Introduction to Image Processing
Final Project

Lane Detection based on dynamic ROI with MATLAB

報告成員：

電機 3B 408415002 林　想

電機 3A 408415005 吳致廣

電機 3B 408420083 陳昀顥

目錄

一、前言

車道線自動偵測在自動駕駛方面佔有重要一席之地，然而畫面上經常受到非車道線的干擾，導致在偵測過程中容易會產生誤判。本次期末專題以霍夫轉換作為偵測車道線條的主要概念，並使用 Matlab 來撰寫程式且不會使用到內建函式:Hough、HoughPeaks、Houghlines，在文中會詳細說明程式內的功能及流程圖，文末呈現實驗結果與分析數據。

二、 欲解決的問題

本次專題我們預期解決的問題有以下幾點:

1. 選取 ROI 區域自動化，並根據圖片灰階值變化選擇合適的範圍。
2. 在車道線偵測上只判別車子中心區域的左右車道線。
3. 繪製車道線條時將左右兩邊底部延伸至圖片下方。

三、所使用的演算法介紹

1. 所有函式 & 程式碼流程圖

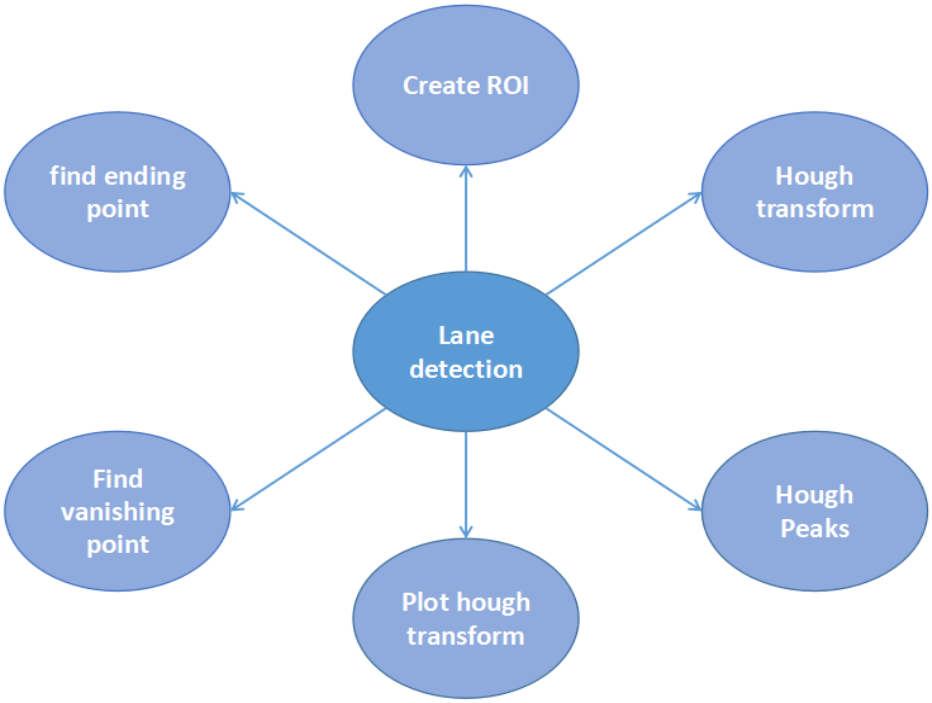

圖 1. 所有函式功能架構圖

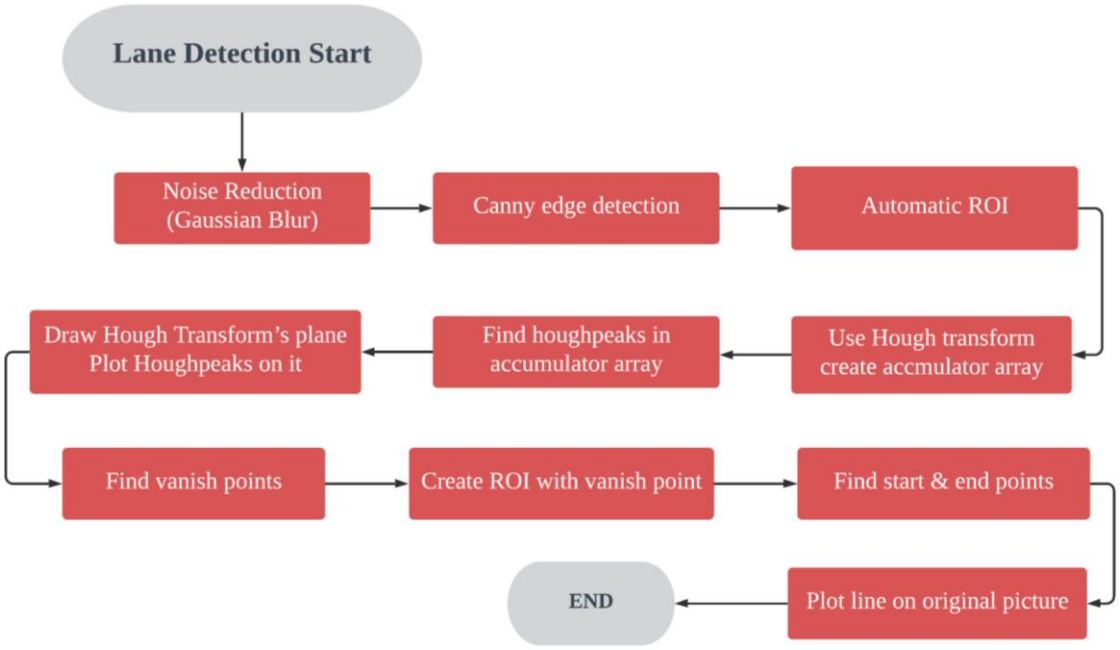

圖 2. Lane detection 流程圖

2. 介紹副函式功能

(1). Image Pre-processing

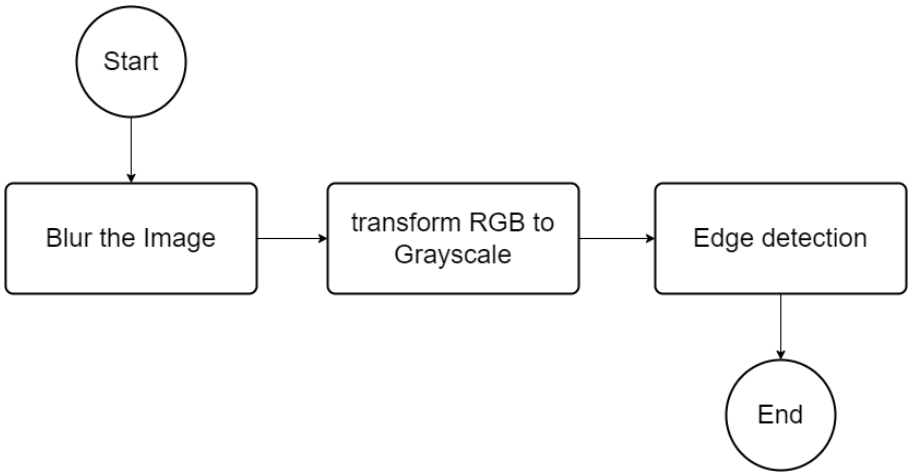

圖 3. 影像前處理流程圖

為了避免原始影像中的雜訊以及周圍的景物會影響到車道線的邊緣偵測，首先以高斯濾波器(Gaussian Filtering)將原始影像過濾一次，再將其轉為灰階影像去做邊緣偵測，這邊使用是 Canny 邊緣偵測，因為其可以調整參數去控制所需要偵測到的強邊緣與弱邊緣，因此有比較好的彈性去做濾波。

(2). create ROI

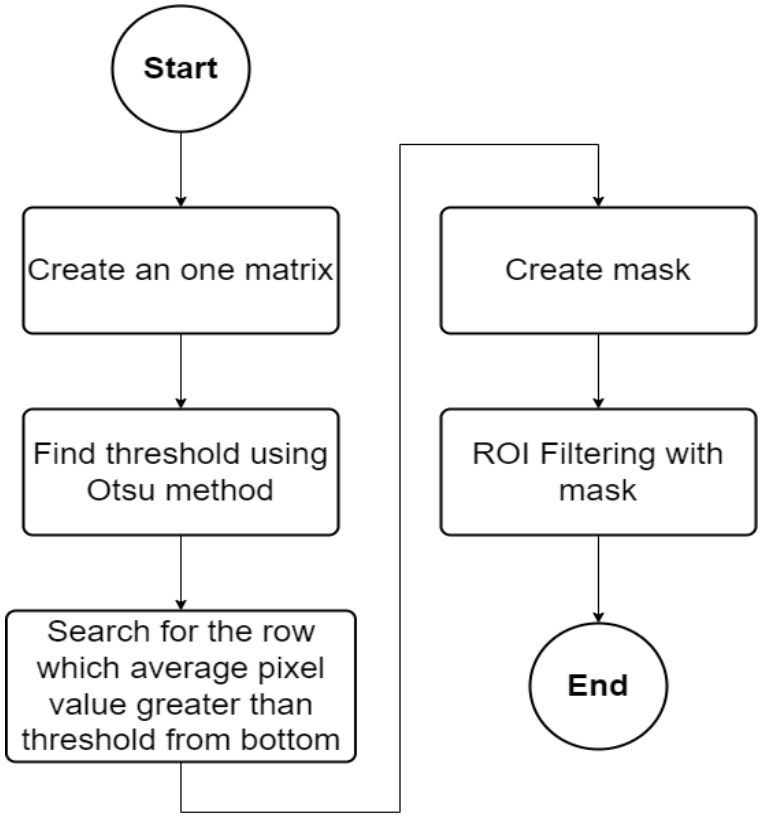

圖 4. Create ROI 流程圖

觀察灰階影像的直方圖(Histogram)會發現像素的值主要會落在兩個區塊--亮區與暗區，透過觀察每列的直方圖，可以發現道路的直方圖主要都是落在暗區的部分，因此只需要找出亮區與暗區之間的交界點，就可以將車道獨立分割出來，把其他不需要的部份去掉。

首先創出一個與原圖尺寸相同的矩陣並將其值設為 1 ，為了在之後建立遮罩(mask)時需要用到，再來透過大津演算法(Otsu's method)將前景與背景(亮區與暗區)的交界點取出來，透過由下往上去計算每列灰階像素的平均值，去和交界點比較，若是平均值小於交界點，代表該列還位於車道內；若是平均值超過交界點，則代表該列已經超出車道的範圍，車道的部分都已經檢查過，就在前的建立好的遮罩上將超出車道的列以上都設為 0 ，最後將遮罩蓋上邊緣偵測後的二值化影像就可以得出所需要的 ROI(如圖 5 所示)。

圖 5. 取 ROI 之結果

(3). Hough transform

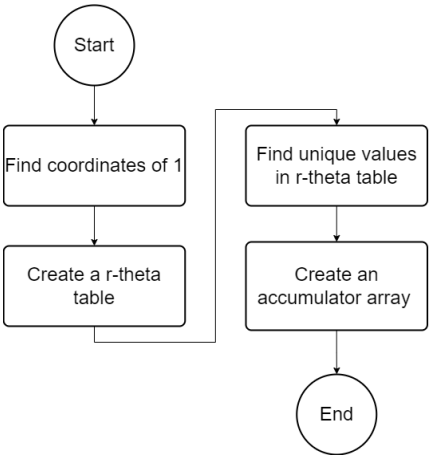

圖 6. 霍夫轉換(Hough transform)流程圖

霍夫轉換主要的目的是要找出通過最多點的線，最主要的線也就是車道線，先找出二值化影像中所有有數值的座標位置，在將其座標帶入霍夫轉換公式內，可以得出通過該點的所有線之 r 與 Θ ，並且將所有不一樣的 r 值取出標示出來， 並根據每個 r 值去建立 accumulator array : H，這裡不取用 20 度以下以及 160 度以上是為了要避免過於平的線被誤認為車道線，以及 90 度的線也不用，避免有路燈、相片邊緣等被偵測誤認為車道線。

(4). Houghpeaks

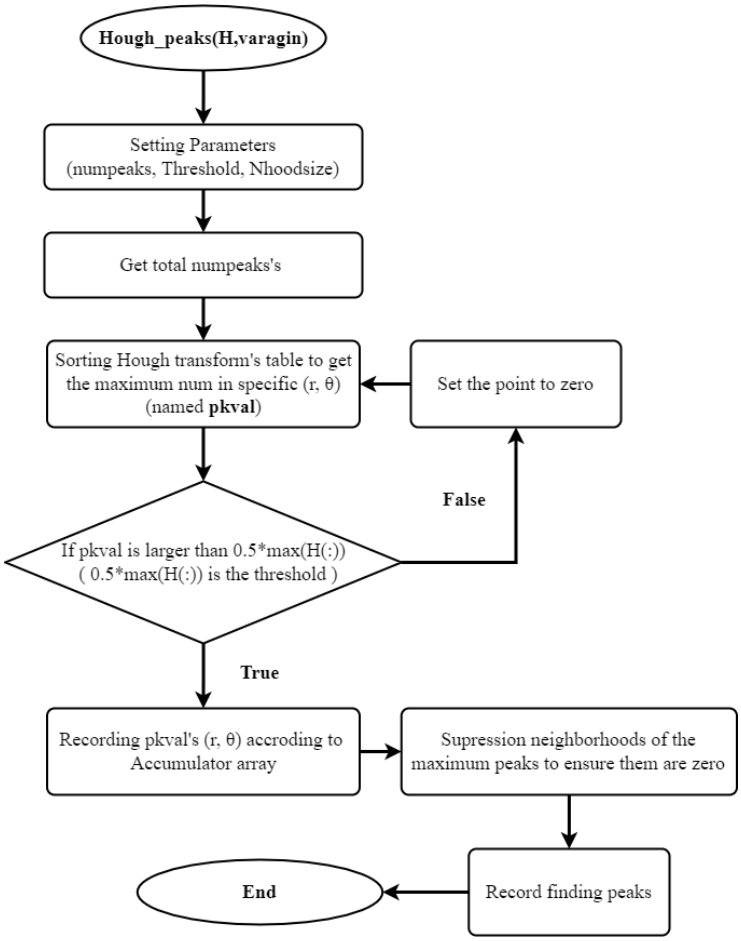

圖 7. Hough_peaks 流程圖

在找尋最大峰值之前，此函式會依據外部條件輸入設定三個參數:numpeaks (輸出欲得到的峰值數量)、Threshold(Accumulator Array 中最大值的一半當作基準值)、NHoodSize(抑制鄰域的大小，在偵測峰值後將該周圍的鄰域設置為 0)。

宣告完上述參數後，設置第一層迴圈跑的次數為 numpeaks 的數量，總共有 x 個就找到 x 個峰值，首先會進行 Sort by Descending ordered 將 H 最大值排在前面，如果大於閾值則透過 find 回傳 Accmulator Array 中符合 Hs(1)的(r, θ)值到暫存變數 h,k 裡面，接著進行 Supression 將該點附近的峰值進行歸零(以確保下次找尋時不會重複找到同一點附近的峰值)，最後將找到的(r, θ)存到 peaks 裡面 (peaks 格式為 peaks x 2 的矩陣，譬如要找 4 個峰值則矩陣大小為 4 x 2)。

(5). Plot_hough_transform

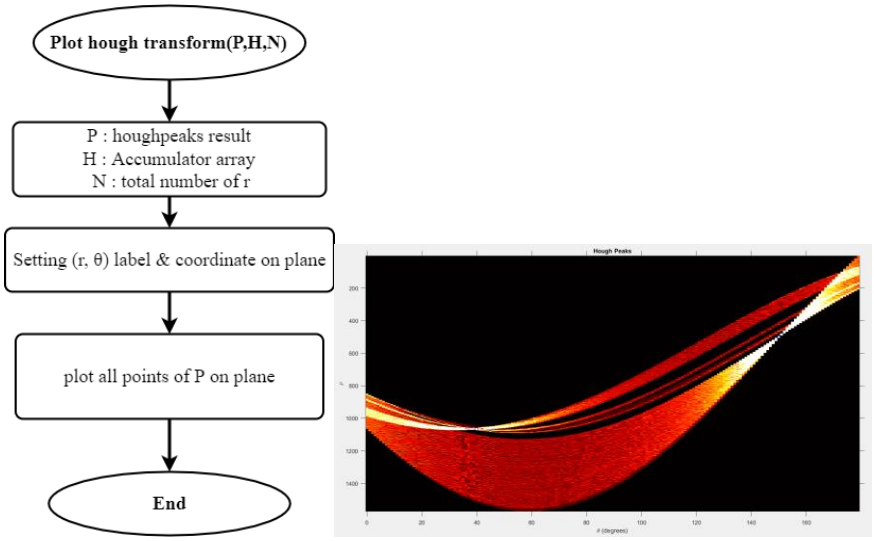

圖 8. Plot_hough_transform 流程圖&示意圖

為了確認最後繪製出來的車道線是否與 houghpeaks 所找到的點一致，因此設計出此函式來表示 Hough transform 的結果於(r, θ)座標上。首先會設定座標參數及座標範圍來建構初體圖，將線條用 colormap(gca,hot)讓線條交會處更加明顯，接著將 P 內的所有座標點標上轉換圖上，並標示成藍色點方便確認(圖 8.)。

由於顧及到 ROI 自動化周圍有殘留不必要的雜訊，在 Hough transform 角度選取方面選擇不計小於 20 度及 160 度的角度，至於 90 度部分去除的原因是怕圖片的左右邊緣過於強烈讓 houghpeaks 誤認為 strongest line(圖 9.)。

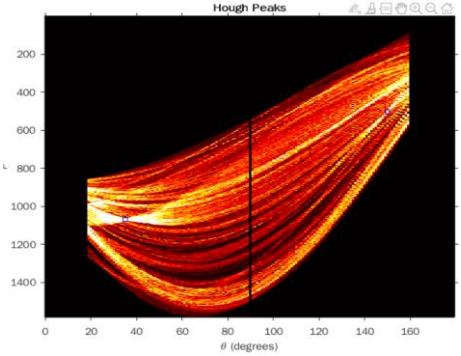

圖 9. 經改良後的 hough_transform & Plot_hough_transform 結果

(6). Find vanishing point

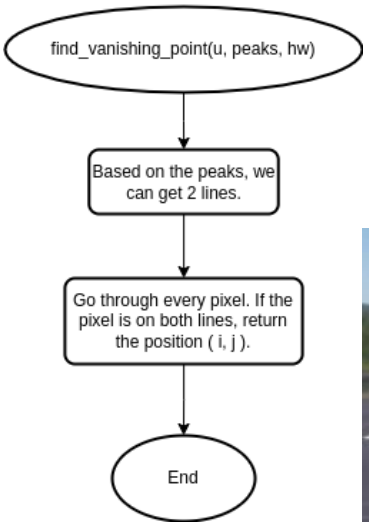

圖 10. find_vanishing_point 流程圖

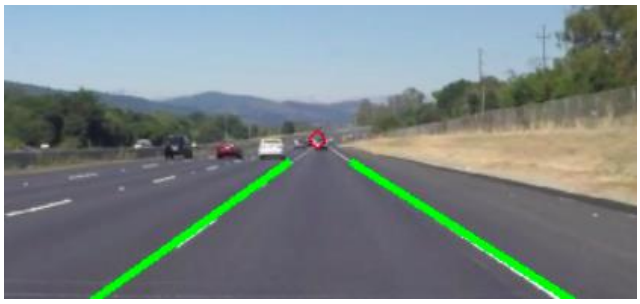

圖 11. Vanishing point 示意圖

為了在接著找起終點時能夠繪製更完整的車道線，在這邊會先尋找消失點 vanishing point。因為車道線會匯集到消失點，所以可以準確地找出車道在圖上分佈的位置。利用 Hough peaks 找到的兩個 peaks，找到兩條線之交點，並回傳交點。消失點的位置如圖 11 的紅色圓圈。

(7). Create ROI with vanishing point

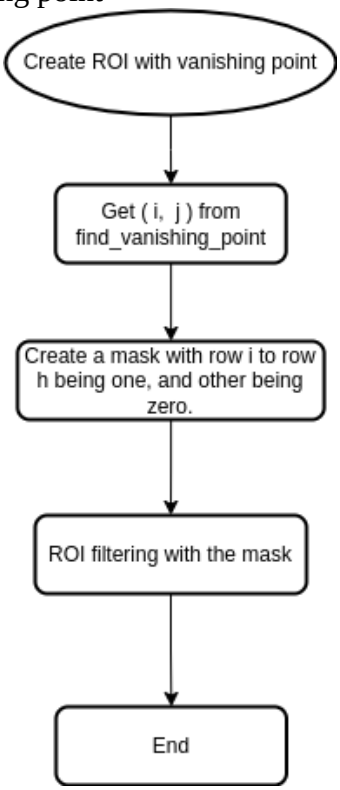

圖 12. Create ROI with vanishing point 流程圖

找到 vanishing point 之後，利用 vanishing point 的 row 值 i 決定 mask 的邊界，row i 以上設為 0，以下設為 1。最後用 mask 對 edge image 做 ROI filtering，得到較好的 ROI 影像。

(8). Find start & end points

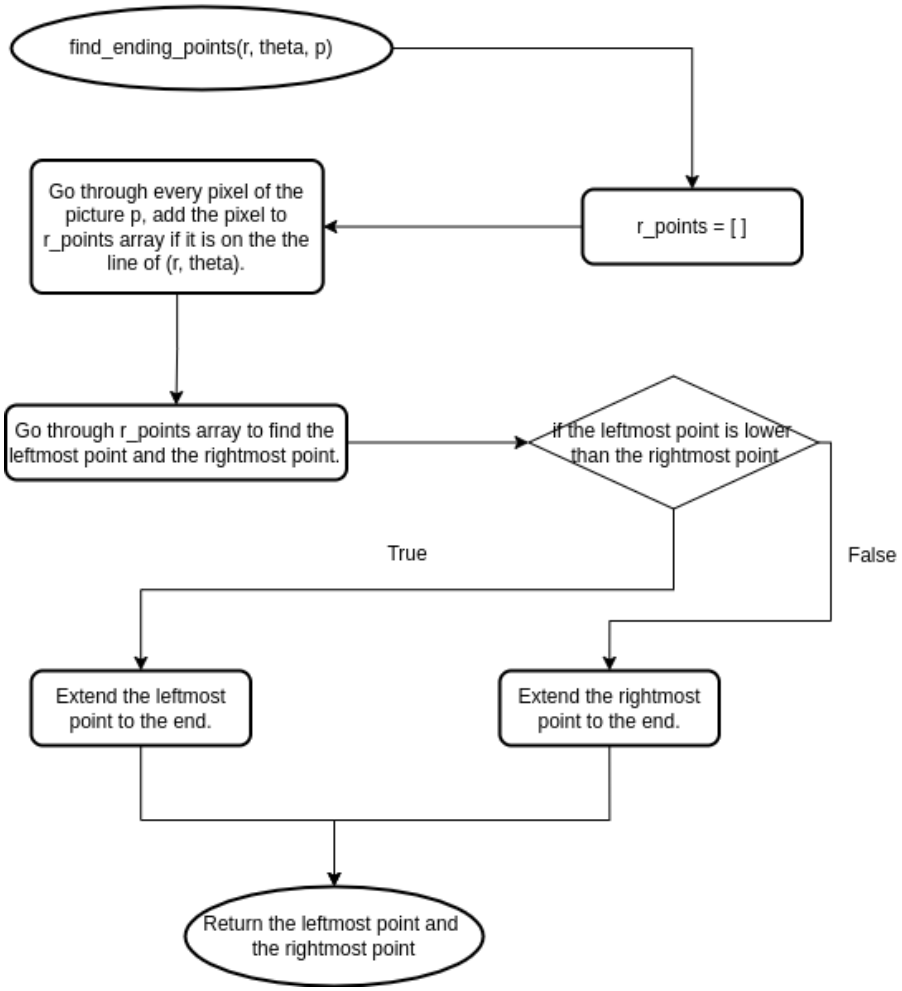

圖 13. Find start & end points 流程圖

本函式會利用 edge 圖片找到車道線的起終點。有三個引數，r theta 與 p，分別為 Hough transform 得到的 r 與 theta, 與欲尋找起終點的 edge 影像 p。首先會找出所有在 r theta 線上且在 edge 影像 p 上為 1 的點，並存在 r_points array 裡面。接著在 r_points array 中找出最左邊與最右邊的點。最後把線的起終點延伸到圖片的最下方，並回傳起終點。

(9). Plot line on the original image

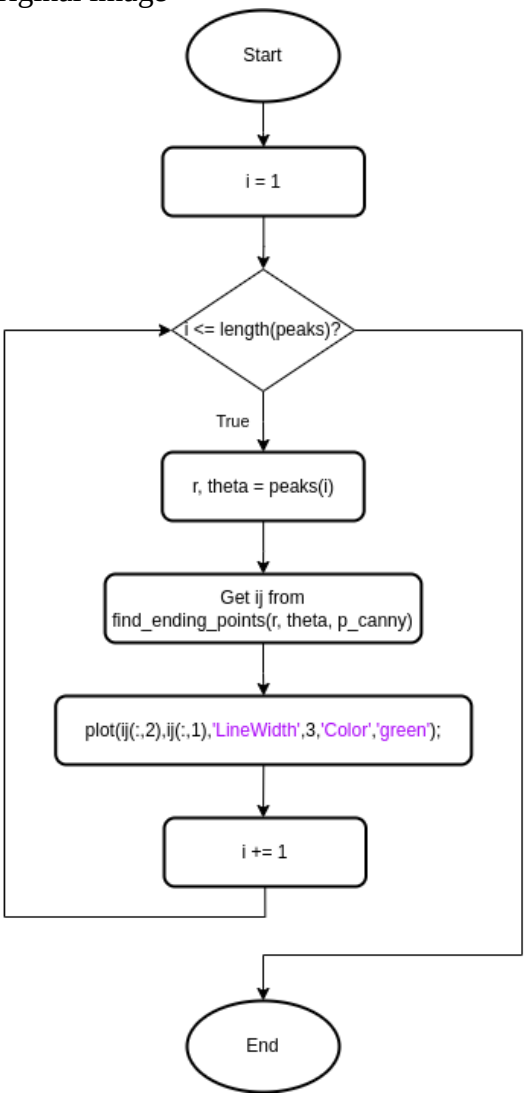

圖 14. Plot line on the original image 流程圖

由 Hough peaks 找到的兩個 peaks，利用 find ending points 找到起終點，最後在原本的影像上，對兩條偵測到的車道線畫上綠色的標記線。

四、實驗結果與分析

1.成功的影像

圖 15. 高速公路車道線偵測

分析：

對與 sample_lane 影像相似的高速公路影像(圖 15.)，能夠忽略旁邊的路標與車子，並擷取到不錯的 ROI，去掉背景，最終標記出正確的車道線。

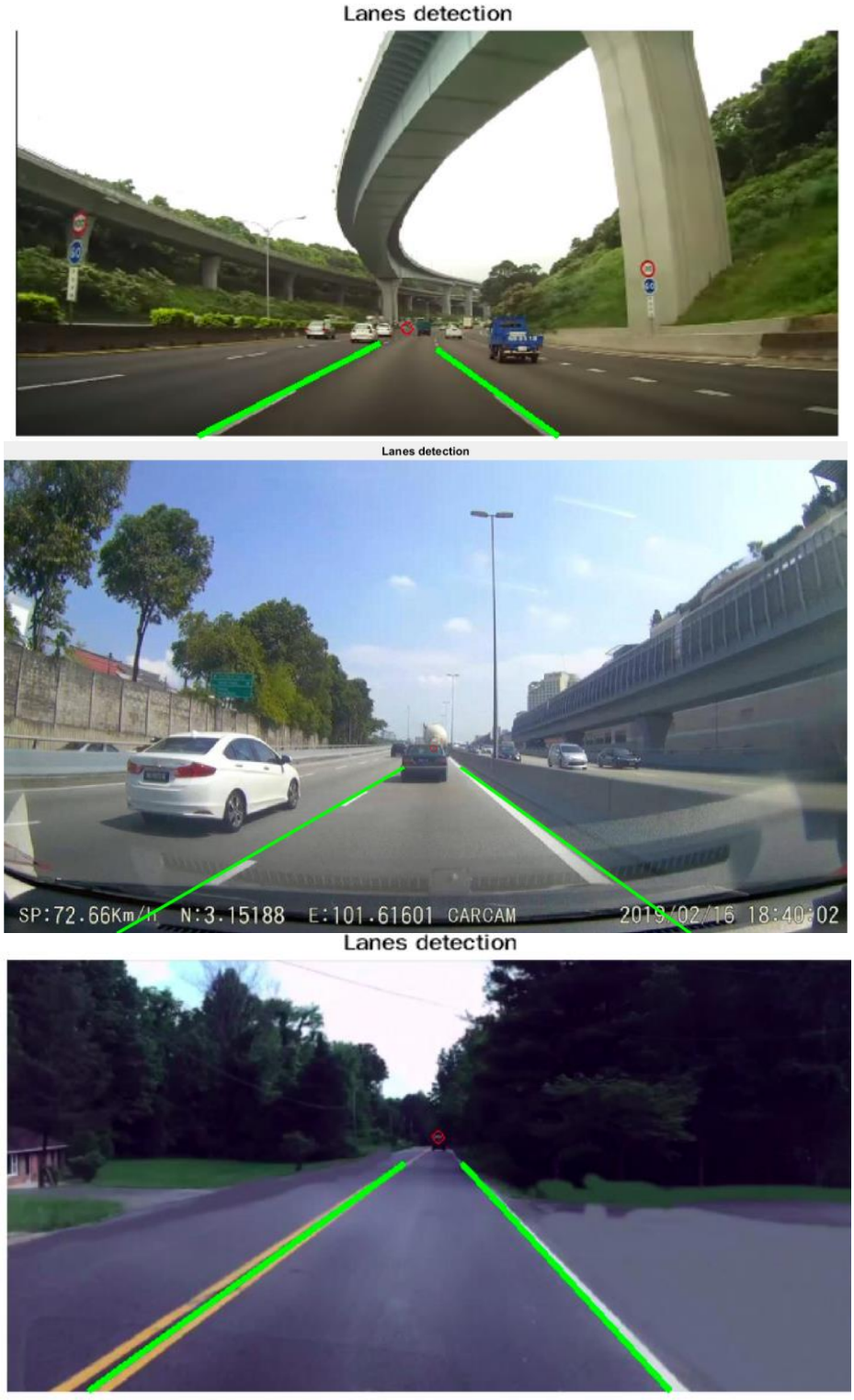

圖 16. (上)(中)(下)一般道路車道線偵測

分析：

由於我們在偵測道路線上有過濾掉角度離 90 度很遠的線，所以可以避免掉很多彎彎曲曲以及旁邊車道等雜訊所形成的線(圖 16.)。

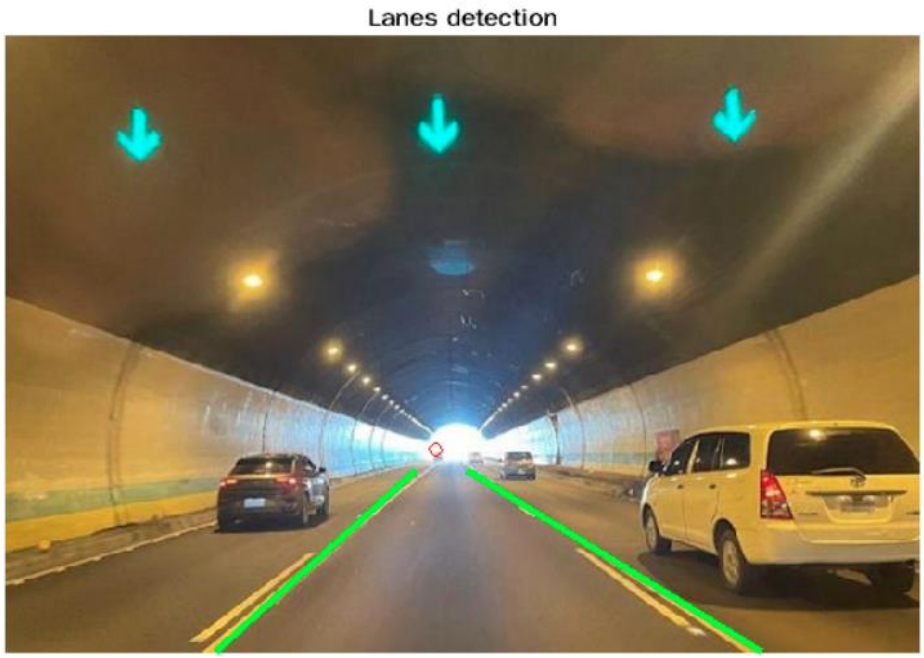

圖 17. 隧道車道線偵測

分析：

在光線明亮的白天，路面與背景的對比夠大(圖 17)，用灰階之亮暗來決定 ROI 可以取得很不錯的效果，進而找到正確的車道線。

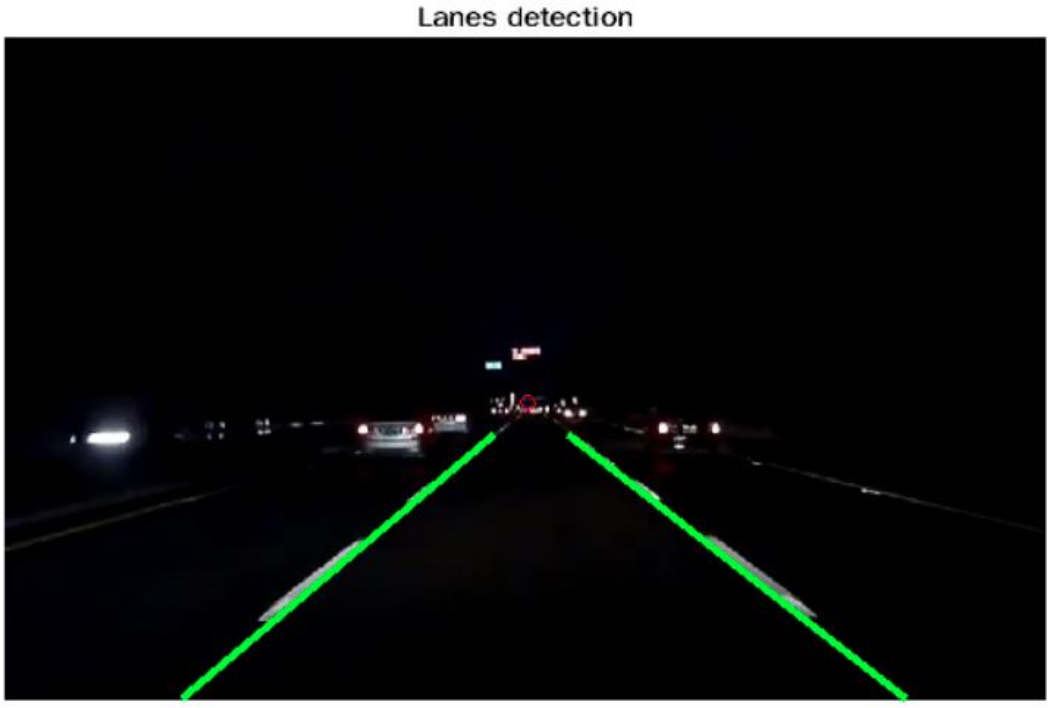

圖 18. (上)(下)夜間車道線偵測

分析：

就算是在晚上，若燈光足夠(圖 18)，使用灰階之亮暗來決定 ROI 的演算法也可以取得不錯的效果，並找到正確的車道線。

2.失敗的影像

(1).道路兩旁資訊過多

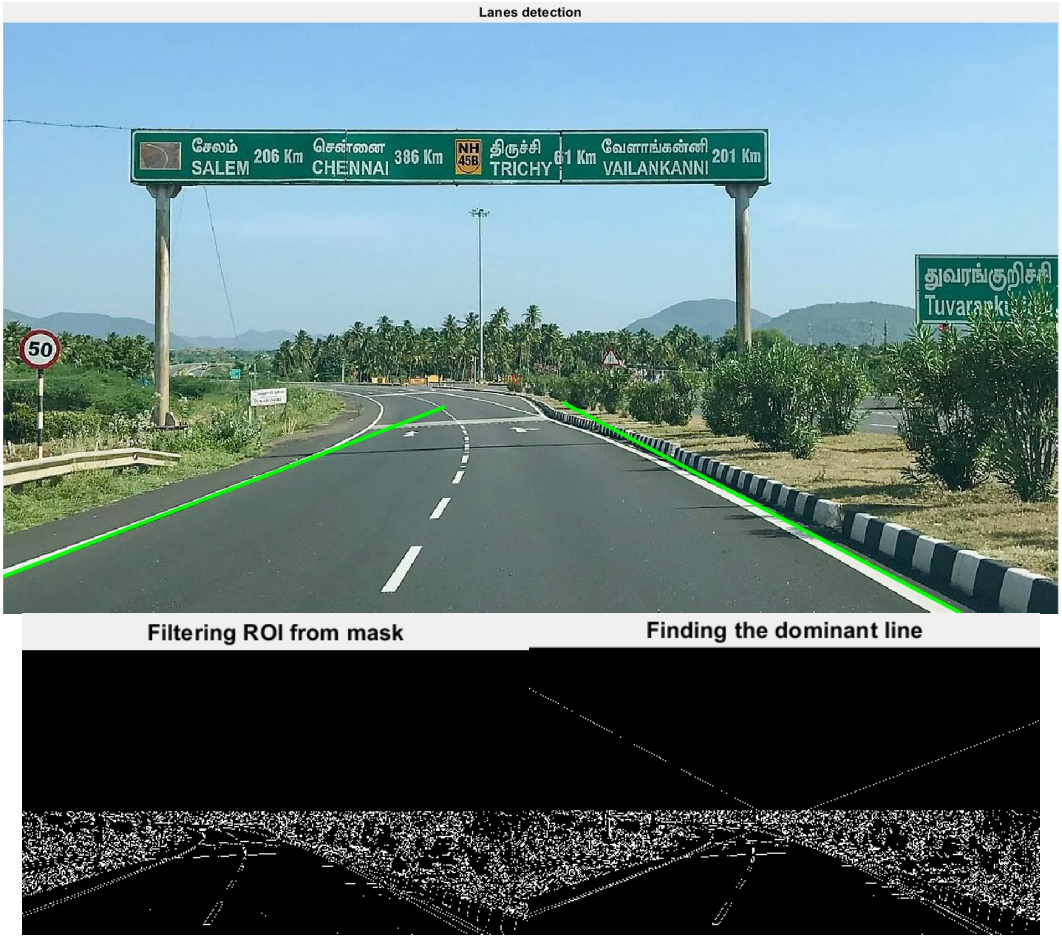

圖 19. 車道兩旁有樹叢之車道線偵測

分析：由於我們的 Mask 是取影像下半部有道路的部分，無法濾掉道路兩旁雜訊(圖 19.)，故會影響車道線的判斷。

(2).場景上暗下亮使 ROI 偵測不到

圖 20. 利用 OSTU's Method 所得到之 global threshold(*255)為 33

-比較其他黑暗場景 ROI 取樣成功過程

圖 21. 利用 OSTU's Method 所得到之 global threshold(*255)為 55

圖 22. 利用 OSTU's Method 所得到之 global threshold(*255)為 77

分析: 在失敗的案例(圖 20.)中可以看到其 global threshold 比成功的案例(圖 21.22.)還要小很多，所求得之 ROI 全黑推測是每一列灰階像素平均值超過其交界點，誤認成超出車道線範圍，因此整個圖被 Mask 刪除只剩全黑部分。

3. 未來欲解決之問題方向:

去除掉道路兩旁之雜訊、對上半部背景過暗之圖片改善 ROI 運算方式。

五、工作項目分配表

項目\成員名字	林 想	吳致廣	陳昀顥
相關資料蒐集	33%	33%	33%
演算法開發	40%	30%	30%
程式撰寫	40%	30%	30%
數據分析與解釋	33%	33%	33%
書面報告	25%	30%	45%

表 1、分工項目圖表

六、參考文獻&資料

[1] https://www.mathworks.com/help/images/ref/houghpeaks.html
[2] https://www.mathworks.com/help/matlab/ref/inputparser.parse.html
[3]https://stackoverflow.com/questions/9916253/hough-transform-in-matlab-without-using-hough-function
[4]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9085354
[5] https://ieeexplore.ieee.org/document/8230303