

Lab03

Background subtraction

Threshold

Connected component

輸入一段影片



輸出成果

Background subtraction



Threshold



Connected component



框出前景



0. 讀取影片

影片(video)是由連續的影像(image)組成，組成影片的影像稱為影格(frame)，影片播放時會不斷呈現新的影格，影格間的時間稱作更新頻率(frame rate)。

- `cap = cv2.VideoCapture(filename)`

- `cap.isOpened()` → 會用 while

- 檢查影片是否被成功讀取

- `ret, frame = capture.read()`

- 不斷讀取來源影格，把資訊寫進frame

- `cv2.imshow("frame", frame)`

- 顯示影片

- `cv2.waitKey(33)` → 可用可不用

- 等待幾毫秒再讀取下一幀

while 那裡面

1. Background Subtraction

- `backSub = cv2.createBackgroundSubtractorMOG2()` → 直接 call python 的 MOG2
 - 創造一個 BackgroundSubtractor
- `fgmask = backSub.apply(frame)`
 - 用在 frame 上面

apply 在每個 frame 上面,
就會把前景 subtract 出來

前景 → 255

背景 → 127

背景 → 0

不需要這個

基本上顏色不太會變, 但亮度比背景暗

2. Threshold

- `shadowval = backSub.getShadowValue()`
 - 找出shadow的值 · default = 127 · 0: background · 255: foreground
- `ret, nmask = cv2.threshold(fgmask, shadowval, 255, cv2.THRESH_BINARY)`
 - 輸出圖片 輸入圖片 threshold最小值 最大值 method

3. Connected Component

- Seed Filling Algorithm: 要用 two pass 的方式

1. 掃描圖像，找到強度為1的點
2. 將此點的強度設為一個label值
3. 向四周擴散找到相連且強度為1的點，並將這些點的強度設為同一label值
4. 重複上述步驟直到沒有強度為1的點

- 同一區域的點會有相同的label值

0	1	1	0	1	0	0
0	1	1	0	1	0	1
1	1	1	0	1	0	1
0	0	0	0	1	1	1
0	1	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	0	0	0

Original Image

0	1	1	0	2	0	0
0	1	1	0	2	0	2
1	1	1	0	2	0	2
0	0	0	0	2	2	2
0	3	0	0	0	0	0
0	3	3	3	3	3	0
0	3	3	3	0	0	0

Connected Components

3. Connected Component

- 找connected component時，要同時計算相連區域面積
- 若相連區域面積 $> T$ ，找出相連區域最外圍的四個邊

NO可能輸出報訊

計算面積，太大的可能
就是報訊

4. 框出物體

- opencv畫矩形

- `cv2.rectangle(影像, (頂點座標), (對向頂點座標), 顏色, 線條寬度)`

↑ 把人的座標打出來



透過跑一個迴圈，知道 label 的 width 和 height



就可畫出矩形

4. 框出物體

- opencv畫矩形
 - `cv2.rectangle(影像, (頂點座標), (對向頂點座標), 顏色, 線條寬度)`