

學生：郭峰豪

指導教授：許鉅秉 教授

國立交通大學交通運輸研究所碩士班

### 摘要

駕駛在進行跟車行為時，往往會受到周遭環境的影響，而過去的文獻大多只針對和前方車輛的相互關係來改變跟車行為。而本文嘗試引用以駕駛者的視覺資訊帶給內心刺激來改變駕駛行為的觀點之文獻，並定義解釋其中心理質量的意義，並以實際資料進行模式參數的校估和驗證。而驗證結果也在一穩定的誤差範圍內。

本模擬模式主要是建立一直線三車道道路，有大型車和小型車兩種車種，再以建構出之模擬模式設立幾種情境進行模擬，模擬的結果可發現，模式可以解釋當駕駛行駛時，鄰近車道車輛的有無造成駕駛的分心，的確會對於整體車流的速度和間距有顯著的影響，又模式可解釋當車輛的車種大小不同時，會對於整體車流的速度造成明顯的影響。

關鍵字：跟車行為

The construction of simulator for car following behavior  
using quantum optical flow theory

student : Fong-Hao Kuo

Advisor : Dr. Jiuh-Biing Sheu

Institute of Traffic and Transprotation  
National Chiao Tung University

ABSTRACT

Drivers are often affected by the surrounding environment when driving. But most existing car-following models assume that the following vehicle is affected by the front vehicle. This study uses the car following model based on quantum optical flow. Define and explain the psychophysical mass for the drivers. And this study constructs the car following simulator. The simulation model is calibrated and tested with the video-based car following data. The simulation results are as follows : The models can explain the drivers' distraction which would affect the speed and spacing of the flowing traffic. And can explain when the car size which would affect the speed of the flowing traffic.

Keywords: Car following behavior,

## 致 謝

在學校經過兩年的學習，由衷感謝指導老師許鉅秉教授的指導，在論文撰寫的過程中，提供許多建議供我參考，並且不只是學業對於其他各方面的關心與鼓勵，使我受益良多。感謝胡守任老師和盧宗成在論文口試期間耐心的指導以及鼓勵，讓我瞭解了自己研究和思考上的不足之處，讓我的論文能夠更加的完整。

感謝交通大學交研所的每一位老師、所辦親切熱心的柳小姐和何小姐、電腦中心的鄭小姐與楊小姐、圖書室的陳小姐、以及班上一起度過這兩年每一位同學與學長姊、學弟妹們。

感謝我多年來的好友扣子、象爺、洪爺、豬毛、小林、偉梁，在那麼趕的時間二話不說車就開出來陪我收集資料，讓我有數據能夠完成我的論文。再感謝其他沒有列舉出來的朋友們，這段時間陪我一起玩樂、吃飯、抱怨，有你們的陪伴讓心理感受到論文的壓力也舒緩了許多。

最後要感謝的是我的父母，他們的關心、養育以及鼓勵讓我的生活能夠不虞匱乏，能夠專心的在自己的學業上，感謝你們這麼多年來的辛勞。

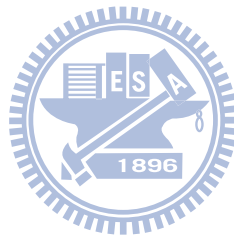
郭峰豪 謹誌于交通大學交通運輸研究所

民國一〇一年六月

# 目錄

中文摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究範圍.....	2
1.4 研究方法.....	2
1.5 研究內容.....	3
1.6 研究流程.....	4
第二章 文獻資料回顧.....	5
2.1 跟車行為.....	5
2.1.1 四大限制方程式.....	5
2.1.2 刺激反應方程式.....	8
2.1.3 行為門檻模式.....	10
2.1.4 二維座標車流模擬模式.....	13
2.2 駕駛人視覺特性與視覺行為.....	14
2.2.1 視覺特性.....	15
2.2.2 視覺行為.....	16
2.3 量子視覺流跟車模式.....	20
第三章 跟車模式邏輯建構.....	22
3.1 系統基本假設.....	22
3.2 跟車模式邏輯.....	22
3.2.1 駕駛視覺區域之計算.....	23
3.2.2 視線邊界和車輛權重之計算.....	24
3.2.3 跟車模式加速度計算.....	26
第四章 程式模擬系統之發展.....	29
4.1 模擬程式架構.....	29
4.2 程式撰寫.....	30
4.3 模擬環境之設計.....	36
第五章 參數校估及模擬結果分析.....	38
5.1 參數校估.....	38

5.2 模式驗證.....	40
5.3 模擬結果情境分析.....	41
第六章 結論與建議.....	55
6.1 結論.....	55
6.2 建議.....	55
參考文獻.....	57
附錄.....	59



## 表目錄

表 3-1 車輛速度和駕駛視線角度關係表.....	24
表 4-1 程式函數說明表.....	31
表 4-2 車量尺寸資料表.....	37
表 5-1 模式參數校估結果.....	42
表 5-2 高流量三種車種比例平均速度 ANOVA 檢定結果.....	47
表 5-3 中流量三種車種比例平均速度 ANOVA 檢定結果.....	47
表 5-4 低流量三種車種比例平均速度 ANOVA 檢定結果.....	47
表 5-5 高流量三種車種比例平均速度多重比較結果.....	48
表 5-6 中流量三種車種比例平均速度多重比較結果.....	48
表 5-7 低流量三種車種比例平均速度多重比較結果.....	48
表 5-8 高流量三種車種比例平均車間距 ANOVA 檢定結果.....	49
表 5-9 中流量三種車種比例平均車間距 ANOVA 檢定結果.....	49
表 5-10 低流量三種車種比例平均車間距 ANOVA 檢定結果.....	49
表 5-11 高流量三種車種比例平均車間距多重比較結果.....	50
表 5-12 中流量三種車種比例平均車間距多重比較結果.....	50
表 5-13 低流量三種車種比例平均車間距多重比較結果.....	50
表 5-14 高流量一般狀況和無旁車每五秒平均速度檢定和平均車間距檢定.....	53
表 5-15 中流量一般狀況和無旁車每五秒平均速度檢定和平均車間距檢定.....	53
表 5-16 低流量一般狀況和無旁車每五秒平均速度檢定和平均車間距檢定.....	53
表 5-17 一般狀況和無旁車每五秒平均速度和平均車間距檢定整理.....	54

## 圖目錄

圖 1-1	研究流程圖.....	4
圖 2-1	同一車道前後車廂對關係圖.....	11
圖 2-2	行為門檻關係圖.....	12
圖 2-3	波的疊加及波長與縱深關係圖.....	19
圖 3-1	目標駕駛視線範圍示意圖.....	23
圖 3-2	鄰近車道視線邊界示意圖.....	25
圖 4-1	車輛行為判斷流程圖.....	30
圖 4-2	主程式 main() 流程圖.....	34
圖 4-3	副程式 check_car() 之一般跟車行為流程圖.....	35
圖 4-4	模擬高速公路幾何特性示意圖.....	37
圖 5-1	台北市仁愛路資料收集路段.....	38
圖 5-2	模擬和實際速度平均絕對離差圖.....	41
圖 5-3	三種警覺心模擬結果速度比較圖.....	42
圖 5-4	三種警覺心模擬結果加速度比較圖.....	42
圖 5-5	一般情況低流量時空圖.....	44
圖 5-6	一般情況中流量時空圖.....	44
圖 5-7	一般情況高流量時空圖.....	44
圖 5-8	高流量三種車種比例每五秒平均速度與平均間距比較圖.....	46
圖 5-9	中流量三種車種比例每五秒平均速度與平均間距比較圖.....	46
圖 5-10	低流量三種車種比例每五秒平均速度與平均間距比較圖.....	46
圖 5-11	高流量一般情況和無旁車之平均速度和車間距比較圖.....	52
圖 5-12	中流量一般情況和無旁車之平均速度和車間距比較圖.....	52
圖 5-13	低流量一般情況和無旁車之平均速度和車間距比較圖.....	52

# 第一章 緒論

## 1.1 研究背景與動機

近年來世界各國由於科技發展、經濟快速成長，使得機動車輛數量快速的增加普及，也因此伴隨著運輸安全、交通壅塞、能源消耗及環境汙染等問題的產生。為了上述問題並且增進旅客與貨物之流暢與經濟生產力，皆致力於發展「智慧型運輸系統」(Intelligent Transportation Systems, ITS)。由中華智慧型運輸系統協會網站得知，增進交通安全、降低環境衝擊、改善運輸效率及提升經濟生產率為發展 ITS 的主要四大目標。在智慧型運輸系統的子系統「先進車輛控制安全系統」(Advanced Vehicle Control and Safety System, AVCSS) 係結合感測器、電腦、通訊、電機及控制技術應用於車輛及道路設施上，協助駕駛人提高行車安全性，增加道路容量，減少交通擁擠。其相關技術包含：防撞警示系統、自動停放車輛、車間與車一路間通訊、自動車輛診斷、自動橫向／縱向控制等。而由先進車輛控制安全系統的相關技術中可察知，許多技術的理論基礎會應用微觀車流理論中的跟車模式。

跟車行為模式是以一般傳統的駕駛跟車行為的研究上，大多假設車流中車輛駕駛的加減速僅會受到前車間的距離和相對速度影響的影響，而在實際上可能會受到其他因素影響。如駕駛在行駛時不只會受到前方車輛的刺激，在視線範圍內其他車道的車輛也可能會帶給駕駛刺激而改變行駛的速率。又當在相同的前車間距且相對速度一樣的情況下，因前車的車種不同而有不同的跟車行為，當前車是大型車輛時駕駛往往會因更強烈的不安全感，而放慢車速以拉大和前車的間距。

一般駕駛在開車時常常會受到前方或周遭車流或其他環境的變化而改變車輛行駛的速度，当前方車距離很近時，駕駛人可能會因感到不安全而放慢車輛的速度，或是在較為壅塞速度較低的車流中，駕駛會選擇貼近前車以防鄰近車道車輛超車。跟車理論就是在車流中沒有車道變換的情形下，利用各種微觀交通行為模式來解釋駕駛的加減速行為。



本研究以 J.B.Sheu(2011)所提出的以量子視覺流理論來解釋駕駛的跟車行為[1]為基礎，針對在一般三車道公路上行駛時的駕駛行為進行模擬與探討，期望在未來發展車流模擬器時能夠更為貼近現實。

## 1.2 研究目的

探討當車輛在多車道公路上行駛時，因視覺受到前車以及左右車流變化的刺激影響，使的駕駛人進行加減速，而在本研究的主要目的在於如何將此駕駛行為模擬模式建構出來，並且加以模擬分析。

對於模式內參數進行明確定義，利用程式語言進行模擬程式的撰寫，並且在經過情境分析以及和實際車流數據資料的驗證後，期望此研究能夠更加接近真實的駕駛跟車行為，能夠讓本研究的跟車行為模式提供於學術界做參考以及後續的研究使用。



## 1.3 研究範圍

本研究著重於高速公路路段上車輛的跟車行為，範圍界定於三車道高速公路的環境下，沒有交流道併入和支出的路段，研究路段為直線車道，車輛包含小客車和大型車。另外本研究不考慮車輛變換車道的行為以及路段中路旁其他干擾因素，且車輛不會有事故發生。

## 1.4 研究方法

使用心理動量來表示駕駛人受到視覺刺激時的感受，即駕駛視線內車輛對於駕駛的質量乘以相對速度，將駕駛人視線範圍內所感受到的車輛所帶來的刺激轉化為每位駕駛人內心所受到的心理動量，再藉由這心理動量來促使駕駛人進行車輛的加減速。駕駛在行駛除了要考慮前方車輛的影響，也必須注意左右鄰近車道車流所帶來的影響。以此為基礎建立模式並且撰寫程式，考慮幾種車流情境來進行車流模擬，接著再對模

擬結果進行分析。

## 1.5 研究內容

本研究之研究內容如下

- (1) 了解與確定研究問題及所需使用的研究工具與方法，並對於研究問題的範圍以及限制進行確認以及界定。
- (2) 蒐集國內外跟車理論和視覺衝擊對於駕駛人心理與行為影響的相關文獻和發展課題，進行整理和探討。
- (3) 根據 J.B. Sheu(2011)所提出之以量子視覺流理論解釋駕駛跟車行為，以此為基礎針對其模式以及想要模擬的情境，撰寫模擬程式。
- (4) 以實際跟車攝影資料作為跟車行為特性資料。
- (5) 以所收集到的資料對於模式進行參數校估及模式驗證。
- (6) 總結論文以及研究心得，並對於研究提出具體的結論和後續研究之建議，以便供後續研究、改進及延伸。



## 1.6 研究流程

本研究之研究流程如下圖所示。

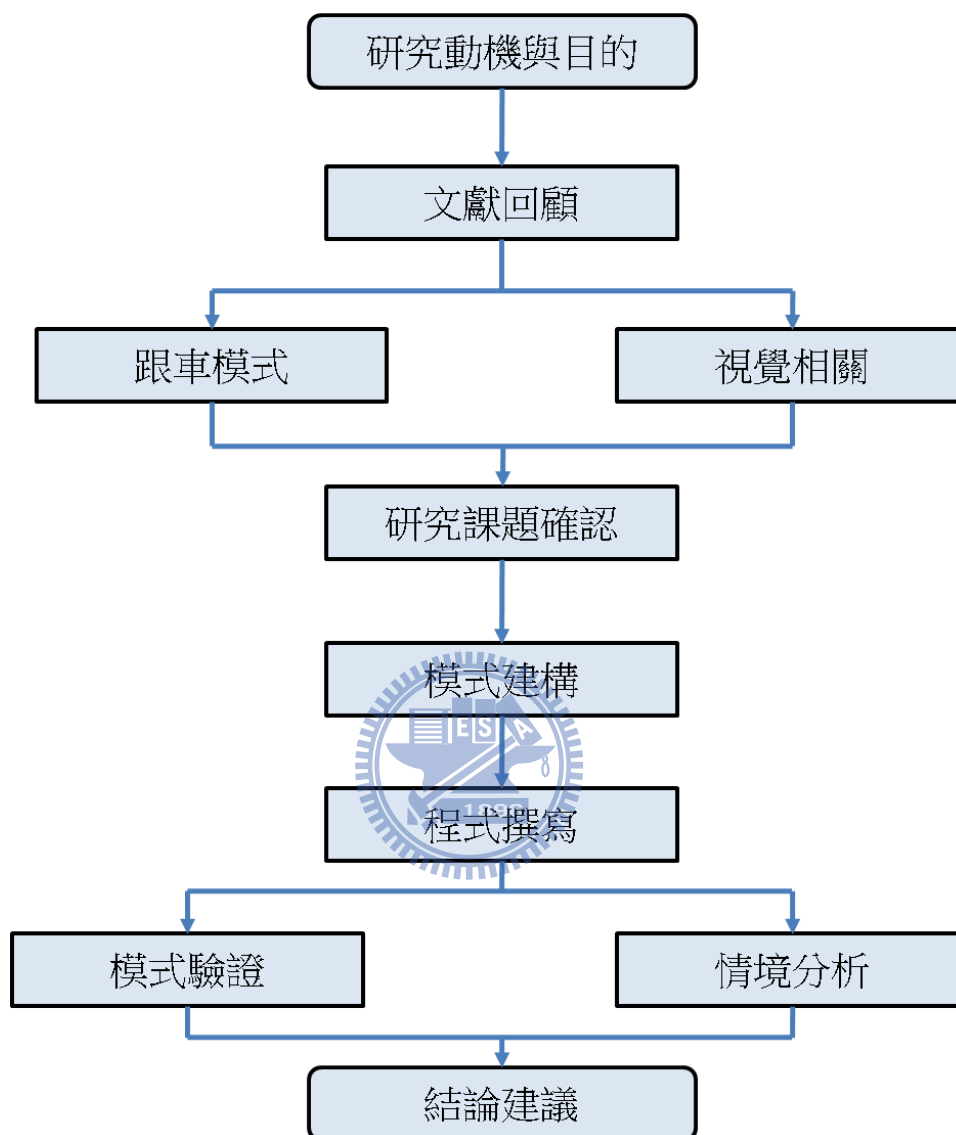


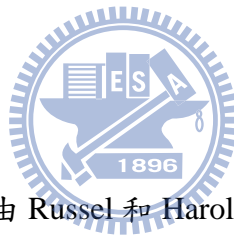
圖 1-1 研究流程圖

## 第二章 文獻回顧

本研究之文獻回顧，主要可分為三個部份進行。分別為回顧過去微觀車流理論中跟車模式的研究內容與發展，車輛行駛時駕駛的視覺特性以及視覺行為研究進行回顧，以及回顧本論文所使用之量子視覺流跟車模式。

### 2.1 跟車行為

跟車行為是在道路上行駛時車輛間的相互關係，後方車輛在行駛的過程中，為了維持和前方車輛的安全間距，不斷的隨著前車調整其行車速率和車輛間距，當突發事件發生時，讓後車能及時煞停且不會與前車發生碰撞。跟車模式的主要內容是當跟車行為發生時，前後車間的速度、距離及加減速等的互動關係，並將這些變數以方程式來加以描述。



#### 2.1.1 四大限制方程式

四大限制方程式於 1963 年由 Russel 和 Harold[5]所提出，指出觀測車輛在跟隨前車行駛的過程中，必須要調整其行車速率，以維持一最小安全距離，以使當發生突發事件時，觀測車輛能夠在前車車尾後方安全煞停，不至於頭尾相撞。此種車輛推進方式主要考慮四種限制：

首先定義最小安全間距  $S$ ，其指前後兩車輛車頭之最短距離，最小安全間距會和跟車速率成正比，當前車速率大於後車時，無需增加車輛間距，反之則須加長間距。公式如下：

$$S = P + K_1 V_F + \frac{K_2}{2D} (V_F - V_L)^2 * C \quad (1)$$

其中，

$P$  : 前車有效車長(前車車長加上車輛停止時的安全間距, 單位為公尺)

$K_1$  : 後車駕駛反應時間

$K_2$  : 常數, 當採 MKS 制時為 1 公尺/秒

$V_t$  : 後車車速(公尺/秒)

$V'_t$  : 前車車速(公尺/秒)

$\bar{D}$  : 後車之平均減速率(公尺/秒)

$C$  : 係數, 當  $V > V'$ ,  $C = 1$

當  $V \leq V'$ ,  $C = 0$

### (1) 間隔限制

間隔限制為兩前後行駛之車輛, 後車保持在安全間距下, 單位時間內所能向前推進之最大距離。當觀測車輛無前車干擾時, 則不需要考慮此項限制。


$$Z_s = X'_t - X_{t-1} - S \quad (2)$$

其中,  $X'_t$  :  $t$  時刻前車車頭位置(公尺)

$X_{t-1}$  :  $t-1$  時後車車頭位置(公尺)

$S$  : 最小安全間距(公尺)

### (2) 加速限制

加速限制為車輛在現有車速下, 單位時間內以期加速能力所能向前推進之最大距離。當車輛欲加速時, 需考慮此項限制。

$$Z_a = \frac{1}{2} [V_{t-1} + (V_{t-1} + \bar{A})] \quad (3)$$

式中,  $\bar{A}$  : 觀測車在車速為  $V_{t-1}$  時所能獲得的加速率( $m/s^2$ )

### (3)停止限制

停止限制是指當車輛於路口遇到紅燈或其他因素必須減速時，單位時間內所能行進的距離。

$$Z_d = \frac{1}{2}V_{t-1} - \frac{\bar{D}}{4} + \left( \frac{\bar{D}}{16} - \frac{\bar{D}}{4}V_{t-1} + \frac{\bar{D}}{2}X \right)^{\frac{1}{2}} \quad (4)$$

式中，X：觀測車至停止點的距離(公尺)

### (4)轉彎限制

車輛於路口轉彎時，為避免受離心力作用所造成的不舒適及翻覆的危險，在開始轉彎時會以較低的速率行駛，待通過轉彎點之後，再進行加速。首先計算車輛轉彎時的最大速率限制V，其公式如下

$$V = \sqrt{\mu g R} \quad (5)$$


其中， $\mu$ ：路面側向摩擦係數

$g$ ：重力加速度

$R$ ：車輛轉彎半徑(公尺)

當在(t-1)時刻轉彎車輛與轉彎點之距離為X時，則可將其分為下面兩種狀況來進行討論：

$$Z_t = \frac{1}{2}V_{t-1} - \frac{\bar{D}}{4} + \left( -\frac{1}{4}\bar{D}V_{t-1} + \frac{1}{16}\bar{D}^2 - \frac{1}{4}V^2 + \frac{1}{2}\bar{D}X \right)^{\frac{1}{2}} \quad (6)$$

此式適用於當t時刻車輛尚未通過轉彎點，且 $Z_t \leq X$ 的情況。

而當t時刻車輛已經通過轉彎點時，若在(t-1)時刻以加速率A加速至t所得到的速率大於轉彎速限V時，則轉彎限制推近距離為下式：

$$Z_t = X + \left( V_t + \frac{1}{2} AT^2 \right) \quad (7)$$

式中，T：為(t-1)秒至 t 秒間已過轉彎點之行使時間(0<T<1)

在考慮過四項限制條件後所得之最小行駛距離，即為車輛在單位時間內所能推進的距離，再依此來更新車輛坐標位置與車速。

### 2.1.2 刺激反應方程式

而在 1998 年 Leutzbach 指出四大限制方程式著重於車輛可推進之安全距離之計算，卻忽略了前後車間交互反應的互動關係。因此後續研究開始針對跟車行為進行探討，強調於跟車時，須經常調整其行車速度以便和前車保持一安全車間距離的動態行為。而此觀念乃建立於「刺激－反應」的基礎之上，故出現了所謂的「刺激－反應方程式」。

此類模式主要之觀念為：駕駛者的反應和來自於外界的刺激之間有函數的關係，當外界刺激越大則駕駛者之反應也越大，且此模式的基本假設不考慮駕駛者超車和變換車道的行為。模式的發展歷史最早可追溯至 1950 年代的 Pipes(1953)和 Forbes(1958)，而後陸續有許多學者投入研究此類模式，美國通用汽車研究群(GM group)為其中的代表。

Pipes 以安全距離的觀念來表示後車的車速與後車和前車間的空間成比例再加上一段距離的關係。此理論的缺點在於車輛在低速或高速的情況下會出現低估的情形，和現實並不相符。

$$d_{n+1}(t) = L_n + \frac{X_{n+1}(t)}{10 \times 1.47} \times L_n \quad (8)$$

$d_{n+1}(t)$ ：時間  $t$  時後車與前車間距(英尺)

$L_n$ ：車長(英尺)

$X_{n+1}(t)$ ：時間  $t$  時之速度(MPH)

Forbes 在跟車行為模式中加入反應時間，使後車用以判斷是否該減速煞車，一般認為前車之後緣與後車之前緣間的時間間隙至少應大於或等於反應時間，所以最小的時間間距的數學關係式等於反應時間加上前車車長距離之行駛時間。如下所式：

$$h_{\min} = \Delta t + \frac{L_n}{\dot{X}_n(t)} \quad (9)$$

式中，

$\Delta t$ ：反應時間(秒)

$L_n$ ：車長(英尺)

$\dot{X}_n(t)$ ：前車在時間  $t$  時之速度(MPH)



美國通用汽車的研究群利用統計方法，由現場照相調查資料的方式，並加入  $m, l$  矩陣建立刺激－反應方程式，隨後又經過許多人的研究修改，共發展出五代的 GM 模式。其中第五代模式可以寫為：

$$\ddot{X}_{n+1}(t + \Delta t) = \frac{\alpha_{l,m} \left[ \dot{X}_{n+1}(t + \Delta t) \right]^m}{\left[ X_n(t) - X_{n+1}(t) \right]^l} \left[ \dot{X}_n(t) - \dot{X}_{n+1}(t) \right] \quad (10)$$

其中，

$\ddot{X}_{n+1}(t + \Delta t)$ ：後車在  $(t + \Delta t)$  時之加減速度

$\dot{X}_{n+1}(t + \Delta t)$ ：後車在  $(t + \Delta t)$  時之速度

$X_n(t)$ ：前車在  $t$  時之距離



$X_{n+1}(t)$ ：後車在  $t$  時之距離

$m$ ：後車駕駛人對於速度反應的敏感度參數

$l$ ：後車駕駛人對於前後車間車間距反應的敏感度參數

$\alpha_{l,m}$ ：跟車敏感度參數

GM 模式主要是認為後車的駕駛者會因為前車給予的刺激，例如相對速度或相對間距的變化而進行反應，如車輛的加減速。在 GM 模式中假設每位駕駛人具有同質性，周遭環境刺激和駕駛行為反應有一個明確存在的關係。

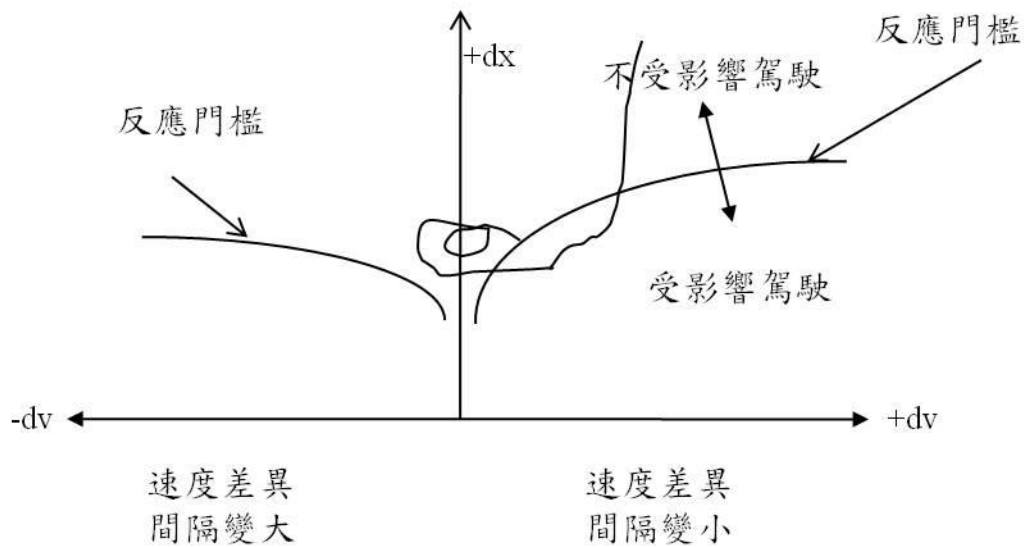
### 2.1.3 行為門檻模式

傳統的刺激反應方程式有兩項不合理的假設。第一點為不論前後兩車距離多遠，都會互相影響。第二點為當前後兩車相對速度不變時，跟車反應即不會改變。然而行為門檻模式是認為，當跟車駕駛感知前車與自身間距及相對速度差距過大時，會加速以接近前車縮短間距，並逐漸由不受影響駕駛範圍進入受影響駕駛範圍；

Hoefs(1974)[3]於實際的觀測中發現，當跟車駕駛者感知過於接近前車感到不安全時，則減速以加大間距；減速後若不符合跟車駕駛者的期望則再加速度，但若當觀測車輛與前車的差距過大時再完全不會受到駕駛者的影響。整個系統就在駕駛者不斷加減速的過程中達到穩定的跟車狀態。而此種駕駛行為所表現出的意向有兩點：

(1)跟車過程處於一微幅振盪調整之間距與速差系統中。

(2)在各反應狀態下，具有如拋物線型之感知門檻。



圖

2-1 同一車道前後車相對運動示意圖[3]

Wiedemann(1974) [4]將「心理—物理間距模式」(Psycho-Physical Spacing Model)觀念引進微觀車流模式中，建構一數學模式，即所謂的「行為門檻模式」。其特點為同時考慮速差與車間距離作為行為變化的決策條件，駕駛者根據路況中的時空變化，對應其決策門檻而進行跟車行為調整。模式基本假設亦為單一車道跟車行駛，不考慮變換車道情形，可將車流狀況分成三個反應區：感知反應區(Perceived Reaction)、無意識反應區(Unconscious Reaction)及無反應區(No Reaction)。了解各感知門檻界線區隔後，可再將其細分為各個決策行為區，如圖 2-2 所示。

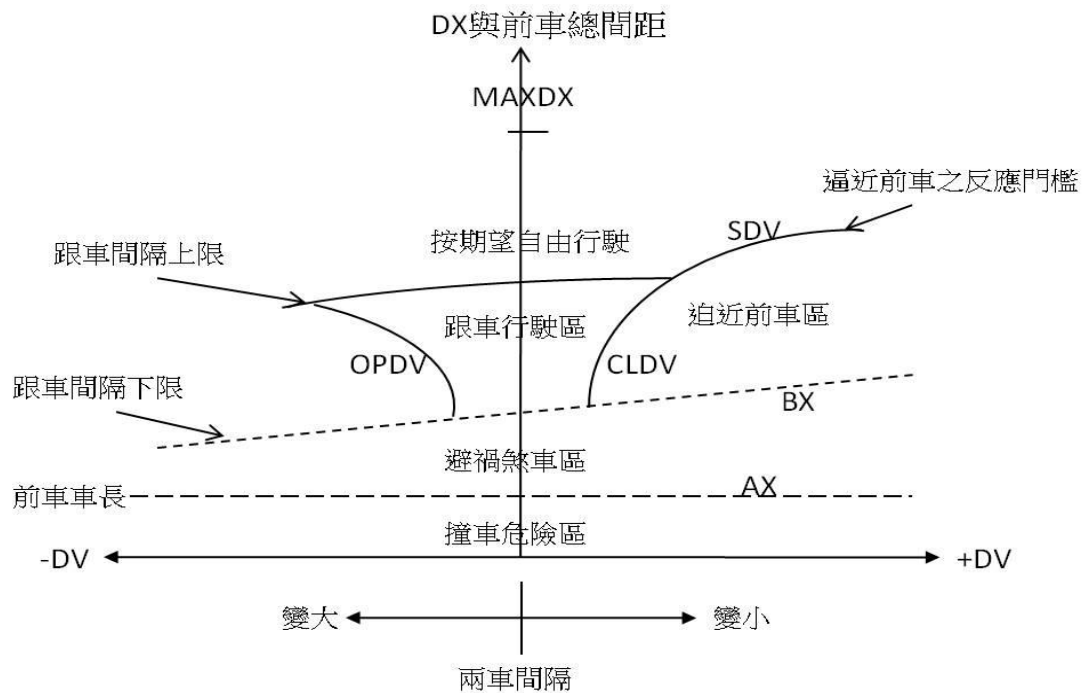


圖 2-2 行為門檻關係圖[4]

各行為門檻的定義如下：

1. 靜態間距( $AX$ )：車輛靜止時後車希望與前車車頭保持之距離。
2. 最小安全間距( $BX$ )：後車車速與前車相近時，欲維持之最小跟車間距。
3. 感知速差門檻( $SDV$ )：在一較大間距下，跟車駕駛者對於速率差異(後車車速減前車車速)之感知門檻。由後車目前間距( $DX$ )與兩個隨機因子(安全因子與估計因子)估計而來。
4. 跟車間距上線( $SDX$ )：為顧及駕駛者判斷間距能力之差異， $SDX$  約在 1.5~2.5 倍最小跟車間距範圍內振盪。
5. 間距漸減速差門檻( $CLDV$ )：在一較小間距間距漸減且速差為正之情況下，速度差異之門檻值，由  $SDV$  與隨機因子計算而得。
6. 間距漸增速差門檻( $OPDV$ )：在一較小間距間距漸增且速差為負之情況下，速度差異之門檻值，其值約為 1~3 倍  $CLDV$  值，此因駕駛者通常對正在遠離物比正在接近物反應較遲緩。

#### 2.1.4 二維座標車流模擬模式[11]

二維座標車流模擬模式是使用車輛可能前進範圍的概念，配合車道寬度限制、車輛動態尺寸限制與其他限制處理混合車流行為。

##### (1) 車輛可能前進範圍

由於混合車流無明顯的跟車行為，車輛不時以非直線方向前進；然車輛斜進角度在操作上因車種而不同，因此衍伸出最大偏向角之觀念。所謂最大偏向角是車輛行進中，在舒適及安全駕駛的限制條件下，單位時間所能偏轉的最大角度。在此模式中，機車是以等速率圓周運動實驗求得最小圓周半徑，汽車則使用公路設計之曲線超高公式求得曲率半徑，兩者在依餘弦定理求取最大偏向角。故車輛最大偏向角 $\theta_M$ 為

$$\theta_M = \frac{1}{2} \cos^{-1} \left( 1 - \frac{L_0^2}{2R^2} \right) \quad (11)$$

式中， $L_0$ ：車輛在速率下單位時間所行駛之弧長(m)

$R$ ：車輛在速率在容許舒適程度下之最小半徑(m)

根據最大偏向角即可定義車輛可能推進範圍為以加速限制 $Z_a$ 為半徑所展開之 $2\theta_M$ 扇形區域。

##### (2) 車輛動態尺寸

動態尺寸指車輛行進中，駕駛人所感受到其他車輛之有效影響長度與寬度，也可視為車輛占用道路之範圍。二維座標車流模式是採用車輛行進時之動態車長以控制前進距離，而動態車長則是依據跟車理論中「跟隨車輛在行駛過程中長調整其速率，使其能與前車保持一安全間距」，各車種動態車長如下式：

$$PLL = PL(TYPE) + K \cdot V_i \quad (12)$$

式中， $PLL$ ：前車之動態車長(m)

$PL(TYPE)$ ：前車之靜態車長，依車種而有所不同(m)

$K$ ：本車駕駛者的反應時間(sec)

$V_i$ ：本車車速(m/s)

動態車寬為車輛行進時之安全車寬，各車種之動態車寬如下式：

$$PWW = PW(TYPE) \cdot (1.0 + 0.2 \cdot V_i / 33.3) \quad (13)$$

### (3) 車輛推進限制

即計算單位時間內，在考慮下列限制條件下，各方向之最大推進長度。

#### a. 路寬限制

即控制車輛斜進時，動態車寬的右側不得超過路寬限制，此處路寬限制式指路面邊緣、中央分向線或車道線。

#### b. 前車限制

包括直向前進限制與斜向前進限制，即在前方與側方之動態車長與車寬影響下，重新計算各方向之最大推進長度。



## 2.2 駕駛人視覺特性與視覺行為

一般駕駛人乃使用視覺感受外面環境再調整自己的駕駛行為，故必須先瞭解人類的視覺與駕駛車輛的特性，底下引用陳俊鴻[8]所整理之視覺文獻。「視覺」(Visual)是人類在接收外界訊息時使用最多的感覺類型，人類大部分的知識均藉由視覺系統而獲得，由於人類雙眼所能見到的視界角度約在 $180^\circ$ 以內，而駕駛人的視力(Visual Acuity)係隨著離視軸角度(視錐角)之不同而異，且在水平及垂直面上的視覺效果亦不同。在水平視錐角方面，視錐角度愈大，駕駛人雖能看得更廣，但不能看得清楚，即視覺辨識力變得較差。駕駛人能看到最清晰的部分是在水平視錐角 $3^\circ \sim 5^\circ$ 之內；在垂直

視錐角方面，其明視狀況僅水平視錐角之1/2~1/3 而已。當角度增加至10 度以上時，對於景象即不能明確地辨識。

車輛行駛過程中，駕駛者需要不斷接收資訊以維持車輛正確與安全的行進，而駕駛者本身擁有不同的感應器，可用來接收在人— 車內部或人— 車外部產生的資訊。內部資訊來自於駕駛者的記憶，或是系統回饋(反應)的結果；外部資訊則藉由視力、聽覺與觸覺取得。其中，外部資訊有90%是利用視覺取得，並且是維持行進方向、偵測障礙物或從號誌、標誌與標線中取得資訊的唯一管道。

### 2.2.1 視覺特性

#### (1) 視力及視野範圍

駕駛狀態中有90%以上之訊息接收是來自視覺，其餘則是由聽覺或觸覺接收。在閱讀狀態下，人類的視角相當窄，約 $3^{\circ}$ ~ $10^{\circ}$ 之間，而在此視圓錐角之外的影像則是由周邊視覺搜尋。靜止狀態下周邊視覺的範圍是：左右各 $90^{\circ}$ 、水平視角以上 $60^{\circ}$ 、水平視角以下 $70^{\circ}$ 。若是在有速度的狀態下，速度30kph 時，雙眼視界角度約為 $100^{\circ}$  度；速度60kph 時，雙眼視界角度約為 $70^{\circ}$  度；當速度為100kph時，雙眼視界角度則為 $40^{\circ}$  度。另外，視覺的焦點(雙眼明視焦距)也與駕駛人視覺及車輛速度有關，視覺焦距將因速率的增加而距離愈遠。因此駕駛人在行駛的過程中，必須不斷移動眼球注視或搜尋，並配合頭部運動來接收各種視覺的資訊。

#### (2) 眼球與頭部之運動

資訊獲得之主要限制條件是眼睛視覺的焦距由一物移轉至另一物之速度。經由研究發現，眼球最多一秒可左右搜尋四次，但在經過一段時間之後，左右搜尋之速度會稍微減緩，約減低至一秒二次。而在一般駕駛狀態下，每秒1 次至1.5 次是較合理之數值。故在交通工程之設計上，標誌標線或號誌之間，應分開有一相當之距離。如車

輛以時速100 公里行進，而視線搜尋之頻率為每秒1.0 至1.5 次，則駕駛人每20 到28 公尺會看見一個標誌、標線或號誌。若是兩交通工程設施之距離小於上述數值，則駕駛人便會錯失該項訊息。雖然眼球左右移動可超過50度，但於一般駕駛狀態中，駕駛人只傾向左右移動眼球各15 度。

### (3) 道路環境對視覺之影響

駕駛者面對複雜的道路環境與交通狀況，不僅需根據道路環境適切地操控自己車輛，同時需注意其他車輛的動向及其他一切可能的突發狀況，以做出及時的反應。但由於視覺能力有限制，無法接收所有可能與行車訊息有關之資訊。即使駕駛者看到了，可能由於未激發注意力之調度，或因視覺負載過重而未予處理，因而錯失了重要的訊息而發生事故。



### 2.2.2 駕駛人視覺行為

當眼球視軸投注在某一固定點上時稱為視覺焦點或凝視。眼球凝視之定義為—用中心視線所注視之物體，當專注於一物體或狀況時，人通常會移動其眼球之注視方向，使影像落於視網膜的中心窩位置，因此當某物被注意時，亦在中心窩凝視下。另外一種定義為—凝視點是駕駛者凝視其眼睛在一物體之特徵上或其他環境之短暫延時(大約是0.1~0.5 秒)，而此時視覺能獲得中心窩視線的最大解析力之駕駛資訊。眼睛凝視點是指駕駛者眼睛落點之移動。在分析眼睛凝視點是否為同一目標物時，須視下列兩因素而定：

- (1) 鄰近的兩個可能眼睛凝視點在一定之時間內；
- (2) 在視覺面上介於兩個可能是眼睛凝視點的一定距離長度。

但當駕駛人眼睛落點之移動，能夠充分說明中心窩視線已移去另一不同之目標物時，則可視為另一新的凝視，而不論上述之延時或距離之限制。在上一節中曾提及雙眼明視之範圍，在0 度時其視力可達最佳，此時即為一般我們所測之視力大小，亦即



視覺焦點所在位置。但在一般道路上駕駛的過程中，視覺焦點外之週邊視覺亦扮演重要的角色。其主要工作為協助中心窩之凝視偵測，並注意視線中心外圍之狀況，當有需要時，引導中心視力(即凝視)去注視，以明辨狀況並決定是否應採取適當反應。而有經驗之駕駛人會利用其外圍之視覺來判定其在車道之位置。

因為以往的視覺文獻皆以實驗的方式來量測駕駛人駕駛中環境對反應或是行為的影響，但沒有完整的理論去對環境和行為做個適當的說明，Robert G.V.Baker(1999)利用量子力學建構視覺流，更近一步的提出視覺流與駕駛行為的不確定性關係，以及利用量子力學中海森堡不確定性原理來解釋人類對速度的誤判。故本文中將會採用Baker的部分模型架構來建構視覺衝擊對人類行為的影響，茲將部分模型架構整理並解釋如下。

依據量子力學視覺流理論，一個人的視野的縱深(X)寬度(Y)和對周遭的相對速度呈現以下特定關係

$$(\Delta x(t))(v_i(t)) = C_x, \forall t \quad (14)$$

$$(\Delta y(t))(v_i(t)) = C_y, \forall t \quad (15)$$

其中， $\Delta x(t)$ 和 $\Delta y(t)$ ：表示駕駛者在時間 $t$ 時視線焦點縱深和寬度的標準差

$v_i(t)$ ：表示在時間 $t$ 時目標車輛 $i$ 沿著 $x$ 方向的行駛速度

$C_x$ 和 $C_y$ ：為兩個分別和 $X$ 和 $Y$ 方向有關的常數

在式(14)(15)中表現出當與周遭環境的相對速度越大，駕駛者的視野縱深和寬度則會越小，同時專注視線也就越集中。相反的當車輛在低行駛速率下會讓增加，導致駕駛的量子光場的不確定性提高。

當目標車輛在車流中駕駛的視線範圍內有其他車輛 $J_Q$ 存在，經過視覺流的影響，視線範圍內的每輛車 $j_q$ 在時間 $t$ 時，會對於觀測車輛駕駛給予其心理動量 $\Phi_{j_Q}(t)$ (psychophysical momentum)和心理動能 $\theta_{j_Q}(t)$ (psychophysical energy)，如下所示：



$$\Phi_{j_Q}(t) = m_{j_Q} \times v_{j_Q \rightarrow i}(t) \times W_i(t), \forall j_Q \in J_Q \quad (16)$$

$$\theta_{j_Q}(t) = \frac{m_{j_Q} \times [v_{j_Q \rightarrow i}(t)]^2 \times W_i(t)}{2}, \forall j_Q \in J_Q \quad (17)$$

其中， $J_Q$ ：在時間 $t$ 時在目標駕駛視線範圍 $Q[\Delta x(t), \Delta y(t)]$ 內的所有車輛

$m_{j_Q}$ ：表示駕駛感受到的淨值量

$v_{j_Q \rightarrow i}(t)$ ：表示在時間 $t$ 時目標車輛 $i$ 與視野內車輛 $j_Q$ 的相對速度

$W_i(t)$ ：表示時間 $t$ 時目標駕駛的警覺心

再接著將心理動量和心理能量轉換為波的形式，也就是波的波長以及振幅，如下

所示：

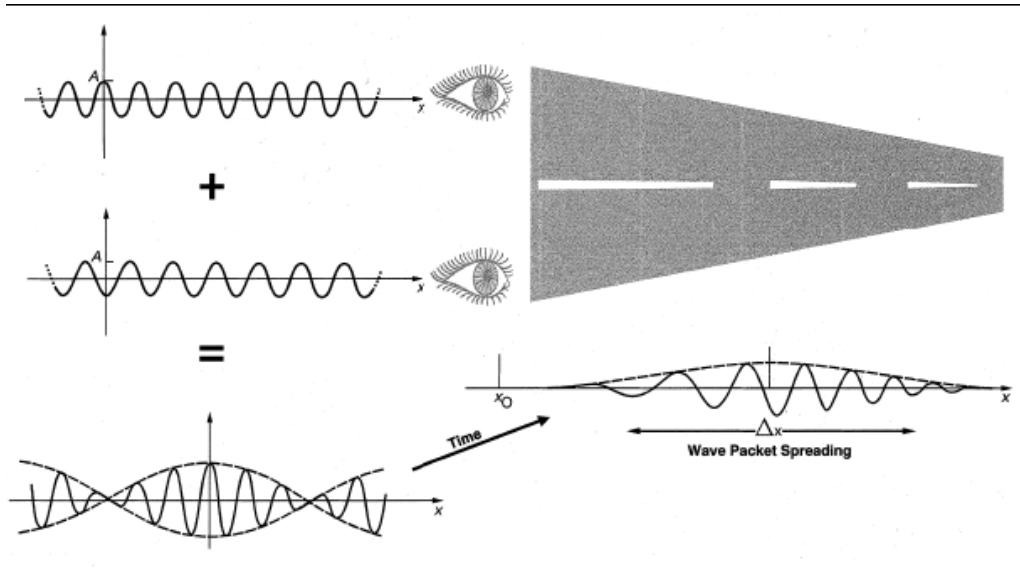
$$\omega_{j_Q}(t) = \frac{h}{\Phi_{j_Q}(t)}, \forall j_Q \in J_Q \quad (18)$$

$$f_{j_Q}(t) = \frac{\theta_{j_Q}(t)}{h}, \forall j_Q \in J_Q \quad (19)$$

其中， $\omega_{j_Q}(t)$ ：表時間 $t$ 時車輛對目標駕駛造成的動量轉換成的波長

$f_{j_Q}(t)$ ：表時間 $t$ 時車輛對目標駕駛造成的能量轉換成的振幅

$h$ ：為一動態常數



資料來源：Baker(1999)[3]

圖 2-3 波的疊加及波長與縱深關係圖

經過以上過程可以將駕駛視野範圍內的每一部車輛，對於駕駛所造成的感受轉換成一條波的形式，一輛車轉換成一條波，並將所感受到的每一條波疊加形成一波包，且 Baker 亦提及波包的波長  $\omega_{jQ}(t)$  和視野縱深  $\Delta x(t)$  成以下關係：

$$\Delta x(t) = \frac{\omega_{jQ}(t)}{2} \quad (20)$$

並將(20)代入(14)，可得到

$$\frac{\omega_{jQ}(t)}{2} (v_i(t)) = C_x \quad (21)$$

再將(18)代入(21)，可得

$$\frac{h}{2\Phi_{jQ}(t)} (v_i(t)) = C_x \quad (22)$$

由(22)式中得知，在駕駛者觀察其在視野範圍內，其周遭環境與車輛的相對速度

越大時，也代表著駕駛這內心受到的動量越大，此時駕駛者便會減速以緩和心理動量。

### 2.3 量子視覺流跟車模式

J.B.Sheu(2011)以 Robert G.V.Baker(1999)的理論為基礎，提出量子視覺流來解釋駕駛的跟車行為，此文獻為本研究的重要骨幹，其主要將駕駛視線範圍內車輛所帶來的心理動量和心理動能，轉換成駕駛行駛時進行的加減速行為。首先把觀測駕駛對於視覺範圍內每一點的專注力利用二維高斯波包( $\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]$ )來表示：

$$\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)] = \frac{e^{-\frac{1}{2} \left[ \frac{[x(t) - \mu_{Q_x}(t)]^2}{(\Delta x(t))^2} + \frac{[y(t) - \mu_{Q_y}(t)]^2}{(\Delta y(t))^2} \right]}}{2\pi \times \Delta x(t) \times \Delta y(t)} \quad (23)$$

其中  $\mu_{Q_x}(t)$  和  $\mu_{Q_y}(t)$  分別是高斯波包 X 和 Y 中央位置。

而對於視線範圍內的車輛所 ( $j_Q, \forall j_Q \in J_Q$ ) 造成的心理動量  $\Phi_{j_Q}(t)$ ，會因位置的不

同給予相對的權重  $\delta_{i, j_Q}(t)$ ，可寫作：

$$\delta_{i, j_Q}(t) = \frac{\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]}{\sum_{\forall j_Q \in J_Q} \Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]}, \forall j_Q \in J_Q \quad (24)$$

接著再依據前述的二維高斯波包( $\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]$ )，來加總計算時間 t 時目標駕駛受到調整後的心理動量  $\Phi_{j_Q}(t)$ ，並且依此來估算在時間  $t + \Delta t$  時，目標駕駛進行相對應的速度調整  $\dot{v}_i(t + \Delta t_i)$ 。其數學式如下：

$$\begin{aligned} \dot{v}_i(t + \Delta t_i) &= \alpha_1 \times \Phi_{j_Q}(t) \\ &= \alpha_1 \times W_i(t) \times \sum_{\forall j_Q \in J_Q} \delta_{i, j_Q}(t) \times [m_{j_Q} \times v_{j_Q \rightarrow i}(t)] \end{aligned} \quad (25)$$

式中， $\alpha_1$ ：表示一參數

$\Delta t_i$ ：表示目標駕駛的反應時間

此量子視覺流為基礎之駕駛刺激反應模式，有三點是值得注意的，第一，此模式原本式適用於事故發生影響區內，因當駕駛者在接近事故時，會因為周遭車況的改變，而不只是專注於前方車輛，亦會考慮其他視野範圍內所感受到的車輛，因此當鄰近車道有事故發生時，較能夠呈現真實的駕駛反應行為。

第二，從心理的觀點來看，每個人在受到同樣心理動量的情況下，會因為駕駛者的警覺心 $W_i(t)$ 不同而影響加減速的大小。當駕駛人越保守，表示警覺心值 $W_i(t)$ 越大，一有車流情況發生變化時，則進行的加減速動作越激烈，反之，若駕駛人越冒險，表示警覺心值 $W_i(t)$ 越小，所進行加減速反應則較小。

第三，考慮當前方車輛分別為小客車以及大型車時這兩種情況。通常可以發現兩者會有很大的不同，當前方是大型車時駕駛會比當前方是小型車時更專注於前方車輛且有著更高的警覺心，主要是因為當前方是大型車時會對駕駛帶來較大的心理壓力以及安全的疑慮，也因此會造成此兩種情況會有著不同的加減速行為。

## 第三章 跟車模式邏輯建構

本章主要會使用 J.B.Sheu(2011)所提出之跟車模式，首先對於整個系統範圍和限制進行界定，其次為了更清楚明瞭的表示跟車模式的邏輯程序，將其分成駕駛視野範圍區域之計算、鄰近車道視野邊界之計算、視野內車輛注意力權重之計算、駕駛跟車行為加減速的計算四個步驟。

### 3.1 系統基本假設

現實世界中的跟車行為，由於會受到周遭環境如人、車、路等因素的影響，為了使本研究能夠合理清晰明瞭。本研究在建立跟車模式建構的同時，會對於系統進行一些基本假設限制。假設限制如下：

1. 假設駕駛皆為並不考慮變換車道和事故的發生。
2. 研究共考慮兩種類型的車輛，分別為小型車和大型車，並且同一種類型的車輛性能和尺寸皆相同。
3. 假設駕駛只會受到同一車道和相鄰車道的影響，並且考量到前車會受到後方車輛阻擋，每一車道只考慮視線範圍內離目標車最近的車輛所帶來的影響。
4. 不考慮天氣因素，即表示路面乾溼狀況、輪胎胎面狀況以及視野狀況皆為狀況良好。

### 3.2 跟車模式邏輯

本文在文獻回顧中有提及本模式之主軸，將人類心裡面所感受到的車流狀況變化，以心理動量來進行表示，而所謂的心理動量是經過堆疊後的波所轉換而產生的，心理動量是駕駛人對於外在車流環境的綜合感受，而由於 Baker 所提的波長 ( $h=6.626 \times 10^{-34} \text{ J}\cdot\text{s}$ )經過動量的轉換之後值非常小，較難去進行計算的工作。因此本文所引用之文獻[1]的模式是引用駕駛人對於外在車流環境的綜合感受的心理動量，

來代替 Baker 所提及的波長。

### 3.2.1. 駕駛視覺區域之計算

為了要確認駕駛視線範圍內的車輛，必須先決定駕駛的視線範圍，而根據量子力學視覺流理論，可定義一個人的視線焦點範圍  $Q(\Delta x(t), \Delta y(t))$  其駕駛的視線焦點座落在點  $(x(t), y(t))$  的機率可由一橢圓形之二維高斯波包  $\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]$  來表示，而  $\Delta x(t)$  為視野縱軸 x 軸之標準差， $\Delta y(t)$  為視野橫軸的標準差。在本論文認為當駕駛在行駛車輛時，會將視線焦點中心放在同一行徑車道的前方車尾座標上避免和前車發生碰撞，因此視線焦點中心  $(\mu_{x_i}, \mu_{y_i}) = (x_{i-1} - l_{i-1}, y_{i-1})$ ，其中  $l_{i-1}$  為前方車輛之車長。並且車輛行駛時還需要注意和前方車輛之間的路面狀況以防有其他突發狀況產生(如物品掉落或是鄰近車道車輛變換車道)，因此將視線縱軸 x 軸的標準差  $\Delta x(t)$  之值設為本車和前車之間距。視線焦點橫軸的標準差  $\Delta y(t)$ ，則是以[8]所提及的車輛在行駛時會因為速度的上升而使得視線的角度下降，以車輛時速 40 公里時視線角度為 100 度、70 公里時為 65 度和 100 公里時為 40 度為基礎，車時速 70 公里為分界做兩個關係式來決定駕駛視角，進而求得視線橫軸之標準差  $\Delta y(t)$ 。駕駛視線範圍之示意圖如圖 3-1。

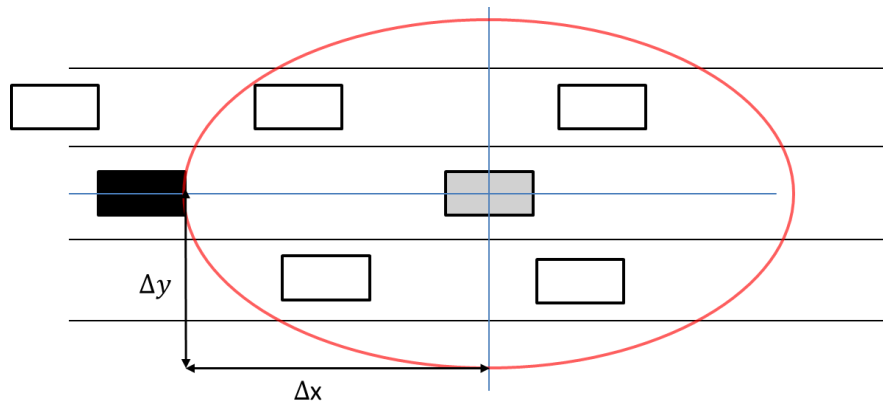


圖 3-1 目標車駕駛視線範圍示意圖

目標車和前車之間距為縱軸之標準差。

$$\Delta x(t) = x_i(t) - x_{i-1}(t) - l_{i-1} \quad (25)$$

$x_i(t)$ ：代表車輛 i 在時間 t 時的車頭位置

$x_{i-1}(t)$ ：代表車輛 i 前車在時間 t 時的車頭位置

$l_{i-1}$ ：代表車輛 i 前車之車長

以[8]所提及車速與駕駛視角關係如表，推得以時速 70 公里為分界之速度和視野橫軸標準差之關係式。

表 3-1 車輛速度和駕駛視線角度關係表

車輛速度(km/h)	駕駛水平視線角度
40	100
70	65
100	40

$$\Delta y(t) = \begin{cases} \Delta x(t) \times \tan(-7/12v_i(t) + 220/3), & \text{if } v_i(t) \leq 70\text{km/h} \\ \Delta x(t) \times \tan(-5/12v_i(t) + 185/3), & \text{if } v_i(t) > 70\text{km/h} \end{cases} \quad (26)$$

其中，

$v_i(t)$ ：為車輛 i 在時間 t 之速度

### 3.2.2 視線邊界和車輛權重之計算

而為了確認鄰進車道之車輛是否在視線範圍內，必需先了解駕駛視線在鄰近車道之邊界，而在前一小節提到駕駛的視線焦點範圍  $Q(\Delta x(t), \Delta y(t))$  為一橢圓形之二維高斯波包，因此可將駕駛視線範圍邊界座標以橢圓方程式來表示：

$$\frac{(y - \mu_y)^2}{\Delta y^2} + \frac{(x - \mu_x)^2}{\Delta x^2} = 1 \quad (27)$$

由於已知在不考慮變換車道的情況下每一車道車輛都是沿著縱軸 x 軸方向前進，

即每一車道車輛之  $y$  座標皆為已知，因此可由駕駛視線邊界的橢圓座標公式推得在  $y$  座標已知時駕駛視線邊界  $x$  座標的位置。駕駛對於鄰近車道的視線邊界可如圖所示。

$$x = \mu_x \pm \sqrt{\Delta x^2 \times \left(1 - \frac{(y - \mu_y)^2}{\Delta y^2}\right)} \quad (28)$$

在決定駕駛的視線範圍後，在三車道高速公路的環境下除了前方車輛，實際上鄰近車道可能會有不只一輛車在視線範圍內，但考量到除了駕駛視線範圍內最靠近目標車輛的車輛以外還會受到其他車輛阻擋而難以去考量其在視線內的範圍，並且考量到車輛相對距離越小時所帶給駕駛的心理動量會越大，對於駕駛的加減速行為影響較大，因此對於鄰近車道的車輛只考慮目標車駕駛視線內離目標車最近的車輛。如圖 3-2 所示。其中黑色的車輛為目標車輛，而灰色的車輛則是在目標車駕駛視線範圍內，會影響目標車輛駕駛跟車行為之車輛。

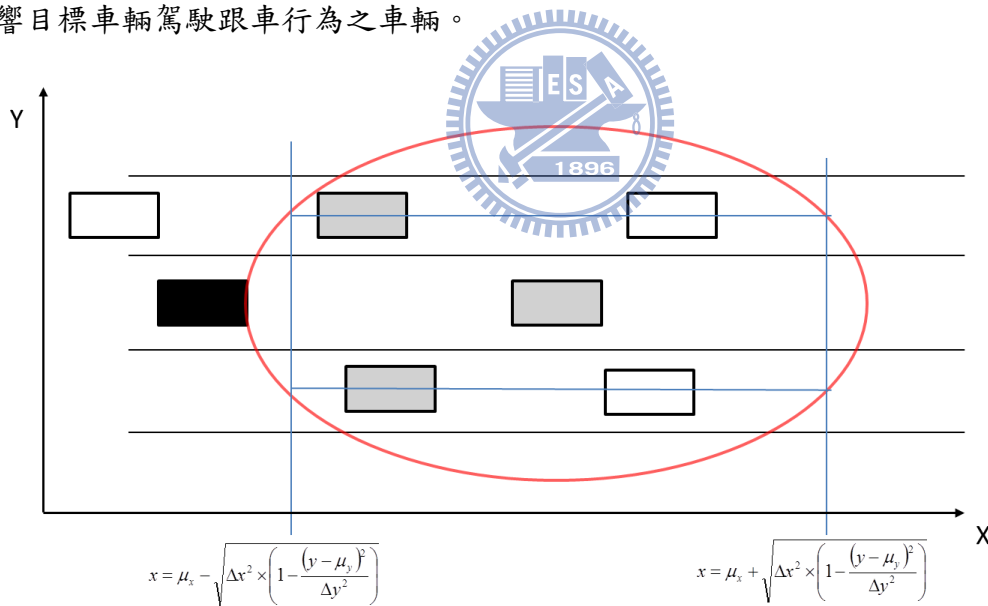


圖 3-2 鄰近車道視線邊界示意圖

當鄰近車道的車輛在視線範圍內並且後一輛車在視線範圍外時，即將此輛車視為在視線範圍內，並且會對目標車輛駕駛造成影響的車輛。



首先當 j 車道的車輛 i 在目標車輛的視線內

$$\mu_x - \sqrt{\Delta x^2 \times \left(1 - \frac{(y - \mu_y)^2}{\Delta y^2}\right)} \leq x_{i,j} \leq \mu_x + \sqrt{\Delta x^2 \times \left(1 - \frac{(y - \mu_y)^2}{\Delta y^2}\right)} \quad (29)$$

且後方車輛 i+1 在視線範圍外時

$$x_{i+1,j} < \mu_x - \sqrt{\Delta x^2 \times \left(1 - \frac{(y - \mu_y)^2}{\Delta y^2}\right)} \quad (30)$$

$x_{i,j}$  : 鄰近車道 j 車輛 i 之 x 座標

$x_{i+1,j}$  : 鄰近車道 j 車輛 i 後車之 x 座標

當上述兩條件皆成立時，即確認 j 車道之車輛 i 位於駕駛的視線範圍之內，接著再利用二維高斯機率聯合分佈函數  $\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]$ ，此函數可用以表示駕駛

視線焦點座落於視野內的車輛座標之二維高斯機率。

$$\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)] = \frac{e^{-\frac{1}{2} \left[ \frac{[x(t) - \mu_{Qx}(t)]^2}{(\Delta x(t))^2} + \frac{[y(t) - \mu_{Qy}(t)]^2}{(\Delta y(t))^2} \right]}}{2\pi \times \Delta x(t) \times \Delta y(t)} \quad (31)$$

將視線內車輛的高斯機率值加總後，可求得駕駛對於視線內每輛車影響駕駛注意力的相對權重，

$$\delta_{i,j_Q}(t) = \frac{\Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]}{\sum_{\forall j_Q \in J_Q} \Omega_{Q[\Delta x(t), \Delta y(t)]}[x(t), y(t)]}, \forall j_Q \in J_Q \quad (32)$$

### 3.2.3 跟車模式加速度計算

使用 J.B.Sheu(2011)所提出之跟車模式，來計算車輛的加減速跟車行為，在文獻

回顧中提到本模式的主軸，其中若將心理質量  $m_{jQ}$  視作定值，模式內將只考慮到車輛間速度的差異，當出現相對速度相同相對間距不同的兩種狀況時，不會因為車輛間距較短而產生較劇烈的加減速反應。本論文考量到駕駛面對不同車種在相對速度相對距離相同時，還是會因為車輛映入駕駛視野面積的大小而對於駕駛產生不同的心理質量。並且考量到車輛之間的距離也會影響到駕駛感覺車輛在視線內所佔的面積範圍，當距離越遠時駕駛所感覺到車輛的面積就會越小，因此本論文將心理質量  $m_{jQ}$  以視線內車輛背面的面積成正比和車輛間距成反比來進行處理。所求得的公式如(33)所示：

$$\begin{aligned} \dot{v}_i(t+1) &= \alpha_1 \times W_i(t) \times \sum_{\forall jQ \in J_Q} \delta_{i,jQ}(t) \times [m_{jQ} \times v_{jQ \rightarrow i}(t)] \\ &= \alpha_1 \times W_i(t) \times \sum_{\forall jQ \in J_Q} \delta_{i,jQ}(t) \times \left[ \frac{tall_{jQ} \times width_{jQ}}{\sqrt{(x_{jQ}(t) - x_i(t))^2 + (y_{jQ}(t) - y_j(t))^2}} \right] \times [v_{jQ}(t) - v_i(t)] \quad (33) \end{aligned}$$

在求得目標車駕駛所期望進行的加減速後，還須考量到實際車輛的物理限制，現實中車輛的加速度和減速度必然有一上限，因此在此定義一最大加速度  $A_{max}$  和最大減速度  $D_{max}$ ，以確保目標車輛不致於產生過大的加減速，也能更貼近真實的車流情形。

而模式雖然考慮到車輛之間間距的影響，但當只有本車和前車存在時，即使車輛間之距離很近，但速率相同時，套用上此模式仍不會有應有的減速拉開安全距離的動作。因此本研究必須導入由[11]所提及安全動態間距的概念，當本車與前車之間距一但小於一動態安全間距時，本車必須以最大減速率煞車確保和前車之間的安全，動態安全間距計算方式如下：

$$X_d + Z_{i-1} - Z_i \geq X_s \quad (34)$$

$$\Rightarrow X_d \geq X_s - Z_{i-1} + Z_i = X_s + \frac{v_i(t)^2}{2d(I_i)} - \frac{v_{i-1}(t)^2}{2D(I_{i-1})} \quad (35)$$

式中， $X_d, X_s$ ：分別代表動態安全間距和靜態安全間距(m)

$Z_i, Z_{i-1}$ ：本車與本車道前車之煞車距離(m)

$v_i(t), v_{i-1}(t)$ ：本車與本車道前車於  $t$  時之行駛速率(m/s)

$d(I)$ ：本車一般減速率( $m/s^2$ )

$D(I)$ ：前車最大減速率( $m/s^2$ )

而間距上的迷思除了過近的狀況，還有過遠的狀況，但在本研究已假設到心理質量和車輛間距成反比的關係，即當間距夠遠時，所受到前車的影響會變得非常小。



## 第四章 程式模擬系統之發展

本章使用 Turbo C 3.0 來撰寫建構第三章所提到的跟車模式邏輯，並且設定程式中所需之道路幾何和車輛特性參數，以利之後模擬車流運行。

### 4.1 模擬模式架構

車輛在高速公路路段推進過程中，因在本研究中不考慮車輛變換車道以及事故的發生，駕駛所可能採用的行為主要可分為自由行駛、一般跟車行為以及緊急煞車三種，以下將再對這些行為做較為詳細的說明。車流行為模擬過程即是根據判斷來決定本車應採用上述何種行為。圖 4-1 為車輛行為判斷流程。

#### 1. 自由行駛

本車為每車道的首車，即模擬模式中每車道所發的第一輛車，因同一車道前方不會有任何的車輛，因此本車將會以期望速率行駛。



#### 2. 一般跟車行為

本車不為所在車道首車，並且和前車之間距尚未小於動態安全間距。以 3.2 節所提之跟車模式內容，調整車輛加減速率。其中依序包括決定本車視野區域範圍、判定視野範圍內之車輛、計算駕駛對於視野範圍內車輛注意利權重最後再進行本車加減速率之計算。

#### 3. 緊急煞車

當本車和前車之間距小於動態安全間距，為了保持動態安全間距，車輛將以最大減速度進行煞車以增加和前車之間距。詳細內容和公式請參照 3.2.3 節之說明。

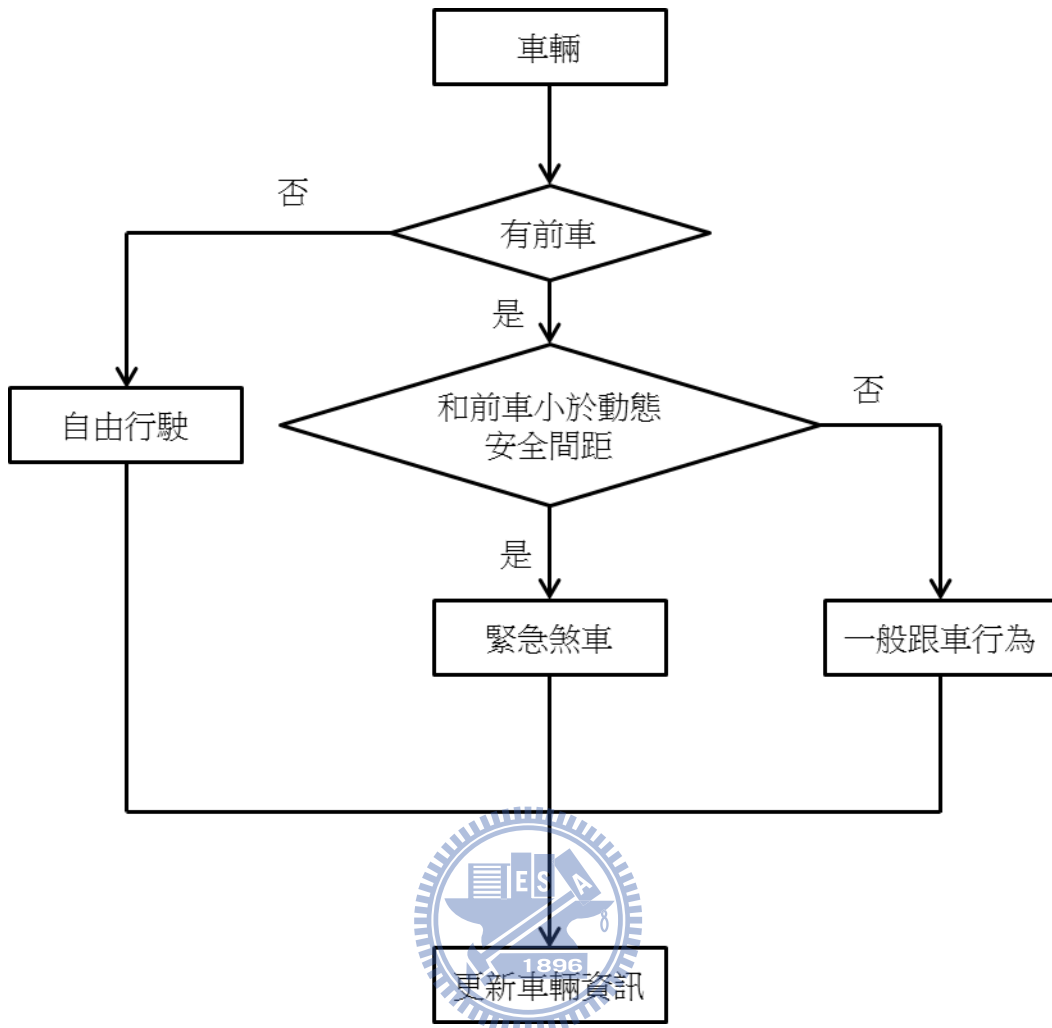


圖 4-1 車輛行為判斷流程圖

## 4.2 程式撰寫

程式中建立一直線三車道之高速公路路段，且三車道之車輛皆在不考慮車道變換的情況下，以第三章所建構之跟車模式進行車輛行進模擬。

程式中主要以 `main()` 控制模式整體的運作，在模擬時間內進行多個副程式，以 `drawRoad()` 進行三車道高速公路的繪製、`dawlane()` 繪製在道路上移動的車輛以及三個 `laneGenerator()` 產生不同車道之車輛和以 `laneTransistion()` 來進行各車道車輛的運行最後再 `show()` 輸出所需要之車流模擬資料。主程式的模擬流程圖如圖 4-1 所示，而對於各個副程式後面會有較詳細之說明。

系統模擬時間採用時階(time step)的方式，由於時間為連續的，若以每秒為單位容易造成誤差，因此以此方式將系統時間改成時階的方式，至於每一時階的長度已經變數化，本模擬程式訂為 0.1 秒，若需要變更只須在程式最前面的 TIMESLICE 的值進行變動即可。

每輛車輛以資料結構 struct \*node 來代表，其中包含車輛 x 軸位置 y 軸的位置、速度、加速度、車輛編號、駕駛視野縱深 dx、駕駛視野寬 dy、前方車輛和後方車輛，以鏈結串列(Linked List)的方式進行，將三個車道車道內的車輛進行串聯，優點在於可直接以指標變數指向或提取前方或後方車輛的資料，方便進行前後車輛相對距離、相對速度等跟車模式常用之參數計算。簡要程式函式說明如表所示：

表 4-1 程式函數說明

函數名稱	函數說明
Main()	主函式
C_laneGenerator()	中央車道車輛產生函式
L_laneGenerator()	左車道車輛產生函式
R_laneGeneration()	右車道車輛產生函式
C_laneTransition()	中央車道車輛運行函式
L_laneTransition()	左車道車輛運行函式
R_laneTransition()	右車道車輛運行函式
check_car	計算中央車道車輛加速度函式
check_carl	計算左車道車輛加速度函式
check_carr	計算右車道車輛加速度函式
v_acc()	車速函式
x_acc()	位置函式
watchtime()	觀看系統時間函式
hittime()	呼叫與對照系統時間函式
checkvision()	計算駕駛視野中心和標準差函式
visionx()	計算視野邊界 x 座標函式
normal()	常態分配函式
GauProb()	二維常態機率函式
car_safe()	動態安全間距函式

drawroad()	繪製道路函式
drawlane()	繪製車輛函式
output()	輸出結果資料函式

以下將說明主要函式知詳細內容：

## Input

主要負責資料輸入，其設定主要是放入主程式 main() 中，參數修改部分包括每車道車流量、大型車和小型車的車輛比例。輸入之每車道車流量為了配合一般使用者使用習慣，輸入每小時每車道之車流量再由程式內部轉換成每分鐘之車流量。

本研究設定兩個重要的參數；每個車道的車流量，用以觀察了解車流量不同時，對於車流整體運行時的影響，並且做為和其他變數改變時比較之基礎，模擬之車流量分為：高流量 1600 輛/小時、中流量 1200 輛/小時、800 輛/小時；大小型車種比例，為了觀察當增加大型車輛時對於車流狀況所造成的影響，模擬分為：不考慮大型車、小型車大型車比 80：20、小型車大型車比 50：50。

## C\_laneGenerator()、L\_laneGenerator()、R\_laneGenerator()

為中央左右車道的車輛產生，車輛產生的方式是以鏈串列(link list)來進行，每個鍊串會有數個車輛(node)，每台車輛之間以指標進行連接，往後若車輛需要進行車道的改變，只需要將指標指到的地方稍做修改即可達到變換車道的目的。

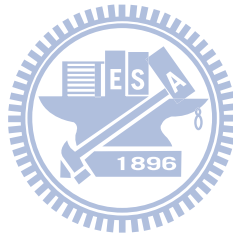
車輛產生的方式是將車間時距(headway)令其為 normal() 函式所回傳之值，將車道車流產生率以每分鐘平均產生多少車輛，轉換成產生車輛所需之秒數平均值，經由常態分配來做為車間時距。車輛到達時間分配的形式決定首先假設是以下瓦松分配的形式來產生，而當平均每分鐘超過 5 台時，可以使用常態分配來近似，因此車輛的產生是以平均每分鐘釋放  $\lambda$  台車，將  $60/\lambda$  作為 Normal 分配的平均值，變異數為 1，來進行車輛的產生。

### C\_laneTransition()、L\_laneTransition()、R\_laneTransition()

此副程式主要是用於控制道路上的車輛，使用第三章提到的跟車模式來計算車輛的加減速以及進行跟車行為。其中由 checkvision()來決定目標車輛駕駛的視線範圍 dx 和 dy，並以 check\_car()得到視線內的車輛併計算各車輛的心理權重值，進而求得目標車輛的加速度。再代入 x\_acc()求得車輛位置和 v\_acc()求得車輛速度，laneTransition 之流程由圖 4-2 所示。

### check\_car()、check\_carl()、check\_carr()

此副程式用於判斷車輛駕駛行為並且計算車輛的加減速，先判定車輛是否前方是否有車輛，當確定此車為首車時，車輛會將速度調整為車輛期望速率。若不為首車則以 dmin()計算動態安全車距，並判斷前方車輛是否小於動態安全車距，若為真，加速度為最大減速度，若為否則判斷視線內車輛權重並計算車輛加速度。





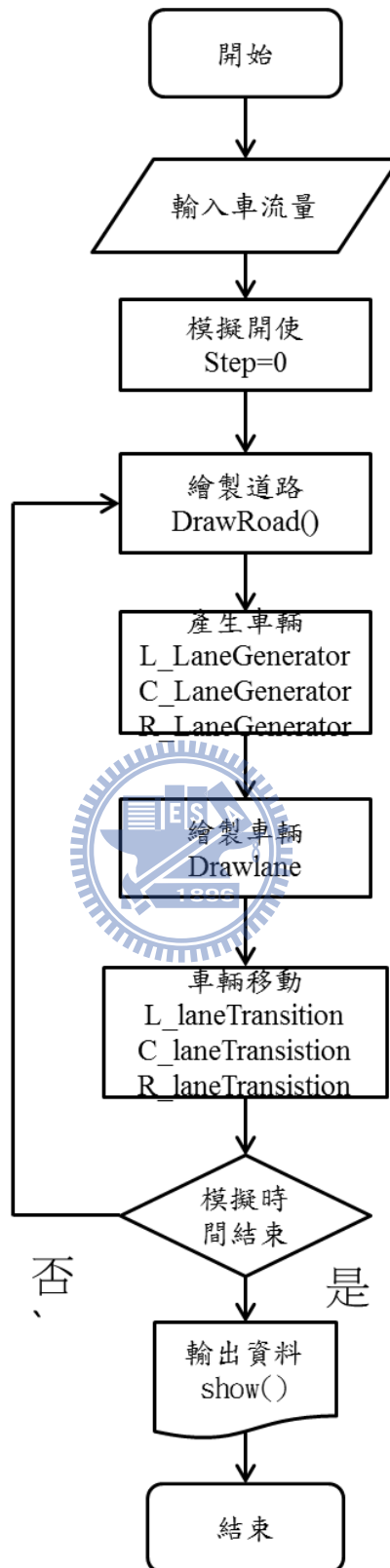


圖 4-2 主程式 main() 流程圖

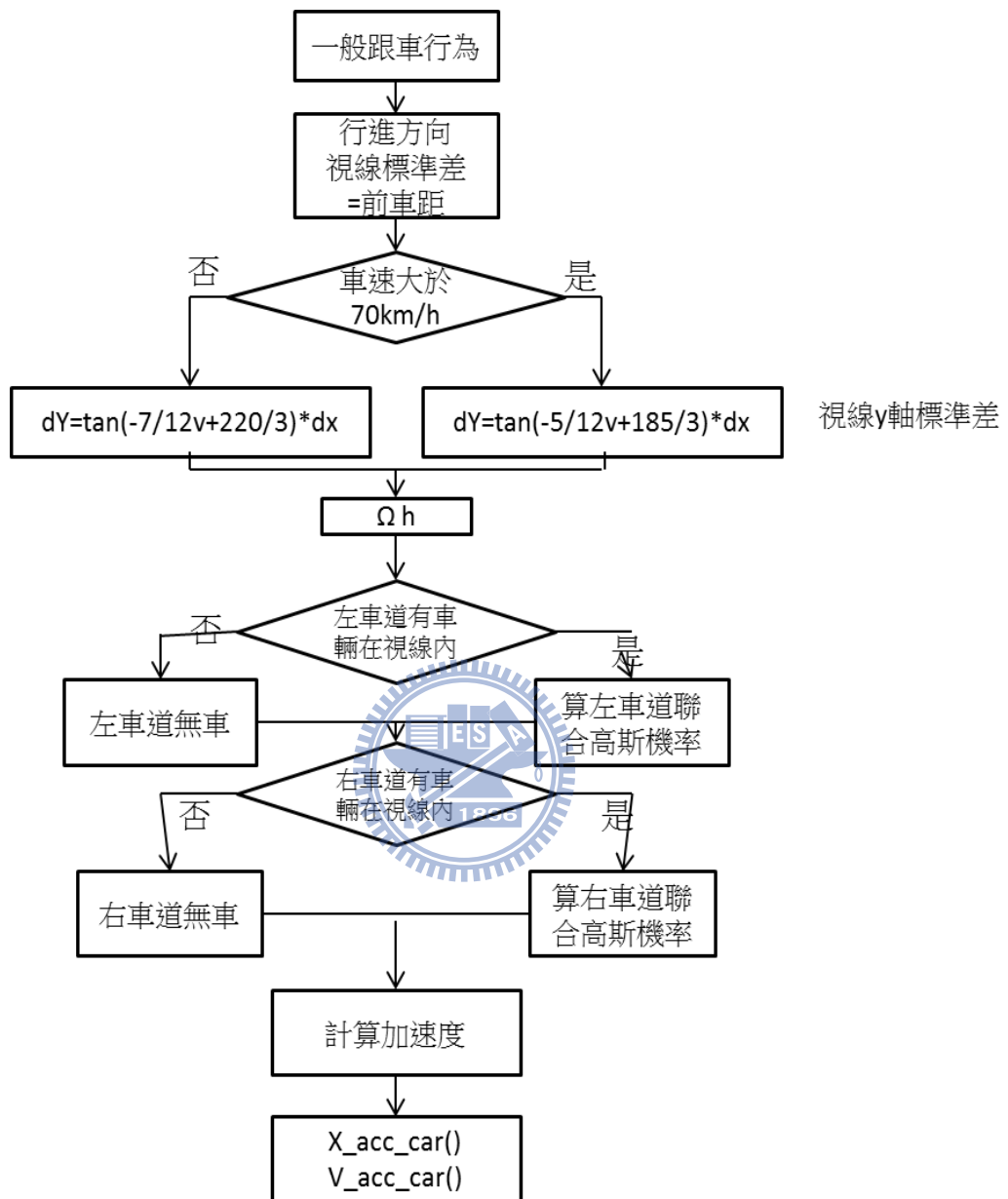


圖 4-3 副程式 chek\_car()之一般跟車行為流程圖

drawRoad()、drawlane()

此二函式為繪圖函式，利用 turbo C 內建之 <graphics.h> 繪圖函式庫，以 drawRoad() 來進行直線三車道之繪製，由於 dos 只能支援 640x480 像素，若想要觀看到正確比例的模擬效果，則需要將每個像素代表一公尺，模擬之路段長度改為 640 公尺，但若想要表達更長的車道長度時，還是可透過程式碼將比例進行縮放，來改變模擬車道的長度。

車輛的部分則是以 darwLane() 來進行繪製，配合 laneTransistion 得到車輛的位置後，以填色的長方形方塊來表示，而大型車和小型車的表示，因為畫面像素的關係，大型車和小型車在螢幕畫面顯示只會有車輛長度的不同，車輛的寬度看起來是相同的。

show()

建立一個 excel 檔，並且以 fprintf() 來決定想要輸出之資料，使用 excel 主要是出於輸出結果整理方便，且可由 Excel 所內建之繪圖功能，繪製所需要的結果關係圖。

### 4.3 模擬環境之設計

本小節主要為程式所需輸入之參數設定，可分為道路幾何特性和車輛特性兩個部分。

#### 1. 道路幾何特性

- (1) 直線三車道高速公路基本路段
- (2) 無坡度、無曲度之直線型車道
- (3) 路段上無收費站、無匝道和無交織路段。
- (4) 每車道寬 3.75 公尺
- (5) 車道長度 1000 公尺。

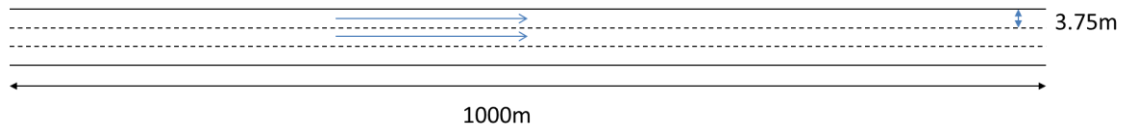


圖 4-4 模擬高速公路幾何特性示意圖

## 2. 車輛特性之設計

本研究將車輛分為小型車和大型車做為研究對象，參考[9]所蒐集整理之資料，而所考慮之車輛特性又分為實體車輛尺寸、車輛加減速率、車輛最大加減速率車輛。

### (1) 實體車輛尺寸

表 4-2 車輛尺寸資料表

單位：公尺

車型	實體車長	實體車寬	實體車高
小型車	4.0	2.0	1.3
大型車	10.0	2.5	4.1

### (2) 車輛最大加減速

車輛的最大加減速是當車輛之車速在乾燥路面下、輪胎狀態良好下之加減速度。而本模擬程式在此是假設大型車和小型車的加減速性能相同，並且假設其最大加速度為  $3.6(m/s^2)$ 、最大減速度為  $7.2(m/s^2)$ 。

## 第五章 參數校估及模擬結果分析

### 5.1 參數校估

#### 資料蒐集

本次實際攝影調查路段為台北市仁愛路四段，光復南路至敦化南路路段的內側車道如圖 5-1 所示，選擇此路段的特色為，路段內側車道只有公車專用道和一般禁行機車道兩車道，並且路段內側車道和外側車道之間有分隔島分隔。選擇此路段的主要原因如下：

1. 除了資料收集的車道本身外，僅鄰接一流量相對較小的公車專用道，減少鄰近車道數，降低周遭環境所帶來的干擾。
2. 內側車道禁止機踏車行駛，因此可防止機踏車對於駕駛的影響。
3. 對向車流已經以分隔島隔開，以降低相對速度較大的對向車流對於駕駛所產生的影響。

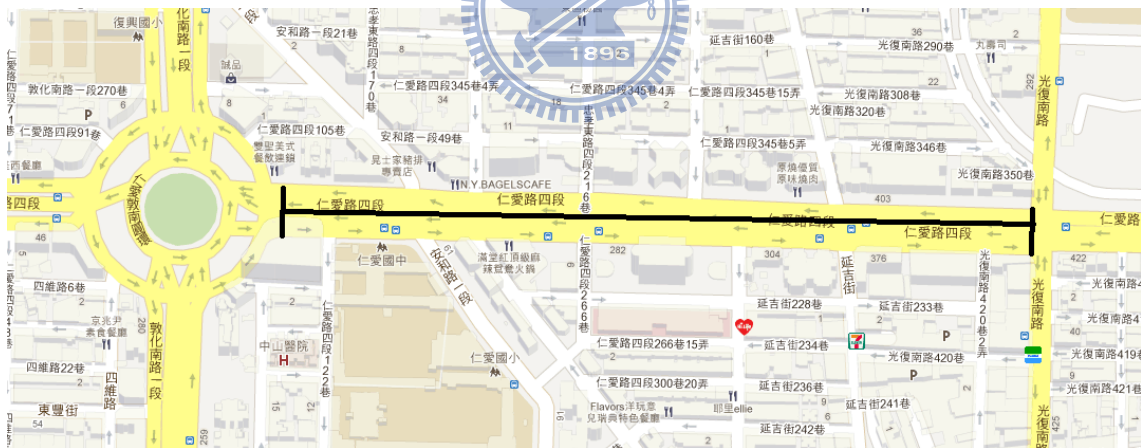


圖 5-1 台北市仁愛路資料收集路段

使用兩台車分別為前方車輛和進行跟車之目標車輛來進行資料收集，每台車輛分別固定一台攝影機同時拍攝車輛行駛時駕駛視野道路狀況及儀表板紀錄車輛時速。並且將兩台攝影機的時間進行校對，以確保車輛在行駛時兩台攝影機所收集到的資料是同步的，讓前車於同一車道自由行駛並且讓目標車跟隨前車。

並且再以攝影所收集到的資料車輛位置車輛速度，我們以攝影機拍攝兩台車在同一時間點儀表板所顯示的車輛速度來求得兩輛車之間的相對速度，以及攝影所記錄到的目標車和前方車輛的車輛位置。較詳細的處理步驟如下：

1. 當兩輛車行駛進實驗路段時，同時開始進行攝影，紀錄影片同一秒的儀表速度即為兩台車同一時間點分別的速度，可求出影片當下兩輛車之相對速度。
2. 在記錄影片某時間點的速率以後，記錄下一秒的車輛速率，以此來做為車輛前一秒的瞬時加速度。
3. 因兩台車輛攝影機固定的位置和角度皆為相同，因此以記錄影片的邊緣和道路的位置進行比較，得到兩台車當下時間點的位置，藉此求得兩輛車的相對距離。

對於車輛駕駛的警覺心 $W$ ，雖然駕駛時可能會因為長時間駕駛疲勞或是當下精神狀況等而產生改變不是固定的，而此次資料收集是短時間駕駛並且告知是做為研究數據校估用之資料，假設駕駛在此種狀況行駛時，會保持高度的警覺心，並且因為是短時間的駕駛，所以不考慮到駕駛可能會因為駕駛疲勞等因素導致駕駛的警覺心改變，本研究將其假設為一固定值 0.8。

#### 模式參數校估

模式的校估以每 5 秒的車輛行駛狀況進行資料的收集，總共有樣本 30 筆，本研究以用 SPSS 統計軟體將設定為  $\alpha=0.05$  進行迴歸分析，分析結果如下表 5-1 所示：

表 5-1 模式參數校估結果

解釋能力 R-Square	係數	未標準化系 數估計值	標準化係數	t 值	顯著性(p 值)
0.290	常數項	-0.042	-	-1.4	0.178
		9.144	0.538	2.709	0.014

由以上所示，R-Square 來看解釋能力不是很好，而常數項的 t 檢定 P 值大於 0.05，顯示無法拒絕常數項不等於 0，因此在此將常數項視為零。係數的 t 檢定 P 值則小於 0.05 明顯拒絕係數 $\alpha_1=0$  的假設，而標準化系數為 0.538 表示感受到的心理動量值與加減速度的相關程度為正的，亦即感受到的心理動量值越大於 0 則加速度越大，但因為要考量實際的加減速值，所以本研究採用未標準化系數分別得出參數常數項為 0、 $\alpha_1=9.144$ 。最後可將跟車模式寫作下式：

$$\begin{aligned} \dot{v}_i(t+1) &= \alpha_1 \times W_i(t) \times \sum_{j_{j_0} \in J_0} \delta_{i,j_0}(t) \times \left[ \frac{tall_{j_0} \times width_{j_0}}{\sqrt{(x_{j_0}(t) - x_i(t))^2 + (y_{j_0}(t) - y_j(t))^2}} \right] \times [v_{j_0}(t) - v_i(t)] \\ &= 9.144 \times 0.80 \times \sum_{j_{j_0} \in J_0} \delta_{i,j_0}(t) \times \left[ \frac{Width_{j_0} \times Tall_{j_0}}{\sqrt{(x_{j_0}(t) - x_i(t))^2 + (y_{j_0}(t) - y_j(t))^2}} \right] \times [v_{j_0}(t) - v_i(t)] \quad (36) \end{aligned}$$



## 5.2 模式驗證

為了瞭解參數校估後的模式，以及模擬程式是否符合真實的駕駛情形，本小節將以模擬程式之模擬結果和實際跟車攝影資料進行比較，實際資料為同一路段重複行駛三次每五秒的資料，一趟十筆共三十筆的車輛行駛資料，以第一個時間點的車速平均值作為模擬的起始速度平均值，以兩車進入起點時的時間差來作為發車時間的平均值。

經過 50 次模擬之後，將後方車輛每五秒的速度和實際跟車之速度進行平均絕對離差，即將同一個時間點的模擬結果後車的車速，和實際跟車資料的車速相減取絕對值後進行平均，以此來觀察每五秒模擬結果車速和實際資料車速的差異，可以得到圖 5-2。

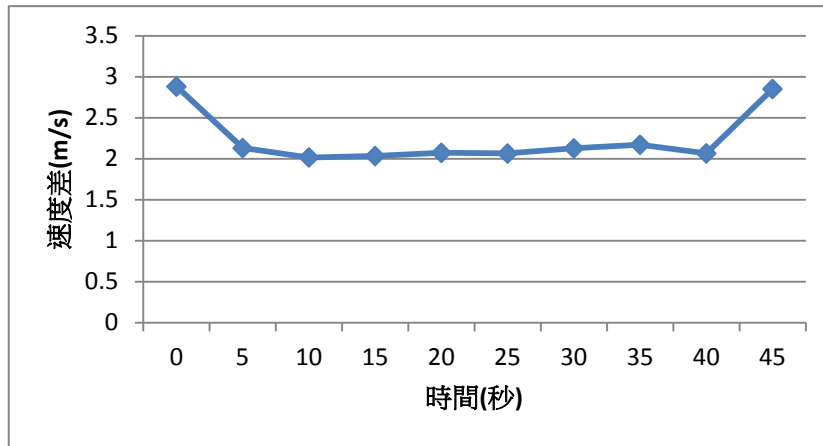


圖 5-2 模擬和實際速度平均絕對離差圖

由圖中觀察可發現除了頭尾以外，平均絕對離差都穩定的在 2 附近變動，而起始和結束時的值特別大，可能是因為在進入觀測路段時實際觀察資料的車輛受到前一個路口紅綠燈的阻擋，讓車輛在進入觀測路段時速率較低，而在最後離開觀測路段時因接近下一個路口，可能受到前方路口為紅燈或是因要進入十字路口而使得車流行駛狀況改變，因此頭尾時間的平均絕對離差會較大。因此我們認為此模擬程式所得的結果，在一個穩定的誤差範圍內。

### 5.3 模擬結果情境分析

將模式寫成程式後，為了探討模式的實用性以及了解模式中輸入參數的改變對於系統之影響，建立了幾種情境對於模擬結果數據進行比較和分析其合理性，並且為了比較模式所帶來影響，因此每個情境得亂數種子皆相同。

首先是針對前後兩輛車輛後車進行跟車時之模擬情形，觀察當驚覺心改變時對車輛駕駛行為的影響，接著再設立幾種情境，觀察模擬程式中央車道整體車流車速和車間距所受到的影響。



## 改變警覺心

在本文參數校估時，是將駕駛的警覺心假設為一固定值 0.8，而我們認為在同樣的車流環境下，當警覺心越大時對於加減速的反應就會越劇烈，因此好奇在模擬程式中是否能夠表現出此一特性，因此假設當警覺心分別為 0.2、0.5、0.8 三種情形，進行跟車模擬，將三種警覺心的車速和加速度進行比較，所得的結果如下圖所示。

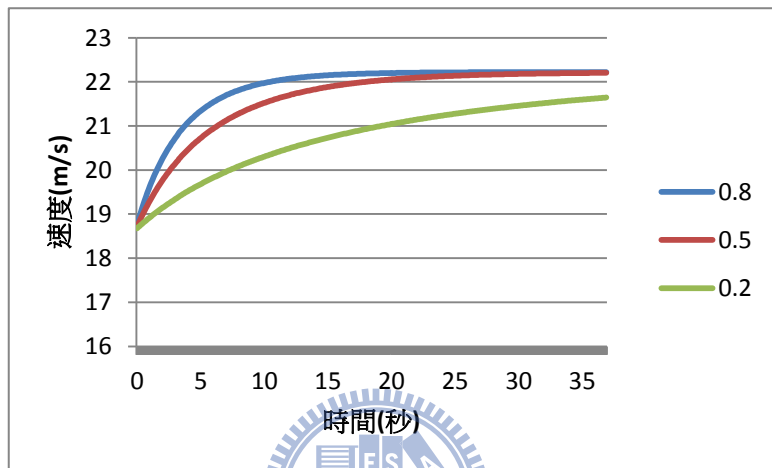


圖 5-3 三種警覺心模擬結果速度比較圖

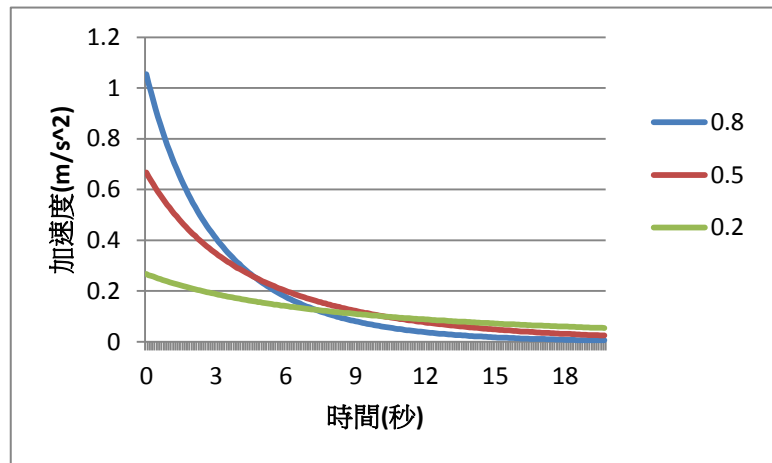


圖 5-4 三種警覺心模擬結果加速度比較圖

可以觀察到三種情況下的車輛的速度都是逐漸得向期望速度逼近，且的確是初期當警覺心越大時，速度的改變量較大，會較快達到和前車的速度相同，也和我們所預

期的結果一致。

### 車流整體狀況

我們改變三種參數來做為車流模擬情境：1.車流量：共分成高流量 1600 輛/小時、中流量 1200 輛/小時、低流量 800 輛/小時。2.車種比例：可分成只有小型車、小型車大型車比 80：20、小型車大型車比 50：50。3.左右車道有車和左右車道無車輛。

比較當車道車流量不同且不考慮大型車時模式所帶來的模擬結果，首先列出時間空間關係圖。



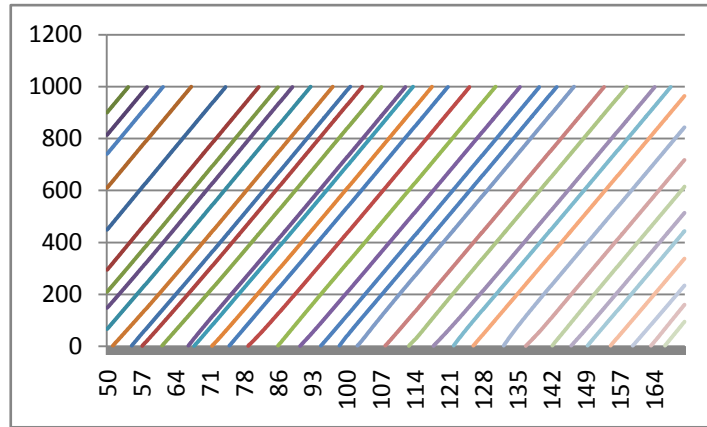


圖 5-5 一般情況低流量時空圖

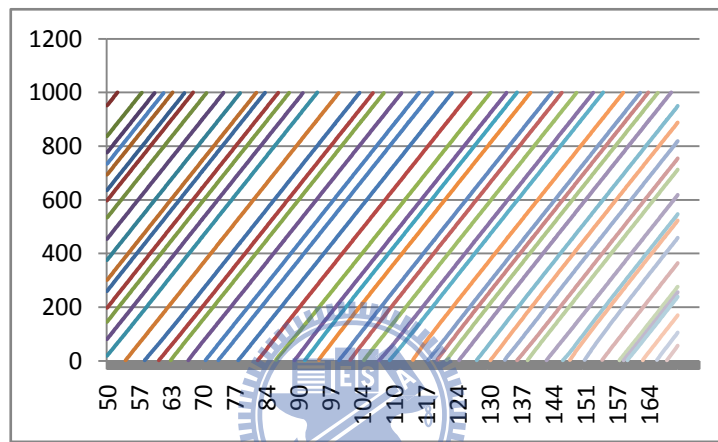


圖 5-6 一般情況中流量時空圖

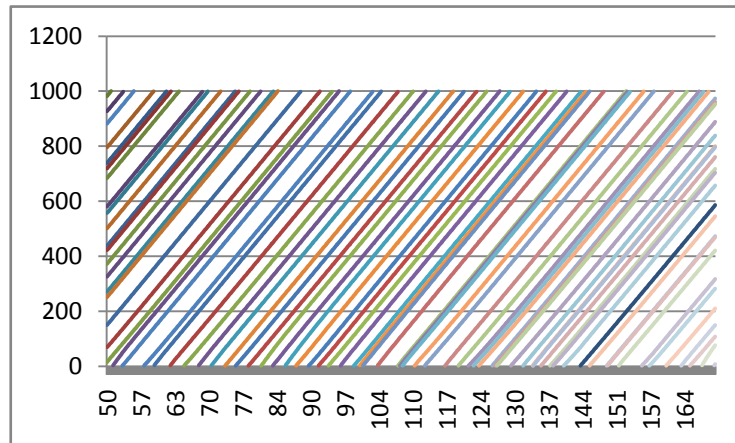


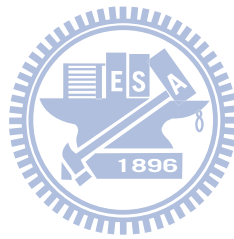
圖 5-7 一般情況高流量時空圖

因為本論文未考慮到車輛變換車道、事故發生、號誌控制等會對於車輛行駛狀況產生較明顯變化的情形發生，也因此車輛的速度會逐漸的朝著預設的期望速度逼近逐漸穩定。因此由時空關係圖圖 5-5、圖 5-6 和圖 5-7 只能夠觀察到因為車流量的不同

車輛的時間間隔也會不同，流量越高前後車輛通過同一個位置的時間間隔會越短，或是當車輛行駛距離尚短時因為車輛速度和期望速度仍有些差距，因此可以看到些許的斜率也就是行車速率的改變。

### 1. 當車流中包含大型車時

本論文之跟車模式有提到車輛的加減速會受到視線內車輛背面面積的影響，在此為了確認是否會受到影響，模擬試驗增加改變小型車和大型車之間的比例除了原本的100%為小型車，又有小型車：大型車 80：20 和 50：50 三種車輛比例。而各流量的小型車和小型車比 80：20 和 50：50 模擬結果的每五秒平均速度和平均車間距比較圖如下所示，其中左圖為平均速度，橫軸為時間(秒)、縱軸為速度(m/s)，右邊為平均車間距，橫軸為時間(秒)、縱軸為距離(m)。



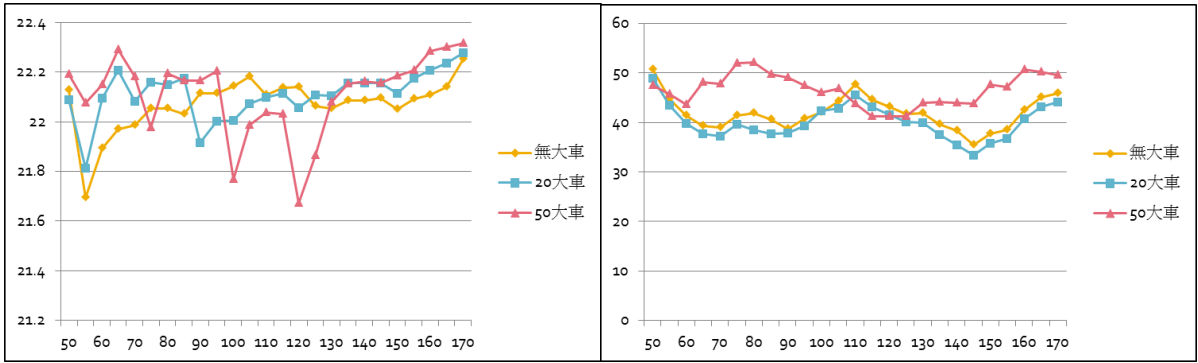


圖 5-8 高流量三種車種比例每五秒平均速度與平均車間距比較圖

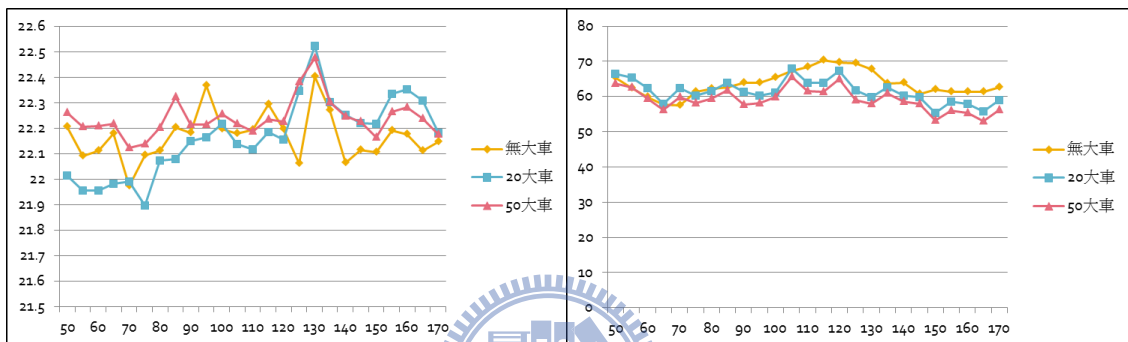


圖 5-9 中流量三種車種比例每五秒平均速度與平均車間距比較圖

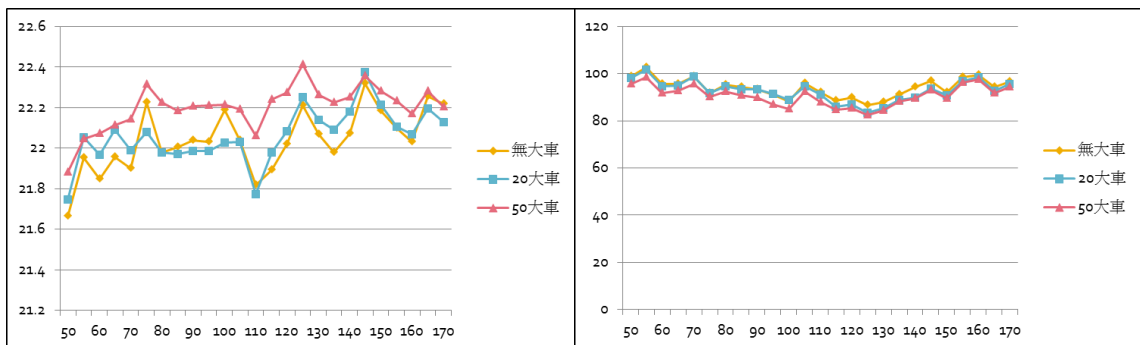


圖 5-10 低流量三種車種比例每五秒平均速度與平均車間距比較圖

由圖中可看出只有高流量情形下，大車比例為 50% 的車輛平均速度和平均車間距的走向特別的不同，但是因為我們想要更明確的瞭解模擬結果是否會因為大型車車輛比例的增加而造成平均速度和平均車間距的平均值改變，因此我們分別對高中低車流

量的三種車種比例進行 ANOVA 檢定，觀察當流量相同車輛比例不同時速度以及車間距的平均值是否會相同。

表 5-2 高流量三種車種比例平均速度 ANOVA 檢定結果

車種分配	變異數	平均數	P 值	有顯著差異
無大型車	0.10583	22.0707	0.440	否
小型車：大型車 80：20	0.09908	22.1085		
小型車：大型車 50：50	0.15997	22.1115		

表 5-3 中流量三種車種比例平均速度 ANOVA 檢定結果

車種分配	變異數	平均數	P 值	有顯著差異
無大型車	0.09433	22.1691	0.034	是
小型車：大型車 80：20	0.15044	22.1648		
小型車：大型車 50：50	0.07442	22.2394		

表 5-4 低流量三種車種比例平均速度 ANOVA 檢定結果

車種分配	變異數	平均數	P 值	有顯著差異
無大型車	0.15250	22.0395	0.000	是
小型車：大型車 80：20	0.13465	22.0592		
小型車：大型車 50：50	0.10865	22.2018		

表 5-5 高流量三種車種比例平均速度多重比較結果

車種 1	車種 2	P 值	有顯著差異
無大型車	小型車：大型車 80：20	0.401	否
無大型車	小型車：大型車 50：50	0.543	否
小型車：大型車 80：20	小型車：大型車 50：50	0.996	否

表 5-6 中流量三種車種比例平均速度多重比較結果

車種 1	車種 2	P 值	有顯著差異
無大型車	小型車：大型車 80：20	0.992	否
無大型車	小型車：大型車 50：50	0.015	是
小型車：大型車 80：20	小型車：大型車 50：50	0.081	否

表 5-7 低流量三種車種比例平均速度多重比較結果

車種 1	車種 2	P 值	有顯著差異
無大型車	小型車：大型車 80：20	0.879	否
無大型車	小型車：大型車 50：50	0.000	是
小型車：大型車 80：20	小型車：大型車 50：50	0.000	是

首先看平均速度的檢定結果，由檢定結果來看反而是中流量和低流量的情況下每

五秒平均速度平均值的 ANOVA 檢定結果有顯著的差異，和前面觀察折線圖認為高流量情況下的平均速度看起來差異最大，反而檢定的結果為沒有顯著的差異。

並且我們好奇中低流量是哪些車種比例之間會有顯著的差異，因此再將這兩種流量分別進行多重比較，在進行比較後發現，無大型車輛和大小型車比例為 50 比 50 時會有明顯差異，可以顯示在中低流量車種比例差異夠大時會對於車輛平均速度產生影響。

表 5-8 高流量三種車種比例平均車間距 ANOVA 檢定結果

車種分配	變異數	平均數	P 值	有顯著差異
無大型車	3.38288	41.8498	0.000	是
小型車：大型車 80：20	3.53919	40.0765		
小型車：大型車 50：50	3.25742	46.5536		

表 5-9 中流量三種車種比例平均車間距 ANOVA 檢定結果

車種分配	變異數	平均數	P 值	有顯著差異
無大型車	3.54032	63.6644	0.000	是
小型車：大型車 80：20	3.27969	61.4871		
小型車：大型車 50：50	3.23498	59.1586		

表 5-10 低流量三種車種比例平均車間距 ANOVA 檢定結果

車種分配	變異數	平均數	P 值	有顯著差異
無大型車	4.06040	93.9442	0.027	是
小型車：大型車 80：20	4.53231	92.5699		
小型車：大型車 50：50	4.35186	90.5975		



表 5-11 高流量三種車種比例平均車間距多重比較結果

車種 1	車種 2	P 值	有顯著差異
無大型車	小型車：大型車 80：20	0.177	否
無大型車	小型車：大型車 50：50	0.000	是
小型車：大型車 80：20	小型車：大型車 50：50	0.000	是

表 5-12 中流量三種車種比例平均車間距多重比較結果

車種 1	車種 2	P 值	有顯著差異
無大型車	小型車：大型車 80：20	0.072	否
無大型車	小型車：大型車 50：50	0.000	是
小型車：大型車 80：20	小型車：大型車 50：50	0.039	是

表 5-13 低流量三種車種比例平均車間距多重比較結果

車種 1	車種 2	P 值	有顯著差異
無大型車	小型車：大型車 80：20	0.501	否
無大型車	小型車：大型車 50：50	0.019	是
小型車：大型車 80：20	小型車：大型車 50：50	0.269	否

接著對於平均間距進行 ANOVA 檢定和多重比較，結果顯示，高中低流量三種車

種比例的平均車間距皆有顯著的不同，再看多重比較的部分，無大型車和小大型車比例為 50：50 兩種情況的比較，三種車流量下皆有顯著的差異。因此由已上的結果來看，可以看道當車輛車種的比例差異夠大的時候，的確會影響駕駛行為改變車輛的平均速度以及平均車間距，也表示此模擬程式的確能夠表現出當車流中的車輛車種改變時，對於駕駛行為會產生影響。

## 2.只針對前車時對於速度的影響。

本論文跟車模式和傳統跟車模式差異最大的地方在於不僅針對和前方車輛的互動關係改變車輛的加減速，因此好奇當目標車輛駕駛只考慮前方車輛的影響進行加減速時，對於模擬結果會產生什麼差異。在此模擬試驗只有中央車道會產生車輛，表示左右車道將不會有車輛對於駕駛產生影響。並以高流量每車道 1600 輛/小時、中流量每車道 1200 輛/小時、低流量每車道 800 輛/小時進行模擬。模擬結果每五秒平均速度以及平均車間距之折線圖如下所示。



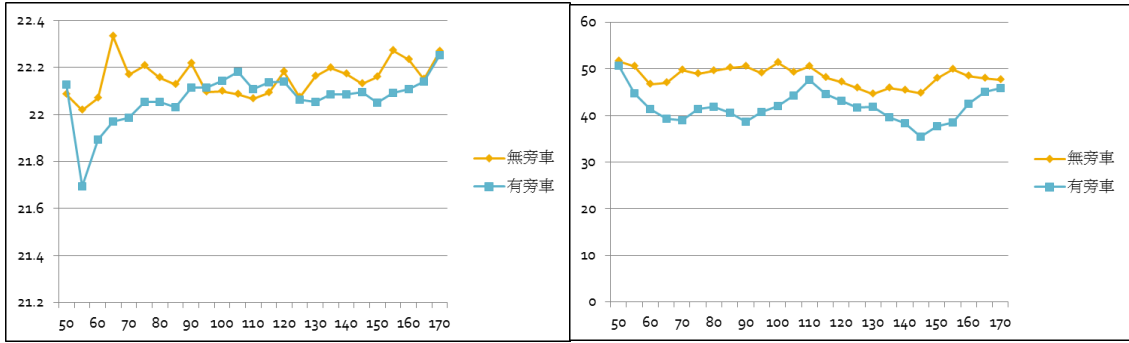


圖 5-11 高流量一般情況和無旁車之平均速度和車間距比較圖

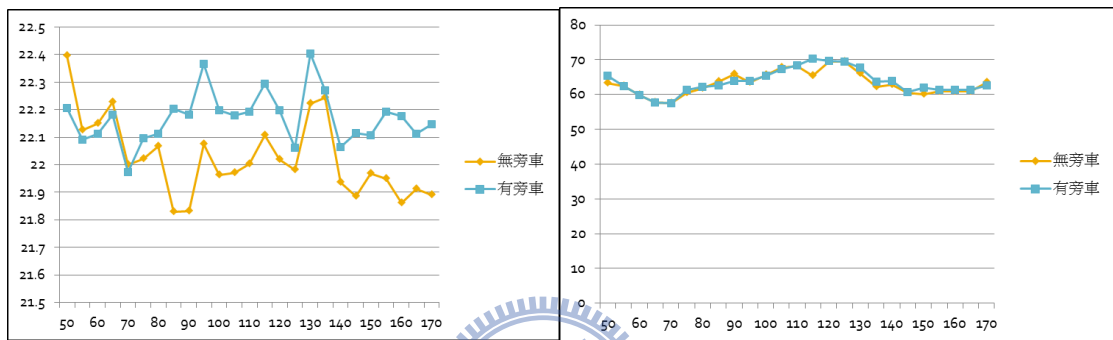


圖 5-12 中流量一般情況和無旁車之平均速度和車間距比較圖

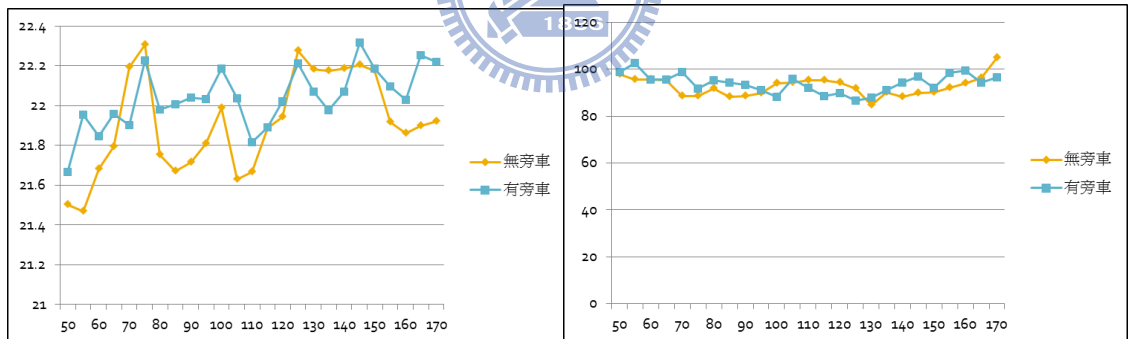


圖 5-13 低流量一般情況和無旁車之平均速度和車間距比較圖

表 5-14 高流量一般狀況和無旁車每五秒平均速度檢定和平均車間距檢定

	平均速度		平均車間距	
	變異數	平均數	變異數	平均數
一般情況下	0.10583	22.1522	3.38288	41.8498
只有前方車輛	0.07528	22.0707	2.04463	48.3283
P 值	0.605	0.003	0.088	0.000
有顯著差異	否	是	否	是

表 5-15 中流量一般狀況和無旁車每五秒平均速度檢定和平均車間距檢定

	平均速度		平均車間距	
	變異數	平均數	變異數	平均數
一般情況下	0.14138	22.1691	3.54023	63.6644
只有前方車輛	0.09433	22.0252	3.35498	63.1849
P 值	0.056	0.000	0.839	0.625
有顯著差異	否	是	否	否

表 5-16 低流量一般狀況和無旁車每五秒平均速度檢定和平均車間距檢定

	平均速度		平均車間距	
	變異數	平均數	變異數	平均數
一般情況下	0.15250	22.0395	4.06040	93.9442
只有前方車輛	0.24631	21.9119	4.17733	92.5574
P 值	0.012	0.033	0.954	0.240
有顯著差異	是	是	否	否

表 5-17 一般狀況和無旁車每五秒平均速度檢定是否顯著差異整理

	平均速度		平均車間距	
	變異數	平均數	變異數	平均數
高流量	否	是	否	是
中流量	否	是	否	否
低流量	是	是	否	否

結果顯示，高中低三種車流量每五秒速度平均值皆有顯著的不同，可以表示當在一般情形和無旁車時，這兩種狀況對於車輛速度影響會有顯著的差異，顯示此模式的確可以反應出當旁邊車道有車輛在行駛時的確會影響到整體車流的速度。而在平均車間距的部分，由車輛平均間距圖和檢定結果可發現，只有高流量情況下的平均車間距會有顯著的差異。由以上的檢定和數據觀察，顯示駕駛在行車時其平均車速，會因為受到周遭環境的影響而有所改變。



## 第六章 結論與建議

### 6.1 結論

在過去的跟車理論文獻中只將對象鎖定為前車，本文跟車模式是參考 Sheu(2011) 所提出的量子力學流跟車模式，考慮駕駛者視覺對心智的負荷，將對象增加為前車和旁車，並且考慮車量對於駕駛視線內所佔的面積，以實際攝影跟車資料進行參數校估，並撰寫模式對模擬結果進行分析。本研究模擬以微觀觀點，採用單位時間掃描方式，處理高速公路上以量子視覺流跟車模式進行車流行為之模擬。模擬範圍為直線高速公路路段，可模擬大型車與小型車混合車流。在程式和模式驗證的部分，和實際資料比較下，可發現模擬結果和實際的資料是在一個穩定的誤差範圍內。而由情境模擬結果發現，確實可以對於旁車對於駕駛所造成的分心影響進行解釋，且可發現對於整體車流速度有明顯的影響。對於改變車流大型車和小型車的組成比例，來觀察車流模擬結果是否會受到影響，結果顯示當車種比例差異夠大時，的確會對於駕駛的行為改變影響車輛的速度和車間距有顯著的差異。由於只是初步的模擬車輛跟車模式的前進，沒有考慮到其他容易改變車輛行為的因素，以利於觀察模式所帶來的影響，因此模擬數據如行車平均速度大都是在首車的行車速率附近變動。

### 6.2 建議

1. 本研究模式進行之參數校估的資料收集，只針對本車和前車，仍有鄰近車道之車流進行干擾，並且資料收集地點是位於市區，車速和模擬時所設定得期望速度有所差距，若能夠選擇在較少車輛的高速公路路段進行資料收集的話，也許會產生不同的結果。
2. 因考量到同一車道之間車輛會互相阻擋，本研究假設每個車道最多只會有一輛車在駕駛的視野內，但實際上車輛可能只會有部分遭到阻擋，仍有部分面積出現在駕駛的視線範圍內，期望將來研究者可探討當車輛受到部份遮擋時，對於駕駛者感受的影響。

響。

3. 模式中並未考量到車道變換的情形，當前方車輛變換車道或是本車變換車道時對於駕駛這會造成何種影響，期望將來研究者能對變換車道時產生的影響進行研究。

4. 模式中所假設之心理質量是以和車輛背面面積成正比和車輛相對距離成反比，可能實際上並不是以這種關係存在，並且如第 2 點所示，並不一定是車的背面映入駕駛者的視野，期望能夠對於駕駛的心理質量能夠有更精確的定義。

5. 本研究假設模式中駕駛的驚覺心為一固定值，然而實際上警覺心會因為駕駛的各種生理心理條件而改變，若能夠配合生理心理測量儀器，得到更多得精密的心理研究資料，代表影響警覺心之因素，使得模式更為準確。



## 參考文獻

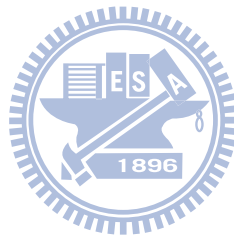
- [1]Jiuh-Biing Sheu, Characterisation of driver behavior during car following using quantum optical flow theory,Transportmetrica, DOI:10.1080/18128602.2011.572571
- [2]Robert G.V. Baker, On the quantum mechanics of optic flow and its application to driving in uncertain environment” Transportation research part F, 1999.
- [3]Hoefs, D.H., Untersuchung des Fahrverhaltens in FahrzeugKolonnen, Forschungsberichte des Institute fur Verkehrswesen, Heft 140,Universitate Karlsruhe,1972.
- [4]Wiedemann, R., Modelling of RTI-elements on multi-lane roads. In the Commission of the European Community, Advanced Telematics in Road Transport, Vol. DG XIII.1991.
- [5]Russel L.M. and M.L. Harold, “Simulation of Traffic Flow to Obtain Volume Warrants for Intersection Control”, HRR15, pp.1-43, 1963.
- [6]Brackstone, M. and McDonald, M., Car-following: a historical review. Transportation Research Part F, 2, 181–196.1999
- [7]邱明峰,高速公路事故發生時非事故車道上駕駛人感受與駕駛行為之研究，國立交通大學交通運輸研究所碩士論文，民國96年6月
- [8]陳俊鴻，以視覺模擬方式研究機車駕駛人於路段中之駕駛行為，台大土木工程學研究所碩士論文，民國92年6月。



[9]張鈞華，模擬一般市區道路事故發生之微觀車流行為，台大土木工程學研究所碩士論文，民國90年6月。

[10]鐘爵旭，駕駛者心智負荷模式之發展及驗證，國立交通大學交通運輸研究所碩士論文，民國97年6月。

[11]周義華、陳天賜，「公車站對道路交通影響之研究—公車站對車流延滯之影響」，運輸計畫季刊，11卷3期，pp.303~340，民國71年9月。



## 附錄 程式碼(turbo c)

```
#include<graphics.h>
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>

#define TOTALSECONDS 180
#define TIMESLICE 0.1
#define TOTALSTEPS (int)(TOTALSECONDS/TIMESLICE)
#define ROADLEN 1000.0
#define MAX_ACC 3.6
#define MAX_DEACC 7.2
#define MAX_SPEED 32
```

```
FILE *o;
```

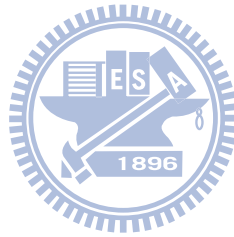
```
struct node
```

```
{
    float x;
    float y;
    int car_seq;
    float length;
    float width;
    float tall;
    float car_type;
    float v;
    float a;
    float ab;
    struct node *head;
    struct node *tail;
};
```

```
float normal(float ave,float dev);
```

```
struct node *c=NULL;
```

```
struct node *l=NULL;
```



```

struct node *r=NULL;
struct node *headc=NULL;
struct node *headl=NULL;
struct node *headr=NULL;
int C_laneGenerator(float,float,float);
int L_laneGenerator(float,float,float);
int R_laneGenerator(float,float,float);
int C_laneTransition();
int L_laneTransition();
int R_laneTransition();
float v_acc_car(float v0,float a,float t);
float x_acc_car(float x0,float v0,float a,float t);
float watchtime();
int hittime(float);
int STEPS;
int checkvision(struct node *car);
int check_car(struct node *car);
int check_carl(struct node *car);
int check_carr(struct node *car);
float visionx(float dxx, float dyy, float yy, float hyy);
float GauProb(float avex, float avey,
               float devx,float devy, float carx, float cary);

```



```

void drawlane()
{
    int x;
    struct node *current;
    current = headl;
    while(current!=NULL){
        if(current->car_type==0){
            setfillstyle(1,YELLOW);
            x=(int)current->x/ROADLEN*1000;
            bar(x,27,x+3,29);
        }
        else if(current->car_type==1){
            setfillstyle(1,YELLOW);
            x=(int)current->x/ROADLEN*1000;
            bar(x,27,x+8,29);
        }
    }
}

```

```

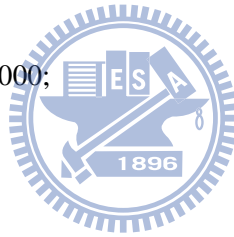
}
else{
setfillstyle(1,YELLOW);
x=(int)current->x/ROADLEN*1000;
bar(x,27,x+11,29);
}
current = current -> tail;
}

```

```

current = headc;
while(current!=NULL){
if(current->car_type==0){
setfillstyle(1,YELLOW);
x=(int)current->x/ROADLEN*1000;
bar(x,35,x+3,37);
}
else if(current->car_type==1){
setfillstyle(1,YELLOW);
x=(int)current->x/ROADLEN*1000;
bar(x,35,x+8,37);
}
else{
setfillstyle(1,YELLOW);
x=(int)current->x/ROADLEN*1000;
bar(x,35,x+11,37);
}
current = current -> tail;
}

```



```

current = headr;
while(current!=NULL){
if(current->car_type==0){
setfillstyle(1,YELLOW);
x=(int)current->x/ROADLEN*1000;
bar(x,43,x+3,45);
}
else if(current->car_type==1){
setfillstyle(1,YELLOW);

```

```

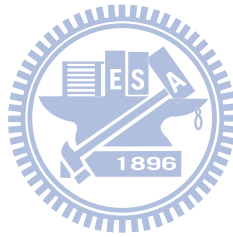
x=(int)current->x/ROADLEN*1000;
bar(x,43,x+8,45);
}
else{
setfillstyle(1,YELLOW);
x=(int)current->x/ROADLEN*1000;
bar(x,43,x+11,45);
}
current = current -> tail;
}
}

```

```

void drawRoad(void)
{
float x;
setlinestyle(0,0,1);
line(0,24,1000,24);
line(0,48,1000,48);
setlinestyle(3,0,1);
line(0,32,1000,32);
line(0,40,1000,40);
setfillstyle(4,63);
}

```



```

void show()
{
struct node *temp;

temp=headc;
fprintf(o,"C-lane:\n");
fprintf(o,"Car\tx\tv\ta\tdx\tdy\thv\ttype\n");
while(temp){
fprintf(o,"%03d:\t%.3f\t%.3f\t%.3f\t%.3f\t",
temp->car_seq,temp->x,temp->v,temp->a,temp->head->v);
fprintf(o,"\n");
temp=temp->tail;
}
}

```

```

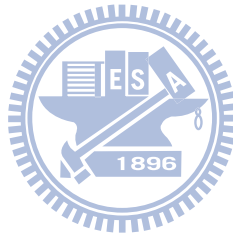
}
temp=headl;
fprintf(o,"L-lane:\n");
fprintf(o,"Car\tx\tv\ta\tdx\tdy\thv\n");
while(temp){
fprintf(o,"%03d:\t%.3f\t %.3f\t%.3f\t%.3f\t",
temp->car_seq,temp->x,temp->y,temp->a,temp->head->y);
fprintf(o,"\n");
temp=temp->tail;
}

temp=headr;
fprintf(o,"R-lane:\n");
fprintf(o,"Car\tx\tv\ta\thv\n");
while(temp){
fprintf(o,"%03d:\t%.3f\t %.3f\t%.3f\t",
temp->car_seq,temp->x,temp->y,temp->a,temp->head->y);
fprintf(o,"\n");
temp=temp->tail;
}

fprintf(o,"-----\n");
}

int main()
{
//    randomize();
float smallcar,bus;
float flow;
struct node *temp;
int k,l;
int graphdriver=DETECT,graphmode=DETECT;
initgraph(&graphdriver,&graphmode,"E:\\TC\\BGI");
if((o=fopen("file01.xls","w"))==NULL){
fprintf(o,"Open file file01.txt for writing error\n");
return -1;
}
}

```



```

printf("Flow is:");
scanf("%f",&flow);
flow/=60;

printf("car percent is:");
scanf("%f",&smallcar);

printf("bus percent is:");
scanf("%f",&bus);

for(STEPS=0;STEPS<TOTALSTEPS;STEPS++)
{
    drawRoad();
    C_laneGenerator(flow,smallcar,bus);
    L_laneGenerator(flow,smallcar,bus);
    R_laneGenerator(flow,smallcar,bus);

    drawlane();
    C_laneTransition();
    L_laneTransition();
    R_laneTransition();
    fprintf(o,"\t\t\t\t\t%.2f\n",watchtime());
    show();
    for(k=0;k<30000;k++)
    for(l=0;l<1000;l++);
    cleardevice();
}
closegraph();
getchar();
return 0;
}

```



```

int C_laneGenerator(float flow,float smallcar,float bus)
{
    static float next_car_time=0;
    static int carseq=0;
    struct node *temp;

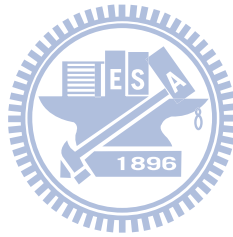
```

```

float aa,tt;
if(c&&((c->x<5)<0)){
next_car_time+=0.1;
return 1;
}
if(hittime(next_car_time)){
temp=(struct node *)malloc(sizeof(struct node));
temp->head=c;
temp->tail=NULL;

if(c!=NULL)c->tail=temp;
else headc=temp;
aa=random(100);
c=temp;
temp->bu=bus;
temp->sma=smallcar;
temp->x=0;
temp->y=10;
temp->v=normal(22.22,2.5);
temp->a=0;
temp->dx=0;
temp->dy=0;
temp->ab=aa;
if((temp->sma>temp->ab)){
temp->car_type=0;
temp->length=4;
temp->width=2;
temp->tall=1.3;
}
else if(aa<(temp->sma+temp->bu)){
temp->car_type=1;
temp->length=10;
temp->width=2.5;
temp->tall=4.1;
}
else if(100>=aa){
temp->car_type=2;
temp->length=9;

```





```

temp->width=2.5;
temp->tall=4.1;
}
temp->car_seq=++carseq;
if(temp->car_seq==1)
temp->v=22.22;

tt=normal(60/flow,1);
if(tt>0.1)

next_car_time+=tt;
else
next_car_time+=0.1;
fprintf(o,"C:%03d is born\n",carseq);
}
}

```

```

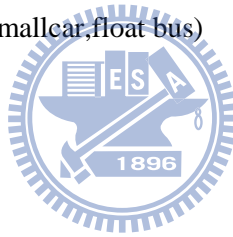
int L_laneGenerator(float flow,float smallcar,float bus)

```

```

{
    static float next_car_time=0;
    static int carseq=0;
    struct node *temp;
    int aa;
    float tt;
    if(l&&(l->x<5)){
        next_car_time+=0.1;
        return 1;
    }
    if(hittime(next_car_time)){
        temp=(struct node *)malloc(sizeof(struct node));
        temp->head=1;
        temp->tail=NULL;
        if(l!=NULL)l->tail=temp;
        else headl=temp;
        aa=random(100);
        l=temp;
        temp->x=0;
        temp->y=6.25;
    }
}

```



```

temp->v=normal(22.22,2.5);
temp->a=0;
temp->dx=0;
temp->dy=0;
temp->ab=aa;
if(aa<smallcar){
temp->car_type=0;
temp->length=4;
temp->width=2;
temp->tall=1.3;
}
else if(aa<(smallcar+bus)){
temp->car_type=1;
temp->length=10;
temp->width=2.5;
temp->tall=4.1;
}
else if(aa<100){
temp->car_type=2;
temp->length=9;
temp->width=2.5;
temp->tall=4.1;
}
temp->car_seq=++carseq;
if(temp->car_seq==1)
temp->v=22.22;
tt=normal(60/flow,1);
if(tt>0.1)

next_car_time+=tt;
else
next_car_time+=0.1;
fprintf(o,"L:%03d is born\n",carseq);
}
}

```



```

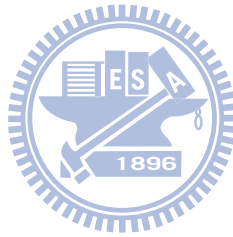
int R_laneGenerator(float flow,float smallcar,float bus)
{

```

```

static float next_car_time=0;
static int carseq=0;
struct node *temp;
int aa;
float tt;
if(r&&(r->x<5)){
next_car_time+=0.1;
return 1;
}
if(hittime(next_car_time)){
temp=(struct node *)malloc(sizeof(struct node));
temp->head=r;
temp->tail=NULL;
if(r!=NULL)r->tail=temp;
else headr=temp;
aa=random(100);
r=temp;
temp->x=0;
temp->y=13.75;
temp->v=normal(22.22,2.5);
temp->a=0;
temp->dx=0;
temp->dy=0;
if(aa<smallcar){
temp->car_type=0;
temp->length=4;
temp->width=2;
temp->tall=1.3;
}
else if(aa<(smallcar+bus)){
temp->car_type=1;
temp->length=10;
temp->width=2.5;
temp->tall=4.1;
}
else if(aa<100){
temp->car_type=2;
temp->length=9;

```



```

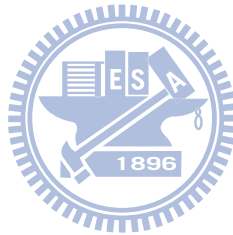
temp->width=2.5;
temp->tall=4.1;
}

temp->car_seq=++carseq;
if(temp->car_seq==1)
temp->v=22.22;
tt=normal(60/flow,1);
if(tt>0.1)

next_car_time+=tt;
else
next_car_time+=0.1;

fprintf(o,"R:%03d is born\n",carseq);
}
}

```



```

int C_laneTransition()
{
    struct node *current;
    current=headc;
    while(current!=NULL){
        checkvision(current);
        check_car(current);
        current->x=x_acc_car(current->x,current->v,current->a,TIMESLICE);
        current->v=v_acc_car(current->v,current->a,TIMESLICE);
        current=current->tail;
    }
}

```

```

int L_laneTransition()
{
    struct node *current;
    current=headl;
    while(current!=NULL){

```

```

    checkvision(current);
    check_carl(current);
    current->x=x_acc_car(current->x,current->v,current->a,TIMESLICE);
    current->v=v_acc_car(current->v,current->a,TIMESLICE);
    current=current->tail;
}
}

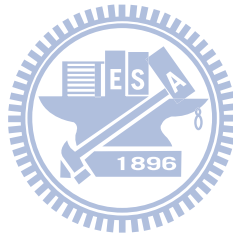
```

```
int R_laneTransition()
```

```

{
    struct node *current;
    current=headr;
    while(current!=NULL){
        checkvision(current);
        check_carr(current);
        current->x=x_acc_car(current->x,current->v,current->a,TIMESLICE);
        current->v=v_acc_car(current->v,current->a,TIMESLICE);
        current=current->tail;
    }
}

```



```
int hittime(float i)
```

```

{
    float j;
    j=watchtime();
    if((i>(j-TIMESLICE/2))&&(i<=(j+TIMESLICE/2)))
        return 1;
    return 0;
}

```

```
float watchtime()
```

```

{
    return STEPS*TIMESLICE;
}

```

```
float x_acc_car(float x0,float v0,float a,float t)
```

```

{
    float x=0.0,v,t1;

```

```

v=v0+a*t;
x=v0*t+0.5*a*t*t;
if(v0>(MAX_SPEED-0.2))
    x=v0*t;
else if(v>MAX_SPEED){
    t1=(MAX_SPEED-v0)/a;
    x=0.5*(2*t-t1)*(MAX_SPEED-v0)+v0*t;
}
else if(v<0)
    x=v0*v0/(2*a);
return x+x0;
}

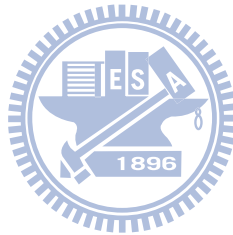
```

```
float v_acc_car(float v0,float a,float t)
```

```

{
    float v;
    v=v0+a*t;
    if(v>MAX_SPEED)
        return MAX_SPEED;
    else if(v<0)
        return 0.0;
    return v;
}

```



```
int checkvision(struct node *car){
```

```

    if(car->head!=NULL)
        car->dx=(car->head->x-car->head->length-car->x+1);
    else{
        return 0;
    }
    if((car->v<(700.0/36))){
        car->dy= tan((-7/12.0*car->v*3.6+220/3)*3.14/180)*(car->dx);
    }
    else if (car->v>=(700.0/36)){
        car->dy= tan((-5/12.0*car->v*3.6+185/3)*3.14/180)*(car->dx);
    }
}

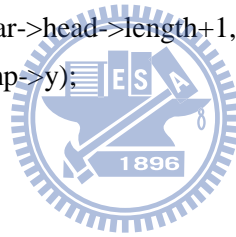
```

```

int check_car(struct node *car)
{
    struct node *temp=NULL;
    struct node *temp1=NULL;
    float leh=0,eh,reh=0,acc=0;

    temp=headl;
    temp1=headr;
    eh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,car->head->x-car->head->length+1,car->head->y);
    while(temp!=NULL){
        if((((temp->x-temp->length-car->head->x+1)*(temp->x-temp->length-car->head->x+1)
<visionx(car->dx,car->dy,temp->y,car->head->y))
&&(((temp->tail->x-temp->tail->length-car->head->x+car->head->length)
*(temp->tail->x-temp->tail->length-car->head->x+car->head->length)
>visionx(car->dx,car->dy,temp->y,car->head->y))))){
            leh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,temp->x-temp->length+1,temp->y);
            break;
        }
        temp=temp->tail;
    }
    while(temp1!=NULL){
        if(((temp1->x-temp1->length-car->head->x+1)*(temp1->x-temp1->length-car->head->x+1)
<visionx(car->dx,car->dy,temp1->y,car->head->y))
&&(((temp1->tail->x-temp1->tail->length-car->head->x+car->head->length)
*(temp1->tail->x-temp1->tail->length-car->head->x+car->head->length)
>visionx(car->dx,car->dy,temp1->y,car->head->y))))){
            reh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,temp1->x-temp1->length+1,temp1->y);
            break;
        }
        temp1=temp1->tail;
    }
}

```

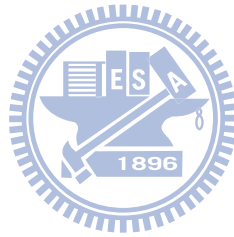


```

acc=7.235*(leh/(leh+eh+reh))*(temp->width*temp->tall/(temp->x-car->x-temp->length+1))*(temp
->v-car->v)
    +eh/(leh+eh+reh)*(car->head->width*car->head->tall/(car->dx))*(car->head->v-car->v)

+reh/(leh+eh+reh)*(temp1->width*temp1->tall/(temp1->x-car->x-temp1->length+1))*(temp1->v-
car->v));
    if(car->head==NULL)
        car->a=0;
    else if((-car->head->v*car->head->v+car->v*car->v)/(2*7.2)>(car->head->x-car->x))
        car->a=-7.2;
    else if(acc>3.6)
        car->a=3.6;
    else if(acc<-7.2)
        car->a=-7.2;
    else
        car->a=acc;
}

```



```

int check_carl(struct node *car)
{
    struct node *temp=NULL;

    float leh=0,ceh=0,acc=0;

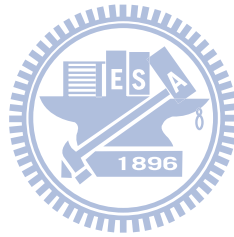
    temp=headc;
    leh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,car->head->x-car->head->length+1,car->head->y);
    while(temp!=NULL){
        if(((temp->x-temp->length-car->head->x+1)*(temp->x-temp->length-car->head->x+1)
<visionx(car->dx,car->dy,temp->y,car->head->y)
&&((temp->tail->x-temp->tail->length-car->head->x+car->head->length)
*(temp->tail->x-temp->tail->length-car->head->x+car->head->length)
>visionx(car->dx,car->dy,temp->y,car->head->y)))){
            ceh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,temp->x-temp->length+1,temp->y);
            break;
        }
    }
}

```



```
temp=temp->tail;
}
```

```
acc=7.235*(leh/(leh+ceh))*(car->head->width*car->head->tall/(car->dx))*(car->head->v-car->v)
+ceh/(leh+ceh)*(temp->width*temp->tall/(temp->x-car->x-temp->length+1))*(temp->v-car->v));
if(car->head==NULL)
car->a=0;
else if((-car->head->v*car->head->v+car->v*car->v)/(2*7.2) > (car->head->x-car->x) )
car->a=-7.2;
else if(acc>3.6)
car->a=3.6;
else if(acc<-7.2)
car->a=-7.2;
else
car->a=acc;
}
```



```
int check_carr(struct node *car)
{
struct node *temp=NULL;
float ceh=0,reh=0,acc=0;

temp=headc;
reh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,car->head->x-car->head->length+1,car->head->y);
while(temp!=NULL){
if(((temp->x-temp->length-car->head->x+1)*(temp->x-temp->length-car->head->x+1)
<visionx(car->dx,car->dy,temp->y,car->head->y)
&&((temp->tail->x-temp->tail->length-car->head->x+car->head->length)
*(temp->tail->x-temp->tail->length-car->head->x+car->head->length)
>visionx(car->dx,car->dy,temp->y,car->head->y))){
ceh=GauProb(car->head->x-car->head->length+1,car->head->y,car->dx,car->dy
,temp->x-temp->length+1,temp->y);
break;
}
```

```

    }
    temp=temp->tail;
}

acc=7.235*(reh/(reh+ceh)*(car->head->width*car->head->tall/(car->dx))*(car->head->v-car->v)
+ceh/(reh+ceh)*(temp->width*temp->tall/(temp->x-car->x-temp->length+1))*(temp->v-car->v));
    if(car->head==NULL)
        car->a=0;
    else if((-car->head->v*car->head->v+car->v*car->v)/(2*7.2)>(car->head->x-car->x))
        car->a=-7.2;
    else if(acc>3.6)
        car->a=3.6;
    else if(acc<-7.2)
        car->a=-7.2;
    else
        car->a=acc;
}

float car_safe(struct node *car)
{
    float dmin;
    struct node*temp;

    dmin=(-temp->v*temp->v+car->v*car->v)/(2*7.2);

    return dmin;
}

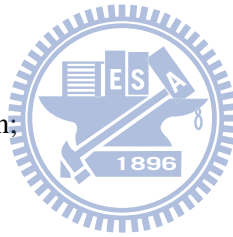
float visionx(float dxx, float dyy, float yy, float hyy)
{
    return (1-(yy-hyy)*(yy-hyy)/(dyy*dyy))*dxx*dxx;
}

```



```
float normal(float mean,float stddev)
```

```
{  
    static double V2,fac;  
    static int phase =0;  
    double S, Z, U1, U2, V1;  
    if(phase)  
        Z=V2*fac;  
    else{  
        do{  
            U1=(double)rand()/RAND_MAX;  
            U2=(double)rand()/RAND_MAX;  
            V1=2*U1-1;  
            V2=2*U2-1;  
            S=V1*V1+V2*V2;  
        }while(S>=1);  
        fac = sqrt(-2*log(S)/S);  
        Z= V1*fac;  
    }  
    phase =1- phase;  
    return((float)Z)*stddev+mean;  
}
```



```
float GauProb(float meanx, float meany,  
              float sttdevx,float sttdevy, float carx, float cary)
```

```
{  
    double AA;  
    double BB;  
  
    AA=-1/2*((carx-meanx)*(carx-meanx)/(sttdevx*sttdevx)+  
            (cary-meany)*(cary-meany)/(sttdevy*sttdevy));  
    BB=exp(AA);  
    return BB/(2*3.14*sttdevx*sttdevy);  
}
```