



逢甲大學學生報告 ePaper

報告題名：

即時影像處理於自走車避障之設計與實現

Obstacle Detection and Dodge of Automatic Vehicle System by
Real-Time Image Processing

作者：陳守信

系級：自控四甲

學號：D9323235

開課老師：陳杏圓

課程名稱：專題(二)

開課系所：自控系

開課學年：95 學年度 第 2 學期



中文摘要

本專題的研究目的於(1)設計並實現一台具有視覺功能的影像伺服控制之自走車，(2)設計影像演算法在特定環境中找出預設障礙物，(3)找出影像座標轉換求得距離，根據所得距離經由閃避法則來控制馬達，以達到閃避障礙物的目的。在硬體實現的過程中，先利用 CCD 以及影像卡，作為影像擷取前端，並以電腦作為影像處理平台，最後將運算結果經由閃避規則，控制兩顆 AC 伺服馬達；而在軟體實現方面，本人以 Borland 公司所發展的 Delphi 7.0 開發環境作為整體介面的設計，其中包含硬體介面控制、影像處理以及閃避法則規劃；結合硬體與軟體的實現整合於自走車上達到閃避障礙物的目的。

關鍵字：自走車避障、影像處理、Image Processing、Obstacle Avoiding、Autonomous Mobile Vehicle。

目次

中文摘要.....	1
第一章 緒論.....	6
1.1 前言.....	7
1.2 文獻回顧與研究動機.....	7
1.3 目標.....	8
1.4 專題結構.....	8
第二章 系統硬體架構.....	9
2-1 系統硬體設備.....	9
2-2 影像辨識系統硬體架構.....	9
2-2-1 CCD.....	11
2-2-2 影像擷取卡.....	11
2-2-3 AC 伺服馬達.....	12
2-2-4 馬達軸卡.....	12
2-2-5 個人電腦.....	12
第三章 障礙物距離計算與數位影像處理.....	13
3-1 距離檢測公式之推導.....	13
3-2 數位影像處理.....	16
3-2-1 影像擷取 (Image Capture).....	17
3-2-2 前處理 (Pre-processing).....	17
3-2-3 二值化 (Threshold).....	20
3-2-4 形態學處理 (Morphological processing).....	22
3-2-5 邊緣偵測 (Edge detection).....	25
3-2-6 影像分割 (Image segmentation).....	28
3-2-7 面積過濾 (Size filtering).....	29

第四章 實驗步驟與結果討論.....	31
4-1 演算法步驟.....	32
4-2 實際測試.....	37
4-2-1 閃避規則.....	37
4-2-2 實驗結果.....	41
4-3 結果討論.....	42
第五章 結論與未來展望.....	43
5-1 結論.....	43
5-2 未來展望.....	43
參考文獻.....	44



圖目錄

圖 2.1	影像辨識系統硬體架構.....	10
圖 2.2	自走車實驗設備.....	10
圖 2.3	自走車照片.....	10
圖 2.4	彩色 CCD 以及變焦倍鏡.....	11
圖 2.5	PCI-V114 內部功能方塊.....	11
圖 2.6	PCI-V114 影像擷取卡.....	12
圖 3.1	避障安全距離判斷的流程圖.....	13
圖 3.2	透視成像原理.....	14
圖 3.3	自走車 CCD 影像座標與世界座標概念圖.....	15
圖 3.4	本專題設定之障礙物.....	16
圖 3.5	本專題影像處理方法.....	17
圖 3.6	影像擷取圖片(包含障礙物).....	17
圖 3.7	彩色影像.....	18
圖 3.8	灰階影像.....	18
圖 3.9	對比度增強的灰階轉換函數.....	19
圖 3.10	對比加強前.....	20
圖 3.11	對比加強後.....	20
圖 3.12	閾值(116-146).....	21
圖 3.13	閾值(40-116).....	21
圖 3.14	$A \oplus B$ (dilation).....	22
圖 3.15	膨脹前以及膨脹後.....	22
圖 3.16	$A \ominus B$ (erosion).....	23
圖 3.17	侵蝕前以及侵蝕後.....	23
圖 3.18	斷開示意圖.....	24

圖 3.19	斷開前以及斷開後.....	24
圖 3.20	閉合開示意圖.....	25
圖 3.21	閉合前以及閉合後.....	25
圖 3.22	各種邊緣偵測法的結果.....	26
圖 3-23	Sobel 遮罩運算子.....	27
圖 3-24	Sobel 的運算方塊.....	27
圖 3.25	Sobel 邊緣偵測前以及結果.....	28
圖 3.26	Sobel 結果經過二值化.....	28
圖 3.27	影像分割示意圖.....	29
圖 3-28	影像分割前與分割後.....	29
圖 3.29	面積過濾示意圖.....	30
圖 3.30	面積過濾前與過濾後.....	30
圖 4-1	Delphi 7.0.....	31
圖 4-2	撰寫完成執行介面.....	32
圖 4.3	演算法步驟顯示結果.....	36
圖 4.4	自走車與障礙物關係俯視圖.....	37
圖 4.5	自走車路徑回歸示意圖.....	39
圖 4.6	自走車閃避流程方塊圖.....	40
圖 4.7	模擬路徑圖.....	41
圖 4.8	自走車實際測試結果.....	41

表目錄

表 2-1 馬達詳規	12
表 3.1 影像座標對應真實(X,Y)之 Data Base	15



第一章 緒論

1.1 前言

電腦視覺應用於行走於室內環境的自走車可以提供人類許多幫助，借助電腦視覺方面之研究的進步，以及電腦本身計算能力的增強，未來我們可以將無人駕駛之自走車應用在公路上的自動駕駛、危險環境探勘、工廠生產線上的自動搬運車等等許多方面。

在以上的應用之中，都必須能夠偵測週遭環境是否有障礙物存在，以防止自走車撞到障礙物。所以如何對於週遭環境提供即時且有效的偵測是否有障礙物，是我們實現無人自走車在未知環境中行走的一項重要工作。其中如何使系統能偵測移動或是靜止中的障礙物並且做出反應，是我們必須要解決的一樣最重要的問題，這項工作對於自走車的導航與避撞來說尤其重要，攸關於自走車安全及是否能順利執行任務。更進一步的來說，自走車如果具備視覺辨識能力，我們就可以設計標誌，經由自走車本身的視覺辨識系統，讓自走車隨機的遵循不同的標誌來執行不同的指令，而不必侷限於預先輸入的電腦指令。

最後，為了要使自走車能更落實在家用環境的應用，如果在簡單的個人電腦上能完成運算，那麼自走車將會更增廣到家用層面，如自動除草機，自動吸塵器等。

1.2 文獻回顧與研究動機

大部份機器人如在未知環境中工作，針對其所使用之檢測用的感測器，有一般常用者有：（一）影像感測器，（二）超音波感測器，（三）雷射或紅外線測距感測器。超音波感測器在使用上會有檢測距離上的限制、較低的測量分辨率以及需要具有對超音波能反映之特性等缺點。而在雷射或紅外線測距感測器而言，若目標物本身不能反射或會吸收發出訊號則不容易被應用[9]。許多研究中有關機器人定位及障礙物檢測上，較多是選擇影像感測器為主要的檢測工具[1][2]，此

即時影像處理於自走車避障之設計與實現

種感測器缺點在於其處理資料量多，造成資料處理時間較長，但此一缺點在現今 CPU 處理速度越來越快以及儲存技術容量越來越大的條件下是可以被克服的。如今影像處理已廣泛地利用於如醫學生物、工廠自動化、機器視覺等。

而關於影像處理對於偵測障礙物與辨識方面有許多方法[15][3]，例如(1)以顏色作為處理方法是將障礙物的顏色（如RGB等）做切割[5]；而(2)以形狀作為處理方法是利用物件之外觀形狀依據的辨識方式，如是先將影像邊緣化之後，再擷取邊界的特徵，如長度、曲率、面積等，或是使用資料庫將影像與預先的標誌樣版做比對。若將以顏色為處理方法和以形狀為處理方法相比較，一般而言以形狀為基礎法較具準確性與精確性。

1.3 目標

本篇論文的目標在發展一個即時的電腦視覺系統，此系統可以偵測障礙物的存在及量測障礙物大小與位置，當偵測出障礙物存在時，可回報障礙物的大小與位置，並且啟動即時閃避動作，以及即時建立及更新自走車所在環境之地圖。本研究的結果可以增進自走車對周遭環境的感知能力，增廣自走車的日常化應用。

1.4 專題結構

本文共分五個章節，各章節簡述如下：

第一章 緒論：簡單介紹國內外機器自走車的發展近況與文獻回顧，以及研究的動機和方向。

第二章 系統硬體架構及控制：首先說明相關的硬體設備及整個影像辨識系統硬體架構的規劃，最後針對影像系統硬體各部份加以說明。

第三章 障礙物距離計算與數位影像處理：整個影像辨識系統流程及針對數位影像處理的基本原理和影像處理操作介面，以及自走車和障礙物之間距離公式轉換，對應顯示 pixel 的位置來加以說明。

第四章 實驗步驟與結果：對本系統的實作流程及結果進行討論及分析。

第五章 結論與建議：對於本文做一總結及對未來發展工作之建議及應用。

第二章 系統硬體架構

本章將說明實驗設備、自走車影像辨識系統及整個系統的硬體架構之規劃。整個系統主要包含：(一)影像擷取平台，(二)影像擷取卡，(三)運算主機(PC)，(四)運動軸卡，(五)馬達控制器以及 AC 伺服馬達，(六)電源系統。

2-1 系統硬體設備

本專題所使用之硬體及實驗設備詳述如下：

1. 一台 CCD (Charge-couple device) 攝影機。
2. 一張影像擷取卡。
3. 一台 PC (P4-3.2G、1G Ram) 。
4. 一張運動軸卡。
5. 兩顆 AC 伺服馬達。馬達控制器。
6. 兩顆 12V 電瓶串聯成 24V，並利用變壓器升壓成 220V 提供電源給其他硬體如 PC、Monitor 使用。

除了上述設備以外，還有整個車體的配線、骨架、輪框膠輪、減速齒輪以及閉震器等機構部分。

2-2 影像辨識系統硬體架構

整個影像辨識系統硬體架構是由五個子部分所構成，(一) 影像感測器，(二) 影像擷取卡，(三) 運動軸卡及馬達控制器，(四) AC 伺服馬達，(五) 個人電腦。由影像感測器擷取影像輸入電腦，經影像處理後並擷取出障礙物資訊，由個人電腦送出指令藉由運動軸卡產生訊號給馬達控制器，以控制 AC 伺服馬達，並且由馬達回授資訊給電腦端。圖 2.1 即為整個硬體架構系統的示意圖、以及圖 2.2 自走車實驗設備、圖 2.3 自走車的照片。接下來分別對圖 2.1 中各方塊說明其功能及操作情形。

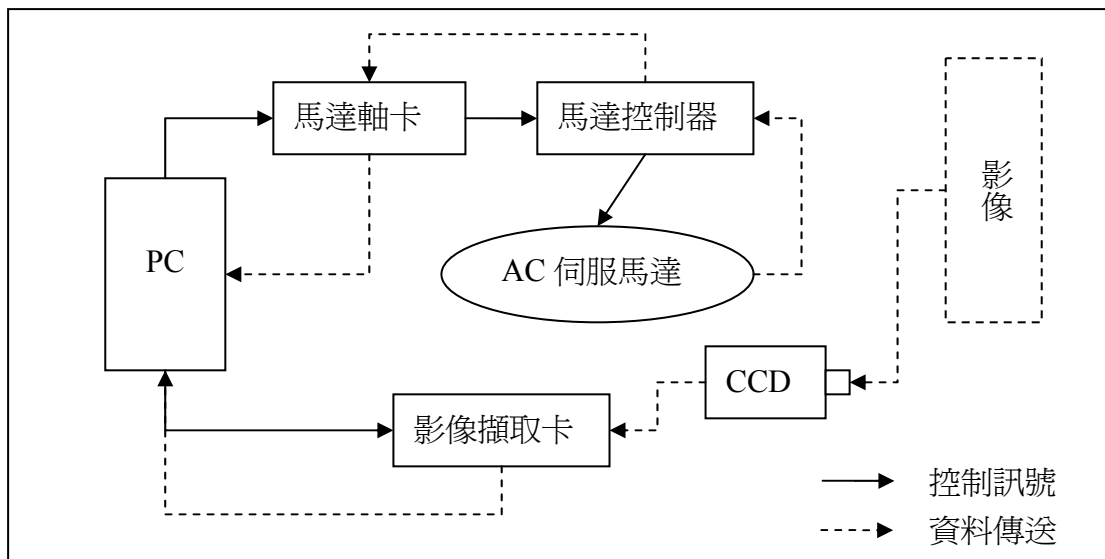


圖 2.1 影像辨識系統硬體架構

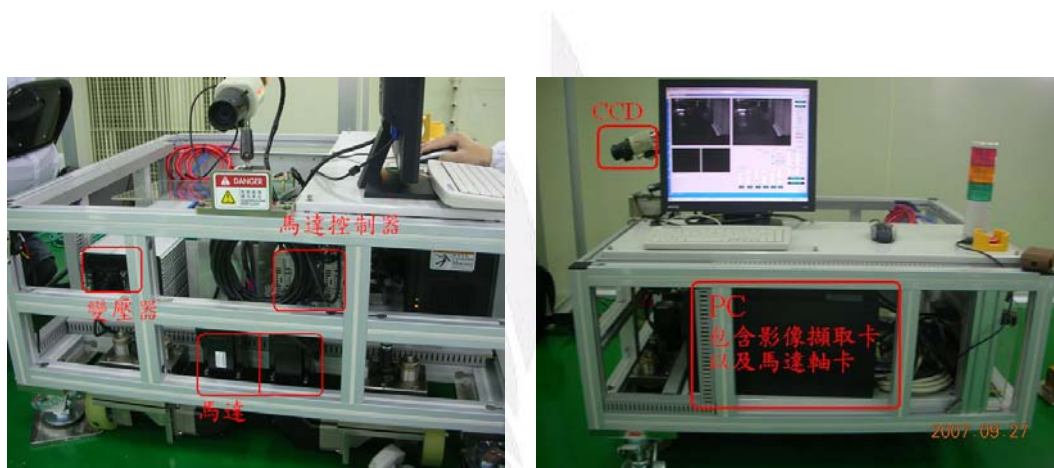


圖 2.2 自走車實驗設備



圖 2.3 自走車照片

2-2-1 CCD

影像感測器選用 Chiper 公司(www.Chiper.com) 生產的 CPT-ND82 高感度高解析度 1/3 吋彩色 CCD，並搭配由 Computar 公司 (www.Computar.com) 生產的 TG10Z101513FCS 變焦倍鏡，其變焦範圍以及可調最大放大倍率為 5m-50mm 和 10X，如下圖 2.3、圖 2.4 所示。



圖 2.4 彩色 CCD 以及變焦倍鏡

CCD 輸出格式為標準的彩色 NTSC/PAL，最大支援每秒 60 張 811*508 影像；另外在倍鏡調整放大率方面，調整為 4 倍，且利用影像演算程式觀察數值變化可手動調整到最清晰的影像。

2-2-2 影像擷取卡

影像擷取卡選用先達公司生產的 PCI-V114，其內部主要結構如圖 2.5 所示。

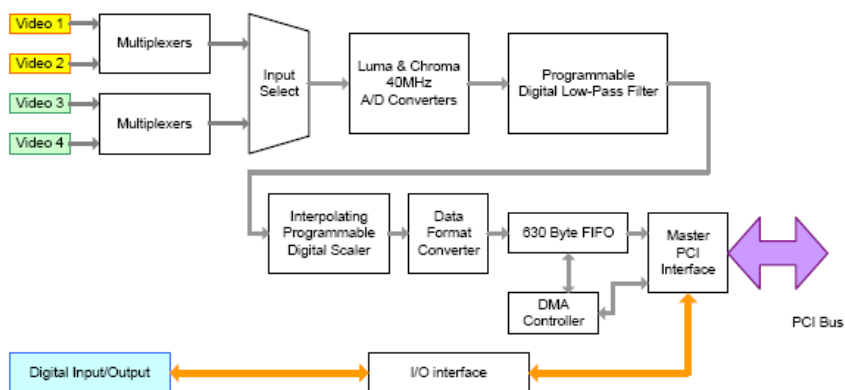


圖 2.5 PCI-V114 內部功能方塊

此卡功能主要是讓類比式輸入的影像訊號經由 A/D (Analog to Digital) 轉換，取樣後轉換成數位式的影像資訊，並輸入個人電腦內以做影像處理及障礙物分析與辨識。此卡另外提供 API (Application Programming Interface) 供影像擷取卡初始化以及設定擷取參數，如圖 2.6 所示。



圖 2.6 PCI-V114 影像擷取卡

2-2-3 AC 伺服馬達

以下為馬達詳規 - Yaskawa SGMAS-08ACA21

System Voltage	W(HP)	N • m	A _{RMS} (A)	Rated Angle Acc. rad / s ²
200V _{AC}	750 (1.01)	2.39	4.4	35600

表 2-1 馬達詳規

2-2-4 馬達軸卡

以下為馬達軸卡特性：

- (一) 4 軸獨立控制。
- (二) 2/3 線性同動控制。
- (三) 2 軸畫圓。
- (四) 4 軸 Encoder 輸入(A/B phase 或 Up/Down)。

2-2-5 個人電腦

個人電腦的程式開發環境以 Intel Pentium 3.0GHz 為中央處理器，2G 的記憶體，Microsoft Windows XP 作業系統，程式撰寫與編譯則是 Borland 公司生產的開發環境 Delphi 7.0。

第三章 障礙物距離計算與數位影像處理

自走車與周圍環境障礙物的距離是自走車避障時重要的參考數據，因此需要由取得的影像資料中找出障礙物外，並要推算出該障礙物與自走車間之距離。本章將推導距離檢測公式，以作為預測障礙物和自走車間之距離的依據。自走車避障全距離判斷的流程圖如圖 3.1 所示：

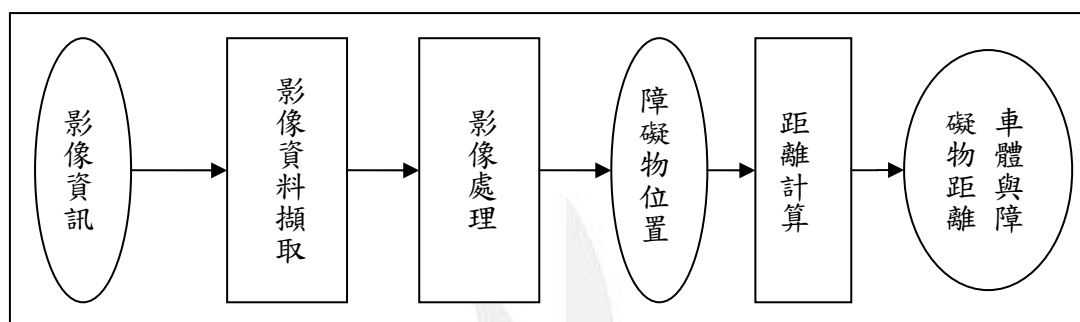


圖 3.1 避障安全距離判斷的流程圖

距離檢測公式之推導的目的在於希望利用影像經由影像處理完取得的資訊代入某一式子後，可以轉換成障礙物和自走車之間的距離，為了檢測正確性，檢測距離是以 2 公尺為參考量測依據。下節將說明公式的推導過程。

3-1 距離檢測公式之推導

本公式主要是利用透視轉換 (Perspective Transformations) 原理來找出障礙和自走車之間的距離。將物體三為點投影到影像平面上，用以觀察三維座標形成在影像平面上的情形。其成像基本模形如圖 3.2 所示：

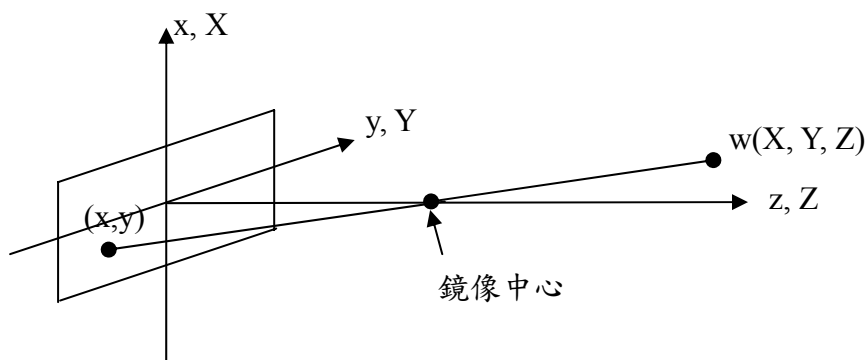


圖 3.2 透視成像原理

圖 3.2 中 (x, y) 為影像感測器平面座標系統，其 z 方向為原點 0 ，而 (X, Y, Z) 為實體世界三維座標系統 (world coordinates)。在影像感測器座標系統 (x, y) 中，假設成像在 x 、 y 平面上，而光軸沿 z 軸，且設成像面的中心在原點處，透鏡的焦點在座標 $(0, 0, \lambda)$ 處， λ 是透鏡焦距 (focal length)。設 (X, Y, Z) 是三維座標中任意點的世界座標，如圖 3.2 中之 w 點所示。若能找出點 (X, Y, Z) 與再成像面上的投影座標 (x, y) 所呈現的關係，便可用平面相似三角形的幾何關係找出實物座標點與影像點的關係，其關係如下：

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda} = \frac{X}{\lambda-Z} \quad (3-1)$$

$$\frac{y}{\lambda} = -\frac{Y}{Z-\lambda} = \frac{Y}{\lambda-Z} \quad (3-2)$$

而上兩式中的負號表示影像點實際上是倒置的。

然而將一個三維座標映射到一個影像平面是一個多對一的轉換，也就是由一個影像點可能會對應到多個世界點的位置。在沒有雙鏡頭達成立體 (3-D) 成像技術 [4][16] 的前提下，為了解決只有一個鏡頭無法確實求得 Z 軸之深度，以及鏡面曲率偏差或誤差，而建立 Data Base 來表示對於影像平面坐標所表示實際的距離，如表 3.1 所示：

x (pixel)	X (cm)	y (pixel)	Y (cm)
-------------	----------	-------------	----------

239+0	0	319+0	0
239+22	20	319+29	20
239+48	40	319+65	40
239+80	60	319+111	60
239+118	80	319+171	80
239+164	100	319+233	100
239+227	120	319+308	120

表 3.1 影像座標對應真實(X,Y)之 Data Base

上述 Data Base 中的 X 、 Y 代表 CCD 的測量點現實位置與中心點所對應的現實位置的距離，而測量點的位置會因為 CCD 本身的最大擷取仰角以及俯角而限制； x 、 y 代表影像平面座標點。此 Data Base 是根據擷取儲存影像為 640×480 來規劃的，也就是在影像平面座標會有 640×480 個 pixel 點，而 CCD 中心點在平面座標位置為 $(239, 319)$ 。在這 Data Base 使用的前提下，所偵測出來的障礙物資訊必須是在現實中的地面點，如圖3.3所示：

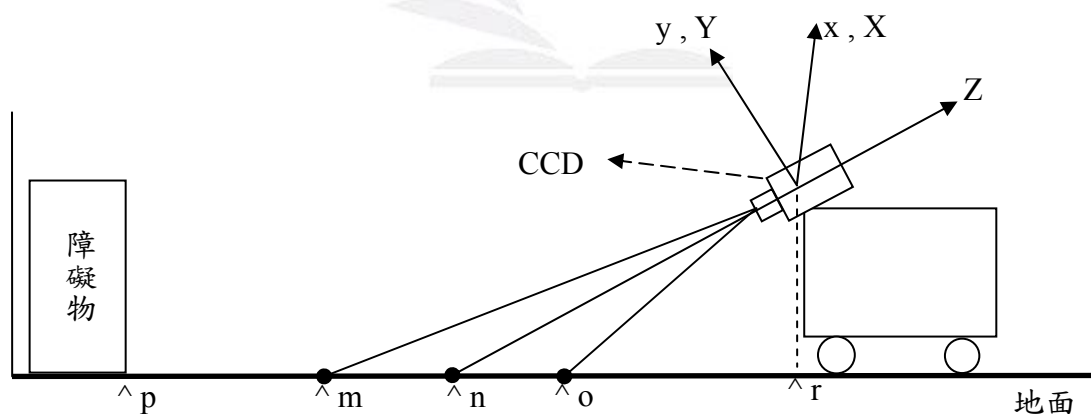


圖 3.3 自走車 CCD 影像座標與世界座標概念圖

圖中 x 、 y 軸以及 X 、 Y 、 Z 軸分別為影像平面座標以及真實的世界座標， m 、 n 、 o 、 p 、 r 分別代表地面上五個假設點：CCD 最高仰角實際點、CCD 最

低俯角實際點、CCD 中心實際點、障礙物底端與地面接觸點、CCD 影像平面與地表垂直線相交點。假設上圖地面與 Z 軸為同一平面，則上圖地面 n 點會在影像平面座標的 (0, 0) 處，而地面 m、n 點分別會在 (0, +y)、(0, -y) 處。假設 p 點為障礙底端與地面接觸的位置，則自走車的位置與障礙物的距離就可以利用 Data Base 中的數值加上距離 \overline{or} 即可求得。所以如何利用影像處理求得障礙物與地面接觸的 p 點是一個首要前提。

3-2 數位影像處理

影像處理方面分為下列幾個部份：影像的擷取、影像前處理、灰階臨界值、邊緣偵測、形態學處理。每一部份處理之目的均是針對影像中的特徵點擷取，也就是上圖 3.3 本專題所設定的障礙物柱子的底端 p 點，用以找出其障礙物特徵點來做安全距離判讀等。而本專題設定之障礙物以及所用的影像處理方法如圖 3.4、圖 3.5 所示

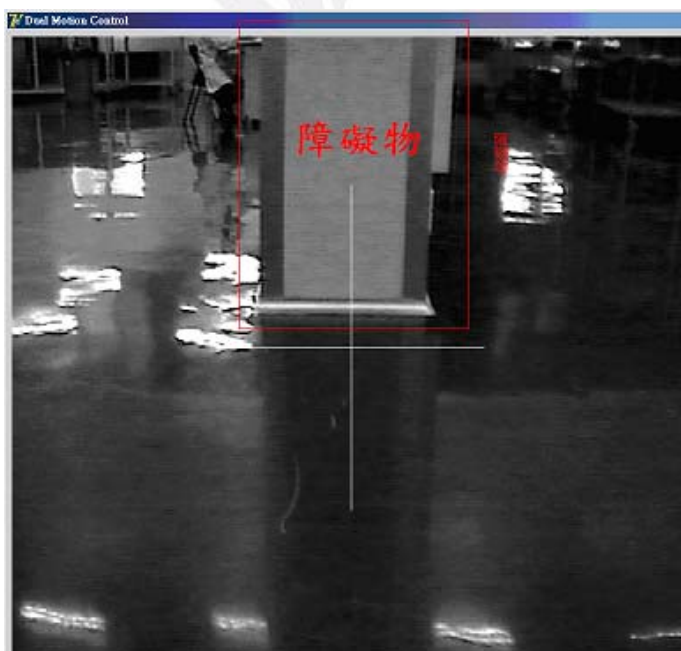


圖 3.4 本專題設定之障礙物

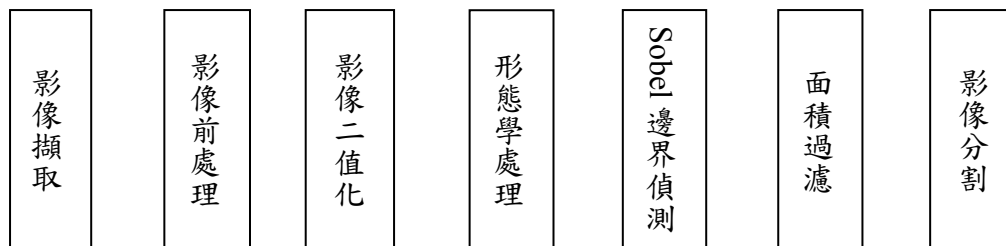


圖 3.5 本專題影像處理方法

3-2-1 影像擷取 (Image Capture)

影像擷取部份，是利用架設於自走車上的彩色 CCD 影像感測器，並以柱子當作為暫定的目標障礙物，並儲存成 640*480 pixel 之彩色影像檔。如圖 3.6 所示。



圖 3.6 影像擷取圖片(包含障礙物)

3-2-2 前處理 (Pre-processing)

前處理在於針對不同的環境下而對影像做不同的前置處理方式，本專題中選擇先將擷取圖片灰階化，再對影像灰階對比加強。

由於擷取後彩色圖片一般都是由 RGB 組成，在本專題由於(1)處理彩色圖片消耗時間較多，以及(2)精確性較不夠，常因為整個環境亮度的影像導致參數微調不容易，所以最後直接在灰階域作影像處理，下圖 3.7、圖 3.8 為彩色以及灰階化影像，彩色轉灰階最簡單的方法如下：

$$I = \frac{R + G + B}{3} \quad (3-3)$$

另外由與考慮到流明的光線方面問題，較正確的彩色影像灰階化公式應為[5]：

$$I = 0.299 * R + 0.587 * G + 0.144 * B \quad (3-4)$$



圖 3.7 彩色影像



圖 3.8 灰階影像

調整灰階對比的目的是在於能使目標障礙物能和背景影像能有更明顯的區分，以利找出目標障礙物。本專題中所謂加強灰階值對比轉換的效果，是把原始

影像中高於灰階度 m 的灰階度變暗，而低於 m 的灰階度變亮，如圖 3.9 所示：

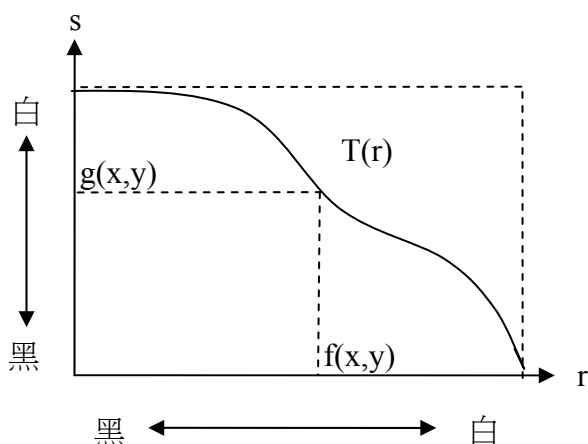


圖 3.9 對比度增強的灰階轉換函數

$f(x, y)$ 表示輸入影像某點的灰階值，而 $g(x, y)$ 表示處理後的影像某點的灰階值，而 r 代表輸入灰階值， s 代表輸出灰階值， T 為轉換函數且 $s = T(r)$ 。如圖 3.7 所示。這種對比加強方法，是將低於 m 的灰階值透過轉換函數像黑色方向壓縮到 s 的窄範圍內；對於高於 m 的灰階值則會發生相反的效應，圖 3.10、圖 3.11 分別為對比加強前以及對比加強後。其中調整灰階值對比公式如下：

$$S_i = a(G_i - \text{avg}) + \text{avg} \quad (3-5)$$

$$\text{avg} = \frac{1}{N_x N_y} \sum_{x=0}^{N_x} \sum_{y=0}^{N_y} F(x,y) \quad (3-6)$$

式(3-5)與(3-6)中：

S_i 代表影像轉換後第 i 個灰階值。

G_i 代表原始影像第 i 個灰階值。

avg 代表原始影像的平均灰階值。

a 代表調整倍率參數。

$F(x,y)$ 代表原始影像座標點 (x, y) 位置之灰階值。

另外做完對比度加強後，其灰階值可能低於 0 或高出 255，爲了下一步正

即時影像處理於自走車避障之設計與實現

確處理，而作了一個正規化的動作，也就是把結果的灰階值位移某個 K 值，再乘以一個 L 倍率，取結果四捨五入。最後使結果正規於 0-255 灰階值之間。而其轉換公式如下：

$$S_{ii} = \text{RoundTo}[L(S_i - K)] \quad (3-7)$$

S_{ii} 為正規化後結果。

$\text{RoundTo}()$ 為四捨五入函數。



圖 3.10 對比加強前



圖 3.11 對比加強後

3-2-3 二值化 (Threshold)

二值化又稱為灰階值分割，將經過灰階化的影像的灰階值分割成只有兩種灰

階度，也就是 0 或 255，其方法就是設定一個閾值 (Threshold)，只要影像灰階值大於閾值，就令其值為 255 或是全亮；反之，小於閾值就令其為 0 或是全暗，這樣就可以得到一個二值化的影像。二值化的影像可以有效的減少資料儲存量，在影像分割時不錯的一個演算法，而在閾值的選取則是根據實驗的環境條件來做調整。其演算法如下：

$$f(x, y) = \begin{cases} 255, & \text{IF } f(x, y) \geq T \\ 0, & \text{IF } f(x, y) < T \end{cases} \quad \text{or} \quad f(x, y) = \begin{cases} 255, & \text{IF } f(x, y) < T \\ 0, & \text{IF } f(x, y) \geq T \end{cases} \quad (3-8)$$

而圖 3.12、圖 3.13 是根據對比加強後用不同閾值來處理的結果：



圖 3.12 閾值(116-146)



圖 3.13 閾值(40-116)

由以上可知，較好的閾值的選取可以有效分離大部分的目標區域，使接下來的邊緣檢測跟面積判斷更加的方便也會比較精準。

3-2-4 形態學處理 (Morphological processing)

形態學最常用的有四種方法，而後兩種方法是由最初兩種方法組合而成。四種分別為：(1) 膨脹(dilation) (2) 侵蝕(erosion) (3) 斷開(opening) (4) 閉合(closing)，以下就四種方法做介紹：

(1) 膨脹 (Dilation)

膨脹運算表示式為：

$$A \oplus B = \{z \mid [(B)_z \cap A] \neq \emptyset\} \quad (3-9)$$

此式表示 A 集合藉由 B 結構元素做迴旋積分，使得 B 和 A 的重疊部分至少有一個元素，圖 3.14 表示其集合狀況。

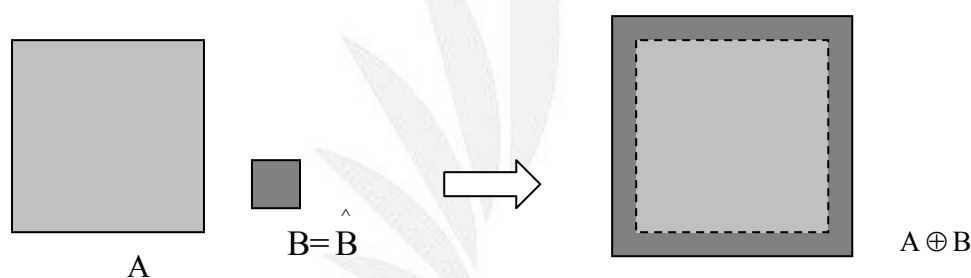


圖 3.14 $A \oplus B$ (dilation)

膨脹最簡單的應用就是在橋接縫隙。而在本專題當中，本人把它利用來擴張選取範圍，以便包含更多的邊界條件，如圖 3.15 所示



圖 3.15 膨脹前以及膨脹後

(2) 侵蝕 (Erosion)

侵蝕運算表示式為：

$$A \ominus B = \{z \mid \hat{(B)}_z \subseteq A\} \quad (3-10)$$

此式指出 A 被 B 的侵蝕是使得 B 位移 z 後仍包含在 A 中所有 z 點的集合，圖 3.16 表示其集合狀況。

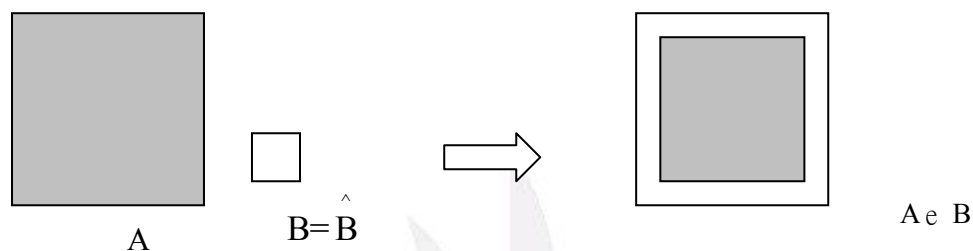


圖 3.16 $A \ominus B$ (erosion)

侵蝕最簡單的應用就是切除連接線以及除去二元影像裡面不相干的細節，尤其是小雜訊。本人專題利用侵蝕的演算法也是基於二值化後還是有些雜訊，利用 erosion 可以除去大部分的小雜訊。如圖 3.17 所示：

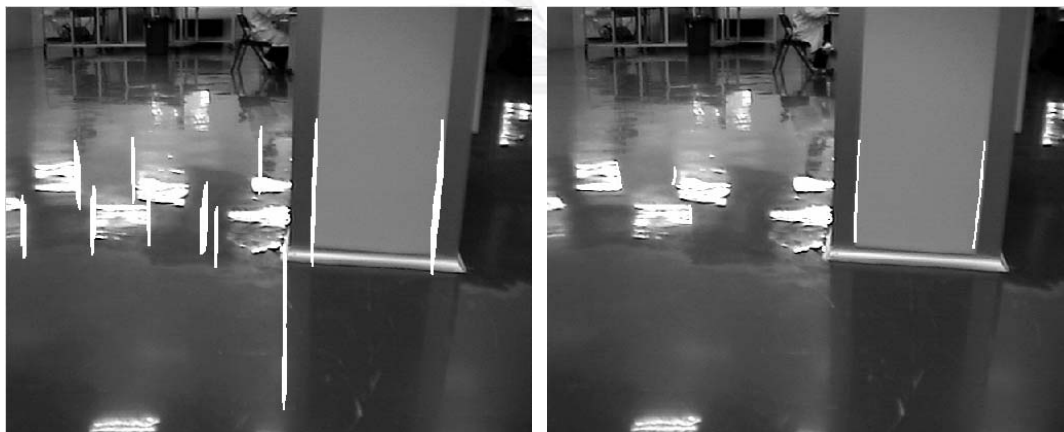


圖 3.17 侵蝕前以及侵蝕後

(3) 斷開 (Opening)

就如開頭所述，後兩種方法為前兩種方法：膨脹、侵蝕所組成。而斷開的意

思就是先侵蝕再在膨脹。此影像演算法可以讓連接點切開，其數學表示式以及表示示意圖圖 3.18 如下：

$$A \circ B = (A \ominus B) \oplus B \quad (3-11)$$

其中 $A \circ B$ 代表集合 A 被結構元素 B 斷開。

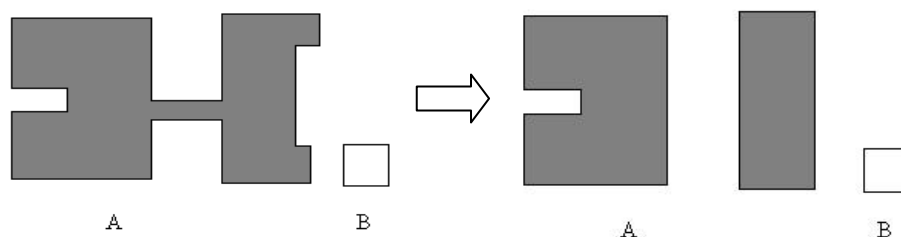


圖 3.18 斷開示意圖

從圖 3.17 並不難發現 A 集中間的连接線被切斷了。在本專題中，就是利用這種斷開方法去除橫縱向的邊緣，如下圖 3.19 所示：



圖 3.19 斷開前以及斷開後

(4) 閉合 (Closing)

與斷開處理方式相反，斷開是先膨脹再侵蝕，其數學表示式以及表示示意圖圖 3.20 如下：

$$A \bullet B = (A \oplus B) \ominus B \quad (3-12)$$

其中 $A \bullet B$ 代表集合 A 被結構元素 B 閉合。

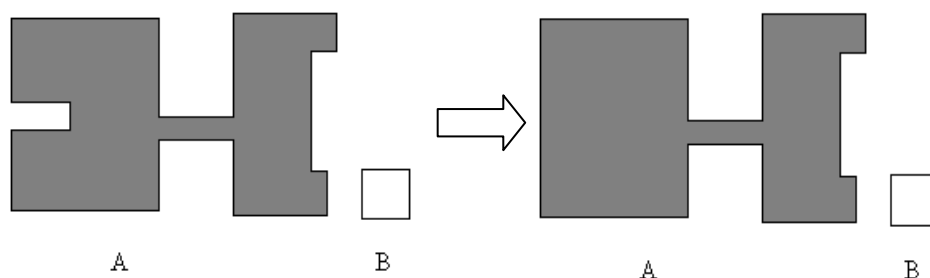


圖 3.20 閉合開示意圖

從圖 3.20 很明顯發現 A 集合左邊中間的縫隙經 opening 處理後閉合起來。在本專題中，就是利用這種閉合方法來把縱向邊緣斷接處接起來，如下圖 3.21 所示：



圖 3.21 閉合前以及閉合後

3-2-5 邊緣偵測 (Edge detection)

對於影像辨識的而言，過多的資訊，例如背景等非要辨識的物體會造成辨識速度緩慢、辨識率低的情況發生，因此物體的邊緣偵測是影像處理一個重要的課題和方法，其目的在於有效且準確的偵測到目標物的邊緣資訊。

如要知道物體的在影像中的形狀、面積、分布狀況等資訊，從邊緣判斷是最直接的，而好的邊緣偵測也是重要的選擇。在影像處理中常用的邊緣偵測方法有；Roberts、Prewitt、Sobel、Laplace、Canny。比較特殊不常見的有：Frei、

Kirsch、Robinson 等。圖 3.22 為各種邊緣偵測法的結果。

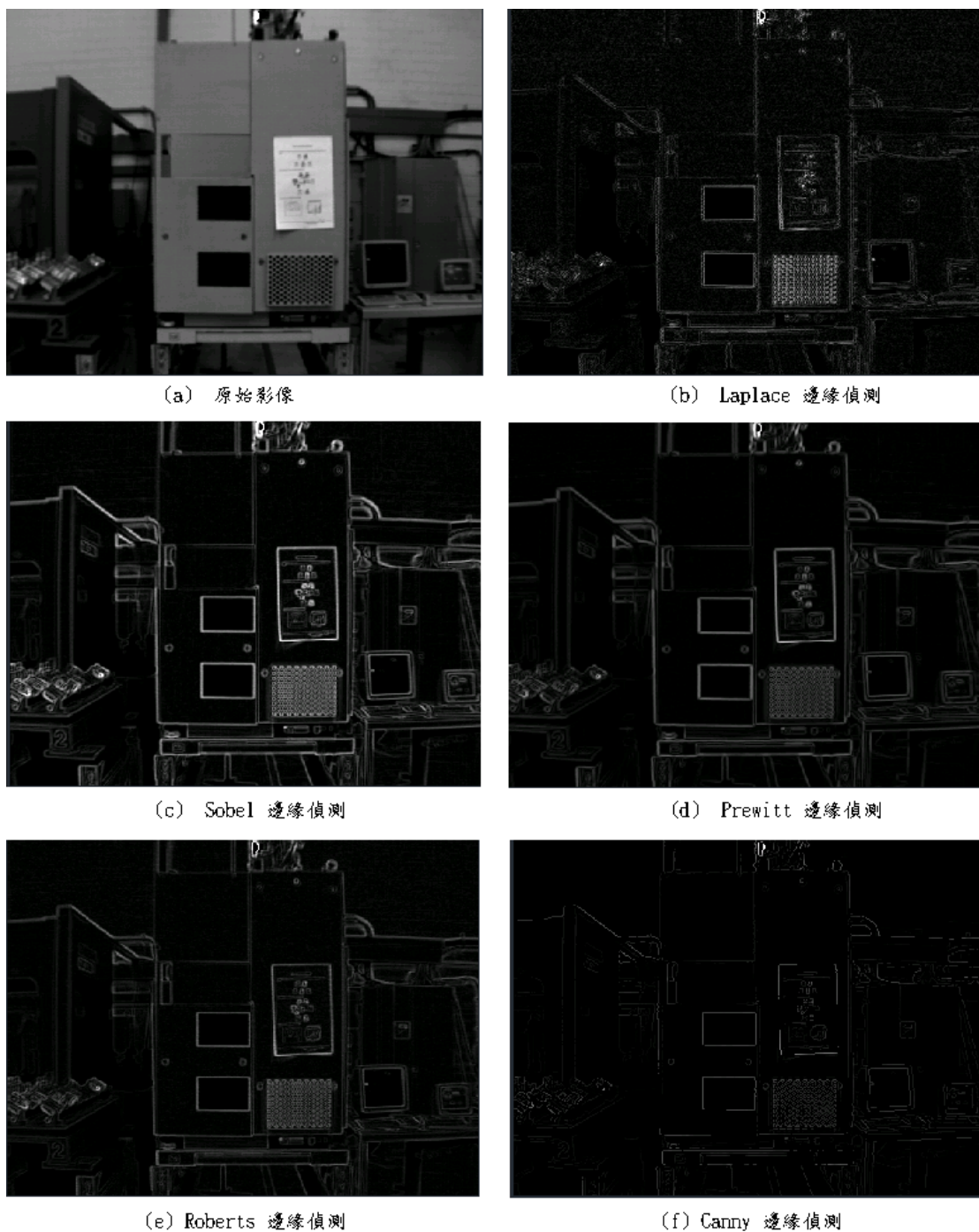


圖 3.22 各種邊緣偵測法的結果

邊緣偵測其實就是求某一個方向的一階導數，即影像 $f(x, y)$ 在 x, y 方向的梯度變化量。

$$\nabla f(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (3-13)$$

本專題中，採用的是 Sobel 邊緣偵測方法來作為物體邊緣的偵測，在 Sobel 邊緣偵測當中，使用的是下列圖 3-23 的遮罩運算子來求影像的梯度值。

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

圖 3-23 Sobel 遮罩運算子

Z ₁	Z ₂	Z ₃
Z ₄	Z ₅	Z ₆
Z ₇	Z ₈	Z ₉

圖 3-24 Sobel 的運算方塊

對於 Sobel 而言，其運算式為下式：

$$\begin{aligned} G_x &= (Z_7 + 2*Z_8 + Z_9) - (Z_1 + 2*Z_2 + Z_3) \\ G_y &= (Z_3 + 2*Z_6 + Z_9) - (Z_1 + 2*Z_4 + Z_7) \end{aligned} \quad (3-14)$$

而對算出來的梯度大小取一臨界值(Threshold)，大於某值的梯度值為 1，反之為 0。圖 3-24 左圖為已經先由影像處理取出所要的障礙物範圍原圖，接下來利用 Sobel 來做邊緣偵測。圖 3-26 為圖 3-25 經過二值化的結果

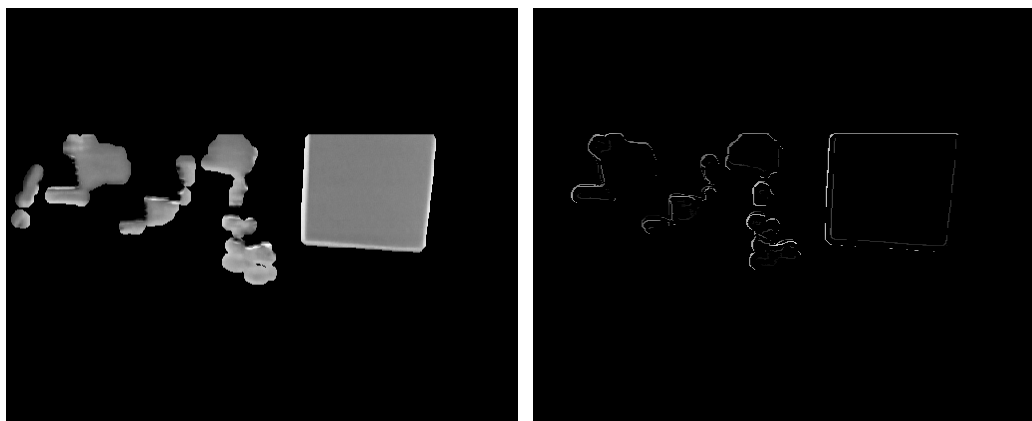


圖 3.25 Sobel 邊緣偵測前以及結果

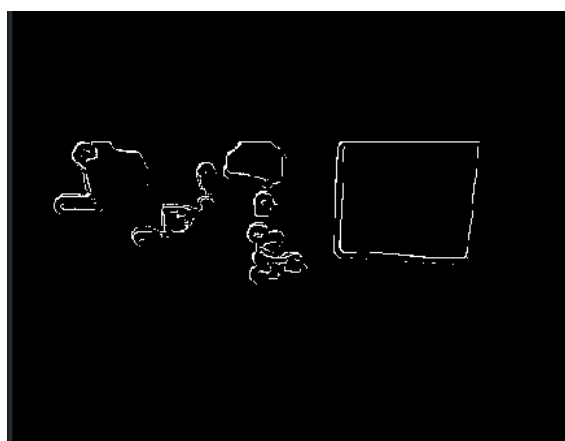


圖 3.26 Sobel 結果經過二值化

3-2-6 影像分割 (Image segmentation)

影像分割是影像處理中極為重要的一環，經過分割後的影像可以個別處理，例如面積、長度判斷等等。本專題影像分割使用在經過二值化篩選後的影像。如圖 3.27 所示，同一連接區域為一個範圍。在演算法上分為二種方法：回歸演算法 (recursive algorithm) 及漸進演算法 (sequential algorithm)，而另外在同區域檢查上分為二種：四鄰點檢查 (4-connectivity) 及八鄰點檢查 (8-connectivity)。本專題使用八鄰點檢查的漸進演算法，然後將每一個分割物獨立到新的影像中，如圖 3-28。

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	1	0
0	1	1	0	0	1	1	1	1	1
	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0
0	0	3	0	0	0	0	2	2	0
0	3	3	0	0	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0

圖 3.27 影像分割示意圖

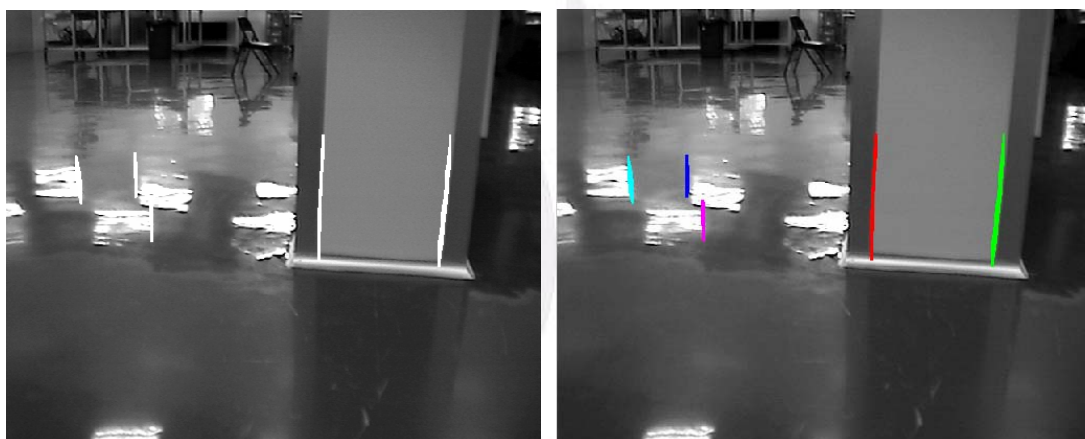


圖 3-28 影像分割前與分割後

3-2-7 面積過濾 (Size filtering)

面積過濾的工作原理是將影像分割完成之分割物，進行面積運算，以每一個像素為單位，如下圖 3.29 所示，分割物 1 的面積是 7 個像素，分割物 2 的面積是 8 個像素，分割物 3 的面積是 3 個像素，我們將面積小於 7 像素的分割物區域濾除。利用分割物區域所構成的面積進行過濾，可以濾除不符合特定面積的分割物，即是將小於一定面積的目標物視為雜訊。本專題中目標是影像中面積最大的分割物為障礙物，因此可以利用面積過濾，濾除較小的分割物，保留影像中最大面積的分割物，特別的是，可以濾除較面積較小的雜訊。圖 3.30 為處

理後結果。

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	1	0
0	1	1	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0

圖 3.29 面積過濾示意圖

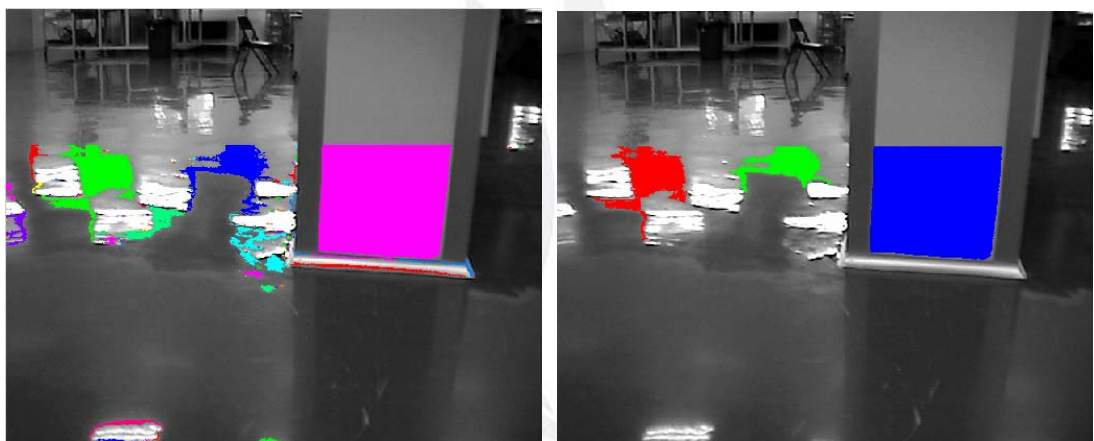


圖 3.30 面積過濾前與過濾後

第四章 實驗步驟與結果討論

本專題所有影像處理以及馬達控制整合，皆是在 Borland Delphi 7.0 開發環境下撰寫，下圖 4-1 是 Delphi 7.0 開發環境以及圖 4-2 為撰寫完成的介面 [16][17][18][19]。

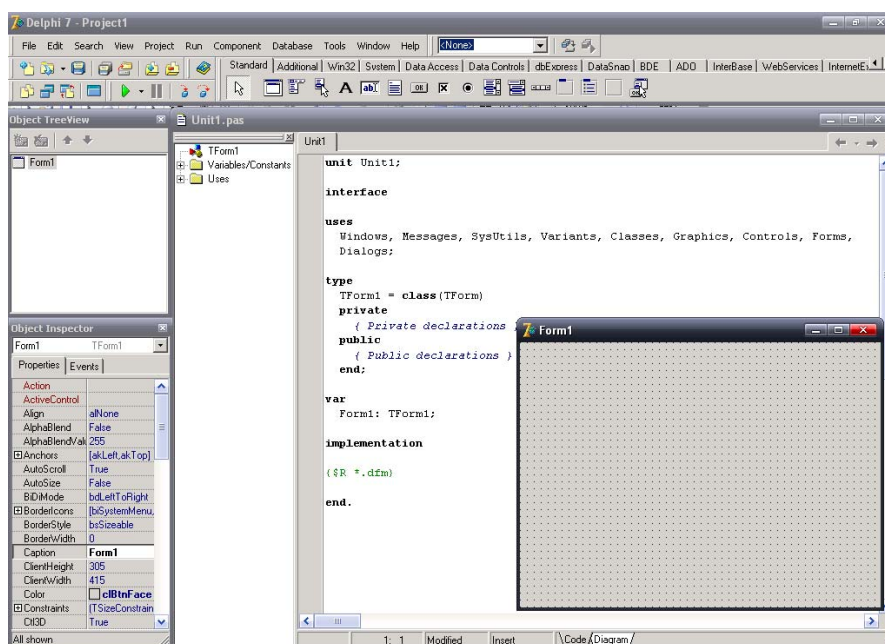


圖 4-1 Delphi 7.0

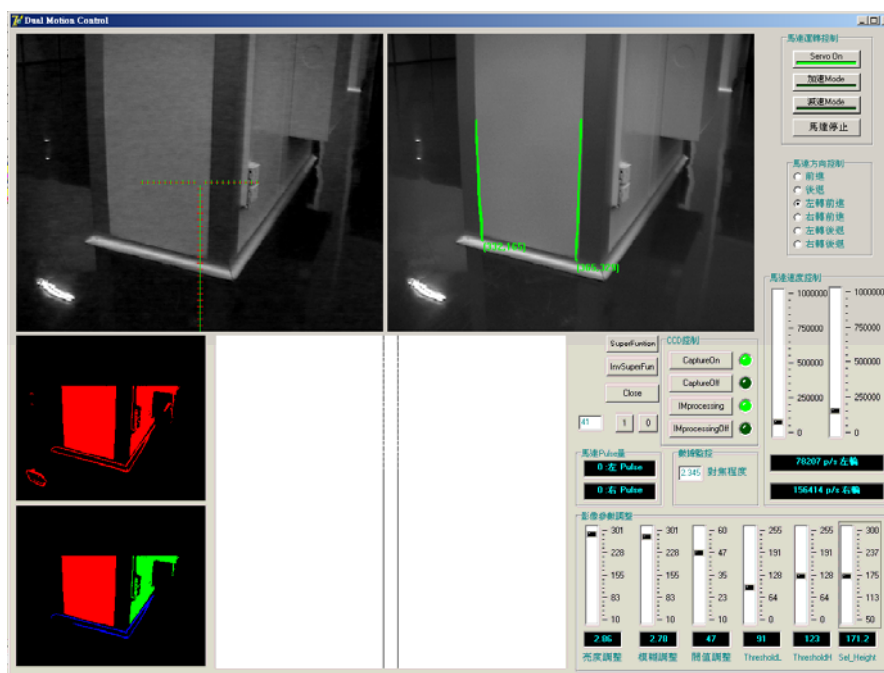


圖 4-2 撰寫完成執行介面

本實驗分為兩部分：第一部份為影像演算法的驗證，利用上述影像演算法針對障礙物進行測試；第二部份為實際測試，並且行走的同時會把偵測到的障礙物位置儲存。

4-1 演算法步驟

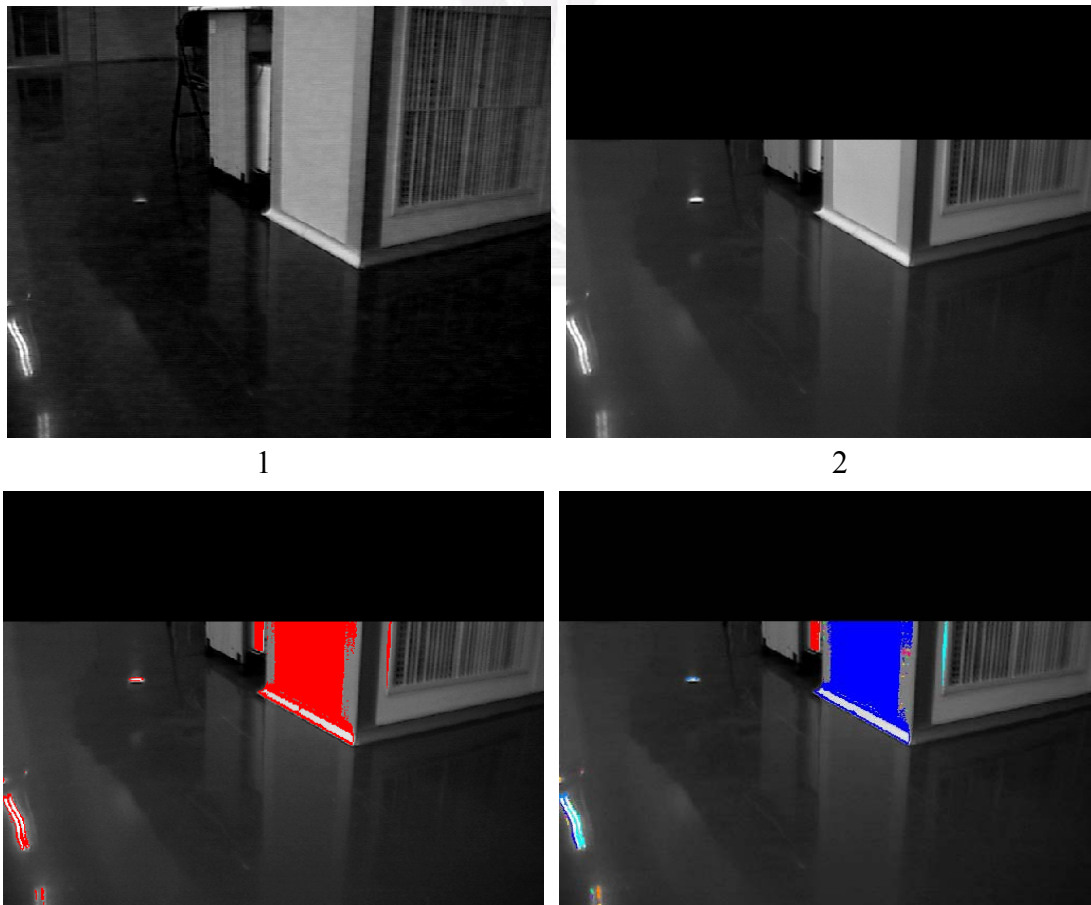
以下為演算法的步驟：

1. 擷取圖片(640*480)並轉化成灰階影像
2. 選取分割範圍，並分割 ($x: 0 \sim 639; y: 150 \sim 479$)
3. 取兩個二值化接近障礙物灰階值的範圍，將結果兩個範圍取邏輯 OR 相加
4. 影像分割
5. 選取面積大於某值
6. Opening
7. Dilation
8. 影像合併(邏輯 OR 相加)
9. 利用 8 與原始圖片 1 做邏輯 AND，選取要分析的圖片範圍

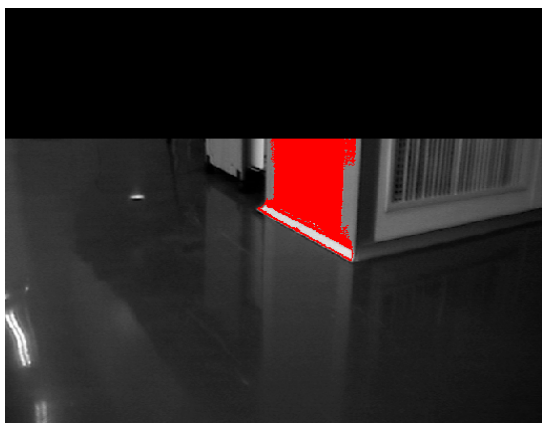
即時影像處理於自走車避障之設計與實現

10. 影像加強使對比度增加
11. Sobel 邊緣偵測
12. 邊緣取二值化
13. Opening
14. 影像分割
15. 選取高度(最小 y 軸與最大 y 軸相減)大於某值
16. Dilation
17. Erosion
18. 再次取高度
19. 選取範圍對低點的座標位置(x, y)並標示

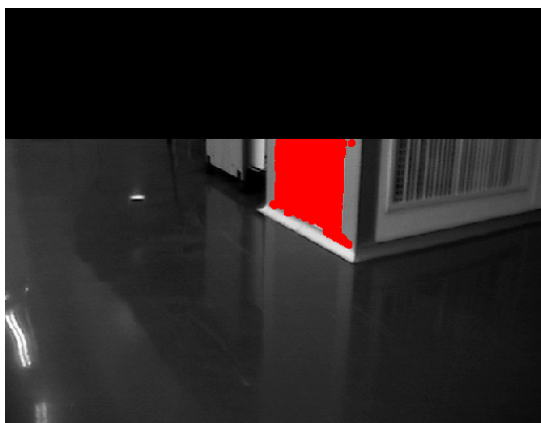
且由求得的座標位置配合距離轉換 Data Base 可以求得目前自走車與障礙物的 X、Z 軸實際距離。圖 4.3 是由以上的演算法得到的結果。



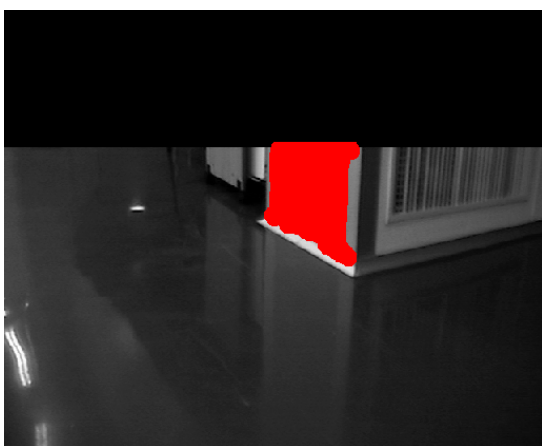
3



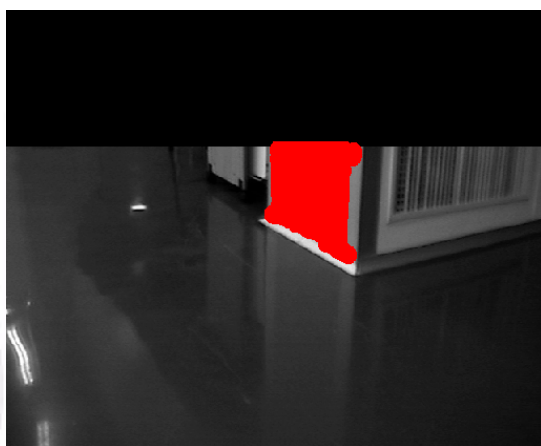
4



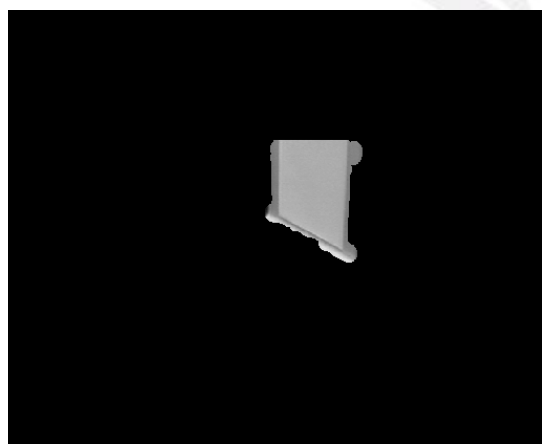
5



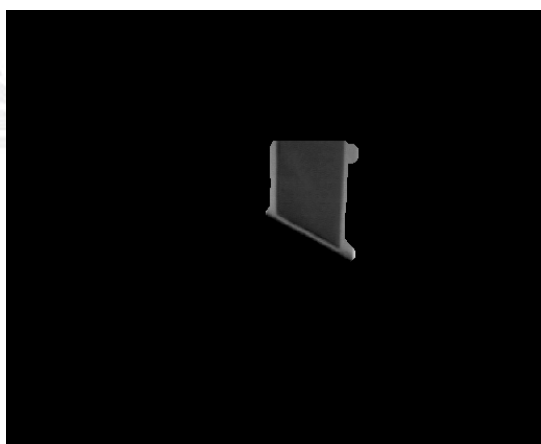
6



7



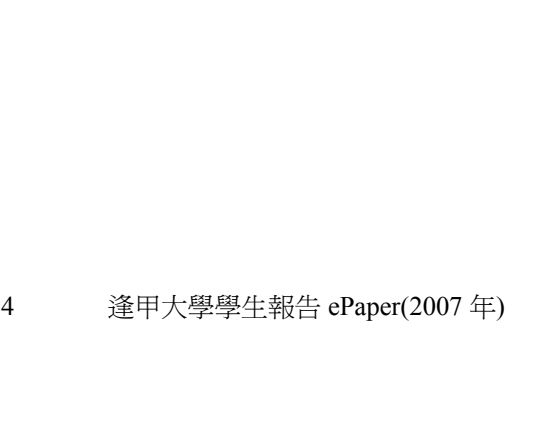
8

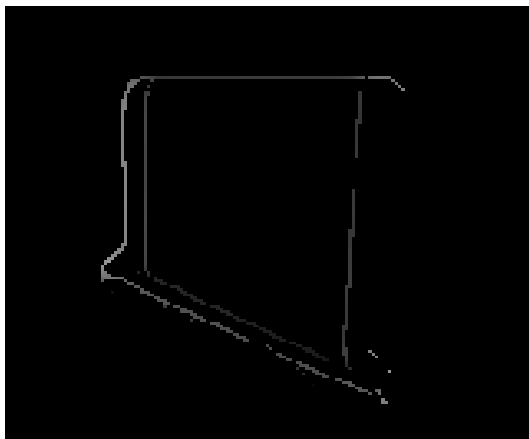


9



10

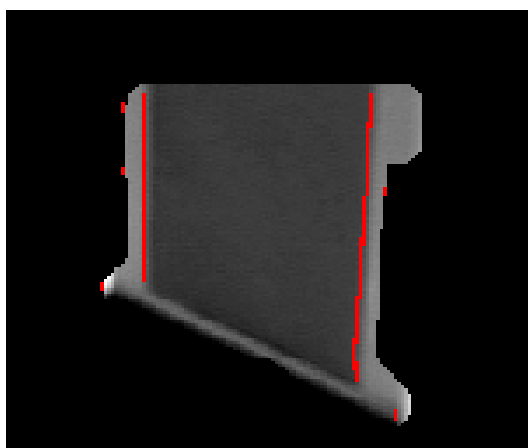




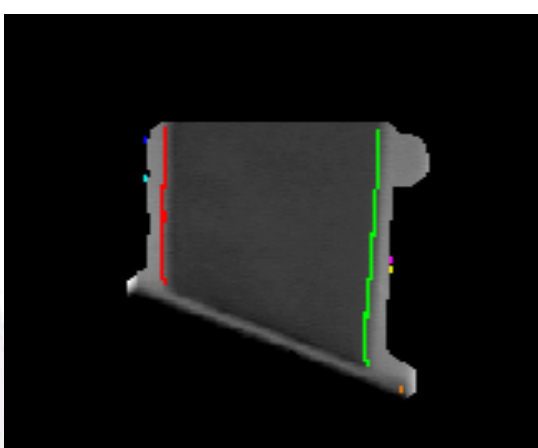
11



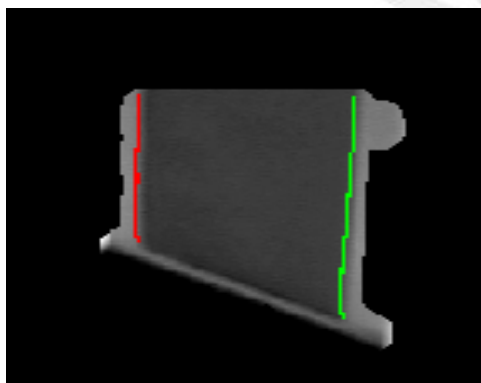
12



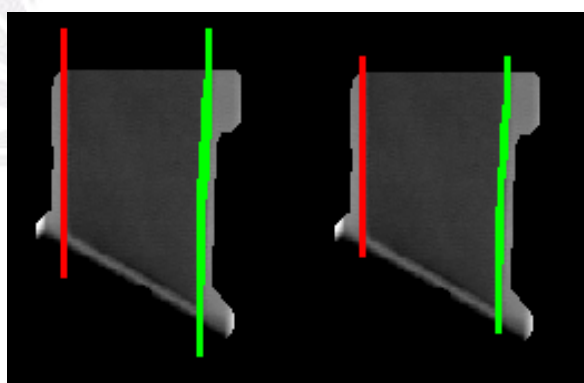
13



14

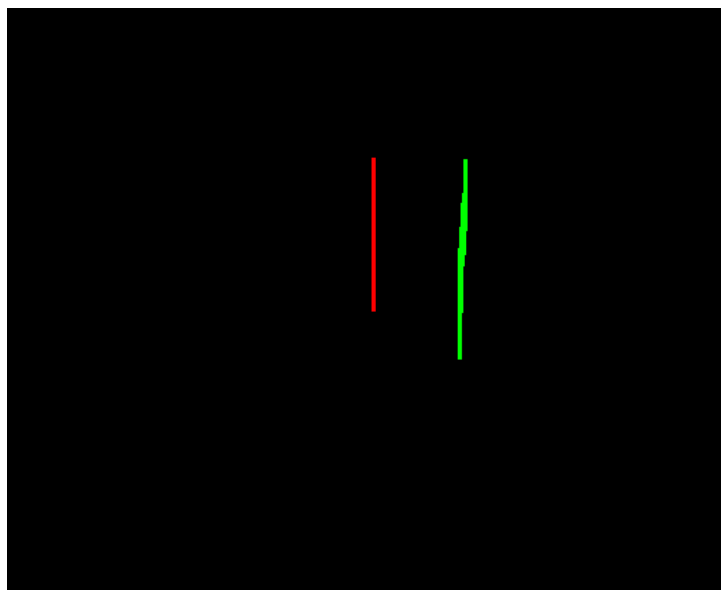


15

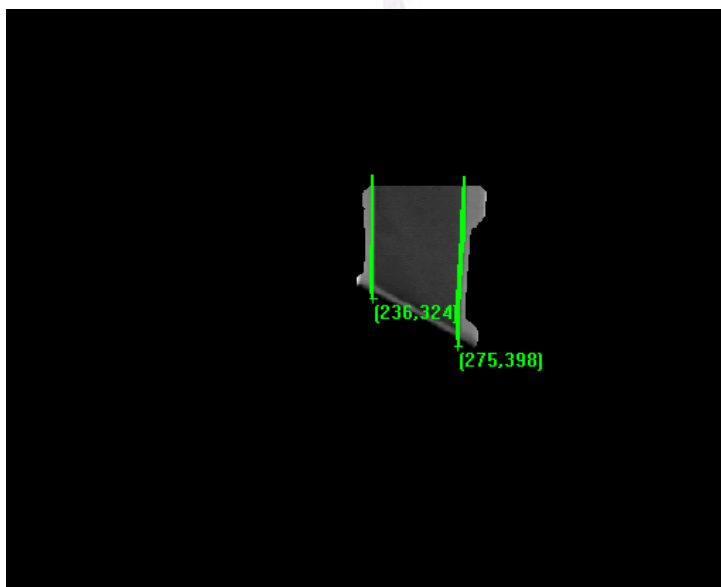


16

17



18



19

圖 4.3 演算法步驟顯示結果

其中圖 4.3-11~17 張圖是圖 4.3-10 影像有資訊部分的放大，從上述影像演算法可以求得實際自走車與障礙物的座標。接下來要說明的是如何使用求得的距離作閃避的動作。

4-2 實際測試

實際測試的環境的預設條件以及參數為：(1)鏡頭下俯 16.5° ，(2)鏡頭放大原始影像 4 倍，(3)轉彎時左右輪的轉速比 1:2，也就是固定轉動角度，(4)事先測試最適迴轉半徑公式，

$$R < 32.406 * \log_{10}(w_i) \quad (4-1)$$

其中

R 代表閃避距離臨界值，也就是最佳的轉彎距離

w_i 代表障礙物寬度

4-2-1 閃避規則

以下是閃避的流程圖，當車子在行走的同時，會判斷前方是否有障礙物，而障礙物判斷的方式是利用抓到兩點障礙物底端點後，利用表 3.1 的 Data Base 計算出寬度 W 、車與障礙物的距離 L 、與障礙物中心的角度 θ 等，並且紀錄以及繪圖至顯示區，下圖 4.4 表示自走車與障礙物關係的俯視圖。

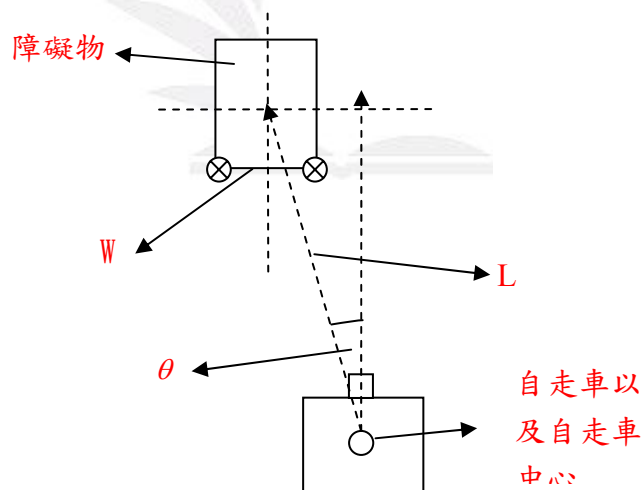


圖 4.4 自走車與障礙物關係俯視圖

在直行路徑當中，如果沒有障礙物則繼續直走，如判斷出障礙的時候，則會繼續判斷障礙物與車子距離 L 是否小於某數值，且車子行走方向與車子中心距離障礙物角度 θ 是否小於某數值，如皆成立則執行閃避動作。

執行閃避會分為左轉或右轉，以圖 4.4 為例，假設圖 4.4 需要執行閃避動作時則會往右轉，由於下列判斷式，假設障礙物在物體左邊，則往右邊閃避，

$$X[P(x, y)] - 319 < 0 \quad (4-1)$$

其中

$X[P(x, y)]$ ，代表障礙物中心點的 X 軸座標值

319 為 CCD，的中心點 (319, 229) 的 X 軸值

由於本專題需要路徑回歸，所以在設計閃避時設定了一個角度積分器，當啟動閃避障礙物同時，以積分的方式在一定的週期加上目前與原先未閃避角度改變的數值

$$\sum_{k=0}^M \theta_k - \theta_0 \quad (4-2)$$

其中

M 為累加歸零的時間， $M \in Z$

當中有一個重要的判斷方塊是預估路徑，下一步的路徑會根據紀錄的障礙物點判斷是否執行路徑回歸，假使角度以及距離都符合安全範圍，如符合條件則執行；如不行則會決定直走或者繼續閃避。而如果直走或是閃避則積分器會持續累加；反之，積分器會反累加，如最後將累積的數值反累加至 0，則路徑將會回歸。下圖 4.5 為自走車路徑回歸示意圖，以及圖 4.6 自走車閃避判斷的方塊圖。

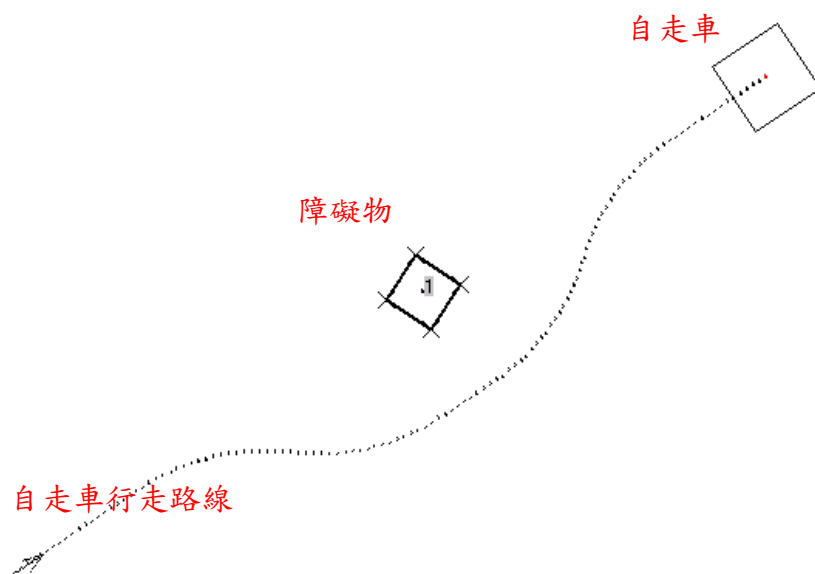


圖 4.5 自走車路徑回歸示意圖



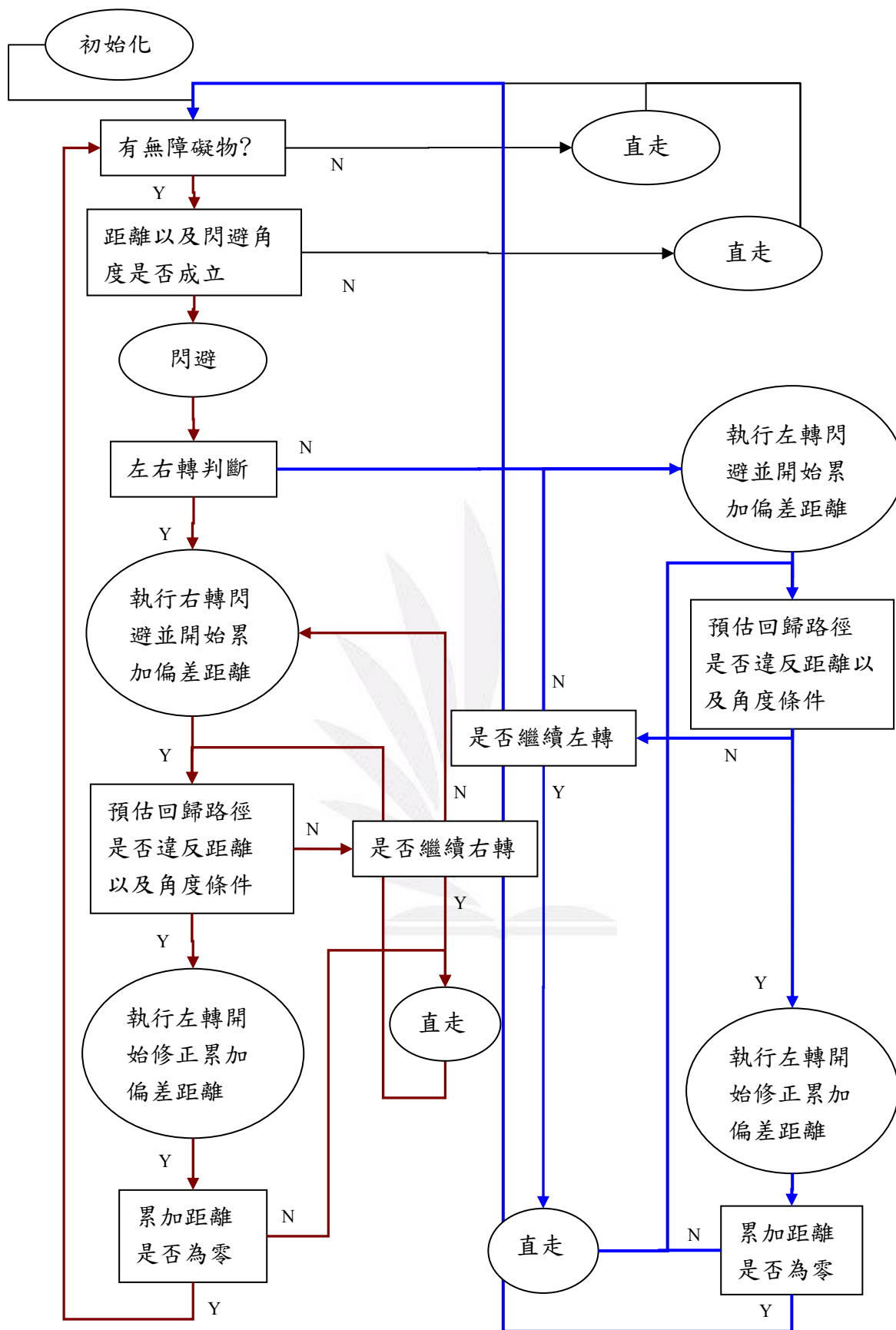


圖 4.6 自走車閃避流程方塊圖

4-2-2 實驗結果

下圖 4.7 圖 4.8 為路徑記錄圖以及實驗拍攝結果

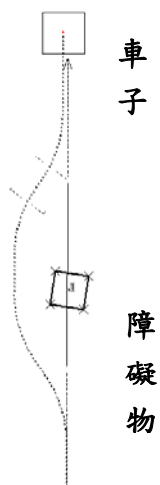


圖 4.7 模擬路徑圖

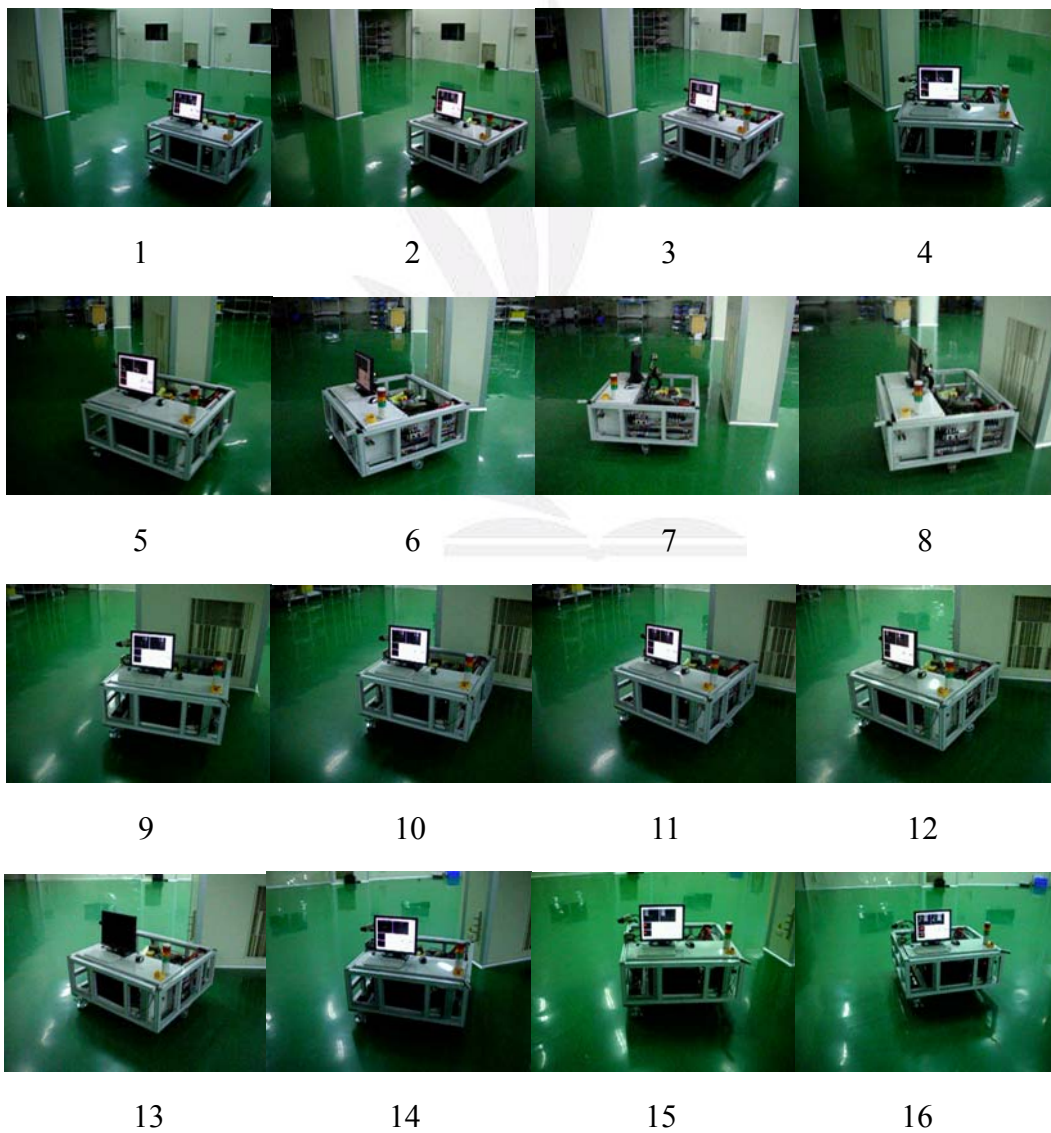


圖 4.8 自走車實際測試結果

4-3 結果討論

整個影像處理在時間方面，處理 1000 張的時間平均約 28.3 秒，每秒大約能夠處理 25 到 35 張照片。另外，調整完參數，包括對比度、二值化的閾值等等之後，在準確率方面，靜態取樣經由 -45 度、45 度、0 度角度切入障礙物以距離 1.5 公尺拍攝障礙物，或是動態行進拍攝，判斷點皆大於三點以上的張數，靜態處理 1000 張圖片誤判張數約為 30 到 50 張，準確率約為 96%；而在行走狀態處理 1000 張圖片誤判張數約為 80 到 120 張，準確率約為 90%。



第五章 結論與未來展望

5-1 結論

在本論文中，吾人利用影像處理應用於自走車即時障礙物偵測工作。影像處理完後檢測出障礙物與地面的交界位置，如此便可以取出障礙物的與自走車的相對關係，使自走車能在行進之間能取得環境資訊。

最後利用已經建立好的距離轉換 Data Base，當自走車行進時，能夠以此Data Base作為即時換算判斷障礙物距離的依據，實驗中也發現並證實這是一個可行的方法。

5-2 未來展望

這是學生第一次接觸到影像處理，加上配合硬體上以及軟體上的研究，有許多的功能可以在日後的擴充加以改善，使影像自走車能有更多功能的運用。如：

1. 在操作介面上：建立多種控制方式，便於驗證用。例如搖桿控制、自動復位、指定位置移動等。
2. 在影像判別上：本研究只能分辨出物體影像的資訊，但對於卻無法判斷是何種什麼物體，未來可以加入一些深入的影像判別演算法，根據物體的特徵來分辨是什麼物體。
3. 在感測器上：本研究所採用的 CCD 只有一個，對於深度資訊以及世界座標，只能針對特定位置障礙物以及建資料表來判斷距離。未來可以採用雙 CCD，利用立體視覺的觀念求得更精確的數值。另外可以加入另外的感測元件如超音波距離感測、加速規、陀螺儀等，取得額外的資訊如距離、轉動角度、行走速度以及加速度，以利於更多的控制和操控精準度。
4. 在遠端遙控上：在未來可以加入遠端遙控操作，使 CCD 所看到的畫面，可經由無線傳輸傳回使用者控制端，使使用者清楚知道遠端資訊，並藉由遠端進行控制，而不必人車一起操作。

參考文獻

- [1] Hill, J. and Park, W.T., “*Real Time Control of a Robot with a Mobile Camera 9/e*”, Washington, D.C., 1997.
- [2] Ramesh, J. , Rangachar, K., and Brian G., “*Machine Vision*”, McGRAW-HILL, 1995
- [3] Rafael C. Gonzalez and Richard E. Woods, “*Digital Image Processing 2/e*”, Addison Wesley Publishing , 2002.
- [4] Trucco, E. and Verri, A. ,” *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall”, New Jersey, 1998.
- [5] <http://www.msu.edu/~redder/colors.html>, Color's RGB.
- [6] 黃敏鋒，人臉追蹤法應用於監控系統之研究，碩士論文，成功大學電機工程學系，民國 91 年。
- [7] 吳至仁，即時障礙物偵測/定位及標誌辨識，碩士論文，成功大工程科學系，民國 90 年。
- [8] 陳饒中，自走車系統之障礙物檢測與位置資料之建立，碩士論文，成功大學工程科學系，民國 90 年。
- [9] 陳浥菁，自走車 DSP 嵌入式影像系統之實現，碩士論文，成功大學航空太空工程研究所，民國 95 年。
- [10] 蕭賢德，電腦視覺系統應用於自走車系統障礙物檢測之研究，碩士論文，國立成功大學工程科學系，民國 89 年。
- [11] 郭桓甫，應用即時影像辨識技術於物體追蹤之無線遙控自走車的設計製作，碩士論文，成功大學航空太空工程研究所，民國 95 年。
- [12] 鄭芝怡，利用電腦視覺在室內走廊環境做障礙物偵測與距離估測，碩士論文，交通大學電機控制工程學系，2002。

- [13] 傅培耕，即時物體追蹤之立體視覺導引自走車，碩士論文，中原大學機械工程學系，2002。
- [14] 蔡博智，影像追蹤方法應用在監控系統之研究，碩士論文，中原大學機械工程學系，2002。
- [15] 繆紹綱，”數位影像處理－活用 MATLAB”，全華，2000。
- [16] 黃文鈺，”Delphi 7 徹底研究”，文魁，2002.
- [17] 洪國勝，”Delphi 7 程式設計(含資料庫)快樂上手”，全華，2003。
- [18] 中村拓男，”Delphi 於繪圖與圖形處理上的實習應用”，博碩，2002。
- [19] Ray, L.原著，劉琍綾編譯，”DELPHI 指令參考手冊”，美商歐萊禮，2001。

