

# Digital Visual Effect 2018 - Project2

## Image Stitching

- B03902017 曹峻寧
- B03902066 陳俞安

### 1. Project Description

#### 1.1 Overview

在這份作業我們主要參考 [1] 實作 image stitching，對齊拼接一串輸入的圖片序列以生成一張環景圖 (panorama)。所有的程式都是以 python3 撰寫。

#### 1.2 Data

我們以腳架固定單眼相機 (Canon 550d)，在相同設定水平旋轉拍攝一連串圖片。我們同時也使用手機拍攝了數組測試資料。

### 2. Method

#### 2.1 Cylindrical Warping

為了能使圖片接合的更好，我們要把輸入圖片投影到圓柱座標系。首先，利用 AutoStich [2] 這個軟體預先求得每張圖片的焦距  $f$ 。接著，使用  $x = f * \tan\left(\frac{x'}{f}\right)$ ,  $y = \frac{y' * \sqrt{x^2 + f^2}}{f}$  做 inverse warping 以避免輸出圖片有很多黑洞。

上式是以 (0,0) 為中心做 warping，我們有微調讓他是以圖片中心為 warping 的中心。此外，把 warping 放在 feature detection 和 matching 前，這樣就不用把偵測出的 feature 再轉換一次座標，且在 warping 完的圖片上做 feature detection 看起來跟在原始圖片的效果並沒有什麼差異，所以推測這麼做並不會傷害 performance。

#### 2.2 Features Detection & Matching

我們使用 Harris corner detector 算法來進行 feature extraction，corner response function  $R = \det M - k(\text{trace} M)^2$  的參數  $k$  設為 0.04，被選定為特徵點的 threshold 設定為最大 response 的  $10^{-4}$ ，為了避免 warping 後的邊界被當作特徵點處理，我們限定特徵點不可離邊界過近。此外，也實作 non-maximal suppression 避免取到過於相近的特徵點。

接著，使用特徵點周遭 9x9 像素的灰階值作為 feature descriptor。我們也有嘗試 Multi-Scale Oriented Patches (MSOP)，但發現效果差不多。(沒有使用 Bias/gain normalization 和 Wavelet transform)

最後，將兩張圖的特徵點用 k-d tree 儲存和計算，並要求對每個特徵點而言，與最好的 matching 的距離要是次好的 0.7 倍以下才算是有效的 match，然後取相鄰距離最短點作為 matching pair，最後依照距離由小到大進行排序、儲存 matching pairs 的座標。

## 2.3 Pairwise Alignment

得到 matching pairs 後，就可以透過 Random Sample Consensus (RANSAC) 算出每兩張圖片的相對位移。在 RANSAC 中，我們假設圖片間只有 translation，因此取  $n=1$  (一對 matching pairs 在兩張圖的座標)，觀察發現 inlier 的機率大概是 0.6 (inlier 定義為依照現行參數位移後、座標距離小於 6 的 pairs)，因此理論上要達到  $P \geq 0.99$  需要試驗  $k \geq 6$  次 (但我們最後是取  $k=50$ )。此外，比起隨機 sample，我們認為理想上、特徵距離較小的 matching pairs 應該是比較好的 match，因此從距離最小的 pair 開始 sample。最後直接選 inliers 最多的參數當作 best translation，沒有重新用所有 inliers 再平均一次是因為我們發現這對結果的影響都小於 1，但我們並沒有要做 sub-pixel 的 translation，所以這麼做沒什麼意義。

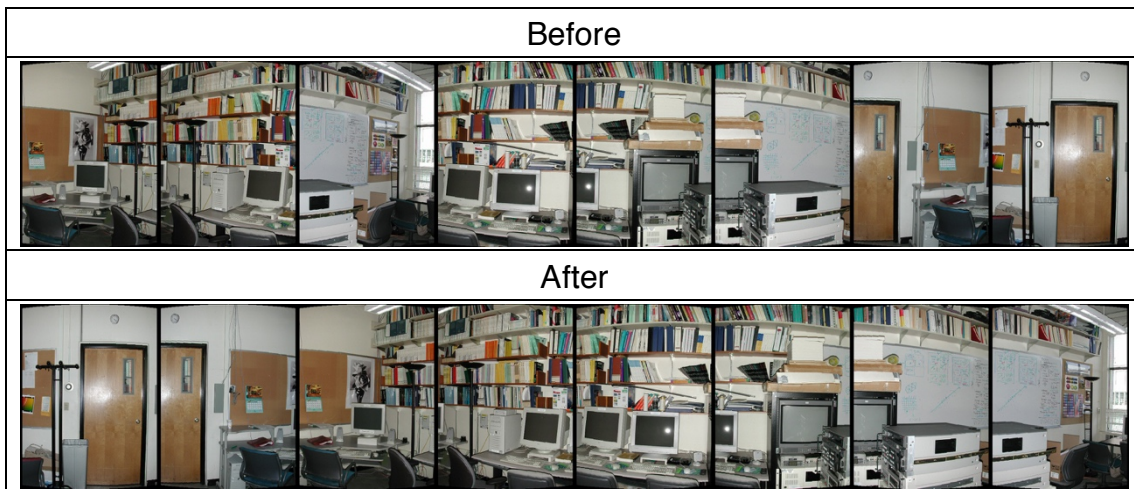
## 2.4 Image Alignment and Blending

在這一步中，我們從第一張圖片開始由左至右拼接出環景圖 (我們預設輸入圖片的序列要是環景圖中由左至右的順序)，每拼完一張圖就需要更新環景圖的座標系以及對正確方向進行擴展。在 overlap area，我們進行 linear blending，並且判斷 pixel 是否為 warping 造成的 margin，如果是就略過它，以免 blending 出的結果有一條一條灰紋。

# 3. Extensions

## 3.1 Images Order Recognizing

在輸入圖片並不是由左至右的順序時，我們實作一種算法去將圖片序列重新排列成正確的順序、就可以用原本的算法生成環景圖。我們將此算法加入在 feature detection 後，使用的做法如下：對於每張圖，計算與其他所有圖的 matching pair，利用 RANSAC 來計算其中 inliers 的個數，當 inlier 個數大於一定值 (100)，則認為此為可能配對，將所有可能配對依 inlier 個數由大排到小，取前二配對並依 translation 值推斷這個兩張圖片的關係 (何者在右邊)，再依這些關係來還原正確圖片順序。下圖是我們將順序被打亂的範例圖片還原出正確順序的例子：



### 3.2 Panorama Refinement

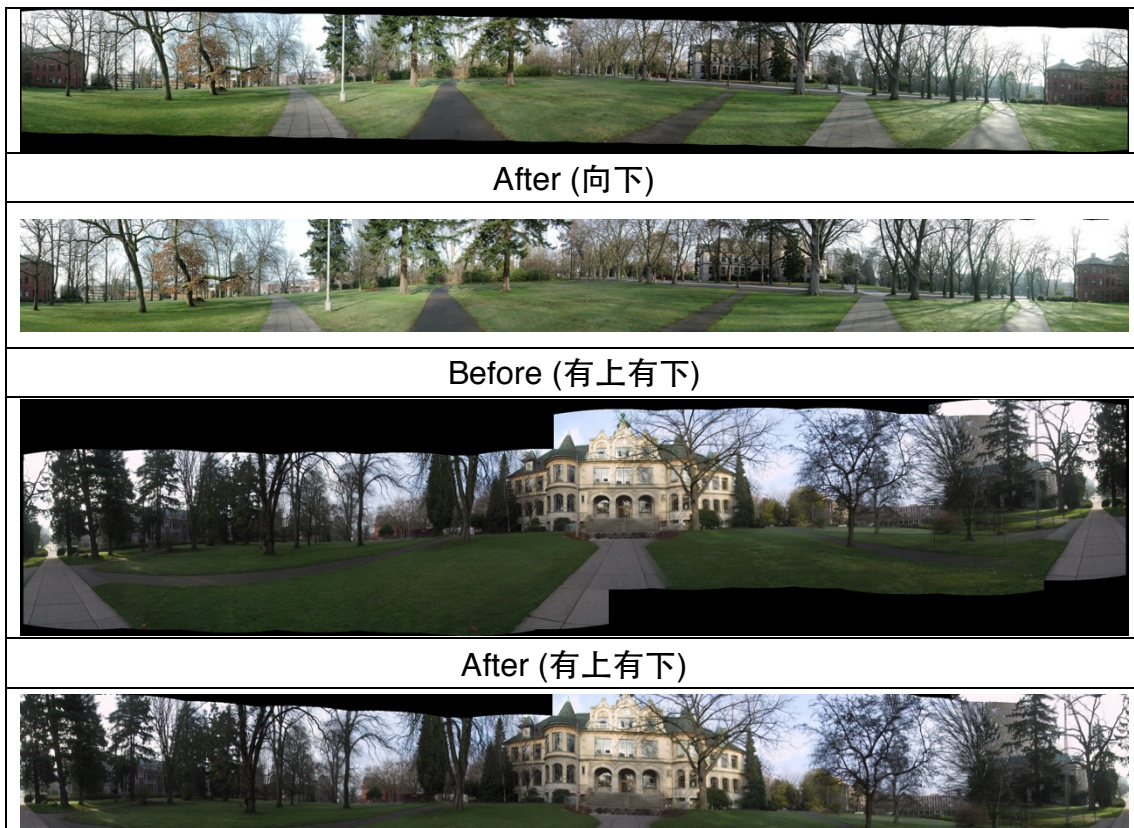
我們也實做了處理 drift 問題的算法，方式是透過生成之環景圖的左上角與右上角座標判斷這張環景圖整體而言是向上還向下 drift，接著透過  $y = y' + mx$  做 inverse warping，其中  $m$  是將環景圖與原圖的高度差除以環景圖寬度得到的斜率。另外有調整一下起始座標、使修正向上 drift 時上方 margin 不會太大。

除了修正 drift 外，我們也做了 adaptive cropping，根據修正完的環景圖的去找它四個角落的座標，依此去猜測 margin 大小進行 crop，會比直接定死一個 cropping 比例來得好很多。

一些結果如下，注意如果圖片 drift 有上有下時，這個做法會表現較差。







可以發現例子 3 從頭尾來看要向下修正許多，使得建築物產生變形，且左半部黑色 margin 往下壓使得 cropping 後的結果還是存在 margin。

## 4. Discussion

### 4.1 Feature Matching

我們發現如果兩張圖重疊區域太少，feature matching 會變得很困難，因此後來拍照時都有特別注意要多留一點 overlap。下圖是一個 matching 較失敗的例子：



### 4.2 Alignment

由於 transform model 假設圖片間只有 translation，如果拍攝的圖片品質不夠好，例如彼此還有旋轉或是視角差異等，就會造成結果不理想。3.2 的表格中的

第一組圖片就是例子，可以發現接合處都會有許多鬼影，但我們的程式在三組範例資料上倒是都有很好的結果，代表 feature matching 與 alignment 本身是沒有問題的。下面是另外兩組例子，意外的是第二組是手持手機拍攝、結果卻比用腳架固定相機的第一組還好，推測第一組除了旋轉外還有仰角的問題、造成圓柱投影不理想（而且可以發現 warping 造成階梯處有不連續的現象）。



## 5.Result

除了前述出現過的 5 組結果外，以下再分別提供一組範例資料的結果以及我們最終拿來當 artifact 的結果，都是我們程式能表現優良的 case：

### 5.1 Example Data



### 5.2 Our Data (selected as the artifact)





## 6. Appendix

### 6.1 Environment Requirement

測試環境為 macOS 與 ubuntu。套件要求為：

- python 3.4+
- OpenCV 3.2+
- numpy 1.13+
- scipy 0.19+
- tqdm 4.21+

### 6.2 File Structure and Execution

繳交的檔案結構如下：

- \*.py
- run.sh
- README.md
- Report.pdf
- data/
  - test\_1/
  - test\_2/
  - test\_3/
  - example\_1/
- artifact.png

執行時僅需 `./run.sh <data_dir>` 即可。

例如，想執行第一組測試資料，則在當前資料夾輸入 `./run.sh ./data/test_1`，運行結果會輸出至至 `./data/test_1/warp/refine_panorama.png`。

## 7. Reference

[1] Brown, Matthew, and David G. Lowe. "Recognizing Panoramas." *ICCV*. Vol. 3. 2003.

[2] <http://matthewalunbrown.com/autostitch/autostitch.html>