

Image Stitching

Cylindrical Projection

首先透過軟體 Autostitch 取得相片之間的 focal length，再將相片從平面投影到圓柱面。

Feature Detection-SIFT

實作參考 SIFT-MATLAB，並將其修改為 Python code 並對其做部分優化

Difference of Gaussian

先建出 Layers of gaussian images 再將兩兩相減會得到 DoG



Extremum

再來則是要比較上層 octave 的 9 個、當前 octave 的 8 個像速點、下層 octave 的 9 個，若此點都比其他 26 個點大或是小就可以進入下一個步驟

Keypoint Positioning

這個步驟是要確定 Extremum 的尺度、位置，並且除去低對比以及邊緣響應的 keypoint

Local image descriptor

為了實現在任何旋轉都可以成功 match，需要透過 gradient 以及 Gaussian images 建出 histogram，建出之後給予適當的平滑比較好，一個 keypoint 可能會

有多個方向，只要大於一定的值就把它當作新的 **keypoint** 可以增加準確性，**descriptor** 用來描述該 **keypoint** 與鄰居之間的關係，最後會生成 128 維的 **descriptor**

Feature Matching

直接使用 Brute-force matching，Feature 找最靠近的跟次靠近的 Features，當最靠近與次靠近的比例小於 0.5 則就是好的 match，並且會做一個 Cross Check 來增加 Feature matching 的可靠性。

RANSAC and Homograph

透過隨機的大量取樣可以找到一條最適合此樣本的 Homograph

Warp Perspective

透過剛剛算出來的 Homograph 可以找到圖片需要的位移，最後再將其利用 transformation 位移至合適的地方。

Blending and stitching

為了讓重疊的地方看起來自然一點，這裡使用了 Liner Blending，依據像素點的位置給予 0~1 之間的權重，最後將兩張圖片 stitching 起來

What did I learn from the project?

實務方面我覺得我的程式邏輯跟理論知識的部分都有不錯的提升，雖然老師在很早之前就說過 SIFT 的實作十分複雜，但我還是不信邪的用 SIFT 寫，我真的寫到心態快要炸裂了，但最後在參考了一翻資料及一些程式之後最後還是有成功

寫了出來，好在最後效果還算滿意，但也可能是因為我用 SIFT 寫所以真的花了非常多時間才實作完成，可能是 P1 的三、四倍，整體 Project 的難度也上升了好幾倍，SIFT 真的是我寫過最難的演算法。

心理方面某種程度上來說也培養我的心態，在面對複雜的演算法的時候要怎麼去處理它，回到最後還是要先把理論搞懂，但我覺得有時候看理論還是很難懂，反而有時候看人家程式碼就能很快知道它在幹嘛，但看別人的程式碼也是一件蠻痛苦的事情，總體來說學習到了不少就是了

References

- <https://github.com/fredzzhang/SIFT-MATLAB> (SIFT-MATLAB)
- https://sbme-tutorials.github.io/2018/cv/notes/9_week9.html
- <https://www.csie.ntu.edu.tw/~cyf/courses/vfx/papers/Brown2003RP.pdf>