

國立台北大學資訊工程學系專題報告

Video Stitching!! 基於 SIFT 的影片縫合系統

專題組員：吳竑璋 郭育丞 楊定澄 黃曉發 陳逸桐

專題編號：PRJ-NTPUCSIE-108-007

執行期間：108 年 09 月至 109 年 05 月

一、摘要

圖像拼接是圖像處理技術的一個重要內容，是一種將多張重疊圖像拼成更大視野的合成圖像技術，有效地解決了圖像視野及解析度的矛盾問題，在醫學圖像處理、航空和軍事領域等方面都有廣泛的應用。

就醫學方面，大幅度的全景圖像能幫助醫生對病灶及其周圍部位的情況進行全面、直觀地觀察。就軍事方面，由於監控器在不同角度及不同條件下，獲取的圖像信息差異較大，對於圖像拼接有更大的需求。

本文研究了圖像拼接的關鍵技術，包括 sift 特徵點檢測、透視彎曲、多波段融合，並使用上述技術完成圖像拼接。而本專題以此為基礎，將影片擷取為一系列的圖像，再透過圖像拼接合成。若是影片中有移動的人物，我們會透過 tensorflow 抓取人物的位置及範圍，再將範圍內的影像利用簡單的 Alpha blend 和成果影像融合，來達到影像拼接及顯示人物移動的軌跡圖。

二、簡介

(一) 研製背景

圖像拼接是指將多張具有重疊區域的圖像進行縫合，構成一張更大視

野的合成圖像。由於近代電子技術和計算機技術的快速發展，影像處理技術也已發展純熟，現在廣泛應用於醫學、軍事、航太技術等各個領域。

(二) 研究目標

我們希望以圖像拼接作為基礎，以影片的形式當作輸入，在經過擷取影片中的圖像後，讓使用者從影片選取出影像，透過圖像拼接合成輸出一合成影像。且若影片中的場景有移動中的人物，我們希望能在縫合後的圖片中，同時顯示不同時間點的人物。

(三) 主要預期效益

當使用者輸入影片後，讓使用者選擇影片拍攝的移動速度，並依選擇的速度設定擷取影像的頻率。完成擷取後，使用者能選取影像，而系統將所選的影像做圖像拼接。

但使用者必須遵守一些規定，像是影像間必須有重疊的場景，及選取的起始影像及結束影像之拍攝角度差需在約 100 度以內。若使用者在規則下選擇好圖片，系統便能將影像做圖像拼接並將所選的影像中的人物均顯示在成果的影像上。

三. 專題進行方式

(一) 人員配置與職責：

本組專題課程由楊定澄學長負責提供其他組員協助，吳竑璋撰寫主程式，郭育丞設計介面與介面程式編寫（及部分主程式），黃曉發與陳逸桐負責部份程式編寫及會議紀錄。

(二) 系統分析與設計摘要：

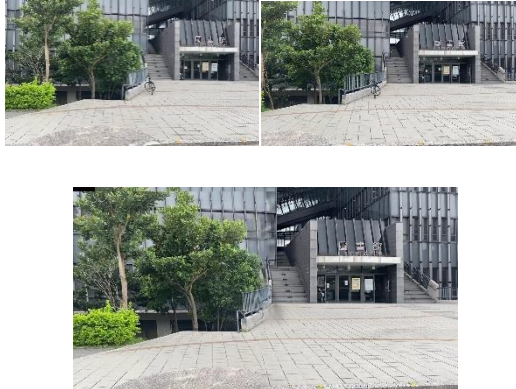


Fig. 1

本系統為基於 SIFT 演算法^[1]的圖片縫合，利用 SIFT 演算法找出特徵點及變換矩陣，再利用透視投影^[5](Warp Perspective)將圖片投影至相對應的位置。而在圖片融合上，我們利用了多波段融合^{[8][9]}(multiband blending)的方式。

在進入主程式前，我們將使用者輸入的影片擷取成圖片，讓使用者選擇在主程式中要使用的圖片。可以選擇圖片及圖片中對象的位置。

下圖為本系統之基礎架構圖：

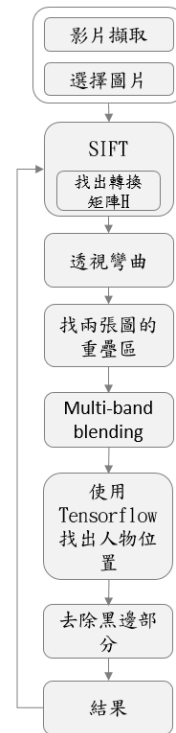


Fig. 2

1. SIFT^[1]:

SIFT(Scale-Invariant Feature Transform)，有四層步驟分別為尺度空間極值偵測、特徵點的篩選、取向分配、及特徵描述。其中以尺度空間極值偵測與特徵描述為其主要特色。

在尺度空間極值偵測中，SIFT 以 Difference of Gaussian 做 LoG 的近似。在正確的數值設定下，DoG 可以做到接近零誤差且大幅降低了運算成本。(Fig.3)

求出 DoG 後，將相鄰的三張分為一組，各組將中間的 DoG 的每一個像素和其相鄰的 8 個像素及上下兩張 DoG 相對應的 9 個像素相比，若為極大或極小值，此像素就被認為一 potential keypoint。(Fig.4)

這一階段的運算讓 SIFT 能偵測到在不同尺度空間都具代表性的特徵點，使其特徵點不受縮放大小的影響 (Scale-invariant)。

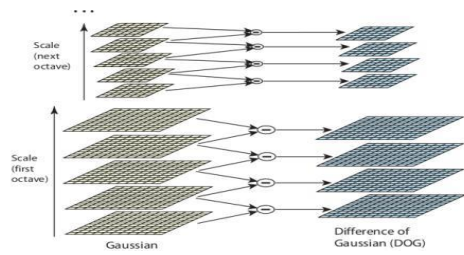


Fig. 3

圖片來源: [1]

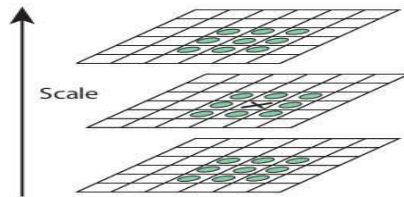


Fig. 4

圖片來源: [1]

由於keypoint 數量龐大，我們必須篩選出足夠強度的特徵點。使用泰勒級數展開，若強度小於 0.03 則放棄該 keypoint。並使用 2x2 Hessian 矩陣計算曲率，若大於 10 則視為 edge 也一樣必須捨棄。

另一個 SIFT 的主要特色為特徵點描述，SIFT 利用計算取向角度及梯度大小來描述特徵點。透過特徵點描述，SIFT 特徵點能不受圖像旋轉影響 (Rotation invariant)。

最後就是特徵點匹配，使用 KNN(K 近鄰演算法)並將 k 值設為 2。並做 ratio test，若第一近距離與第二近距離之比大於 0.8 則不被採用。



Fig. 5

圖片來源: <https://www.cnblogs.com/Jessica->

<jie/p/8622449.html>

2. 透視彎曲^[5]:

透視彎曲是將成像投影到一個新的平面，也稱作投影映射。如下圖，通過透視彎曲將 ABC 變換到 A'B'C'

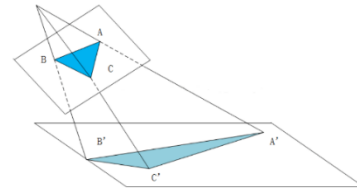


Fig. 6

Opencv 中透視彎曲的公式為：

$$\text{dst}(x, y) = \text{src} \left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}} \right)$$

src - 輸入的影像

dst - 輸出的影像

M - 轉換矩陣

3. 多波段融合^{[8][9]}:

多波段融合是將圖像分解為不同頻率的圖像的疊加，在不同的頻率上，應該使用不同的權重(遮罩)來進行融合，在低頻部分應該使用波長較寬的加權信號，而在高頻部分應該使用較窄的加權信號。

算法如下:

- (1) 計算輸入圖像的高斯金字塔。
- (2) 將(1)中得到的高斯金字塔，依據下列方法，計算得出圖像的拉普拉斯金字塔。

$$L_i = G_i - \text{expand}(G_{i+1})$$

(註:本系統利用 cv.resize()及線性差值法實踐 expand()方法)

- (3) 將處於同層級的拉普拉斯金字塔進行 Alpha blending(線性融合)。
- (4) 將高層的拉普拉斯金字塔依次擴

展直至和原圖相同尺寸(scale)。
(5)將(4)中得到的圖像疊加，則得到最終的輸出圖像。

4. Tensorflow

為了找出人物的座標，我們使用 google 釋出的 Tensorflow 版本的 Object detection API，裡面提供多種模型供使用者選擇，使用者可評估速度/準確率的要求來選擇合適的模型，而本專題為了追求快速，使用了 SSD 模型。



Fig. 7

5. 判斷範圍

(1)找重疊區域:

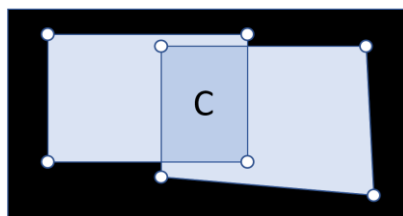


Fig. 8

在找重疊區域時，我們將兩張圖的八個頂點座標取出，分別排序 x 座標及 y 座標。而重疊的區域經過實驗統計後，我們發現可以取 x, y 排序後的中間兩個值所圍成的區域(C)。

(2)找去除黑邊區域:

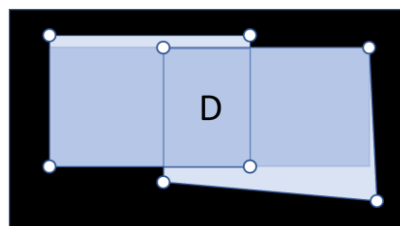


Fig. 9

去除黑邊時，為了讓黑邊不影響連續縫合的效果。我們只保留不包含黑邊的最大矩形。而我們同樣利用兩張圖的八個頂點座標排序後，x 座標(左右)取排序後的第二及倒數第二的值，y 座標(上下)則取排序後的第四及第五的值所圍成的區域。

(三) 主要困難與解決之道:

1. 縫合多張圖片時的黑邊影響:

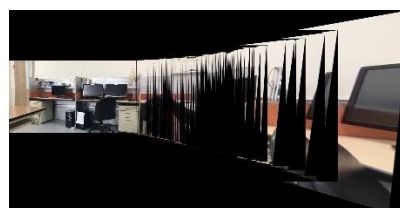


Fig. 10

在縫合多張圖片時，如果在每一輪的結果中都有多餘的黑邊，就會大幅的影響連續縫合圖片的結果。為了解決這一個問題我們需要捨棄部分的圖片內容，只保留去除黑邊部分的最大矩形。

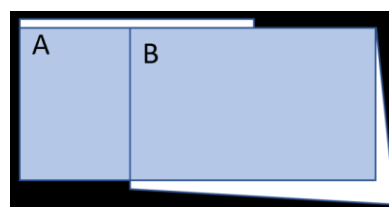


Fig. 11

如上圖所示，我們只保留藍色區域中的圖片當作下一輪的輸入，將其餘白色的部分捨棄，讓連續縫合圖片時不會在圖片中出現黑邊。



Fig. 12

2. 加入多波段融合後出現雜訊:



Fig. 13

使用多波段融合的方法會將 Laplacian pyramid 中低頻至高頻的六層相加，因此在部分的像素中會因數值超出範圍而產生雜訊(相加後若某像素值超出 RGB 值的範圍 0~255，會自動將值以環型的方式轉換)(如下圖)。



Fig. 14



Fig. 15

3. 顯示重疊區域的對象

在縫合圖片時，我們會將經透視彎曲轉換的圖片作為基底，再將基準圖疊置在上方，因此下方圖片上的對象就無法顯示。

起初我們想利用調整多波段融合中的遮罩(權重)，但我們無法確定所要顯示的對象(人物)所在的位置，而無法確定遮罩的位置與大小需如何調整。而若為了顯示所有可能位置的人物而將遮罩範圍放大，則會造成 Ghosting(物體及場景變為半透明)。

最後我們決定使用 google 釋出的 Tensorflow 版本的 Object detection API，利用 tensorflow 中的模型做圖像辨識，將偵測到的位置及範圍內的影像利用簡單的線性融合的方式和結果圖融合。雖然解決方法和起初努力的方向不同，但透過這個方式我們也做到了設定的目標。

4. 物體角度變化:

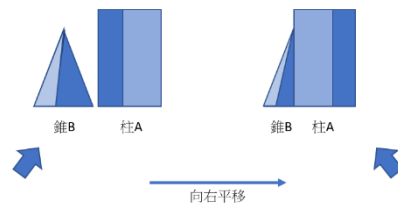


Fig 1

我們在實驗過程中，試過了許多鏡頭平移拍攝的影片及測資，然而使用 SIFT 的結果都不理想。我們經過多次的測驗及討論後才發現是因為使用平移拍攝的方式，會使畫面中的物體(場景)的相對位置及角度發生大幅的變化，雖然 SIFT 對於 3D 的視角變化有一定的容忍度，但仍無法承受較長距離平移造成的視角變化及相對位置的改變。因此我們決定將程式的目標鎖定為旋轉拍攝的影片。

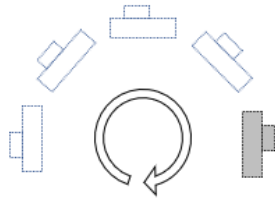


Fig 2

三、 成品

四、 結語及展望

本次專題，我們實作出一套圖像拼接軟體。利用 SIFT、WarpPerspective、Multiband 等方法，實現拼接以及融合的目的。

為了重複顯示出影片中的人物，使用了 tensorflow 來辨識人物，雖然我們利用 SSD 模型達成了和目標相似的結果，但仍是一種替代方案。未來期待能找到合適的融合演算法，讓我們能在融合的過程中做到目標物的顯示，提供使用者快速實用的應用程式。

另一方面，我們也希望能將這次專題的系統利用網頁的模式，提供使用者一個方便使用的介面。

五、 銘謝

感謝教授這一年來的悉心指導，引導我們方向，協助我們分工，提供資源和建議，一步步帶領我們，幫助我們完成專題。

感謝學長在上學期專題上的帶領，給我們解釋本專題的原理和程式上的指導。

最後，感謝所有為專題付出過努力的人，才能讓我們有今天成果。

六、 參考文獻

[1] Alexander Mordvintsev & Abid

K. “Introduction to SIFT (Scale-Invariant Feature Transform) ”, 2013

[2] 閒着學點甚麼, “opencv-使用 SIFT or SURF 實現圖像拼接”, Aug, 2019

[3] Vagdevi Kommineni, “Image Stitching Using OpenCV”, Oct 12, 2018

[4] 程式前沿, “opencv-python 提取 sift 特徵並匹配”, Jul 27, 2018

[5] opencv devteam, “warpPerspective” retrieved from

https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html (Apr, 2014)

[6] Matthew Brown, “AutoStitch: a new dimension in automatic image stitching”, Jul, 2018

[7] on2way, “Python 下 opencv 使用笔记 (三) (图像的几何变换)”, Jul, 2015

[8] smallflyingpig, “图像拼接——多波段融合(multiblend)” retrieved from

<https://blog.csdn.net/smallflyingpig/article/details/61200497> (Mar, 2017)

[9] Zhe Zhu, Songhai Zhang, Jiaming Lu, Ralph R Martin, Minxuan Wang, Hantao Liu and Shimin Hu. A Comparative Study of Algorithms for Realtime Panoramic Video Blending. arXiv:1606.00103v2 [cs.CV] 24 Dec 2016