

影像拼接作業

姓名: 李珮慈

學號: 00657124

日期: 2020/05/20

● 方法

1. 拼接 dataset1.mp4

我讓圖片從中間往左右兩側拼接，方式為先拼完右邊，再拼左邊。

因為要從中間開始，所以 frame_num 要從 frame_total 的一半開始算。

剛開始找中心點為 result.shape 的一半，視覺上似乎是在中心點了，但實際上不是，那是 imshow 視窗的中心點。應該是要找

result.shape[1]-frame.shape[1]的一半才是拼接圖片正確的中心點。

而後方-190 是在做一點點誤差修正，讓兩邊左右拼接的長度看起來比較平均。

Frame_num+=1 表示該 while 是向右拼

Frame_num-=1 表示該 while 是向左拼

```
if frame_num == total_frame // 2:
    T = np.eye(3)
    T[0,2] = (result.shape[1]-frame2.shape[1]) / 2 - 190 ##取中間值再做點誤差修正
    T[1,2] = 0
    result = cv2.warpPerspective(frame2,T,(result.shape[1],result.shape[0])).astype(np.float)
    t_count= cv2.warpPerspective(ones,T,(result.shape[1],result.shape[0])).astype(np.float)
    count += t_count.astype(np.float)
    disp = result.copy()
    cv2.imshow('stitched image',disp.astype(np.uint8))

    frame1 = frame2
    kp1 = kp2
    dt1 = dt2
```

2. 拼接 dataset1 之圖片

使用老師的範例，將輸入改為 dataset1 的圖片，由右往左開始拼接。由於照片只有 81 張，因此將原本的 total_frame 改為 81，而拼接部分之 code 也只跑 81 次，每一次都去讀取 dataset 內的照片編號，再做拼接動作。

由於此資料集只有 81 張照片，比原本影片少，所以調整了輸出的 width，從原本的 width/4*3 改為 width/4*2，多出來的黑色的部分就消失了。

```
total_frame = 81
```

```
while i in range(81):
    frame2=cv2.imread(".\\dataset1\\dataset_1_%d.jpg"%(i))
    print(".\\dataset1\\dataset_1_%d.jpg"%(i))
    frame2 = cv2.resize(frame2,(frame2.shape[1]//4,frame2.shape[0]//4))
```

● 結果

1. 影片拼接

https://youtu.be/2Chsp_IFarI

2. 圖片拼接

<https://youtu.be/eTcYc5s-aZw>

3. Github

https://github.com/muyumiyal201/stitch_image

● 結論

若使用拼接的方式，越到後面拼出來的圖片就越模糊，甚至被拉寬。從中間往兩側拼接的話，兩邊失真的效果就會下降。比由右到左或由左到右來的好看。

● 參考

老師的範例 code