

高等電腦視覺 HW#4

學號：107318050

姓名：黃佳婷

本次作業為實現 Tracking，使用 C++和 opencv3.4 完成這次的作業，程式執行流程就是按 q 框出倉鼠區域後觀看相關資訊及結果，案 c 則下跳到下一幀畫面，直接執行 exe 檔框完 ROI 跑完影片後就會看到存下來的影片了，報告裡的圖片都是結果影片中的截圖，因為影片太大，所以檔案裡面都不附影片，要執行 exe 檔的話要把影片放到同一個資料夾。

#Program_4-1.avi

Opencv3.4 可以用的內建 tracking 演算法有 KCF、MIL、TLD、MedianFlow、Boosting、MOSSE，"1.avi"基本上用哪個演算法執行都沒有太大的問題，tracking 效果都還算不錯，在這裡我是套 **KCF**，基本上都有順利追蹤到。

以下是每 30 個 frame 擷取一次的結果圖，由以上結果可以看到 30 張有成功抓取的為 30 張,故準確度為 100%





#Program_4-2.avi

BOOSTING Tracker：最後決定使用的演算法，移動到背景之後有機率定在背景，有經過之後又接回來了，跑回右上角之後，框框有機率會一直定格在右上角，或是變成在抓手。遇到背景比較複雜的部份效果不好，如果目標物沒有在經過框框的話可能就接不回來了，執行速度稍微比較慢，報告截圖是我最好的一次結果。

MIL Tracker：速度比 Boosting 快，但有一半會跑掉。

KCF Tracker：一開始經過喬巴後就直接定在背景沒再回來過。

MedianFlow Tracker：不行，後面會跑掉。

TLD Tracker：正面大致都抓得到，轉到背面就抓不到了，正面有時候也抓不到。

MOSSE Tracker：到後面都在抓手不是在抓倉鼠。

以下是每 30 個 frame 擷取一次的結果圖，由以上結果可以看到 31 張有成功抓取的大約 10、11 張,故準確度為 32.25%





心得：

這次做 inverse warping 其實卡了很久，主要是卡在寫入 BMP 檔格式的時候一直出問題，只是卡了很久的輸出結果最後也是有問題的，讓我覺得相當絕望，應該是中間有公式代錯但是我沒有發現到。因為第一題的教訓，第二題程式整個重新寫，改成讀寫都用 raw 檔處理，整體輕鬆愜意了許多，只是變成察看結果需要用 XnView 不太方便，下次有機會再改成輸出可以直接查看的格式，第二題主要的障礙是卡在看 paper 看了很久，把所有的變數搞清楚之後再

實現成為程式，這之間也花了相當久的時間，想出這個拼接影像方法的人真的相當了不起。