

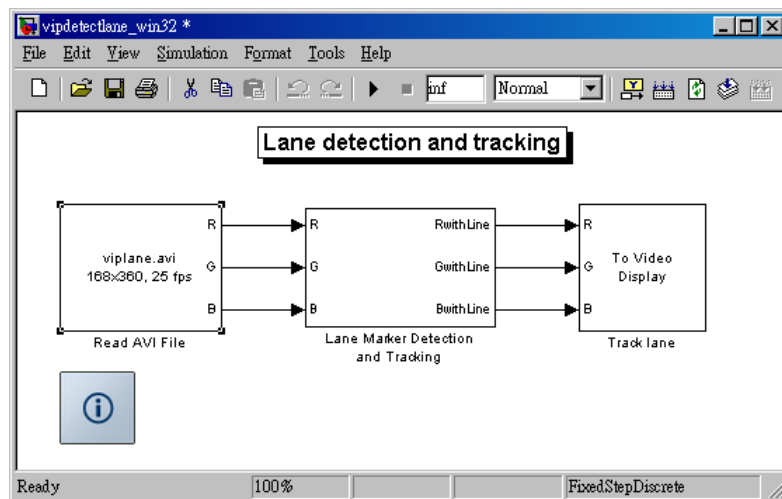
範例：車道偵測與追蹤

本範例模擬由車輛上所擷取的道路影像資料，先經過色彩空間的轉換後，再利用霍氏轉換 (Hough Transform) 偵測車道的邊線，並經由邊線的夾擠 (控制) 追蹤車道。

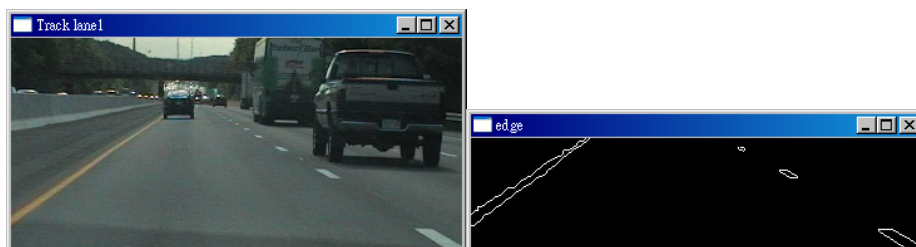
Simulink 是一個建構在時間的基礎上，以模組為基礎可開發設計的程式語言。以本範例而言，由主程式 (圖一) 可以清楚看出其架構：

- (1) 原始的影像資料檔案 (viplane.avi) 由 VIP Blockset 裡的「Read AVI File」block 讀取後，分為 R、G、B 三個通道 (channel) 輸出
- (2) 主要演算法「Lane Marker Detection and Tracking」處理
- (3) VIP Blockset 裡的「To Video Display」block 播放影像資料檔案

本文中將詳細介紹以上三部份，包括如何在 Simulink 中讀取影像資料檔案、對主要演算法「Lane Marker Detection and Tracking」內部程式解說、與最後的輸出。輸入的原始影像檔案如圖二 (A) 所示，圖二 (B) 則為霍氏線性直線加入在二值化影像檔案中的結果。



圖一 車道辨識與追蹤 Simulink Model



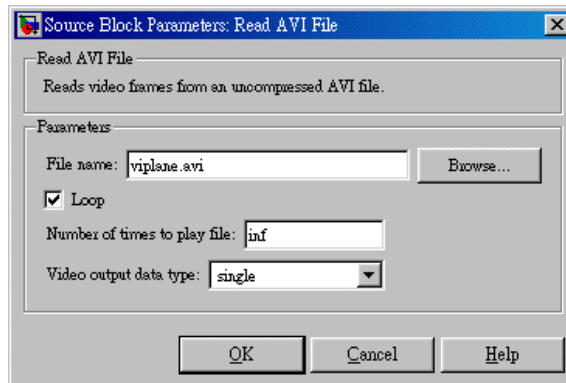
圖二 (A) 輸入原始影像檔案 圖二 (B) 二值化影像之霍氏線性直線

◆如何在 Simulink 中讀取影像資料檔案

Simulink 的 Video and Image Processing Blockset (模塊函式箱) 中的「Source」模塊群組中，

可以找到「Read AVI File」的 Block（模塊）。加入 Simulink model 之後點選進入模塊的參數設定視窗，填寫欲讀取的 AVI 檔案名稱與播放方式等參數（圖三）。

在 Parameter 中，File name 可經由「Browse」瀏覽選取已貯存的 AVI 格式檔案，決定是否以「Loop」方式重複讀取；然後選擇播放的次數（Number of times to play file，本範例播放無限次：inf）。最後選擇影像檔案輸出的資料型態，本範例為 single。



圖三 「Read AVI File」模塊參數設定視窗

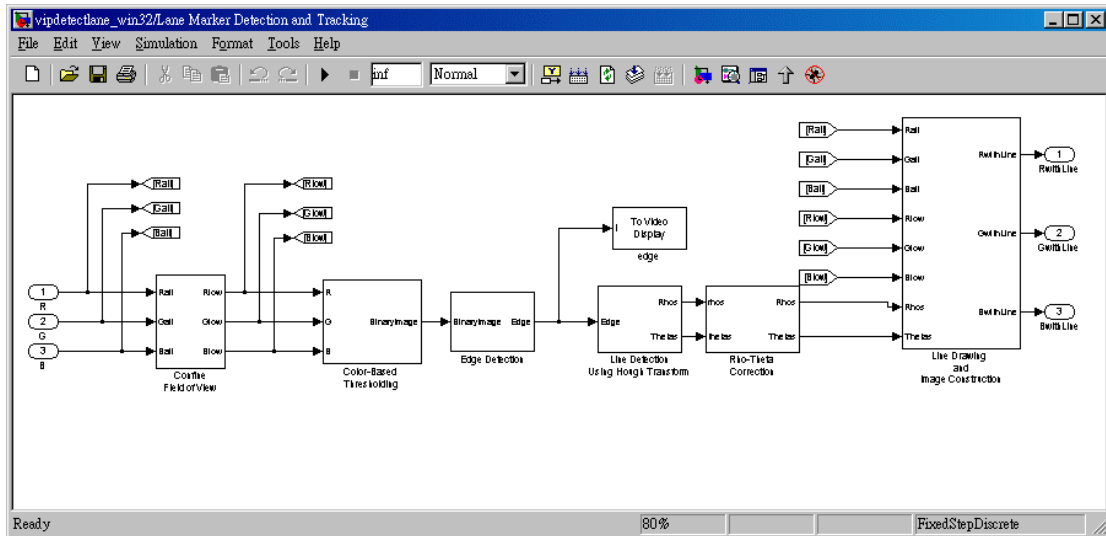
◆主要演算法「Lane Marker Detection and Tracking」內部程式解說

此程式中，又可分為下列部分：

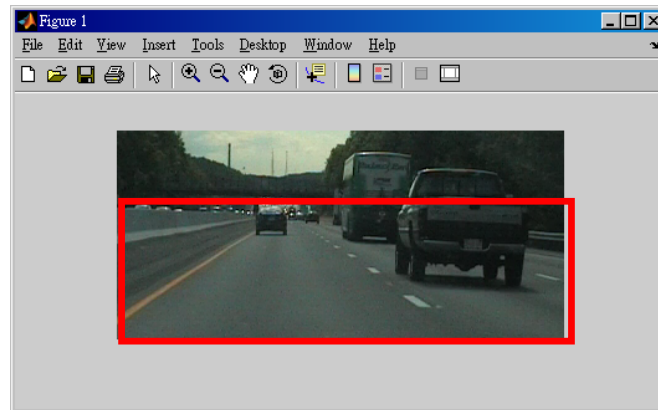
- (1) 色彩空間（Color Space）由三原色 RGB 轉為 YCbCr，並將影像二值化
- (2) 經過轉換的影像資料做邊緣偵測（Edge Detection）
- (3) 霍氏轉換（Hough Transform）
- (4) 在原始影像檔案上加上霍氏線性直線（Hough Lines）的標示

以上步驟，可經由點選進入主要演算法模塊後（圖四）清楚得知。

由「Read AVI File」的模塊可知，影像資料的像素（pixel）為 168x360；讀取 viplane.avi 檔案中的第一張影像資料觀察（圖五）可以發現，與車道辨識相關的影像資料只分佈於影像的下半張（框起部分）。因此，可將與車道辨識實際相關之影像資料擷取出來處理與辨識。擷取出用於辨識的影像資料範圍為：影像矩陣中第 81 列至 136 列（Row），第一行至最後一行（Column）。



圖四 車道辨識與追蹤之主要演算法 Simulink Model

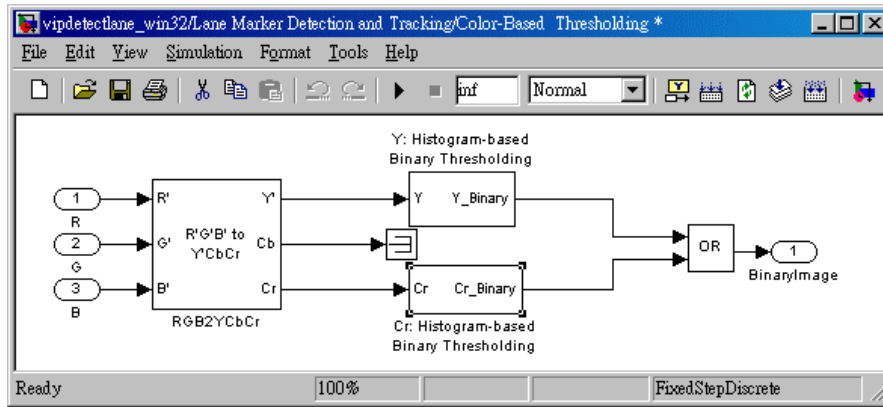


圖五 原始檔案中第一張影像資料圖

(1) 色彩空間 (Color Space) 由三原色 RGB 轉為 YCbCr，並將影像二值化

將色彩空間為 RGB 三原色的影像資料讀進至「RGB2YCbCr」的模塊，此模塊支援三種色彩空間的轉換，分別為：RGB 轉 YCbCr、YCbCr 轉 RGB、RGB 轉 intensity 三種。由於 RGB 色彩上彼此的相關性太大且與亮度有關，用於影像辨識效果並不如 YCbCr 的色彩空間，因此先將三原色的色彩空間轉換後再做後續演算法處理。

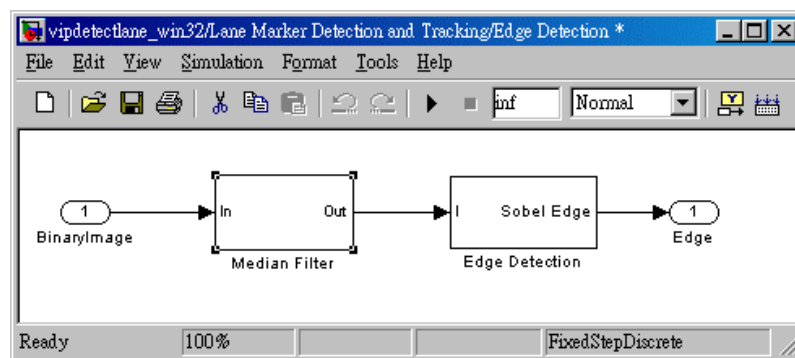
為了讓後續的邊緣偵測處理更有效果，在將色彩空間轉換為 YCbCr 後，再轉為二值化的影像；利用轉出的 Y (亮度) 與 Cr (色度，紅色)、忽略 Cr (色度，藍色) 後，以統計分析的方式找出兩者轉為二值化的最佳門檻後，轉為二值化影像資料如圖六所示。



圖六 影像資料 RGB 三原色轉二值化影像之演算法 Simulink Model

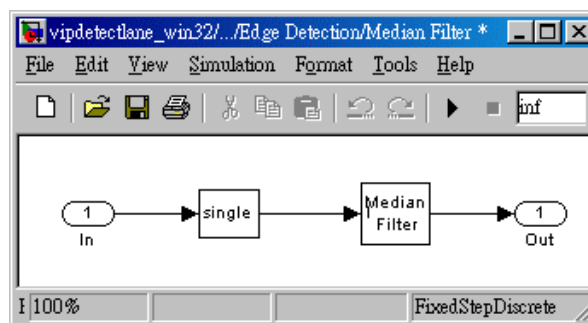
(2) 經過轉換的影像資料做邊緣偵測 (Edge Detection)

先將轉換過的二值化影像資料 (Binary Image) 讀入，經過一個中通濾波器 (Median Filter) 處理後，接續做邊緣偵測處理，然後輸出。如圖七所示。



圖七 二值化影像邊緣偵測之演算法 Simulink Model

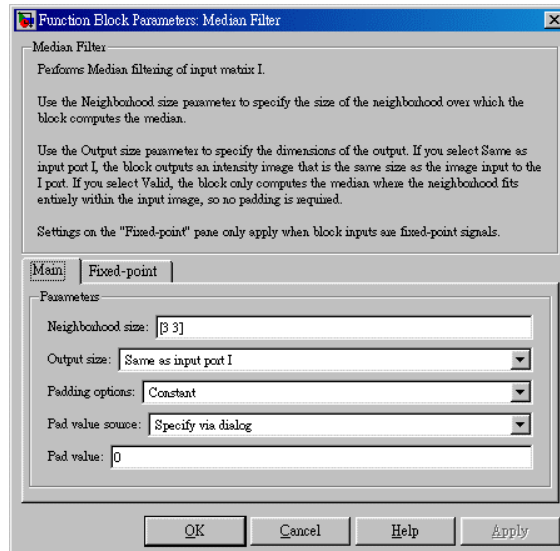
圖八為圖七中，中通濾波器模塊的內部程式。由程式可知：二值化的影像資料在處理之前，先將其輸入的影像資料型態轉成 single，再由中通濾波器處理，其功能為降低影像中雜訊的干擾，以利於後續之車道偵測；然後輸出。



圖八 中通濾波器之演算法 Simulink Model

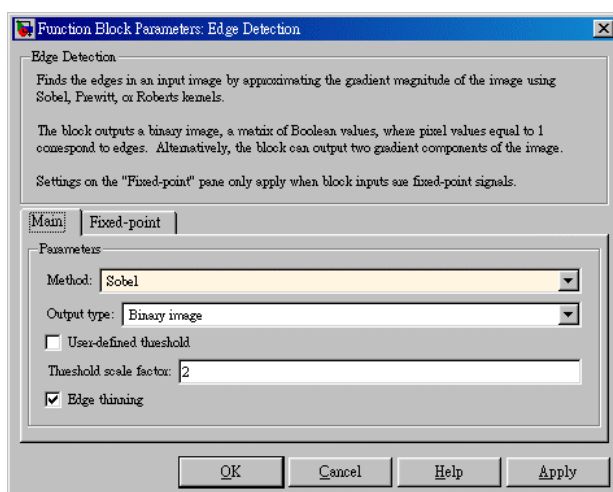
所謂的中通濾波器 (Median Filter) 是指在有雜訊的影像資料矩陣中，找一個適當大小的濾

波矩陣 (Neighborhood size)，將此矩陣中所有像素值依其大小進行排列；然後，以此矩陣中的中位數取代此矩陣中的所有像素值。如此對影像矩陣中每一像素順序進行。若濾波矩陣總元素為一偶數值，則計算中間兩個像素值的平均值以使用。因此，中通濾波器能很平均地將影像資料的重點表示出來。範例中所選取之濾波矩陣為一個 3x3 的矩陣。如圖九所示。

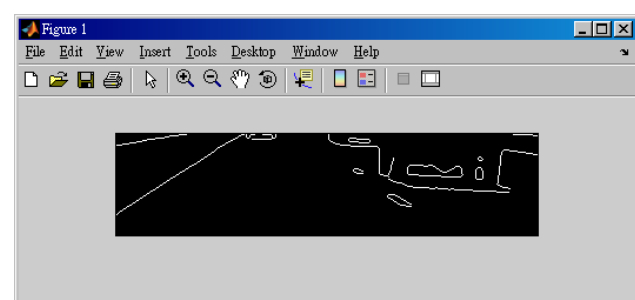


圖九 「Median Filter」模塊參數設定視窗

將已經降低雜訊的二值化影像資料做邊緣偵測。Simulink Video and Image Processing Blockset 提供三種邊緣偵測的演算法：(1) Sobel (2) Prewitt (3) Roberts。範例中所使用的為第一種：Sobel。邊緣偵測亦可調整經其處理後所輸出的資料型態及 Threshold scale factor，不同的 Threshold scale factor 將影響邊緣偵測輸出的邊緣線條顯示。Threshold scale factor 表示在此參數以下的影像資料將被當作雜訊而濾去。因此，Threshold scale factor 設定越高，被當作雜訊而濾掉的影像資料越多；若將 Threshold scale factor 調整過高，將無法顯示邊緣偵測的結果。參數調整如圖十，邊緣處理過後的影像如圖十一所示。



圖十 「Edge」模塊參數設定視窗



圖十一 二值化影像邊緣偵測示意圖

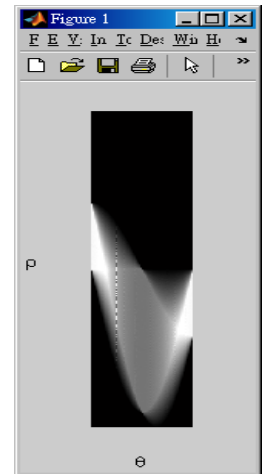
(3) 霍氏轉換 (Hough Transform)

霍氏轉換是一種找尋二值化影像檔案中線性直線的轉換式。其原理是將 $x-y$ 座標平面上的線性直線轉換成 θ - ρ 座標平面上的點。假設在 $x-y$ 座標平面上有兩條線性直線 $y_1=mx+b$ 與 $y_2=ax+c$ ；而 θ - ρ 座標平面上之 θ ，為原 $x-y$ 座標平面上直線之斜率， ρ 則為原 $x-y$ 座標平面上直線之截距，則原本的 y_1 與 y_2 兩條直線將在 θ - ρ 座標平面上將轉為 (m,b) 與 (a,c) 兩個點。

Simulink 中 Hough Transform 將會轉換出三項參數：Hough 矩陣、 θ 向量與 ρ 向量，其中 Hough 矩陣即為 θ 向量與 ρ 向量的集合。Hough 矩陣之圖示如右圖十二。

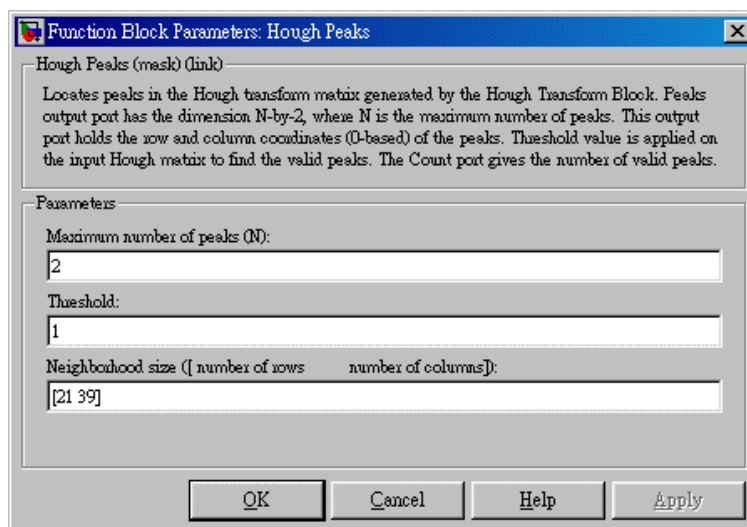
因此，由圖十二可知，若圖形中交點越多就表示在原始影像資料中，相同或接近的斜率與截距的線性直線越多。觀察 *viplane.avi* 影像檔案中，最明顯的為左下方的車道邊界連續直線，以及右下方車道與車道間分隔之間斷直線；由此可知，這些線段在經過霍氏轉換後，必定會因為相同的斜率與相近的截距而大量的交會於一點或其附近成為峰值 (Peaks)。

反之，找出霍氏轉換的峰值即可找到車道的邊線。



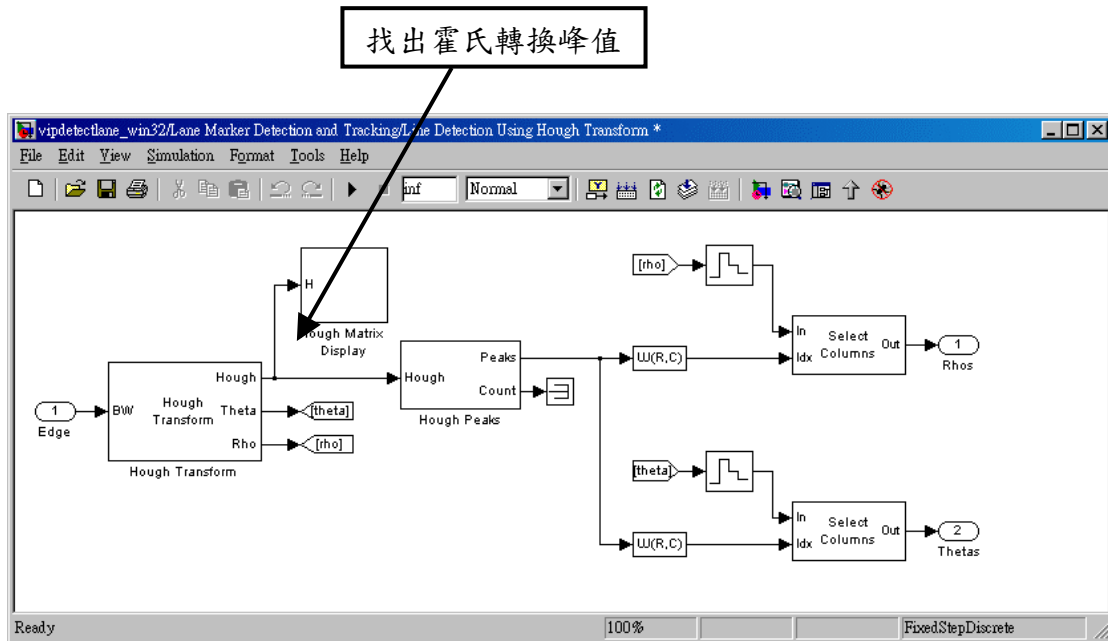
圖十二 霍氏矩陣表示圖

所謂的霍氏峰值 (Hough Peaks) 即為霍氏矩陣中 (如圖十一) 交集密集的点。而 Maximum numbers of peaks (N) 則是選擇霍氏峰值的個數。當 Maximum numbers of peaks (N) 為 1 時，得出的即為霍氏矩陣中交集最頻繁的直線；若 Maximum numbers of peaks (N) 為 2 時，則會找出交集最頻繁與次頻繁的兩條直線。由於車道辨識所要求的直線為車道左右兩側的邊界線與分隔線，因此霍氏峰值選擇為 2，以找尋此兩條車道線。參數設定如圖十三所示。



圖十三 「Hough Peaks」模塊參數設定視窗

由於車道的邊線只有左右下方各一條，因此霍氏轉換的峰值只需要找出兩個，便等於找到欲偵測的車道邊界。



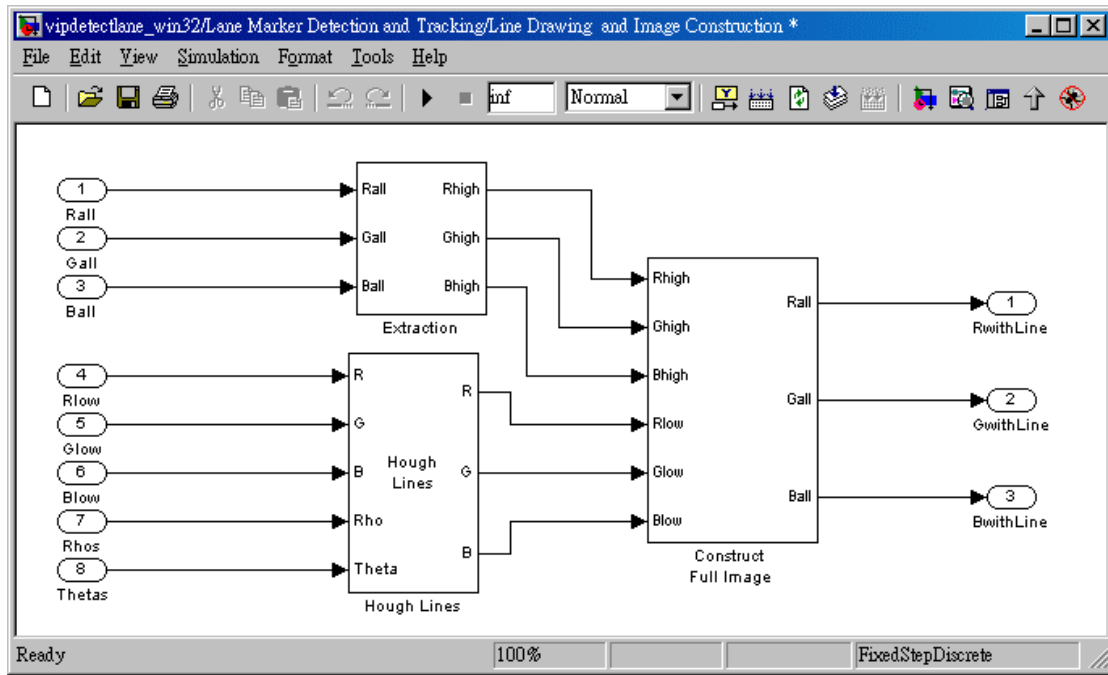
圖十四 霍氏轉換與霍氏峰值之演算法 Simulink Model

圖十四則為影像檔案經過邊緣處理後經霍氏轉換及霍氏峰值找尋之演算法。

(4) 在原始影像檔案上加上霍氏線性直線 (Hough Lines) 的標示

嚴格來說，如圖十五的次系統 (Subsystem) 並不屬於演算法的一部份，目的只是為了把找到的車道線標註在原始的影像之上，以供結果的觀察與顯示。

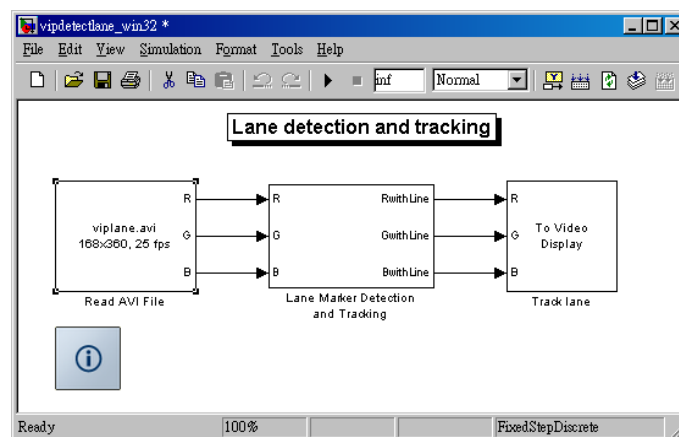
在霍氏轉換式得到的霍氏矩陣中所找出的兩個峰值即為兩條線性直線，Simulink 中「Hough Lines」的模塊即可將已知的 theta 與 rho 值直接與做霍氏轉換相同維度的原始影像資料合併。由於做霍氏轉換原始影像資料是經過割取的，因此必須將分割出的原始影像資料檔案加回，才可得出與 viplane.avi 原始檔案相同亦加入已求取之霍氏線性直線之影像資料檔案。其演算法如圖十五所示。



圖十五 分割影像與加入霍氏直線影像資料疊合之演算法 Simulink Model

◆ Simulink 中播放影像資料檔案

Simulink Video and Image Processing Blockset 中有「Sink」的模塊群組，其中可選擇「To Video Display」的模塊。此模塊支援 RGB 色彩空間的 video 檔案播放。其圖示如圖一最右邊的模塊。



圖十六 車道辨識與追蹤 Simulink Model