

instruction to run

產生 parrington 的 RANSAC(14):

```
python final.py parrington ransac
```

產生 parrington 的 shift(18):

```
python final.py parrington
```

algorithm

1. Feature detection

Multi-Scale Harris corner detector

$$\mathbf{H}_l(x, y) = \nabla_{\sigma_d} P_l(x, y) \nabla_{\sigma_d} P_l(x, y)^T * g_{\sigma_i}(x, y)$$

$$\nabla_{\sigma} f(x, y) \triangleq \nabla f(x, y) * g_{\sigma}(x, y)$$

$$f_{HM}(x, y) = \frac{\det \mathbf{H}_l(x, y)}{\text{tr} \mathbf{H}_l(x, y)} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

對 img 做 gaussian 並求 gradient，找出每個點的 f，保留 f 值為 10 以上的點當做 candidate。
共做 4 層。

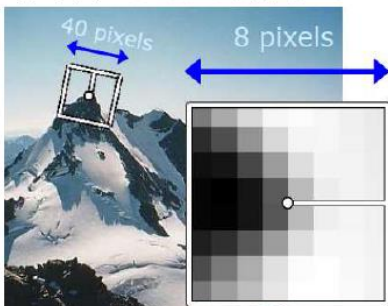
Non-maximal suppression

對所有 candidate，找出 f * 0.9 仍大於自己的點的最近距離。
將距離由短到長排列，前 point_num 個點做為 feature。

2. Find descriptors

MSOP

取每個 feature 周圍 40*40 的點，每 5*5 個做一個取樣，得到一 8*8 大小的 descriptor



3. Cylindrical projection

將每張圖利用已知的 focal 做 projection。

$$x' = s\theta = s \tan^{-1} \frac{x}{f}$$
$$y' = sh = s \frac{y}{\sqrt{x^2 + f^2}}$$



4. Feature matching

Exhaustive search

對每個 feature，找另一張圖中和其 descriptor 均方差最小(d1)和第二小(d2)的 feature，若 d1/d2 小於一定比例且 d1 不大於所有 feature 中 d1 最小值的一定比例時，則 match 產生。

5. Image matching

這次使用了兩種方式來決定圖片的位移。

shift

每張圖片不做變形，只做x, y的位移。

取每一對 match 的位移當作圖片位移，看在哪個位移下其他 match 的誤差總和為最小，

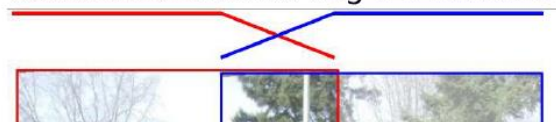
RANSAC

在 max_iteration 的次數下 隨機選 4 個 match 並形成一轉換矩陣 (homo)，以在該 homo 轉換後在一定誤差範圍內的其他 match 數量當作標準，找出最佳homo，並將整張圖片用 homo 轉移。

6. Blending

linear blending

使用線性方式做兩張 img 重疊處的blending



Discussion

以下是 parrington 經過 cylindrical projection 後用 shift 做 image matching 的結果。(18張imgs結合)



以下是 parrington 不經過 cylindrical projection 用 RASAC 做 image matching 的結果。(14張imgs結合)

