

# Digital Visual Effects

## project #2: Image Stitching

組別: 26

組員: R10522606 曾柏翔

R10522611 周昱辰

### 1. 介紹

現今市面上的相機較難做到拍攝大角度的環景照片，若要用一般相機得到該效果就需要透過多張影像拼接而成。若直接將影像拼接，會因為焦距、重疊等誤差使照片看起來仍是多張獨立照片。

因此本專題目的就在於利用多張不同角度的照片，經過影像投影修正照片焦距、找出重疊處特徵點做影像對齊、最後將影像混合。使用該方式就可以修正不同角度照片之間的誤差，做出大角度環景效果。

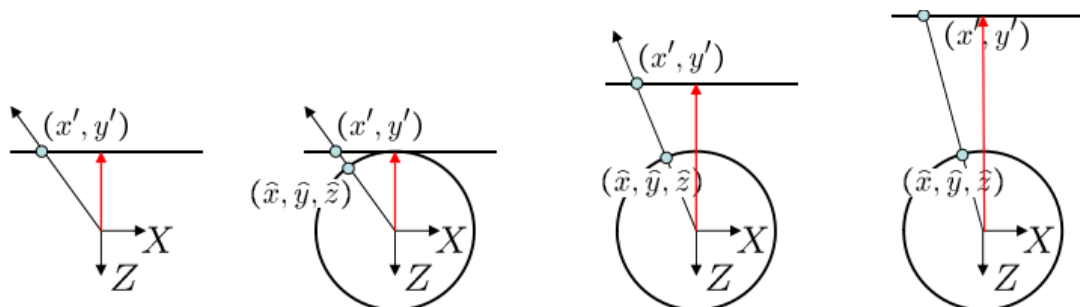
### 2. 方法

我們先收集了 8 張不同角度的照片，為了讓之後在做 image matching 更方便，首先把影像投影到圓柱座標上，使影像間只有平移的關係。接著我們參考 Harris corner detector 的方式完成 Feature Detection，再利用 Non-Maximum Suppression 方法來刪去多餘的特徵點。接著利用 8 x 8 大小的視窗去掃描特徵點附近資訊來做 Feature description。將剛剛得到的 feature points 做距離比對篩選，做出 Feature Matching。再利用 RANSAC 計算平移距離，最終將照片照著距離做貼合來完成全景影像。

#### 2.1. Cylinder Warping

因為全景影像的原理就是利用不同角度拍攝出環場效果，所以在做偵測特徵點或將照片接起來之前，先投影到圓柱體上會較符合其原理，並且可以得到較好的效果。我們投影的方式，是參考老師講義中的公式，先將圖片中心設為原點，再利用以下公式：

$$x' = s\theta = \tan^{-1}\left(\frac{x}{f}\right) \quad ; \quad y' = sh = s \frac{y}{\sqrt{x^2 + f^2}} \quad ; \quad s = f \text{ give less distortion}$$



## 2.2. Feature Detection

這部分我們是利用 Harris corner 來做，其原理是偵測出「兩條邊的交點形成一個角」當作是特徵點。我們是先將所有的 pixel 做一次與二次微分，再利用 Gaussian filter 來求出 corner response。但因為有些 corner 的特徵不是很明顯，所以我們設定一個門檻值，過濾掉低於門檻值的點。將剩下的點對 feature candidates 計算出 local maximum，最後經過 Non-Maximum Suppression 方法來刪去多餘的點，剩下的點即為求得的特徵點。

## 2.3. Feature Description

接著利用上述得到的特徵點，我們必須得到描述特徵點的方法，也才能在不同影像中找尋相似的特徵點。我們採用的方式是利用一個視窗去掃描特徵點，將特徵點附近的資訊作為該點的描述方式。視窗大小則需要適中，太大會導致效果對高頻雜訊有影響，太小則無法詳細描述該點特徵，我們經過測試選擇 8 x 8 為最終視窗大小。

## 2.4. Feature Matching

接著我們有特徵點的描述方式，就可以透過該資訊找尋兩張影像中對應的特徵點。我們是利用 Euclidean distance 找出兩張影像中特徵點距離最小的作為對應關係，而且該距離必須小於次小距離的 0.8 倍，才會被定為正確的對應。

## 2.5. Image Matching

利用前一步驟求出的特徵點對應關係後，計算出兩張影像重疊部分需要多少平移方能接在一起。為了減少 feature mismatching 造成的誤差，我們利用了 RANSAC 技巧：首先隨機選擇數個點去產生 translation model，再用隨機的點來解出 translation model，最後計算多少點落於門檻內並為之打個分數。重複上述步驟，分數最高者即為最合適的 translation model。

## 2.6. Image Stitching

完成上面所有步驟後，我們就將兩兩照片根據 translation model 做合併。並且在所有照片合併完成後，將最終輸出的照片上下弧形部分進行適度的切除，即可得到全景照片。

# 3. 結果討論與比較

## 3.1. 成果展示

首先呈現 2 張老師所提供的 test data，2 組照片皆有調整參數，來配合其焦距、照片尺寸等不同的狀況。可以看出來結果都不錯，因此接著實作我們所拍攝的照片。

Parrington



Csie



接著呈現的是 Visualize feature，如下圖所示為 original image 、feature point 和 feature matching。

Original image (1)



Feature point (1)



Original image (2)



Feature point (2)





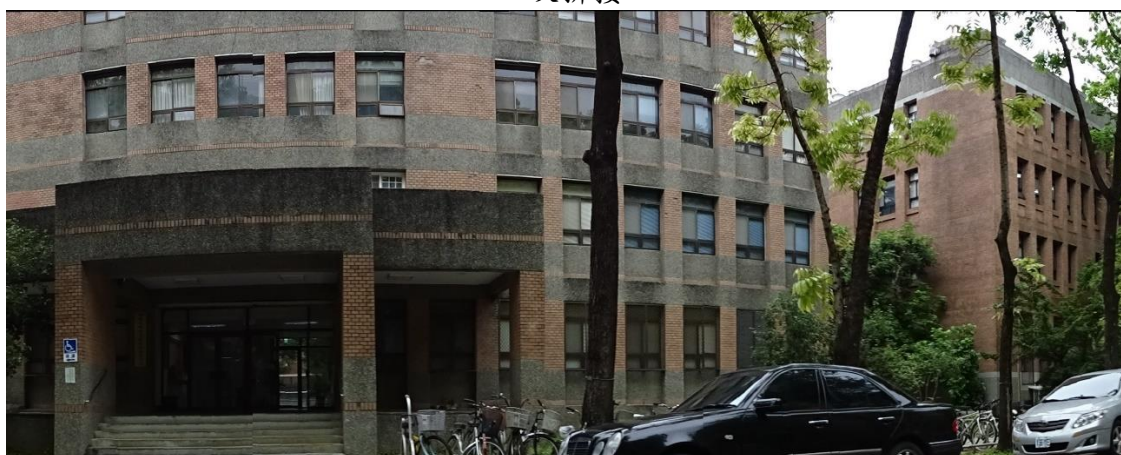
Feature matching



最終呈現自行拍攝的 8 張照片拼接的過程，可以看到隨著照片不斷的兩兩拼接，也使得長寬的比例逐漸改變，下圖所呈現的依序為兩張原始圖片一次拼接、兩次拼接、三次拼接(全景圖)。照片並非所有過程，僅拿部分做為示意。



一次拼接



二次拼接



三次拼接(全景圖)

### 3.2. 問題與困難

在程式撰寫上遇到的問題是數學公式的運算與邏輯的部分，首先是由於這次的圖片拼接牽扯到很多數學的計算，因此在 programming 的時候也需要不斷的拿紙筆進行驗算與確認。再來就是圖片拼接包含了多個步驟，因此在 code 中需要不斷的來回呼叫與承接，因此在初期會不斷地因為邏輯上的錯誤，而導致 Error 的發生。

而完成 code 後，遇到的困難就是實際照片的拍攝，原先使用手持拍攝，會有嚴重的晃動現象，因此改成了腳架拍攝，來改善手振的問題。但隨之而來的問題是光線，由於拍攝的範圍蠻廣的，因此造成每張照片會有亮暗的差異，導致拼接後的圖片，會有明顯的分界線。最終改成到戶外拍攝，因為太陽光的強度比較強，因此每張照片幾乎都是非常明亮，即可改善明顯分界線的問題。

### 4.心得與結論

這次的專題實作了將圖片拼接成一張全景圖，是一項在日常生活中已經十分常見的技術。以前都直接拿手機，就能很快拍出全景圖，甚至有時候還會抱怨上下出現黑色瑕疵很糟糕，但這次實作後才明白其中的困難之處，先不說拼接後的完美程度，光是在時間上就有很明顯的差距。

透過這次的專題以及老師上課的講解，真正的瞭解到 Image Stitching 的原理與步驟，讓我們對於相機的功能背後的原理又有更深的認識。

### 5.參考文獻

- [1] DigiVFX 課程講義
- [2] Szeliski and Shum's SIGGRAPH 1997 paper
- [3] AutoStitch <http://matthewalunbrown.com/autostitch/autostitch.html>
- [4] Panoramic Image Stitching <https://reurl.cc/55kKAz>