Advanced Computer Vision _ Final Project

Jia-Ting Huang 107318050

Electrical Engineering, National Taipei University of Technology

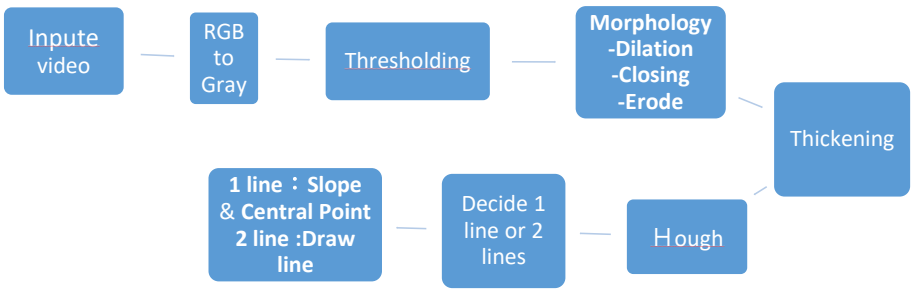
Taipei, Taiwan
t107318050@ntut.org.tw

I. INTRODUCTION

此期末報告之目標為找出影片中的目標線段畫出中線，並與每部影片的 ground truth 做距離比對（Error），計算出每個 frame 的錯誤率以及執行時間。本篇報告所使用編譯環境是 Visual Studio 2017，以及 OpenCV 3.4.0。

II. Methods of encoding

A.Flow Char



B Image pre-processing

在擷取影像時，由於環境的影響，譬如:光源干擾、震動、…等等，導致影像品質不佳，或是影像中有我們不需要的資訊，這時就必須透過一些方法對影像做合適的處理，達到去除雜訊、增強影像…等的效果，藉此改善影像的品質，將我們需要的資訊凸顯出來，使影像可以用於後續的影像分析，這個過程

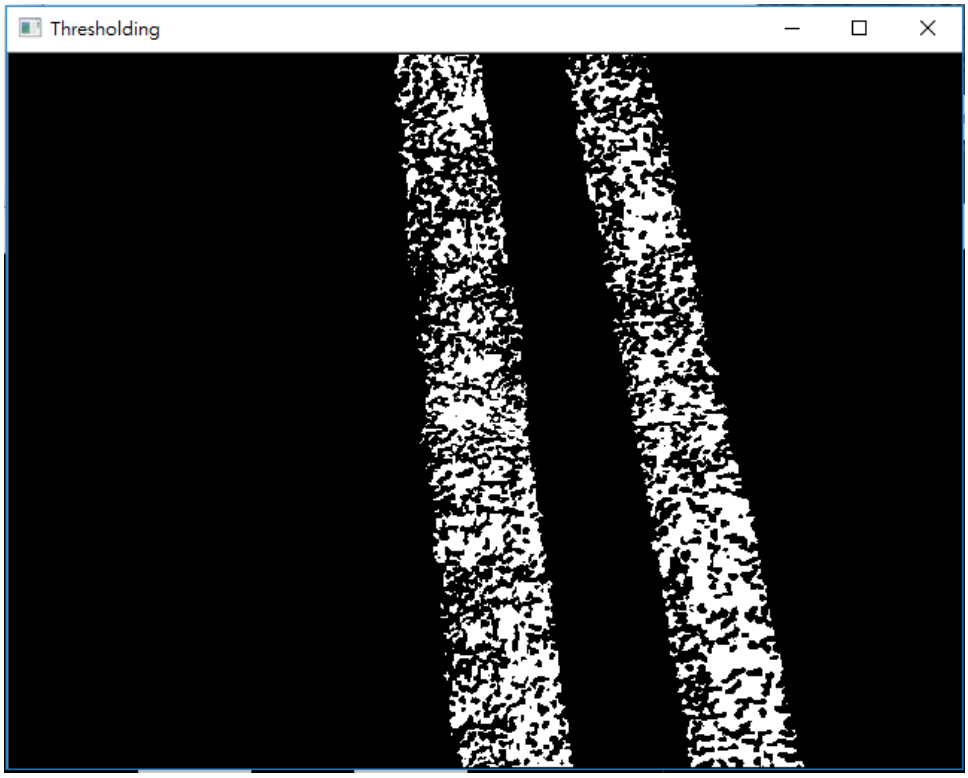
稱之為前處理。

1. RGB to Gray

先將採色影項轉為灰階，減少處理的資訊量。

2. Thresholding

將上一步得到的灰階圖做二值化(閥值為 145)，介此過濾大部份之雜訊，留下虛要的白線部份。



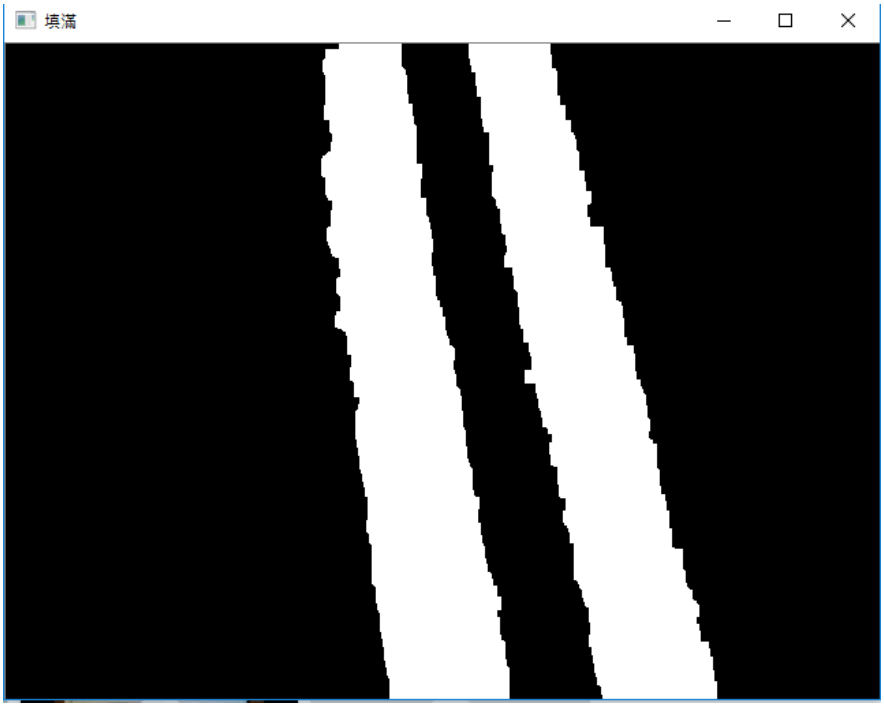
3. Morphology

利用型態學將路線缺失的部份填補回來。

膨脹(Dilation)：將透過膨脹將整體讓沒有連接的部份變粗，讓破碎的線段/塊更靠近，以達到填補的效果。

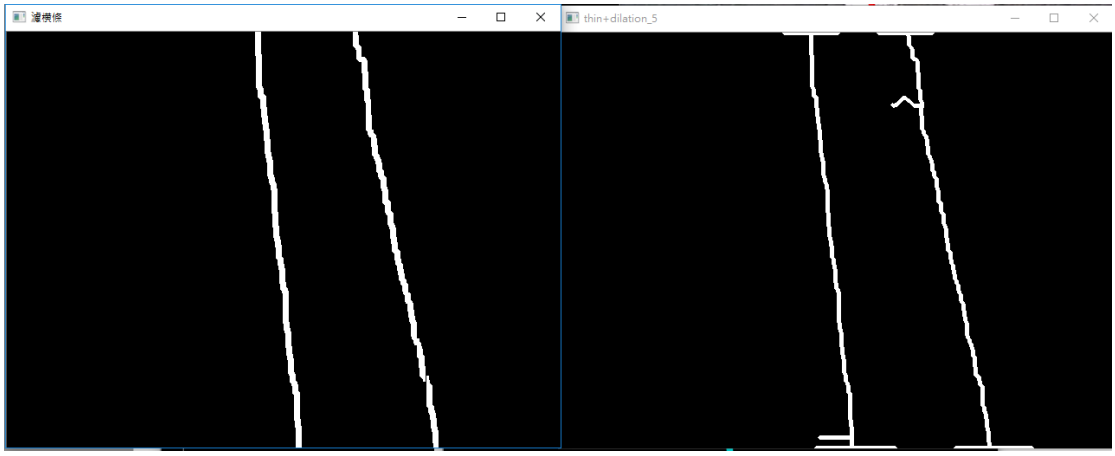
閉合(Closing)：將路線裡的小孔閉合起來，以達到填補之效果。

侵蝕(Erode)：將剛剛膨脹的部份做侵蝕，可達到去雜訊以及將路線的寬度復原至接近原來的形狀。



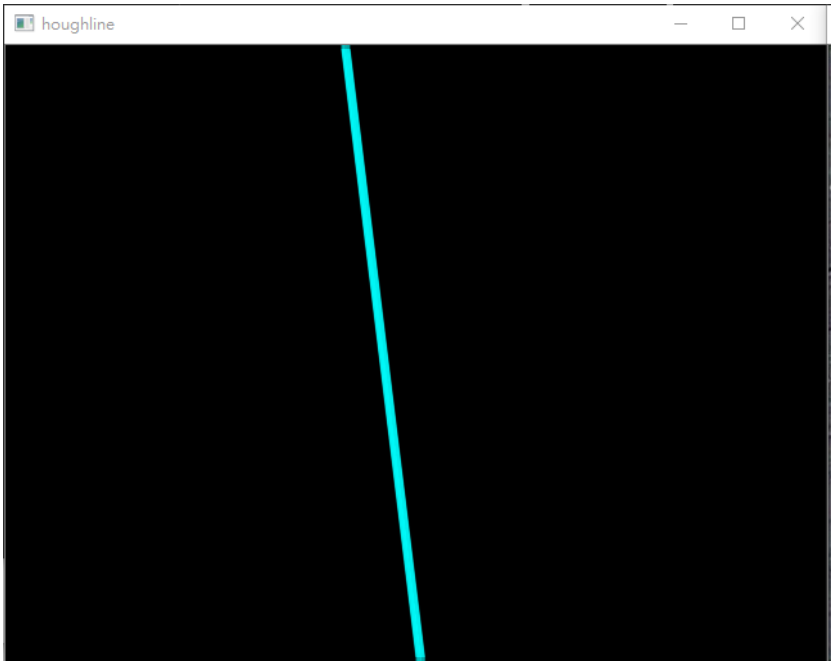
4. Thickening

將剛剛做完形態學之路線做細化，得到最中央的那條線。



5. Hough Transform (Line)

Lines 是求得目標線的上、下頂點座標。結果可能是由多個線段合併，故利用已知兩點反推出直線方程式再畫出延伸到影像頂點之直線。其中參數值 160 是一個閥值，要超過此閥值才會認定是一條直線。參數 240 是認定之線段長超過 240（影像長的一半）才會儲存資訊，確保不必要的小線段顯示(如下圖中左邊兩條線段不會被畫出)。最後一個參數 20 為線段之間的距離超過 20 以上為兩個不同之線段。(結果如下圖所示)



6. 決定線之數量

當霍夫線偵測到兩條路線時會產生較多的霍夫線，設一個閥值判斷為一條路線或兩條路線。

7. 去除非目標線段

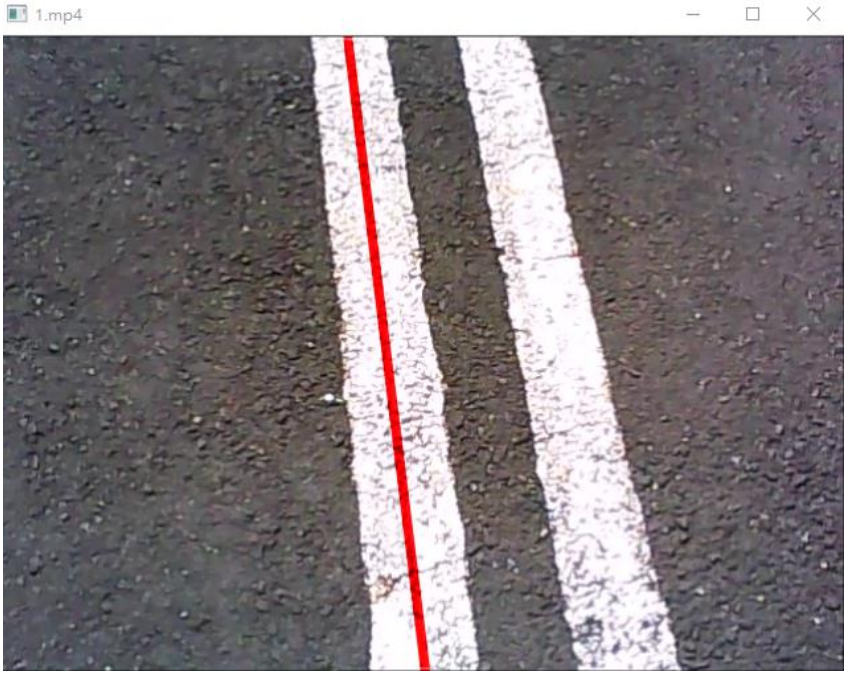
在搜尋目標線時，有時會同時偵測到其它線段，所以利用兩種特徵做篩選。分別是使用第一個 frame 線段的斜率以及中心點做為基準點，並在之後的每個 frame 更新基準點的資訊，讓基準點保持在最新的狀態(結果如下圖所示)。故此方法十分仰賴第一個 frame 的準確率，假使第一個 frame 抓取錯誤，則之後的結果可能都會是錯的。

(1) Slope

因為函式 HoughLinesP 的輸出值為線的兩端點，所以利用這兩點回推出線段方程式，求得斜率並延長線段至整個 frame。再將求得之斜率與第一個 frame 的斜率做相減，如果差值大於 0.3 則移除此條線段。

(2) Central Point

得之線段兩端點的 x 座標點，將他們相加除以二後與第一個 frame 的中心點做比較。設之條件：彼此差值大於 25 以上，則將此線段去除。



III. RESULT OF THE EXPERIMENTS

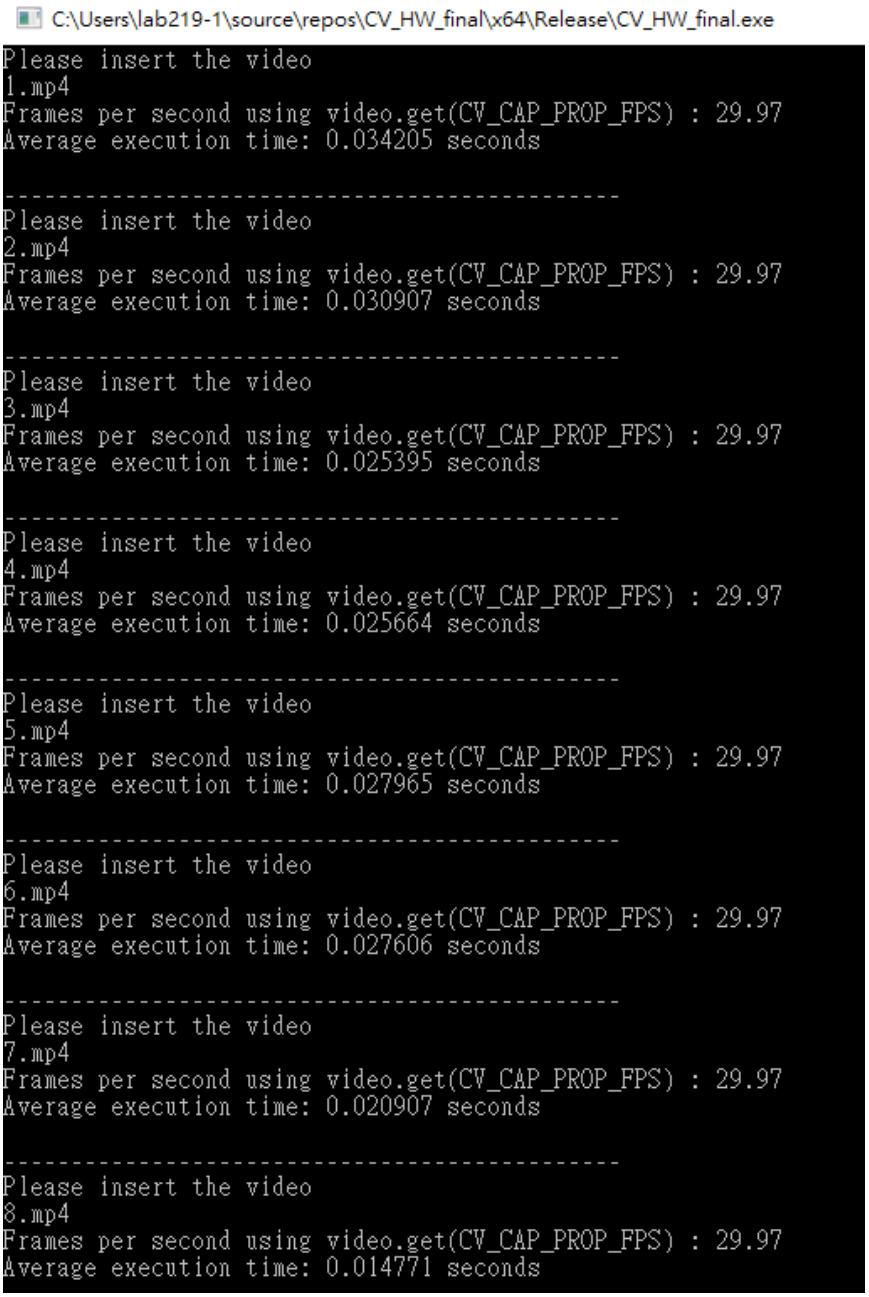
A. The fps and average of executive time of each video

將每一個 frame 的執行時間相加後除上總 frame 數。FPS 參照[4]，套用函式 video.get(CV_CAP_PROP_FPS);

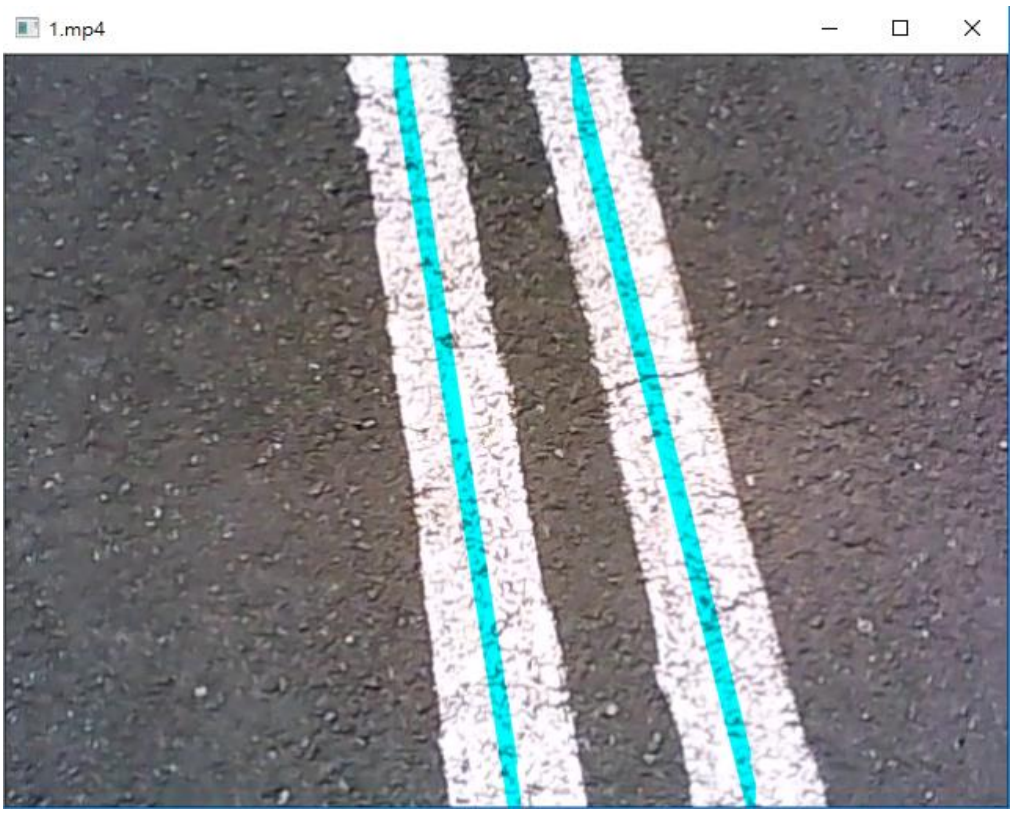
TABLE I. THE FPS AND AVERAGE OF EXECUTIVE TIME OF EACH VIDEO

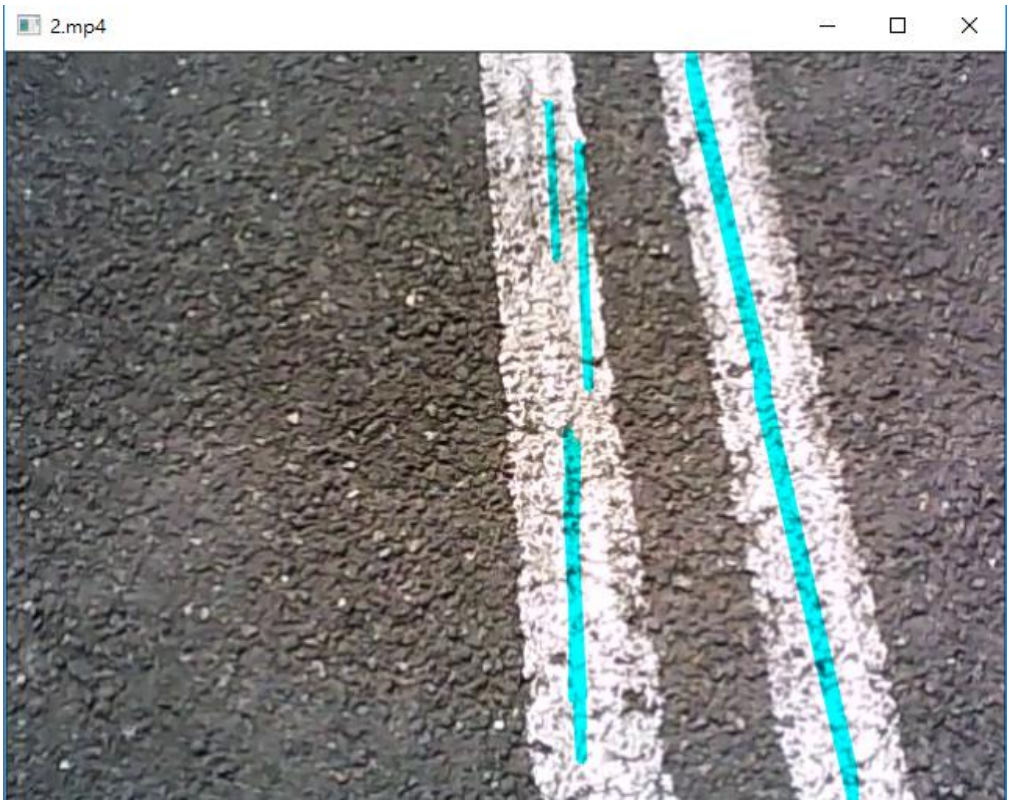
Video_number	The average of executive time (sec)	FPS
1	0.035	30
2	0.034	30

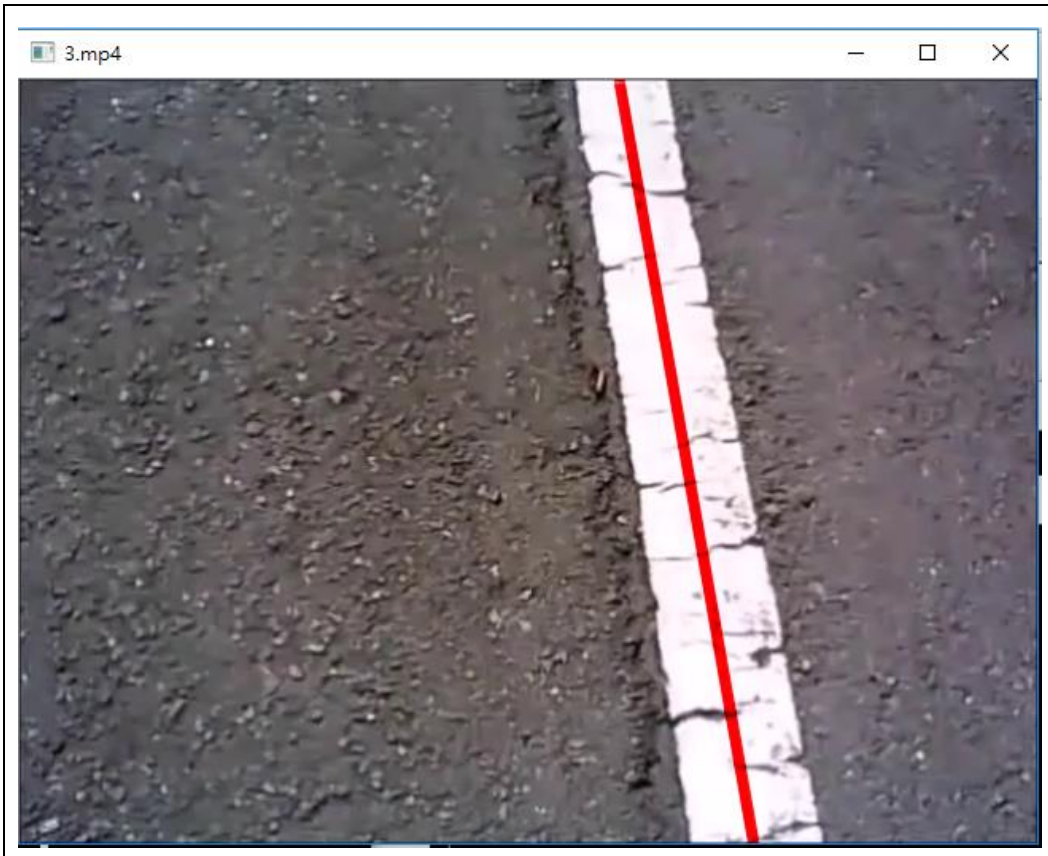
3	0.026	30
4	0.025	30
5	0.028	30
6	0.027	30
7	0.022	30
8	0.014	30
Total_AVG	0.026	30



B. Result of 8 video

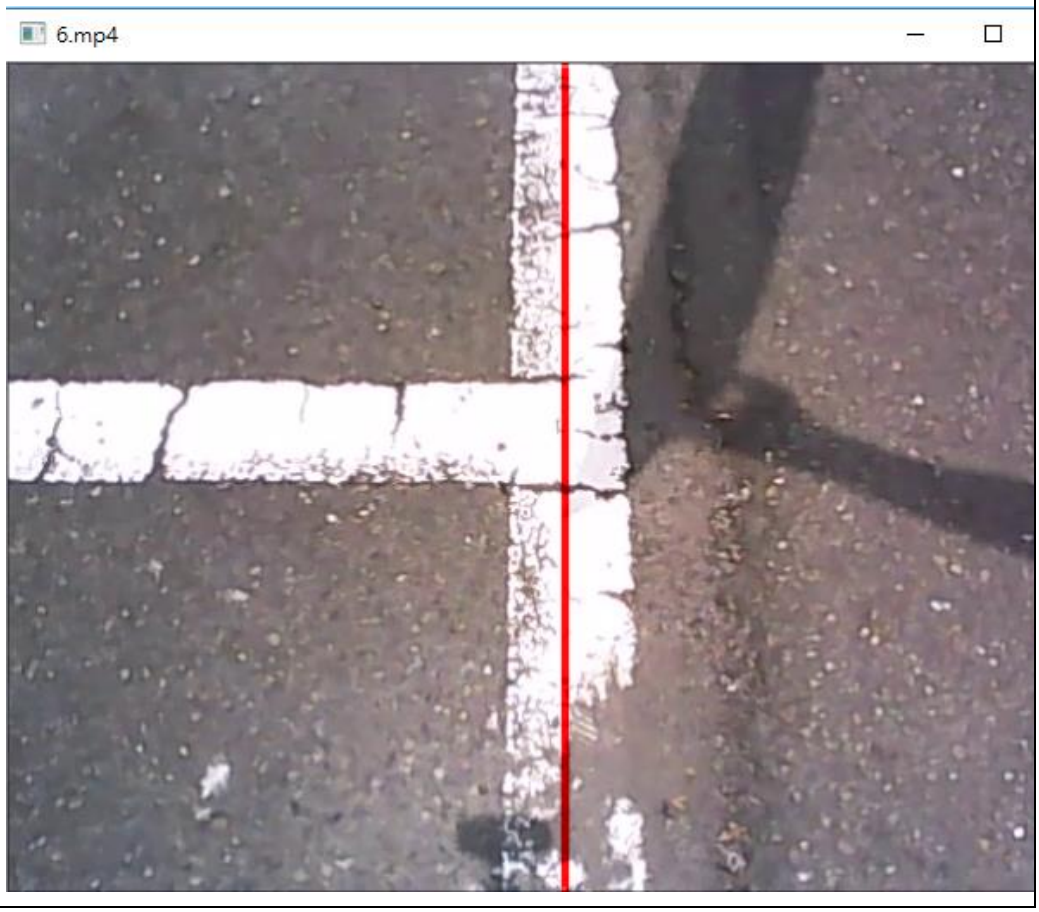


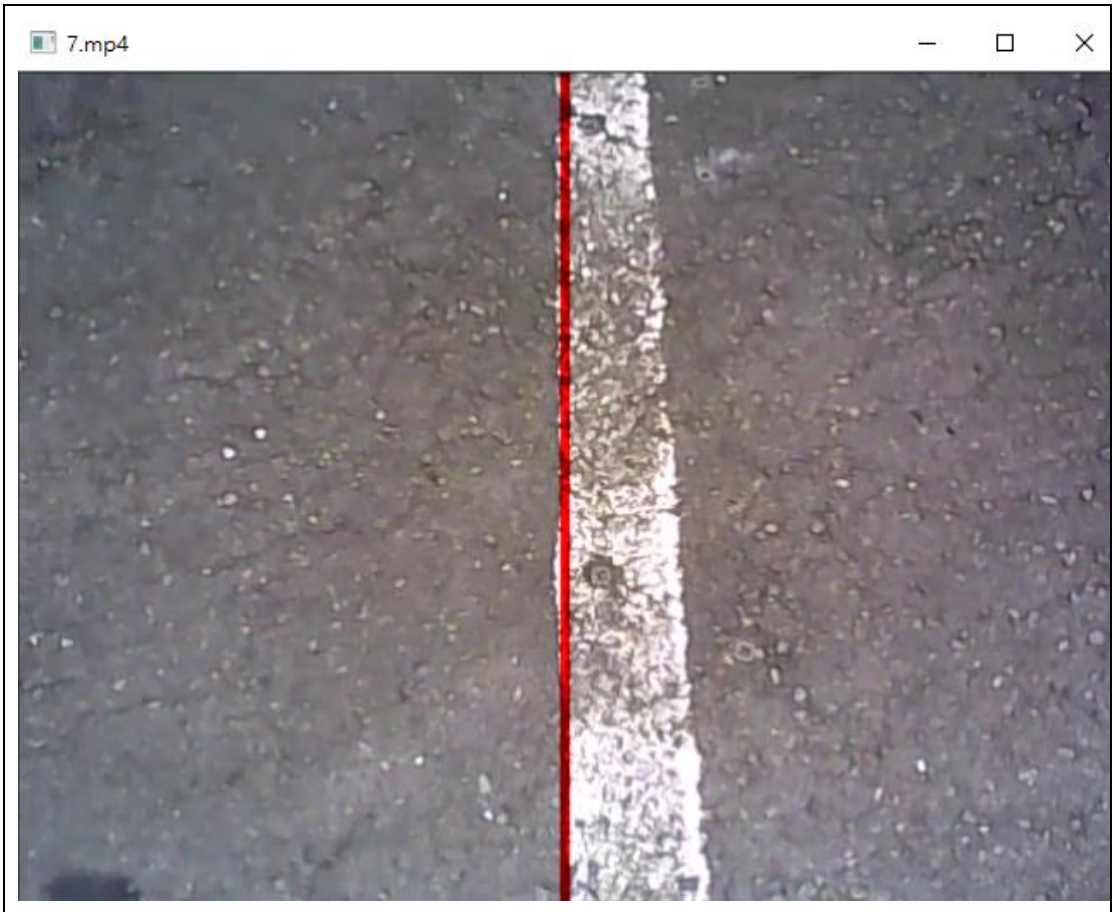



3.mp4


4.mp4


5.mp4


6.mp4


7.mp4


8.mp4

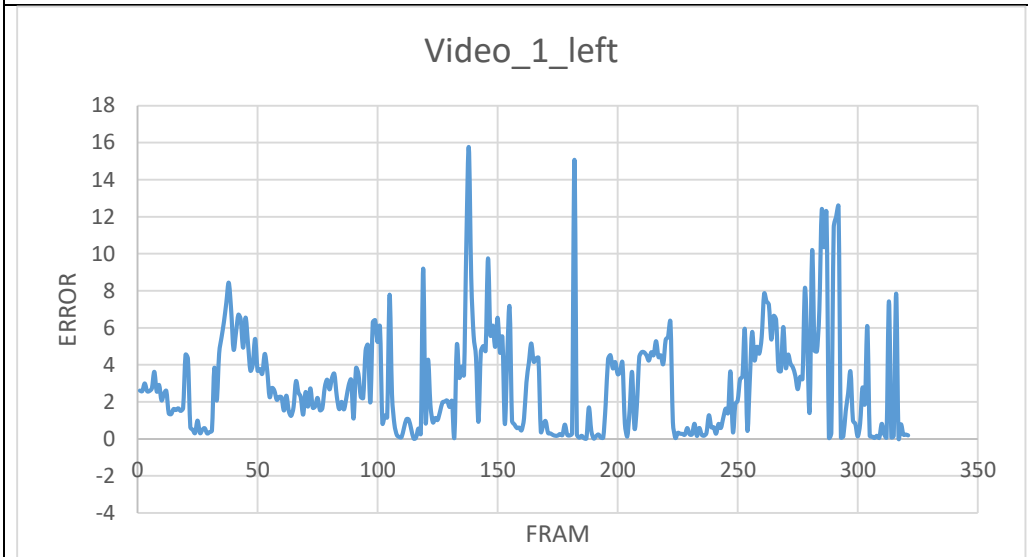
C. The error trajectory between result and ground truth

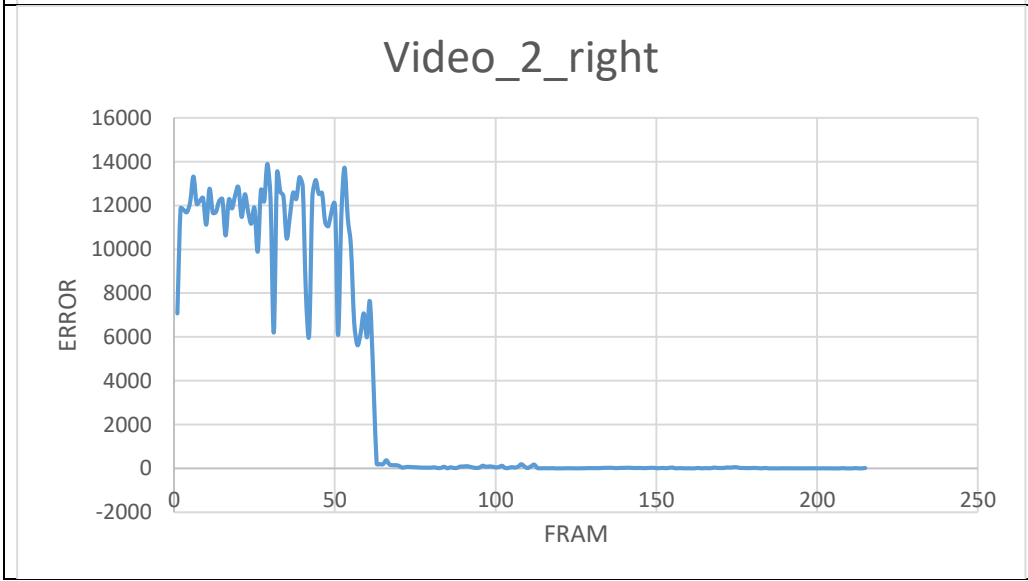
使用題目上所給予的公式計算出：

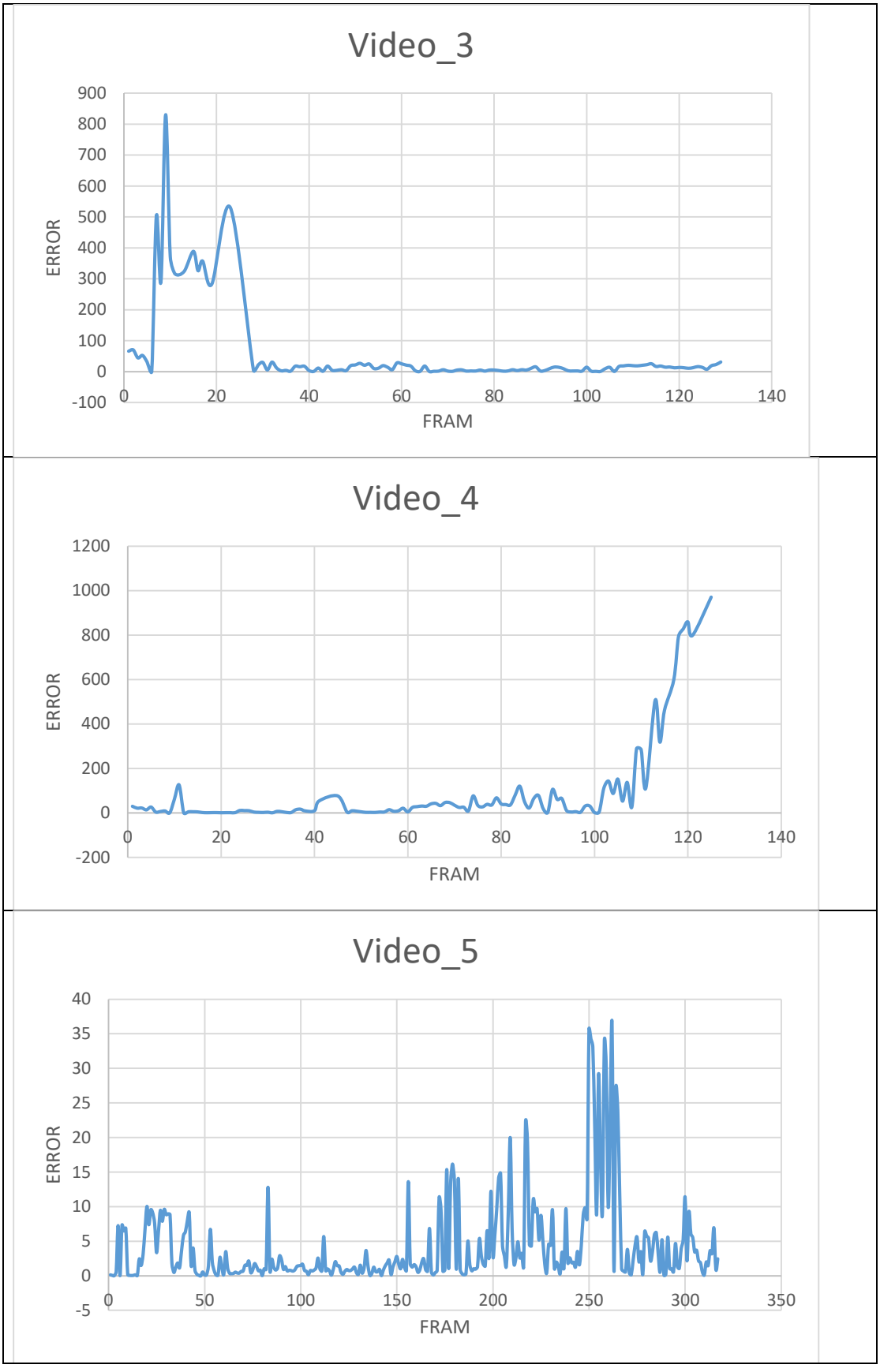
Error= $\left(\sqrt{(p1_x - g1_x)^2 + (p1_x - g1_x)^2} + \sqrt{(p2_x - g2_x)^2 + (p2_x - g2_x)^2}\right)$：

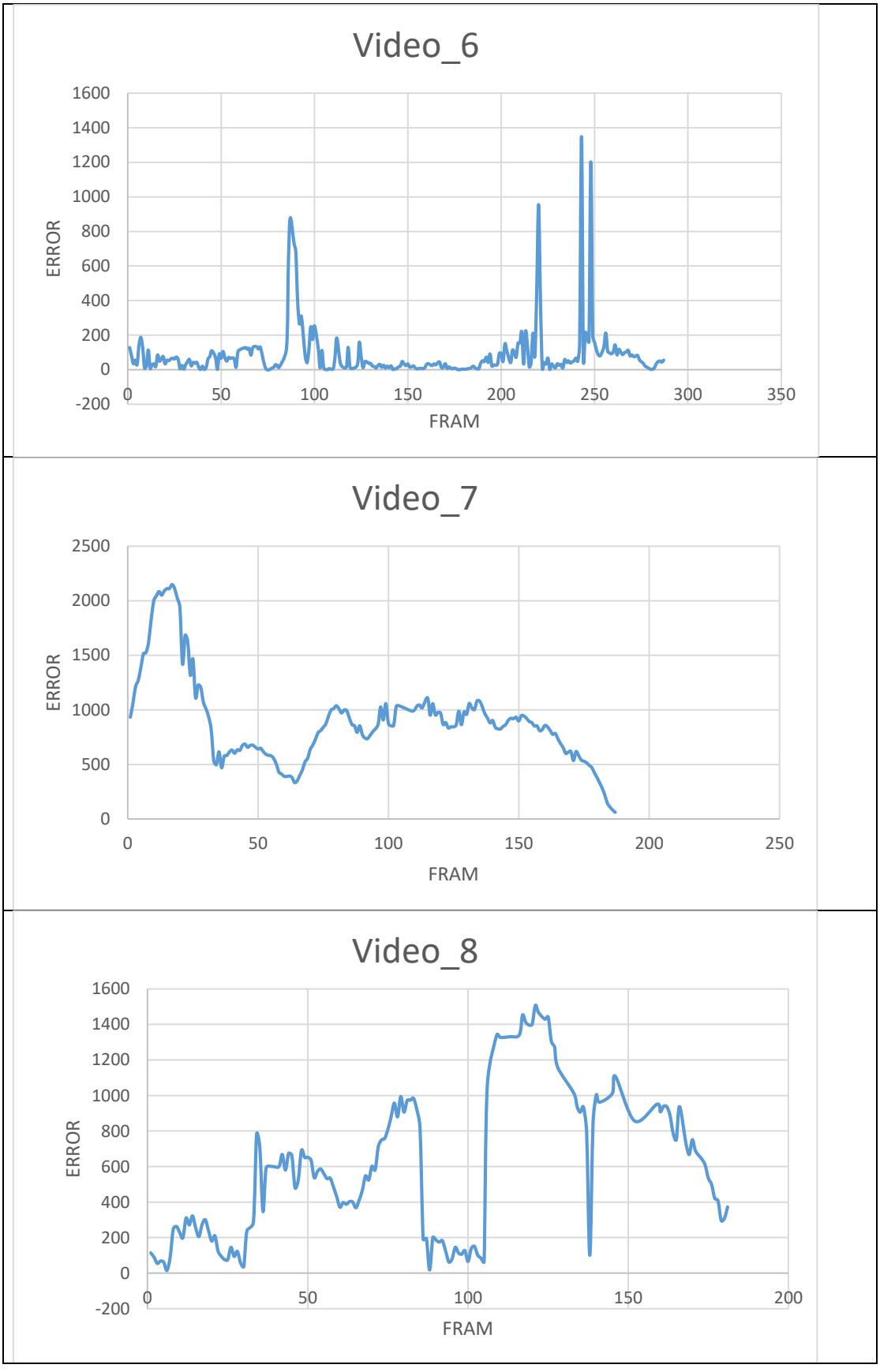
影片 1 的值基本上不會差異太大，結果還算不錯；影片 2 的一開始會亂跳，會從左邊切到右邊去，所以一開始的 ERROR 值會很大；影片 3 則是一開始會有點亂跳，後來就還蠻穩定的；影片 4 的後面因為預到車格的關係，在加上前處理可能沒很好，所以斜率跑掉了；影片 5 的部份結果還算不錯，遇到車格也沒什麼偏掉；影片 6 可能受到音影跟線跳偶爾有破碎，所以線常常會亂跳；影片 7 和 8 路線本來就是破碎的，所以線會左右亂跳或一開始根本沒在中間，導致 ERROR 值超級大；其實我覺得這次的作業相當可惜，如果前處理能在做得更完善益點，畫線結果應該會改善許多。

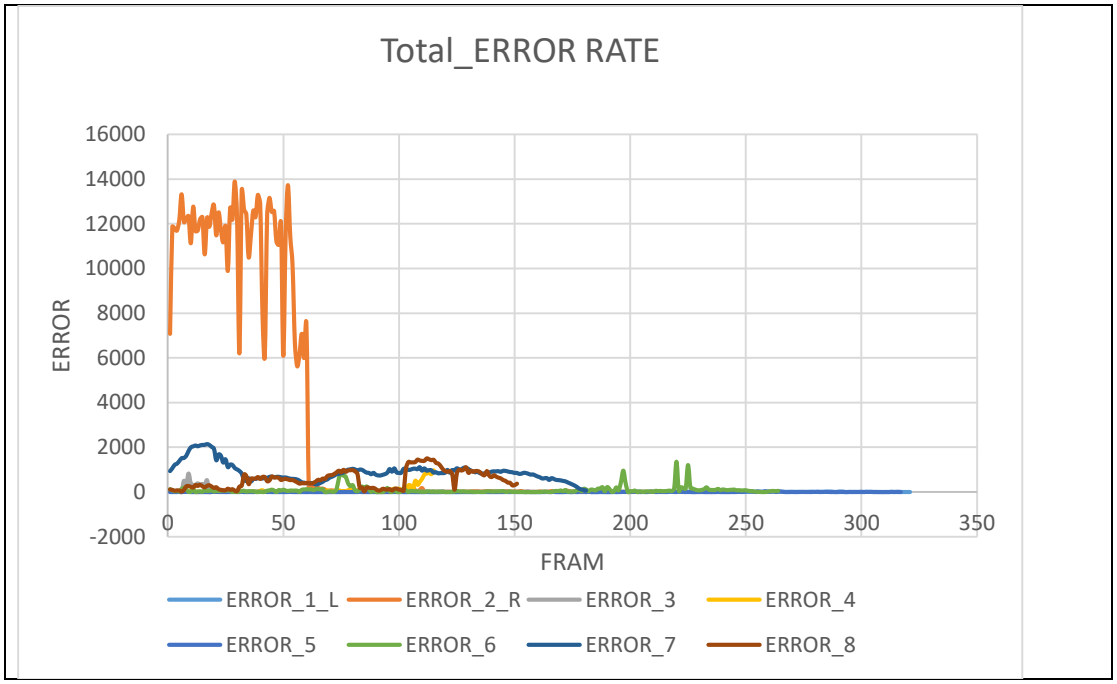
ERROR 圖表






Video_3
900
800
700
600
500
400
300
200
100
0
-100
ERROR
0
20
40
60
80
100
120
140
FRAM
Video_4
1200
1000
800
600
400
200
0
-200
ERROR
0
20
40
60
80
100
120
140
FRAM
Video_5
40
35
30
25
20
15
10
5
0
-5
ERROR
0
50
100
150
200
250
300
350
FRAM


Video_6
1600
1400
1200
1000
800
600
400
200
0
-200
ERROR
0
50
100
150
200
250
300
350
FRAM
Video_7
2500
2000
1500
1000
500
0
ERROR
0
50
100
150
200
250
FRAM
Video_8
1600
1400
1200
1000
800
600
400
200
0
-200
ERROR
0
50
100
150
200
FRAM



References

[1] cvtColor and threshold function [Online]. https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html

[2] Eroding and Dilating[Online]. https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

[3] Morphological Transformations [Online]. https://docs.opencv.org/3.4.2/d4/d76/tutorial_js_morphological_ops.html

[4] HoughlineP function [Online]. https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghlinesp#houghlinesp

[5] Video function [Online]. https://stackoverflow.com/questions/10475198/retrieving-the-current-frame-number-in-opencv

[6] Convert HoughLinesP output line format to HoughLines output line format [Online]. https://stackoverflow.com/questions/40638778/convert-houghlinesp-output-line-format-to-houghlines-output-line-format

[7] Thickening [Online]. https://blog.csdn.net/FunnyWhiteCat/article/details/80670332