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

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/~cyy/courses/vfx/papers/Brown2003RP.pdf>