

2017 CV Homework02 Report

資科碩一 / 0556087 / 謝禎原

Outline :

- 程式流程
- 執行結果

程式流程 :

Step 1. 圖片讀取 , SIFT

把 Target 也視為其中一張拼圖, 一併和 Sample 計算 Key points 和 descriptors .

Step 2. 2 – NN (找出各 Kp 的 matches)

針對各個拼圖以及 Target Key points (K_i), 計算和所有 Sample 中 Key points (K'_j) descriptors 的歐式距離 , 取最短的兩個 Key points 作為該 Key points (K_i)的 matches .

$K_i \rightarrow$ (距離 K_i 最近的 K'_j , 距離 K_i 第二近的 K'_j)

Step 3. RANSAC (找出最佳的映射矩陣)

- 終止條件 (擇一即可) :
 - 該 Matrix 的 inlier Ratio > 0.8 .
 - 若跑了超過 2000 回, 則選擇 inlier Ratio 最高的 Matrix .
- Inlier ?
 - 當由該 Matrix 映射出來的點座標和由 KNN 找出來的 matches 座標其中一個的「座標距離」小於 5 即可(有兩個)算是 inlier .
- Output :
 - Puzzle 們到 Sample 的映射矩陣 (H_i) .
 - Target 到 Sample 的映射矩陣 (H') .

Step 4. Wrapping (把 Puzzle 貼成和 target 大小一樣的圖上)

先算出各個 puzzles 最終要使用的映射矩陣(F_i) (puzzle 到 target), 也就是 $H'.inv() * H_i$ (因為 H' 是 target 到 sample , 因此反過來需要取 inverse !).

兩種方法 :

■ Forward wrapping :

從 Puzzle 出發去算要貼在 Result 的哪個座標 .

發現最後會有某些座標沒有 RGB 值的問題, 因此
最後算出對應座標填顏色時不是只填該 Pixel 的顏色, 而是該 Pixel 的九宮格一併填那個顏色 (但最後發現好像還是有黑洞).

■ Backward wrapping :

從 Result 出發去找(注意, 是乘上 `Fi.inv()`)每個點該填什麼 RGB 值 .

執行結果 :

Backward wrapping :



Forward wrapping :

