

元智大學工程學院機械工程學系
Department of Mechanical Engineering
College of Engineering
Yuan Ze University

孤輪阿罵的飆速輪椅
機械工程實務進度報告 1

1100826 王子晨

1100854 蘇威全

1100862 施廷翰

1100812 魏羽暘

1100861 盧昀序

指導教授：江右君、翁芳柏、余念一、吳昌暉

目錄

1 緒論	1
1.1 問題敘述	1
1.2 設計規範	1
1.3 任務分工	2
2 設計理念	2
2.1 車體設計	2
2.2 風扇與葉片設計	5
2.3 電機與轉速	6
2.4 風道與結構設計	7
2.5 空氣動力學設計	7
2.6 演算法設計	8
3 模擬與分析	27
3.1 影像偵測演算法模擬	27
4 製程與實作	32
5 測試與驗證	32
5.1 風扇測試	32
6 結論與反思	34
7 參考文獻	34
8 附錄	35
8.1 程式碼	35
8.2 物料清單與發票收據	35

圖目錄

1	競賽跑道	1
2	車體結構草圖	2
3	影像辨識系統草圖	3
4	轉向、燈光系統草圖	3
5	自動控制系統草圖	4
6	輪椅草圖	4
7	硬體架構圖	5
8	影像循線演算法流程圖	9
9	車輛速度定義	9
10	影像預處理流程圖	9
11	相機拍攝之原始影像	10
12	車輛座標系與像素坐標系關係之示意圖	10
13	透視變換流程圖	11
14	尚未滑動之偵測視窗	14
15	滑動後之偵測視窗	14
16	擬合後曲線	15
17	擬合後曲線與車輛座標系之 x 軸	16
18	偏差角度 $angle(\theta)$ 隸屬函數	18
19	位置偏移量 $distance(mm)$ 隸屬函數	19
20	前進速度 V_x 隸屬函數	20
21	側向數度 V_y 隸屬函數	22
22	角速度 ω 隸屬函數	23
23	不同偏差角度及偏移量對應之前進速度 V_x	25
24	不同偏差角度及偏移量對應之側向數度 V_y	26
25	不同偏差角度及偏移量對應之角速度 ω	26
26	尚未進行透視變換之原始圖片	27
27	經透視變換後影像	27
28	實際測量地板尺寸	28
29	待測試之原始圖片	28
30	經灰階處理之圖片	29
31	經高斯模糊處理之圖片	29
32	經二值化處理之圖片	30
33	經邊緣偵測之圖片	30
34	尚未滑動之偵測視窗	31
35	滑動後偵測視窗	31
36	擬合後曲線與車輛行駛方向比較	32
37	風扇開度對應理論風速曲線圖	33

38	風扇開度對應電流曲線圖	33
39	風扇開度對應功率曲線圖	33
40	風扇開度對應推力曲線圖	34
41	風扇開度對應力效曲線圖	34
42	bom 表	36

表目錄

1	風扇規格	6
2	V_x 控制規則表	24
3	V_y 控制規則表	24
4	ω 控制規則表	24
5	透視變換誤差	28

1 緒論

1.1 問題敘述

- 測試開始前車輛需整車放置於起跑區中，感測器可與框線對齊。
- 車輛完全自主循跡，以自主風力驅動，不得以遙控或遠端修改程式控制車輛。
- 測試時間內可隨時返回起跑區重新測試，取單趟最短完成時間為期末展現分數依據。測試時間為 5 分鐘。
- 測試過程中有下述違規事項者，返回起跑區域重新開始，期間不停錶
- 未照循跡行進：
 1. 行進過程中，車體有部分離開賽道範圍(以區域框線外緣為準)
 2. 於暫停區和終點區時，車體全部投影面積需靜止於區內至少 3 秒
 3. 以外力碰觸或影響車體

競賽跑道規格如下圖所示。

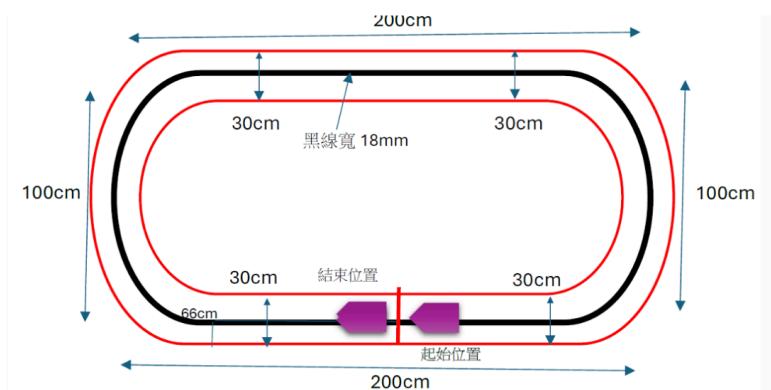


圖 1：競賽跑道

1.2 設計規範

1. 車體大小不超過 A4 ($21.0\text{ cm} \times 29.7\text{ cm}$)。
2. 車輛僅用風力驅動，不得有其他動力來源。至少 3 秒
3. 扇葉需自行設計製造，且須帶有保護裝置以防扇葉飛出造成傷害。

4. 車體載重設計須外加負載 250g，模擬車手重量。
5. 最終總成本不得超過 3000 元新台幣，且所有元件皆需發票或加工證明，不可外包。
6. 機電系統需自行組裝與測試，可購買市售相關零組件。

1.3 任務分工

- 1100826 王子晨：演算法設計、期末報告撰寫
- 1100854 蘇威全：風扇設計、流力分析
- 1100862 施廷翰：車體設計
- 1100812 魏羽暘：簡報製作
- 1100861 盧昀序：製作阿罵

2 設計理念

2.1 車體設計

2.1.1 車體結構

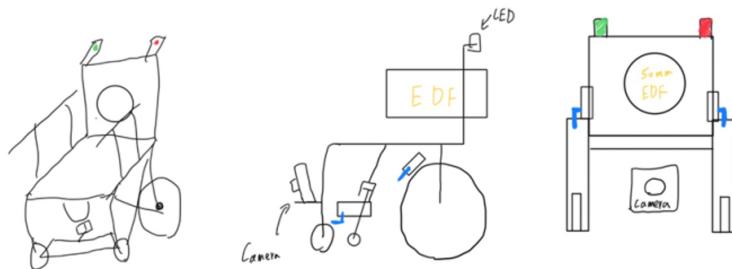


圖 2: 車體結構草圖

2.1.2 影像辨識系統

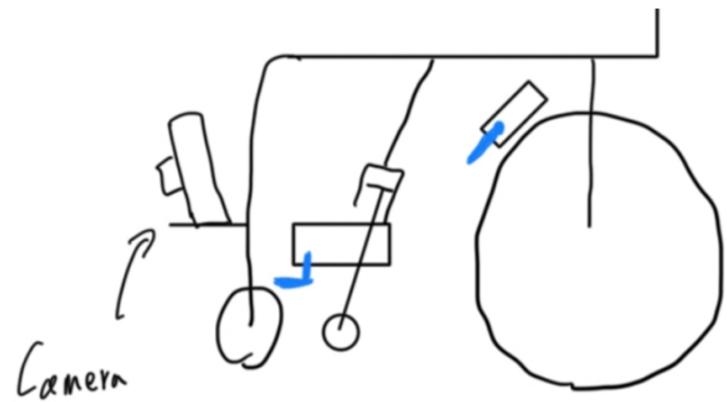


圖 3: 影像辨識系統草圖

2.1.3 轉向、燈光系統

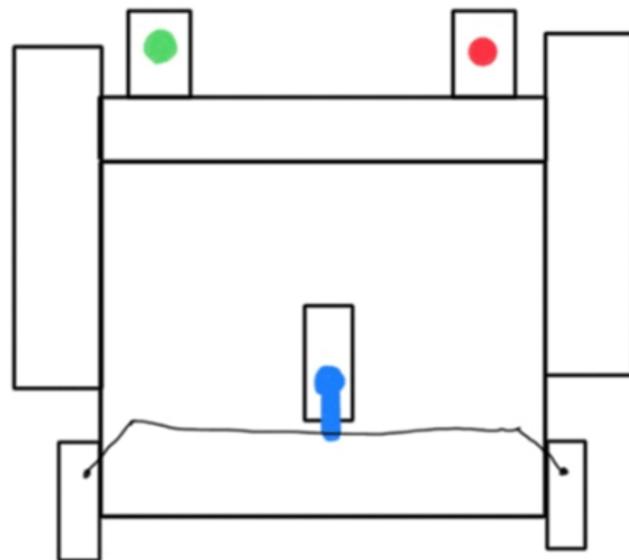


圖 4: 轉向、燈光系統草圖

2.1.4 自動控制系統



圖 5：自動控制系統草圖

2.1.5 輪椅草圖

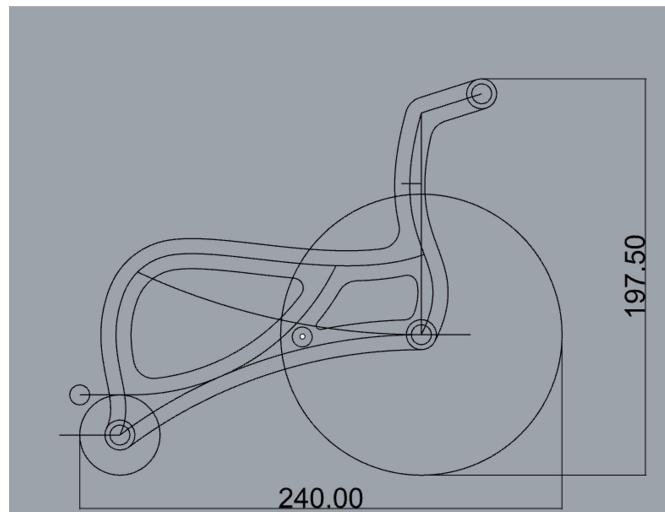


圖 6：輪椅草圖

2.1.6 硬體架構

本系統以 Raspberry Pi 3B+ 作為核心控制器，負責影像處理、邏輯判斷與用戶介面等高層運算任務。考量到 Raspberry Pi 使用 3.3V TTL 電平，而 Arduino 採用 5V TTL，因此選擇透過 UART 通訊介面進行資料交換，可有效簡化接線、降低相容性問題，並避免馬達啟停時瞬間電流對主控板造成干擾。

在系統架構中，所有感測元件（如攝影機、IMU、按鍵等）皆直接連接至 Raspberry Pi，由其統一進行資料擷取與分析；而 Arduino 則專職負責執行層控制任務，包括：控制 EDF 風扇啟動與停止與控制伺服馬達與四連桿機構實現轉向動作。此種分層架構不僅有助於程式模組化管理，也能提升系統穩定性與維護彈性。

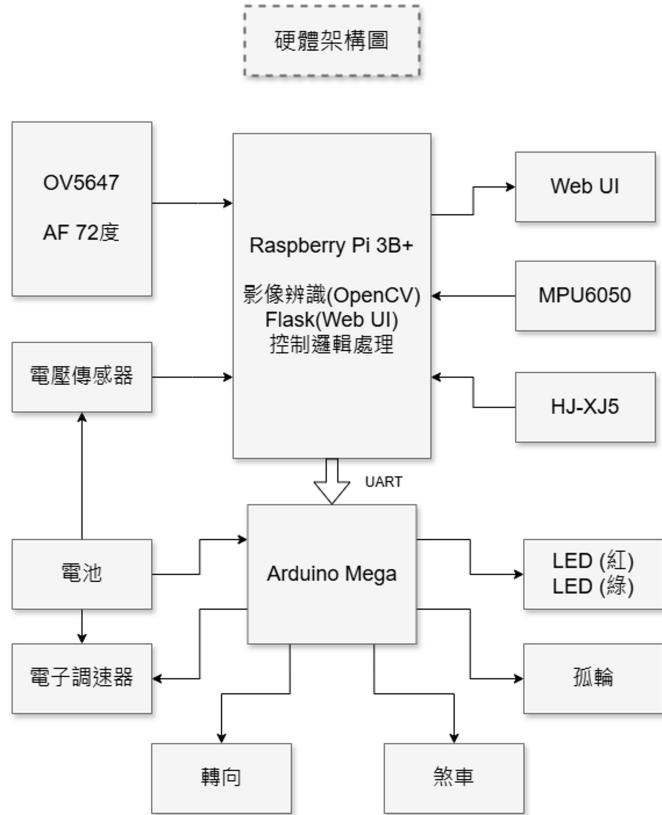


圖 7: 硬體架構圖

2.2 風扇與葉片設計

2.2.1 動力來源

風扇通常由電動機（馬達）驅動，將電能轉換為機械能，使葉片旋轉。

2.2.2 風扇設計

葉片的形狀與角度（攻角）決定了風扇的效率與風量。葉片旋轉時，根據伯努利原理與動量守恆，空氣從低壓區流向高壓區，形成風流。葉片數量、長度與彎曲度也影響氣流模式（軸流、離心或混合流）。

2.2.3 葉片設計

葉片設計包含三部分，葉片數量、葉片形狀和角度、葉片長度與寬度。葉片數量方面，少葉片（35片）適合高速旋轉，風速快，但風壓較低。多葉片（7片以上）：提供較高的風壓，適合對抗阻力的應用（如散熱器、空調）。

葉片形狀與角度這方面葉片的攻角（Blade Angle）影響空氣推動力。較大攻

角 ($20\text{--}45^\circ$) 提供較高風壓，但阻力增加，可能降低轉速。較小攻角 ($10\text{--}20^\circ$) 提升風速，適合低阻力環境。扭曲葉片，可減少渦流，提高效率。仿生學設計(如模仿鯨魚胸鰭、鷹翼)可減少空氣阻力，提高流動效率。

葉片長度與寬度方面加長葉片可提高風量，但需注意馬達負載。加寬葉片有助於提升風壓，適用於高阻力場景。

我們的設計策略是根據應用需求選擇葉片數量，確保風壓與風量的平衡。

2.3 電機與轉速

風力大小與轉速成正比需選擇高效能電機，如：無刷直流電機 (BLDC)：效率高、壽命長，適合高速風扇。感應電機 (Induction Motor)：適用於大功率風扇。

我們使用的風扇規格如表 1 所示：

表 1: 風扇規格

項目	規格
風扇直徑 D	$50\text{ mm} = 0.05\text{ m}$
葉片數	11
EDF 型號	50 mm EDF
馬達 KV 值	4500 KV
電池電壓	$3S = 11.1\text{ V}$

理論轉速計算如下：

$$\begin{aligned}\text{RPM} &= KV \times V = 4500 \times 11.1 \\ &= 49950\text{ RPM}\end{aligned}\tag{1}$$

風扇出口速度計算式 2， V_e 為出口速度，RPM 為馬達轉速，D 為風扇直徑， K_e 為出口速度係數(通常介於 0.1 至 0.2)。

$$\begin{aligned}V_e &= K_e \cdot \text{RPM} \cdot D \\ &= 0.1 \cdot 49950 \cdot 0.05 \\ &= 249.75\text{ m/s}\end{aligned}\tag{2}$$

2.4 風道與結構設計

2.4.1 風扇外框與導流設計

- 加裝導風罩 (Shroud) 可集中氣流，提高風速與效率。
- 收縮式進風口 (如噴嘴設計) 可增加流速，提高出風動能。
- 擾流板 (Guide Vanes) 可減少渦流，提升氣流穩定度。

2.4.2 風扇類型選擇

- 軸流風扇 (Axial Fan) 適合長距離送風 (如電腦散熱、風扇塔)。
- 離心風扇 (Centrifugal Fan) 適合高風壓應用 (如空調、渦輪增壓器)。
- 混流風扇 (Mixed Flow Fan) 結合兩者優勢，提高風壓與風量。

2.4.3 風扇導流罩

主要作用是提升風扇的推力與效率，其原理涉及氣流控制、減少渦流、提高風壓與風速。導流罩的幾個關鍵機制為增加氣流速度 (噴射效應) 與減少渦流與能量損失。風扇在運轉時，葉片末端 (特別是無外框的風扇) 會產生強烈的端部渦流 (Tip Vortex) 會導致氣流擾動降低風壓與能量損耗使推力下降。

導流罩可封閉葉片端部，減少氣流洩漏，進而提高風扇效率。目前打算設計變截面導流罩 (Converging Nozzle)，入口與出口皆有變化，可提升氣流效率，減少能量損耗。

2.5 空氣動力學設計

- 降低車身阻力 (Drag Reduction)
- 流線型設計，避免平直表面，減少風阻
- 使用導流罩 (Shroud) 引導風扇氣流，提高推進效率
- 穩定性
- 風扇的反作用力可能影響車輛平衡，需要考慮重心設計
- 若車輛速度高，可加裝尾翼，減少側風影響

推力計算如下所示， $\rho = 1.225(kg/m^3)$ 、 $V_i = 0$ 、 $D = 0.05(m)$ ：

$$\begin{aligned}
 T &= \dot{m} \cdot (V_e - V_i) \\
 &= \rho \cdot A \cdot V_e \cdot (V_e - V_i) \\
 &= \rho \cdot \frac{1}{4} \cdot D^2 \cdot \pi \cdot V_e \cdot (V_e - V_i) \\
 &= 1.225 \times \frac{\pi}{4} \times 0.05^2 \times 249.75 \times (249.75 - 0) \\
 T &= 150.0296(N)
 \end{aligned} \tag{3}$$

當汽車行駛時，會受到空氣阻力， CD 為阻力係數， ρ 為空氣密度， A 車體迎風面， v 為車速，其公式如式 4。

$$FD = \frac{1}{2} CD v^2 \rho A \tag{4}$$

降低空氣阻力的方法包含流線型車身設計（減少迎風面積）、降低車輛高度（減少底部亂流）、使用導流板（減少渦流與尾流）。

汽車在高速行駛時，車身可能會產生升力 (Lift)，導致車輛不穩定而 F1 賽車則透過空氣動力學產生下壓力 (Downforce) 來增加抓地力。升力公式如式 4。

$$FL = \frac{1}{2} CL v^2 \rho A \tag{5}$$

減少升力的方法包含加裝尾翼 (Spoiler)、底部擴散器 (Diffuser)、前擾流板 (Front Splitter)。加裝尾翼可改變氣流方向，增加下壓力。底部擴散器，增加地面效應，減少氣流擾動。前擾流板可提升車頭穩定性。

風扇驅動車是靠風扇推進，需要考慮：

1. 風扇的推力與效率：使用導流 (Shroud) 來提高推力。
2. 車身形狀：流線型設計可減少阻力，讓風扇推動更有效。
3. 尾流管理：避免車後氣流分離，降低拖曳力，提高速度。

2.6 演算法設計

循線演算法將結合影像偵測與模糊控制實現，並包含預處理、偵測、模糊控制三部分。演算法流程圖如 8 圖所示。



圖 8: 影像循線演算法流程圖

演算法的輸入為影像，演算法的輸出為車輛為 V_x 、 V_y 與 ω ，定義如下圖 9 所示。

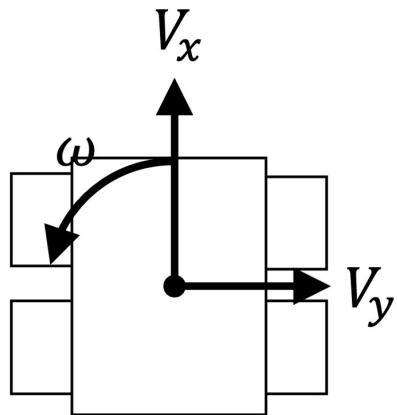


圖 9: 車輛速度定義

2.6.1 預處理

此部分要將影像拍攝影像處理成可供影像偵測之影像，包括透視變換、灰階處理、高斯模糊、二值化、Canny 邊緣偵測五步驟，將接收到的影像進行處理，提供後續偵測使用。

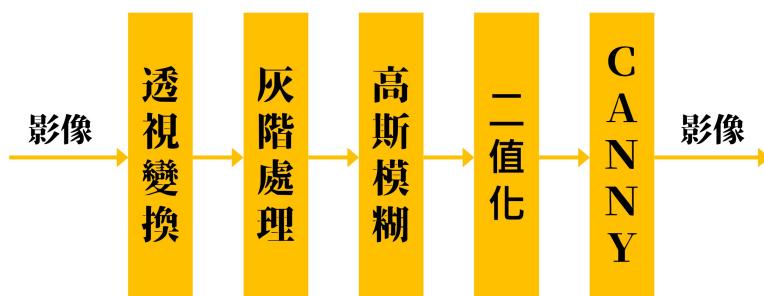


圖 10: 影像預處理流程圖

首先，對捕獲的影像進行透視變換，將影像轉換為以車輛為原點的二維座

標系統，這樣有助於確定車輛與道路的相對位置，並為後續的計算提供準確的基準。公式 8 及示意圖(如圖 11、圖 12 所示)如下：

$$\begin{bmatrix} x_{car} \\ y_{car} \end{bmatrix} = f(\begin{bmatrix} x \\ y \end{bmatrix}, W, H, \theta_{camera}, HFOV, WFOV) \quad (6)$$

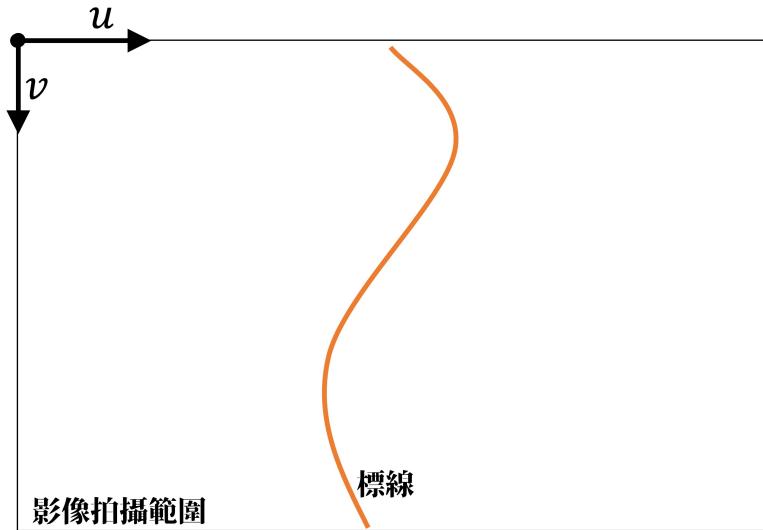


圖 11：相機拍攝之原始影像

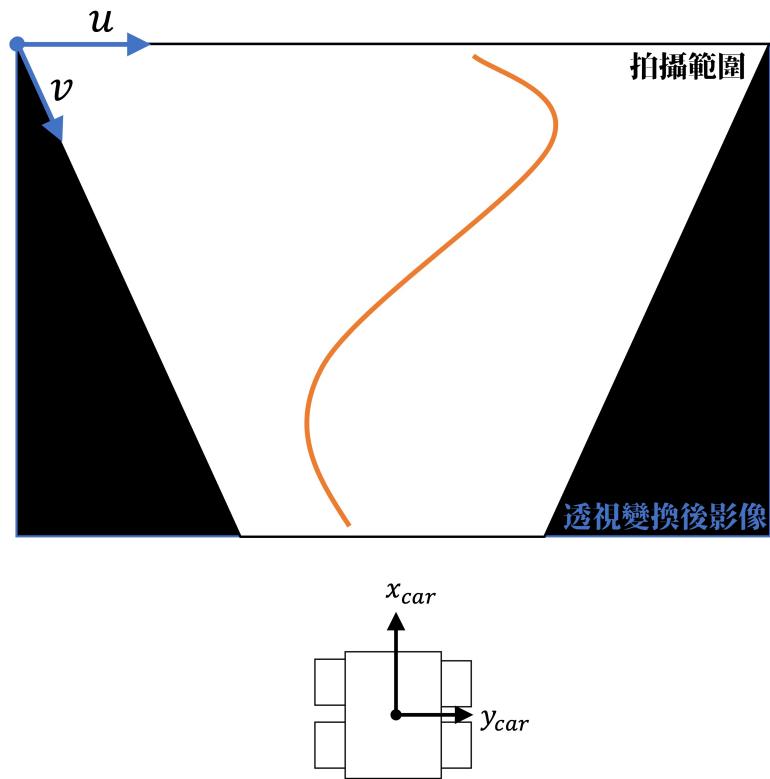


圖 12：車輛座標系與像素坐標系關係之示意圖

將利用鏡頭往斜下的角度拍攝。採用逆透視變換 (Inverse Perspective Mapping, IPM)[1][2][3] 技術處理影像，可以將像素座標轉換為地面座標，取得影像中精確的距離。

透視變換的核心在於去除透視效應，以便將影像中的道路線條轉換為平行的直線。這樣的變換有助於車輛識別路徑，使後續演算法能夠更容易地進行路徑規劃與車道偵測。

首先將影像像素座標系 (u, v) 轉換為圖像座標系 (x, y) 。加入相機的水平視角 $HFOV$ 與相機的垂直視角 $VFOV$ 、相機俯仰角 θ_{camera} ，將圖像座標轉換為相機座標 $(x_{camera}, y_{camera}, z_{camera})$ 。接著將相機座標系轉換為車輛標系 $(x_{car}, y_{car}, z_{car})$ ，以取得拍攝影像中內容物與車輛之相對位置。流程圖如下：

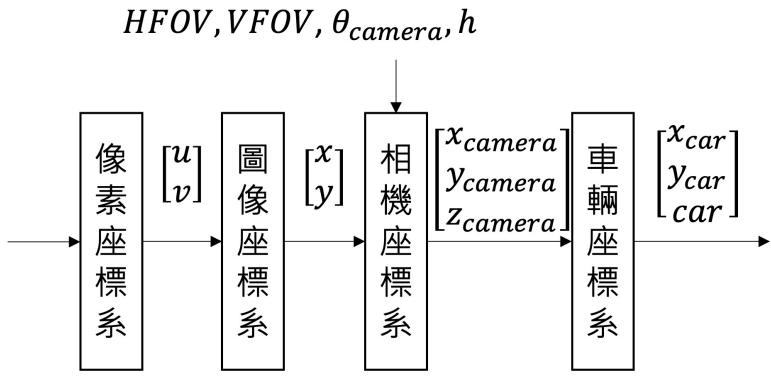


圖 13: 透視變換流程圖

將像素座標轉換為無人機相對於地平面的世界座標時，過程涉及相機的內參矩陣 K 、旋轉矩陣 R 以及相機高度 h 。要算相機的內參矩陣首先需得出影像的水平焦距 f_x 、影像的垂直焦距 f_y 和影像的中心座標 (c_x, c_y) 。

影像的水平焦距 f_x 公式如式 (7)，W 為影像的寬度，HFOV 為相機的水平視角。影像的垂直焦距 f_y 公式如式 (8)，H 為影像的寬度，VFOV 為相機的垂直視角。

$$f_x = \frac{W}{2 \cdot \tan\left(\frac{HFOV}{2}\right)} \quad (7)$$

$$f_y = \frac{H}{2 \cdot \tan\left(\frac{VFOV}{2}\right)} \quad (8)$$

影像的中心座標 (c_x, c_y) 公式如式 (9)、式 (10)：

$$c_x = \frac{W}{2} \quad (9)$$

$$c_y = \frac{H}{2} \quad (10)$$

式(11)為相機內參矩陣 K 。

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

旋轉矩陣 R 用來表示相機相對於世界座標系的旋轉。此處忽略車輛進行中的晃動。旋轉矩陣 R 如下：

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{camera}) & -\sin(\theta_{camera}) \\ 0 & \sin(\theta_{camera}) & \cos(\theta_{camera}) \end{bmatrix} \quad (12)$$

將像素座標系 (x, y) 轉換為圖像座標系的公式如式(13)：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (13)$$

將圖像座標系轉換為相機座標系的公式如式(14)：

$$\begin{bmatrix} x_{camera} \\ y_{camera} \\ z_{camera} \end{bmatrix} = R \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (14)$$

將相機座標系轉換為車輛座標系 $(x_{car}, y_{wcar}, z_{wcar})$ 的公式見式(15)、式(16)，此公式即可獲已車輛為原點的座標系統，取得每影像中每一像素與車輛的絕對距離。

$$x_w = \frac{x_c}{z_c} \quad (15)$$

$$y_w = \frac{y_c}{z_c} \quad (16)$$

灰階處理將彩色影像轉換為灰階影像，這樣能有效減少計算量，同時保留影像的基本結構特徵，便於後續處理。這樣的轉換方式基於人眼對不同顏色敏感度的權重選擇，使得轉換後的影像仍然保持較高的可辨識度。

高斯模糊用來去除影像中的噪聲，這有助於平滑影像，減少後續處理中的干擾，尤其是在邊緣檢測時。高斯模糊的公式如下：

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (17)$$

其中， σ 為標準差，決定模糊的強度。較大的 σ 值會導致更強的平滑效果，但可能會模糊掉重要的邊緣資訊。

二值化將影像轉換為二值圖像，即只保留黑白兩種顏色，這有助於強調出車道線等重要特徵，便於後續的分析與處理。固定閾值法：設定一個固定的閾值 T ，當像素強度高於 T 時設為白色，否則設為黑色。固定閾值法的公式如下。

$$I_{bin}(x, y) = \begin{cases} 1, & I(x, y) > T \\ 0, & I(x, y) \leq T \end{cases} \quad (18)$$

最後使用 Canny 邊緣偵測來提取影像中的邊緣信息，這有助於識別出車道的邊界或其他障礙物，為自走車的導航提供關鍵信息。

2.6.2 偵測

圖像經過預處理後，接下來的步驟是偵測車道標線，並計算車輛的偏移量與偏移角度。

將首先，從影像中掃描特定的區域，通過檢測這些區域的像素值來獲取相關的幾何特徵。在偵測車道標線時，並非整張影像都具有相同的重要性，因此可以選擇關鍵區域進行分析。例如，車輛前方的道路中央區域可能包含主要的行駛標線，而影像上方區域則可能包含過多的背景資訊，對偵測無實質幫助。將使用滑動視窗法，使用固定大小的窗口在影像中移動，對每個區域進行局掃描，以提取潛在的車道標線位置，並將視窗滑動偵測最多白色像素之位置。

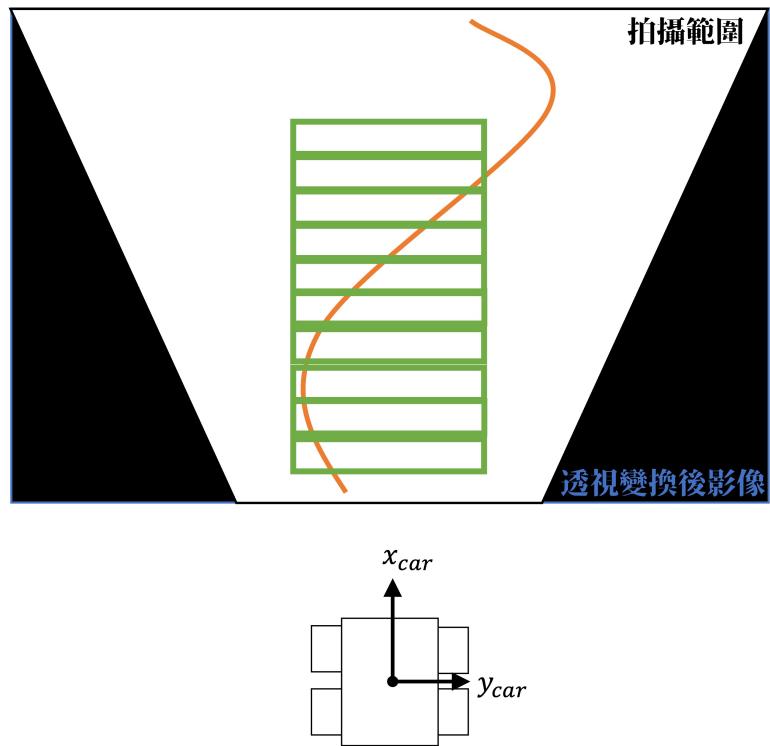


圖 14: 尚未滑動之偵測視窗

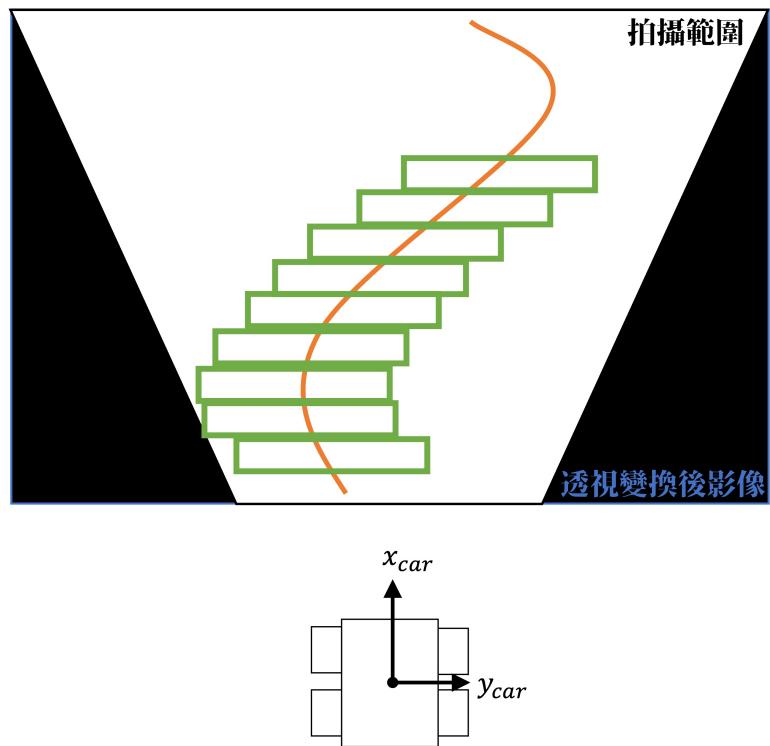


圖 15: 滑動後之偵測視窗

據掃描到的像素資料，使用多項式擬合來擬合出車道線或相關物體的曲線，這樣可以準確地描述其形狀。當獲取到標線像素點後，下一步是利用數學模型

來擬合標線的形狀。由於車道標線通常具有一定的連續性與平滑度，可以使用多項式擬合（Polynomial Fitting）來描述其形狀。

對於普通道路，車道標線通常可以近似為二次曲線：

$$y = ax^2 + bx + c \quad (19)$$

其中， a, b, c 為待求的係數。使用最小二乘法（Least Squares Method, LSM）來找到最適合這些數據點的曲線。

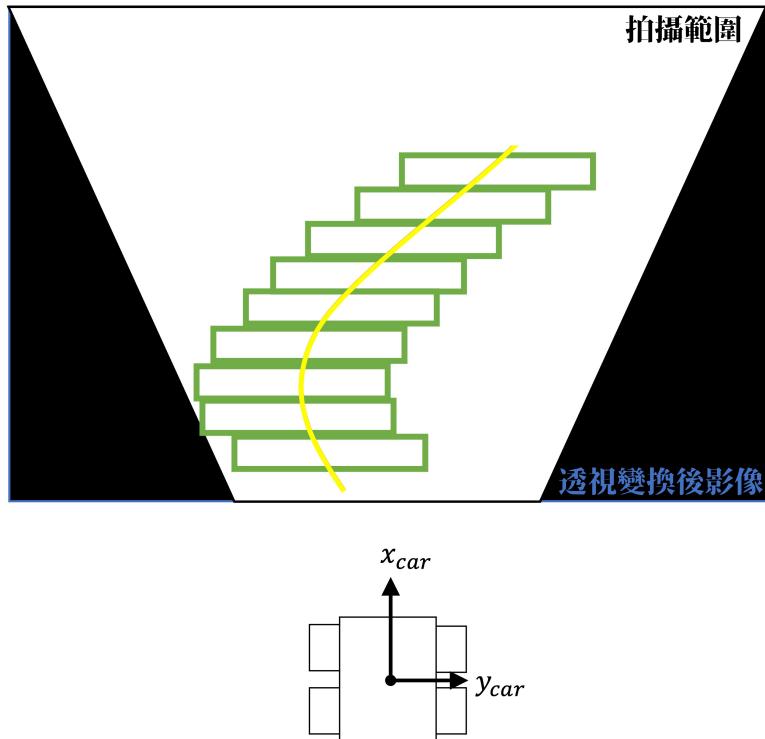


圖 16: 擬合後曲線

最後即可根據擬合的曲線與車輛座標系之 x 軸計算車輛的偏移量與偏移角度。

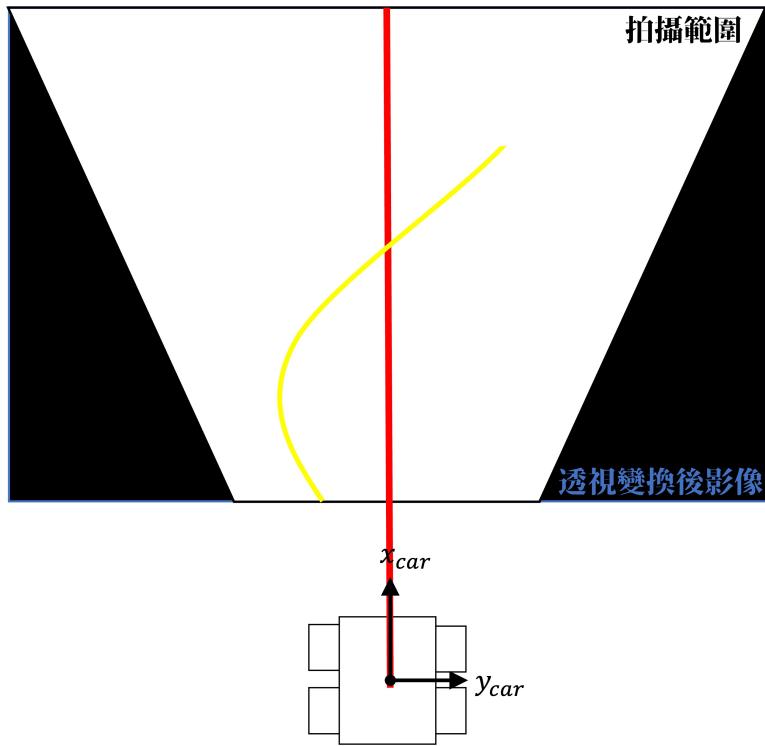


圖 17: 擬合後曲線與車輛座標系之 x 軸

根據擬合的曲線，計算出車輛當前的偏移量（與目標軌跡的水平距離）和偏移角度（與車道或目標方向的夾角）。這些數據將作為車輛調整行駛路徑的依據。

2.6.3 模糊控制

首先定義輸入變數與輸出變數。輸入變數為偏差角度 $angle(\theta)$ 及位置偏移量 $distance(mm)$ 。輸入變數為前進速度 V_x 、側向數度 V_y 與角速度 ω 。

首先定義偏差角度 $angle(\theta)$ 及位置偏移量 $distance(mm)$ 的隸屬函數。

在本系統中，輸入變數包含偏差角度 $angle(\theta)$ 與位置偏移量 $distance$ (mm)，需先將其模糊化處理。其定義如下：

$$\theta \in [-30^\circ, 30^\circ], \quad \theta \in angle = \{BL, SL, Z, SR, BR\} \quad (20)$$

$$d \in [-50 \text{ mm}, 50 \text{ mm}], \quad distance \in position = \{BL, SL, Z, SR, BR\} \quad (21)$$

上述模糊集合分別表示：

- BL：大左 (Big Left)

- SL : 小左 (Small Left)
- Z : 居中 (Zero)
- SR : 小右 (Small Right)
- BR : 大右 (Big Right)

每個模糊集合都有對應的隸屬函數 $\mu_A(x)$ ，其映射如下：

$$\mu_A(x) : X \rightarrow [0, 1] \quad (22)$$

偏差角度 $angle(\theta)$ 的隸屬函數如下：

$$angle_{BL}(x) = \begin{cases} 1 & x \leq -30 \\ \frac{x+30}{15} & -30 < x \leq -15 \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

$$angle_{SL}(x) = \begin{cases} \frac{x+30}{15} & -30 < x \leq -15 \\ \frac{-x}{15} & -15 < x \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

$$angle_Z(x) = \begin{cases} \frac{x+15}{15} & -15 < x \leq 0 \\ \frac{15-x}{15} & 0 < x \leq 15 \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

$$angle_{SR}(x) = \begin{cases} \frac{x}{15} & 0 < x \leq 15 \\ \frac{30-x}{15} & 15 < x \leq 30 \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

$$angle_{BR}(x) = \begin{cases} \frac{x-15}{15} & 15 < x \leq 30 \\ 1 & x \geq 30 \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

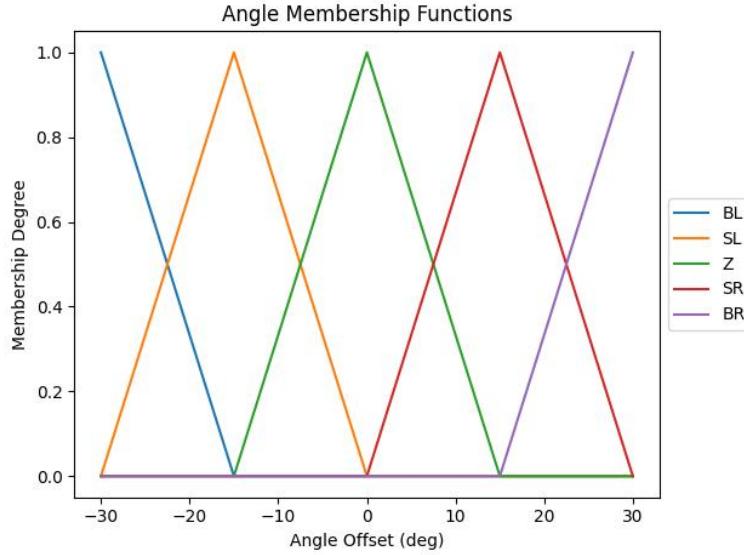


圖 18: 偏差角度 $angle(\theta)$ 隸屬函數

位置偏移量 $distance(mm)$ 的隸屬函數如下：

$$distance_{BL}(x) = \begin{cases} 1 & x \leq -50 \\ \frac{x+50}{25} & -50 < x \leq -25 \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

$$distance_{SL}(x) = \begin{cases} \frac{x+50}{25} & -50 < x \leq -25 \\ \frac{-x}{25} & -25 < x \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

$$distance_Z(x) = \begin{cases} \frac{x+25}{25} & -25 < x \leq 0 \\ \frac{25-x}{25} & 0 < x \leq 25 \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

$$distance_{SR}(x) = \begin{cases} \frac{x}{25} & 0 < x \leq 25 \\ \frac{50-x}{25} & 25 < x \leq 50 \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

$$\text{distance}_{\text{BR}}(x) = \begin{cases} \frac{x-25}{25} & 25 < x \leq 50 \\ 1 & x \geq 50 \\ 0 & \text{otherwise} \end{cases} \quad (32)$$

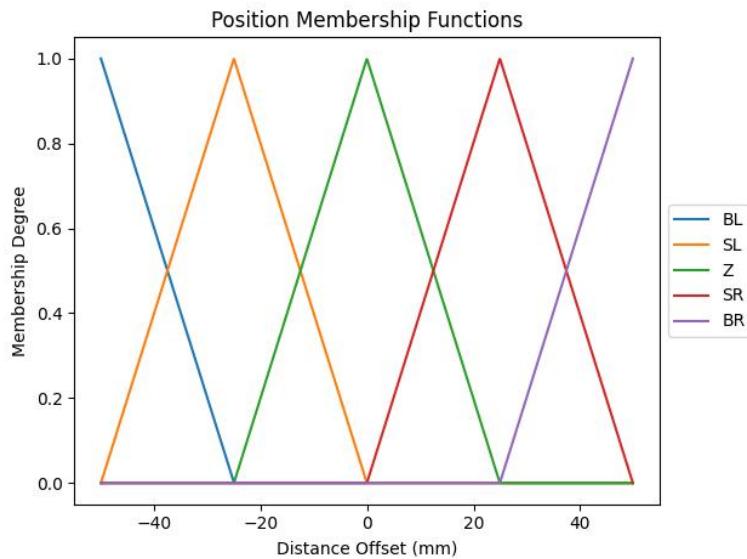


圖 19: 位置偏移量 $\text{distance}(mm)$ 隸屬函數

接著定義前進速度 V_x 、側向數度 V_y 與角速度 ω 的隸屬函數。

前進速度 V_x 定義為：

- S (Slow) : 慢速
- M (Medium) : 中速
- F (Fast) : 快速

前進速度 V_x 的隸屬函數如下：

$$Vx_S(x) = \begin{cases} 1 & 0 \leq x \leq 50 \\ \frac{100-x}{50} & 50 < x \leq 100 \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

$$Vx_M(x) = \begin{cases} \frac{x-50}{50} & 50 < x \leq 100 \\ \frac{150-x}{50} & 100 < x \leq 150 \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

$$Vx_F(x) = \begin{cases} \frac{x-100}{50} & 100 < x \leq 150 \\ 1 & 150 < x \leq 200 \\ 0 & \text{otherwise} \end{cases} \quad (35)$$

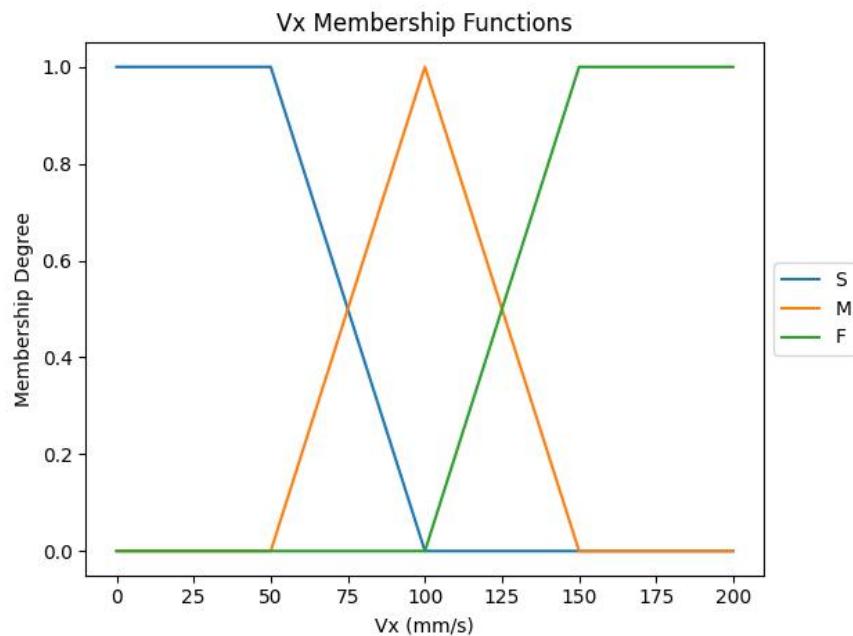


圖 20: 前進速度 V_x 隸屬函數

側向數度 V_y 定義為：

- LL (Leftmost) : 最左
- L (Left) : 左
- Z (Zero) : 中間
- R (Right) : 右
- RR (Rightmost) : 最右

側向數度 V_y 的隸屬函數如下：

$$\text{Vy}_{LL}(x) = \begin{cases} 1 & -50 \leq x \leq -40 \\ \frac{-20-x}{20} & -40 < x \leq -20 \\ 0 & \text{otherwise} \end{cases} \quad (36)$$

$$\text{Vy}_L(x) = \begin{cases} \frac{x+30}{15} & -30 < x \leq -15 \\ \frac{-x}{15} & -15 < x \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (37)$$

$$\text{Vy}_Z(x) = \begin{cases} \frac{x+15}{15} & -15 < x \leq 0 \\ \frac{15-x}{15} & 0 < x \leq 15 \\ 0 & \text{otherwise} \end{cases} \quad (38)$$

$$\text{Vy}_R(x) = \begin{cases} \frac{x}{15} & 0 < x \leq 15 \\ \frac{30-x}{15} & 15 < x \leq 30 \\ 0 & \text{otherwise} \end{cases} \quad (39)$$

$$\text{Vy}_{RR}(x) = \begin{cases} \frac{x-20}{20} & 20 < x \leq 40 \\ 1 & 40 < x \leq 50 \\ 0 & \text{otherwise} \end{cases} \quad (40)$$

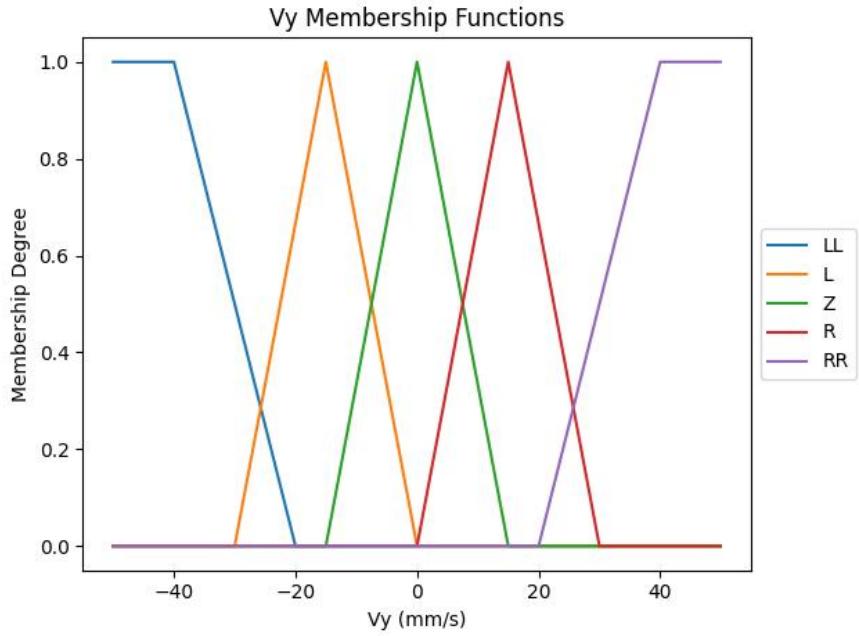


圖 21: 側向數度 V_y 隸屬函數

角速度 ω 定義為：

- CCW2 (Counterclockwise 2)：強烈逆時針
- CCW (Counterclockwise)：輕微逆時針
- Z (Zero)：無旋轉
- CW (Clockwise)：輕微順時針
- CW2 (Clockwise 2)：強烈順時針

角速度 ω 的隸屬函數如下：

$$\omega_{CCW2}(x) = \begin{cases} 1 & -20 \leq x \leq -15 \\ \frac{-10-x}{5} & -15 < x \leq -10 \\ 0 & \text{otherwise} \end{cases} \quad (41)$$

$$\omega_{CCW}(x) = \begin{cases} \frac{x+15}{5} & -15 < x \leq -10 \\ \frac{-x}{10} & -10 < x \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (42)$$

$$\omega_Z(x) = \begin{cases} \frac{x+10}{10} & -10 < x \leq 0 \\ \frac{10-x}{10} & 0 < x \leq 10 \\ 0 & \text{otherwise} \end{cases} \quad (43)$$

$$\omega_{CW}(x) = \begin{cases} \frac{x}{10} & 0 < x \leq 10 \\ \frac{15-x}{5} & 10 < x \leq 15 \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

$$\omega_{CW2}(x) = \begin{cases} \frac{x-10}{5} & 10 < x \leq 15 \\ 1 & 15 < x \leq 20 \\ 0 & \text{otherwise} \end{cases} \quad (45)$$

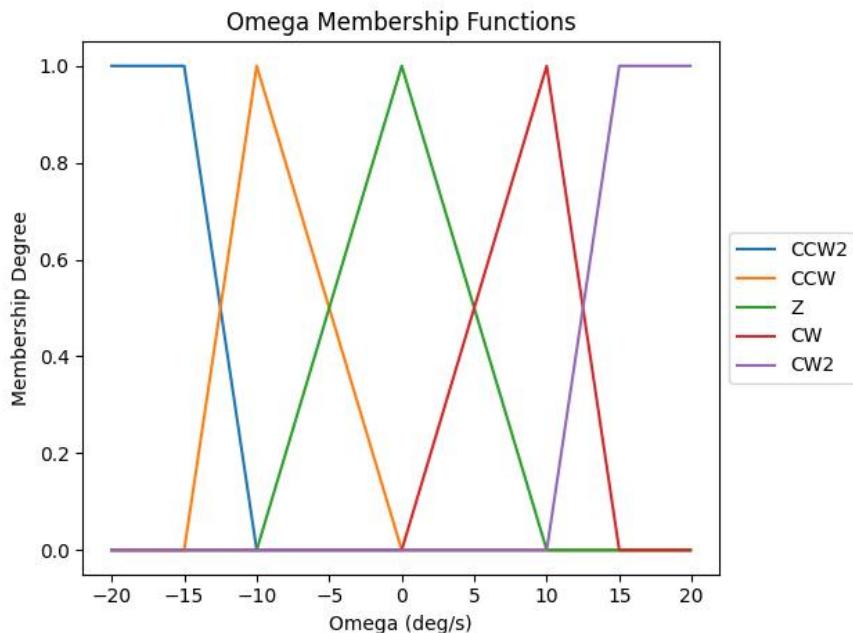


圖 22: 角速度 ω 隸屬函數

接著是定義控制規則庫， V_x 之控制規則表、 V_y 控制規則表與 ω 控制規則表，如下所示。

表 2: V_x 控制規則表

Angle($^\circ$) \ Position(cm)	BL	SL	Z	SR	BR
BL	S	S	M	S	S
SL	S	M	F	M	S
Z	M	F	F	F	M
SR	S	M	F	M	S
BR	S	S	M	S	S

表 3: V_y 控制規則表

Angle($^\circ$) \ Position(cm)	BL	SL	Z	SR	BR
BL	RR	R	Z	L	LL
SL	RR	R	Z	L	LL
Z	RR	R	Z	L	LL
SR	RR	R	Z	L	LL
BR	RR	R	Z	L	LL

表 4: ω 控制規則表

Angle($^\circ$) \ Position(cm)	BL	SL	Z	SR	BR
BL	CW2	CW2	CW2	CW2	CW2
SL	CW	CW	CW	CW	CW
Z	Z	Z	Z	Z	Z
SR	CCW	CCW	CCW	CCW	CCW2
BR	CCW2	CCW2	CCW2	CCW2	CCW2

觸發強度是由這兩個模糊集合的隸屬度中的較小者所決定。

$$\alpha_{ij} = \min (\mu_{\text{angle}}^{A_i}(\theta), \mu_{\text{position}}^{B_j}(x)) \quad (46)$$

最後進行去模糊化，計算對每個輸出變數 z ，計算：

$$z^* = \frac{\int z \cdot \mu(z) dz}{\int \mu(z) dz} \quad (47)$$

其中， $\mu(z)$ 是合併後的輸出模糊集合的隸屬度。

因為有數組輸出，利用 Alpha-Cut (α -截集) 的方式，排除異常值，僅保留後 80% 結果來避免極端值干擾。

對輸出值集合 z_i ，排序後選取前 $N \times \alpha$ 個，再取平均：

$$\bar{z}_\alpha = \frac{1}{N_\alpha} \sum_{i=1}^{N_\alpha} z_i \quad (48)$$

最後即可取得不同偏差角度 $angle$ 及位置偏移量 $distance$ 對應之前進速度 V_x (圖 23)、側向數度 V_y (圖 24) 與角速度 ω (圖 25)。

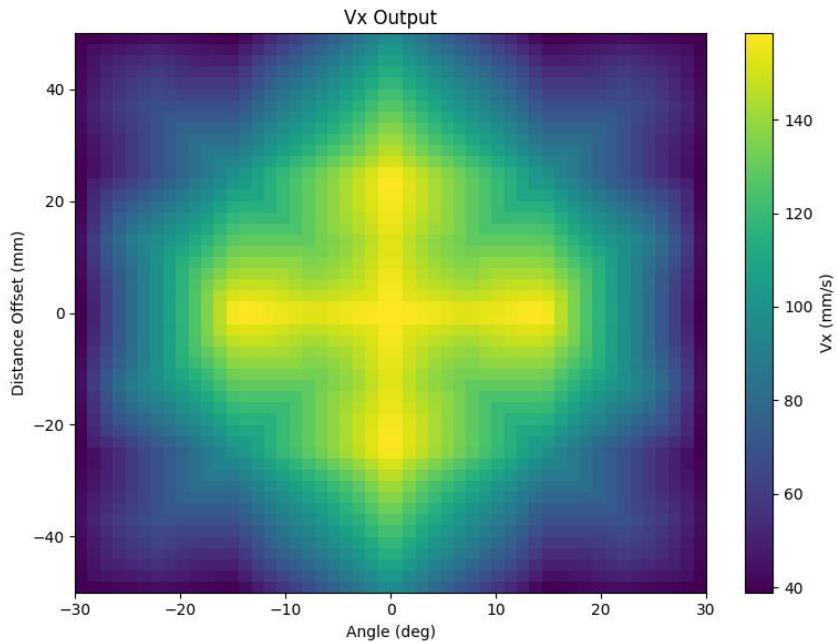


圖 23: 不同偏差角度及偏移量對應之前進速度 V_x

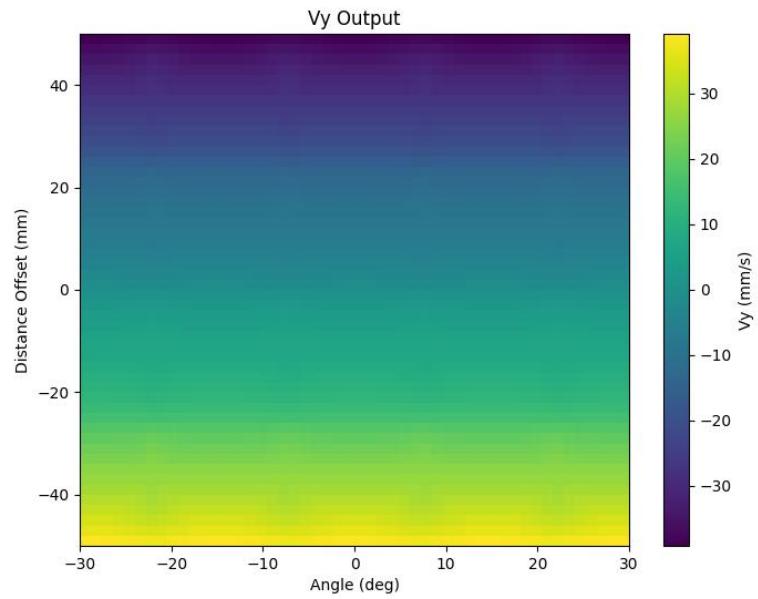


圖 24: 不同偏差角度及偏移量對應之側向數度 V_y

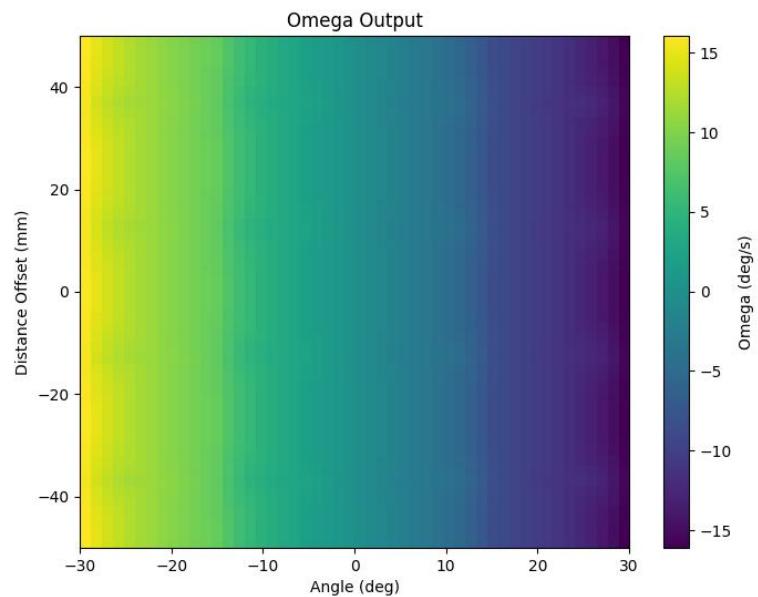


圖 25: 不同偏差角度及偏移量對應之角速度 ω

3 模擬與分析

3.1 影像偵測演算法模擬

首先是進行影像透視變換實際測試，並利用逆透視變換取得影像中特定像素點在以車輛為原點的平面座標系之座標。

影像透視變換測試結果如圖 27所示。將實際拍攝影像投影至以車輛為原點的平面上，即可計算拍攝目標物與車輛之距離，圖中座標單位為公尺。



圖 26: 尚未進行透視變換之原始圖片



圖 27: 經透視變換後影像

下圖 (如圖 28所示) 與下表 (如表 5所示) 為逆透視變換之結果，與計算之誤差，可知若參數設置準確，絕對誤差大約為 7mm 左右，且測試時相機距離地面 1.5 公尺。若離地面更近，絕對誤差理論上會更小。



圖 28: 實際測量地板尺寸

表 5: 透視變換誤差

MAE(m)	0.007
MRE(%)	1.168

接下來是進行影像偵測之模擬，測試影像處理方式是否可行，先由行車圖片進行測試，圖 29為原始圖片，圖 30為經灰階處理之圖片，圖 30為經高斯模糊處理之圖片。



圖 29: 待測試之原始圖片



圖 30: 經灰階處理之圖片



圖 31: 經高斯模糊處理之圖片

圖 32為經二值化處理之圖片，圖 33為經邊緣偵測之圖片，在偵測邊緣前會在經過一次高斯模糊，再次過濾二值化後產生的雜訊。



圖 32: 經二值化處理之圖片

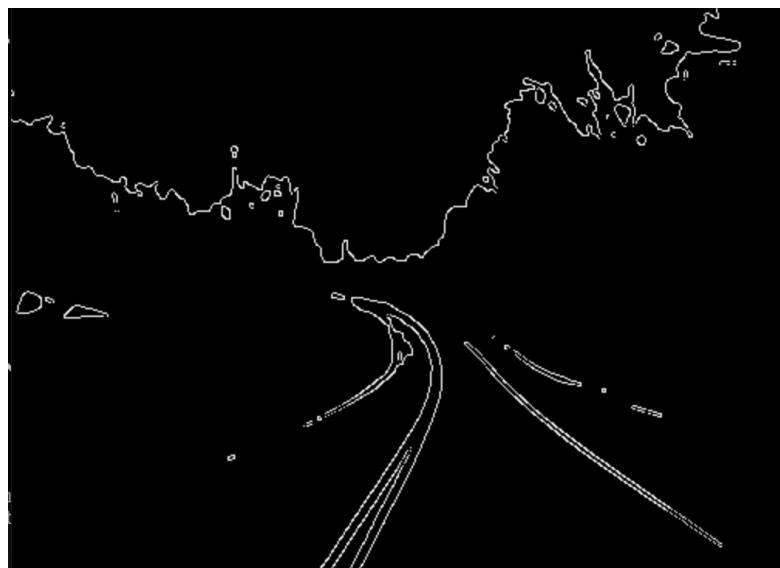


圖 33: 經邊緣偵測之圖片

經過 canny 邊緣偵測後的圖片，只有黑白兩種像素，標線邊緣為白色，其他均為黑色，因此可利用白色像素進行偵測。下圖為測試之結果。

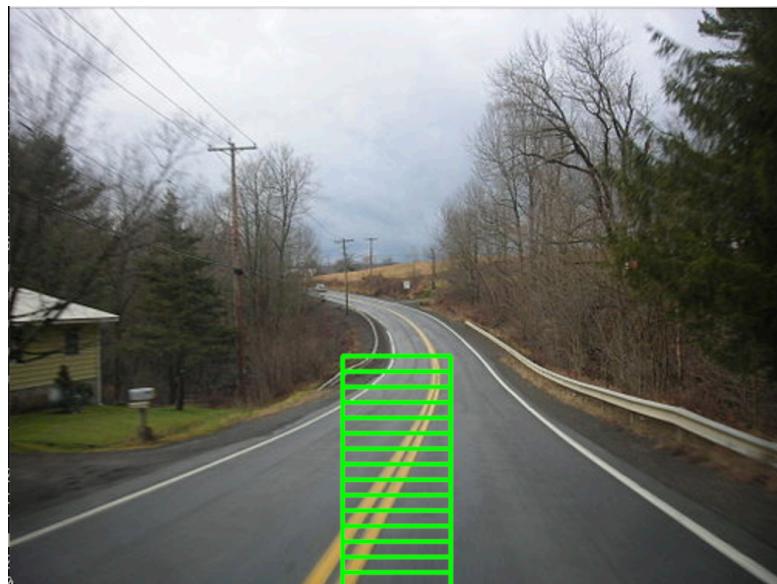


圖 34: 尚未滑動之偵測視窗

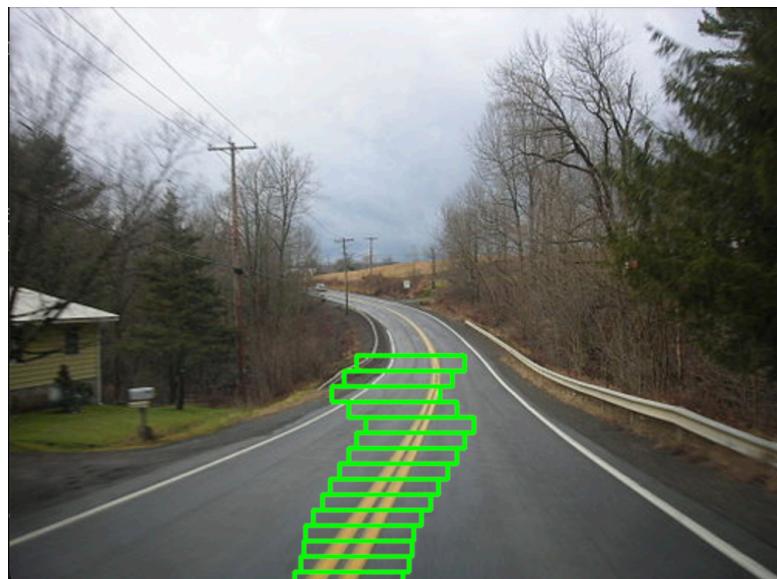


圖 35: 滑動後偵測視窗

下圖為實際擬合曲線之結果，可見圖片下方的部分大致有擬合出來，但上方的標線有偏離目標標線的現象。



圖 36: 擬合後曲線與車輛行駛方向比較

因為測試圖判沒有做透視變換處理，藉由觀察，可看到為最頂端的幾個偵測視窗，有偵測到其他條線，因此有偏離目標的現象，最後擬合的曲線在那部分也相較下方沒有完全貼合目標。由此可知透視變換之處理有其必要性。

經這幾項測試，判此演算法之可行性，可用於循線自走車。

4 製程與實作

5 測試與驗證

5.1 風扇測試

以下是對風扇進行實際測試。測試內容包含風扇開度對應理論風速(如圖37所示)、對應電流(如圖38所示)、開度對應功率(如圖39所示)、對應推力(如圖40所示)與對應力效(如圖41所示)。

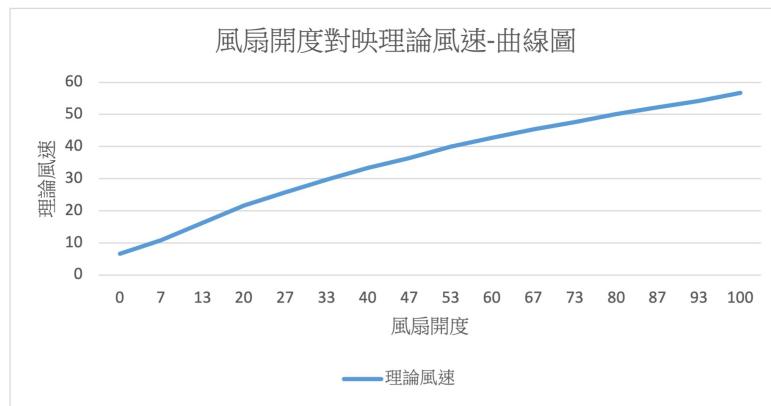


圖 37: 風扇開度對應理論風速曲線圖

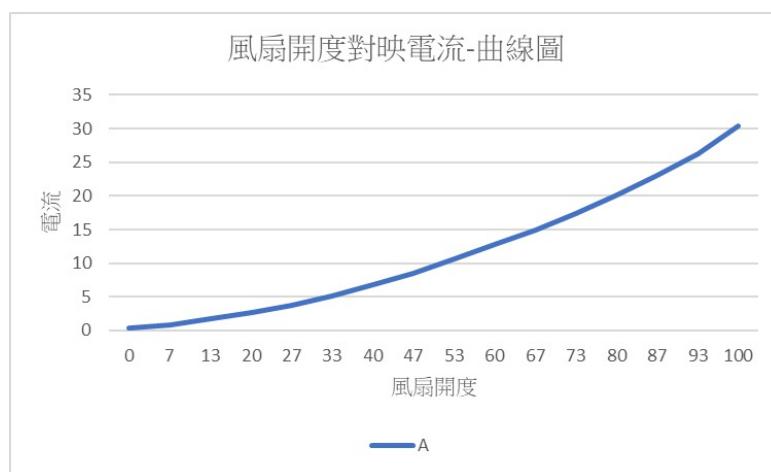


圖 38: 風扇開度對應電流曲線圖

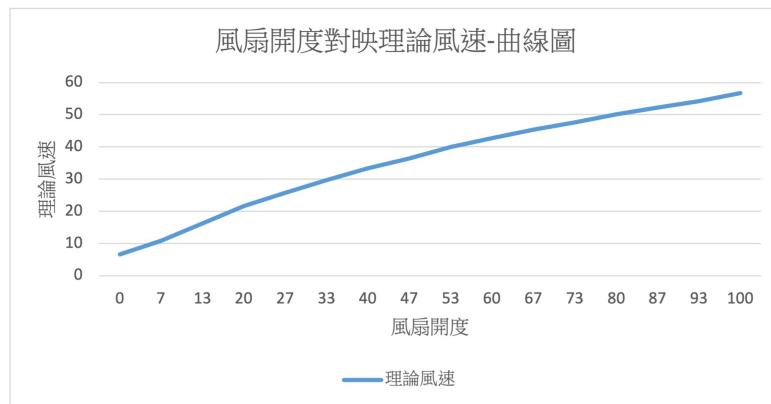


圖 39: 風扇開度對應功率曲線圖

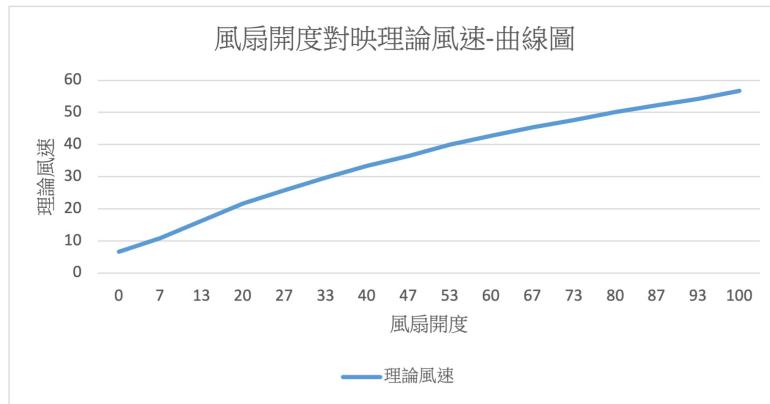


圖 40: 風扇開度對應推力曲線圖

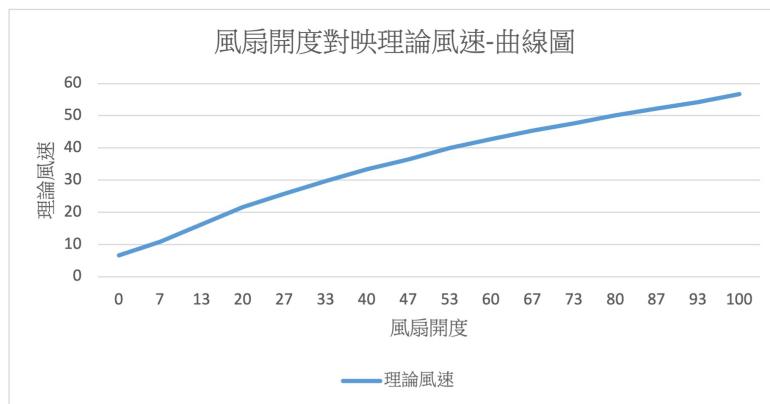


圖 41: 風扇開度對應力效曲線圖

6 結論與反思

7 參考文獻

- [1] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, “Estimation of number of people in crowded scenes using perspective transformation,” *IEEE*, vol. 31, no. 6, 2001.
- [2] M. Bertozzi, A. Broggi, and A. Fascioli, “Stereo inverse perspective mapping: Theory and applications,” *Image and vision computing*, vol. 16, no. 8, pp. 585–590, 1998.

- [3] H. A. Mallot, H. H. Bülow, J. J. Little, and S. Bohrer, “Inverse perspective mapping simplifies optical flow computation and obstacle detection,” *Biological cybernetics*, vol. 64, no. 3, pp. 177–185, 1991.

8 附錄

8.1 程式碼

影像偵測演算法以 Python 撰寫，報告則使用 LaTeX 系統進行編輯與排版；所有原始程式碼均公開於王子晨的 GitHub 儲存庫，以利後續參考與驗證。網址如下：

https://github.com/PrinceLego/Practice_Of_Mechanical_Engineering.git

8.2 物料清單與發票收據

BOM					
名稱	數量	單價	使用的材料	來源	價格
Raspberry pi 3B	1	1980			1980
Arduino Mage	1	725			725
Lipo 3S	1	325			325
40A ESC	1	400			400
50mm EDF	1	800			800
SG90	2	50			100
MG995	1	150			150
MPU6050	1	90			90
OV5647	1	280			280
循跡傳感器	1	300			300
電壓傳感器	1	20			20
電流傳感器	1	60			60
降壓板	3	30			90
測速感知器	2	30			60
50A斷路器	1	400			400
保險絲盤	1	340			340
保險絲5A	5	3			15
保險絲10A	5	3			15
保險絲15A	5	3			15
保險絲20A	5	3			15
保險絲30A	5	3			15
保險絲40A	5	3			15
電線 - 12# - 黑	3	48			144
電線 - 12# - 紅	3	48			144
電線 - 18# - 黑	3	32			96
電線 - 18# - 紅	3	32			96
ANS保險絲座	2	80			160
ANS - 50A	1	30			30
ANS - 60A	1	30			30
ANS - 80A	1	30			30
LED - Green	2	10			20
LED - Yellow	2	10			20
LED - Red	2	10			20
緊急開關	1	40			40
啟動開關	1	40			40
杜邦線	40	2			80
					7160

圖 42: bom 表