

國立臺灣師範大學科技與工程學院電機工程學系

碩士論文

Department of Electrical Engineering  
College of Technology and Engineering  
National Taiwan Normal University  
Master's Thesis

四旋翼無人機結合視覺辨識障礙物與自主路徑導航  
Vision-Based Obstacle Detection and Autonomous Path  
Navigation for Quadrotor UAVs



柯紀為

Ji-Wei Ke

指導教授：呂藝光 博士

Advisor: Yih-Guang Leu, Ph.D.

中華民國 114 年 7 月

July 2025

## 誌 謝

時光荏苒，碩士兩年的旅程即將劃下句點，回首這段充實的時光，內心充滿感激。在此，我想誠摯地感謝一路上陪伴與支持我的每一位師長、同儕與家人，因為有你們，讓我能無畏地面對挑戰，逐步走向目標。

首先，衷心感謝我的指導教授呂藝光教授。從研究初期的方向引導，到每一次的悉心指正，呂老師總是以開放而嚴謹的態度，激發我對學術研究的熱情。除了學術上的啟發，老師對團隊合作、問題解決及未來職涯的指引，更讓我受益良多。在老師的指導下，我不僅完成了屬於自己的研究成果，也學會了如何面對未知、解決難題，這將成為我人生道路上寶貴的資產。

感謝實驗室的學長們高碩、佑駿、軒博，總在我面臨困難時耐心地給予的建議，幫助我更深入理解研究背後的技術與邏輯。也感謝我在實驗室的夥伴們哲瑋、昱賢、彥澂、鼎鈞、翊翔、凱鴻、孟緯、逸旻，在一起學習與討論的過程中，我們互相成長、彼此鼓勵，無論是課業、研究還是生活上的陪伴，都是我碩士生活中最珍貴的回憶。

此外，也感謝每一位參與討論與合作的同學，不同的觀點與思維方式，拓展了我的眼界，也讓研究之路更加豐富多元。

最後，我要感謝我的家人。感謝你們長期以來無條件的支持與鼓勵，是你們成為我最堅強的後盾，讓我得以全心投入學習與研究。在我迷惘或低潮時，家人的理解與陪伴，讓我能夠重新找回方向、持續前行。

紀為 謹誌於

中華民國一〇四年七月

# 四旋翼無人機結合視覺辨識障礙物與自主路徑導航

學生：柯紀為

指導教授：呂藝光

國立臺灣師範大學電機工程學系碩士班

## 摘 要

本研究實現一套具備即時避障與自主導航能力之四旋翼無人機系統，整合視覺辨識、空間分析與路徑規劃技術，以提升無人機在複雜環境中的適應性。

首先，在障礙物辨識方面，應用了 Farneback 光流法，並結合光流補償提升移動物體偵測準確度。隨後，採用 DBSCAN 類聚法與區塊劃分方式，分析影像中障礙區域的分佈與面積，進一步轉換為 3D 空間中的實體位置資訊。在路徑規劃部分，使用 BFS 廣度優先搜尋演算法進行障礙區域探索，並結合反投影法將 2D 影像座標轉換至 3D 空間中，提供動態避障資訊。針對整體導航策略，採用快速探索隨機樹演算法 (RRT\*) 產生初始路徑，再依序跟隨生成的航點進行飛行引導。在導航過程中，利用機載鏡頭即時取得影像，建立光流場以進行障礙物偵測，並即時分析四旋翼飛行器與障礙物之間的相對關係，實現動態調整與安全避障。

實驗結果驗證，本研究所提出之系統可穩定進行視覺辨識、動態避障與空間導航，具有良好之實用性。

關鍵字：四旋翼無人機系統、動態避障、廣度優先搜尋演算法、導航。

# Vision-Based Obstacle Detection and Autonomous Path Navigation for Quadrotor UAVs

Student: Ji-Wei Ke

Advisor: Yih-Guang Leu

Department of Electrical Engineering  
National Taiwan Normal University

## ABSTRACT

This study presents the implementation of a quadrotor unmanned aerial vehicle (UAV) system capable of real-time obstacle avoidance and autonomous navigation. The system integrates visual perception, spatial analysis, and path planning techniques to enhance UAV adaptability in complex environments.

For obstacle detection, the Farnebäck optical flow method with flow compensation is applied to enhance moving object detection. DBSCAN clustering and grid-based segmentation are used to analyze obstacle areas, which are then mapped into 3D space. A breadth-first search (BFS) algorithm explores obstacle regions, and a back-projection method converts 2D image coordinates into 3D positions for dynamic obstacle avoidance. The Rapidly-exploring Random Tree Star (RRT\*) algorithm generates the initial navigation path. During the navigation process, real-time images are captured by the onboard camera to construct an optical flow field for obstacle detection. The relative relationship between the quadrotor UAV and obstacles is analyzed in real time, enabling dynamic adjustment and safe obstacle avoidance.

Experimental results show that the system achieves stable visual recognition, obstacle avoidance, and navigation performance, demonstrating strong potential for practical UAV applications.

**Keywords:** Unmanned aerial vehicle system, dynamic obstacle avoidance, Breadth-First Search, navigation.

# 目 錄

誌 謝.....	i
目 錄.....	iv
表 目 錄.....	vi
圖 目 錄.....	vii
第一章 緒論.....	1
1.1 研究動機與背景.....	1
1.2 研究目的.....	2
1.3 研究方法.....	2
1.4 論文架構.....	3
第二章 文獻探討與回顧.....	4
2.1 無人機避障之演進.....	4
2.2 基於視覺的障礙物辨識.....	4
2.3 路徑規劃.....	5
第三章 障礙物偵測與路徑規劃方法.....	7
3.1 障礙物辨識.....	7
3.1.1 Farneback 光流法.....	8
3.1.2 Image Pyramid 影像金字塔.....	11
3.1.3 DBSCAN 類聚法.....	14
3.1.4 動態障礙物與光流補償.....	16
3.1.5 劃分與障礙面積計算.....	19
3.2 動態路徑規劃.....	20
3.2.1 BFS 廣度優先搜尋演算法.....	20
3.2.2 Deprojection 反投影法.....	23
3.3 靜態路徑規劃.....	27

3.3.1 Rapidly-exploring Random Tree 演算法 .....	27
3.3.2 Rapidly-exploring Random Tree Star 演算法 .....	30
第四章 四旋翼無人機自主導航系統 .....	33
4.1 自主導航系統之架構 .....	34
4.2 硬體架構 .....	34
4.2.1 Realsense 鏡頭模型 .....	36
4.2.2 NVIDIA Jetson Nano 開發套件 .....	38
4.2.3 GPS 元件 .....	38
4.3 軟體架構 .....	39
4.3.1 障礙物辨識規劃 .....	39
4.3.2 動態路線規劃 .....	41
4.3.3 靜態路線規劃 .....	42
4.3.4 四旋翼無人機控制系統 .....	43
第五章 實驗設計與結果 .....	45
5.1 障礙物辨識實驗 .....	45
5.1.1 靜態障礙物辨識實驗 .....	45
5.1.2 動態障礙物辨識實驗 .....	52
5.2 無人機避障實驗 .....	59
5.2.1 靜態障礙物避障實驗 .....	61
5.2.2 動態障礙物避障實驗 .....	67
5.3 自主導航實驗 .....	74
第六章 結論與未來展望 .....	85
6.1 結論 .....	85
6.2 未來展望 .....	85
參考文獻 .....	86

## 表 目 錄

表 4-1 硬體元件模組表.....	37
表 5-1 靜態障礙物辨識實驗結果.....	47
表 5-2 靜態障礙物辨識實驗數據.....	51
表 5-3 靜態障礙物辨識實驗結果.....	54
表 5-4 大型動態障礙物辨識實驗數據.....	58
表 5-5 小型動態障礙物辨識實驗數據.....	59
表 5-6 靜態障礙物避障實驗結果.....	62
表 5-7 靜態避障實驗數據.....	66
表 5-8 動態障礙物避障實驗結果.....	68
表 5-9 大型動態避障實驗數據.....	73
表 5-10 小型動態避障實驗數據.....	74



## 圖 目 錄

圖 3-1 小區塊範圍的障礙物輪廓.....	10
圖 3-2 大區塊範圍的障礙物輪廓.....	10
圖 3-3 影像金字塔技術結合 Farneback 光流法示意圖(作者繪製).....	11
圖 3-4 實際觀測環境.....	13
圖 3-5 未套用影像金字塔的障礙物輪廓.....	13
圖 3-6 套用影像金字塔的障礙物輪廓.....	13
圖 3-7 未做前景分割的完整輪廓.....	15
圖 3-8 前景分割後的前景輪廓.....	16
圖 3-9 前景分割後的背景輪廓.....	16
圖 3-10 未做光流補償的動態物體輪廓.....	18
圖 3-11 做光流補償的動態物體輪廓.....	18
圖 3-12 滿足遮蔽閾值的網格區塊.....	19
圖 3-13 實際觀測環境.....	21
圖 3-14 障礙輪廓和 BFS 推導過程(作者繪製).....	22
圖 3-15 紅色相鄰的空曠面積.....	22
圖 3-16 有兩處空曠區域.....	23
圖 3-17 找到最大面積和移動方向.....	23
圖 3-18 針孔攝像頭模型(作者繪製).....	24
圖 3-19 透過 BFS 演算法獲得影像平面的移動路徑.....	25
圖 3-20 透過反投影法將障礙物和路徑轉換成三維空間.....	25
圖 3-21 前視圖.....	26
圖 3-22 右視圖.....	26
圖 3-23 上視圖.....	26
圖 3-24 RRT 演算法流程(作者繪製).....	28

圖 3-25 RRT 演算法結果(作者繪製).....	28
圖 3-26 Google Maps 提供的三維建築物.....	29
圖 3-27 三維建築資料套用 RRT 演算法.....	29
圖 3-28 RRT*演算法流程(作者繪製).....	31
圖 3-29 套用 RRT 演算法的路徑輸出.....	31
圖 3-30 套用 RRT*演算法的路徑輸出.....	32
圖 4-1 四旋翼無人機(作者拍攝).....	33
圖 4-2 整體系統架構.....	34
圖 4-3 硬體架構.....	35
圖 4-4 Intel RealSense D435i(作者拍攝).....	37
圖 4-5 深度精度.....	37
圖 4-6 NVIDIA Jetson Nano(作者拍攝).....	38
圖 4-7 Neo-M8N GPS 元件(作者拍攝).....	39
圖 4-8 障礙物辨識流程.....	40
圖 4-9 動態路線規劃流程.....	42
圖 4-10 靜態路線規劃流程.....	43
圖 4-11 四旋翼無人機控制流程.....	44
圖 5-1 靜態障礙物辨識實驗流程.....	46
圖 5-2 實驗示意圖.....	46
圖 5-3 障礙物 X 軸變化圖.....	49
圖 5-4 障礙物 Y 軸變化圖.....	50
圖 5-5 障礙物 Z 軸變化圖.....	50
圖 5-6 動態障礙物辨識實驗流程.....	53
圖 5-7 實驗示意圖.....	53
圖 5-8 障礙物 X 軸變化圖.....	56
圖 5-9 障礙物 Y 軸變化圖.....	56

圖 5-10 障礙物 Z 軸變化圖.....	57
圖 5-11 避障實驗流程圖.....	60
圖 5-12 靜態障礙物避障實驗示意圖.....	61
圖 5-13 障礙物 X 軸變化圖.....	64
圖 5-14 障礙物 Y 軸變化圖.....	64
圖 5-15 障礙物 Z 軸變化圖.....	65
圖 5-16 動態障礙物避障實驗示意圖.....	67
圖 5-17 障礙物 X 軸變化圖.....	71
圖 5-18 障礙物 Y 軸變化圖.....	71
圖 5-19 障礙物 Z 軸變化圖.....	72
圖 5-20 自主導航實驗流程圖.....	75
圖 5-21 建築物輪廓圖.....	76
圖 5-22 靜態路徑規劃圖.....	76
圖 5-23 無人機實際路線圖.....	77
圖 5-24 無人機位置與導航路線間的距離變化圖.....	77
圖 5-25 發生避障事件之位置.....	78
圖 5-26 飛行階段.....	79
圖 5-27 障礙物初現階段.....	79
圖 5-28 避障決策階段.....	79
圖 5-29 避障完成階段.....	79
圖 5-30 障礙物 X 軸變化圖.....	80
圖 5-31 障礙物 Y 軸變化圖.....	80
圖 5-32 障礙物 Z 軸變化圖.....	81
圖 5-33 發生避障事件之位置.....	81
圖 5-34 飛行階段.....	82
圖 5-35 障礙物初現階段.....	82

圖 5-36 避障決策階段.....	82
圖 5-37 避障完成階段.....	82
圖 5-38 障礙物 X 軸變化圖.....	83
圖 5-39 障礙物 Y 軸變化圖.....	83
圖 5-40 障礙物 Z 軸變化圖.....	84



# 第一章 緒論

## 1.1 研究動機與背景

隨著無人機 (Unmanned Aerial Vehicles, UAVs) 技術的快速發展, 其應用範疇已從傳統的軍事偵察與地形建模, 逐步擴展至商業運輸、航空影像與日常生活中的應用。受益於通訊、控制與計算能力的提升, 以及硬體成本的下降, 無人機正迅速成為時代中具代表性的自主移動平台。然而, 無人機在低空複雜環境中自主導航的能力仍面臨諸多挑戰, 特別是在缺乏高精度傳感器的條件下, 如何有效地感知環境並執行即時避障, 是一項極需解決的問題。

在各式無人機中, 四旋翼飛行器因具備垂直起降與懸停能力, 在動態環境中具備高度靈活性, 成為自主飛行研究的熱門平台。然而, 其受限於載重與續航力, 往往無法搭載大型感測器。因此, 如何利用輕量化的機載設備, 來實現導引與避障, 成為現今研究的重要方向。

過去的文獻中, 視覺型避障方法主要分為單眼視覺[1]、立體視覺[2]與光流法[3]三大類。單眼視覺雖具備低硬體要求, 但深度資訊不足; 立體視覺能提供高精度的深度, 其系統複雜度與運算量較高; 而光流法能夠透過連續影像估計場景變化, 兼顧效能與運算效率, 成為低成本即時避障的可行方案。特別是在光流法中, 透過光流補償與區域分析等技術, 可進一步提升障礙物偵測的準確性與穩定性。

另一方面, 在路徑規劃與導航策略方面, 雖已有大量研究探討路線導航與即時避障的整合應用, 然而, 如何結合預先規劃的路徑與飛行中的環境資訊, 並適應動態障礙物, 仍是一項挑戰。

基於上述背景與需求，本研究致力於開發一套結合靜態路徑規劃與動態視覺避障的四旋翼自主導航系統，實現在複雜環境下的穩定飛行與即時路徑修正，以提升無人機在實際應用中的可靠性。

## 1.2 研究目的

本研究旨在開發一套具備即時避障與自主導航能力的四旋翼無人機系統，整合光流分析、深度感知與路徑規劃技術，提升無人機於複雜環境中的飛行安全與導航效率。透過視覺感測與動態路徑調整，實現高效能的自主飛行控制。

## 1.3 研究方法

本研究將結合靜態與動態的處理方式進行自主導航和即時避障，以實現四旋翼無人機於未知環境中的自主導航能力。首先，在導航初始階段，透過快速探索隨機樹（Rapidly-exploring Random Tree Star, RRT\*）演算法[4]，針對任務區域預先建立一條初始導航路徑，並產生一系列導航點以引導飛行。

於實際飛行過程中，系統透過機載鏡頭持續擷取連續影像，並應用Farneback 光流法[5]計算場景中的光流場，用以偵測移動中的潛在障礙物。此外，透過光流補償的技術，降低移動時所造成的干擾，提升障礙物偵測的精度。針對偵測結果，利用 DBSCAN 類聚法[6]與區塊劃分方式分析影像中障礙區域的分佈與面積，並結合深度資訊將其投影至三維空間中，取得實體障礙物位置。

動態路徑的更新與避障控制，系統整合廣度優先搜尋（Breadth-First Search, BFS）演算法，快速搜尋可通行區域，搭配飛行路徑與障礙物的相對位置動態調整導航路線。

達成即時感知、分析與導航控制功能，實現具有環境適應性與路徑靈活調整能力的四旋翼無人機導航系統。

## 1.4 論文架構

本研究共分為六個章節，各章內容分述如下：

第一章 緒論：詳述本研究背景及動機、研究目的、研究方法。

第二章 文獻探討：探索過去無人機避障技術的發展脈絡，整理障礙物辨識的方法與常見路徑規劃技術，作為本研究方法設計的理論基礎。

第三章 障礙物偵測與路徑規劃方法：介紹本研究所採用的障礙物偵測技術和路徑規劃方法。

第四章 四旋翼無人機自主導航系統：說明整體系統架構，分別介紹所使用的硬體設備與軟體設計，包括靜態與動態路線規劃系統以及無人機控制機制。

第五章 實驗設計與結果：透過障礙物辨識、無人機避障、自主導航三個實驗，來分析系統實用性。

第六章 結論與未來展望：對於本研究實驗結果總結，並提出未來研究的改善方向。

## 第二章 文獻探討與回顧

本章節主要回顧無人機避障技術的發展歷程，並探討目前主流的障礙物辨識方法與路徑規劃演算法。藉由相關研究的整理與比較，奠定研究理論的基礎。

### 2.1 無人機避障之演進

隨著無人機應用的普及，如何在複雜環境中進行自主導航與避障，已成為智慧型飛行器研究的重要議題。早期的避障方法多依賴感測器進行障礙偵測與環境建圖[7]。然而，這些感測元件大多體積龐大、成本高昂，不適用於輕量型四旋翼飛行器。因此，研究重心逐漸轉向以低成本為主的視覺導向避障系統。透過機載鏡頭取得的影像資訊，進行場景解析與路徑規劃，能大幅降低硬體成本並提升系統整合度[8][9]。

### 2.2 基於視覺的障礙物辨識

在無人機自主導航系統中，視覺感知技術是實現環境理解與障礙物辨識的核心。隨著深度學習與感測器的進步，視覺障礙辨識逐漸從傳統影像處理方法，發展至結合深度資訊與時序動態分析的技術架構。目前常見的辨識策略可大致分為單目影像分析、立體視覺，以及光流法三類，各有其適用條件與技術限制。

單目影像分析由於其硬體架構簡單、重量輕巧，特別受到小型無人機平台的青睞。這類方法透過卷積神經網路（CNN）對影像內容進行端到端學習[10]，預測像素對應的深度或障礙物區域。然而，單目系統常面臨尺度不確定性問題，需透過環境假設、語意補償或其他感測器輔助提升其可靠性。

相較之下，立體視覺系統透過雙鏡頭取得視差圖，可直接推估場景中物體的相對深度。生成深度圖後，便可重建雲點圖，並藉由空間分割辨識障礙物區域[11]。這類系統可提供較高精度的三維資訊，但也需進行鏡頭標定與同步處理，且在強光或低紋理環境中容易出現視差。

另一種適合動態場景的方法為光流法。光流分析利用連續影像之間的像素位移計算相對運動向量，進而推測障礙物的存在與移動方向。Farneback 光流法能提供完整的場景運動資訊，更適合應用於空間中障礙物的偵測[12]。

## 2.3 路徑規劃

完成障礙物辨識後，無人機須具備有效的路徑規劃能力，才能確保其避開危險區域並抵達目標。路徑規劃系統可區分為靜態規劃與動態避障。靜態路徑階段主要產生一條大致可行的飛行路徑，而動態避障則需根據實際飛行過程中的感測回饋，即時調整行進方向。

靜態路徑規劃以取樣的探索法為主，其中快速探索隨機樹（RRT）是一種能在空間中迅速建立路徑的演算法，藉由隨機點擴展樹狀結構，不需網格化處理即可適應地形[13]。然而，傳統 RRT 所生成的路徑通常不夠平滑，因此改良版本 RRT\* 應運而生，透過不斷重新連接並優化節點間的距離，以提升路徑品質與可行性[14]。

針對動態避障需求，廣度優先搜尋（BFS）演算法則提供一種簡潔且有效率的格狀空間搜尋方式，能快速找出通行方向與障礙區域分布，在即時影像中快速決策[15]。此外，反投影法（Back Projection）也在避障中扮演關鍵角色，它可將影像座標下的避障資訊投影回三維空間，進而整合路徑更新與方向修正[16]。



## 第三章 障礙物偵測與路徑規劃方法

本章節介紹該研究中所採用的障礙物偵測與路徑規劃技術。首先，將說明視覺式障礙物偵測的原理與實作流程，包含光流法在環境中的應用與光流補償的細節等。接著，介紹如何透過類聚與區塊分析方式，有效判斷潛在障礙物區域及其空間分布特性。針對避障與路線選擇部分，將深入探討 BFS 廣度優先搜尋在即時導航中的角色，並結合反投影技術將影像資訊轉換至真實三維空間，實現空間感知的功能。

最後，本章將介紹靜態路徑規劃中所採用之快速探索隨機樹 (RRT) 及其改良版本 RRT\* 演算法，說明其在複雜空間中建立可行路徑的流程與優勢。本章內容將為後續系統整合與實驗驗證奠定技術基礎。

### 3.1 障礙物辨識

本研究首先透過光流法辨識影像中的運動資訊，利用 Farneback 光流法以產生密集光流場。為了排除無人機自身運動造成的干擾，加入光流補償技術，去除全域背景運動，突顯障礙物的相對位移。接著，結合類聚法辨識出可能的障礙物區塊，並估算其位置與面積，作為後續避障與路徑規劃的依據。

### 3.1.1 Farneback 光流法

Farneback 光流法是一種密集型光流估計方法 (Dense Optical Flow) [17]，其核心概念是對影像中一個局部區域做二次多項式擬合，得出目標像素亮度，再藉由前後影像間的變化來推估該像素的運動向量。

Farneback 光流法引入局部圖像的方法，計算每個像素點的運動向量，適用於大規模場景中的運動估計。圖像中的亮度可視為一個二維函數  $f(x, y)$ ，以目標像素為中心，構建一個局部座標系。對此亮度函數進行二次多項式展開後，可近似表示為：

$$f(x, y) \approx r_1 + r_2x + r_3y + r_4x^2 + r_5y^2 + r_6xy \quad (3-1)$$

整理可得：

$$f(x, y) = \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} r_4 & r_6/2 \\ r_6/2 & r_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} r_2 \\ r_3 \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix} + r_1 \quad (3-2)$$

最後表示為：

$$f(x) = x^T Ax + b^T x + c \quad (3-3)$$

其中， $x = \begin{bmatrix} x \\ y \end{bmatrix}$  為二維座標向量， $A = \begin{bmatrix} r_4 & r_6/2 \\ r_6/2 & r_5 \end{bmatrix}$  為  $2 \times 2$  的對稱矩陣， $b = \begin{bmatrix} r_2 \\ r_3 \end{bmatrix}$  為  $2 \times 1$  的矩陣。此處的係數只針對在目標像素點  $(x, y)$ ，對於其他點並不適用。將式(3-1)重寫為：

$$f(x, y) = P(x, y)^T r \quad (3-4)$$

其中， $P(x, y) = [1 \ x \ y \ x^2 \ y^2 \ xy]^T$  為六維基底向量，而  $r = [r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6]^T$  為對應的六維係數向量。然而在實際運算中，起初並不知道係數向量  $r$  的具體數值，因此需要根據區域內實際像素的亮度值  $I(x, y)$  來推算。

將區域內所有像素點建立一組聯立方程式，每個像素對應一個線性方程，整合後可得：

$$Mr = g \quad (3-5)$$

其中

$$M = \begin{bmatrix} P(x_1, y_1)^T \\ P(x_2, y_2)^T \\ \vdots \\ P(x_n, y_n)^T \end{bmatrix} \quad g = \begin{bmatrix} I(x_1, y_1) \\ I(x_2, y_2) \\ \vdots \\ I(x_n, y_n) \end{bmatrix} \quad (3-6)$$

$M$ 是由各點的基底向量  $P(x, y)^T$  所組成的矩陣， $g$  是由各點的實際亮度  $I(x, y)$  組成的矩陣。為了加強擬合的穩定性與中心像素的影響力，套用加權最小平方法，帶入權重矩陣  $W$  求解出係數向量  $r$ ：

$$r = (M^T W M)^{-1} M^T W g \quad (3-7)$$

接著對兩張連續影像中的同一區域，分別擬合出其亮度函數：

第一張影像亮度函數為：

$$f_1(x) = x^T A_1 x + b_1^T x + c_1 \quad (3-8)$$

第二張影像亮度函數為：

$$f_2(x) = x^T A_2 x + b_2^T x + c_2 \quad (3-9)$$

影像中的區域發生位移  $d$  後，表示為：

$$f_2(x) = f_1(x + d) \quad (3-10)$$

將式(3-10)展開後可得：

$$f_1(x + d) = (x + d)^T A_1 (x + d) + b_1^T (x + d) + c_1 \quad (3-11)$$

整理後可得：

$$f_1(x + d) = x^T A_1 x + (b_1 + 2A_1 d)^T x + (d^T A_1 d + b_1^T d + c_1) \quad (3-12)$$

帶入式(3-10)和式(3-9)比較係數可得：

$$A_2 = A_1 \quad (3-13)$$

$$b_2 = b_1 + 2A_1 d \quad (3-14)$$

$$c_2 = d^T A_1 d + b_1^T d + c_1 \quad (3-15)$$

式(3-13)中，矩陣  $A$  代表局部亮度的二次變化特性。在僅有平移發生的情況下，影像區域內的亮度結構視為穩定，故近似前後幀的矩陣  $A$ ，使  $A_2 = A_1$ 。由式(3-14)可求得光流向量  $d$  為：

$$d = \frac{1}{2}A_1^{-1}(b_2 - b_1) \quad (3-16)$$

透過這種方式，我們可以從亮度函數中的係數變化推估出像素的運動向量。將上述方法套用至整張影像中的每一個像素位置，並彙整所有對應的運動向量  $d = [u, v]^T$  集合，即可構成一張完整密集光流場。

在 Farneback 光流法中，局部區域的大小選擇對光流估計結果具有顯著影響。如圖 3-1，較小的取樣區域能更精細地擬合局部亮度變化，對於小尺度的變化具有更高的解析能力。然而，由於參與擬合的像素數較少，容易受到雜訊影響，導致光流場出現不穩定或抖動。

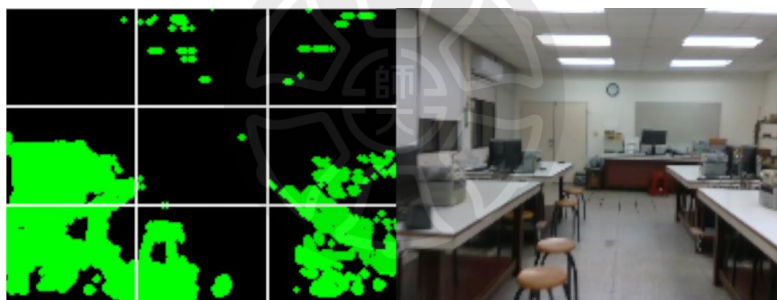


圖 3-1 小區塊範圍的障礙物輪廓

相反地，如圖 3-2 所示。較大的區域範圍可涵蓋更多像素，提高擬合的穩定性，進而產生更平滑的光流場。然而，大範圍擬合也會導致局部細節的運動被過度平滑化，難以準確捕捉物體邊界或小區域內的微小變化。

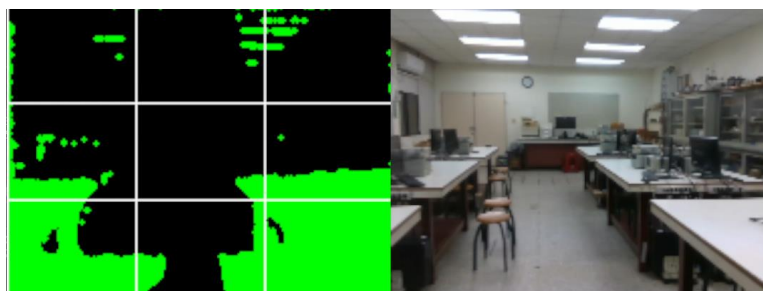


圖 3-2 大區塊範圍的障礙物輪廓

### 3.1.2 Image Pyramid 影像金字塔

上述問題說明了：固定區域大小難以同時應對大範圍位移與細部變化的需求。為此，本研究結合影像金字塔（Image Pyramid）技術，以達到從粗略估計到精細修正的多階段重建策略。

Farneback 光流法基於區域二次多項式擬合進行光流估計，因此其取樣區域大小必須在捕捉大位移與維持精細邊緣之間取得平衡。然而，若僅使用單一解析度進行估計，對於快速移動的物體容易出現錯估或模糊，進而影響整體光流場品質。

如圖 3-3 所示，影像金字塔的核心概念是將原始影像按比例逐層縮小，產生一組從低解析度到高解析度的影像集合。在流程中，首先選定一對相鄰時間點的影像  $image(t)$  與  $image(t + dt)$ ，並分別構建其金字塔結構 [25]。

以第  $L$  層為解析度最低的層為起點，Farneback 光流法首先在這層進行光流估計。所得到的光流向量  $u^L$ ，為初步的粗略估計，將用於後續高解析度層的補償與微調。在低解析度下，原始的大範圍位移將轉化為較小的像素位移，這使得固定大小的取樣區域能更有效地捕捉其整體運動趨勢，提升穩定性。

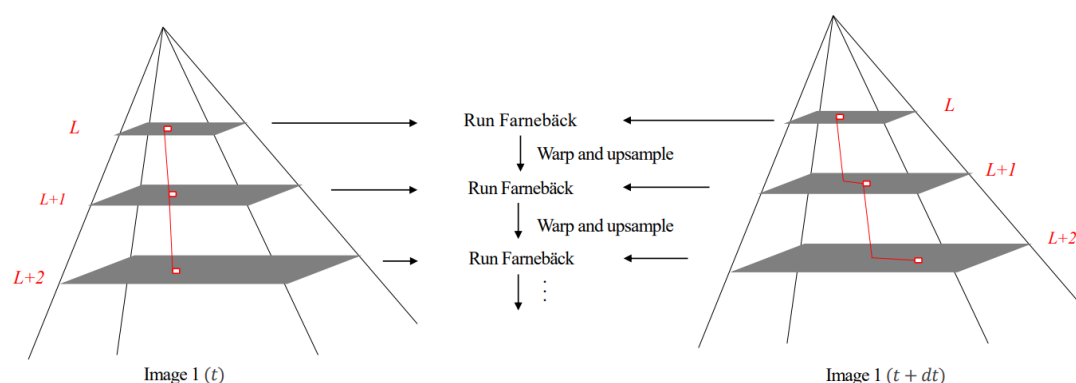


圖 3-3 影像金字塔技術結合 Farneback 光流法示意圖(作者繪製)[25]

接下來將這組粗略的光流向量推進至較高解析度的第  $L + 1$  層，先進行上採樣（upsample）操作。透過雙線性插值，將光流場對應插值至新解析度，表示如下：

$$\tilde{u}^{L+1} = \text{Upsample}(u^L) \quad (3-17)$$

其中 *Upsample* 表示內插函數，用於將第  $L$  層估計的光流轉換至第  $L + 1$  層的像素上，使其可直接應用於新解析度。

接著進行的是扭曲（warp）操作，將第  $L + 1$  層的  $\text{image}_{t+dt}^{L+1}$  影像依據目前的光流進行像素對齊，形成補償影像  $\hat{\text{image}}_{t+dt}^{L+1}$ ：

$$\hat{\text{image}}_{t+dt}^{L+1} = \text{image}_{t+dt}^{L+1}(x - \tilde{u}^{L+1}) \quad (3-18)$$

$x$  為第  $L + 1$  層像素座標，此操作將  $t + dt$  時刻的像素依光流向量平移至預期位置，使其更貼近第  $t$  幀的對應內容，讓後續的估計聚焦於補償後的殘差部分。

在完成補償後，將  $\hat{\text{image}}_{t+dt}^{L+1}$  與原始  $\text{image}_t^{L+1}$  進行新一輪光流估計，獲得殘差光流向量  $\Delta u^{L+1}$ ，並將其與上採樣結果  $\tilde{u}^{L+1}$  合併，即得第  $L + 1$  層的光流：

$$u^{L+1} = \tilde{u}^{L+1} + \Delta u^{L+1} \quad (3-19)$$

這一流程會逐層向下推進，直至原始解析度為止。在每一層中，光流場都會在初步補償後進行局部細節估計與誤差修正，這種多層次的估計流程，使 Farneback 光流法在面對複雜運動場景時，能兼顧整體穩定性與局部解析力。

圖 3-4 為實際觀測的環境，從圖 3-5 中可以發現未使用影像金字塔時，光流向量在細節處容易受到雜訊干擾，導致整體光流場雜亂，產生破碎的雜訊。而圖 3-6 套用了影像金字塔後，系統會先在低解析度下進行光流估計，濾除了雜訊，提供一個平滑且更接近整體運動趨勢的初步估計。接著，

向上層細化時再逐層修正細節，使得整體光流場具備更高的穩定性與一致性。



圖 3-4 實際觀測環境

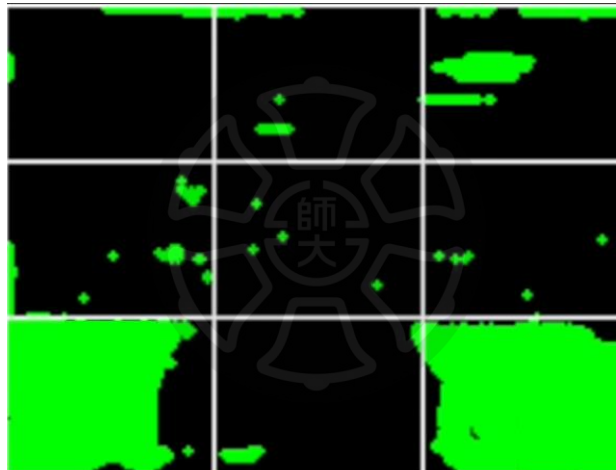


圖 3-5 未套用影像金字塔的障礙物輪廓

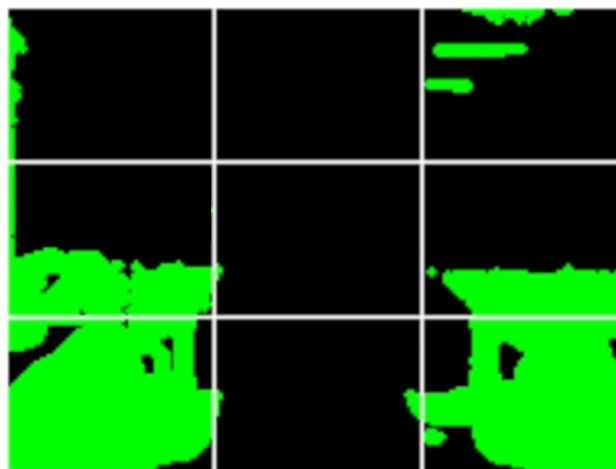


圖 3-6 套用影像金字塔的障礙物輪廓

### 3.1.3 DBSCAN 類聚法

在處理前景分離的任務中，如何從大量的像素運動資訊中擷取結構化的區域，是關鍵。為了從密集的光流場中有效區分前景與背景，我們採用 DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 類聚法[18]，結合像素運動行為進行區域類聚，進而強化前景輪廓與遮蔽區塊的可辨識性。本節將詳細介紹 DBSCAN 的理論基礎，並延伸至光流場的應用調整與實際效果。

DBSCAN 類聚法主要依賴兩個參數，鄰域半徑  $\varepsilon$ ，為任一資料點的搜尋範圍；最小點數的門檻  $MinPts$ ，定義形成「核心點」所需的鄰近點數。對任一點  $P$ ，若其鄰域內包含至少  $MinPts$  個點，則  $P$  為核心點。若某點位於某核心點的鄰域內，則為邊界點；否，則為雜訊點。若從某一核心點可透過鄰域遞延到另一點，則二者屬同一群集。

DBSCAN 類聚法最大的優點是：能夠辨識任意形狀的群集，且對雜訊有良好的排斥能力。然而，其距離度量預設僅考慮空間座標的鄰近性，無法應用於具有「向量屬性」的資料，如光流場中的像素運動行為。

在光流場中，每一像素點不僅具有空間位置  $(x, y)$ ，同時對應一個運動向量  $(u, v)$ 。若僅基於像素的位置進行類聚，無法將具有相似運動模式的像素聚為同群，尤其在靜態背景與前景移動物體的分界處，會造成邊界的模糊與輪廓的缺失。

因此，我們將 DBSCAN 類聚法輸入特徵向量擴展為：

$$P_i = (x_i, y_i, u_i, v_i) \quad (3-20)$$

並重新定義空間中的距離度量函數為：

$$D(P_i, P_j) = \alpha \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \beta \sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} \quad (3-21)$$

其中  $(x_i, x_j)$  為像素空間座標； $(u_i, v_j)$  該像素的光流向量； $\alpha$  控制空間鄰近性的重要程度； $\beta$  控制運動向量相似度的重要程度。

透過此「空間-運動混合的距離」，DBSCAN 類聚法可類聚出「位置相近且運動相似」的像素區域，進而有效辨識出畫面中具有運動趨勢的物體。

在實作上， $\alpha$  與  $\beta$  的選擇影響了類聚的結果。若  $\alpha \gg \beta$ ，偏向空間位置主導；若  $\alpha \ll \beta$ ，則對運動敏感但可能產生碎裂分群。經實驗調整後，本研究設定  $\alpha = 0.1$ 、 $\beta = 0.9$ ，使類聚對運動趨勢更為敏感。

本研究所採用之 DBSCAN 類聚法，透過整合像素之空間座標與光流向量，建立具備空間-運動混合的特徵資料集。然而，類聚過程本身並不具備前景與背景的分類能力。在無額外處理的情況下，光流場中有運動行為的區域皆有可能被類聚為有效的群集並繪製出輪廓，如圖 3-7，導致前景與背景輪廓在同一畫面中共同呈現。

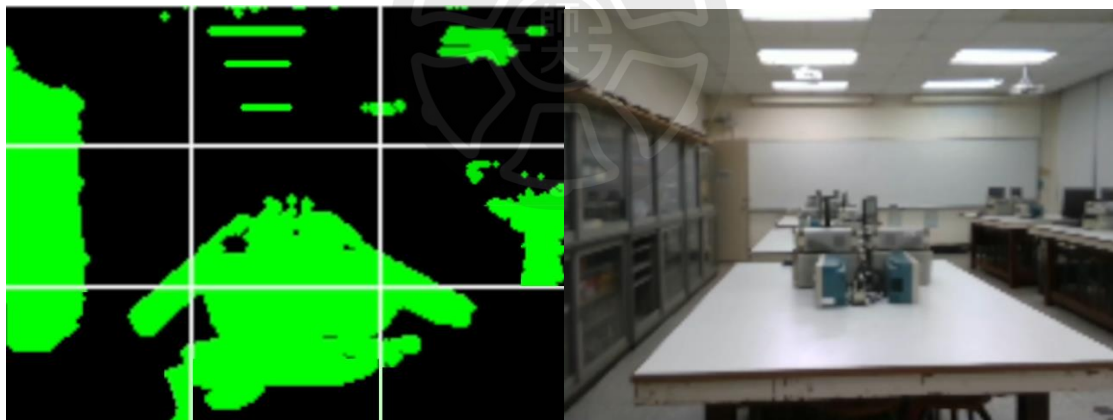


圖 3-7 未做前景分割的完整輪廓

因此，本研究根據障礙物通常伴隨較大運動向量幅度之特性，於類聚完成後實施閾值篩選機制。具體而言，僅保留光流向量幅度高於指定閾值之群集作為最終前景輸出，如圖 3-8 和圖 3-9 所示，為分割後的前景與背景輪廓。最終將所獲得的前景輪廓，將作為後續空間分析之基礎，導入區塊劃分與障礙面積計算機制，以提供避障策略所需之障礙物分佈資訊。

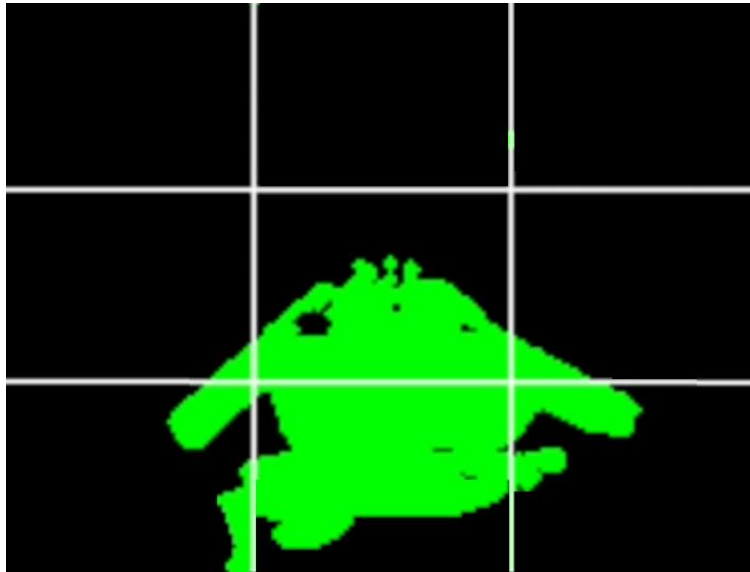


圖 3-8 前景分割後的前景輪廓

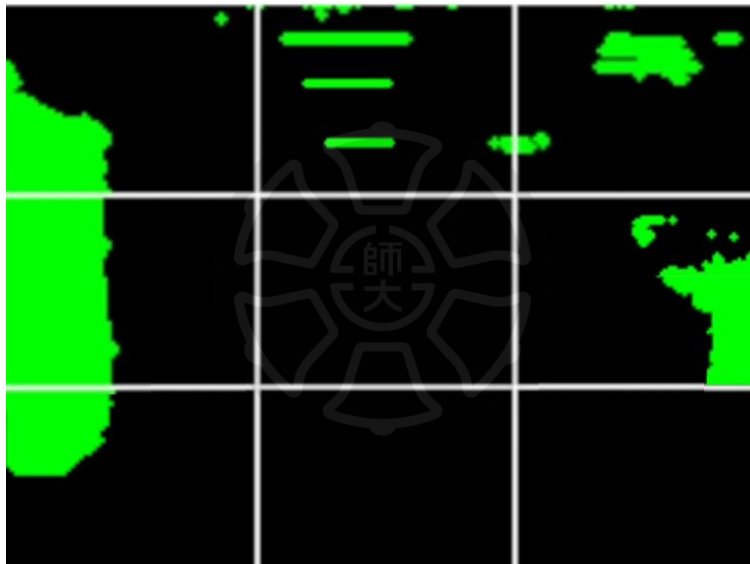


圖 3-9 前景分割後的背景輪廓

### 3.1.4 動態障礙物與光流補償

本研究所採用之 Farneback 光流法結合 DBSCAN 類聚法，能有效在影像中辨識出潛在障礙物。然而，在動態場景中，系統仍面臨挑戰，難以準確辨識出場景內獨立運動的物體。

由於無人機本身運動所產生的全域光流與場景中移動物體造成的局部光流重疊，導致整體光流場混雜，進而影響類聚法對獨立運動物體的辨識能力，常出現輪廓破碎或偵測不完整的問題。

針對此一問題，本研究引入深度資訊作為輔助強化對動態障礙物辨識能力[19]。系統首先透過深度攝像頭獲取場景中的深度圖，並計算連續幀之間的深度變化量。深度變化顯著的區域通常對應於靠近無人機的物體或正在移動的障礙物。針對這些區域，系統進行光流補償。

利用 RealSense 所提供之深度圖資訊，對前後兩幀的深度影像進行逐像素比對，計算絕對深度變化量  $\Delta D$ ：

$$\Delta D(x, y) = |D_t(x, y) - D_{t-1}(x, y)| \quad (3-22)$$

若某像素位置的深度變化量  $\Delta D$  大於閾值  $\theta_D$ ，該區域即視為可能存在動態物體，進而被納入光流補償的處理範圍。

接著，為排除無人機自體運動對光流的影響，需建立背景光流模型。利用 IMU 所提供的線速度  $v_t$  與角速度  $\omega$ ，並結合相機深度值  $Z(x, y)$  與內參參數，估算該像素在理想背景下應產生之背景光流  $v_{bg}$ ，其近似表示如下：

$$v_{bg}(x, y) = \frac{1}{Z(x, y)} Av_t + B\omega \quad (3-23)$$

其中

$$A = \begin{bmatrix} f_x & 0 & -x \\ 0 & f_y & -y \end{bmatrix} \quad B = \begin{bmatrix} \frac{xy}{f_x} & -\frac{f_x^2 + x^2}{f_x} & y \\ \frac{f_y^2 + y^2}{f_y} & -\frac{xy}{f_y} & -x \end{bmatrix} \quad (3-24)$$

其中矩陣  $A$  反映了平移速度對光流的影響； $B\omega$  則描述了角速度產生的光流場。

完成背景光流建模後，將 Farneback 光流法估得之實際光流  $v_{fb}(x, y)$  與  $v_{bg}$  相減，即可得補償後光流  $v_{res}$ ：

$$v_{res}(x, y) = v_{fb}(x, y) - v_{bg}(x, y) \quad (3-25)$$

此殘差光流  $v_{res}$  主要反映場景中獨立運動物體所產生的額外運動分量。最終，系統將殘差光流導入 DBSCAN 進行類聚，以辨識出場景中動態障礙物之位置與輪廓，達到更精確且穩定的動態感知效果。

由圖 3-10 與圖 3-11 之比較可明顯看出光流補償機制對於動態障礙物偵測之效益。圖 3-10 顯示未經補償之情境下，整體光流場混雜，動態物體輪廓破碎。相較之下，圖 3-11 中透過引入 IMU 所提供之自體運動資訊並執行光流補償後，背景光流干擾被有效消除，殘差光流更能凸顯場景中獨立移動物體之運動特徵，使得 DBSCAN 類聚結果更加集中、連貫，動態障礙物之輪廓亦更加完整與清晰。

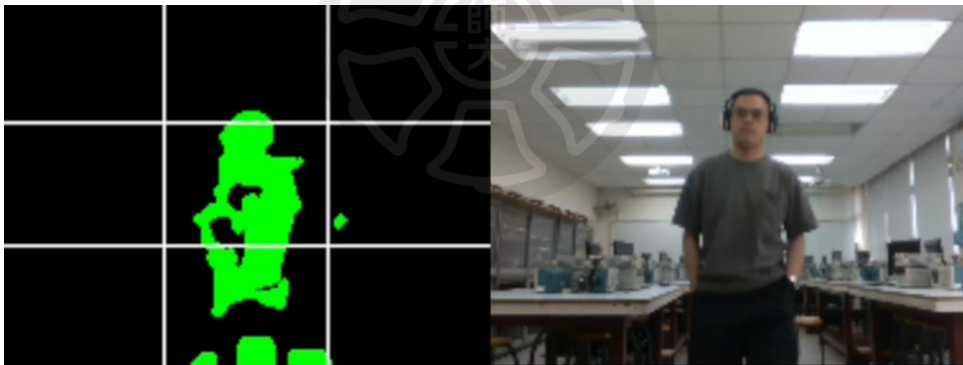


圖 3-10 未做光流補償的動態物體輪廓



圖 3-11 做光流補償的動態物體輪廓

### 3.1.5 劃分與障礙面積計算

為量化前景障礙物在畫面中的分佈與空曠區域，本研究於前景分割完成後，實施畫面區域劃分與障礙面積計算之步驟。

本研究將影像畫面劃分為  $5 \times 5$  的網格區塊，共 25 格，每格對應特定空間位置，作為障礙判斷與後續路徑選擇的基本單位。對每一格區塊，統計其內部落於前景遮蔽區之像素數，並以該區塊總像素數作為基準，計算遮蔽比例。

為避免因局部雜訊或小面積干擾造成誤判，本研究設定遮蔽閾值，僅當某區塊內前景像素數占總像素數超過此閾值時，判定該格為「障礙遮蔽區」。

如圖 3-12，經由遮蔽判斷與網格劃分處理後，系統能夠有效篩選出滿足遮蔽閾值的網格區塊，並記錄其對應之三維中心座標位置。建立完整的空間佔據圖，作為後續動態路徑規劃與避障決策的重要依據。

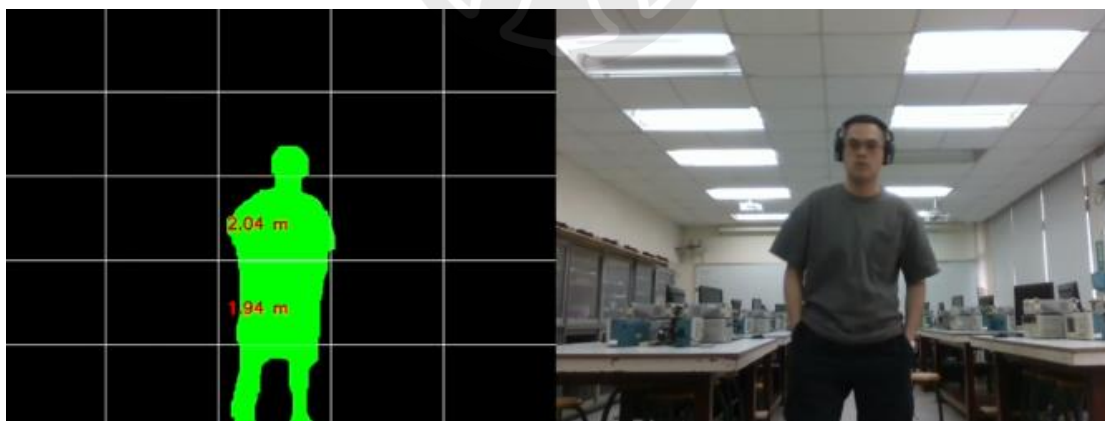


圖 3-12 滿足遮蔽閾值的網格區塊

## 3.2 動態路徑規劃

為實現無人機在動態環境中的自主避障與導引，本研究設計一套結合 BFS（廣度優先搜尋）與 Deprojection（反投影）的動態路徑規劃系統。該系統基於前述的動態障礙物偵測與網格遮蔽資訊，對空間中的可通行區域進行分析與導航決策。

### 3.2.1 BFS 廣度優先搜尋演算法

在本研究設計的動態避障與路徑規劃系統中，廣度優先搜尋演算法被應用於畫面空間中，辨識可通行的最大連續空間區域[20]。為利於區域化分析，系統首先將影像畫面等分為  $G \times G$  的網格，其中  $G \in N$  為指定的網格數，本研究設為  $G = 5$ 。每一網格區塊之尺寸如下：

$$\text{Grid Width} = \frac{W}{G} \quad \text{Grid Height} = \frac{H}{G} \quad (3-26)$$

其中  $W$  與  $H$  為輸入影像的寬與高。根據深度影像中遮蔽區域的像素佔比，定義遮蔽狀態矩陣  $grid(i, j)$  為：

$$grid(i, j) = \begin{cases} 1, & \text{第}(i, j)\text{區塊遮蔽比例} \geq \theta \\ 0, & \text{other} \end{cases} \quad (3-27)$$

其中  $\theta \in [0, 1]$  為遮蔽比例閾值，本研究設定為  $\theta = 0.75$ 。將遮蔽比例高的區塊視為障礙物，其餘區塊視為可通行空間。

為辨識最大空曠區塊，本研究以 BFS 對  $grid \in \{0, 1\}^{G \times G}$  值為 0 之格子進行搜尋，鄰接方向定義為四連通集合：

$$N_4 = \{(-1, 0), (1, 0), (0, -1), (0, 1)\} \quad (3-28)$$

對任一初始空格  $(i, j)$ ，其相鄰可通行格子的集合為：

$$C_{(i, j)} \subseteq \{(x, y) \in [0, G]^2 | (x, y) \text{ 與 } (i, j) \text{ 連通, 且 } grid(x, y) = 0\} \quad (3-29)$$

BFS 演算法會依以下條件擴展連通集合：

$$(x', y') \in N_4(x, y) \wedge \text{grid}(x', y') = 0 \wedge \neg \text{visited}(x', y') \quad (3-30)$$

其中  $N_4(x, y)$  表示點  $(x, y)$  的四個鄰居點。最終，選出最大面積的空曠區域：

$$C^* = \text{argmax} |C_{(i,j)}| \quad (3-31)$$

並計算其中所有格子的重心，推估無人機應前進的方向。此方法可於動態環境下即時辨識通行方向，並提供導航演算法進行路徑更新與避障控制。

如圖 3-13，實驗於真實室內環境中進行，人物作為主要遮蔽障礙物，對無人機前方視野造成部分遮擋。圖 3-14 中可觀察到系統已成功辨識出遮蔽物體的輪廓範圍，並透過 BFS 廣度優先搜尋演算法，自畫面左上角起進行連通區塊的探索。演算法依序擴展相鄰區塊，並在過程中排除所有遮蔽區域，確保搜尋結果僅涵蓋可通行的空曠區域。最終辨識出最大連通的空曠區塊，如圖 3-15，其中紅色區域即為目前畫面中最大且相對安全的可飛行面積。

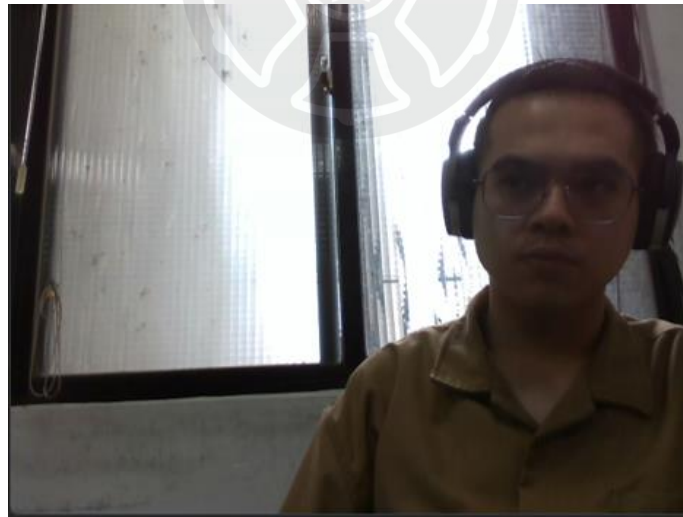


圖 3-13 實際觀測環境

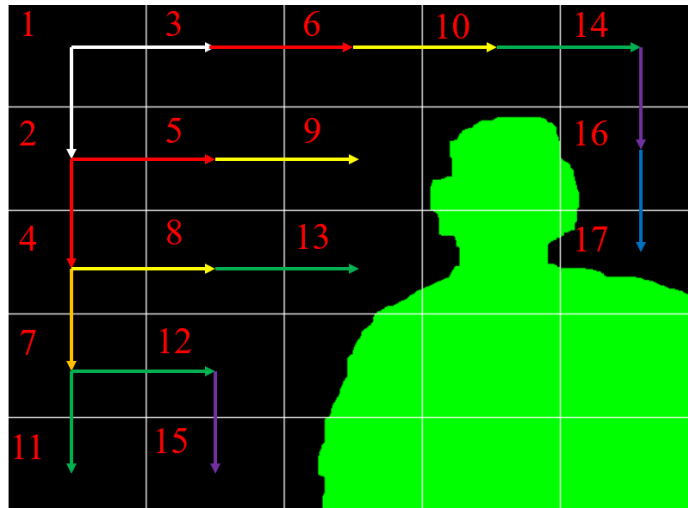


圖 3-14 障礙輪廓和 BFS 推導過程(作者繪製)



圖 3-15 紅色相鄰的空曠面積

如圖 3-16，當場景中出現多個不連續的空曠區域時，系統會對每個獨立區塊分別進行廣度優先搜尋，計算其相連區塊的總面積。經過比較後，選擇面積最大的區域作為當前最可行的移動方向依據。圖 3-17 中以紅色標記顯示出所選之最大空曠區塊，並透過黃色箭頭指出其重心方向，作為無人機前進的導航參考。此作法不僅能有效排除遮蔽障礙對導航的干擾，也能在多重候選空間中自動選擇最佳通行路徑，強化系統在動態避障應用中的穩定性與即時反應能力。



圖 3-16 有兩處空曠區域

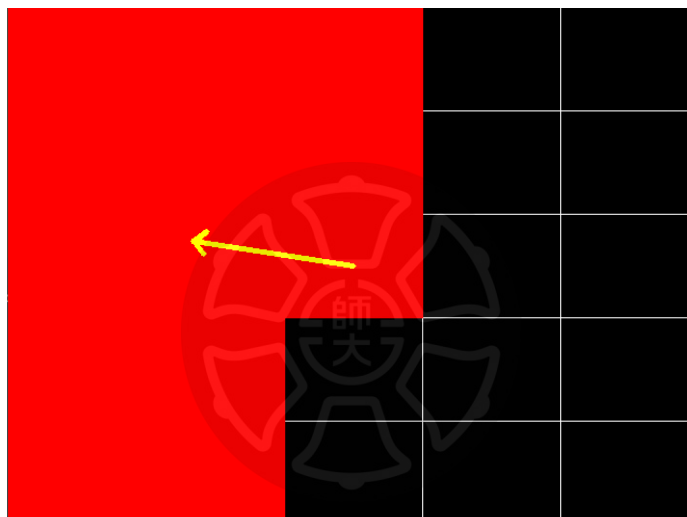


圖 3-17 找到最大面積和移動方向

### 3.2.2 Deprojection 反投影法

在三維電腦視覺中，攝像頭所獲得的影像僅是場景中三維資訊經由投影（Projection）至影像平面的二維結果[21]。此投影過程由攝像頭的成像模型所決定，當中最常見的是針孔攝像頭模型（Pinhole Camera Model）。

如圖 3-18，場景中的一個三維點  $(X, Y, Z)$  在經過相機投影後，其影像上的對應像素座標  $(u, v)$  可由以下公式獲得：

$$\frac{X}{Z} = \frac{(u-c_x)}{f_x} \quad \frac{Y}{Z} = \frac{(v-c_y)}{f_y} \quad (3-32)$$

整理後可得：

$$u = \frac{f_x \cdot X}{Z} + c_x \quad v = \frac{f_y \cdot Y}{Z} + c_y \quad (3-33)$$

其中  $f_x, f_y$  為相機的焦距， $c_x, c_y$  為主點 (principal point) 座標，這些參數可透過相機內部獲得。

相對地，當已知影像中某一像素點  $(u, v)$  的對應深度值  $z$ ，則可進行反投影 (Deprojection) 操作，將此點從影像座標還原回相機座標系下的三維空間座標  $(X, Y, Z)$ 。此過程稱為「反投影」，公式如下：

$$X = \frac{(u-c_x) \cdot z}{f_x} \quad Y = \frac{(v-c_y) \cdot z}{f_y} \quad (3-34)$$

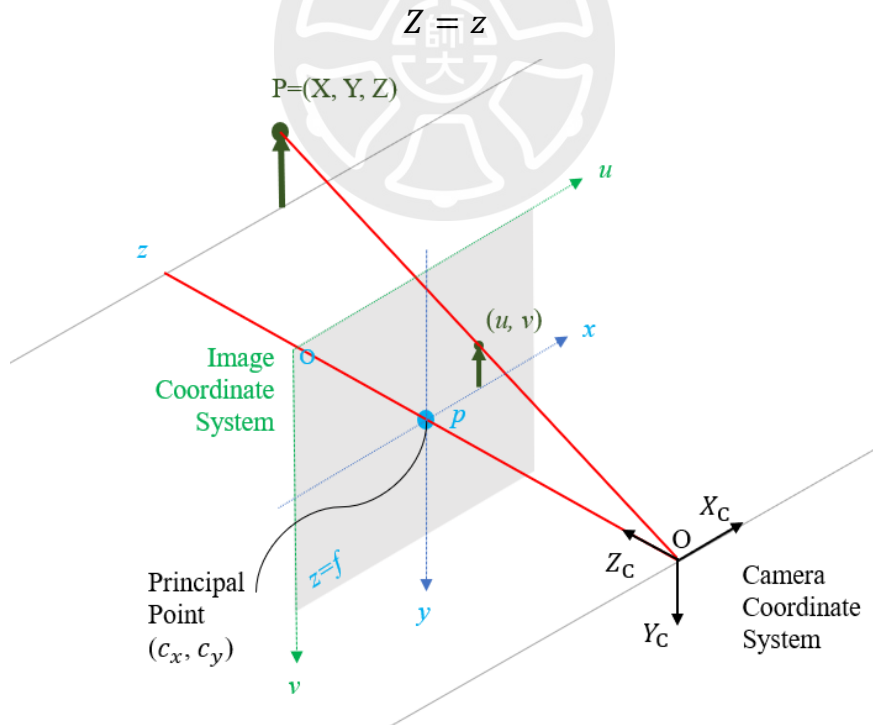


圖 3-18 針孔攝像頭模型(作者繪製)[21]

延續前一章所述，本系統首先排除遮蔽面積達閾值的障礙區塊後，於剩餘區塊中尋找最大連通的通行區域，並計算該區域之重心位置，作為前進的方向。接著，透過 Deprojection 反投影法，將此通行路徑從影像平面轉換至對應的三維空間座標，進而獲得具體的導航向量[22]。

如圖 3-19，紅色區域為透過 BFS 找出的最大空曠連通區，而黃色箭頭即為由該空間重心所推導的最佳移動方向。圖 3-20 至圖 3-23 實現了從影像座標空間到三維世界座標的完整轉換，使系統不僅能在影像上辨識空曠區，更能在實體空間中做出具體且精確的路徑規劃，提升避障導航的實用性與可靠性。

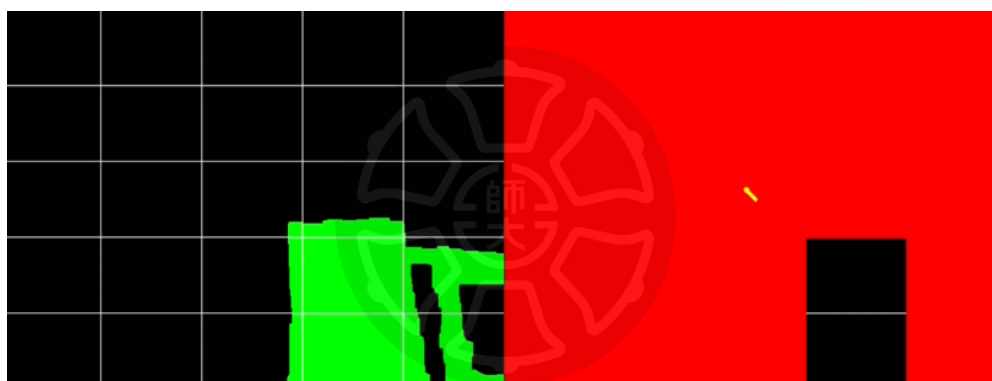


圖 3-19 透過 BFS 演算法獲得影像平面的移動路徑

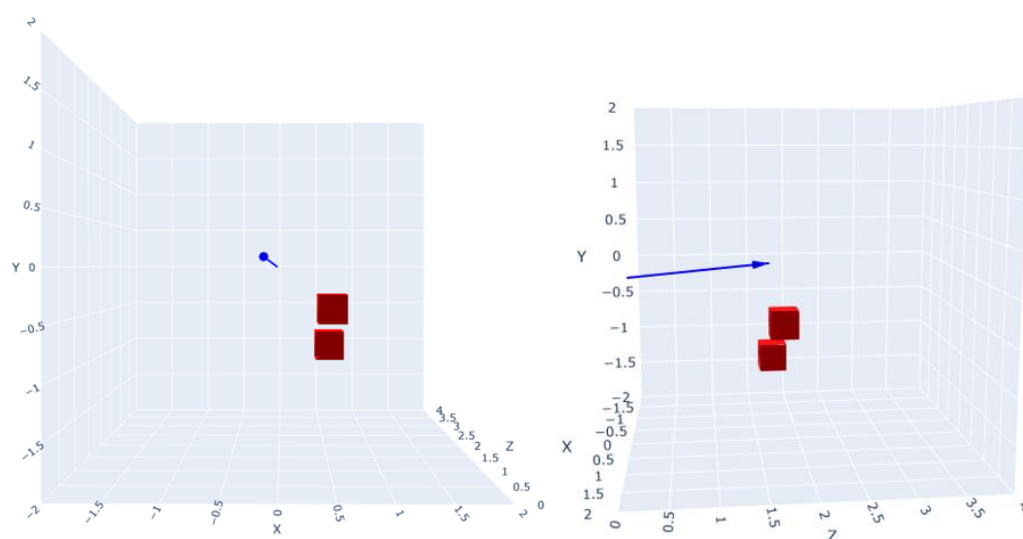


圖 3-20 透過反投影法將障礙物和路徑轉換成三維空間

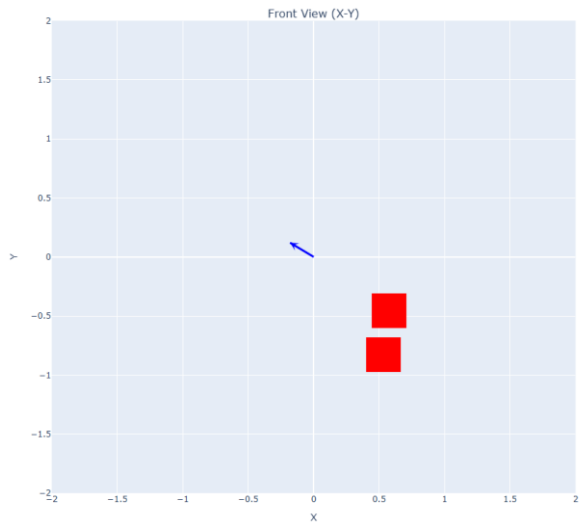


圖 3-21 前視圖

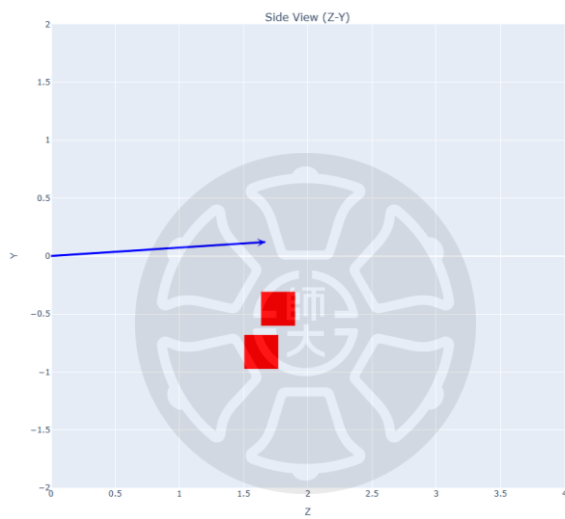


圖 3-22 右視圖

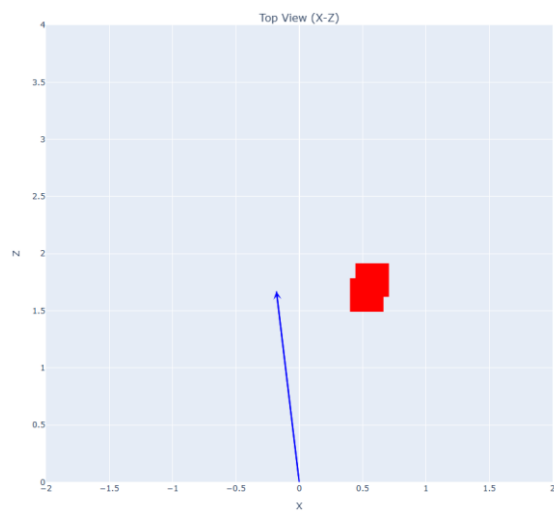


圖 3-23 上視圖

### 3.3 靜態路徑規劃

在動態環境中即時避障雖然至關重要，但若系統能於任務初始即建立一條完整且可行的靜態導航路徑，則有助於整體移動效率與決策穩定性之提升。本研究所設計之靜態路徑規劃機制，建立在已知的地圖資訊之上，透過演算法搜尋可行通路，以規劃出一條從起點至目標點的移動路徑。

藉由靜態與動態路徑規劃機制的雙重整合，系統不僅能預見整體任務目標與最終路徑形貌，更能即時因應突發性障礙或環境改變，使無人機能有效地完成自主導航任務，並具備高度環境適應能力與路徑修正彈性。

#### 3.3.1 Rapidly-exploring Random Tree 演算法

RRT (Rapidly-exploring Random Tree) 為一種取樣式路徑規劃演算法，適用於空間中的可行路徑搜尋問題[23]。其核心思想是透過在空間中隨機取樣點，並以快速擴展的方式建立一棵樹狀結構，使搜尋能有效涵蓋整體空間，進而找到一條從起點至目標點的可行通路。

演算法初始時僅包含起始節點，接著不斷隨機產生新的空間取樣點，並尋找距此點最近的樹中節點，嘗試從此節點向該隨機點擴展一段固定長度的邊，若該擴展邊未穿越任何障礙物，即將新節點加入樹中。如此迭代進行，RRT 可逐步拓展覆蓋整體自由空間，並在某次擴展時觸及終點區域，即表示成功找到一條可行路徑。

如圖 3-24 所示，RRT 演算法的運作流程可透過視覺化圖例說明。紅色圓點為起始節點，而黃色圓點則為目標終點，中間的黑色區塊代表空間中的障礙物。演算法首先在可行空間中隨機生成一個藍色圓點作為擴展目

標，接著搜尋當前樹中距離該目標點最近的節點，並由此節點向目標點方向延伸一段固定距離，產生新的紫色節點，構成樹狀結構的延伸。

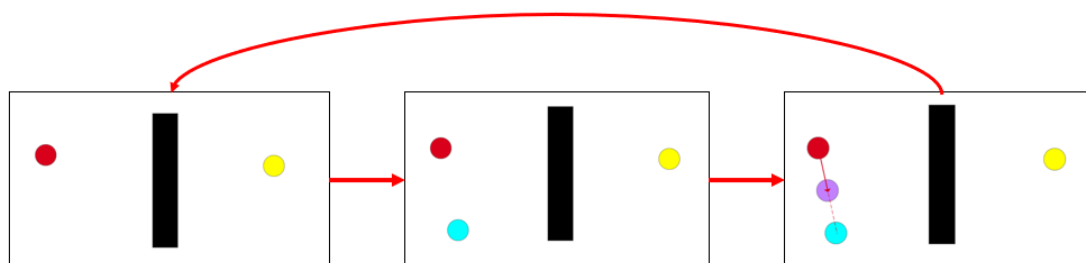


圖 3-24 RRT 演算法流程(作者繪製)

如此不斷重複隨機取樣與向外擴展的過程，樹將逐漸涵蓋整個自由空間，並嘗試繞開障礙物向目標點靠近。圖 3-25 中可見，透過多次生長，RRT 最終形成一條繞過黑色障礙物的節點連線路徑，並朝向黃色終點前進。

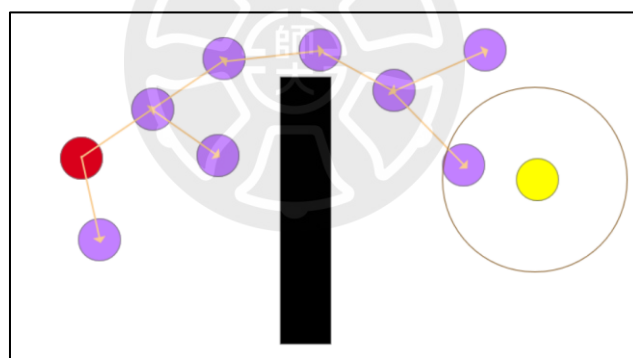


圖 3-25 RRT 演算法結果(作者繪製)

本研究將該演算法應用於真實世界的三維地圖環境中。如圖 3-26，透過 Google Maps 提供的三維建築物輪廓資料[28]，本研究能取得地理環境中建築物的實際分布與形狀。這些資訊可轉換為 RRT 規劃所需的障礙區域模型，使路徑規劃不再侷限於理想化或模擬環境，而能貼合實際場域。

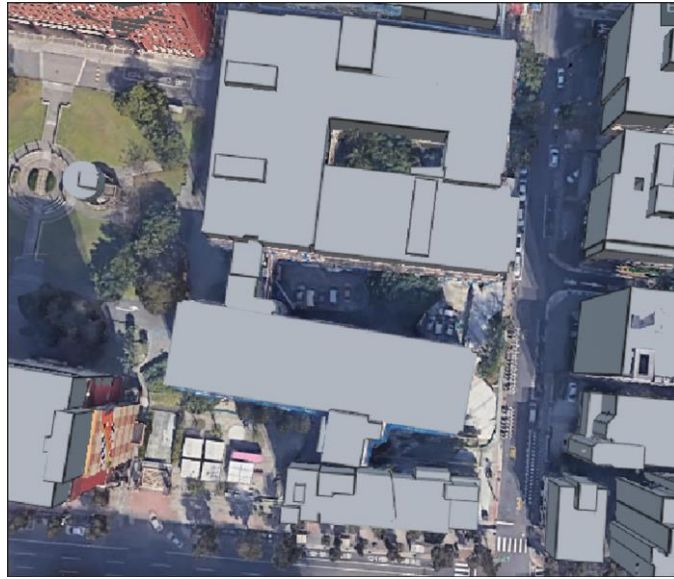


圖 3-26 Google Maps 提供的三維建築物[29]

基於此三維建築資料，RRT 演算法可在障礙物之間搜尋可行的通行空間，自起始位置出發，向目標位置延展樹狀結構，逐步繞過建築輪廓並建立一條可行路徑，如圖 3-27。當樹成功連接起點與終點後，系統將擷取該條路徑上的節點作為導航航點，供無人機依序追蹤飛行，進行避障導航。



圖 3-27 三維建築資料套用 RRT 演算法

### 3.3.2 Rapidly-exploring Random Tree Star 演算法

雖然 Rapidly-exploring Random Tree (RRT) 演算法因其簡單與高效的隨機採樣特性，廣泛應用於空間中的路徑規劃，但其本身也存在限制。最主要的缺點在於缺乏路徑最優化能力。

RRT 著重於迅速建立一棵涵蓋可行區域的樹狀結構，但在連接至目標點後，所產生的路徑往往只是一條可行路徑，而非最短或成本最低的路徑。整體結構容易產生不必要的繞行與彎折，導致規劃出的結果冗長、不自然，甚至無法實用於真實應用。

為了克服上述問題，本研究採用 RRT\* (Rapidly-exploring Random Tree Star) 演算法[24]。首先，從空間中隨機採樣一點，作為欲擴展的目標位置，並尋找當前樹中與其距離最近的節點，再沿此方向延伸出新的節點。不同於原始 RRT 演算法，RRT\* 演算法不僅與最近節點連線，而是在新節點產生後，會以此新節點為中心搜尋一個固定半徑範圍內的所有節點，這些節點構成其鄰近集合。

接下來，演算法會在鄰近集合中挑選一個「最佳父節點」，從起點至新節點累積成本最小者，再將新節點插入樹中並與該點連線。此步驟確保每次新增的節點皆盡可能擁有最低的成本。

完成新增後，RRT\* 接著進行重布線過程：對於新節點鄰近集合內的其他節點，演算法會判斷是否能透過新節點重新連線，進而降低它們的總成本。若是，則更新這些節點的父節點並重新連線。

圖 3-28 展示了 RRT\* 演算法流程。從左上圖開始，新節點  $X_{new}$  被插入現有的 RRT 中，並尋找其鄰近集合。接著，中上圖與右上圖顯示在搜尋半徑內所有候選節點的連線評估過程，最終選擇最佳父節點建立新邊。

接續的下方則顯示重布線的機制，原本與其他父節點相連的節點若經由  $X_{new}$  可獲得更低成本，則會調整其父節點以實現成本最小化。此一機制在演算法執行過程中持續進行，使得整體路徑結構得以不斷優化。

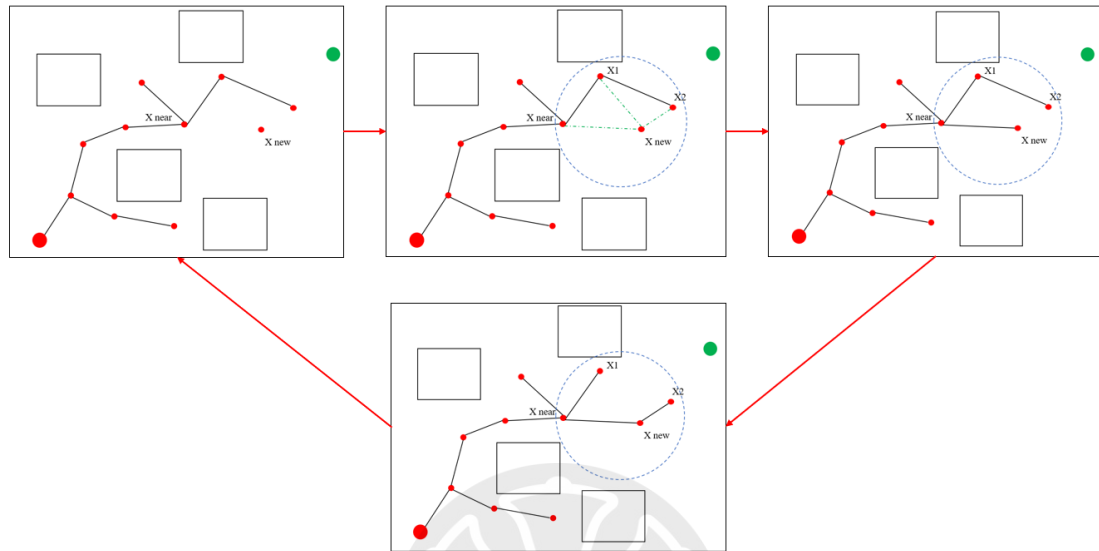


圖 3-28 RRT\*演算法流程(作者繪製)

圖 3-29 與圖 3-30 分別呈現套用 RRT 與 RRT\* 演算法後所產生的路徑結果。從圖 3-29 中可觀察到，RRT 所生成的路徑較為曲折、不平滑，且缺乏成本最小化的規劃策略，並未對整體路徑品質進行優化。

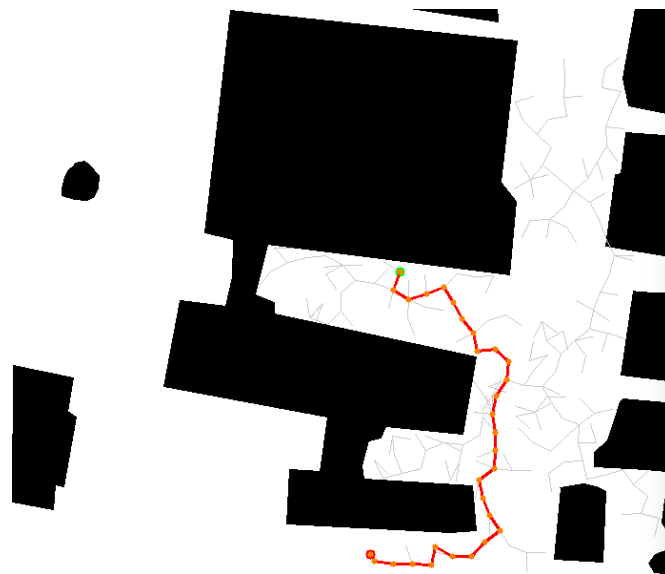


圖 3-29 套用 RRT 演算法的路徑輸出

相較之下，圖 3-30 所示 RRT\* 演算法則透過鄰近節點重連與重布線機制，持續調整並優化整體樹狀結構，使所產生的路徑更為順暢且總長度顯著縮短，展現出其趨近最短路徑的特性。

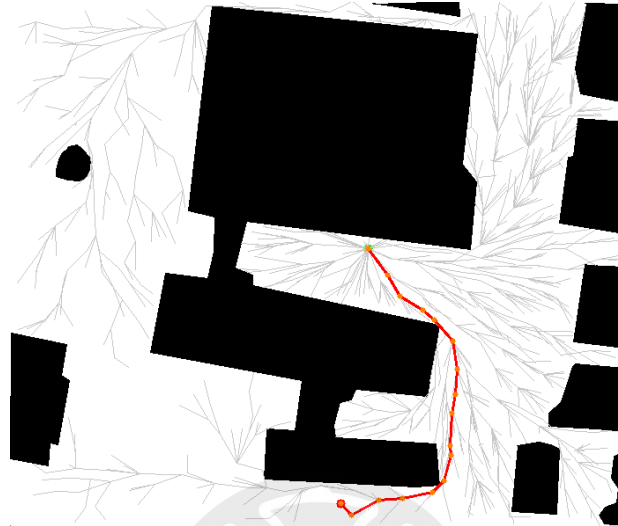


圖 3-30 套用 RRT\*演算法的路徑輸出

## 第四章 四旋翼無人機自主導航系統

在本章節中，我們將介紹本研究所設計的四旋翼無人機自主導航系統之整體架構。該系統整合感測、資料處理與控制三大核心功能，目標為實現無人機於未知環境中即時感知障礙物並進行自主避障與導航。系統架構主要由硬體與軟體兩部分所組成，如圖 4-1 所示。



圖 4-1 四旋翼無人機(作者拍攝)

## 4.1 自主導航系統之架構

本研究所實現之自主飛行系統，整體架構如圖 4-2。系統核心採用 NVIDIA Jetson Nano 作為高階運算平台，負責即時處理來自感測器的資訊，並執行動態與靜態路徑的規劃。Jetson Nano 與微處理器間透過藍牙模組進行資料傳輸，將導航的決策結果傳送至微處理器 MCU。

MCU 作為飛行控制的核心，負責接收 Jetson Nano 傳來之指令後，並執行兩項控制任務，高度控制：根據導航高度目標調整無人機垂直方向之推力。和速度控制：依據目標方向與速度進行機體運動控制，實現平穩前進或轉向。最終，MCU 將 PWM 輸出至各馬達，完成無人機之自主飛行。

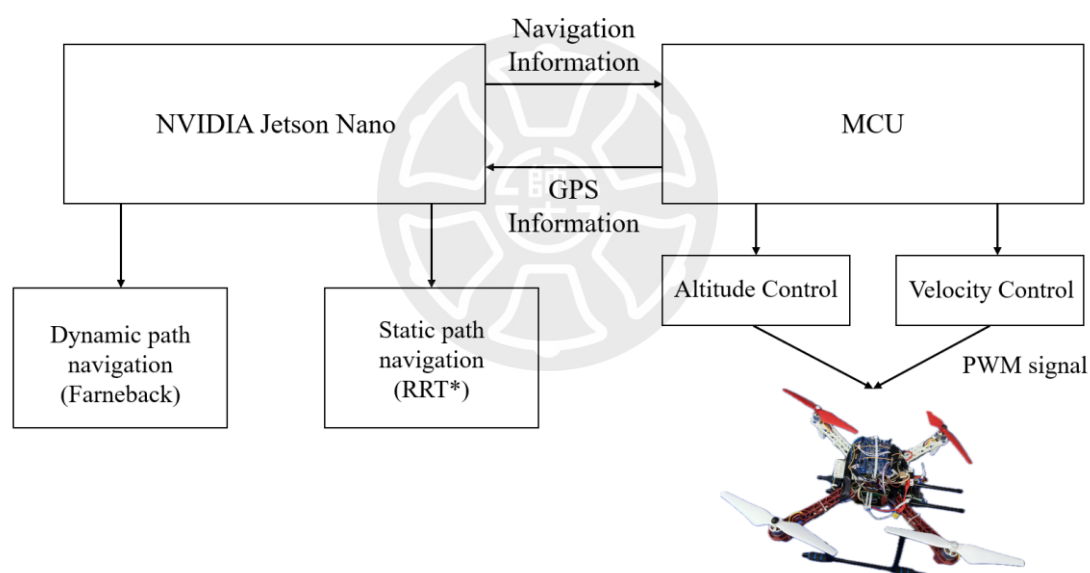


圖 4-2 整體系統架構

## 4.2 硬體架構

本研究所設計之自主導航四旋翼系統，整體硬體架構如圖 4-3 所示，系統主要可區分為「路徑規劃模組」以及「四旋翼飛行平台」兩大部分。

在路徑規劃模組中，核心處理採用 NVIDIA Jetson Nano 作為運算平台，負責即時處理視覺資訊與深度數據，並執行動態與靜態路徑規劃演算法。Jetson Nano 透過 USB 介面與 Intel RealSense D435i 深度相機連接，能同步擷取彩色影像與對應之深度資訊，為後續之障礙物辨識與動態物體分析提供基礎資料來源。為使導航決策結果得以傳至飛控模組，Jetson Nano 內建藍牙通訊功能，並藉由 HC-06 模組與 MCU 實現雙向資料交換，使得高階導航決策與姿態控制能有效協同運作。

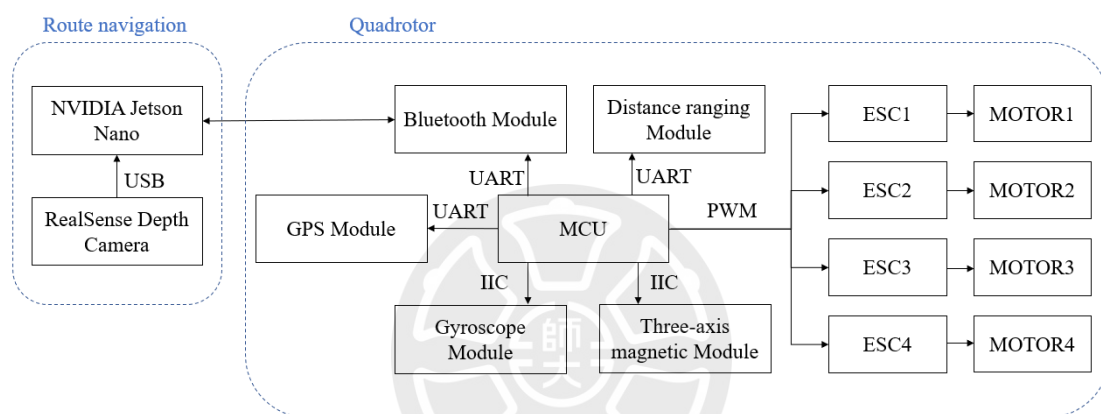


圖 4-3 硬體架構

四旋翼本體部分則以 ESK32-30501 (HT32F52352) [26]微控制器為核心控制平台，並整合多項感測模組與飛行控制元件。系統藉由 MPU6050 陀螺儀模組與三軸磁力計，提供飛行器在三維空間中的姿態感知，透過 I<sup>2</sup>C 通訊方式傳送至 MCU 進行融合處理，實現穩定飛行控制。同時，系統搭載 PV3953L1 雷射測距模組以即時獲取與地面或障礙物之相對距離，並整合 Neo-M8N GPS 模組提供位置資訊，使導航系統能針對全域位置進行修正與調整。在動力控制方面，MCU 控制四組無刷直流馬達 (kv950) 並透過四個電子速度控制器 (ESC) 輸出 PWM 訊號調整轉速，以驅動四個旋翼，完成飛行姿態與速度控制任務。

整體系統之使用材料彙整如表 4-1 所示，涵蓋所有核心元件之型號與數量，包括處理器模組、感測器模組、執行器與電源模組等，建構出具備感測、決策與控制能力的一體化自主飛行平台。

表 4-1 硬體元件模組表

型號	數量
ESK32-30501(HT32F52352)	1
NVIDIA Jetson-Nano	1
Camera(Intel Realsense D435i)	1
Gyroscope sensor(MPU6050)	1
Bluetooth Module(HC-06)	1
ESC:Electronic Speed Controller	4
Brushless Dc Motor(kv950)	4
AA Battery	3
Battery 14.8V 1850mAh	1
Distance ranging Module(PV3953L1)	1
Buck Converter (LM2596S DC-DC)	1
GPS Module(Neo-M8N)	1
Three-axis magnetic Module(GY-271)	1

#### 4.2.1 Realsense 鏡頭模型.

本研究所採用之深度相機為 Intel RealSense D435i 圖 4-4，該鏡頭具備同步輸出彩色影像與深度資訊之能力，能有效支援移動平台於複雜環境中之三維空間感知[27]。



圖 4-4 Intel RealSense D435i(作者拍攝)

RealSense D435i 採用主動式立體視覺結構，搭配紅外線投影器與雙目紅外鏡頭，可於室內與弱光環境下維持穩定的深度估測。其深度感測範圍可達 0.2 公尺至 3 公尺，其深度誤差隨距離增加略有提升，如圖 4-5，整體誤差值可於實際距離的 3% 以內。並具備  $87^{\circ} \times 58^{\circ}$  的視角範圍，足以涵蓋無人機於飛行過程中前方的主要視野。

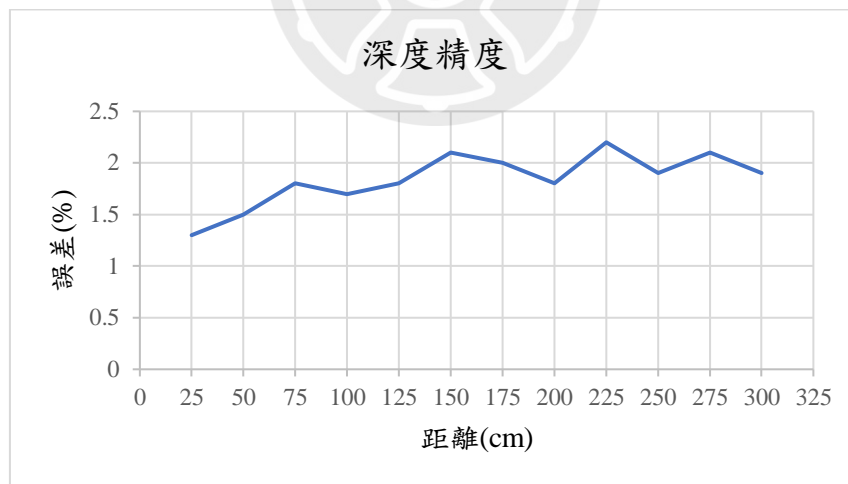


圖 4-5 深度精度

此外，該鏡頭支援 USB 3.0 高速傳輸介面，可於 30 FPS 下穩定輸出 1280x720 的影像，其輸出之彩色與深度影像將作為動態障礙物辨識與光流估測之基礎。

## 4.2.2 NVIDIA Jetson Nano 開發套件

如圖 4-6，在本研究中，Jetson Nano 作為整體導航系統的中央運算單元，負責接收來自 RealSense 深度相機與所提供的感測資訊，並即時執行影像處理、光流分析、動靜態障礙物辨識與路徑規劃等核心演算法。

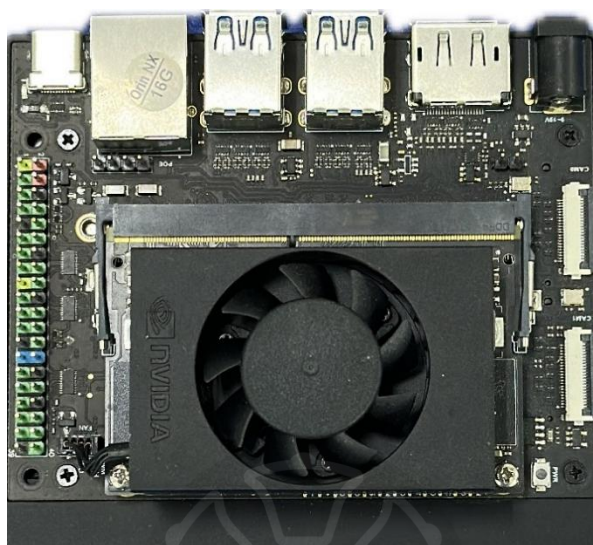


圖 4-6 NVIDIA Jetson Nano

此外，Jetson Nano 搭配藍牙模組與 MCU 通訊，將運算結果轉換為行動指令下達至飛行控制系統。

## 4.2.3 GPS 元件

如圖 4-7，本研究所採用之 GPS 元件為 Neo-M8N，該模組具備高靈敏度與快速定位能力，能在多變且遮蔽程度高的環境中提供穩定且準確的定位資訊[28]。Neo-M8N 具有 10 Hz 的更新頻率，可即時回報經緯度、海拔高度、速度、航向等資訊，並透過 UART 介面與 MCU 進行資料傳輸。



圖 4-7 Neo-M8N GPS 元件

在本系統中，Neo-M8N GPS 元件主要用於提供無人機的即時絕對位置，作為地理座標的基準點。

### 4.3 軟體架構

本系統軟體架構整合了影像處理、三維空間運算、路徑規劃與飛行控制等模組，形成一套具備即時感知與決策能力的無人機自主導航平台。整體流程自影像輸入開始，依序經由障礙物辨識、動靜態路徑規劃模組推導可行移動方向，最終由控制系統輸出對應的飛行指令。以下將詳細說明各子系統功能與其在整體架構中的角色。

#### 4.3.1 障礙物辨識規劃

本系統以 RealSense 深度相機作為輸入來源，搭配金字塔式 Farneback 光流法與深度差異分析，達成即時障礙物的偵測與分離。整體辨識流程如圖 4-8。

首先，系統從深度相機讀取連續兩幀的深度與彩色影像，並將其按比例進行金字塔降解析度，接著，利用金字塔影像對進行 Farneback 光流估計，計算每一像素點的移動向量，並計算前後兩幀之深度變化。

為提升動靜分離的準確性，系統判斷是否存在劇烈的深度變化。若是，將視為場景中存在動態物體，需進行背景光流建模與補償處理。此階段透過推估背景區域光流分布，對整體光流場進行補償修正，排除因攝影機自動運動所造成的漂移，留下動態物體所造成的相對運動向量。隨後，透過 DBSCAN 類聚演算法對補償後光流場進行聚類處理，提具一致運動特徵的前景障礙物輪廓。對於有效的類聚區塊，系統會計算其平均移動方向向量，並建立前景遮蔽物輪廓圖。為避免雜訊誤判，流程中也設有閾值過濾機制，將運動幅度過小的區塊自動排除。最終，輸出完整的前景輪廓圖，供後續的路徑規劃進行即時避障決策。

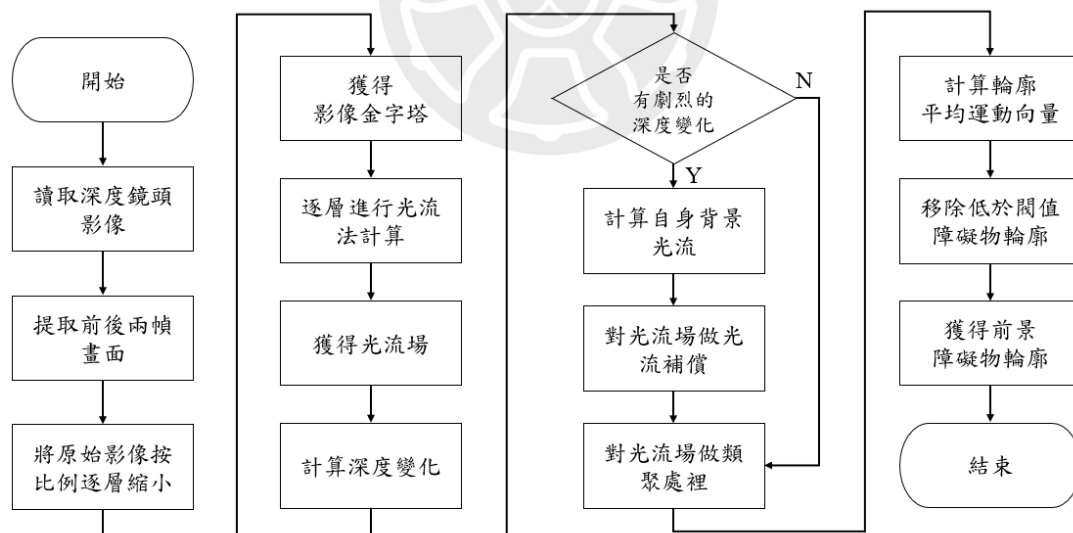


圖 4-8 障礙物辨識流程

### 4.3.2 動態路線規劃

本系統負責根據前一階段障礙物辨識所輸出的輪廓圖，進行即時避障區塊判斷與三維導引方向推算。透過網格分割與區塊聚合分析，結合深度資訊與影像反投影技術，推導出無人機應在三維空間中轉向或避開的動態導航向量，整體流程如圖 4-9。

首先，系統根據障礙輪廓圖進行畫面分割，將影像區域劃分為 5x5 的網格。使障礙物的區域分佈具備位置資訊，並利於區塊分析與導航路徑推導。接著，系統檢查每一區塊的遮蔽像素比例來評估該區塊是否為障礙區。若未達到門檻，則重新等待下一幀輸入；若達門檻則進入區塊分析階段。接下來會提取遮蔽輪廓內的所有障礙塊，並將其從原始網格中排除，以取得剩餘可通過的網格位置。此階段採用廣度優先搜尋演算法進行連通區塊標記，並取得所有可通過的網格區塊。若可通過區塊數為複數時，系統會比較各區塊面積，選取面積最大的區塊作為主要通行通道，並擷取該區塊的幾何重心作為導航基準點。最後，系統結合相機內部參數，將該選定重心進行反投影，轉換為真實世界中的三維座標點，並據此計算出三維導引向量，提供控制系統進行姿態調整或避障操作。

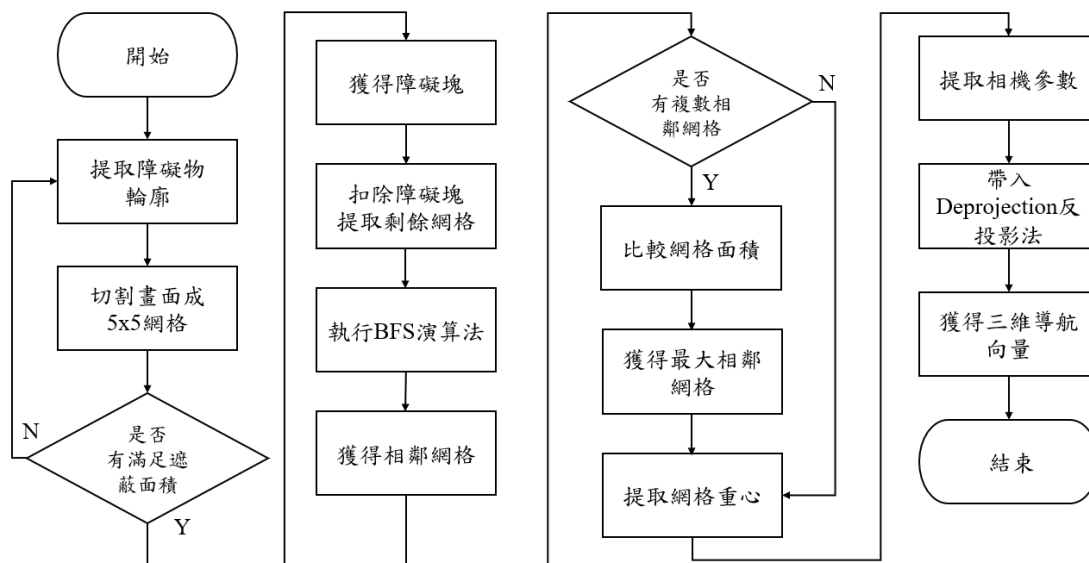


圖 4-9 動態路線規劃流程

### 4.3.3 靜態路線規劃

本研究採用 RRT\* (Rapidly-exploring Random Tree Star) 演算法作為靜態環境下之導航路線規劃核心，流程如圖 4-10 所示。首先，系統將載入三維環境中的建築物資訊，建立障礙物模型。使用者需輸入起點與終點座標作為導航規劃的邊界條件，並由系統開始隨機產生新的取樣點以探索可行空間。

在每一次迭代中，RRT\* 會先尋找當前樹狀結構中距離該取樣點最近的節點，並嘗試從該節點朝取樣點方向延伸一個新的節點。當新節點生成後，演算法將於其附近半徑範圍內搜尋所有已存在的節點，進行鄰近節點的搜尋，結合成本進行綜合評估，以尋找可成為「最佳父節點」的连接候選。在連線新節點後，系統將檢查是否能透過該新節點來降低其鄰近節點的總成本；若能達成成本下降，則會重新連接這些節點至新節點，進行樹

的重布線。演算法持續進行上述取樣與優化步驟，直到成功連接至終點為止，最終即能獲得一條由起點至終點、且避開建築物的導航路徑。

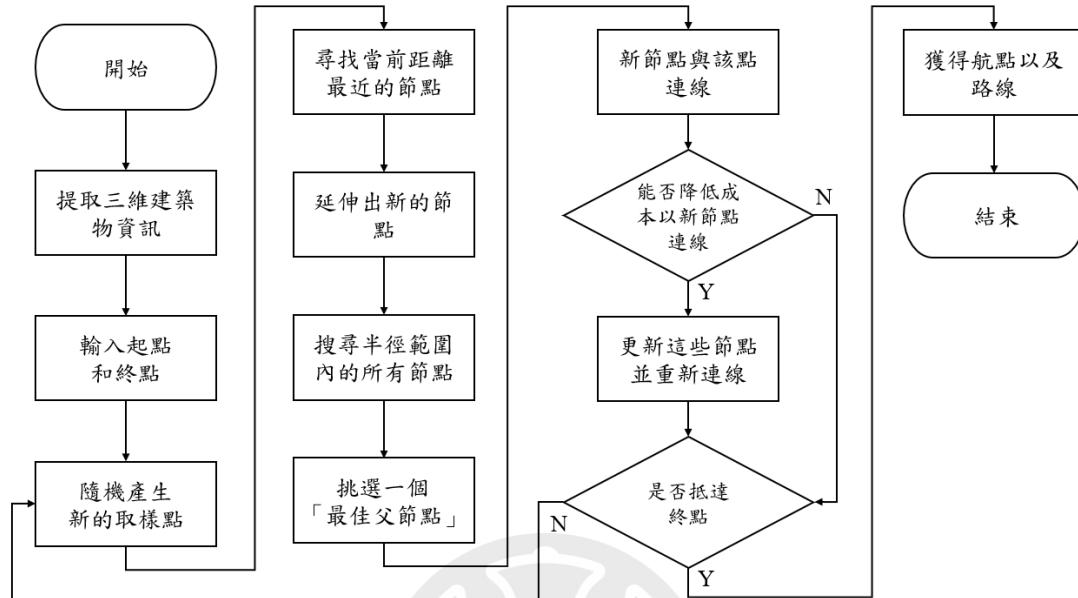


圖 4-10 靜態路線規劃流程

#### 4.3.4 四旋翼無人機控制系統

本研究所設計的四旋翼無人機控制系統，整合了 GPS、磁力計與馬達輸出控制，並透過藍牙與 Jetson Nano 進行即時通訊與導航任務接收，其詳細流程如圖 4-11。系統初始化後，首先與 Jetson Nano 建立藍牙連線，確保無線通訊穩定。完成連線後，無人機將進入懸停等待階段。

在等待過程中，無人機會持續回傳自身的 GPS 位置與磁場方向資訊至 Jetson Nano，作為導航任務判斷之依據。Jetson Nano 根據環境分析與預先的路徑規劃，決定下達的導航指令。接收到導航指令後，控制系統會依據指令，更新無人機的控制參數。這些參數涵蓋姿態修正、偏航調整與推進力分配，進而產生對應的 PWM 訊號，輸出至馬達進行動態控制。藉

由持續回饋位置與方位資料，並根據導航指令即時調整飛行控制參數，四旋翼無人機可達成自主導航與移動控制。

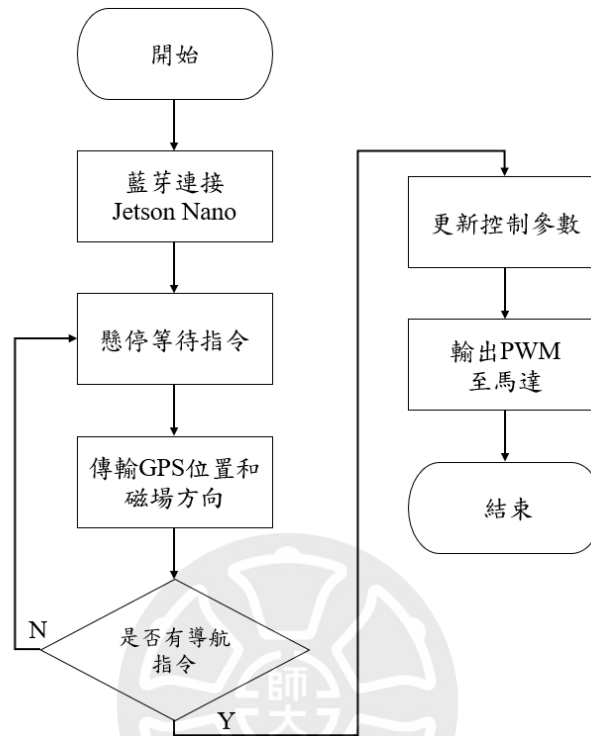


圖 4-11 四旋翼無人機控制流程

## 第五章 實驗設計與結果

本章旨在驗證本研究提出之系統在實際應用中的可行性與效能。實驗內容主要分為三個部分，分別為：障礙物辨識實驗、無人機避障實驗，以及自主導航實驗。各項實驗皆於控制環境下進行，並記錄系統在不同條件下的反應與表現，以佐證其穩定性與實用性。

### 5.1 障礙物辨識實驗

為驗證本系統於不同場景下對障礙物之辨識能力，本章將障礙物辨識實驗依障礙物特性分為靜態障礙物辨識實驗與動態障礙物辨識實驗，分別說明實驗設計、評估指標與預期效能。

#### 5.1.1 靜態障礙物辨識實驗

本實驗旨在驗證無人機搭載深度鏡頭與光流演算法之障礙物感知能力，評估其在不同距離下對障礙物輪廓的辨識完整性。

圖 5-1 和圖 5-2 為本次靜態障礙物辨識實驗的整體流程和實驗示意圖。實驗開始時，啟動無人機與深度鏡頭，系統會開啟障礙物辨識流程。無人機從預設起點開始向前移動，沿途每 50 公分紀錄一次當前視野下的障礙物影像資訊，並同步擷取辨識到的障礙物輪廓。現場記錄位置下的真實障礙物外型，作為對照依據。

為評估系統對不同尺寸障礙物之適應性，本實驗設計了兩種尺寸進行測試，分別為大型障礙物(60\*90cm)和小型障礙物(30\*60cm)，兩種障礙物皆為平面結構，設置於相同高度與位置，並保持正對無人機飛行方向，以確保辨識條件一致。

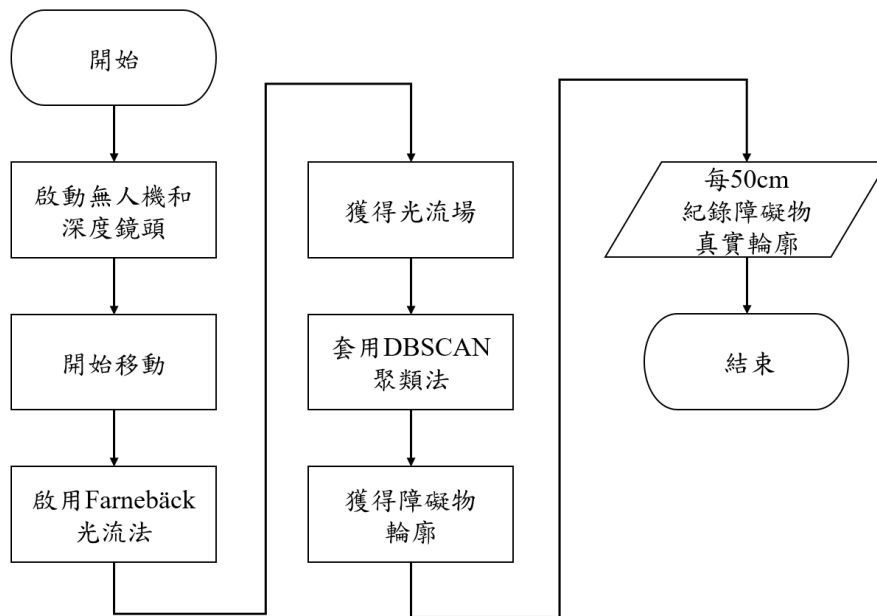


圖 5-1 靜態障礙物辨識實驗流程

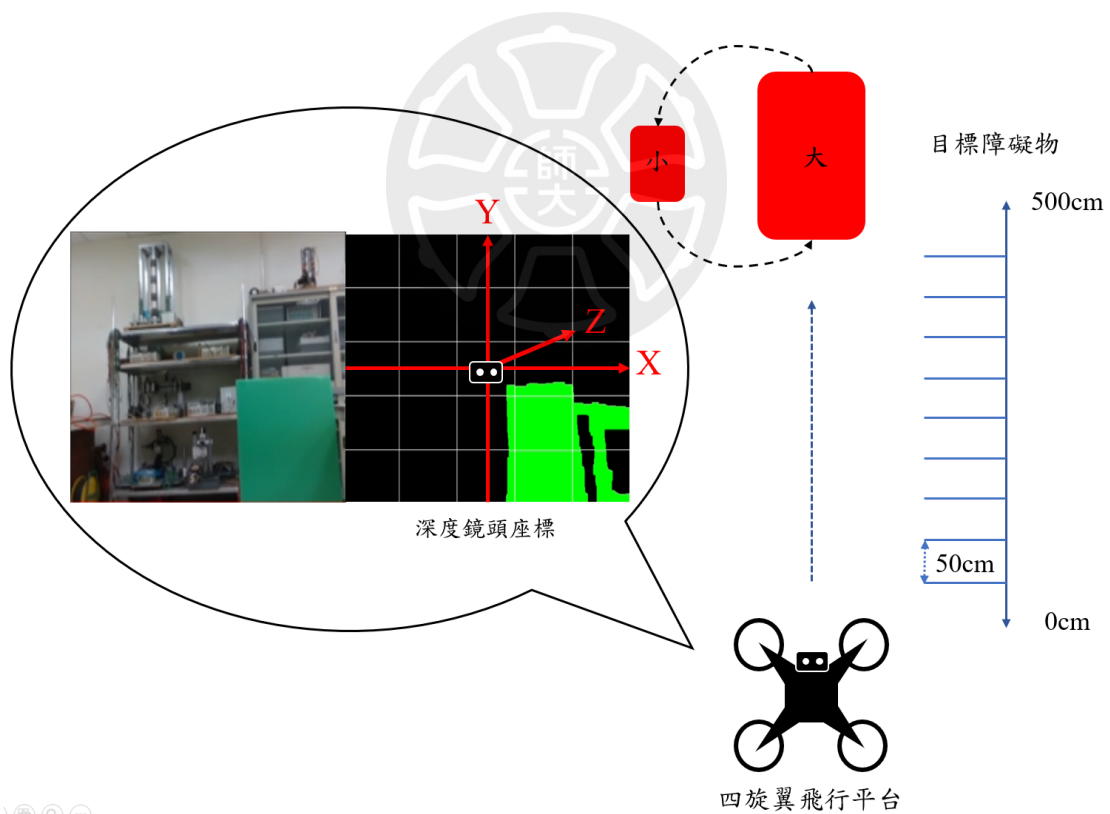


圖 5-2 實驗示意圖

本實驗的最終評估指標為輪廓覆蓋率 (Contour Coverage Rate)，即系統所辨識之障礙物輪廓與實際障礙物外型在空間中的重疊程度。此指標可有效反映系統在不同距離與條件下對障礙物外型掌握的完整性，作為整體障礙物辨識效能的核心衡量標準。

實驗中，為確保實驗數據的穩定性與代表性，每個位置皆進行了 25 次重複測試，整體辨識過程從 500 公分至 50 公分，共涵蓋 10 個距離點 2 種尺寸 25 次測試，共取得 500 筆辨識輪廓數據，最終統計各距離下的平均輪廓覆蓋率，以完整反映系統辨識性能。

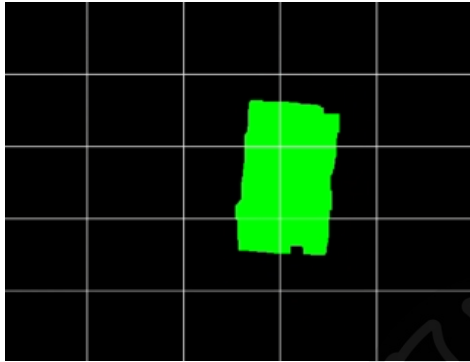
表 5-1 呈現系統於特定時間點所擷取之影像與遮蔽區塊辨識結果，並附上該時刻所計算之障礙物空間座標資訊，提供辨識性能的直觀對照。

表 5-1 靜態障礙物辨識實驗結果

實驗圖	動作說明
<div style="text-align: center;">  <p>上為無人機真實視野 下為障礙物辨識結果</p> </div>	<p>第 0 秒， 啟動無人機與深度鏡頭</p>



上為無人機真實視野  
下為障礙物辨識結果



第 0.2 秒，  
無人機開始移動，  
並透過辨識流程獲得 500cm 處障礙  
物輪廓。

記錄其輪廓與真實畫面。

障礙物位置：

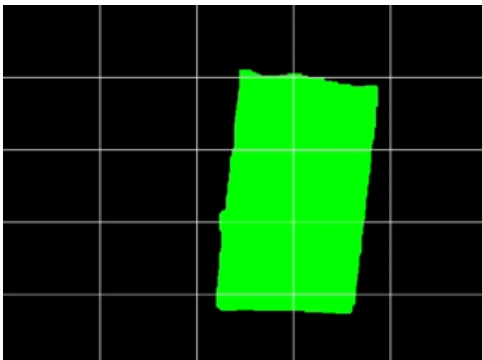
x: 0.4 m

y: -0.3 m

z: 5.21 m



上為無人機真實視野  
下為障礙物辨識結果



第 4.8 秒，  
持續移動直至記錄完 50cm 處輪廓。

障礙物位置：

x: 0.31 m

y: -0.21 m

z: 0.53 m

靜態障礙物辨識實驗結束。

從圖 5-3 至圖 5-5 可觀察到，障礙物在無人機逐步接近過程中的三維座標變化，本研究建構之視覺定位系統中，為統一障礙物座標判讀方式，定義相機的座標軸如下：相機正前方為 Z 軸正方向，右側為 X 軸正方向、左側為 X 軸負方向；上方為 Y 軸正方向、下方為 Y 軸負方向。其中，X 與 Y 軸座標皆呈現微幅的位移，主要於無人機接近障礙物後，影像中所辨識的輪廓逐漸清晰，使得系統所計算的障礙物質心位置更加準確，導致座標出現微小的調整。另一方面，Z 軸即障礙物與無人機之間的深度距離，則呈現穩定趨勢，由約 5 公尺逐步縮短至 0.5 公尺，顯示無人機在前進過程中，系統能夠穩定追蹤障礙物的距離資訊。

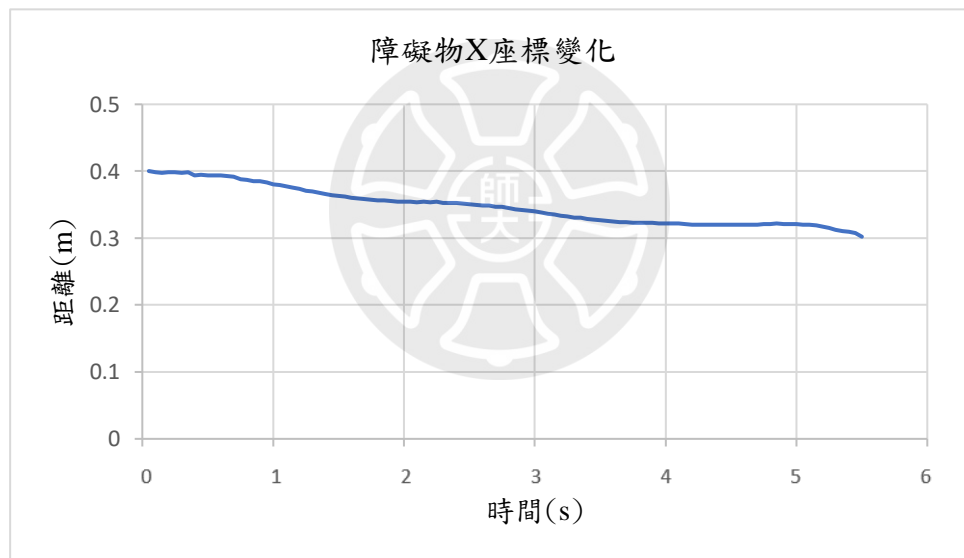


圖 5-3 障礙物 X 軸變化圖

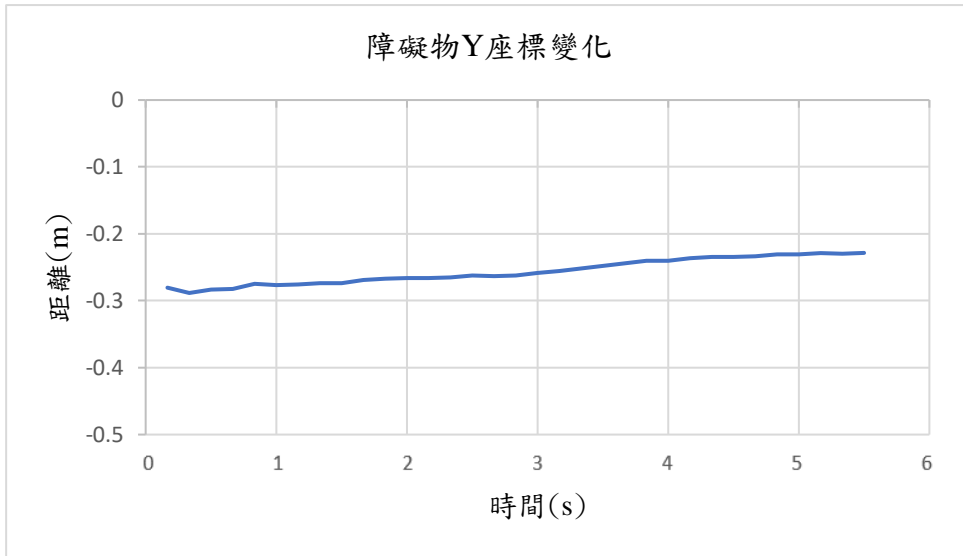


圖 5-4 障礙物 Y 軸變化圖

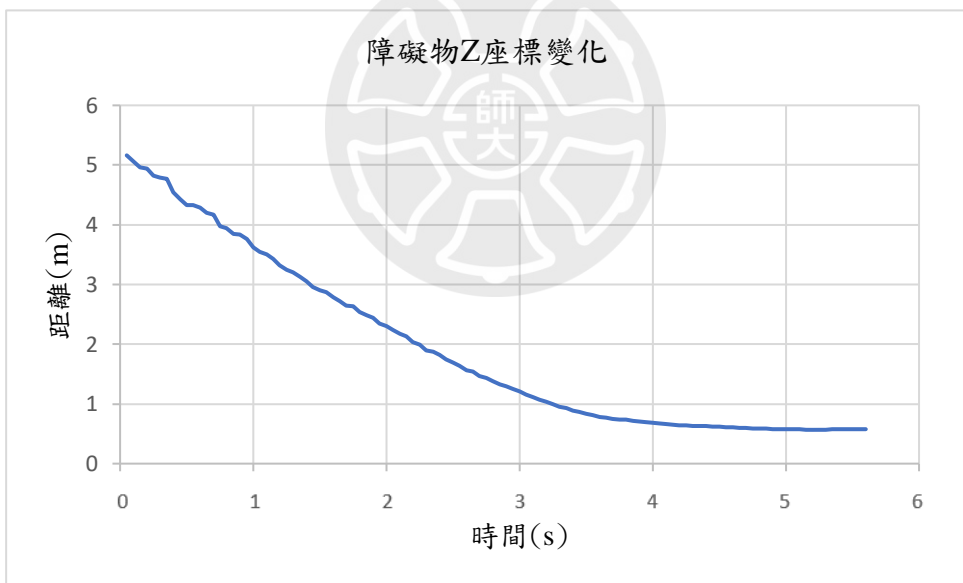


圖 5-5 障礙物 Z 軸變化圖

從表 5-2 數據中可發現，輪廓覆蓋率隨距離減少有提升的趨勢。在 500 公分遠處，受限於障礙物在影像中的佔比較小，解析度較低，導致影像細節不足，使光流運算結果容易產生模糊與失真，使得輪廓辨識效能相對不穩定。其中，大型障礙物平均輪廓覆蓋率為 78.5%，小型則僅為 68.2%；隨著距離逐步縮短，至 350 公分時，各類障礙物輪廓覆蓋率均突破 80%，在此距離下，系統已能穩定辨識障礙物的主要外型。

表 5-2 靜態障礙物辨識實驗數據

	距離位置(cm)	平均輪廓覆蓋率(%)	
		大型障礙物	小型障礙物
靜態障礙物辨識實驗	500	78.5	68.2
	450	81.3	72.5
	400	86.6	76.8
	350	91.7	81.3
	300	92.2	85.9
	250	96.1	89.7
	200	95.5	92.6
	150	98.3	93.1
	100	98.7	96.8
	50	99.2	97.5

### 5.1.2 動態障礙物辨識實驗

本節動態障礙物辨識實驗旨在驗證本研究所建構之結合影像光流與深度資訊之動態辨識流程，於多樣移動情境下對障礙物輪廓之擷取能力。實驗中，障礙物自無人機側方接近，並模擬三種常見人類移動模式，分別為漫步（0.5 至 0.8 m/s）、快走（1.0 至 1.2 m/s）與跑步（1.8 至 2.1 m/s），對應低速、中速與高速等不同級距之移動體。同時無人機本身也以穩定速度向前移動，模擬真實飛行過程中雙方皆具相對運動的情境。

圖 5-6 和圖 5-7 為實驗的整體流程和實驗示意圖，系統會於無人機啟動與深度鏡頭初始化後，開始擷取畫面與深度資訊。當無人機與障礙物皆啟動移動後，進入主要辨識階段。首先透過 Farneback 光流法計算像素運動資訊。為排除來自無人機自我移動所產生的背景干擾，系統會估計自身背景光流並進行光流補償，藉此凸顯場景中具有相對運動的物體。補償後的光流場交由 DBSCAN 聚類法分析，擷取動態障礙物輪廓。

在進行障礙物輪廓擷取後，系統每前進 50 公分即進行一次輪廓記錄，並與實驗中的障礙物真實輪廓作比較，計算輪廓覆蓋率，作為評估辨識準確度的依據。為全面探討系統對不同尺度障礙物的辨識能力，本實驗針對兩種尺寸進行個別測試與紀錄。

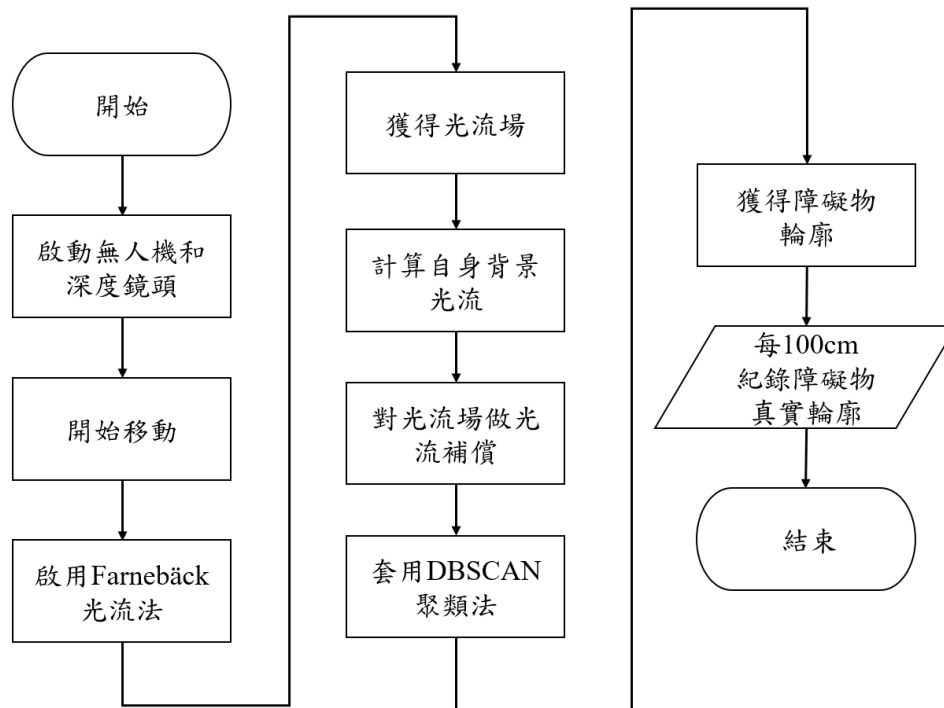


圖 5-6 動態障礙物辨識實驗流程

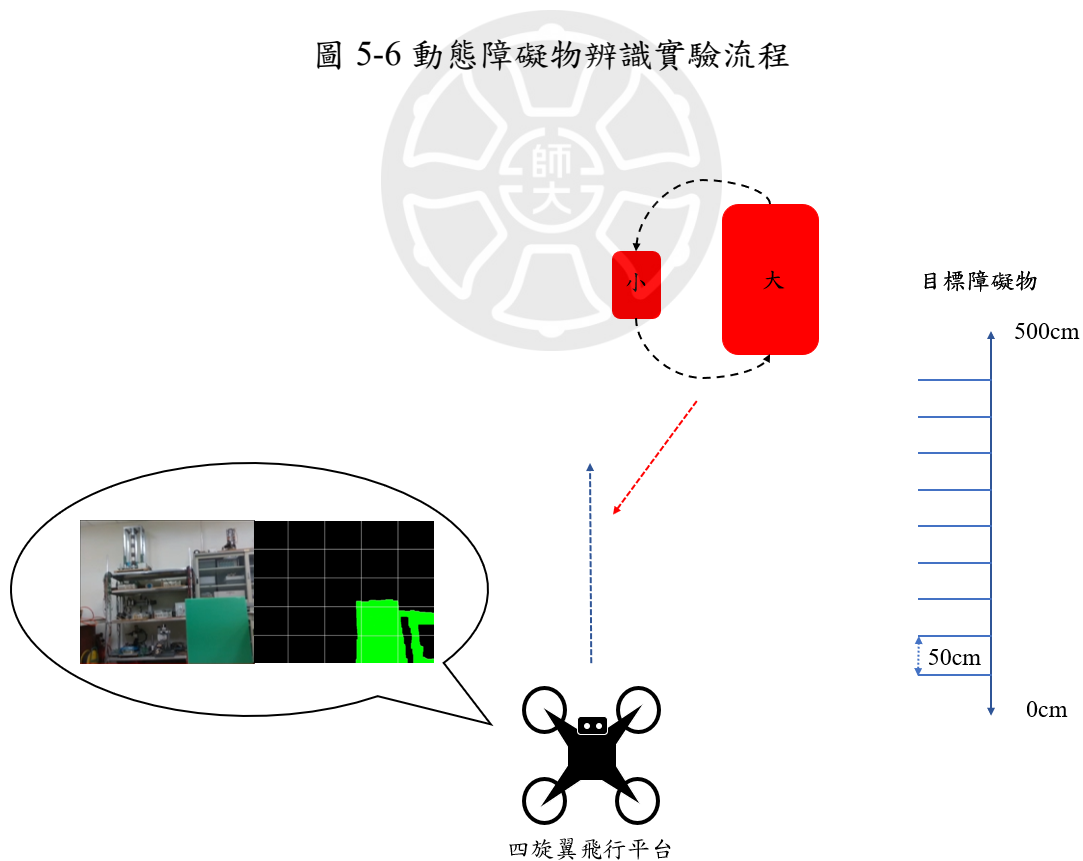


圖 5-7 實驗示意圖

為確保數據的統計穩定性與實驗的重現性，每組距離與速度條件下皆重複進行 25 次測試，最終取各距離位置輪廓覆蓋率的平均值作為代表數據。整體共涵蓋兩種尺寸及三種速度，並在多段距離下做 25 次測試，累積大量輪廓辨識結果，作為後續性能分析之依據。

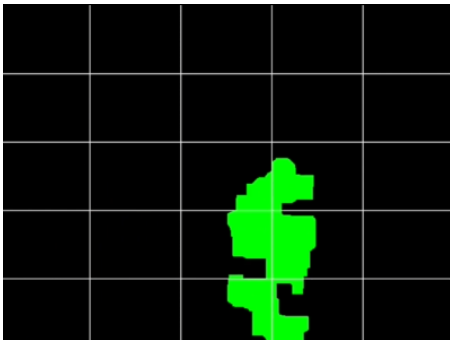
表 5-3 呈現系統於特定時間點所擷取之影像與遮蔽區塊辨識結果，並附上該時刻所計算之障礙物空間座標資訊，提供辨識性能的直觀對照。

表 5-3 靜態障礙物辨識實驗結果

實驗圖	動作說明
<div data-bbox="306 853 715 1160" data-label="Image"> </div> <p data-bbox="365 1182 655 1263">上為無人機真實視野 下為障礙物辨識結果</p> <div data-bbox="296 1272 727 1603" data-label="Image"> </div>	<p data-bbox="794 1144 1139 1249">第 0 秒， 啟動無人機與深度鏡頭。</p>



上為無人機真實視野  
下為障礙物辨識結果



第 0.4 秒，  
障礙物和無人機開始移動，  
並透過辨識流程獲得障礙物輪廓。  
記錄其 500cm 處輪廓與真實畫面。  
障礙物位置：

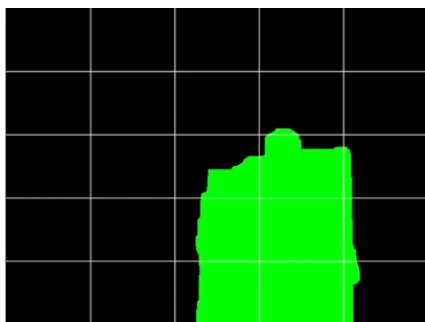
x: 0.52 m

y: -0.46 m

z: 5.23 m



上為無人機真實視野  
下為障礙物辨識結果



第 3.3 秒，  
持續移動且紀錄資料，  
直到距離 50cm 處，  
紀錄最後一筆輪廓與真實畫面。

障礙物位置：

x: -0.12 m

y: -0.31 m

z: 0.48 m

動態障礙物辨識實驗結束。

從圖 5-8 與圖 5-10 可觀察障礙物在動態接近過程中的空間座標變化情形。X 軸座標由約 0.6 公尺逐步下降至 -0.08 公尺，反映出障礙物自無人機右側進入視野並逐漸接近中心位置的軌跡。此種橫向位移變化，與障礙物移動方向一致，證實系統能正確追蹤其側向接近過程中的空間變化，

並即時更新定位結果。另一方面，Z 軸深度變化則展現出穩定且一致的遞減趨勢，從約 5.8 公尺逐漸下降至 0.5 公尺，顯示障礙物與無人機間的距離持續縮短。

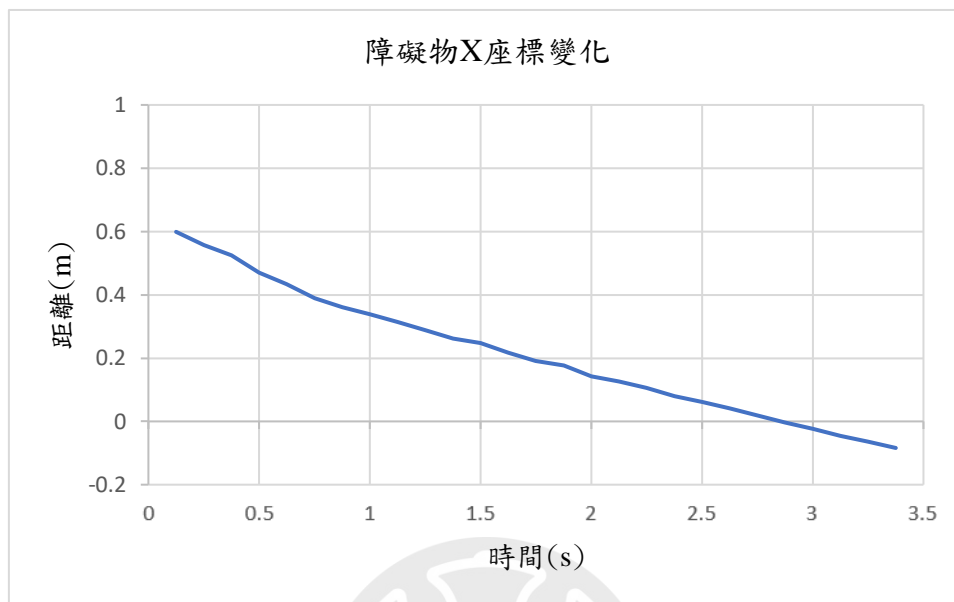


圖 5-8 障礙物 X 軸變化圖

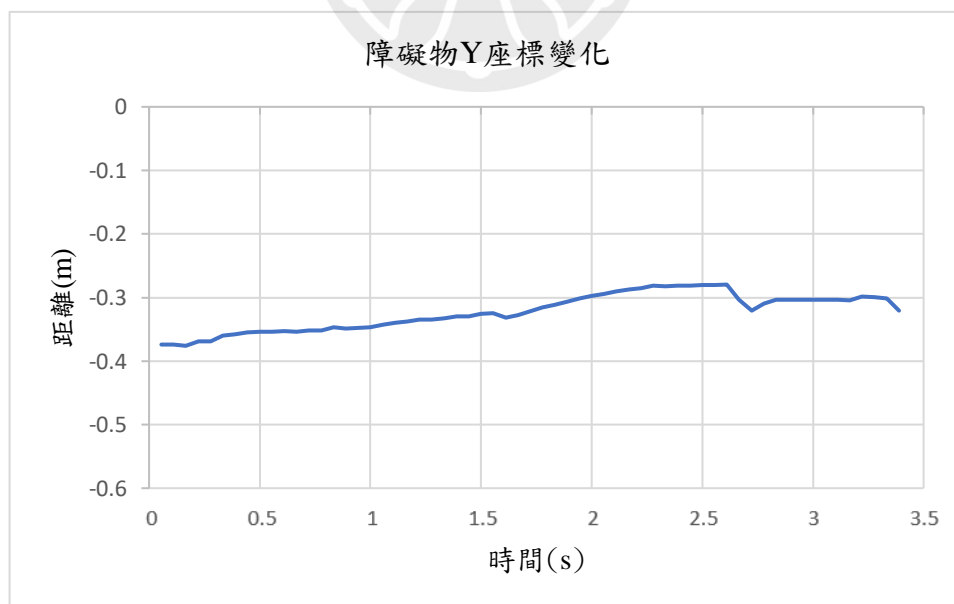


圖 5-9 障礙物 Y 軸變化圖

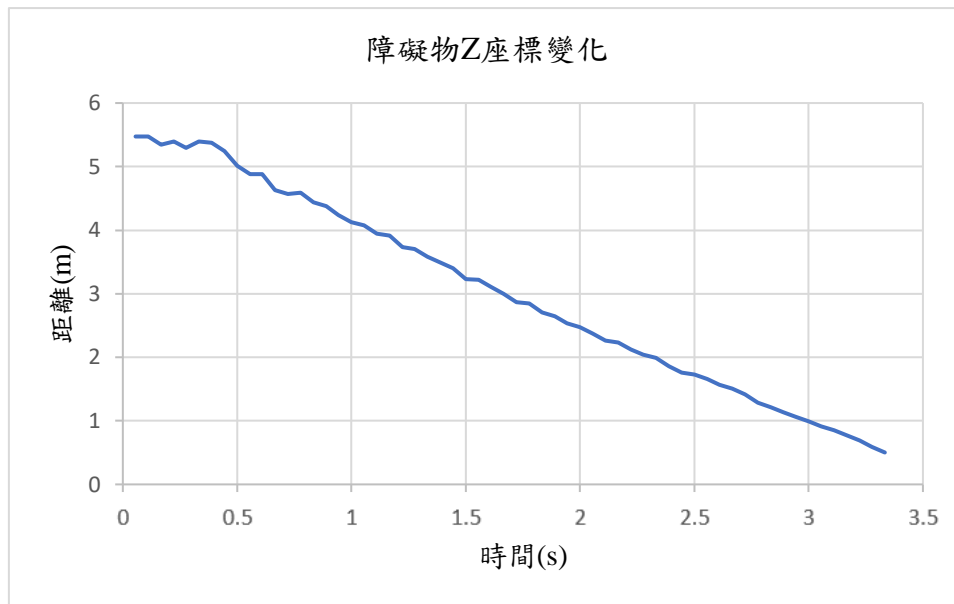


圖 5-10 障礙物 Z 軸變化圖

從表 5-4 和表 5-5 所呈現之結果可觀察到，中速移動下輪廓覆蓋率表現最為穩定且整體平均值最高。以大型障礙物為例，當距離至 300 cm 以內時，覆蓋率可達 90% 以上，小型障礙物亦有超過 85% 的表現。此顯示在中速下，動態物體所產生的光流向量大小適中、方向穩定，明顯區分於背景，利於系統進行有效補償與清晰聚類。

相對而言，當障礙物速度過高時，由於 Farneback 光流法在高速運動條件下常出現光流向量估計錯誤、細節丟失等現象。這些情況將導致光流場品質下降，使後續的類聚輪廓出現偏移甚至偵測失敗，進而造成輪廓覆蓋率顯著下降。此外，在低速中，儘管系統運算上較為穩定，但由於障礙物產生的光流向量過小，與背景向量接近，容易被誤判為靜態背景，使得類聚後的遮蔽輪廓不完整，影響覆蓋率表現。綜上所述，本系統在中等速度下具有最佳的動態障礙物辨識效能。

表 5-4 大型動態障礙物辨識實驗數據

移動模式 距離位置(cm)	平均輪廓覆蓋率(%)		
	低速	中速	快速
500	78.2	83.0	74.2
450	79.8	84.6	75.9
400	81.5	86.2	77.6
350	82.4	88.7	79.8
300	83.3	91.2	82.3
250	85.5	93.1	84.7
200	87.2	95.0	87.2
150	88.7	95.8	88.5
100	90.3	96.2	89.7
50	91.8	97.0	90.2

表 5-5 小型動態障礙物辨識實驗數據

		平均輪廓覆蓋率(%)		
		低速	中速	快速
距離位置(cm)	移動模式			
	500		69.9	75.0
450		72.7	77.3	63.7
400		75.5	79.5	65.8
350		77.2	82.7	68.0
300		78.8	85.8	70.3
250		79.9	86.9	72.1
200		81.0	88.0	74.2
150		82.8	90.4	76.5
100		84.8	92.8	78.8
50		86.5	94.2	80.3

## 5.2 無人機避障實驗

本節在說明本研究設計之無人機避障實驗流程，該流程適用於後續靜態障礙物與動態障礙物之避障測試。

實驗流程如圖 5-11，首先，無人機與深度鏡頭啟動後開始移動，並啟用障礙物辨識流程。系統擷取出障礙物輪廓，透過畫面切割和遮蔽塊定位獲得障礙物在影像中的位置區域。接著透過 BFS 演算法分析，獲得空曠區域重心位置。再透過反投影技術將這些位置轉換為實際空間座標，獲得三維導引向量，系統以此導引向量，指引無人機朝向空曠區域偏移，完成避障決策與動作。

靜態與動態障礙物實驗皆遵循上述流程，其主要差異僅在於障礙物是否具移動性，詳細情境將分別於 5.2.1 與 5.2.2 節中說明。

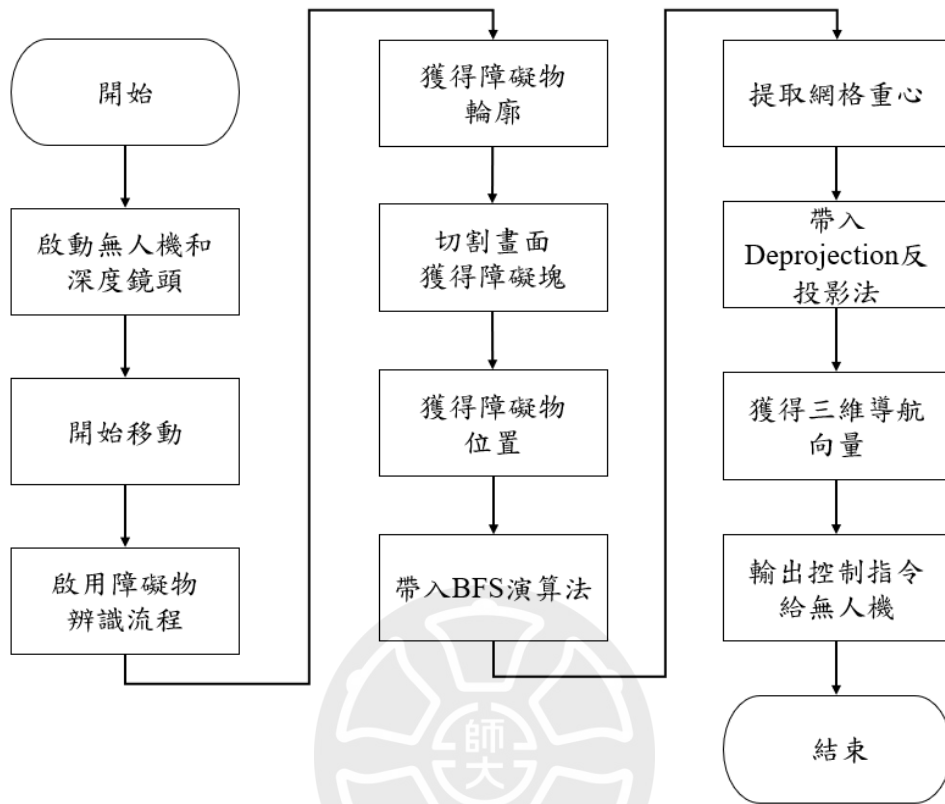


圖 5-11 避障實驗流程圖

本次避障實驗的主要目標為評估本系統於不同情境下的避障成功率與所需的最小緩衝距離，探討其穩定性與實用性。為達此目的，本研究規劃兩個面向進行量化分析。首先，針對最小避障距離分析，實驗設計讓無人機在障礙物位置固定的條件下，從不同起始距離出發，執行完整的避障流程，並統計各距離下的避障成功率，藉此推估本系統得以穩定完成避障行為所需的最短反應距離。

其次，進行避障成功率分析，透過變動起始距離與障礙物尺寸等條件，重複測試避障行為並紀錄成功與失敗的比例，作為系統整體避障能力、決策穩定性與適應場景變化能力的衡量指標以評，估本研究所提出導航策略之實用性與穩定性。

## 5.2.1 靜態障礙物避障實驗

本實驗旨在驗證本系統於靜態障礙物環境中的避障能力與反應極限，藉以推估最小安全緩衝距離，並觀察障礙物尺寸變化對避障成效之影響。實驗場景如圖 5-12，無人機面對固定於前方的靜態障礙物，障礙物設置於距離 500 公分處，並依序調整其大小，包括「小型」與「大型」兩種尺寸，模擬不同障礙物尺寸的避障情境。在每一組障礙物尺寸條件下，無人機將從多個起始距離出發執行避障任務，透過辨識系統與導引向量的生成進行即時避障。每個起始距離條件下皆重複進行 25 次實驗，以統計該距離之避障成功率，進而推估在不同障礙條件下，系統所需的最小緩衝距離。

本實驗透過「起始距離」與「障礙物尺寸」兩項變因，分析本系統在靜態障礙環境下的反應效能與適應能力。最終得出在不同障礙程度下系統可穩定避障的安全距離區間，作為日後應用於複雜環境時之參考依據。

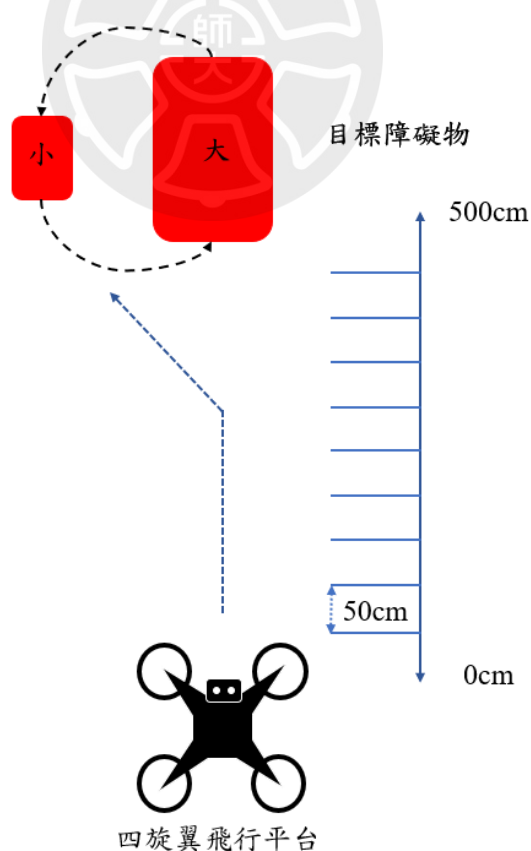

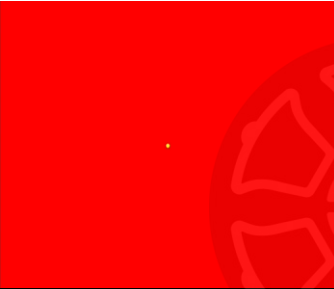
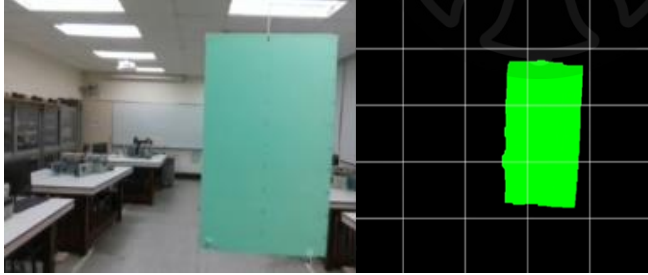
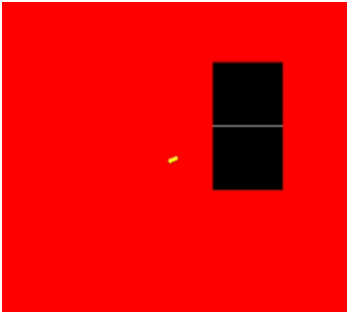

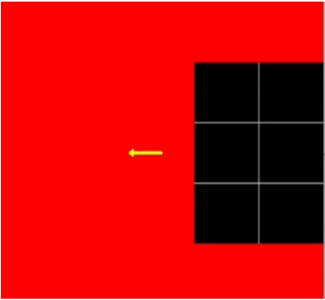

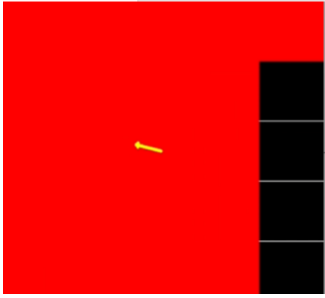


圖 5-12 靜態障礙物避障實驗示意圖

表 5-6 中顯示無人機於不同時間點之辨識結果、避障決策及空間定位數據，能有效觀察障礙物位置變化對系統導引的影響。

表 5-6 靜態障礙物避障實驗結果

實驗圖	動作說明
 <p>上為無人機真實視野和障礙物辨識結果</p> <p>下為空曠區域與導引向量</p> 	<p>第 0 秒， 啟動無人機與深度鏡頭。</p>
 <p>上為無人機真實視野和障礙物辨識結果</p> <p>下為空曠區域與導引向量</p> 	<p>第 0.53 秒， 無人機開始移動， 啟用障礙物辨識流程， 取得障礙物輪廓， 障礙物位置： x: 0.13 m y: -0.28 m z: 4.22 m 帶入避障路徑規劃， 獲得導航向量， 輸出給無人機開始偏移。</p>

 <p>上為無人機真實視野和障礙物辨識結果</p> <p>下為空曠區域與導引向量</p> 	<p>第 2.3 秒， 無人機持續避障， 隨著不斷接近障礙物， 獲得更明顯的輪廓， 障礙物位置： x: 0.45 m y: -0.36 m z: 1.72 m 使偏移力道越發強烈。</p>
 <p>上為無人機真實視野和障礙物辨識結果</p> <p>下為空曠區域與導引向量</p> 	<p>第 3.1 秒， 無人機持續避障， 直到通過障礙物， 障礙物位置： x: 0.61 m y: -0.37 m z: 0.95 m 靜態障礙物避障實驗結束。</p>

根據圖 5-13 至圖 5-15 所示之障礙物座標變化圖，可進一步分析無人機於靜態障礙環境中之感知與避障行為。實驗初期，障礙物位於無人機中央附近，隨著無人機啟動避障策略，機體逐漸朝左上方移動，導致障礙物於影像中呈現出向右下方遠離的相對運動。此變化反映在座標變化上，

X 軸座標由約 0.1 公尺上升至 0.65 公尺，顯示障礙物於影像中逐漸偏向右側；Y 軸座標則由 -0.2 公尺下降至約 -0.38 公尺，顯示障礙物向畫面下方位移。Z 軸部分，座標自約 5.1 公尺下降至 0.65 公尺，顯示無人機與障礙物之間的實際距離持續縮短，直到系統完成避障。上述變化趨勢與無人機實際飛行路徑相符，驗證本系統能有效感知障礙物的空間相對位置，並於靜態障礙物情境中及時作出適當避障反應。

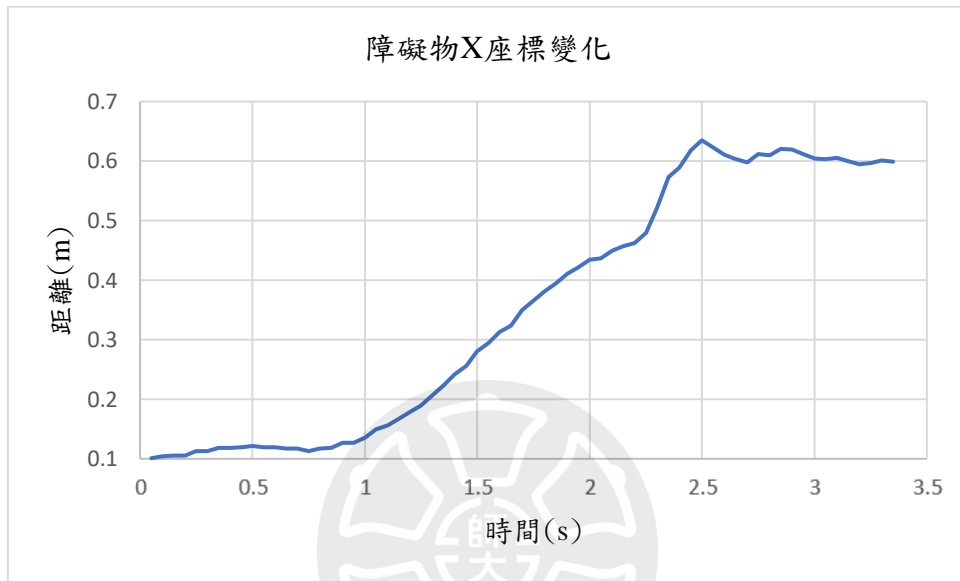


圖 5-13 障礙物 X 軸變化圖

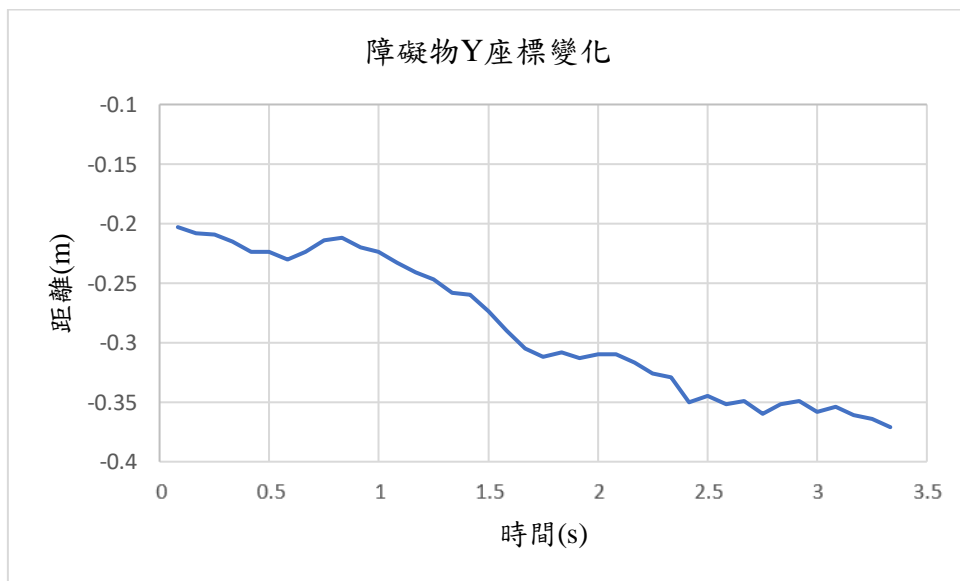


圖 5-14 障礙物 Y 軸變化圖

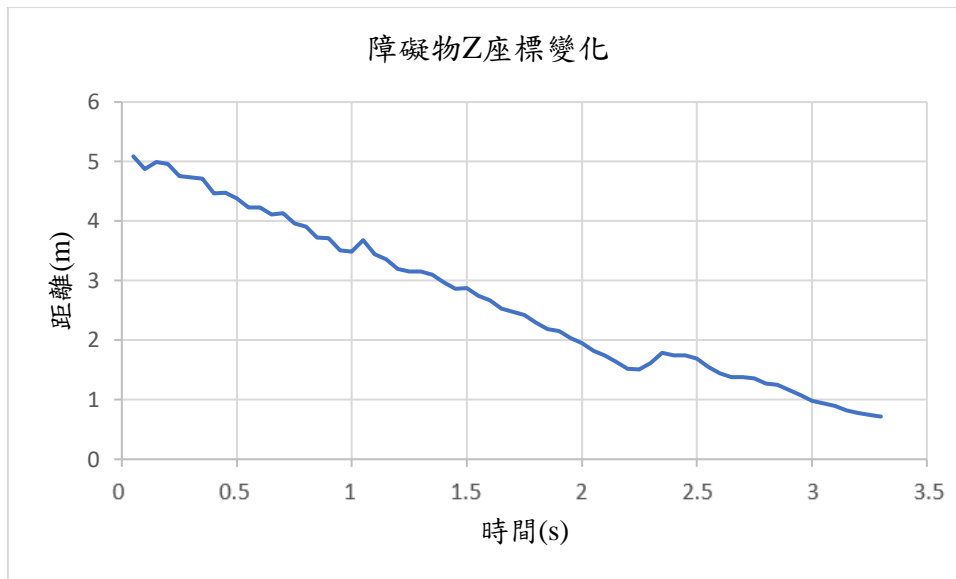


圖 5-15 障礙物 Z 軸變化圖

從表 5-7 實驗數據中可觀察，無人機在進行靜態障礙物避障時，其成功率會隨起始距離縮短而逐步下降，且障礙物尺寸越大，對系統避障的挑戰越高。

在小型障礙物條件下，無人機於 350 公分以上的起始距離皆能穩定完成避障任務，成功率維持在 100%；當起始距離縮短至 300 公分後，成功率略為下降至 96%，顯示此時系統仍具備相當的即時辨識與反應能力。至 100 公分以下，避障成功率快速下滑，特別是 50 公分時僅有 24%，顯示此時系統的反應時間與可用轉向空間嚴重受限，已接近反應極限。

在大型障礙物條件下，儘管於 450 公分起始距離仍可維持 100% 成功率，但在 400 公分開始出現下降，至 200 公分時已降至 80%。由於大型障礙物佔據畫面視角較多，使得可供偏移的空間變小，進一步影響避障動作的成功與即時性。當起始距離僅剩 150 公分時，成功率驟降至 24%，而在 50 公分時更僅剩 8%，顯示此距離下已幾乎無法完成避障動作。

表 5-7 靜態避障實驗數據

	起始位置(cm)	避障成功率(%)	
		大型障礙物	小型障礙物
靜態障礙物避障實驗	500	100.0	100.0
	450	100.0	100.0
	400	96.0	100.0
	350	88.0	100.0
	300	92.0	96.0
	250	92.0	88.0
	200	80.0	92.0
	150	24.0	76.0
	100	24.0	36.0
	50	8.0	24.0

根據靜態障礙物辨識實驗結果可知，當障礙物距離無人機小於 400 公分時，系統對於大型與小型障礙物皆能維持平均 80% 以上之輪廓覆蓋率，顯示其具備良好的辨識能力，視為本系統可穩定執行避障任務的門檻。

綜合上述兩項實驗數據結果推論，當障礙物距離位於 400 公分至 200 公分之間時，系統具備足夠之辨識精度與反應空間，可穩定產生導引向量並完成避障動作。該距離區間定義為本研究系統於靜態障礙物環境中進行穩定避障操作之「安全距離區間」。

## 5.2.2 動態障礙物避障實驗

為驗證系統在現實動態環境中的避障能力，本實驗設計模擬障礙物於無人機視野中移動之情境，評估系統面對相對運動下障礙物遮蔽的即時偵測與避障反應能力。相較於先前靜態障礙物實驗，本實驗更貼近實際應用場景中常見的行人、自行車等移動目標所構成的風險情境。

如圖 5-16，障礙物分為「大型」與「小型」兩種，於側面向無人機方向移動，障礙物移動速度分為三種常見人類移動模式，為漫步（0.5 至 0.8 m/s）、快走（1.0 至 1.2 m/s）與跑步（1.8 至 2.1 m/s），分別對應低速、中速與高速移動體。實驗中，無人機以固定速度穩定向前飛行，系統持續進行即時感測與避障判斷。

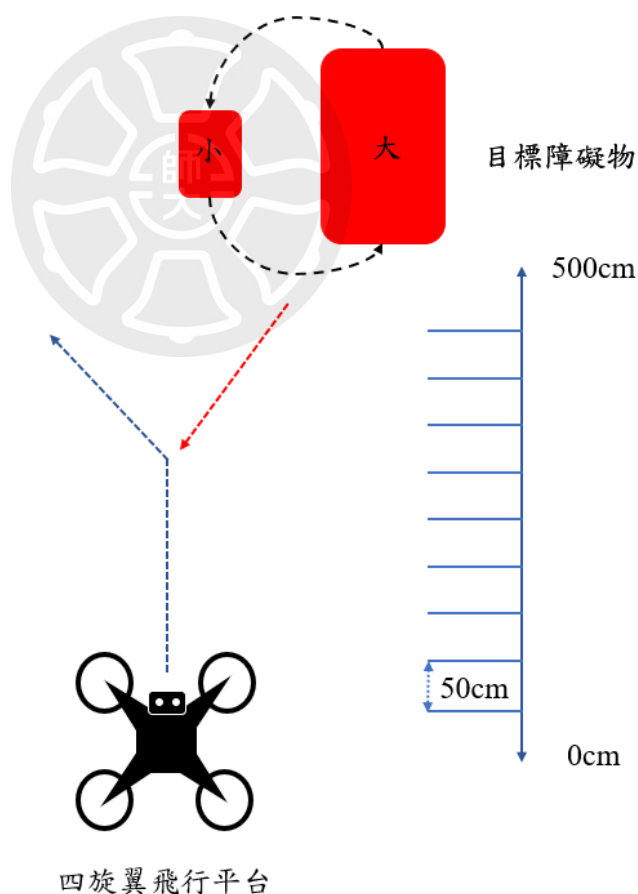


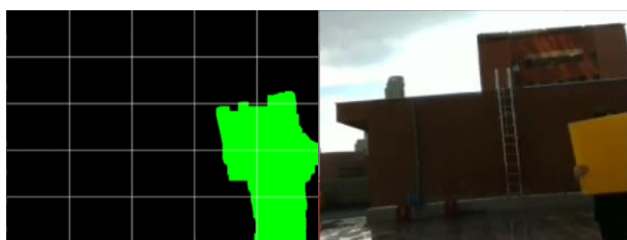
圖 5-16 動態障礙物避障實驗示意圖

為評估系統的最小安全避障距離，每當障礙物距離無人機達設定門檻時，系統將執行避障決策流程。若障礙物尚未進入該距離門檻，則系統不啟動避障，以此方式逐步縮短觸發距離，以觀察不同速度與障礙條件下，系統最低的緩衝距離。本實驗重複 25 次測試於各速度條件下之避障反應，記錄障礙物辨識畫面與避障是否完成，統計各條件下避障成功率，並據此分析系統在動態環境中之安全距離。

表 5-8 中顯示無人機於不同時間點之辨識結果、避障決策及空間定位數據，能有效觀察障礙物位置變化對系統導引的影響。

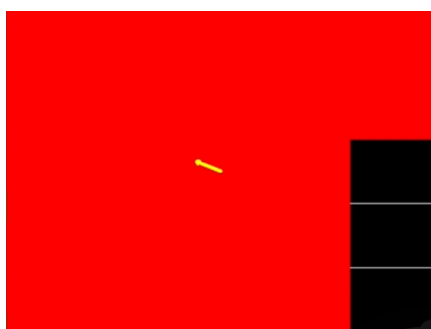
表 5-8 動態障礙物避障實驗結果

實驗圖	動作說明
<div data-bbox="256 1077 906 1323" data-label="Image"> </div> <p data-bbox="309 1361 855 1397">上為無人機真實視野和障礙物辨識結果</p> <p data-bbox="405 1435 759 1471">下為空曠區域與導引向量</p> <div data-bbox="376 1503 786 1809" data-label="Image"> </div>	<p data-bbox="940 1290 1283 1391">第 0 秒， 啟動無人機與深度鏡頭。</p>



上為無人機真實視野和障礙物辨識結果

下為空曠區域與導引向量

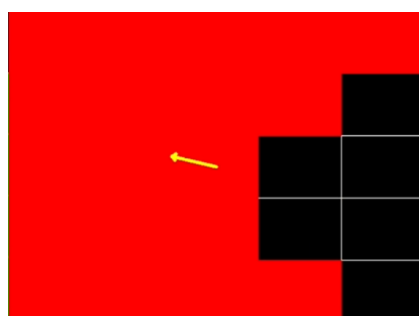


第 0.52 秒，  
無人機開始移動，  
啟用障礙物辨識流程，  
取得障礙物輪廓，  
障礙物位置：  
x: 0.55 m  
y: -0.41 m  
z: 4.25 m  
計算空曠區重心，  
無人機開始些微偏移。

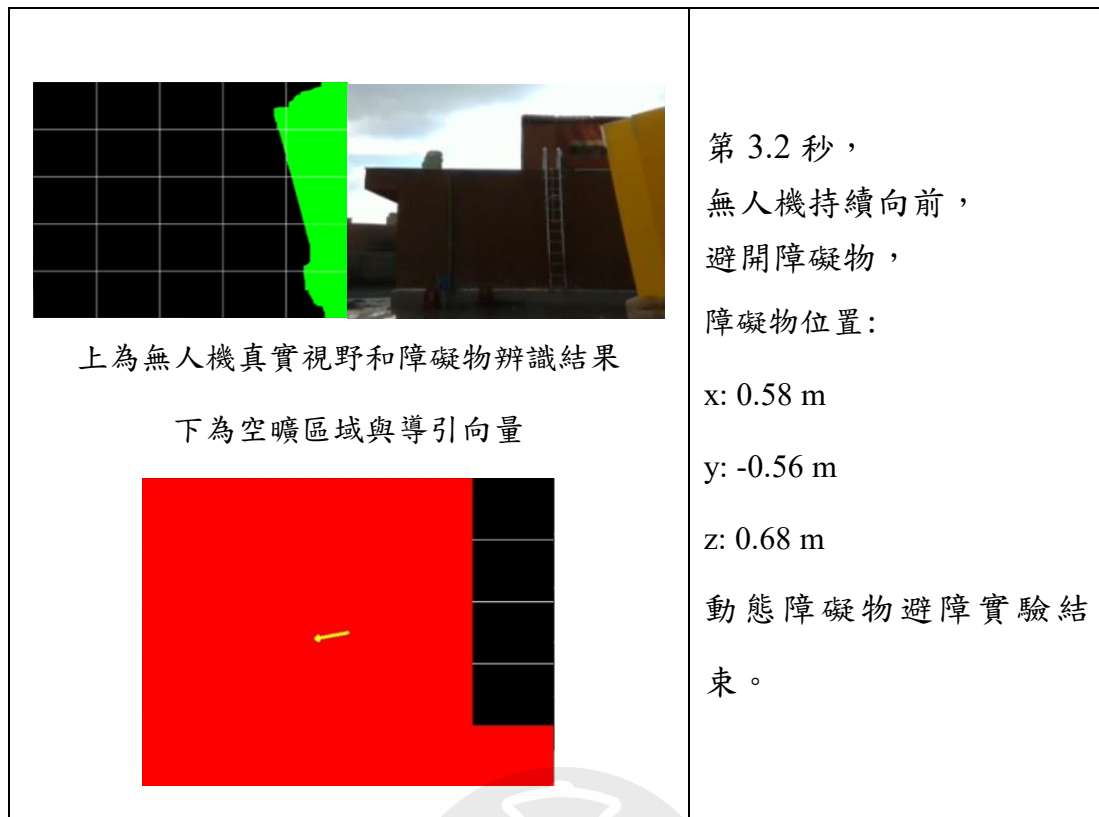


上為無人機真實視野和障礙物辨識結果

下為空曠區域與導引向量



第 1.8 秒，  
無人機持續向前，  
隨著障礙物靠近，  
獲得更明顯的輪廓，  
障礙物位置：  
x: 0.17 m  
y: -0.48 m  
z: 2.38 m  
計算空曠區重心，  
無人機開始大幅偏移。



根據圖 5-17 至圖 5-19 所示之障礙物座標變化圖，可進一步分析無人機於動態障礙環境中之感知與避障行為。實驗中，無人機持續向前移動，而障礙物自右側斜向接近畫面，最初出現在影像右下角，隨著雙方接近，障礙物逐漸靠近畫面中心，接著觸發避障機制，使無人機自左上方進行閃避。

在座標變化上，X 軸初始值約為 0.75 公尺，代表障礙物位於視野右側，接著下降至約 0.17 公尺，顯示障礙物逐步向畫面中央靠近，最終再度上升至 0.7 公尺，反映無人機成功由左側避開後，障礙物重新偏移至右側視野邊緣。Y 軸則呈現下降的趨勢，說明無人機於避障過程中採取向上方的迴避動作，使障礙物在畫面中位置逐漸下移。Z 軸距離則從 5.09 公尺下降至 0.3 公尺，顯示無人機與障礙物之間的實際距離持續縮短，直到系統完成避障。上述變化趨勢與無人機實際飛行路徑相符，驗證本系統能有效感知障礙物的空間相對位置，並於動態障礙物情境中及時作出適當避障反應。

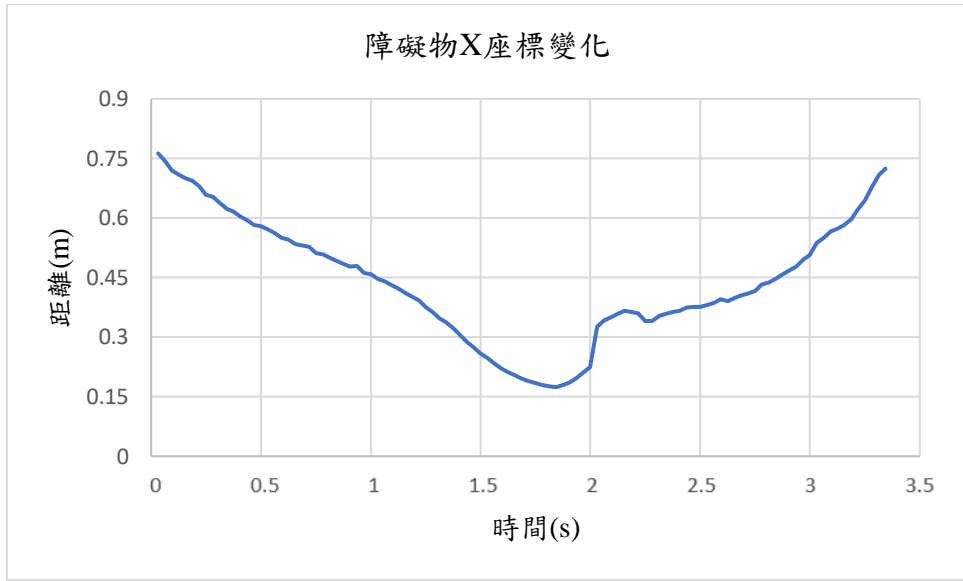


圖 5-17 障礙物 X 軸變化圖

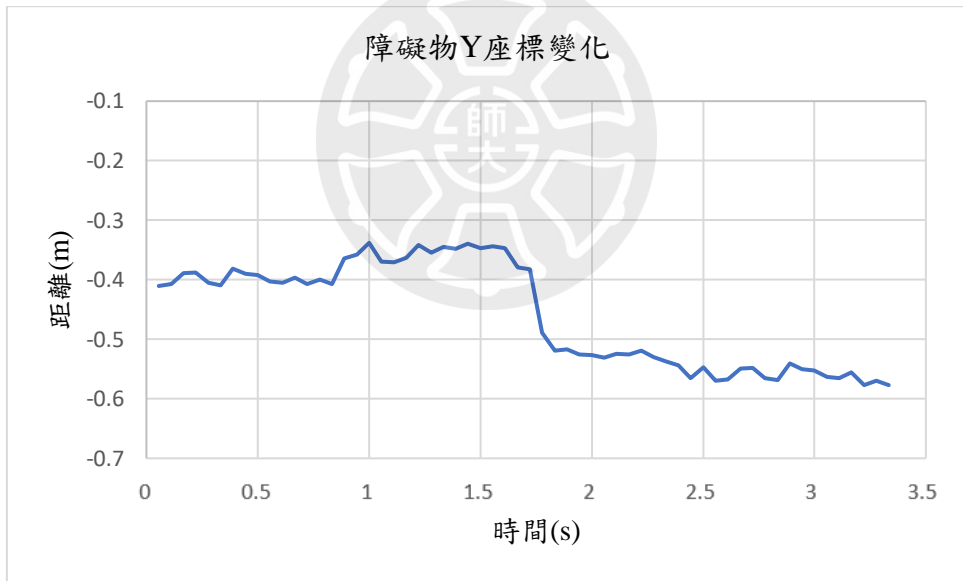


圖 5-18 障礙物 Y 軸變化圖

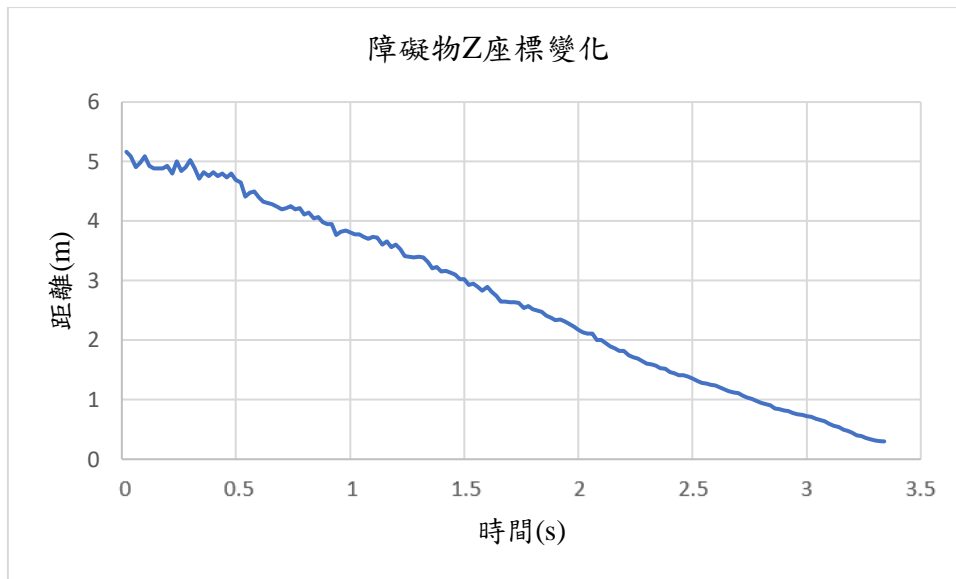


圖 5-19 障礙物 Z 軸變化圖

根據表 5-9 與表 5-10 數據顯示，當障礙物與無人機的距離愈短，系統可用以偵測、判斷與執行避障動作的反應時間亦隨之縮短，導致避障成功率明顯下降。在障礙物與無人機距離為 150 cm 時，避障成功率普遍低於 40%，反映出此距離已接近系統的反應極限。

此外，障礙物速度亦對避障成效產生顯著影響。高速移動的障礙物將大幅壓縮系統處理與反應的時間，導致避障表現下降。大型障礙物於 300 cm 距離與 180 cm/s 速度條件下的避障成功率僅為 76.0%，顯著低於同距離條件下 30 cm/s 速度的成功率 92.0%。

由此可見，障礙物接近速度越快，系統越需提早啟動避障流程以確保安全。針對障礙物尺寸的影響，實驗數據指出小型障礙物在近距離條件下的避障成功率略高於大型障礙物。在 100 cm 距離下，小型障礙物於 30 cm/s 的成功率為 36.0%，相較於大型障礙物的 32.0% 略高。此現象可能與面積較小有關，讓系統能更順利完成導引調整。然而，隨著障礙物速度提升，此優勢逐漸消失。

表 5-9 大型動態避障實驗數據

		避障成功率(%)		
		低速	中速	高速
距離位置(cm)	移動模式			
	500		100.0	100.0
450		100.0	100.0	96.0
400		100.0	96.0	92.0
350		92.0	84.0	80.0
300		92.0	88.0	76.0
250		76.0	68.0	48.0
200		72.0	68.0	44.0
150		36.0	44.0	20.0
100		32.0	40.0	16.0
50		24.0	24.0	8.0

表 5-10 小型動態避障實驗數據

		避障成功率(%)		
		低速	中速	高速
移動模式	距離位置(cm)			
		500	100.0	100.0
	450	100.0	100.0	100.0
	400	100.0	96.0	96.0
	350	100.0	84.0	80.0
	300	92.0	88.0	80.0
	250	84.0	68.0	60.0
	200	84.0	68.0	56.0
	150	40.0	36.0	32.0
	100	36.0	32.0	24.0
	50	28.0	24.0	16.0

綜合動態障礙物辨識實驗結果，當障礙物距離位於 400 公分至 250 公分之間時，系統具備足夠之辨識精度與反應空間，可穩定產生導引向量並完成避障動作。該距離區間可定義為本研究系統於動態障礙物環境中進行穩定避障操作之「安全距離區間」。

本研究結果證實該系統具備處理動態障礙物的能力，唯在高速且近距離的極端條件下存在避障成功率下降之風險。

### 5.3 自主導航實驗

為驗證系統整合視覺感知、障礙物辨識與即時導航導引功能的整體效能，本節設計自主導航實驗情境，無人機於真實環境中自主飛行並完成目

標導航任務之能力。實驗目標為觀察系統是否能在不依賴人工遙控的情況下，自主規劃可行路徑、避開障礙物，並準確抵達終點。

在本實驗中，導航區域以 Google Map 所提供的三維地圖作為環境建模基礎，結合 RRT\* (Rapidly-exploring Random Tree Star) 演算法進行靜態路徑規劃。使用者可於任意地點手動設定起點與終點，系統將據此生成一條避開靜態障礙物的初始航路，包含多個連續航點組成的飛行路徑，導航區域內可包含數個靜態障礙物與動態障礙物。系統會結合障礙辨識流程與動態路徑規劃，進行障礙物辨識與避障。

在實際導航過程中，如圖 5-20，系統首先計算當前位置與目標路徑航點之間的差距，推導出前進方向與控制指令。此時系統會同步啟用障礙物辨識流程，若辨識到潛在障礙物，則觸發即時避障模組，並產生一條新的動態避障路徑以繞行障礙。若偵測無障礙，則依原路徑持續前進，當無人機成功抵達目標位置時，導航任務結束。

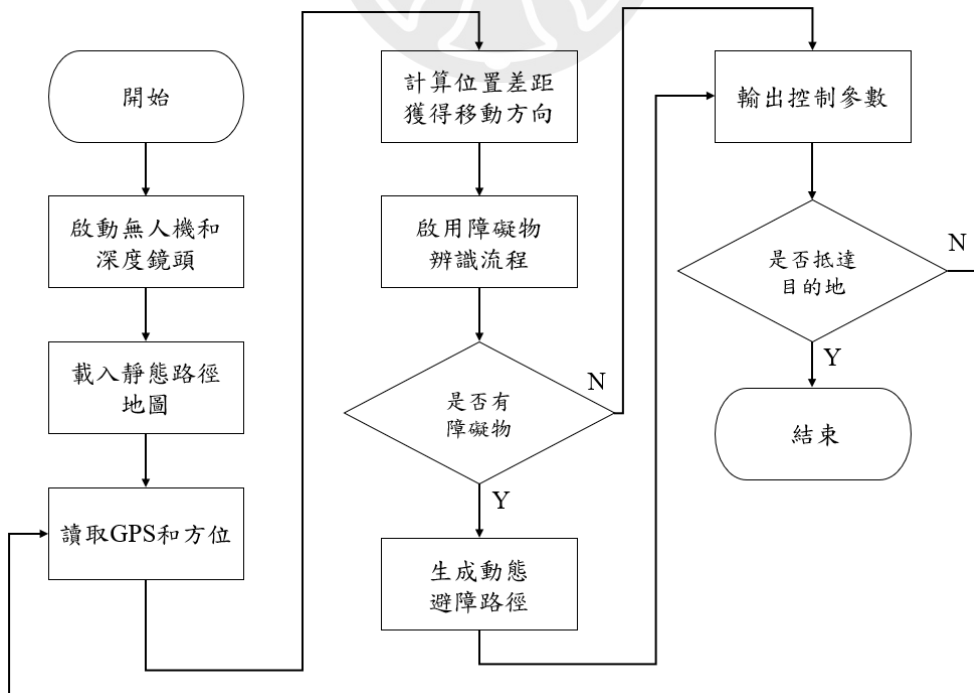


圖 5-20 自主導航實驗流程圖

圖 5-21 為透過 Google Maps 所提供的三維建築物輪廓資料為此次的實驗場地。圖 5-22 為帶入 RRT\*演算法後所得的靜態路徑圖。圖 5-23 顯示無人機於實地飛行過程中的實際航跡，並與規劃路徑進行對照。圖中紅色路線代表靜態路徑規劃所產生的路徑，而黃色節點則為無人機透過 GPS 整合定位後，實際執行導航所經過的位置點。實際飛行路線大致貼合原始規劃結果，唯有靠近建築物或障礙物時，路徑會出現偏移，此為系統動態路徑規劃於飛行途中修正路徑所導致。

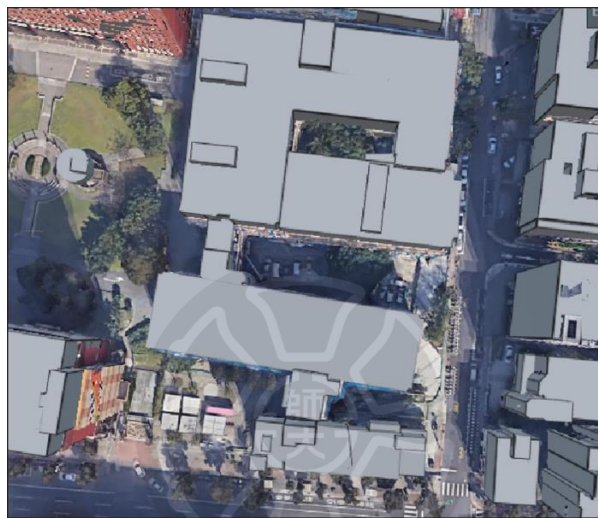


圖 5-21 建築物輪廓圖



圖 5-22 靜態路徑規劃圖



圖 5-23 無人機實際路線圖

圖 5-24 可觀察到無人機在自主導航過程中，其位置與導航路線之間的距離變化呈現一定的趨勢與波動。起始階段距離大致維持在約 3 公尺左右，並於 5 至 10 秒之間出現一波明顯的上升波動，推測此時系統偵測到潛在障礙物，觸發避障機制，導致無人機短暫偏離原導航路徑。接續在 13 至 20 秒之間再次出現距離上升的情形，同樣反映第二次避障事件的發生。兩次避障反應皆顯示系統能即時調整飛行路徑以繞行障礙，維持飛行安全。之後於 30 秒後，距離逐漸趨於穩定，並大致維持於 0.5 至 1.5 公尺範圍內，主因為導航路線鄰近建築物結構，限制無人機的空間，使其貼近原始路徑飛行直至任務結束，

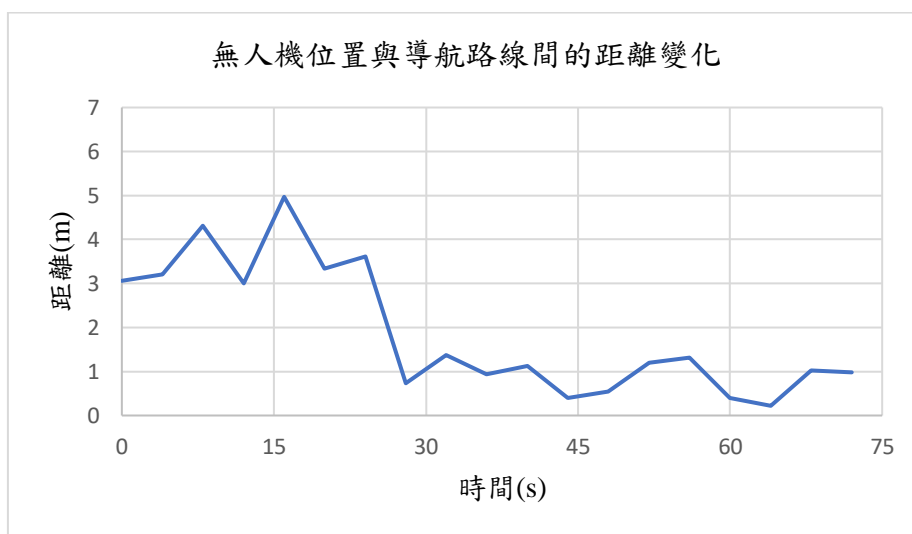


圖 5-24 無人機位置與導航路線間的距離變化圖

圖 5-25 至圖 5-29 展示無人機於飛行過程中實際遭遇障礙物時，系統即時辨識並進行避障的動態流程。圖 5-25 表示發生避障事件的位置。下方四張圖為飛行過程中不同時間點的畫面與辨識結果，左側為無人機的實際畫面，右側為系統所辨識之空曠區域與導引向量視覺化結果，中間為障礙物辨識結果。

在圖 5-26 中，無人機尚未辨識任何障礙物，整體視野空曠，系統未偵測到明顯障礙區塊，中間黑色網格畫面中無障礙物輪廓，無需修正航向。在圖 5-27 中，障礙物進入視野右側，系統透過辨識流程辨識出一群像素，系統計算空曠區分布，透過空曠區重心，產生導引向量向左偏移，指示無人機調整航向避開障礙物。中間黑色網格畫面中，可見右方開始出現綠色障礙物輪廓，導引向量已有些微向左偏移。在圖 5-28 中，障礙物輪廓擴大。此時，右側圖中導引向量清晰指向右方。在圖 5-29 中，無人機依據導引向量進行側向位移，順利通過遮蔽區範圍。畫面中可見障礙物已脫離視線，綠色輪廓退至右側，導引向量逐漸回復至正前方，系統重新恢復原始導航路徑。



圖 5-25 發生避障事件之位置(25°01'37.6"N 121°31'48.4"E)



圖 5-26 飛行階段

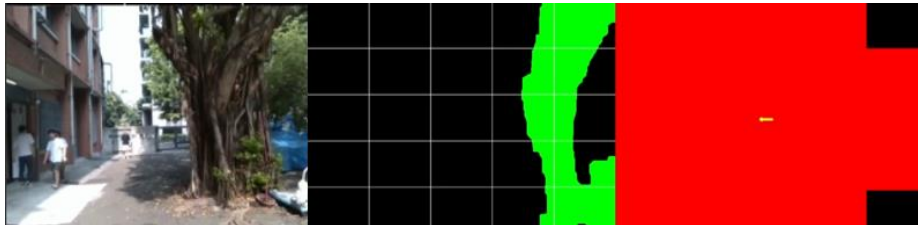


圖 5-27 障礙物初現階段

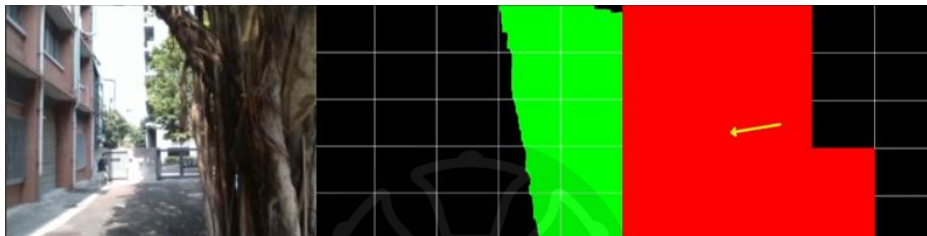


圖 5-28 避障決策階段

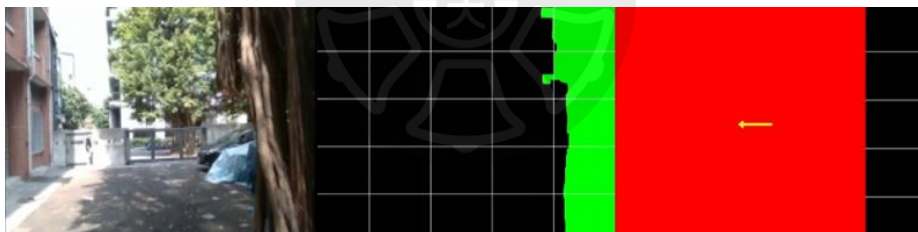


圖 5-29 避障完成階段

如圖 5-30 至圖 5-32 所示，顯示自主導航過程中，無人機面對右側障礙物時，其觀測到的障礙物座標隨時間變化的情形。在實際過程中，障礙物自無人機右側進入視野，系統於偵測後觸發避障，導引無人機朝左下方進行避障。如圖 5-30 所示，X 軸部分，障礙物初期位於約 0.5 公尺，代表其位於畫面偏右側，隨後因持續接近而下降至約 0.3 公尺，避障後又上升至約 0.53 公尺，呈現一段先靠近後遠離的變化趨勢。圖 5-31 則顯示 Y 軸由 0.28 公尺上升至 0.39 公尺，主因為無人機向左下閃避障礙物，使得

障礙物於影像中呈現出向上偏移的相對運動。圖 5-32 則呈現明顯的下降趨勢，顯示障礙物由遠而近，直到避障完成後才逐漸脫離視野。

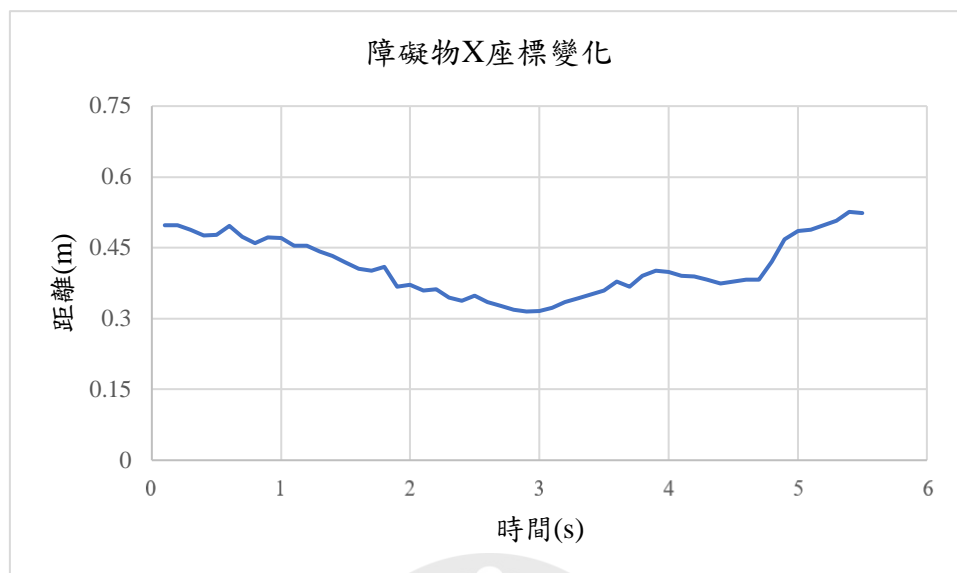


圖 5-30 障礙物 X 軸變化圖

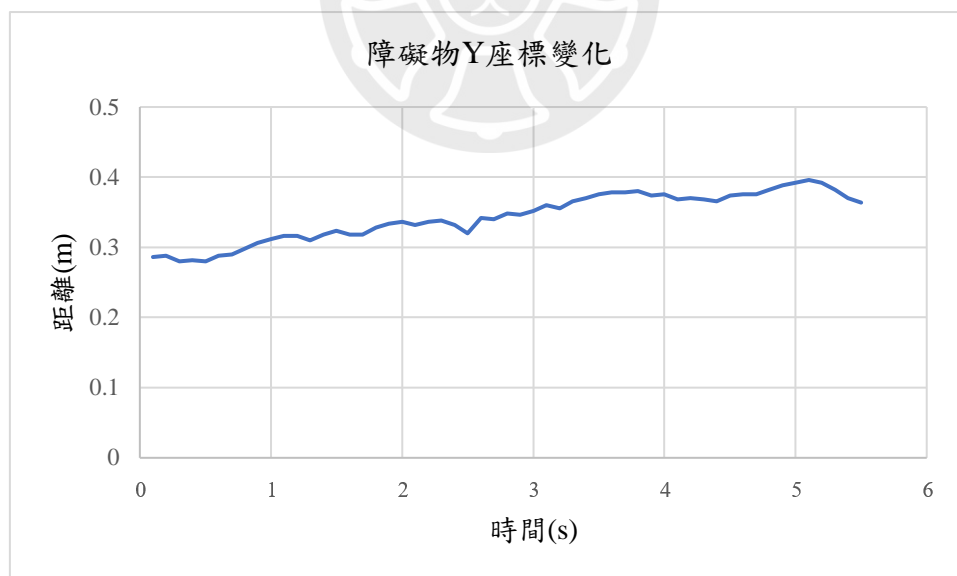


圖 5-31 障礙物 Y 軸變化圖

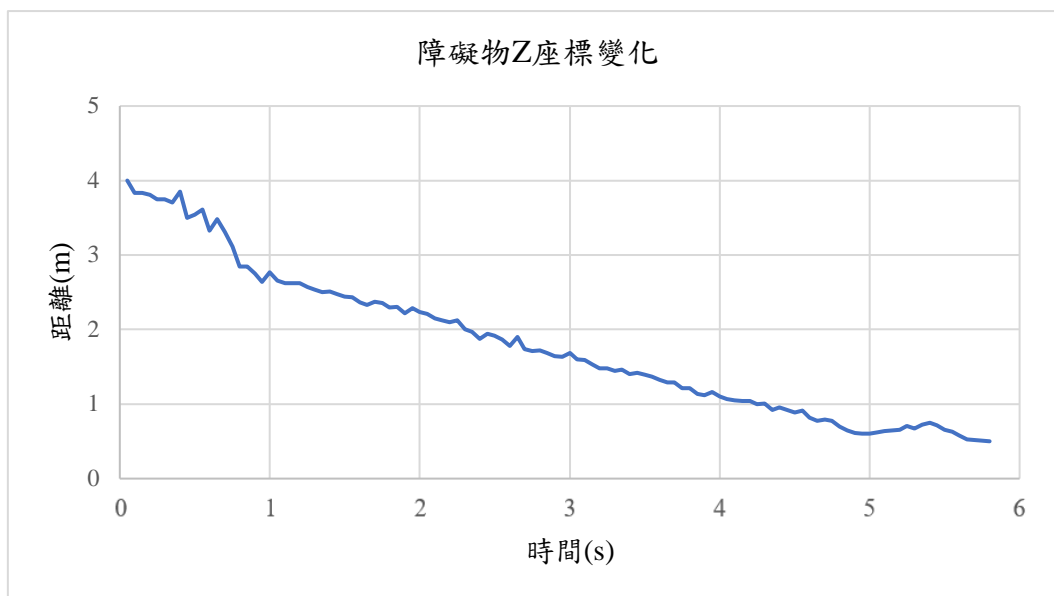


圖 5-32 障礙物 Z 軸變化圖

圖 5-33 為系統於飛行過程中遭遇來自下方障礙物的位置。此情境下的避障反應如圖 5-34 至圖 5-37 所示。整體處理流程與前述案例相似，當障礙物出現在畫面下方時，系統透過光流法快速辨識遮蔽輪廓，並計算出導引向量指向上方空曠區域。無人機依據此導引資訊即時向上偏移航向，成功繞行下方障礙物，最終恢復原始路徑。



圖 5-33 發生避障事件之位置(25°01'37.5"N 121°31'48.8"E)



圖 5-34 飛行階段

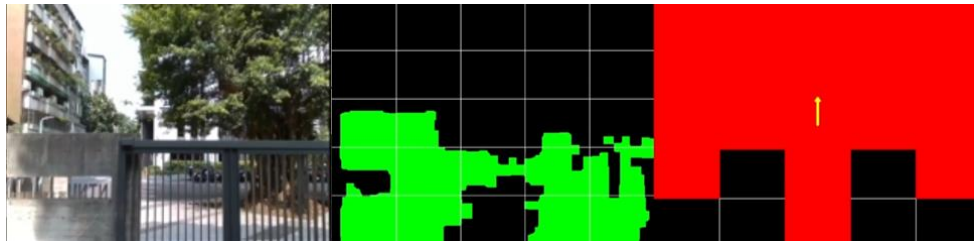


圖 5-35 障礙物初現階段

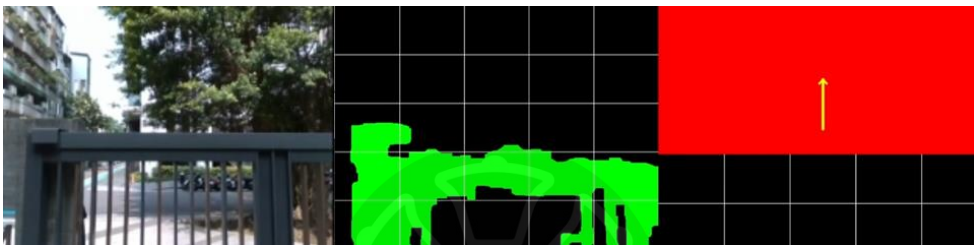


圖 5-36 避障決策階段

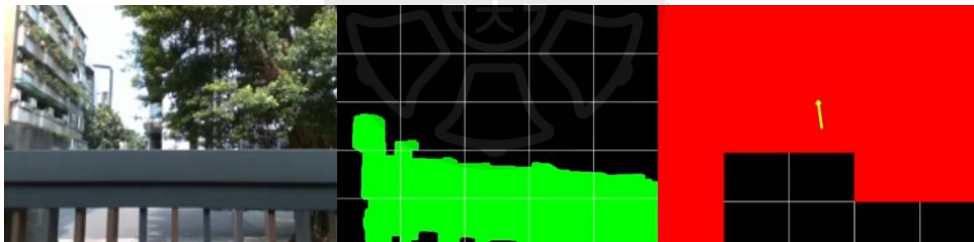


圖 5-37 避障完成階段

如圖 5-38 至圖 5-40 所示，顯示自主導航過程中，無人機面對下方障礙物時，其觀測到的障礙物座標隨時間變化的情形。在實際過程中，障礙物起初由畫面下方進入無人機視野，隨後無人機啟動避障策略並透過上升高度進行閃避，使障礙物在畫面中呈現出持續向下移動的相對運動。如圖 5-38 所示，X 軸的變化相對平穩。圖 5-39 則顯示 Y 軸由約 -0.28 公尺穩定下降至約 -0.48 公尺，呈現明顯的向下位移趨勢，對應於無人機上升閃避所造成的相對運動。圖 5-40 則呈現明顯的下降趨勢，顯示障礙物由遠而近，直到避障完成後才逐漸脫離視野。

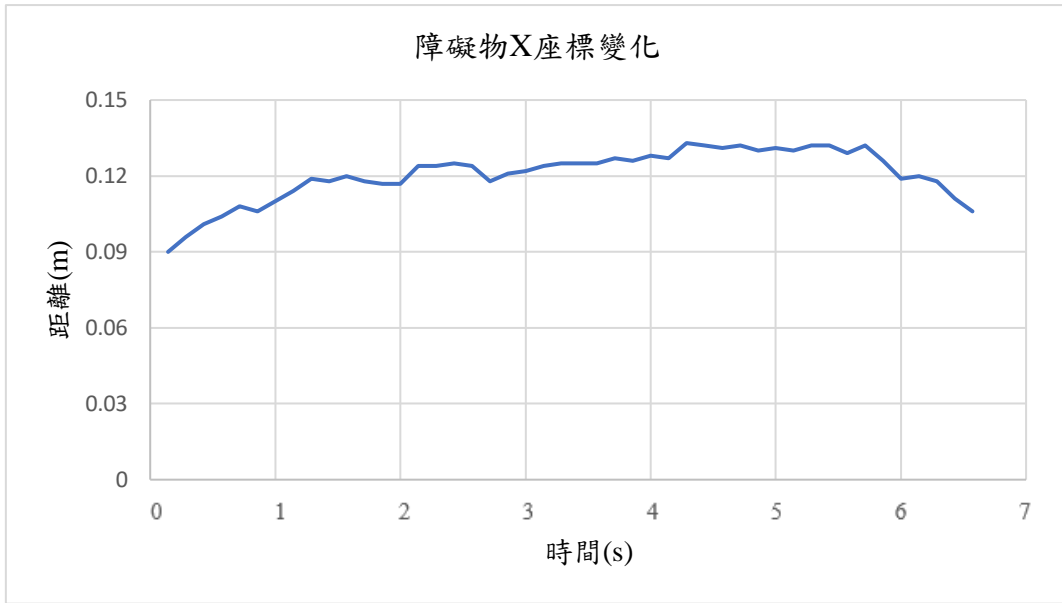


圖 5-38 障礙物 X 軸變化圖

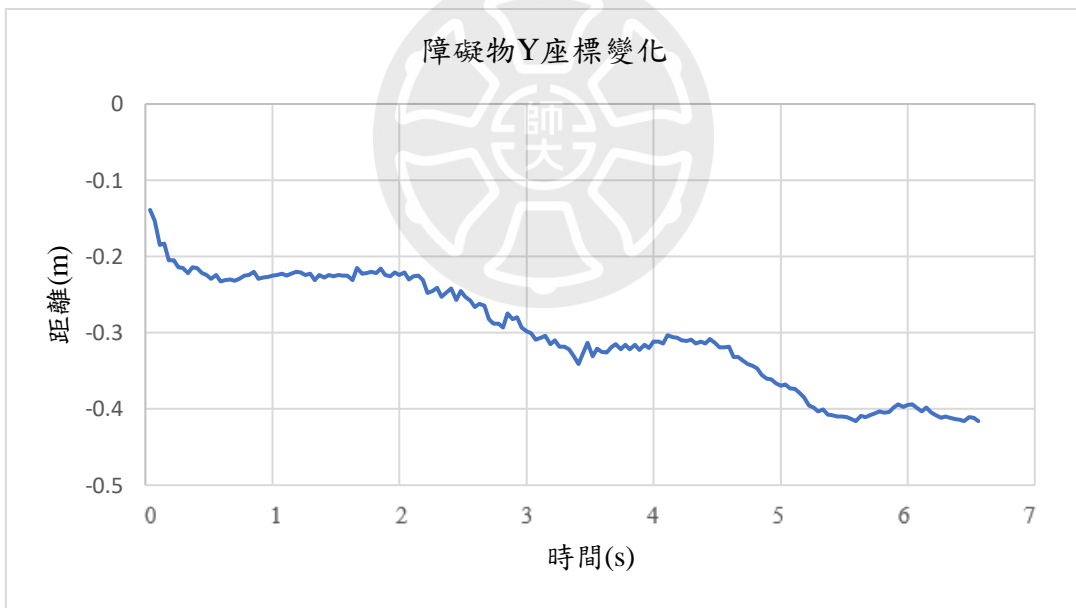


圖 5-39 障礙物 Y 軸變化圖

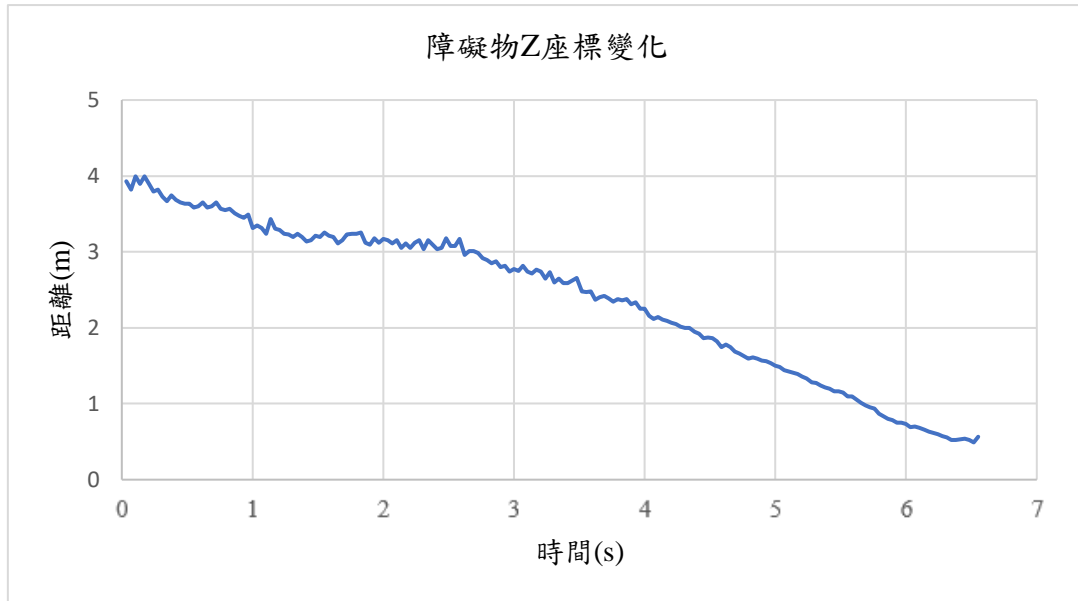


圖 5-40 障礙物 Z 軸變化圖

實驗結果顯示，無人機可在不依賴人工遙控的情況下，準確追蹤靜態規劃路徑，並於前方出現障礙物時即時進行感測、判斷並重新導引，成功避開障礙並回歸原始航線或修正後航線，持續朝向目標前進。

整體而言，本系統在整合導航、感測與控制三大功能後，具備良好的自主導航表現，能應對包含靜態與動態障礙物的複雜場域，具備應用於真實場景中的可行性。

## 第六章 結論與未來展望

### 6.1 結論

本研究針對無人機於複雜環境中進行自主避障與導航的需求，設計並實作一套結合視覺感知與路徑規劃的導航系統。系統整合動態與靜態路徑規劃，具備即時場景感知與避障決策能力。

在動態規劃方面，採用 Farneback 光流法搭配影像金字塔強化多尺度運動估計，並以 DBSCAN 類聚辨識障礙物輪廓。以光流補償技術凸顯動態障礙物，結合遮蔽區面積估算與 BFS 尋找空曠區域，並透過反投影獲取空間座標以生成避障路徑。在靜態規劃方面，導入 RRT\* 演算法建立從起點至終點的初始導航路徑，避開已知靜態障礙，提升整體導航效率。

經由實驗驗證後顯示其於靜態與動態障礙環境中均具備良好的辨識準確率與避障成功率。在真實場景的自主導航實驗中，系統亦可依據 GPS 導航與即時視覺感知自動調整航向，安全繞行障礙物並準確抵達目標，顯示其具備高度的實用潛力。

### 6.2 未來展望

1. 目前使用光流補償方式進行動態物體辨識，在高速場景中仍可能出現破圖或延遲。未來可引入深度學習架構進行時序運動特徵學習，提升對高速運動目標的辨識準確率。
2. 未來導入多感測器融合技術，增強在低光、反射或多障礙環境中的穩定性與準確性。
3. 現階段可以透過障礙物位置，估算其移動方向與速度。未來可導入強化學習，讓系統從經驗中學習最佳避障策略，以強化決策彈性與應變能力。

## 参 考 文 献

- [1] S. Karlsson, C. Kanellakis, S. S. Mansouri and G. Nikolakopoulos, "Monocular Vision-based Obstacle Avoidance Scheme for Micro Aerial Vehicle Navigation," *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, pp. 1321-1327, 2021.
- [2] Zhao Chunhui, Jiang Shan, Zhu Haifeng, Zhu Kaixuan, Lyu Yang and Zhang Qingchun, "Aerial target detection and avoidance for UAV based on stereo vision," *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, pp. 1670-1675, 2016.
- [3] P. Gao, D. Zhang, Q. Fang and S. Jin, "Obstacle avoidance for micro quadrotor based on optical flow," *2017 29th Chinese Control And Decision Conference (CCDC)*, Chongqing, China, pp. 4033-4037, 2017.
- [4] Q. Chen and M. Wang, "An improved RRT\* algorithm for mobile robots path planning," *2022 China Automation Congress (CAC)*, Xiamen, China, pp. 4334-4339, 2022.
- [5] Z. Ma, T. Wang, S. Xu, X. Mu, Q. Wang and Q. Guo, "Moving object Detection Based on Farneback Optical Flow," *2023 42nd Chinese Control Conference (CCC)*, Tianjin, China, pp. 7350-7355, 2023.
- [6] F. Santoro, S. Pedro, Z. -H. Tan and T. B. Moeslund, "Crowd analysis by using optical flow and density based clustering," *2010 18th European Signal Processing Conference*, Aalborg, Denmark , pp. 269-273, 2010.
- [7] B. Chen, Z. Cai, Z. Xiao, J. Yu and L. Liu, "Real-time Detection of Dynamic Obstacle Using Laser Radar," *2008 The 9th International Conference for Young Computer Scientists*, Hunan, China, pp. 1728-1732, 2008.

- [8] S. Hrabar, "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 807-814, 2008.
- [9] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," in *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80-92, Dec. 2011.
- [10] F. Liu, C. Shen, G. Lin and I. Reid, "Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024-2039, 1 Oct. 2016.
- [11] M. Perrollaz, J. -D. Yoder and C. Laugier, "Using obstacles and road pixels in the disparity-space computation of stereo-vision based occupancy grids," *13th International IEEE Conference on Intelligent Transportation Systems*, Funchal, Portugal, pp. 1147-1152, 2010.
- [12] Z. Ma, T. Wang, S. Xu, X. Mu, Q. Wang and Q. Guo, "Moving object Detection Based on Farneback Optical Flow," *2023 42nd Chinese Control Conference (CCC)*, Tianjin, China, 2023, pp. 7350-7355.
- [13] S.M. LaValle and J.J. Kuffner, "Rapidly-exploring random trees: A new tool for path planning," *Technical Report, Computer Science Department*, Iowa State University, 1998.
- [14] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [15] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," in *Computer*, vol. 22, no. 6, pp. 46-57, June 1989.

- [16] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, pp. 573-580, 2012.
- [17] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," *in Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA)*, Halmstad, Sweden, pp. 363–370, 2003.
- [18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise, " *in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, Portland, OR, USA, pp. 226–231, 1996.
- [19] S. Zingg, D. Scaramuzza, S. Weiss and R. Siegwart, "MAV navigation through indoor corridors using optical flow," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, pp. 3361-3368, 2010.
- [20] Goerzen, C., et al. "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. " *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1–4, pp. 65–100, 2009.
- [21] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision, " 2nd ed. Cambridge, U.K.: Cambridge University Press, 2004.
- [22] Z. Zhang, "A flexible new technique for camera calibration," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.

- [23] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects, " *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, Eds. Wellesley, MA: AK Peters, pp. 293–308, 2000.
- [24] J. Tian, T. Chao, M. Yang, J. Zhu and S. Wang, "A path planning algorithm based on improved RRT\* for UAVs," *2022 IEEE International Conference on Unmanned Systems (ICUS)*, Guangzhou, China, pp. 1-6, 2022.
- [25] B. K. P. Horn, "Robot Vision," MIT Press, 1986.
- [26] HT32F52342 \_datasheet, from  
[https://www.holtek.com.tw/WebAPI/11842/HT32F52342\\_52\\_Datasheetv150.pdf/448fb111-4158-4893-9d88-60f02ed96526](https://www.holtek.com.tw/WebAPI/11842/HT32F52342_52_Datasheetv150.pdf/448fb111-4158-4893-9d88-60f02ed96526)
- [27] Intel® RealSense™ Depth Camera D435i \_datasheet, from  
<https://www.intelrealsense.com/depth-camera-d435i/>
- [28] Neo-M8N \_datasheet, from  
[https://content.u-blox.com/sites/default/files/NEO-M8-FW3\\_DataSheet\\_UBX-15031086.pdf](https://content.u-blox.com/sites/default/files/NEO-M8-FW3_DataSheet_UBX-15031086.pdf)
- [29] 3D map information, from  
[https://www.google.com/maps/@25.0267794,121.5297674,188m/data=!3m1!1e3?entry=tту&g\\_ep=EgoyMDI1MDcxNS4xIKXMDSоASAFQAw%3D%3D](https://www.google.com/maps/@25.0267794,121.5297674,188m/data=!3m1!1e3?entry=tту&g_ep=EgoyMDI1MDcxNS4xIKXMDSоASAFQAw%3D%3D)