

國立交通大學

資訊科學與工程研究所

碩士論文



基於立體視覺之即時指向系統之誤差分析

Error Analysis of a Real-time Vision-based Pointing System

研究生：蔡易達

指導教授：莊仁輝 教授

中華民國九十六年六月

基於立體視覺之即時指向系統之誤差分析

Error Analysis of a Real-time Vision-based Pointing System

研究生：蔡易達

Student：Yi-Ta Tsai

指導教授：莊仁輝

Advisor：Jen-Hui Chuang

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

基於立體視覺之即時指向系統之誤差分析

學生：蔡 易 達

指導教授：莊 仁 輝 博士

國立交通大學

資訊科學與工程研究所

摘 要

在本論文中，我們基於立體視覺，利用兩台攝影機實作一套即時指向系統。此系統避開以往複雜的三維重建方式，改以影像分析技術，平面投影轉換及三維幾何等數學理論，作為系統實作的基礎。而與傳統三維重建方法一樣，本系統亦存有誤差的問題，故本論文在介紹系統運作流程後，先檢視本系統的所有誤差來源，了解所擷取影像特徵的誤差範圍。而後，再由已知影像上的誤差範圍，探討影像誤差與最終重建誤差之間的關係，也探討了在最大影像誤差的情況下，重建點的誤差上限會如何分佈。最後則介紹我們嘗試在系統中降低誤差的方法，並探討這些方法對靜態與動態指向點重建精確與穩定度之改進。由實驗結果可知，我們將能根據誤差分析之結果，來設計使系統能精確且穩定運作的使用者介面。

Error Analysis of a Real-time Vision-based Pointing System

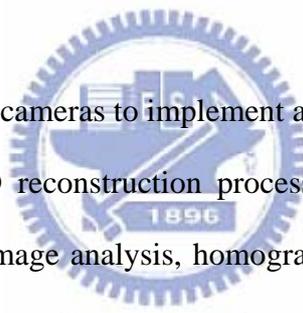
Student : Yi-Ta Tsai

Advisor : Dr. Jen-Hui Chuang

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT



In this thesis, we use two cameras to implement a real-time vision-based pointing system. Without complex 3D reconstruction processes, the implementation of the pointing system is based on image analysis, homography, and 3D geometry. Similar to 3D reconstruction approaches, the reconstruction errors also exist in the proposed pointing system. Based on the procedure of the system, we first study the error sources, and the range of extraction errors of image features. Then, we investigate the relation between image errors and the ultimate reconstruction errors, and try to find the worst-case error range in the latter with respect to a fixed magnitude of the former. Finally, we develop some methods to reduce the error range for both static and dynamic pointing actions. Based on the derived characteristic of error range of reconstruction, as observable from the experimental results, pointing systems with more accurate and stable user interface can be developed.

誌謝

本論文得以順利完成，首先必須要感謝我的指導教授莊仁輝老師。因為老師在研究上的耐心教導，讓我對研究的領域更加了解；在生活上的叮嚀及勉勵，更成為我未來生活的助力。再者必須要感謝三位口試委員，李嘉晃教授、賴飛熊教授、及雷欽隆教授的建議及指導，讓本論文的內容更加完整、更加充實。

謝謝實驗室的所有伙伴，尤其是同居的佳昱、怡伶、郁婷、光兆與宗穎。由於大家的相互扶持及鼓勵，使得大家得以一起順利畢業。也謝謝學弟妹及學長姐們，由於你們帶給大家的歡樂，使得實驗室的氣氛一直很和樂，也因此使得在研究的生活並不會孤單及煩悶。在此要特別感謝泓宏及肇宏學長，在我程式及學習上有任何問題時，你們總是不吝惜的指導，讓我在研究上可以更上一層樓。

謝謝所有資科 94 的伙伴，從大學以來大家的一同學習，每年節日的聚會，凝聚了一股令人稱羨的向心力。也要特別謝謝俊毅、慶臻、明潛、銘丞、冠婷等同學在生活上的陪伴及學業上的切磋。1896

也謝謝我的社團伙伴，尤其是玉芳、承穎、及柏翰。感謝你們的陪伴與關心，點滴我皆感受在心裡。最後要感謝從高中一直以來的好友，建今、育瑋、以恆及家瑋等，認識你們這群好友真的很開心。

最後，感謝一直以來支持我、為我操心的家人，以及所有幫助過我的朋友們，僅以此篇表達我誠摯的謝意。

目錄

中文摘要.....	I
英文摘要.....	II
誌謝.....	III
目錄.....	IV
表目錄.....	VI
圖目錄.....	VII
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 相關研究.....	2
1.3 系統流程.....	4
1.4 各章簡介.....	5
第二章 系統建置.....	6
2.1 硬體環境.....	6
2.1.1 系統運作空間.....	6
2.1.2 軟硬體配備規格.....	7
2.1.3 實驗器材.....	8
2.2 系統運作.....	8
2.2.1 影像擷取.....	8
2.2.2 指向物追蹤辨識.....	10
2.2.3 指向點重建.....	12
2.3 總結.....	13
第三章 系統誤差來源之探討.....	14
3.1 座標量測造成的誤差.....	14
3.2 取像造成的誤差.....	15
3.3 利用 PCA 方法造成的誤差.....	16
3.4 經由 Homography 計算造成的誤差.....	19
3.5 總結.....	22
第四章 影像誤差與重建誤差之關係.....	23
4.1 相關數學理論介紹.....	24
4.1.1 平面投影轉換.....	24
4.1.1.1 齊次線性解.....	24
4.1.1.2 求解平面投影轉換矩陣.....	25
4.1.2 相關三維幾何運算介紹.....	26
4.1.2.1 三維空間中平面方程式表示法.....	27
4.1.2.2 三維空間中三平面求交點.....	27
4.2 由不同誤差半徑與不同角度做探討.....	28

4.2.1 一維雜訊，垂直與水平方向之探討.....	28
4.2.2 固定距離，由不同方向作探討.....	30
4.3 在影像最大誤差的情況下求重建點的誤差上限.....	32
4.4 應用.....	39
4.4.1 由使用者不同站位來探討.....	40
4.5 結論.....	48
第五章 靜態與動態指向誤差之探討.....	49
5.1 部分系統實作設計介紹.....	50
5.2 減少影像與重建誤差的四種方法.....	50
5.2.1 平均重建點以求減少誤差.....	51
5.2.2 平均相同時間的影像點以求減少誤差.....	52
5.2.3 平均多點影像點以求減少誤差.....	53
5.2.4 平均前一秒影像點以求減少誤差.....	53
5.3 四種方法在靜態指向上的實驗.....	53
5.4 四種方法在動態指向上的實驗.....	60
5.5 結論.....	62
第六章 結論與未來展望.....	63
6.1 結論.....	63
6.2 未來展望.....	63
參考文獻.....	65



表目錄

表 1.1	D. Laberge.[3]所提出的系統效能表。.....	3
表 2.1	實作系統所使用的硬體規格表。.....	7
表 4.1	以 90 度倍數角、15 度倍數角與切點所得的重建點誤差上限數據表。 (單位：像素)	38
表 4.2	三次實驗所得的重建點誤差上限範圍數據表。(單位：像素)	43
表 5.1	四種方法所得的重建點分佈範圍數據表。(單位：像素)	57
表 5.2	四種方法所得的重建點分佈範圍數據表。(單位：像素)	59
表 5.3	四種方法平均的誤差範圍。(單位：像素)	59



圖目錄

圖 1.1	簡略系統流程圖。.....	5
圖 2.1	系統實驗實景。.....	6
圖 2.2	兩台攝影機在不同視角所拍攝擷取的畫面。.....	7
圖 2.3	實驗中當作指向物的雷射筆。.....	8
圖 2.4	僅僅執行影像擷取程式時，一秒鐘的時間，兩連接埠與使用的影像畫格緩衝區的紀錄檔。其格式自左至右為編號、時間、連接埠編號、影像畫格緩衝區的記憶體位置。.....	9
圖 2.5	執行所有系統程序時，一秒鐘的時間，連接埠與使用的影像畫格緩衝區的紀錄檔。.....	10
圖 2.6	將圖 2.2 先轉至 HSI 色彩模型，再作二值化後的結果。.....	11
圖 2.7	從圖 2.6 找出指向物，並將之標示出來。.....	12
圖 2.8	指向物追蹤辨識流程圖。.....	12
圖 2.9	影像平面與參考平面轉換示意圖。.....	13
圖 2.10	指向系統的投影布幕，可看得到指向點與滑鼠游標。.....	13
圖 3.1	左相機所拍攝得到的連續兩張影像，(a) 為 I_{t-1} ，(b) 為 I_t 。....	15
圖 3.2	將圖 3.1 轉換至 HSI 色彩模型後，相減得到的結果，即 I_{dif} 。....	16
圖 3.3	將圖 2.7 的雷射筆資料點放大。.....	16
圖 3.4	固定 x 座標，看 y 座標的範圍，(a) 為實驗一，(b) 為實驗二。.....	17
圖 3.5	將圖 3.4 (b) 中的其中一群資料放大。.....	18
圖 3.6	x 座標為時間軸，看 y 座標的變化。(a) 為實驗一，(b) 為實驗二。.....	18
圖 3.7	左右兩攝影機的視角影像，紅色圈起來的部份為特徵點。.....	19
圖 3.8	參考平面的八個特徵點座標位置示意圖。 $P_1 \sim P_8$ 為八個特徵點。 C_1 與 C_2 則分別代表左右兩相機中心的位置。.....	20
圖 3.9	指向平面（布幕）的座標示意圖。.....	21
圖 3.10	指向平面與螢幕間的轉換示意圖。.....	21
圖 4.1	由指向物至指向點之三維幾何示意圖。.....	23
圖 4.2	改變 I_{LS} 的座標，讓雜訊分佈在垂直方向。(a) 為 I_{LS} 的座標， (b) 為重建的滑鼠座標。.....	29
圖 4.3	改變 I_{LS} 的座標，讓雜訊分佈在垂直方向。(a) 為 I_{LS} 的座標， (b) 為重建的滑鼠座標。.....	30
圖 4.4	改變 I_{LS} 的座標，觀察在不同角度上，對重建點的影響。(a) 為 I_{LS} 的座標，(b) 為重建的滑鼠座標。.....	31
圖 4.5	改變 I_{RS} 的座標，觀察在不同角度上，對重建點的影響。(a) 為 I_{RS} 的座標，(b) 為重建的滑鼠座標。.....	32

圖 4.6	將 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 90° 倍數角取樣的分佈圖。.....	33
圖 4.7	I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 90° 倍數角取樣的所有重建點分佈圖。.....	33
圖 4.8	I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 90° 倍數角取樣，重建點誤差上限圖。 (a) x 、 y 座標範圍分別設定在 $0\sim640$ 與 $0\sim480$ 之間，(b) 將重建點誤差上限放大觀察。.....	34
圖 4.9	將 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 15° 倍數角取樣的分佈圖。.....	35
圖 4.10	I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 15° 倍數角取樣，重建點誤差上限圖。 (a) x 、 y 座標範圍分別設定在 $0\sim640$ 與 $0\sim480$ 之間，(b) 將重建點誤差上限放大觀察。.....	35
圖 4.11	I_{LS} 與 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求內公切線與切點，綠色與紅色點為切點，線則為兩內公切線。(a) 為 I_{LS} 與 I_{LE} 的模擬圖，(b) 為 I_{RS} 與 I_{RE} 的模擬圖。.....	36
圖 4.12	I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求內公切線與切點，重建點誤差上限圖。(a) 將重建點誤差上限放大觀察，(b) x 、 y 座標範圍分別設定在 $0\sim640$ 與 $0\sim480$ 之間。.....	36
圖 4.13	比較重建點的誤差上限，將在 90 度倍數角上取樣、 15 度倍數角上取樣，及以切點來求重建點誤差上限畫在一起。.....	37
圖 4.14	I_{LS} 與 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求外公切線與切點，綠色與紅色點為切點，線則為兩外公切線。(a) 為 I_{LS} 與 I_{LE} 的模擬圖，(b) 為 I_{RS} 與 I_{RE} 的模擬圖。.....	38
圖 4.15	(a) I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求外公切線與切點，重建點誤差上限圖，(b) 將內公切線與外公切線造成的誤差畫在一起。..	39
圖 4.16	第一次實驗的站位。(a) 左影像視角，(b) 右影像視角。.....	40
圖 4.17	第一次實驗所計算出來的誤差上限。.....	41
圖 4.18	相對於第一次實驗，往右移約 80 公分的站位。(a) 左影像視角，(b) 右影像視角。.....	41
圖 4.19	第二次實驗所計算出來的誤差上限。.....	42
圖 4.20	相對於第一次實驗，往左移約 80 公分的站位。(a) 左影像視角，(b) 右影像視角。.....	42
圖 4.21	第三次實驗所計算出來的誤差上限。.....	43
圖 4.22	第一次實驗指向物在影像上的位置。(a) 左影像視角，(b) 右影像視角。.....	44
圖 4.23	第一次實驗的重建點分布情況與誤差上限。.....	44
圖 4.24	第一次實驗左右影像做二值化後的結果。.....	45
圖 4.25	第二次實驗指向物在影像上的位置。(a) 左影像視角，(b) 右影像視角。.....	45
圖 4.26	第二次實驗的重建點分布情況與誤差上限。.....	46

圖 4.27	第二次實驗左右影像做二值化後的結果。.....	46
圖 4.28	第三次實驗指向物在影像上的位置。(a) 左影像視角，(b) 右影像視角。.....	47
圖 4.29	第三次實驗的重建點分布情況與誤差上限。.....	47
圖 4.30	第三次實驗左右影像做二值化後的結果。.....	48
圖 5.1	獲取影像時間點的分佈圖。.....	49
圖 5.2	將重建點以二、三、四、五點平均，再設定給滑鼠座標。(a) - (d) 分別為二、三、四、五點的平均。藍色與紅色分別表示重建點的 x 、 y 座標， x 軸表示第幾次的重建。.....	51
圖 5.3	將 I_{LS} 、 I_{LE} 、 I_{RS} 、 I_{RE} 以二、三、四、五點平均，拿平均值作重建，再設定給滑鼠座標。(a) - (d) 分別為二、三、四、五點的平均。藍色與紅色分別表示重建點的 x 、 y 座標，而綠色與黑色點則分別表示 I_{LS} 、 I_{LE} 、 I_{RS} 、 I_{RE} 的 y 座標值， x 軸表示第幾次的重建。.....	52
圖 5.4	系統畫面截圖。分別為左右兩攝影機視角的畫面。而圖中我們以藍色線畫上的，是找到的指向物。.....	54
圖 5.5	利用 5.2.1 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	54
圖 5.6	利用 5.2.2 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	55
圖 5.7	利用 5.2.3 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	55
圖 5.8	利用 5.2.4 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	56
圖 5.9	系統畫面截圖。分別為左右兩攝影機視角的畫面。.....	57
圖 5.10	利用 5.2.1 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統.....	57
圖 5.11	利用 5.2.2 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	58
圖 5.12	利用 5.2.3 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	58
圖 5.13	利用 5.2.4 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。...	59
圖 5.14	利用 5.2.1 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。.....	60
圖 5.15	利用 5.2.2 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。.....	60
圖 5.16	利用 5.2.3 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。.....	61

圖 5.17 利用 5.2.4 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。
.....61



第一章 緒論

1.1 研究動機

近幾十年來，「人機互動」是相當熱門的研究領域，而人機介面的應用更是廣泛。舉凡指向系統 (Pointing System)、虛擬實境 (Virtual Reality)、人臉辨識、手勢與肢體語言的辨識等，皆屬於其範圍。在本論文中，主要則著重在「指向系統」的部分。

「指向系統」近幾年發展出很多應用方向，一般較為廣泛應用的「簡報系統」，即是一例。通常在大型的會議上，都會使用簡報系統，演講者透過雷射筆射出紅點，此系統即會偵測此紅點，來判斷指向的位置並且輸入到電腦中，當此點停留在事先定義的區域時，代表某些特定的指令輸入(如上下換頁)。另外，「智慧型家電操控系統」也是指向系統的應用之一，先藉由分析場景空間，知道家電設備與空間的相對位置，然後透過指向來操控家電系統。例如當使用者指向電視機，則電視機會被開啟，如果對著電視機比劃其他手勢，則可以上下轉台。

不僅如此，指向系統還有其他許多不同的應用面，如電玩遊戲。在現今許多遊戲機台中，有些是利用感應器來感應玩家的動作，以辨別玩家的動作軌跡。其所須搭配的設備，包括搖桿、手槍等，有些還是較為昂貴。如果可以以立體視覺的概念作定位，搭配影像處理的技術，則像射擊遊戲，只需要知道玩家手的朝向，加上射擊的動作，即可完成，既不需要昂貴的設備，也可以由不同手勢與遊戲有不同的互動。而像是「導覽系統」，現在幾乎都是讓使用者以觸碰感應、電腦滑鼠操控、或播放影片，來進行瀏覽。但當需要對一批遊客同時進行解說時，解說員如果使用觸碰感應式面版操控，則常會擋住使用者視線。如果使用指向系統，則解說員即可站在較遠的位置，以手的方向作定位，既不會擋住遊客視線，亦可以方便的和硬體設施作互動。

從上述的應用情境中可知道，根據不同的目的，系統所需要的準確度亦不相

同。在簡報系統中，如果演講者需透過指向系統來操控滑鼠，在螢幕的正確位置下指令，則準確性就必須要高，否則將無法正確地下指令；射擊遊戲亦是需要較高準確度的。至於家電操控系統，使用者只需要知道這些家電在環境中的相對位置。由於家電體積較為龐大，所以只要朝向大略正確，即可作出正確的操控，其準確度就不需要這麼高了。

無論何種指向系統，首先都必須要求指向的正確性，再者系統必須要穩定。當系統能穩定且正確地判斷朝向後，系統即能追隨使用者的指向，而正確地移動。

1.2 相關研究

在指向系統的相關研究中，大致上是利用二維平面間的轉換關係，或以立體視覺找出三維空間相對位置的方法，來計算出指向點。而在此節中，將介紹幾篇與指向系統相關研究的論文，說明其系統的方法，也探討其系統效能與誤差。

R. Sukthankar et al.[1]、J.-F. Lapointe. et al.[2]、D. Laberge et al.[3]提出的雷射筆簡報系統，都只需要利用一台相機，是基於二維平面轉換的系統。在這些系統中，都是利用平面投影轉換原理，來取得相機平面、投影平面與顯示平面間的對應關係，然後把這些對應關係記錄下來，以供系統後續使用。當系統取得有包含雷射點的影像時，經由影像處理的技術，找出雷射點所在位置，然後再算出其在顯示平面的位置。當雷射點停留在事先定義的區域時，即代表某些特定的輸入指令。在這些系統中，雷射光點在影像上的亮度必須要比背景亮，否則不易偵測雷射點。而在[3]中提及的相關系統效能，則如表 1.1 所示，由於直接偵測雷射點的關係，所以其準確度高，誤差範圍只在 ± 3 個像素點；當雷射筆是放置在固定的平面上時，誤差更縮小至 ± 1 個像素點，且平均每秒約跳動一次。而相機與投影平面間的距離、相機與投影機之間的角度、相機的解析度等，皆是影響準確度的原因之一。

表 1.1 D. Laberge.[3]所提出的系統效能表。

Parameter	Result
Range	Line of sight
X accuracy	± 2 screen pixels typical
Y accuracy	± 3 screen pixels typical
Latency	52ms \pm 8ms average
Update	60Hz \pm 1Hz (Overall) 605Hz \pm 5Hz (Algo. only)
Jitter	± 3 screen pixels typical

在指向系統中，也有很多作法是利用手來當作指令或判斷朝向。A. A. Argyros et al.[4]所提出的系統中，先由影像中的色彩分佈 (color distribution) 來判斷出包含膚色的部分，再利用貝式分類器 (Bayesian classifier) 來判斷出影像中手的座標，最後求出指尖的位置。然後定義一些特殊的手勢動作 (如手掌打開伸出五指，代表按下滑鼠左鍵)，藉以操控滑鼠，而此方法必須同時使用兩隻手。此系統亦可使用兩台相機，根據 epipolar constraint 找到手在兩相機影像中的相對位置，再重建出在空間中的位置。對於後者這種使用兩台相機，使用者只需使用一隻手即可。

S. Sato et al.[5]所提出的系統，是利用手指尖與指根的連線方向，來當作朝向。在系統中，先從影像中找出指尖，然後利用手指頭的樣板 (template) 來找出指根的位置，有了指根與指尖的位置後，利用三維重建的方法去求出指向位置。Y. P. Hung et al.[6]所提出的系統，則是考慮利用兩種方法來得到朝向，一種是利用手指的些許像素點求其朝向，另一種則是利用指尖與眼睛的連線方向，來當作指向的方向。而從此篇論文的描述中知道，用第一種方法作定位，很容易被雜訊影響。而利用指尖與眼睛的連線方向的話，效果則較好，誤差約在一個像素點左右。而由於利用 IIS (Institute of Information Science) head 這套立體視覺系統，當指尖已經到達影像的邊界時，可以藉由回饋，更新相機參數，進而控制相機的朝向，如此一來，在相機的視角範圍內，皆可得到正確的朝向。

而 E. Hosoyam et al.[7]所提出的系統則是家電操控應用的一例。在此篇論文

中，他們必須事先切割立體空間，將空間分割成好幾塊立方體，所有的家電設備則位在這些立體空間中。在系統執行前，必須事先知道這些家電設備的位置。他們先在影像中找出包含膚色的區域，然後辨別出臉部與手臂的位置，利用事先定義好臉部重心與肩膀的距離，再根據已求得的臉部重心，即可算出肩膀的座標位置。由於他們利用立體相機（stereo camera）的緣故，所以可以取得影像的深度資訊，根據深度影像就能算出肩膀與手臂在三維空間中的座標位置。而肩膀與手臂的連線方向，即定義為朝向。若在朝向上有家電設備的話，則可以根據遠端控制器來控制該家電。而整體的系統反應時間，平均為 2.74 秒，標準差為 1.34 秒。

從以上我們介紹的論文中不難發現，[1][2][3]是直接偵測雷射點，當作指向點，[4]是偵測手勢，來操控滑鼠；[5][6]則分別是以指尖與指尖和眼睛的連線方向，定義指向，[7]則是以肩膀與手臂的連線方向來定義指向。在本篇論文中，則是利用三維空間平面相交的觀念，先利用兩平面相交求指向方向，再與一目標平面相交求指向點。所以，本論文所使用的方法與以上所介紹的論文有很大的差異。



1.3 系統流程

有別於在前一節所提及的方法，本論文中的系統是參考[8][9]的作法，以立體視覺為基礎，簡略系統流程圖如圖 1.1 所示，利用影像擷取卡擷取兩台攝影機連續輸入的畫面，進行特定指向物的追蹤，並且正確地判斷出指向物所指的方向。而後，再根據事先量測的空間座標資訊，以投影幾何轉換的方式，將指向物所指的方向重建在三維空間座標上，再與目標平面交出指向點。最後將得到的指向點指定給滑鼠游標，以此方法來判斷指向點是否準確。

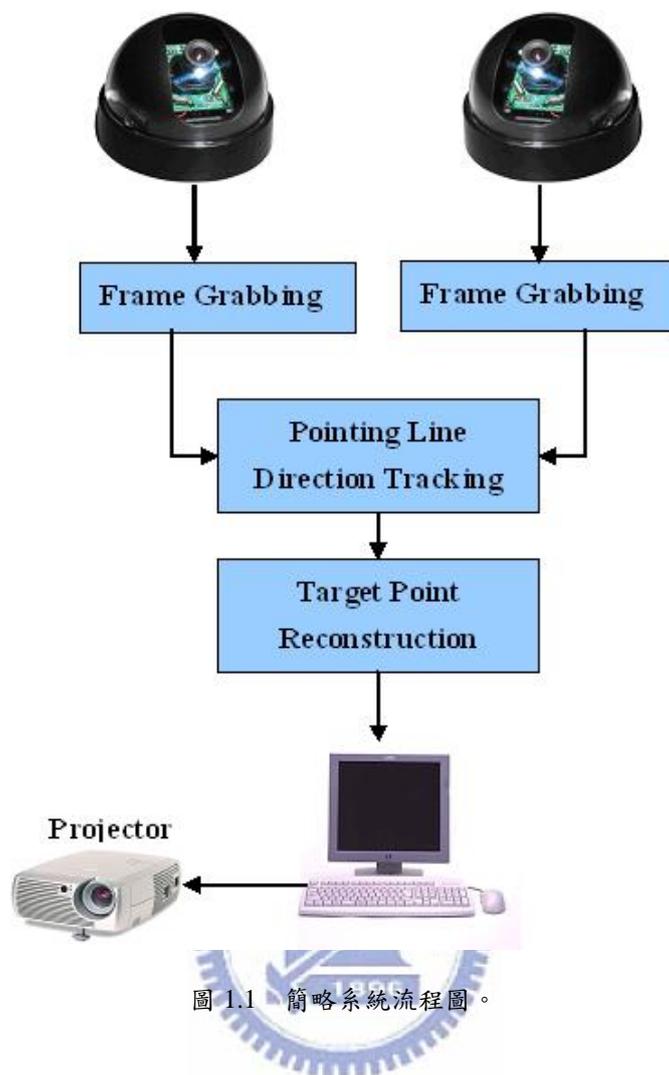


圖 1.1 簡略系統流程圖。

1.4 各章簡介

本論文總共分為六章，除了本章為緒論外，第二章為系統建置，將介紹本論文所用的軟硬體設備與環境，並且說明本系統的細部運作流程。根據[8][9]的實驗結果得知，指向物追蹤結果的好壞會影響系統的重建結果，故為了使重建結果穩定，我們參考[9]的實驗結果，調整 HSI 數據。另外，從[8][9]得知，系統會有一些誤差產生，故在第三章將詳細探討系統誤差來源，以及我們如何減少誤差。第四章將介紹本系統在不考慮誤差來源的情況下，影像上的誤差與重建出指向點的誤差間的關係。第五章則介紹四種不同方法來計算最後的指向點位置，並對靜態指向與動態指向分別作探討，以及這四種不同方法的結果比較。最後則會為本篇論文作個總結與本研究未來可應用的方向。

第二章 系統建置

本論文研究之目的，是以立體視覺為基礎，來開發的一套即時指向系統。在本章中，會先對於系統實施的工作環境，以及所使用的軟硬體配備規格與實驗器材做簡單的介紹；然後再從影像擷取至最後的指向點重建，一一描述本系統內部的運作流程。

2.1 硬體環境

2.1.1 系統運作空間



圖 2.1 系統實驗實景。

本系統的工作環境分為兩部分，包含中控室和實驗場所。中控室中包含本系統程式的電腦，圖 2.1 為我們的實驗場所外觀，包含有投影機、投影布幕，遠端連接中控室電腦的筆記型電腦，以及兩台攝影機，一台在使用者站位的左後方，另一台在使用者站位的右前方，視角大小可參考圖 2.2。由於是以立體視覺為基礎，所以使用者站位必須在兩視角皆能拍攝得到的地方，否則系統將無法運作。



圖 2.2 兩台攝影機在不同視角所拍攝擷取的畫面。

2.1.2 軟硬體配備規格

在本系統中，我們所使用的硬體配備，包含兩台 PTZ 攝影機，分別設置在使用者的左後與右前方，另外在中控室的電腦主機中，插有一張影像擷取卡，此擷取卡支援四個視訊訊號輸入。系統是在 Windows XP 作業系統上開發，程式平台為 Microsoft Visual Studio 2005，所使用的程式語言為 C#，相關規格整理於表 2.1。

表 2.1 實作系統所使用的硬體規格表。

項目	規格
PTZ 攝影機	D7720 高速球型攝影機 P/T/Z : 360°/90°/16X Electronic Shutter : 1/60~1/30k sec Minimum Illumination : 1 lux
影像擷取卡	ADLINK Angelo RTV 24 影像擷取卡 Format : NTSC Resolution : CIF (320×240) Frame rate : 30fps
主機	PC CPU : Pentium4 3.2GHz RAM : 2.0GB OS : Windows XP SP2 Platform : Microsoft Visual Studio 2005

2.1.3 實驗器材

我們是利用一支套有特殊顏色紙張的雷射筆，當作我們的指向實驗器材，如圖 2.3 所示，長度約 40 公分，寬度約為 2 公分。之所以會選用雷射筆來當作我們的指向物，是因為雷射筆所射出的紅色點，可當作我們最後重建點的目標點，也就是希望最後的重建點可以和雷射筆所射出的紅色點吻合。由於使用者是將雷射筆整體握住，會妨害我們辨識雷射筆，所以我們將雷射筆套上具有特殊顏色的紙張，以方便辨識，且能加快我們處理計算的速度。



圖 2.3 實驗中當作指向物的雷射筆。

2.2 系統運作

在上節中，已介紹了在本篇論文中所用到的配備。在本節中，我們將完整的介紹系統的運作流程。簡略流程如圖 1.1 所述，而底下就針對這三部分分別作探討。

2.2.1 影像擷取

在實驗場所中的兩台 PTZ 攝影機分別取得視訊訊號後，便將之傳送至影像擷取卡的兩個連接埠 (port)。從表 2.1 可知，影像擷取卡的畫格速率 (frame rate) 為每秒 30 張，即 30fps，所以若系統只有執行影像擷取的部分，則每個連接埠每秒約會傳送 30 張影像。圖 2.4 是當僅執行影像擷取程式時，於一秒鐘的時間之內，兩連接埠與使用的影像畫格緩衝區 (frame buffer) 的紀錄檔。在圖 2.4 中我

們觀察到，使用中的兩個連接埠，各自有兩塊影像畫格緩衝區（連接埠 0：54165504/56066048，連接埠 1：57966592/59867136），而同一個連接埠，其影像畫格緩衝區是會輪替使用的。但兩個連接埠並非一定是交替使用，有可能會發生連續取同一個連接埠上的影像（如：編號 150/151，171/172）。從連接埠得到的影像會放在其中一塊影像畫格緩衝區。在系統運作中我們是利用 callback 的方式，意即當某一畫格緩衝區滿的時候，會產生中斷（interrupt），告知系統可取得此畫格緩衝區的資料，也就是我們的影像，最後我們再利用影像擷取卡所提供的函示庫，即可取得影像。

圖 2.4 是系統只有執行影像擷取的情況，但當執行所有系統程序時，如圖 2.5 所示，每秒每個連接埠平均約只有 10 張影像，雖然如此，其速度大致不會影響整個系統的運作。

120	下午 05:39:45	0 54165504	149	下午 05:39:45	1 59867136
121	下午 05:39:45	1 59867136	150	下午 05:39:45	0 56066048
122	下午 05:39:45	0 56066048	151	下午 05:39:45	0 54165504
123	下午 05:39:45	1 57966592	152	下午 05:39:45	1 57966592
124	下午 05:39:45	0 54165504	153	下午 05:39:45	0 56066048
125	下午 05:39:45	1 59867136	154	下午 05:39:45	1 59867136
126	下午 05:39:45	0 56066048	155	下午 05:39:45	0 54165504
127	下午 05:39:45	1 57966592	156	下午 05:39:45	1 57966592
128	下午 05:39:45	0 54165504	157	下午 05:39:45	0 56066048
129	下午 05:39:45	1 59867136	158	下午 05:39:45	1 59867136
130	下午 05:39:45	0 56066048	159	下午 05:39:45	0 54165504
131	下午 05:39:45	1 57966592	160	下午 05:39:45	1 57966592
132	下午 05:39:45	0 54165504	161	下午 05:39:45	0 56066048
133	下午 05:39:45	1 59867136	162	下午 05:39:45	1 59867136
134	下午 05:39:45	0 56066048	163	下午 05:39:45	0 54165504
135	下午 05:39:45	1 57966592	164	下午 05:39:45	1 57966592
136	下午 05:39:45	0 54165504	165	下午 05:39:45	0 56066048
137	下午 05:39:45	1 59867136	166	下午 05:39:45	1 59867136
138	下午 05:39:45	0 56066048	167	下午 05:39:45	0 54165504
139	下午 05:39:45	1 57966592	168	下午 05:39:45	1 57966592
140	下午 05:39:45	0 54165504	169	下午 05:39:45	0 56066048
141	下午 05:39:45	1 59867136	170	下午 05:39:45	1 59867136
142	下午 05:39:45	0 56066048	171	下午 05:39:45	0 54165504
143	下午 05:39:45	1 57966592	172	下午 05:39:45	0 56066048
144	下午 05:39:45	0 54165504	173	下午 05:39:45	1 57966592
145	下午 05:39:45	1 59867136	174	下午 05:39:45	0 54165504
146	下午 05:39:45	0 56066048	175	下午 05:39:45	1 59867136
147	下午 05:39:45	1 57966592	176	下午 05:39:45	0 56066048
148	下午 05:39:45	0 54165504	177	下午 05:39:45	1 57966592

圖 2.4 僅僅執行影像擷取程式時，一秒鐘的時間，兩連接埠與使用的影像畫格緩衝區的紀錄檔。其格式自左至右為編號、時間、連接埠編號、影像畫格緩衝區的記憶體位置。

102	下午	05:37:11	1	58228736	123	下午	05:37:12	0	56328192
103	下午	05:37:11	0	56328192	124	下午	05:37:12	1	60129280
104	下午	05:37:11	1	60129280	125	下午	05:37:12	1	58228736
105	下午	05:37:11	0	54427648	126	下午	05:37:12	0	54427648
106	下午	05:37:11	1	58228736	127	下午	05:37:12	1	60129280
107	下午	05:37:11	0	56328192	128	下午	05:37:12	0	56328192
108	下午	05:37:11	1	60129280	129	下午	05:37:12	1	58228736
109	下午	05:37:11	0	54427648	130	下午	05:37:12	0	54427648
110	下午	05:37:11	1	58228736	131	下午	05:37:12	1	60129280
111	下午	05:37:11	0	56328192	132	下午	05:37:12	0	56328192
112	下午	05:37:11	1	60129280	133	下午	05:37:12	1	58228736
113	下午	05:37:11	0	54427648	134	下午	05:37:12	0	54427648
114	下午	05:37:11	1	58228736	135	下午	05:37:12	1	60129280
115	下午	05:37:11	0	56328192	136	下午	05:37:12	0	56328192
116	下午	05:37:11	1	60129280	137	下午	05:37:12	1	58228736
117	下午	05:37:11	0	54427648	138	下午	05:37:12	0	54427648
118	下午	05:37:11	1	58228736	139	下午	05:37:12	1	60129280
119	下午	05:37:11	0	56328192	140	下午	05:37:12	0	56328192
120	下午	05:37:11	1	60129280	141	下午	05:37:12	1	58228736
121	下午	05:37:11	0	54427648	142	下午	05:37:12	0	54427648
122	下午	05:37:11	1	58228736	143	下午	05:37:12	1	60129280
					144	下午	05:37:12	0	56328192

圖 2.5 執行所有系統程序時，一秒鐘的時間，兩連接埠與使用的影像畫格緩衝區的紀錄檔。

2.2.2 指向物追蹤辨識

在上一小節中，我們所擷取到的影像是 RGB24 的彩色影像，而因為我們希望不要受到光線太多的干擾，所以我們將之轉換至 HSI 色彩模型 (color model)，將色調 (hue)、飽和度 (saturation)、亮度 (intensity) 三者分離。而因為我們是利用特定顏色 (紅色) 的紙張來做辨識，所以我們可以事先量測該顏色在影像上所呈現出來的 HSI 範圍，進而當作我們已知的條件來使用。由於實驗環境與 [9] 相同，故根據 [9] 得知，在不同的光源下，量測到 HSI 的範圍約分別為 (340°~20°)、(0.5~0.9)、以及 (0.35~0.7)。而因為我們希望所有指向物在影像上的點皆能夠盡量被保留，所以我們放寬條件限制，將色調的範圍設定在 (300°~40°)，飽和度的範圍設在 (0.2~1.0)，亮度的範圍設定在 (0.3~1.0)。

接著我們設一門檻值，將影像二值化，以圖 2.2 為例，二值化後的影像如圖 2.6 所示。由於雷射筆在影像上的像素應該是不間斷的，所以先利用連接物體標記 (connected components labeling) 的方式，將影像中所有不間斷的區塊分別

記錄下來。我們觀察指向物在形狀上有兩項特性，(1) 筆直且非彎曲的物體、(2) 長寬度比例必須使整體呈現細長型。利用這兩項特性，將所有不間斷的區塊做篩選，符合此兩項特性的區塊中，取體積最大者，即可選出包含有指向物的區塊。圖 2.7 就是在這一連串步驟後，從圖 2.2 中找到包含有指向物的區塊。

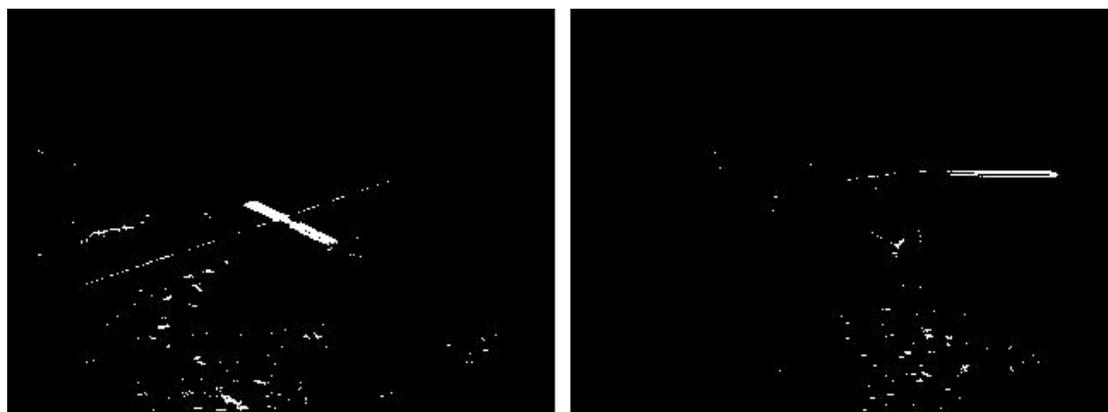
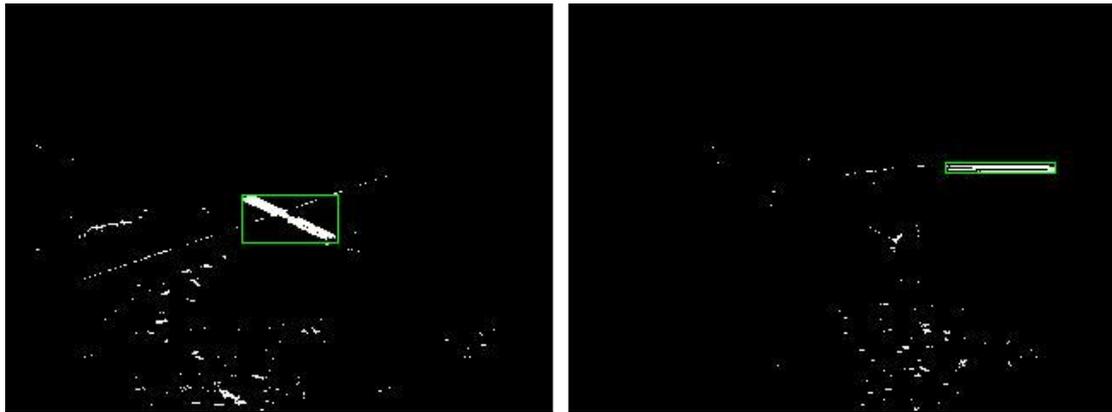


圖 2.6 將圖 2.2 先轉至 HSI 色彩模型，再作二值化後的結果。

接著，我們利用主成份分析法 (principal components analysis, 簡稱 PCA) [10]，求出代表此區塊內所有資料點間關係的共變異數矩陣 (covariance matrix)，再於特徵向量分解 (eigenvector decomposition) 後，選取特徵值 (eigenvalue) 較大者所對應的特徵向量 (eigenvector)，作為指向物在影像上的朝向。因為此朝向可以用直線方程式來表示，所以我們只需要記錄在朝向線上的兩點即可。其作法是紀錄朝向線與區塊邊界的交點，如以圖 2.7 為例，即找出的朝向線與綠色 bounding box 的兩個交點，我們以 $\{I_{LS}, I_{LE}\}$ 、 $\{I_{RS}, I_{RE}\}$ 來分別表示左右兩影像的兩交點。指向物追蹤辨識的流程圖，整理於圖 2.8。



(a)

(b)

圖 2.7 從圖 2.6 找出指向物，並將之標示出來。

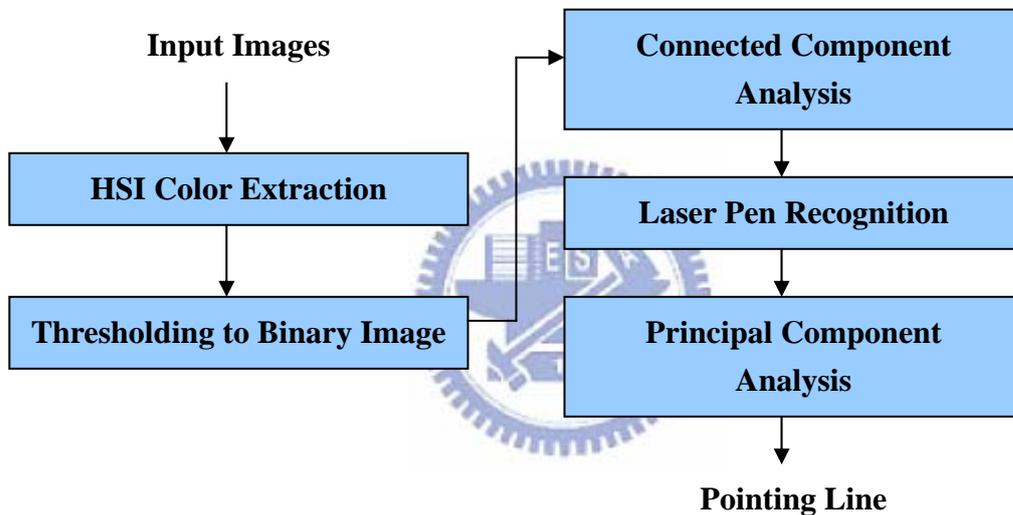


圖 2.8 指向物追蹤辨識流程圖。

2.2.3 指向點重建

至此我們得到指向物在二維影像中的位置，接著則需要將這二維座標轉換至三維空間座標系中，我們利用的是投影幾何轉換 (homography) 的方法。首先我們選取兩攝影機視角照得到的地面為參考平面 (reference plane)，且分別事先算好各影像平面 (image plane) 上四點與參考平面對應四點的轉換關係，如圖 2.9 所示，如此我們就能夠計算出指向物投影在參考平面上的座標，並分別以 $\{R_{LS}, R_{LE}\}$ 、 $\{R_{RS}, R_{RE}\}$ 表示在左右影像上，指向物的直線上兩點，投影在參考平

面上的座標。然後分別各自再跟相機中心 C_L 、 C_R 交出兩平面 π_L 、 π_R ，最後這兩平面再與投影布幕 π_P 三平面共同交出一點，這點即代表我們的指向點，指向點重建步驟與更詳細的討論，可見第四章。

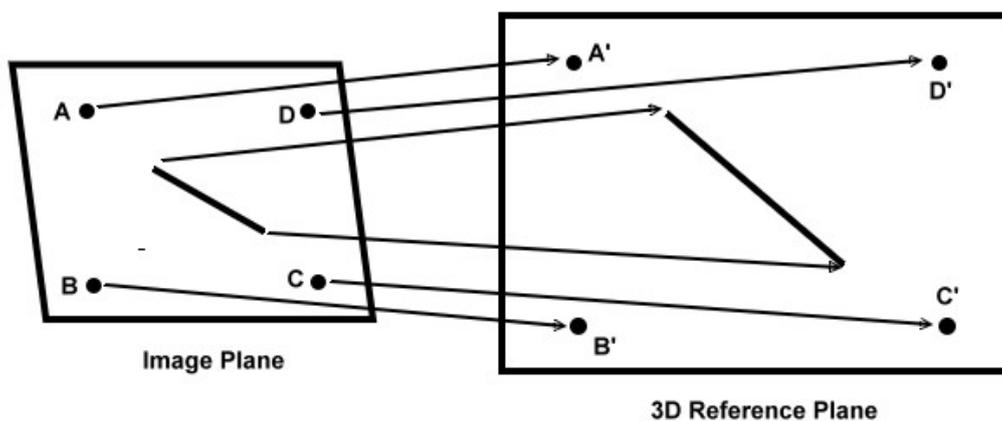


圖 2.9 影像平面與參考平面轉換示意圖。

2.3 總結

在本章中，介紹了我們系統所用的硬體配備，以及系統運作的初步流程。在圖 2.10 中的紅色點即是雷射筆的指向點，在該指向點右下方者為計算所得的滑鼠游標。在此種方法下，系統可以正常的運作，也使得我們有了指向系統的雛形。



圖 2.10 指向系統的投影布幕，可看得到指向點與滑鼠游標。

第三章 系統誤差來源之探討

在本論文所提出的系統中，我們將系統所算出來的指向點指定給滑鼠游標位置，藉此來判別指向點與雷射筆所射出的紅色點是否吻合。然而，在我們實驗之後發現，滑鼠游標位置與紅色點並不吻合，這表示著本系統會產生誤差，此一誤差包含了機械誤差與系統計算誤差。所以在本章中，我們將探討在本系統中所有的誤差來源，以及我們對這些誤差將如何進行改善。但機械誤差並不在我們的研究範圍內，所以在此我們並不作探討。

3.1 座標量測造成的誤差

首先，當我們要將影像平面上的點，重建至三維空間上時，必須先將之投影至參考平面（可參考 2.2.3 節的說明）。由於我們是利用平面投影轉換的方法，所以必須知道實驗場景中的一些特徵點在世界座標系（world coordinate）及在影像平面的座標，才有辦法事先算出影像平面與參考平面間的轉換關係。然而，因為這些特徵點的世界座標數據，我們是以人為測量的方式得到的，所以勢必會有一些誤差產生，而這些特徵點在影像上的座標，也是由人為所估計出來的，所以得到的投影轉換矩陣（homography matrix），並非百分之百的準確。

當我們要作重建指向點時，是在三維空間中，三個平面求其交點，其中兩平面 π_L 、 π_R ，必須分別通過兩相機中心 C_L 、 C_R （可參考 2.2.3 節的說明），一般相機中心都是利用傳統重建的方法，求出內外部參數來得到，但在此我們直接以人為量測的方式估計，以省去作校正的麻煩，卻也因此而產生誤差。再者，將電腦螢幕投影在布幕上，也需要知道投影布幕與電腦螢幕間的轉換關係，因此必須量測投影布幕的世界座標位置。

我們知道，在現實生活中，只要透過人為量測，勢必會有誤差產生，故我們透過多次的量測，希望盡量降低人為因素所造成的誤差。

3.2 取像造成的誤差

從第二章所介紹的內容知道，因為在攝影機端接收到的訊號是類比訊號，而我們從影像擷取卡上所得到的影像是數位訊號，由此可以知道訊號本身已經被做過取樣 (sampling)，所以訊號本身已經有失真了，但這種誤差是無可避免的，所以在此我們並不多作探討。

另外，在實驗場所的光源是一般的日光燈，雖然背景環境皆相同，但因為光源的閃爍，且光源彼此閃爍的時間又並非同步，故在系統流程中，便會產生雜訊。我們以其中一台攝影機為例，對實驗場景連續取得兩張影像，如圖 3.1 所示， I_{t-1} 與 I_t ，將 I_t 與 I_{t-1} 相減，得 $I_{dif} = I_t - I_{t-1}$ ，由於是靜態影像，且實驗場景固定，我們預期 I_{dif} 會得到一張全黑的影像，但如圖 3.2 所示，其結果並不如我們所預期，而會有相異的資料點的產生，亦即所謂的雜訊。經觀察發現，這些雜訊點並非週期性的產生，其分佈點也不一致。



圖 3.1 左相機所拍攝得到的連續兩張影像，(a) 為 I_{t-1} ，(b) 為 I_t 。

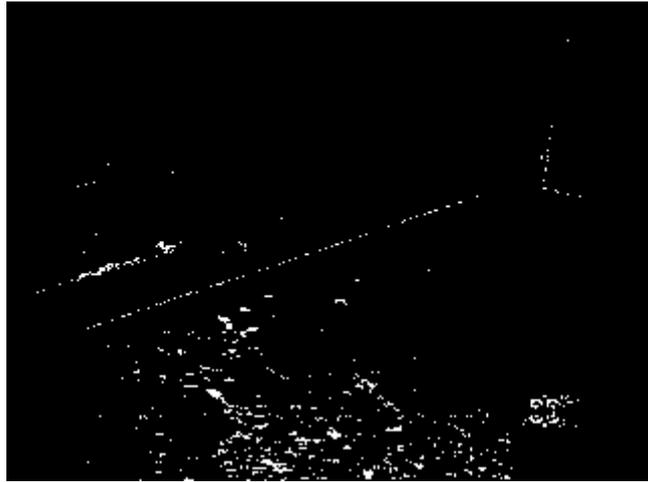


圖 3.2 將圖 3.1 轉換至 HSI 色彩模型後，相減得到的結果，即 I_{dif} 。

除此之外，我們將圖 2.7 (a) 的雷射筆部分，放大來觀察，如圖 3.3 所示，可以發現雷射筆的資料點，並非完全的精確，雖然我們將 HSI 限制條件放寬，希望所有屬於雷射筆的資料點皆能被保留，但我們發現，如紅色線條圈起來的部份，邊緣還是會有些破碎的情況，有些應該屬於雷射筆的資料點卻呈現黑色，如藍色線條圈起來的部份所示。而也因為雜訊的影響，也會使得一些非雷射筆的資料點，卻成為其資料點，這些的誤差皆會對接下來的流程產生影響。



圖 3.3 將圖 2.7 的雷射筆資料點放大。

3.3 利用 PCA 方法造成的誤差

在指向物的追蹤辨識中，我們是利用 PCA 的方法求其指向線，(參考 2.2.2

節的說明)，因為 PCA 的方法可以將所有給予的點，求出其位置之最大變異量方向，而此方向即可定義出指向線的朝向。

由於我們找出雷射筆的資料點是利用連接物體標記的方式，所以只要雷射筆的邊界點，其外圍有和雷射筆上下左右其一方向相鄰 (4-neighbor) 的雜訊點，皆會被誤判成雷射筆的資料點。我們再以圖 3.3 為例，從圖中我們可以發現，在兩箭頭所指處會有疑似非雷射筆的資料點發生，而在 PCA 計算時，這些與雷射筆相鄰的雜訊點皆會被當成雷射筆的資料點作處理。

底下，我們以靜態影像作觀察。首先，我們固定雷射筆的位置，如果沒有任何雜訊的影響，則雷射筆的朝向應當是不會改變的，意即我們得到的交點應該是固定的。但經由實驗發現，交點會一直的改變。為了方便觀察，我們固定 $\{I_{LS}, I_{LE}\}$ 、 $\{I_{RS}, I_{RE}\}$ 的 x 座標，來看 y 座標的範圍。我們作了多次的實驗發現，如圖 3.4 所示，在此我們只列出其中兩次的結果，(a)(b) 分別表示兩次的實驗，圖中 x 軸為 x 座標， y 軸為 y 座標，我們以 (a) 為例，圖上看到的四群分別為 $\{I_{LS}, I_{LE}\}$ 、 $\{I_{RS}, I_{RE}\}$ 的座標。從結果發現， y 座標的值並不會固定，不過其大小會落在一個範圍內，我們將其中變動範圍最大者來放大來觀察，如圖 3.5 所示，得到的 y 座標範圍約在 2 個像素 (pixels) 大小。

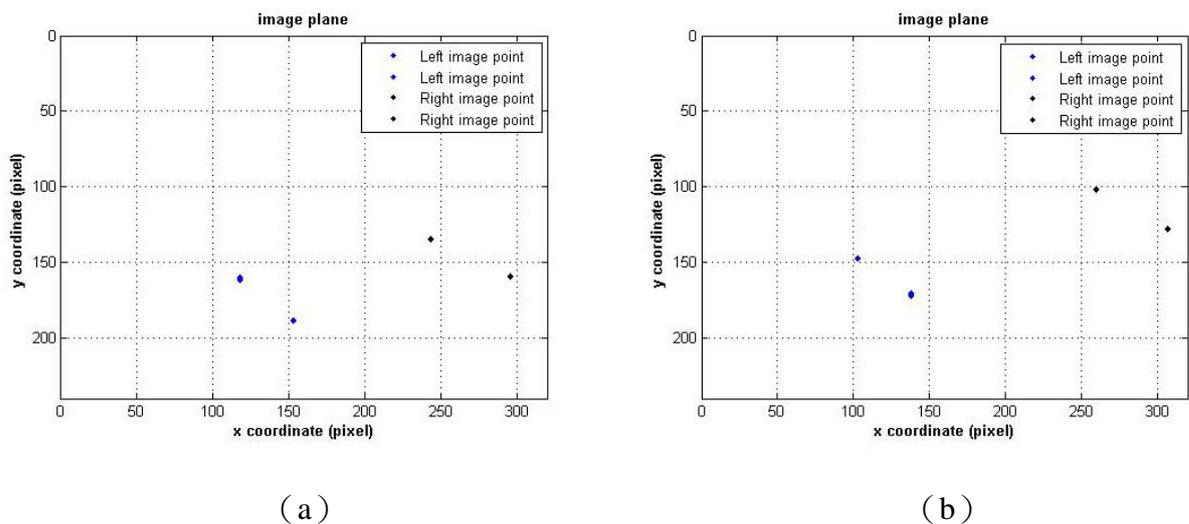


圖 3.4 固定 x 座標，看 y 座標的範圍，(a) 為實驗一，(b) 為實驗二。

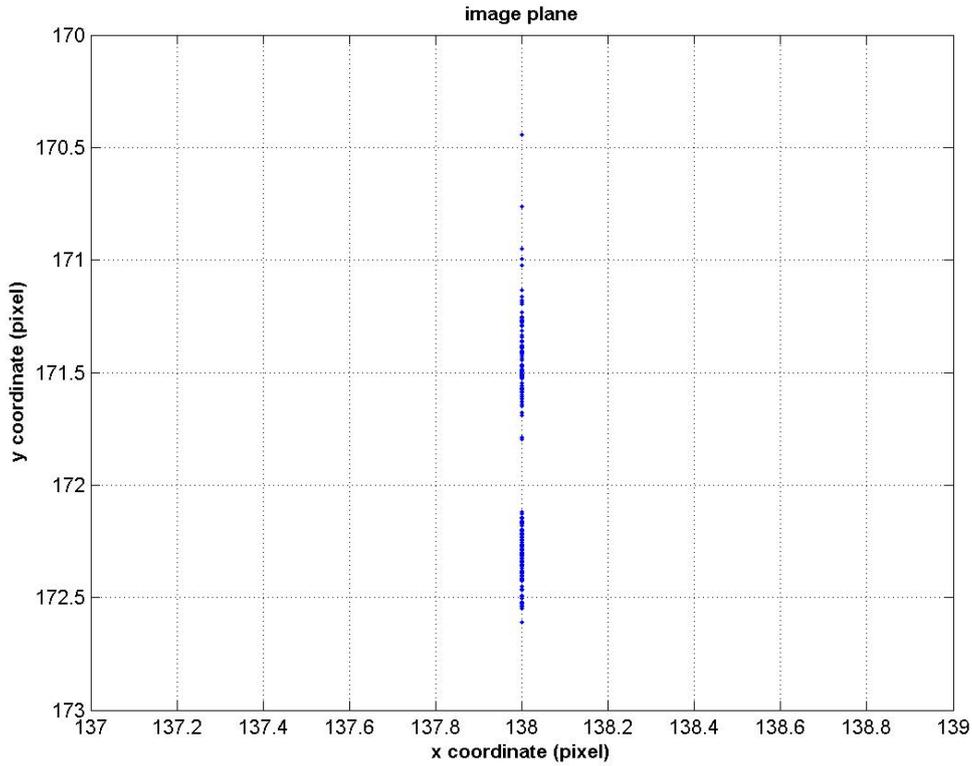


圖 3.5 將圖 3.4 (b) 中的其中一群資料放大。

我們再從時間與 y 座標變化的角度來觀察，圖 3.6 為圖 3.4 中， y 座標隨時間變化的表示圖。圖中我們可以發現， y 座標是跳動不定的，而且其跳動是沒有週期性的，是不規律的跳動。而原因主要是來自於光源不穩定，以及取樣裝置會有雜訊產生的影響，所以在影像上雷射筆的像素點一直在改變，導致 PCA 算出來的結果一直改變。

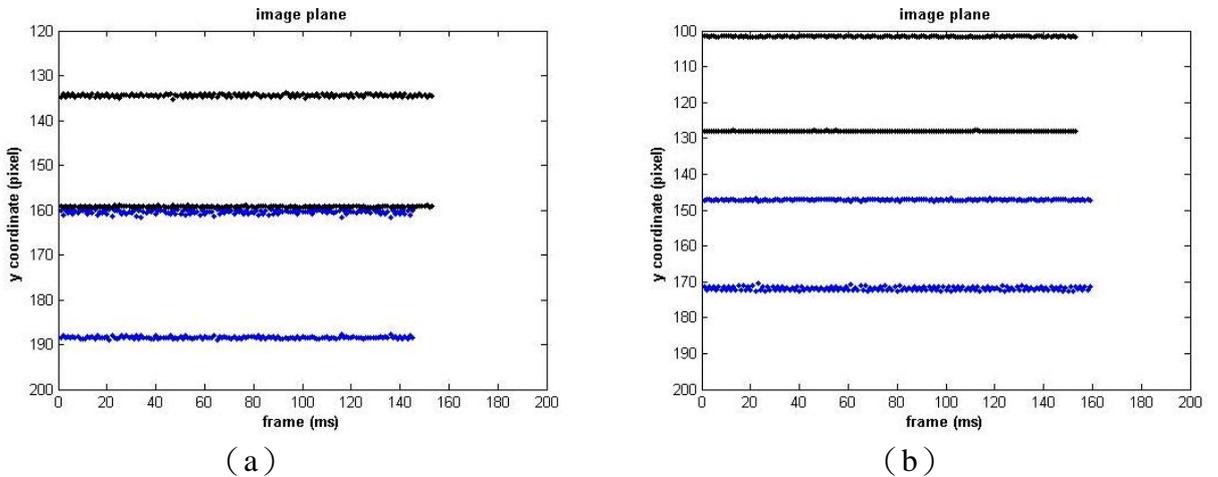


圖 3.6 x 座標為時間軸，看 y 座標的變化。(a) 為實驗一，(b) 為實驗二。

3.4 經由 Homography 計算造成的誤差

在整個系統當中，有兩個部份是利用平面投影轉換，分別為從影像平面轉換至參考平面，和從指向平面（即布幕）轉換至螢幕，而底下將針對這兩次的轉換所造成的誤差分別作探討。

如圖 3.7 所示，左右影像分別表示兩攝影機的視角，因為兩攝影機的視角無法同時照到相同的參考平面，所以左右兩攝影機的參考平面是分開的。在圖上分別以紅色點圈起來的四個標記點，即為我們用來作平面投影轉換的四個特徵點。此四個特徵點我們在影像上是以手動方式去點選，而在地面上的座標我們則是以量測的方式取得（可參考 3.1 節說明），參考平面的八個特徵點在世界座標系中的座標位置示意圖如圖 3.8 所示，而 C_1 與 C_2 則分別代表左右兩相機中心的位置。

以此方式，左右相機所得到的 3×3 平面投影矩陣 H_L 、 H_R ，分別為：

$$H_L = \begin{bmatrix} -0.00231184 & 626042929 & -0.00137555 & 370383911 & -0.340319115902899 \\ -0.00176139008075235 & -0.00516421 & 929625191 & 0.94029008 & 261871 \\ 6.8813341479990 \times 10^{-7} & -1.07825415 & 75954 \times 10^{-5} & -0.000671808121441 & \end{bmatrix}$$
$$H_R = \begin{bmatrix} -0.00178461532197717 & 0.00479595448795168 & 0.105894274445466 \\ -0.00186676553362327 & -0.00022812316168039 & 0.994362050347551 \\ 1.53311725714096 \times 10^{-7} & 7.3934444947710 \times 10^{-6} & 0.000890240732828 \end{bmatrix}$$

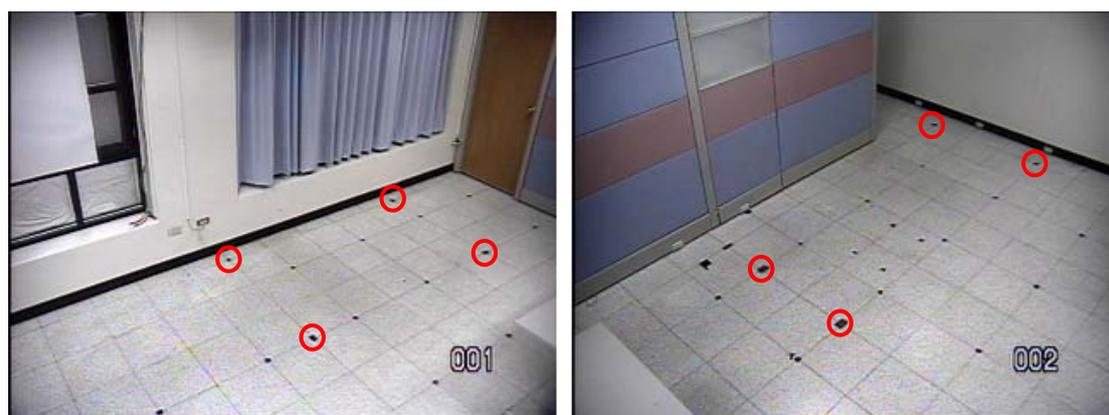


圖 3.7 左右兩攝影機的視角影像，紅色圈起來的部份為特徵點。

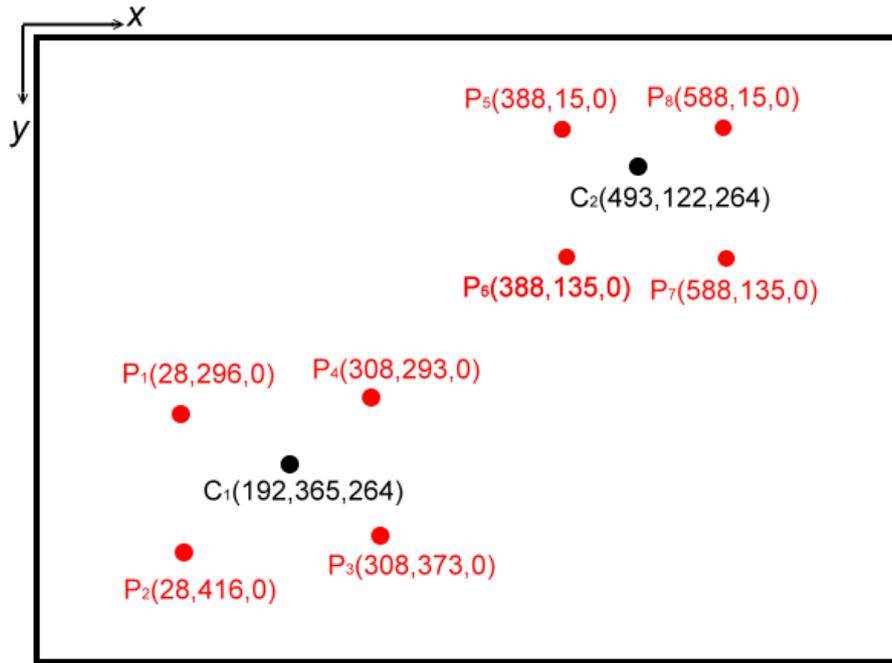


圖 3.8 參考平面的八個特徵點座標位置示意圖。 $P_1 \sim P_8$ 為八個特徵點， C_1 與 C_2 則分別代表左右兩相機中心的位置。

由影像平面與參考平面的座標關係，可以知道，從影像上的座標，投影至參考平面，因為是由小範圍轉換至大範圍，所以如果有誤差，則誤差會有被放大的效果。同樣的，當我們從指向平面要轉換至螢幕的座標時，也是利用平面投影轉換。我們量測圖 3.9 中四個紅色標記點的座標位置，分別為(115,243)、(295,243)、(295,108) 與 (115,108)，然後求出此四點與螢幕角落四點的轉換關係。由於我們將投影機所投射出來的螢幕平面四個角點分別設定為 (0,0)、(640,0)、(640,480) 以及 (480,0)，所以我們將圖 3.9 中的 $P_1 \sim P_4$ ，分別與此四角點作對應，所得到的 3×3 平面投影矩陣為：

$$H = \begin{bmatrix} 0.00374911967435655 & 0 & -0.427399642876532 \\ 0 & -0.00373572996122158 & 0.90404665061629 \\ 0 & 0 & 0.00108958790535711 \end{bmatrix},$$

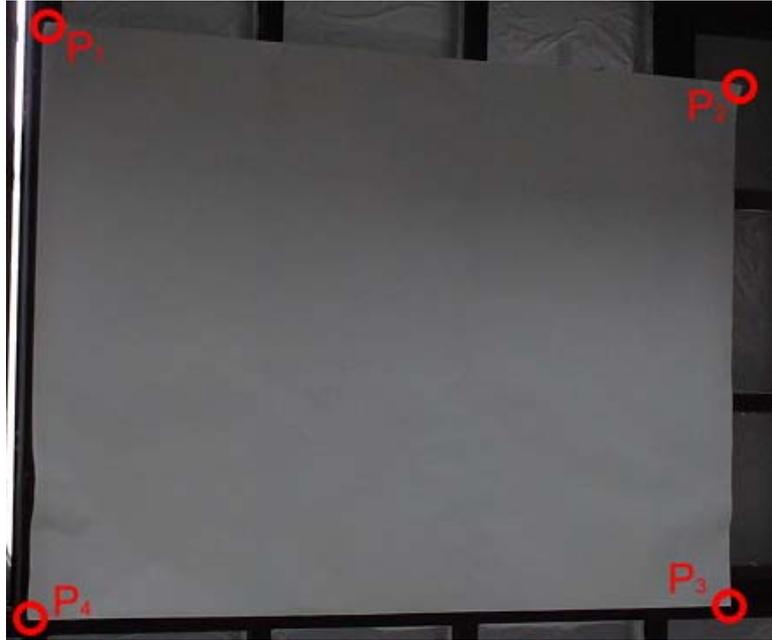


圖 3.9 指向平面（布幕）的座標示意圖。

轉換示意圖則如圖 3.10 所示，由圖上我們可以發現，長與寬分別從 180 轉成 640，135 轉成 480，皆分別放大了 32/9 倍。而歸究其原因，主要是因為單位的不同，在世界座標系當中，我們是以公分為單位，而在螢幕上則是以像素為單位，而由小至大的轉換，只要有誤差，就會有被放大的效果。而經由平面投影轉換所造成的誤差對系統所造成的影響，在下一章會有更深入的探討。

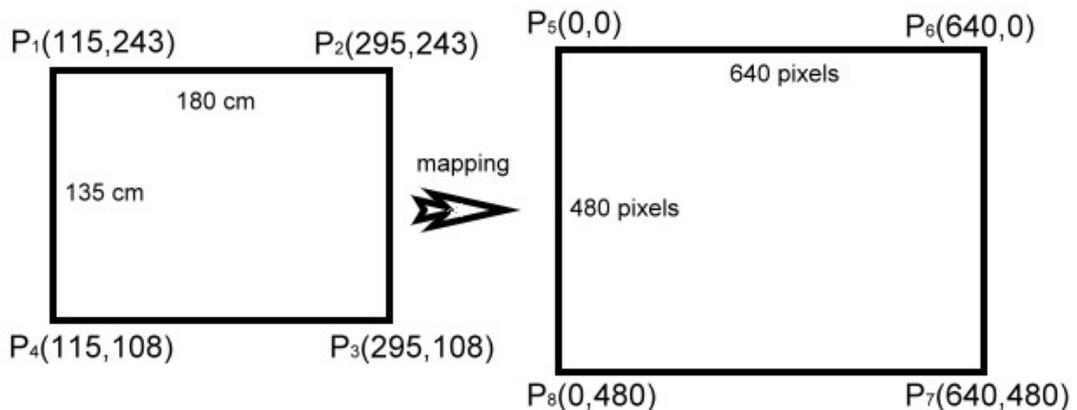


圖 3.10 指向平面與螢幕間的轉換示意圖。

3.5 總結

在本章中，我們說明了本系統所有可能的誤差來源。硬體設備所造成的誤差是我們無法避免的。而在座標量測上，我們盡可能的避免人為上的疏忽，盡量使得量測準確。而光源所產生的雜訊，以及 Homography 轉換所產生的計算誤差，將在最後實驗章節時說明我們如何改善，降低誤差對我們系統的影響。



第四章 影像誤差與重建誤差之關係

從第二章的內容可以知道（參考 2.2.3 節），當我們求得指向物在影像上兩端點的二維座標後，便利用投影幾何轉換（homography）的方法將二維座標轉換至三維空間座標上。在此，我們再一次的簡略說明流程，以便於本章說明。

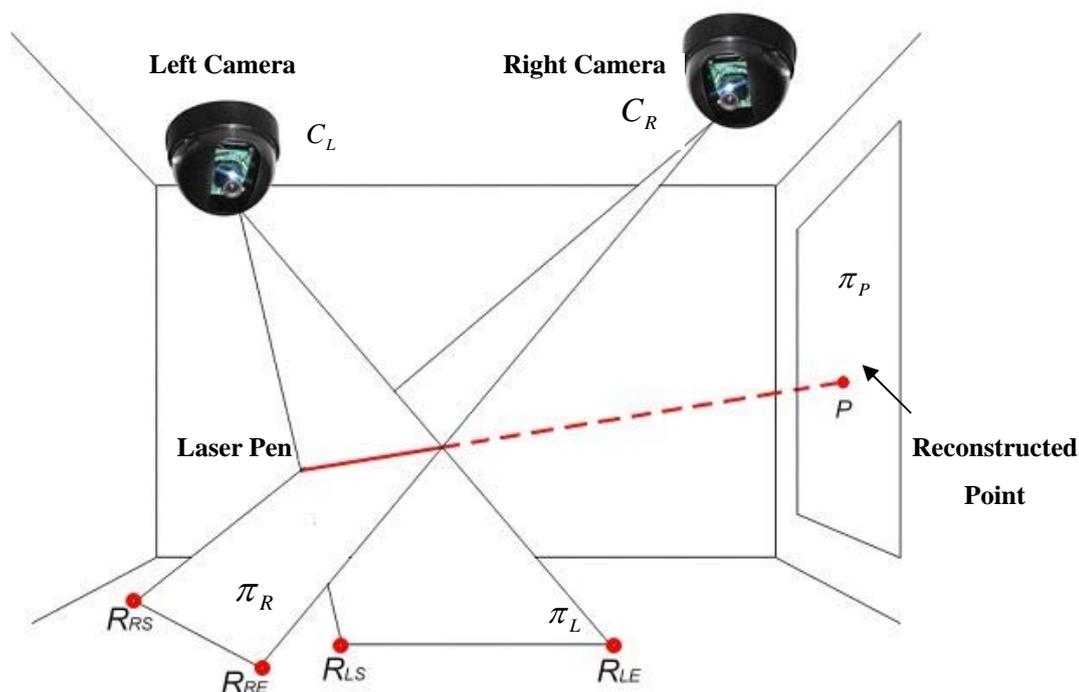


圖 4.1 由指向物至指向點之三維幾何示意圖。

圖 4.1 是重建指向物與其指向點的示意圖。當我們得到指向物在影像上兩端點的座標後，經由平面投影轉換，可以得到圖 4.1 上的 $\{R_{LS}, R_{LE}\}$ 、 $\{R_{RS}, R_{RE}\}$ 此四個座標點， $\{R_{LS}, R_{LE}\}$ 、 C_L 交出平面 π_L ， $\{R_{RS}, R_{RE}\}$ 、 C_R 交出平面 π_R ， π_L 、 π_R 與指向平面 π_P 三平面交於一點 P ，此點 P 即是我們的重建點。由於我們將此點設定給滑鼠游標，故必須再作一次平面投影轉換，將 P 轉換至螢幕座標上。

由第三章的探討可以知道，在影像端的種種誤差，會導致重建點的不正確。所以在本章中，我們要探討在不考慮誤差來源時，影像上的誤差與重建點（滑鼠座標）誤差之間的關係。我們會先對此流程中所使用的數學觀念作介紹，然後由

不同的角度出發來作影像誤差與重建的探討，再由已知影像上的最大誤差求重建誤差的上限，最後將探討在此種誤差分析之應用。

4.1 相關數學理論介紹

我們在不同平面間的轉換，是使用平面投影轉換的觀念，而在重建的部分，則是利用三維幾何的觀念。所以在此節，我們則先對相關的數學理論，作簡單的介紹，而詳細的數學理論與證明，可參考[11][12]。

4.1.1 平面投影轉換

平面投影轉換的簡單定義為，有兩平面 π_A 、 π_B ，要將一群位在 π_A 平面上的點集合 x_i ，透過透視投影對應到另一平面 π_B 上的點集合 x_i' 。其間存在一個 3×3 的投影轉換矩陣 H ，可將 x_i 中的每個點轉換至 x_i' 中對應的點。如果以齊次座標表示法表示，而 x_i 與 x_i' 分別表示為 $x_i = (u_i, v_i, w_i)^T$ 、 $x_i' = (u_i', v_i', w_i')^T$ ，則 x_i 、 x_i' 與 H 間的關係為 4.1 式：

$$Hx_i = \lambda x_i' \quad (4.1)$$

其中 λ 是不為 0 的比例常數， H 則是可逆 (non-singular) 矩陣。

4.1.1.1 齊次線性解

我們以外積的形式表示 4.1 式，則(4.1)可以寫成：

$$x_i' \times Hx_i = 0, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (4.2)$$

x_i' 以齊次座標表示法來表示，且將 H 的第 j 列以 h^{jT} 表示，再將 4.2 式外積展開，則可以推導出 4.3 式：

$$x_i' \times H x_i = \begin{bmatrix} v_i' h^{3T} x_i - w_i' h^{2T} x_i \\ w_i' h^{1T} x_i - u_i' h^{3T} x_i \\ u_i' h^{2T} x_i - v_i' h^{1T} x_i \end{bmatrix} \quad (4.3)$$

因為 $h^{jT} x_i = x_i^T h^j$, $\forall j \in N$, $1 \leq j \leq 3$, 藉由此關係式化簡後, 4.3 式可以寫成以 H 列向量為變數的式子, 如 4.4 式:

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \\ -v_i' x_i^T & u_i' x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (4.4)$$

4.4 式雖然是由三個方程式構成, 但只有前兩個方程式是線性獨立的 (linear independent), 所以只需要利用 4.4 式中, 前兩個方程式即可, 故(4.4)成為:

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (4.5)$$

至此已經推導出以 h 為變數的方程式, 接下來則要把 h_i , $1 \leq i \leq 9$ 解出, 即可解出 H 。

4.1.1.2 求解平面投影轉換矩陣

投影轉換矩陣 H 有九個未知數, 但因為其中一個為縮放係數 (up to scale), 故其自由度為 8, 也就是至少需要八條方程式才能解出 H 。而因為二維平面有 x 和 y 座標, 所以給予一組對應點, 就可以得到兩條方程式, 故需要四組對應點, 即能求出 H , 會完全符合這四組點的對應關係。而求解 H 的方法不只一種, 我們只介紹在本系統中的作法:

- (1) 每組 x_i 與 x_i' 中的對應點, 可從 4.5 式中得到一個 2×9 的矩陣 A_i 。
- (2) 因為有 4 組對應點, 則可得到 4 個 2×9 的矩陣 A_i , 將這些 A_i 結合成一個 8×9 的矩陣 A 。
- (3) 將 A 做 SVD 分解, 得到 $A = UDV^T$, 其中 D 為 A 的 singular value 所構成的對

角矩陣(diagonal matrix)、 V 為 singular vector 所構成的正交矩陣(orthogonal matrix)。我們所要求的 H 是最小的 singular value 所對應的 singular vector，一般而言，SVD 分解後的 D 會將 singular value 由大到小排列，意即 V 中第 9 行的九個元素： $[h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9]^T$ ，為構成 H 的元素。

(4) $h_i, 1 \leq i \leq 9$ 已計算出，則根據 4.2 式 H 的定義，即能求出 H 。

4.1.2 相關三維幾何運算介紹

在此節中，我們將針對系統中所使用的相關三維幾何運算作介紹，而我們會用到的符號表示先整理如下：

I_{LS} ：由左相機所得的指向線起始點，在影像平面上的座標點。

I_{LE} ：由左相機所得的指向線終點，在影像平面上的座標點。

I_{RS} ：由右相機所得的指向線起始點，在影像平面上的座標點。

I_{RE} ：由右相機所得的指向線終點，在影像平面上的座標點。

R_{LS} ： I_{LS} ，在三維參考平面上的投影點。

R_{LE} ： I_{LE} ，在三維參考平面上的投影點。

R_{RS} ： I_{RS} ，在三維參考平面上的投影點。

R_{RE} ： I_{RE} ，在三維參考平面上的投影點。

C_L ：左相機中心。

C_R ：右相機中心。

π_L ：由 R_{LS} 、 R_{LE} 與 C_L 三點所交出的平面。

π_R ：由 R_{RS} 、 R_{RE} 與 C_R 三點所交出的平面。

π_p ：指向平面。

P ：由 π_L 、 π_R 與 π_p 三平面所交出的點。

4.1.2.1 三維空間中平面方程式表示法

平面方程式皆可以一般式 $ax+by+cz=d$ 來表示，其法向量為 x 、 y 、 z 項的係數所組成，記成 $\vec{N}=(a,b,c)$ 。因為 \vec{N} 與平面上的任意向量皆垂直，故可以用平面上的兩向量的外積求得 \vec{N} 。我們以左相機為例，先由 R_{LS} 與 R_{LE} 求出 $\overrightarrow{R_{LS}R_{LE}}$ ， R_{LS} 與 C_L 求出 $\overrightarrow{R_{LS}C_L}$ ，經由此兩向量外積可求得法向量 \vec{N} ，算式為：

$$\vec{N}=(a,b,c)=\alpha(\overrightarrow{R_{LS}R_{LE}}\times\overrightarrow{R_{LS}C_L}) \quad (4.6)$$

α 為比例常數，一般我們皆當作 1 來計算。接下來將 R_{LS} 或 R_{LE} 帶入 $ax+by+cz=d$ 即可求出 d 。因為參考平面為 $z=0$ ，若 $R_{LS}=(x_{LS},y_{LS},0)$ ，則：

$$d=ax_{LS}+by_{LS}+c\cdot 0 \quad (4.7)$$

d 求出後則平面 π_L 即求出， π_R 亦由相同的方法求出。

4.1.2.2 三維空間中三平面求交點

在三維空間中，給予任意三個平面，其三平面間的相互關係有以下幾種：

- (1) 三平面皆平行。
- (2) 兩平面平行，第三平面與此兩平面各交於一條線。
- (3) 三平面兩兩交於一直線，而這三直線不相交。
- (4) 三平面交於一直線。
- (5) 三平面交於一點。

而因為兩平面 π_L 、 π_R 和指向平面 π_p 在系統中的相對位置固定，必定不會平行，故我們可以確定在系統中只會發生上述(5)的情況。接下來我們只需要再用 Cramer's rule 即可解出交點。

我們是利用三平面求交點，所以只要固定其中兩平面，就會交出一條線，而重建點勢必會位在這條線上。由於 π_p 是固定的，所以重建點的位置會由 π_L 與 π_R 決定。又在本系統中 C_L 與 C_R 固定，所以， R_{LS} 、 R_{LE} 、 R_{RS} 與 R_{RE} 此四點的位置，

會決定重建點的位置。而 R_{LS} 、 R_{LE} 、 R_{RS} 與 R_{RE} 則分別是由 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 作一次投影轉換所得，故接下來，我們要探討 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 變化時，對重建點造成的影響。

4.2 由不同誤差半徑與不同角度做探討

以下我們所作一個實驗，是先將左右兩影像拍攝下來，而計算出 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 的座標位置約在 (188,158)、(247,189)、(159,142) 與 (226,155)，並利用上述的方法求得重建座標為 (497.235,363.03)。上述的這五個座標點，我們則當作我們在 4.2 與 4.3 節中，探討的基準點。

在此節，我們要探討，當我們固定基準點，再改變 I_{LS} 、 I_{LE} 、 I_{RS} 或 I_{RE} 其中一點之位置，讓此點座落於與基準點 ± 1 個像素點的位置範圍內，分別探討其對重建點位置的影響。



4.2.1 一維雜訊，垂直與水平方向之探討

首先，我們欲觀察雜訊分佈在垂直方向時，對重建出來的滑鼠座標位置有何影響。我們固定 I_{LE} 、 I_{RS} 與 I_{RE} 在基準點上，只改變 I_{LS} 的座標，讓雜訊垂直分佈在以基準點為中心， ± 1 個像素點的範圍內，間隔為 0.2 個像素點上，如圖 4.2(a) 所示，重建出來的滑鼠座標位置則為圖 4.2(b)。圖中所標示的黑色「x」為以基準點作重建得到的重建點。在圖 4.2(a) 中，我們讓藍色點與紅色點成對稱分佈，可以發現，重建出來的滑鼠座標位置亦呈現對稱分佈。且當 I_{LS} 距離基準點越遠，則得到的重建點離基準的滑鼠座標點也越遠， x 座標的最大差距約為 12 個像素點， y 座標的最大差距則約為 9 個像素點。這些重建點皆座落在一直線上，此直線是由 π_R 與 π_P 的交線所決定。

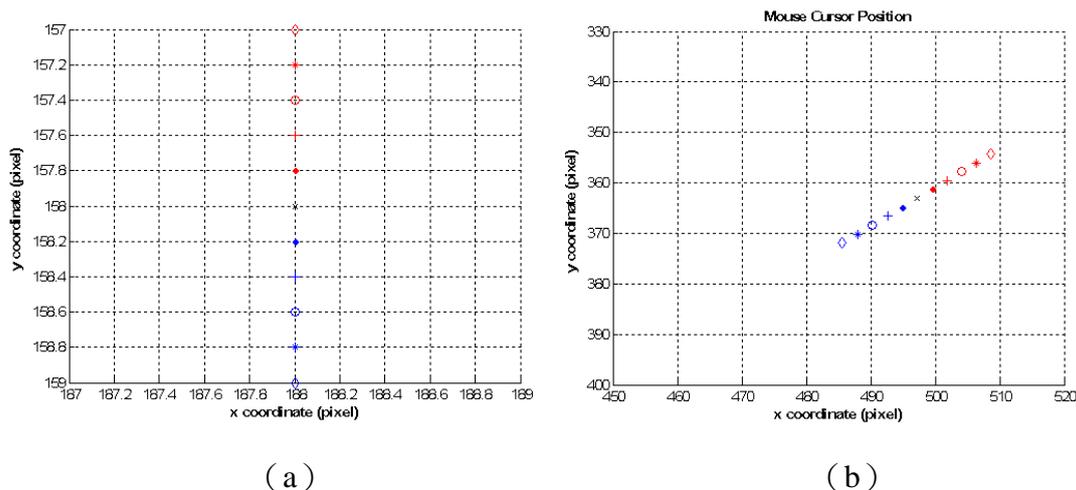


圖 4.2 改變 I_{LS} 的座標，讓雜訊分佈在垂直方向。(a) 為 I_{LS} 的座標，(b) 為重建的滑鼠座標。

接著，我們將雜訊分佈在水平方向，觀察水平方向的雜訊對重建出來的滑鼠座標位置有何影響。我們一樣固定 I_{LE} 、 I_{RS} 與 I_{RE} 在基準點上，只改變 I_{LS} 的座標，讓雜訊水平分佈在以基準點為中心， ± 1 個像素點的範圍內，間隔為 0.2 個像素點上，如圖 4.3 (a) 所示，重建出來的滑鼠座標位置則為圖 4.3 (b)。相同的，在圖 4.3 (a) 中，我們讓藍色點與紅色點成對稱分佈，可以發現，重建出來的滑鼠座標位置亦呈現對稱分佈。且在影像上的指向點距離基準點越遠，則得到的重建點離基準的滑鼠座標點越遠，不過其誤差卻比在垂直方向的小， x 與 y 座標皆在 10 個像素點內， y 座標甚至在 5 個像素點左右。

我們改變 I_{LS} 、 I_{LE} 、 I_{RS} 或 I_{RE} 其中一點，將雜訊分佈在垂直與水平方向上，由多次的實驗發現，距離基準點越遠，則所得到重建點離基準的滑鼠座標點越遠，但其誤差範圍卻不盡相同。從垂直與水平方向的實驗中，我們發現，在垂直方向的雜訊對系統重建點的影響較大，也因此引發了我們探討，雜訊分佈在其他的方向對系統重建點的分布與誤差有什麼影響。

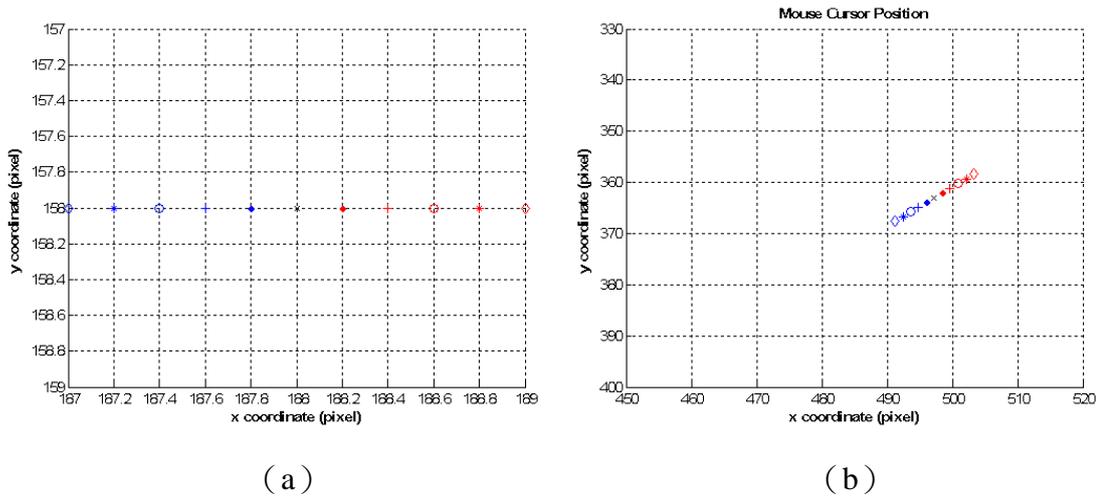
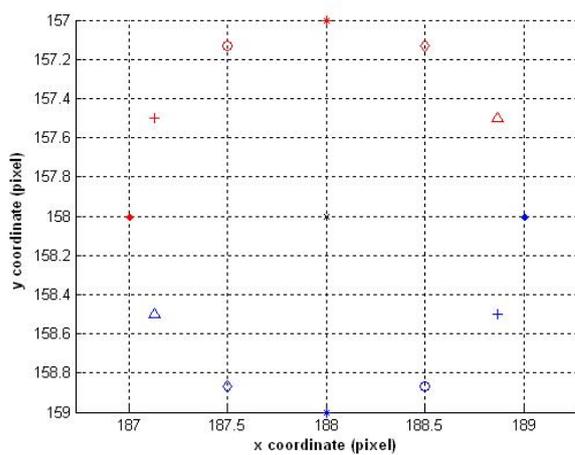


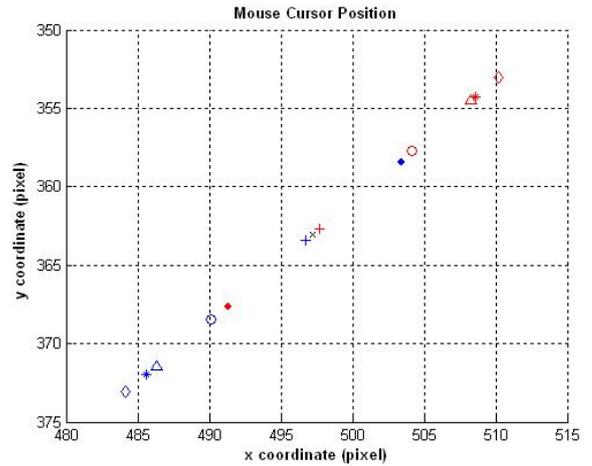
圖 4.3 改變 I_{LS} 的座標，讓雜訊分佈在垂直方向。(a) 為 I_{LS} 的座標，(b) 為重建的滑鼠座標。

4.2.2 固定距離，由不同方向作探討

探討完在垂直及水平方向的影像點，對重建點的影響後，我們固定距離，來探討相同距離但不同角度的影像點對重建點的影響。我們依舊只改變 I_{LS} ，以利我們作比較。我們在以基準點為圓心，半徑為 1 的圓上，每 30° 角取樣一次，圖 4.4 (a) 為 I_{LS} 的分佈圖，而重建點則為圖 4.4 (b) 所示。經觀察發現，當 I_{LS} 的點分佈在對稱於基準點的位置上時，其重建點也會對稱於基準點。而不同的角度，其距離基準點的誤差也不同。由圖 4.4 (b) 可以發現，當 I_{LS} 在 30° 角與 210° 角時，所產生的誤差最小， x 座標的誤差約為 0.5 個像素點， y 座標的誤差則約為 0.4 個像素點。但在 120° 角與 300° 角時，則會產生最大誤差， x 座標的誤差約為 13 個像素點， y 座標的誤差則約為 10 個像素點。



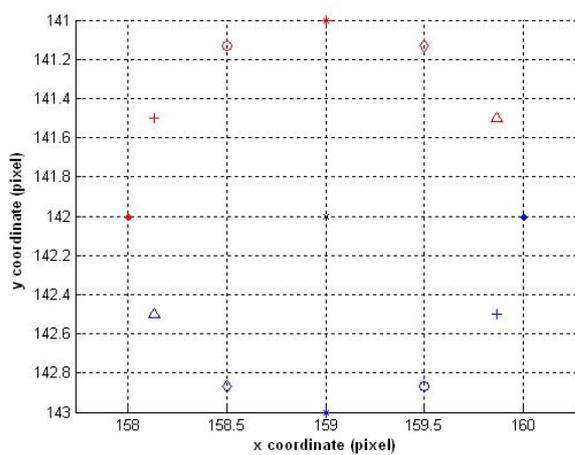
(a)



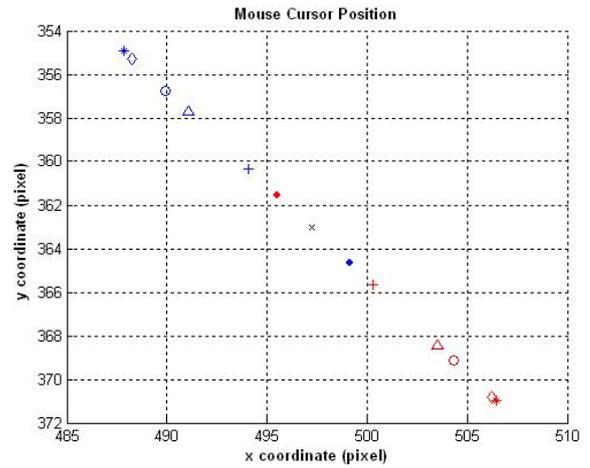
(b)

圖 4.4 改變 I_{LS} 的座標，觀察在不同角度上，對重建點的影響。(a) 為 I_{LS} 的座標，(b) 為重建的滑鼠座標。

接著我們固定 I_{LS} 、 I_{LE} 與 I_{RE} ，改變 I_{RS} ，一樣以基準點為圓心，在半徑為 1 的圓上，每 30° 角取樣一次。圖 4.5 (a) 為 I_{RS} 的分佈圖，而重建點則為圖 4.5 (b) 所示。由圖 4.5 (b) 可以發現，其重建點分佈也是呈現對稱，當 I_{RS} 在 0° 角與 180° 角時，所產生的誤差最小， x 座標約為 2 個像素點， y 座標則約為 1.5 個像素點。但在 90° 角與 270° 角時，則會產生最大誤差， x 座標的誤差約為 10 個像素點， y 座標的誤差則約為 8 個像素點。在圖 4.4 中，因為固定 I_{RS} 與 I_{RE} ，所以 π_R 會被固定，故重建點會分佈在 π_R 與 π_P 的交線上；同理，在圖 4.5 中，則是固定 I_{LS} 與 I_{LE} ，所以 π_L 會被固定，故重建點會分佈在 π_L 與 π_P 的交線上。



(a)



(b)

圖 4.5 改變 I_{RS} 的座標，觀察在不同角度上，對重建點的影響。(a) 為 I_{RS} 的座標，(b) 為重建的滑鼠座標。

經由不同角度的簡單模擬，我們發現在不同的角度上，所造成的誤差會不一樣。而經由上述簡單的模擬，引發了我們探討， I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 的座標如何分佈，會造成重建點的最大誤差。

4.3 在影像最大誤差的情況下求重建點的誤差上限

在 4.2 節的模擬皆是只改變 I_{LS} 、 I_{LE} 、 I_{RS} 或 I_{RE} 其中一點，不過因為無法預期在影像上的雜訊，會導致 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 怎麼變化，但經由圖 3.5 (參考 3.3 節) 可知，雜訊在影像上造成約為 2 個像素點的誤差，所以可以將雜訊分別分佈在以 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 為圓心，半徑為 1 的圓上作探討。由於我們並不知道 I_{LS} 、 I_{LE} 、 I_{RS} 與如何分佈，會造成重建點的最大誤差，所以我們一步步的作逼近，來找出重建點的誤差上限。

首先，如圖 4.6 所示，我們將 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} ，分別以基準點為圓心，半徑為 1，在 90° 倍數角點上取樣，然後各取 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 一點，作 4^4 次的組合，分別求出重建點，其重建點如圖 4.7 所示。我們再從這些重建點求出 x 座標和 y 座標的最大與最小值，這就構成重建點的誤差上限。

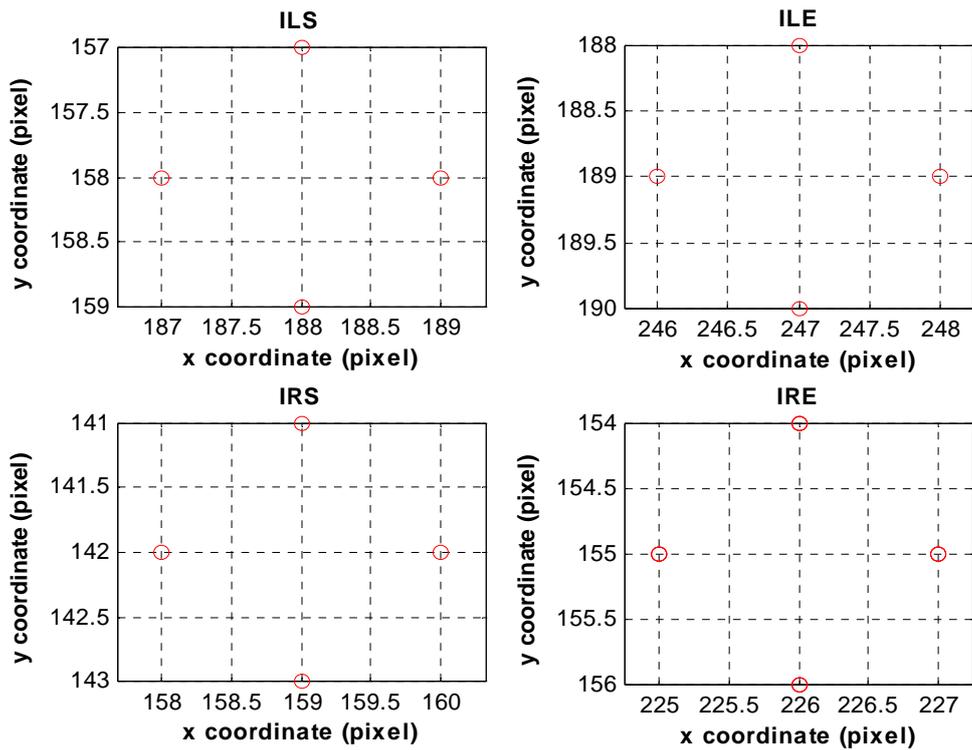


圖 4.6 將 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 90° 倍數角取樣的分佈圖。

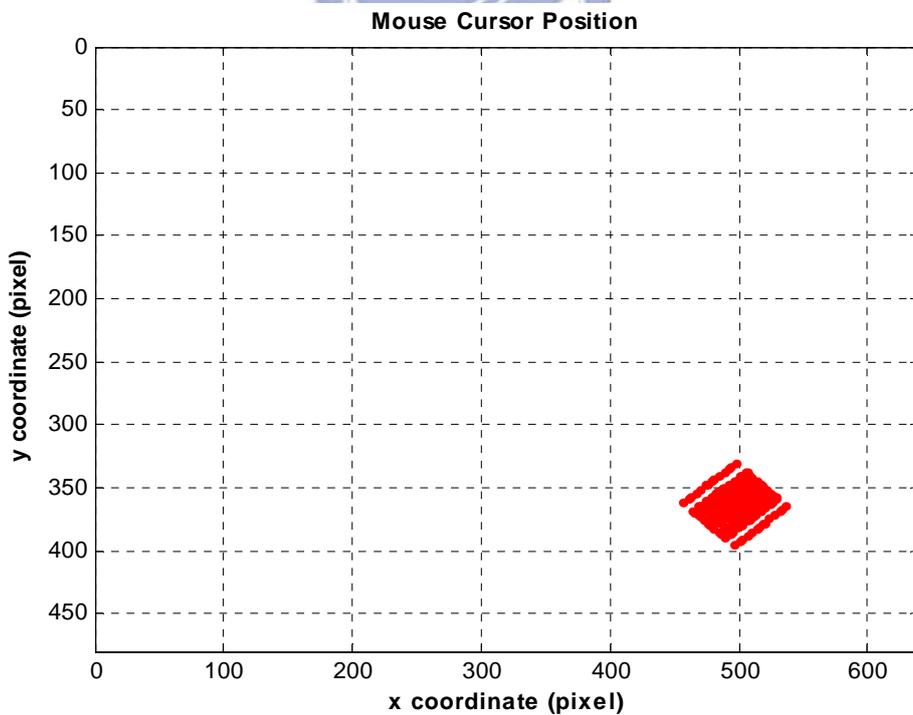


圖 4.7 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 90° 倍數角取樣的所有重建點分佈圖。

圖 4.8 則是求出的重建點誤差上限，(a) 是將 x 座標和 y 座標範圍分別設定在 $0 \sim 640$ 與 $0 \sim 480$ 之間，模擬在螢幕畫面滑鼠分佈的誤差範圍，(b) 則是將重建點誤差上限放大觀察。

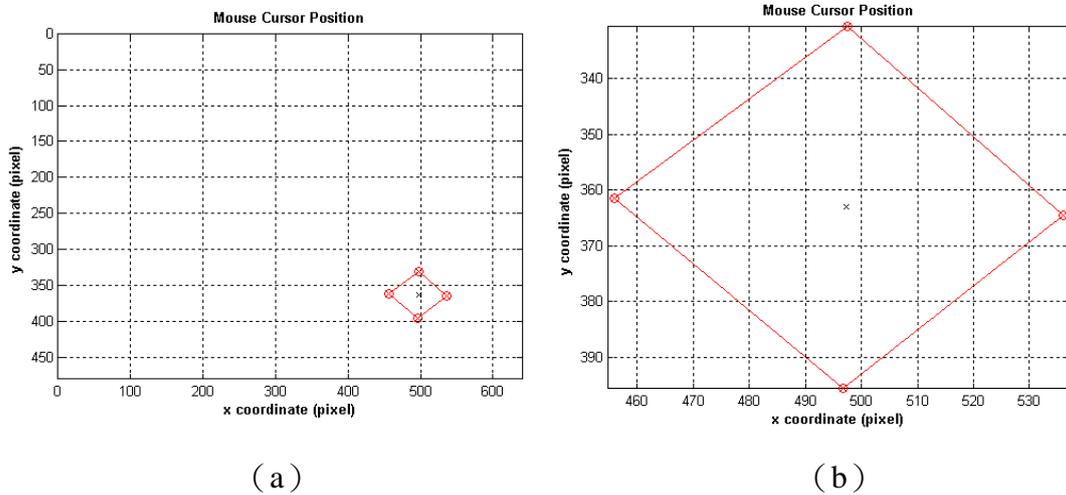


圖 4.8 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 90° 倍數角取樣，重建點誤差上限圖。(a) x 、 y 座標範圍分別設定在 $0 \sim 640$ 與 $0 \sim 480$ 之間，(b) 將重建點誤差上限放大觀察。

接著我們增加在圓上的取樣點，每 15° 角取樣一次， I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 取樣分佈圖如圖 4.9 所示。我們以相同的方法計算重建點，得到的誤差上限為圖 4.10。與圖 4.8 比較，圖 4.10 所得到的誤差上限比較準確。我們觀察到，當我們取樣的越密，則誤差上限越準確，而在此引起我們關注的是， I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別取樣在哪個點時，會構成圖 4.8 (b) 與圖 4.10 (b) 中的這些誤差上限點？而經由我們多次的模擬，發現當 $\overline{I_{LS}I_{LE}}$ 、 $\overline{I_{RS}I_{RE}}$ 的斜率為最大與最小時，所得到的重建點，會是構成誤差上限。

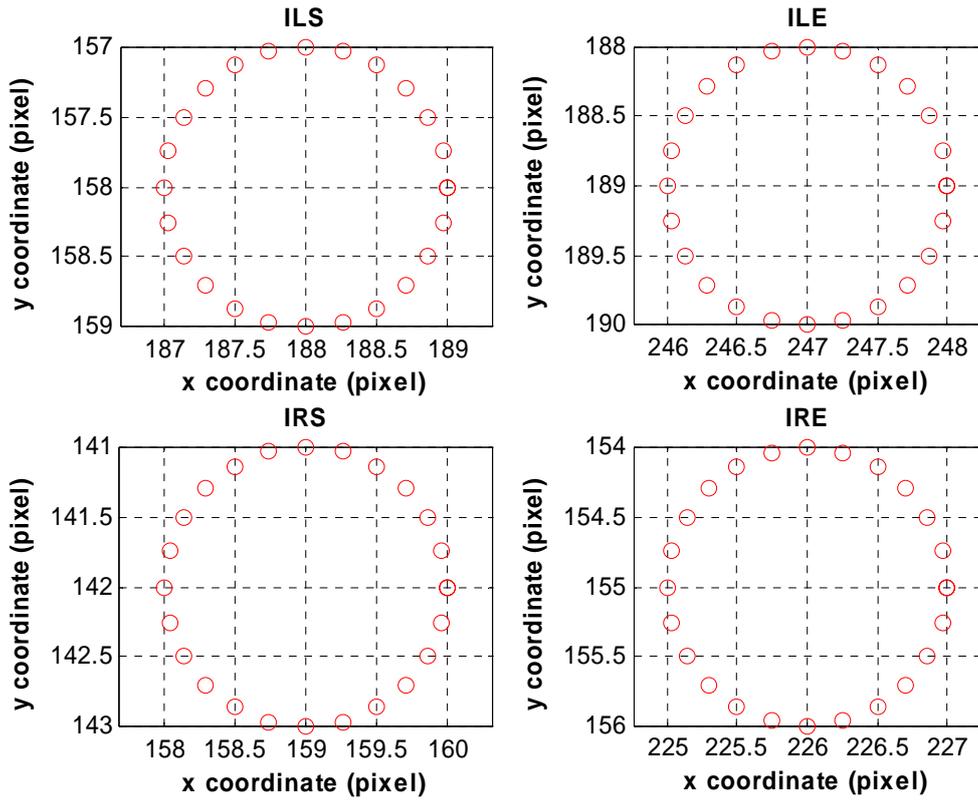


圖 4.9 將 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 15° 倍數角取樣的分佈圖。

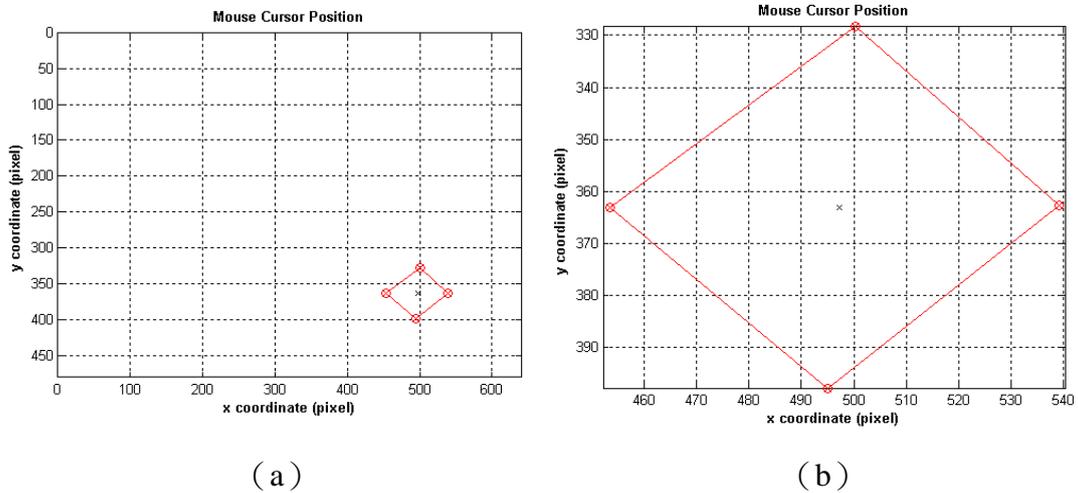


圖 4.10 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 分別在 15° 倍數角取樣，重建點誤差上限圖。(a) x 、 y 座標範圍分別設定在 $0 \sim 640$ 與 $0 \sim 480$ 之間，(b) 將重建點誤差上限放大觀察。

我們所模擬的四個圓，是以指向物在左右影像中所得到的端點為圓心，半徑為 1 所構成的。故從相對位置來看， I_{LS} 與 I_{LE} 所構成的兩圓的關係是外離、 I_{RS} 與 I_{RE} 所構成的兩圓的關係亦是外離。因為指向物夠長，兩端點在影像上的距離夠

遠，且兩圓的相對位置並非一上一下，故此兩圓的內公切線，分別是構成兩圓上點連線的最大與最小斜率。所以我們可以直接以數學理論求得內公切線與兩圓的切點，再由這些切點作重建，即可找出重建點的最大誤差。

我們延續上述的模擬，分別求出內公切線與切點，然後求得重建點的誤差上限。圖 4.11 為所得到的切線與切點，而圖 4.12 則為由這些切點所求得重建點的誤差上限。

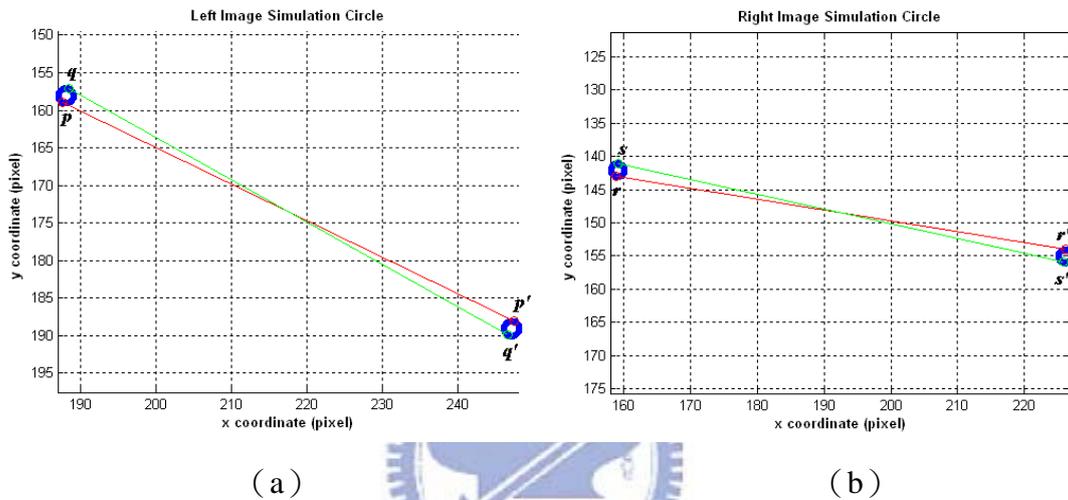


圖 4.11 I_{LS} 與 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求內公切線與切點，綠色與紅色點為切點，線則為兩內公切線。(a) 為 I_{LS} 與 I_{LE} 的模擬圖，(b) 為 I_{RS} 與 I_{RE} 的模擬圖。

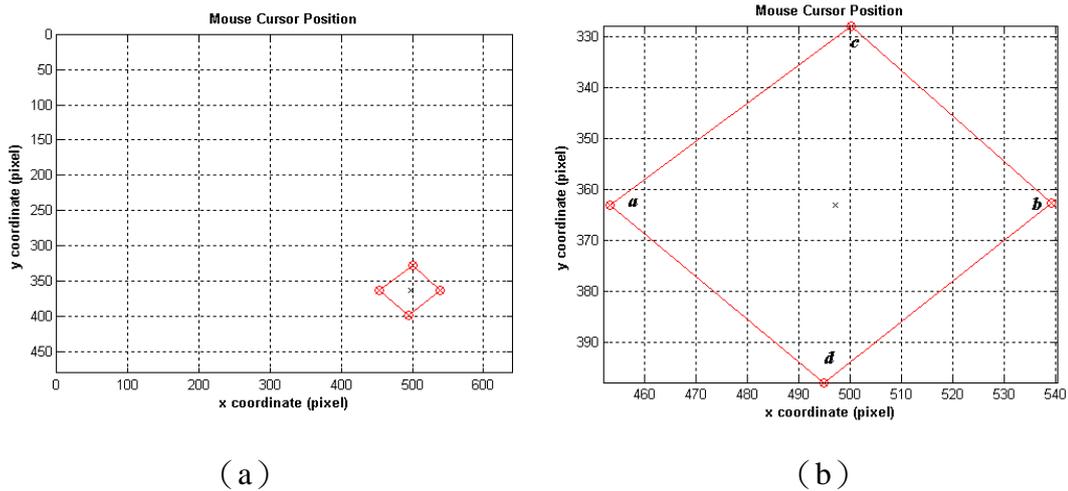


圖 4.12 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求內公切線與切點，重建點誤差上限圖。(a) 將重建點誤差上限放大觀察，(b) x 、 y 座標範圍分別設定在 0~640 與 0~480 之間。

經由觀察可發現圖 4.11 中的 $\{p, p', r, r'\}$ 四點構成圖 4.12 (b) 中的 a 點， $\{q, q', s, s'\}$ 、 $\{p, p', s, s'\}$ 與 $\{q, q', r, r'\}$ 則構成了 b 、 c 與 d 點。所以假若我們在左影像中 I_{LS} 與 I_{LE} 分別固定於 p 與 p' ，則所得到的重建點會落在圖 4.12 (b) 中 a 點與 c 點的連線上，同理若 I_{RS} 與 I_{RE} 分別固定於 s 與 s' ，則所得到的重建點會落在圖 4.12 (b) 中 b 點與 c 點的連線上，其餘同理可得。

我們將以 90° 倍數角、 15° 倍數角取樣，以及以切點來重建的重建點誤差上限畫在一起，如圖 4.13 所示。藍色線為 90° 倍數角所畫出的誤差上限，黑色線為 15° 倍數角所畫出的誤差上限，而紅色線則為以切點畫出的誤差上限。可以發現幾乎看不到黑色線，因為黑色線與紅色線幾乎重疊，也就表示以 15° 倍數角取樣所得到的誤差上限與以內公切線所得到的誤差上限差距很小。

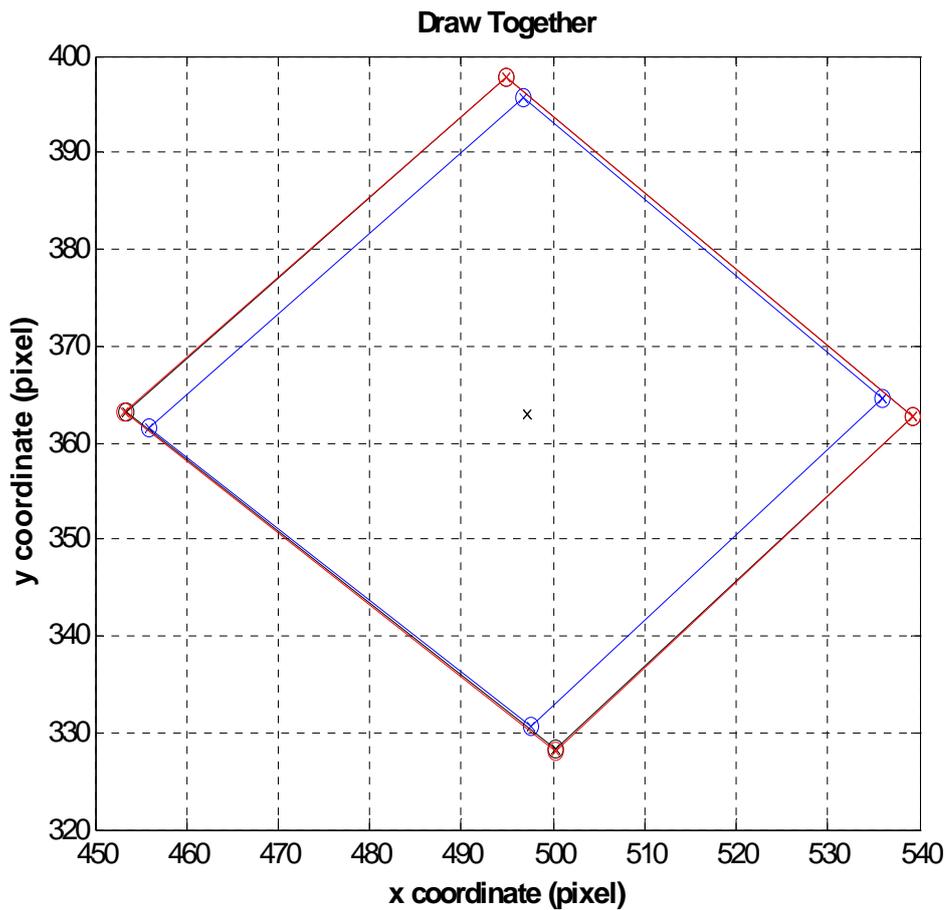


圖 4.13 比較重建點的誤差上限，將在 90 度倍數角上取樣、 15 度倍數角上取樣，及以切點來求重建點誤差上限畫在一起。

我們把以 90° 倍數角、 15° 倍數角取樣，以及以切點來重建的重建點誤差上限數據整理如表 4.1。可以發現以切點求得的誤差上限為最大，但 15° 倍數角幾乎與切點求得的誤差上限重合，誤差只在小數點下幾位。

表 4.1 以 90° 度倍數角、 15° 度倍數角與切點所得的重建點誤差上限數據表。（單位：像素）

	xMax	xMin	yMax	yMin
90 degree	535.9857	455.96	395.6483	330.5393
15 degree	539.2251	453.4022	397.8248	328.1907
內公切線	539.2422	453.2395	397.8823	328.1027

在我們的系統中，內公切線反應在指向物上，是使用者以指向物的一半長度為中心，將指向物作最大角度旋轉。而外公切線反應在指向物上則為最大的平移，故也因此引發我們探討，若 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 剛好座落在外公切線與圓的切點上，則重建點會有多大的誤差範圍。所以我們分別求出外公切線與圓的切點，然後求得重建點的誤差上限。圖 4.14 為所得到的切線與切點，而圖 4.15 則為由這些切點所求得重建點的誤差上限。

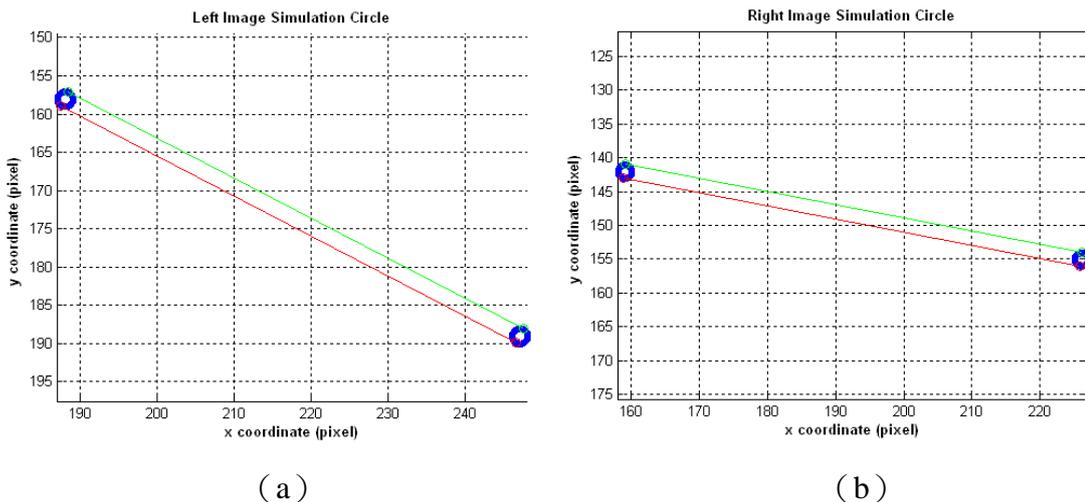
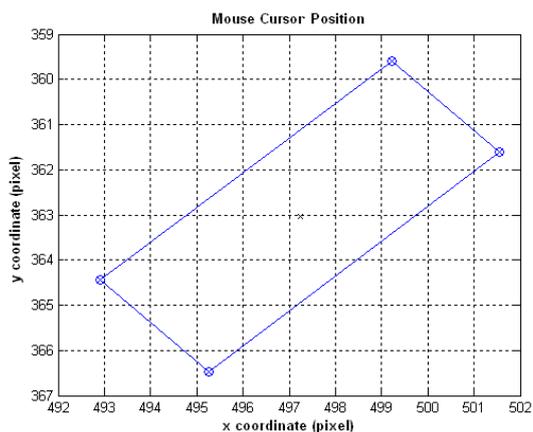
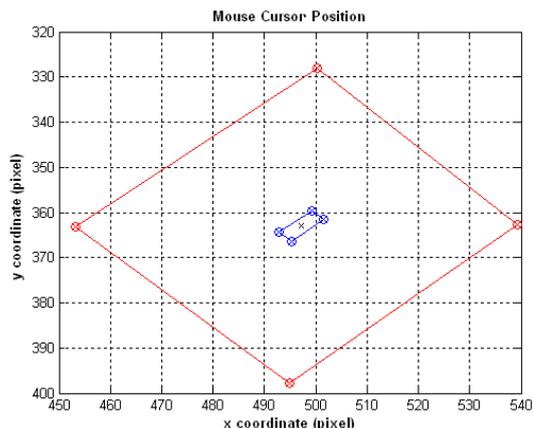


圖 4.14 I_{LS} 與 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求外公切線與切點，綠色與紅色點為切點，線則為兩外公切線。(a) 為 I_{LS} 與 I_{LE} 的模擬圖，(b) 為 I_{RS} 與 I_{RE} 的模擬圖。



(a)



(b)

圖 4.15 (a) I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 以數學理論求外公切線與切點，重建點誤差上限圖，(b) 將內公切線與外公切線造成的誤差畫在一起。

由圖 4.15 (a) 可以發現，重建點的 x 座標和 y 座標誤差範圍，皆在 10 個像素點下，遠遠小於內公切線所造成的重建誤差。因此我們得知，指向物旋轉所造成的誤差遠大於平移所造成的誤差。

經過多次的模擬後發現，直接利用內公切線與圓的切點來求重建點，即會產生最大誤差上限。所以接下來的探討，就直接以內公切線與圓的切點來求重建點的誤差上限。

4.4 應用

在 4.3 節的探討，我們的四個基準點是分別設定在 (188,158)、(247,189)、(159,142) 與 (226,155)。而當使用者的身高、手舉起的高度、以及站的位置不同，則指向物在左右影像上的座標亦不同。意即我們可以根據使用者的身高，或不同的站位，給予不同的基準點，來求取重建點的誤差上限，然後給予使用者回饋，告知使用者在固定的環境下，使用本系統時，該如何站位會使得所得到的誤差上限最小。

本系統只需要同時使用兩台攝影機，而假若在實驗場景中，有多台以上的攝影機時，我們可以根據不同的使用者站位，來搭配選取不同位置的攝影機，使得

誤差上限最小。但由於在我們的實驗場景中，只有兩台固定在天花板上的攝影機，故在此就不從不同的攝影機位置作探討。底下我們就只針對使用者在不同的站位下作探討。

4.4.1 由使用者不同站位來探討

我們底下作了三次的實驗，讓同一位使用者在兩攝影機的視角內，分別站在不同的位置。我們藉由拍攝左右兩影像，分別求出指向物在影像中的座標，以此座標當作圓心，半徑為 1，再求出內公切線與兩圓的切點來求重建點的誤差上限。

圖 4.16 為第一次實驗的站位，我們將這次的站位當作基準點。而圖 4.17 則為以切點計算出來的重建點誤差上限。



圖 4.16 第一次實驗的站位。(a) 左影像視角，(b) 右影像視角。

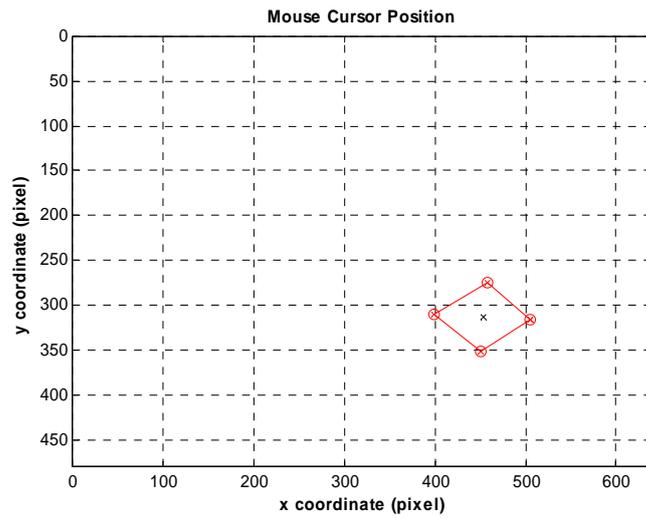


圖 4.17 第一次實驗所計算出來的誤差上限。

接著，我們往右移動約 80 公分的距離，重新再作一次實驗，相機視角圖為圖 4.18 所示。而圖 4.19 則為以切點計算出來的重建點誤差上限。



圖 4.18 相對於第一次實驗，往右移約 80 公分的站位。(a) 左影像視角，(b) 右影像視角。

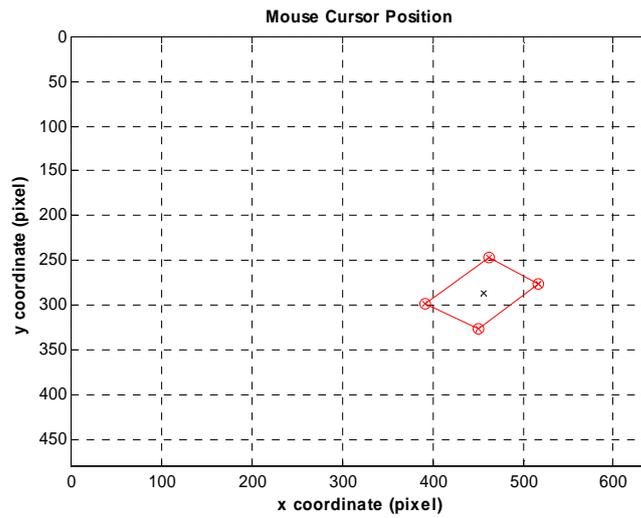


圖 4.19 第二次實驗所計算出來的誤差上限。

接著我們往基準點左方移動約 80 公分的距離，重新作一次實驗，相機視角圖為圖 4.20 所示。而圖 4.21 則為以切點計算出來的重建點誤差上限。



圖 4.20 相對於第一次實驗，往左移約 80 公分的站位。(a) 左影像視角，(b) 右影像視角。

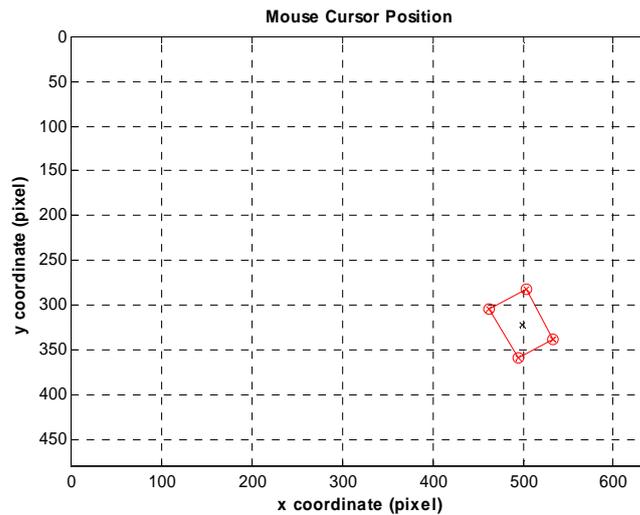


圖 4.21 第三次實驗所計算出來的誤差上限。

我們將這三次實驗的誤差上限數據表整理如表 4.2。可以發現，當使用者的站位不同，指向物在影像中的座標位置亦不同，所以所得到的誤差上限也會不同。若以這三次的實驗為例，我們會建議使用者站在第三次實驗的站位，即圖 4.20 的位置，則所得到的誤差會最小。

表 4.2 三次實驗所得的重建點誤差上限範圍數據表。（單位：像素）

	1 st exp	2 nd exp	3 rd exp
X Range	106.3633	125.1726	71.5868
Y Range	75.4534	79.5892	76.4874

由上述的實驗可以知道，指向物不同的影像座標，所得到的誤差上限會有所不同。在此引發我們想探討的是，指向物在影像上的座標如何分佈，以及什麼原因導致所得到的重建結果，會使得系統最不穩定。故我們將指向物放置在影像中較為邊緣的位置，以及較易受到雜訊影響的位置，做了幾次的實驗，來觀察重建出來的結果，在此我們只放其中三次的結果。

首先我們擺在如圖 4.22 的位置，在左影像呈現是比較偏左下角，而右影像則是偏右上角，重建出來的結果如圖 4.23 所示。外圍四點是誤差上限，而在外圍四點中間的紅色點則為實際上所得到的重建點。可以發現所得到的重建範圍不

大。而圖 4.24 是圖 4.22 得到的二值化影像，經由觀察，我們發現指向物在影像上的位置是比較不受雜訊所干擾的，所以每張畫格找到指向物的座標幾乎是一樣的，故其重建結果較好。



(a)

(b)

圖 4.22 第一次實驗指向物在影像上的位置。(a) 左影像視角，(b) 右影像視角。

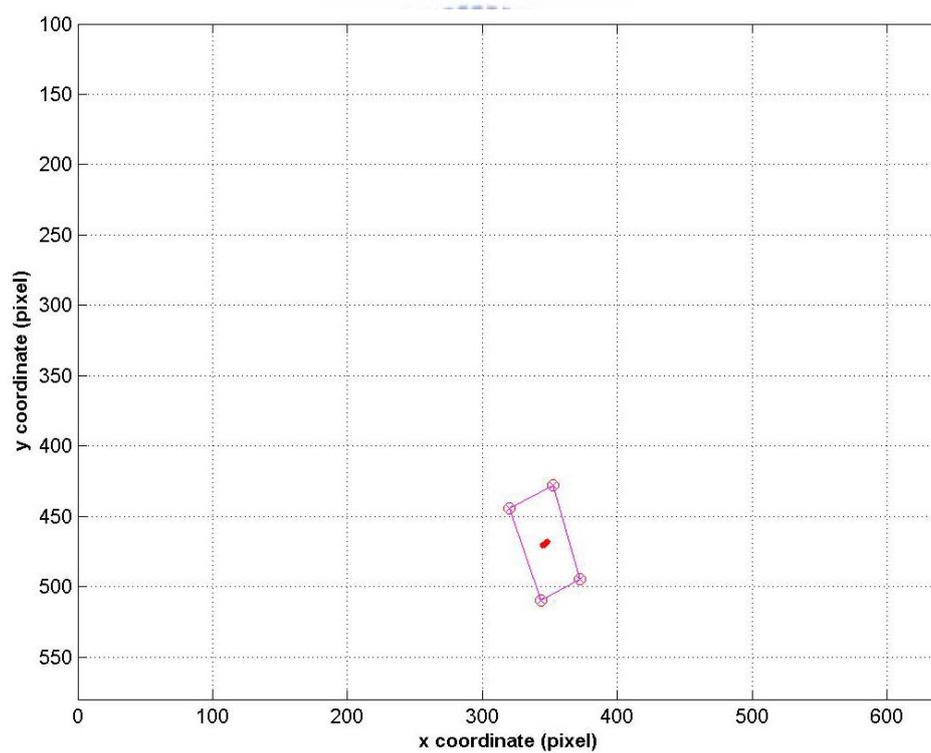


圖 4.23 第一次實驗的重建點分布情況與誤差上限。



圖 4.24 第一次實驗左右影像做二值化後的結果。

接著我們擺在如圖 4.25 所示的位置，在左圖中是偏右下角的位置，而重建出來的結果如圖 4.26 所示。其重建結果點範圍明顯有變大，誤差上限也較大，我們從二值化影像來觀察，如圖 4.27 所示，可以發現雖然左影像的指向物可以很完整的找出來，但右影像中的指向物，卻有很多應該是資料點但卻被判斷為非資料點，且這些點並非固定位置，所以在每張畫格利用 PCA 方法算出指向物端點時，就會一直改變，故重建的結果也會比較不穩定，導致重建結果的範圍會比較廣。

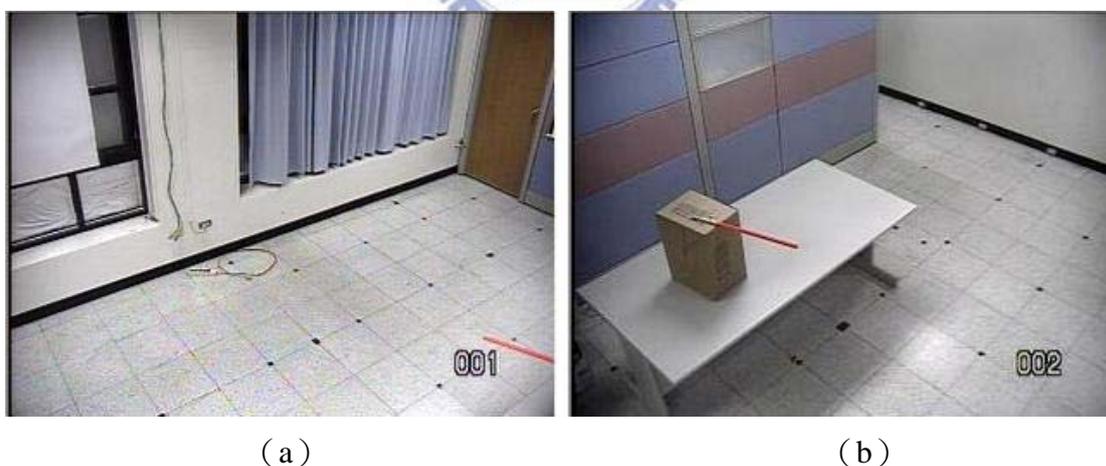


圖 4.25 第二次實驗指向物在影像上的位置。(a) 左影像視角，(b) 右影像視角。

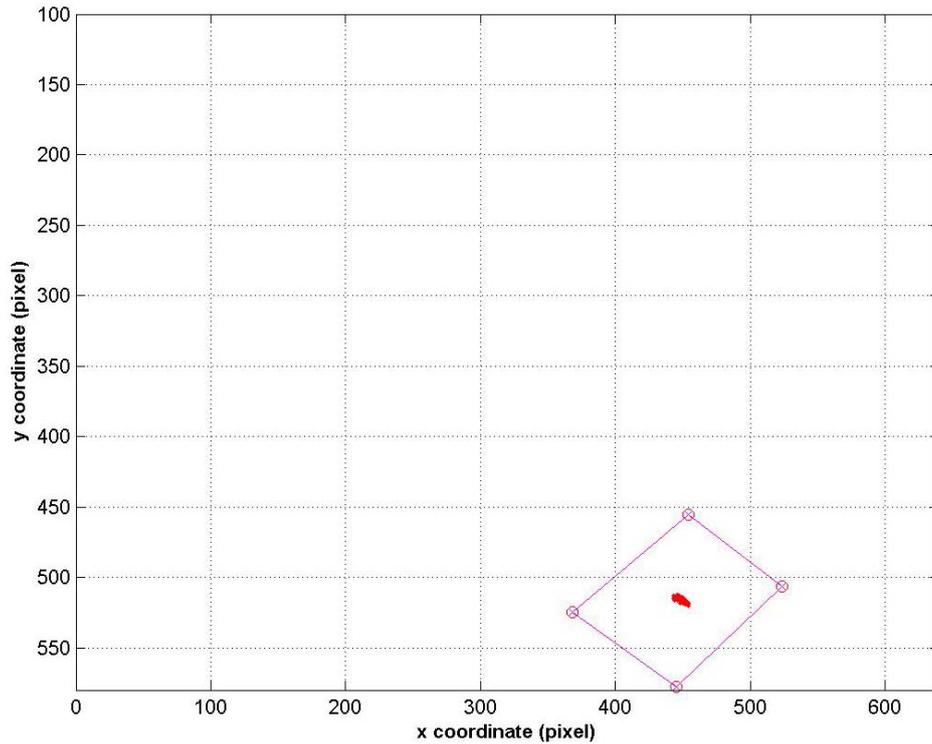


圖 4.26 第二次實驗的重建點分布情況與誤差上限。



圖 4.27 第二次實驗左右影像做二值化後的結果。

接著我們將指向物擺在如圖 4.28 所示的位置，其重建出來的結果如圖 4.29 所示。可以發現，重建結果誤差分佈比較廣，且誤差上限也比較大。從二值化影像中觀察，如圖 4.30 所示，可以發現指向物在左影像受雜訊嚴重干擾，且指向物在右影像中之資料點十分稀疏，所以所得到的重建誤差分佈會比較大，意即當指向物擺在圖 4.28 的位置，滑鼠游標會一直跳動。而因為左影像受雜訊嚴重干擾，使得在左影像中資料點經由 PCA 所得到的兩端點一直在變化，右影像資料點雖然稀疏，但因為較沒有受到雜訊的干擾，PCA 所得到的兩端點比較固定，

所以圖 4.29 重建點的分佈方向，才會是偏左上右下的線段，這也印證了在圖 4.11 與圖 4.12 間切點與重建點間的關係。



圖 4.28 第三次實驗指向物在影像上的位置。(a) 左影像視角，(b) 右影像視角。

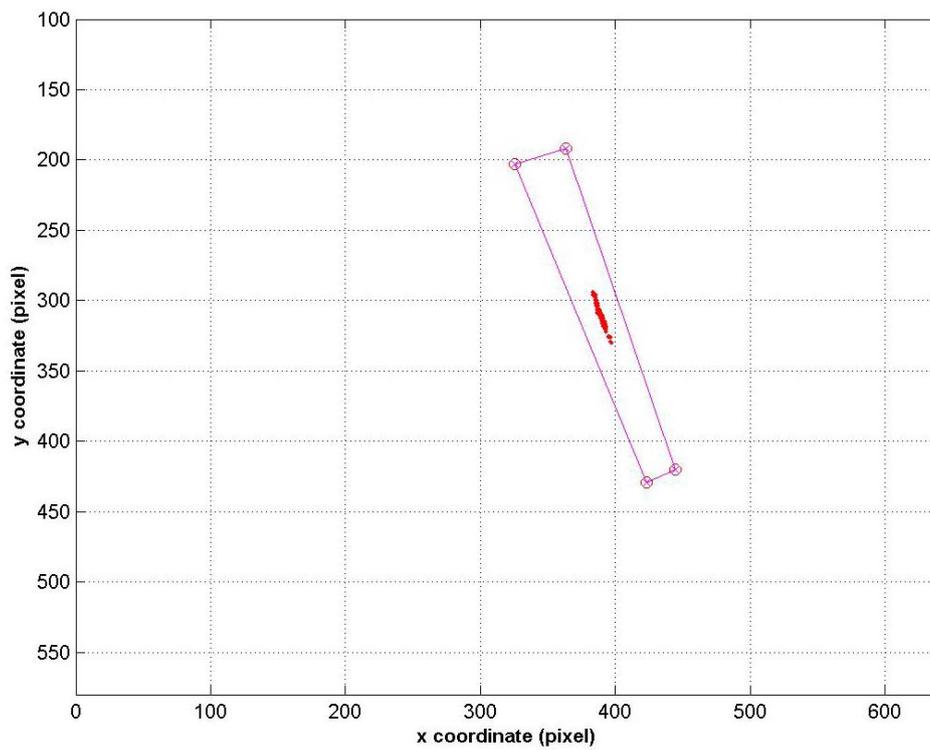


圖 4.29 第三次實驗的重建點分布情況與誤差上限。

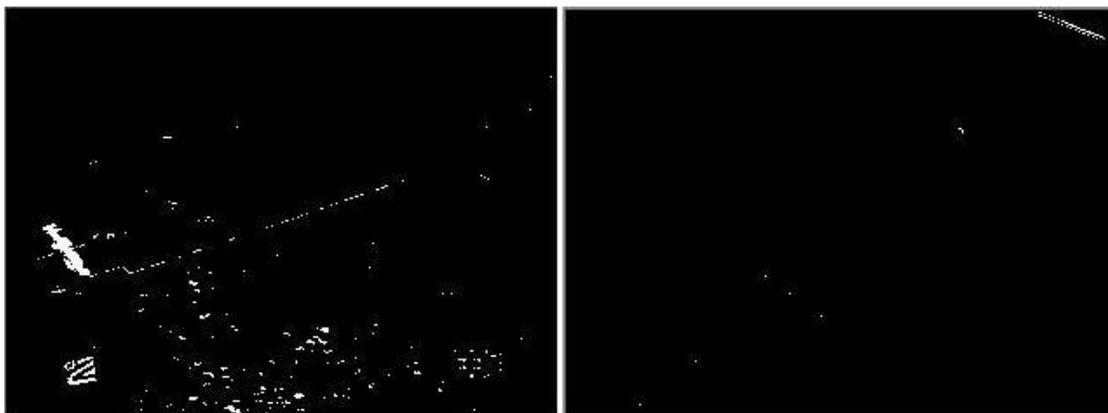


圖 4.30 第三次實驗左右影像做二值化後的結果。

由以上的實驗可以知道，在使用此系統時，我們可以先根據當時的光源，以及環境在光源下的雜訊分佈，然後再依照使用者的身高資訊，告知使用者站位，降低雜訊對系統的影響，並且提高使用者在使用此系統時的穩定度。

4.5 結論

在本章中，我們利用模擬的方式，來探討影像誤差與重建誤差之間的關係。從指向物在影像上的座標，至最後得到重建點並設定給滑鼠游標，整個流程我們以平面投影轉換和三維幾何運算的理論為背景來完成。從模擬的結果可以得知，平移指向物所產生的誤差小於旋轉指向物所產生的誤差。而在系統中，只要在影像端產生 ± 1 個像素點的誤差，且剛好座落於內公切線與圓的切點上，則重建點的誤差範圍有可能多達 100 多個像素點。所以如何在影像端減少誤差，使得每次指向物在影像上的座標變化小，進而降低重建點的誤差，則會在下章作探討。

第五章 靜態與動態指向誤差之探討

在第二章系統運作一節所介紹的內容，是一個初步的流程。由於我們是以立體視覺為基礎，所以我們必須同時有左右兩影像上指向物的座標才能作重建。但因為程式在一個時間點只能執行一個指令，而且我們是以 callback 的方式來取得影像，故我們無法在同一個時間點，從兩攝影機同時獲取左右兩影像。我們總共作了九次實驗，如圖 5.1 所示，藍色點與紅色點分別代表左右影像， x 座標代表的是擷取到影像的時間點， y 座標則為實驗的次數。由圖中可以得知，左右兩影像不可能完全在同一時間上取得，時間差大約為 $1/30$ 秒。所以，我們必須記錄取得影像的時間點，以便接下來系統利用。

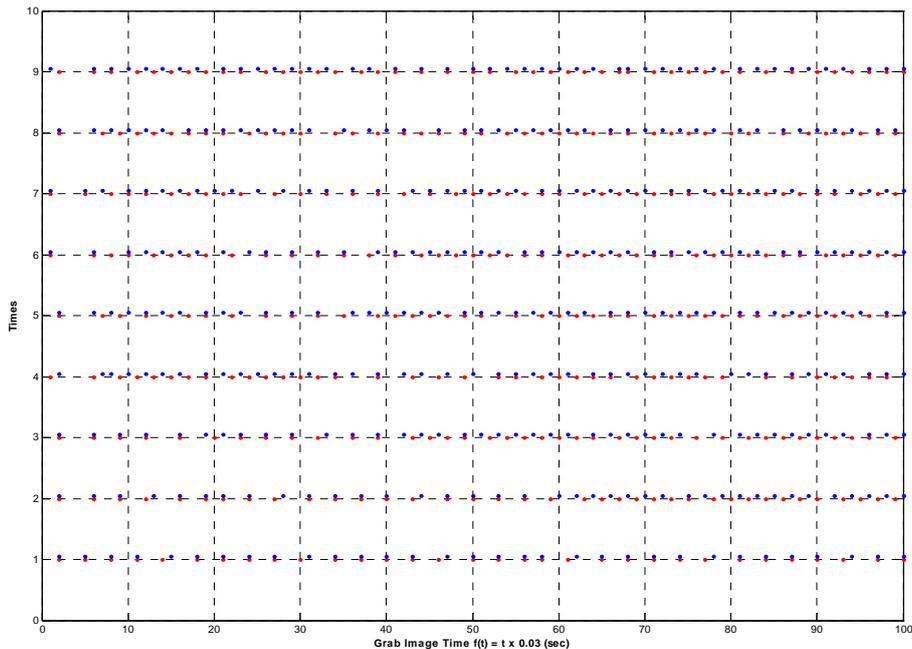


圖 5.1 獲取影像時間點的分佈圖。

而從第四章的結論，可以得知，假設在影像上有 1 個像素點的誤差，有可能會使重建時產生 100 個以上的像素點誤差，這樣的誤差會使得系統無法運作。因此，在本章中，我們會先介紹系統的一些實作設計，包括在系統中嘗試降低誤差的四種方法，以及這些方法在靜態指向與動態指向下所得到的結果。

5.1 部分系統實作設計介紹

在系統中，我們使用兩個計時器。其中一個計時器，用來觸發重建，我們設定為40毫秒。意即每間隔40毫秒，我們會挑選現有的指向物的座標作重建，來取得重建點的位置，並且設定給滑鼠游標。另一個計時器，則是用來記錄擷取到影像的時間點。我們設定為每1/30秒增加1，利用此機制模擬畫格速率，意即假若擷取到左右兩影像時的時間，在系統記錄上是相同的，則此兩影像之間取得的間隔差必定小於1/30秒。

另外，當我們得到指向物在影像上的兩端點座標時，即 $\{I_{LS}, I_{LE}\}$ 或 $\{I_{RS}, I_{RE}\}$ ，我們利用佇列（queue）此資料結構，將指向物的兩端點座標，以及擷取到影像時的時間記錄下來，我們使用兩個佇列 Q_L 、 Q_R ，分別記錄左右影像的資訊。會選擇用佇列是因為其先入先出（first in first out）的特性，剛好符合時間有先後次序的特性，所以較早擷取到的資訊，會記錄在佇列的前端，會較早處理。

利用計時器以及佇列的設計，我們就可以將 $\{I_{LS}, I_{LE}, t_{I_L}\}$ 與 $\{I_{RS}, I_{RE}, t_{I_R}\}$ 記錄下來， t_{I_L} 與 t_{I_R} 分別表示擷取到左右兩影像時的時間。待計時器觸發重建時，即可從佇列挑選，比較 t_{I_L} 與 t_{I_R} 的值，來判斷此兩張影像的擷取時間是否很相近。

5.2 減少影像與重建誤差的四種方法

從第四章我們知道，因為使用多次平面投影轉換的技巧，在整個系統中會使得重建誤差放大。因為攝影機的角度固定，所以從影像平面轉換至參考平面的投影轉換矩陣就會固定。又因為從指向平面轉換至螢幕座標的投影轉換矩陣也固定，所以我們也無法藉由改變此投影矩陣的數值來減少誤差。所以，我們只能從影像上或最後的重建座標著手，使得最後重建出來的座標點能夠盡量準確。在[8][9]中，當每次要作重建時，是直接找尋 Q_L 與 Q_R 中，第一個 $|t_{I_L} - t_{I_R}| \leq 3/30$ 秒的點來作重建，但重建的結果相當的不穩定。所以在此，我們嘗試了四種方法來

降低誤差，進而使重建的結果穩定，分別在下面四小節作介紹。

5.2.1 平均重建點以求減少誤差

當計時器觸發要重建時，我們從[8][9]之 Q_L 與 Q_R 中找出第一個 $t_{I_L} = t_{I_R}$ 的 $\{I_{LS}, I_{LE}, t_{I_L}\}$ 與 $\{I_{RS}, I_{RE}, t_{I_R}\}$ 作重建，重建後把重建點記錄下來，待重建了幾次後，將這些重建點作平均，再將平均值設定給滑鼠游標。

我們固定指向物的朝向，作幾次的實驗，讓其指向同一點，將重建點分別作二、三、四、五個點的平均。如圖 5.2 所示，雖然皆指向同一點，但可以發現重建出來的結果（ x 、 y 座標），不管以二、三、四、五個點作平均，皆無法呈現一直線。原因在於每次重建點的跳動並不是呈現週期性的分佈，仔細觀察發現重建出來的結果會呈現斷層的分佈，即某段時間皆重建在相同點上，這是因為當計時器觸發重建時， Q_L 與 Q_R 並還沒有新的資料點產生，故我們以前一次重建時的資料來取代，所以不管作幾點平均，皆是拿相同的數值作平均，故在重建端作平均對於減低誤差所造成的效果並不顯著。

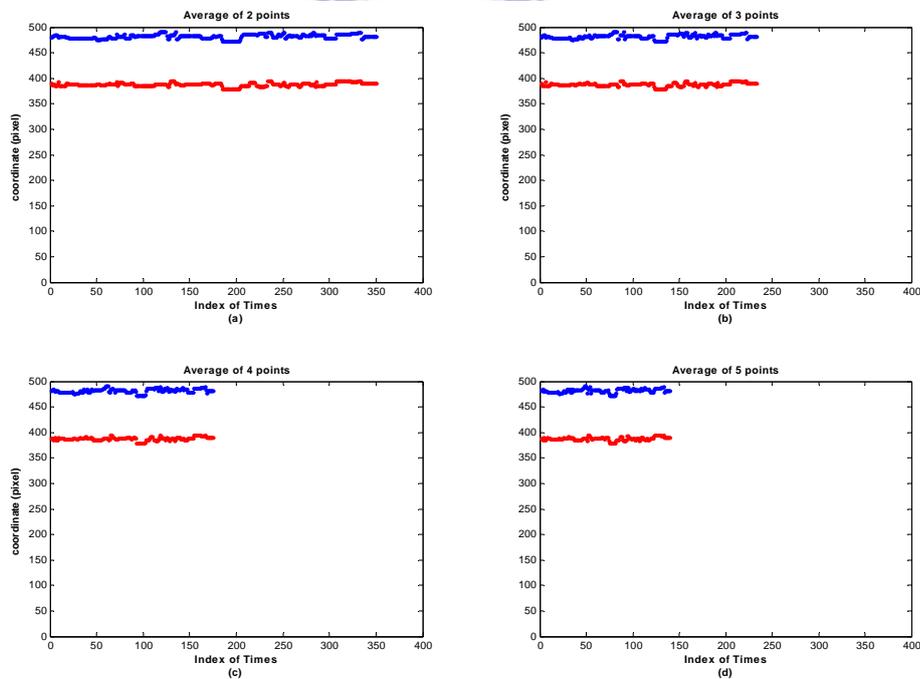


圖 5.2 將重建點以二、三、四、五點平均，再設定給滑鼠座標。(a) - (d) 分別為二、三、四、五點的平均。藍色與紅色分別表示重建點的 x 、 y 座標， x 軸表示第幾次的重建。

5.2.2 平均相同時間的影像點以求減少誤差

由於作完重建後再透過平均的方式減少誤差，所得到的效果不佳，故我們嘗試在尚未作平面投影轉換前，對 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 作平均。當計時器觸發要重建時，我們從 Q_L 與 Q_R 中找出前幾個 $t_{I_L} = t_{I_R}$ 的 $\{I_{LS}, I_{LE}, t_{I_L}\}$ 與 $\{I_{RS}, I_{RE}, t_{I_R}\}$ 資料，將 I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 各自分別作平均，再將平均後的結果作重建，所得到的重建點設定給滑鼠游標。

如同 5.2.1 節的實驗，我們固定指向物的朝向，作幾次的實驗，讓其指向同一點，我們也以二、三、四、五點來作平均。如圖 5.3 所示，在圖上可觀察出兩現象，首先，因為與 5.2.1 節一樣是取相同時間的資料點作平均，故其結果會呈現斷層的分佈。再者，雖然圖下方的綠色與黑色點幾乎呈現一直線，但放大觀察發現會有小數點下的差距，由第四章可知，雖然影像端的誤差很小，但重建端誤差會被放大，所以重建座標看起來會跳動。反應在滑鼠端，就會造成游標不穩定。

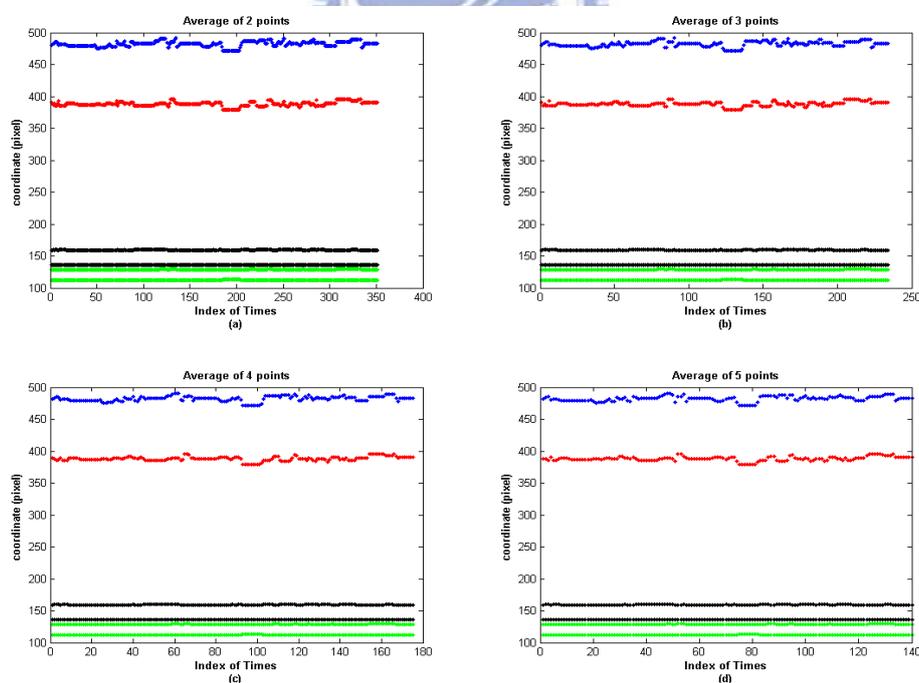


圖 5.3 將 I_{LS} 、 I_{LE} 、 I_{RS} 、 I_{RE} 以二、三、四、五點平均，拿平均值作重建，再設定給滑鼠座標。(a) - (d) 分別為二、三、四、五點的平均。藍色與紅色分別表示重建點的 x 、 y 座標，而綠色與黑色點則分別表示 I_{LS} 、 I_{LE} 、 I_{RS} 、 I_{RE} 的 y 座標值， x 軸表示第幾次的重建。

5.2.3 平均多點影像點以求減少誤差

在前兩方法中，我們都必須先挑選 $t_{I_L} = t_{I_R}$ 的資料點，只要 $t_{I_L} \neq t_{I_R}$ 的資料點，我們則丟棄不予使用。而在此小節的方法中，當計時器觸發要重建時，我們會直接取出佇列中的前幾個資料作平均，平均後的數值再作重建。意即我們不管 $t_{I_L} = t_{I_R}$ 或 $t_{I_L} \neq t_{I_R}$ ，直接從兩佇列分別取出相同個數的資料點作平均，然後將在左右佇列中的第一點丟棄。由圖 5.1 中可以發現，每一秒鐘的間隔，大約會有 9 點資料點，所以在實驗中，利用此方法時，我們是取佇列中的前 9 個資料作平均。

5.2.4 平均前一秒影像點以求減少誤差

在前一小節介紹的方法中，我們是在每次重建時，都會取左右兩佇列中，最前端的 9 點作平均。而在此小節的作法則是，當計時器觸發要重建時，會先取得當下的時間點 t ，然後從左右佇列中選取，在此時間點前一秒鐘內的所有資料點來求平均，即 $t-1 \leq t_{I_L} \leq t$ ， $t-1 \leq t_{I_R} \leq t$ ，平均後的數值再作重建。所以每次重建時，選取的個數點不一定會相同。

5.3 四種方法在靜態指向上的實驗

我們在實驗場景中，將指向物固定在某定點，如圖 5.4 所示。然後分別以 5.2 節所介紹的四種方法作多次的實驗。由於我們最後是將重建點設定給滑鼠游標，而滑鼠游標的座標值是以整數為單位，所以在我們求得重建點後，我們將數值從浮點數以四捨五入的方式轉為整數。圖 5.5 至圖 5.8 分別是以 5.2.1 節至 5.2.4 節介紹的四種方法，在圖 5.4 中的位置所重建出來的座標點。



圖 5.4 系統畫面截圖。分別為左右兩攝影機視角的畫面。而圖中我們以藍色線畫上的，是找到的指向物。

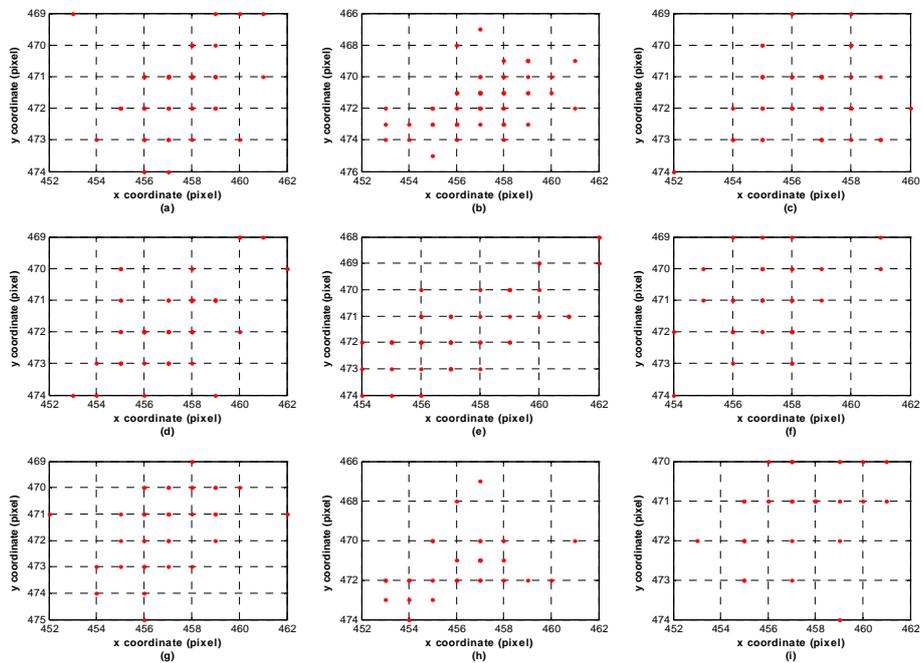


圖 5.5 利用 5.2.1 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

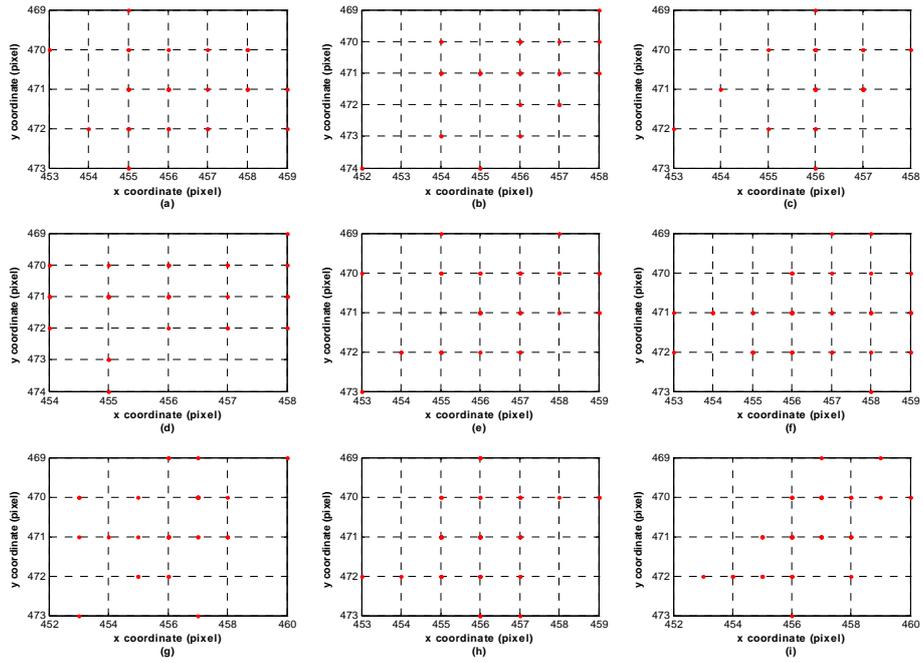


圖 5.6 利用 5.2.2 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

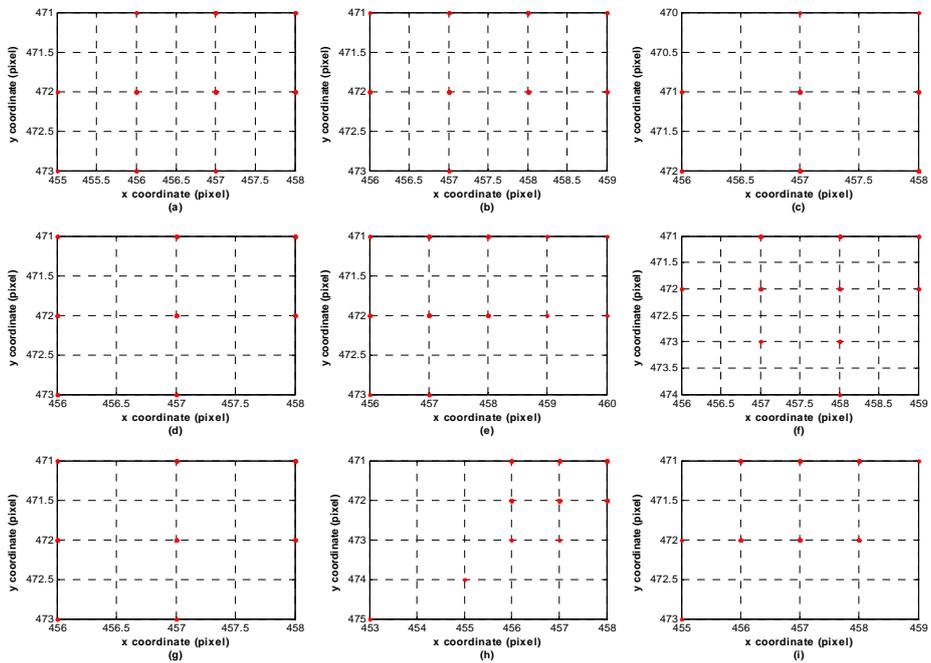


圖 5.7 利用 5.2.3 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

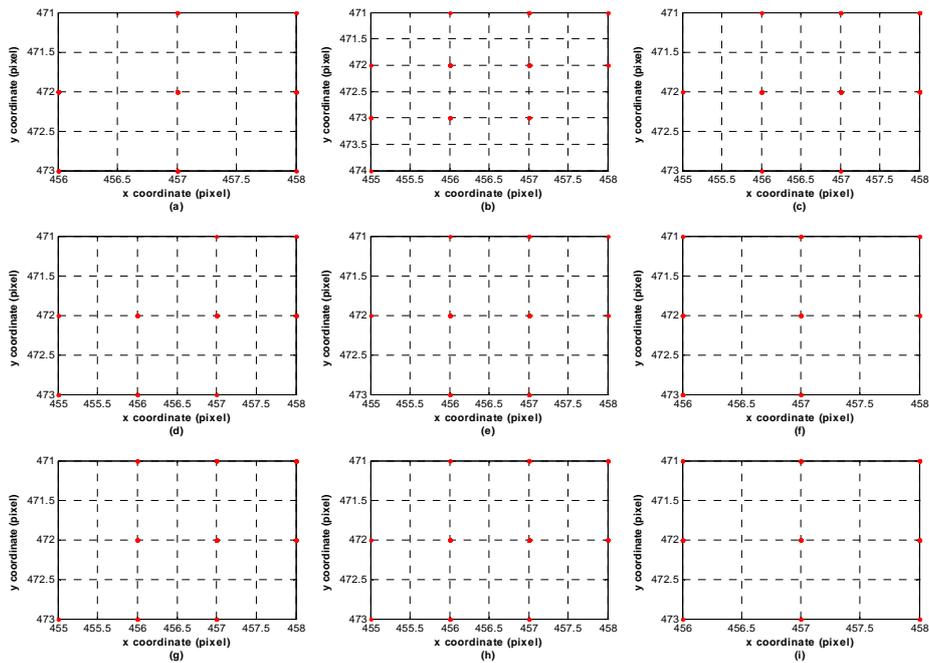


圖 5.8 利用 5.2.4 節的方法，在圖 5.4 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

從圖 5.5 我們可以觀察出，利用 5.2.1 節所介紹的方法，在九次實驗中，其重建點分佈較為凌亂，也會得到差異較大的數值，反應在滑鼠游標上，表示游標會在這些點上不停的跳動，以使用者的角度，就會認為此系統不夠穩定。而圖 5.6，即利用 5.2.2 節所介紹的方法，其重建點的分佈範圍略有減少，表示雖然滑鼠游標依然會跳動，但其跳動點會比較固定。所以在影像端經由平均後， I_{LS} 、 I_{LE} 、 I_{RS} 與 I_{RE} 在每次重建時，數值會比較接近，重建出來的值差就會比較小。

5.2.3 節和 5.2.4 節的方法反應在靜態指向上差別並不大，因為在第三種方法中我們是取佇列的前 9 點作平均，每次做完重建，則會把佇列的最前端點丟棄。因為從圖 5.1 中可以發現，每一秒鐘約有 9 個資料點，而第四個方法是當下時間前一秒的資料點，在每次重建時，也約為以 9 個資料點作平均，並且在每次觸發重建的時間點，在佇列中約新增加一個點，所以在靜態指向上的差別並不大。表 5.1 為此次實驗四種方法的 x 、 y 座標數值分佈整理。

表 5.1 四種方法所得的重建點分佈範圍數據表。

(單位：像素)

	Method 1	Method 2	Method 3	Method 4
X Range	453~462	452~460	453~460	455~458
Y Range	467~475	469~474	470~475	471~474

我們將指向物固定在另一個位置，如圖 5.9 所示，再作一次相同的實驗。

圖 5.10 至圖 5.13 分別是以 5.2.1 節至 5.2.4 節介紹的四種方法，在圖 5.9 中的位置所重建出來的座標點。



圖 5.9 系統畫面截圖。分別為左右兩攝影機視角的畫面。

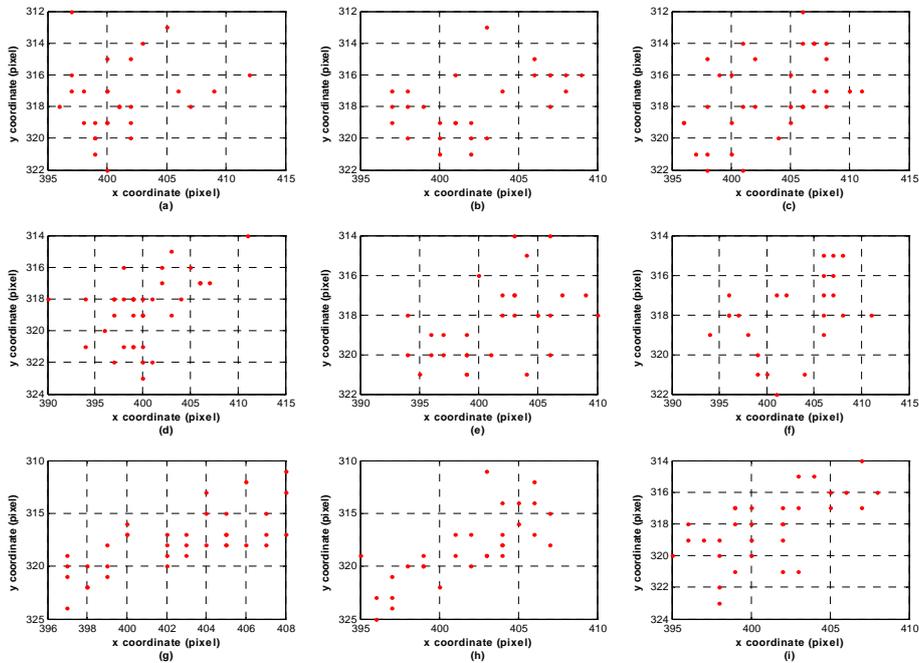


圖 5.10 利用 5.2.1 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

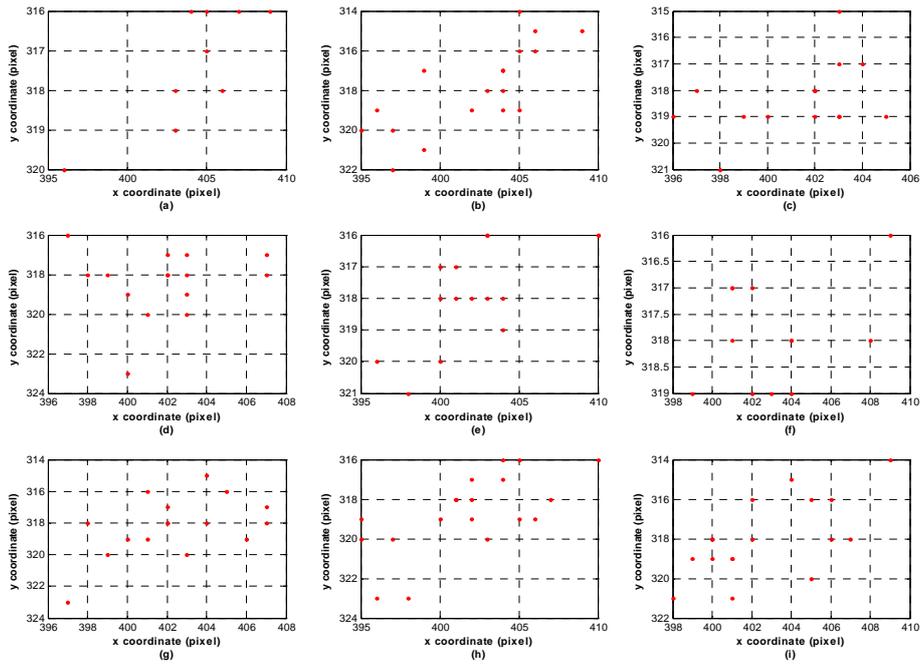


圖 5.11 利用 5.2.2 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

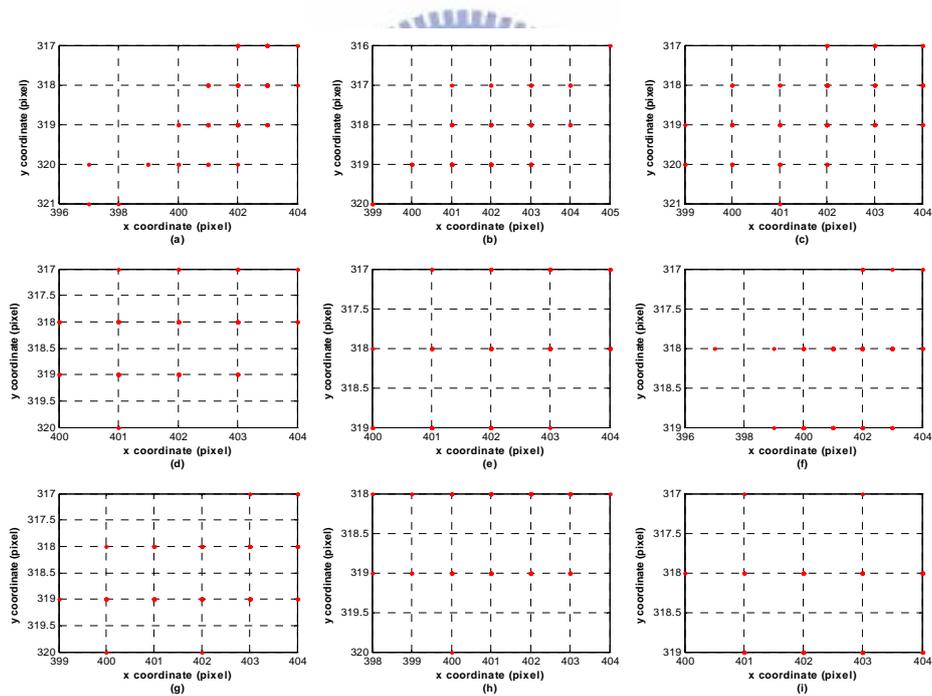


圖 5.12 利用 5.2.3 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

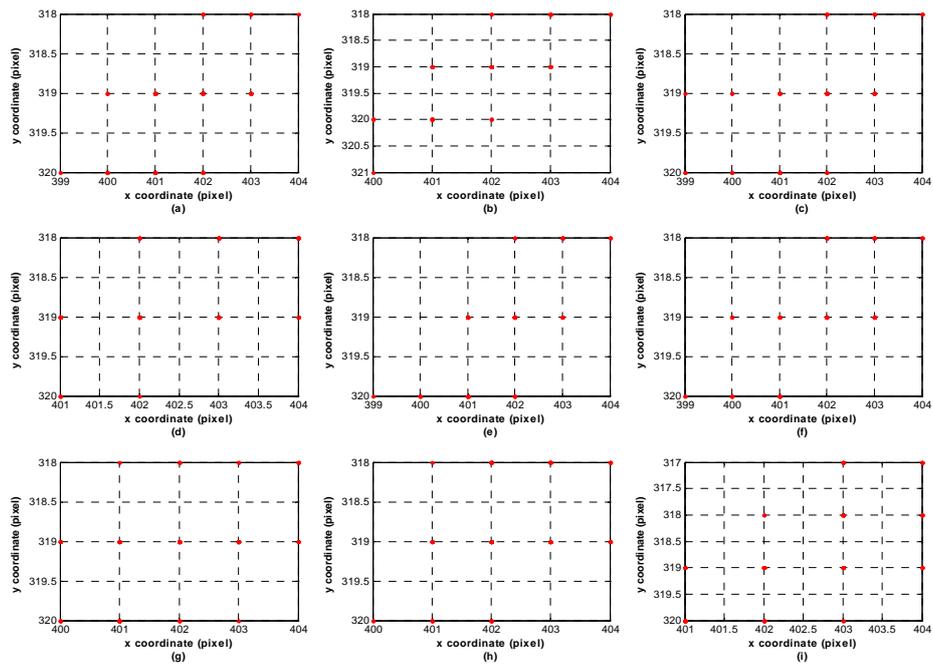


圖 5.13 利用 5.2.4 節的方法，在圖 5.9 中的位置，重建出來的分佈點。(a) - (i) 分別表示系統在同個位置執行 9 次所得到的結果。

表 5.2 四種方法所得的重建點分佈範圍數據表。(單位：像素)

	Method 1	Method 2	Method 3	Method 4
X Range	394~412	395~410	397~404	399~404
Y Range	311~325	314~323	316~321	317~321

表 5.2 為第二次實驗四種方法的 x 、 y 座標數值分佈整理。從此次的數據可以發現，誤差範圍比第一次的實驗大，其原因是指向物所在的位置，誤差的干擾較為嚴重。我們在多作幾次實驗後，歸納四種方法大致的誤差範圍，如表 5.3 所示，可以發現法四所得到的誤差範圍是最小的。

表 5.3 四種方法平均的誤差範圍。(單位：像素)

	Method 1	Method 2	Method 3	Method 4
X Range	12	10	7	4
Y Range	10	8	5	3

5.4 四種方法在動態指向上的實驗

在動態指向中，由於在實驗場景中無法精確地讓指向物固定在同一平面上移動，所以我們讓使用者站在固定的位置，手持指向物分別沿著鉛直及水平的方向，在不同的鉛直線與水平線中比劃。我們分別以 5.2.1 節至 5.2.4 節所述的四種方法，在鉛直與水平方向各作了 6 次的實驗，每次的實驗時間固定約 15 秒鐘，而 5.2.1 節與 5.2.2 節所介紹的方法中，是以五點平均來做底下的實驗。圖 5.14 至圖 5.17 分別表示 5.2.1 節至 5.2.4 節所述的方法，在鉛直與水平方向的實驗結果。紅色線是沿著鉛直方向的實驗，藍色線是沿著水平方向的實驗，但為了方便觀察，在描繪鉛直線的實驗點時，我們將 x 、 y 座標軸對調來描繪，(a) - (f) 則分別表示 6 次的實驗。

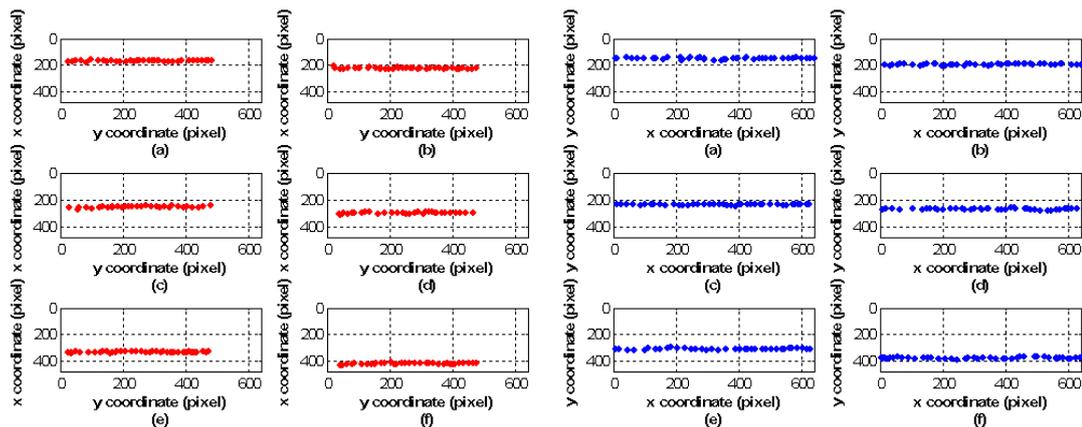


圖 5.14 利用 5.2.1 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。

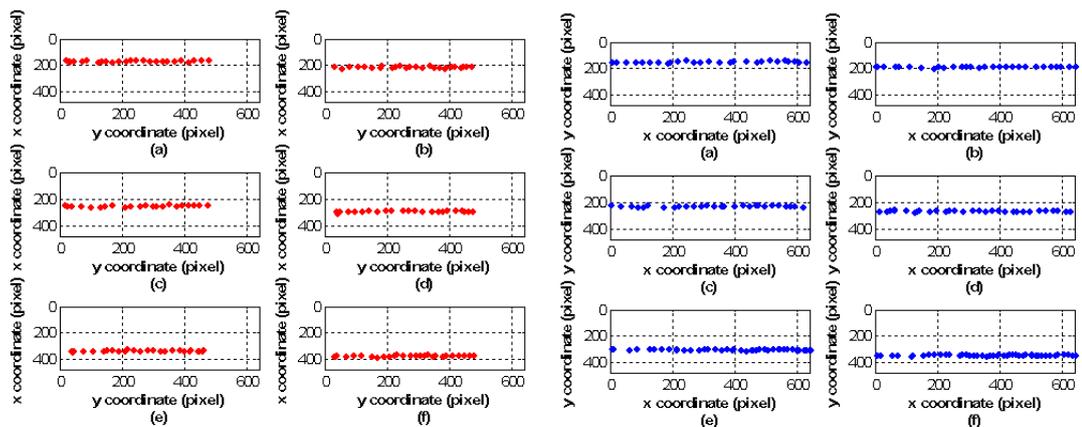


圖 5.15 利用 5.2.2 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。

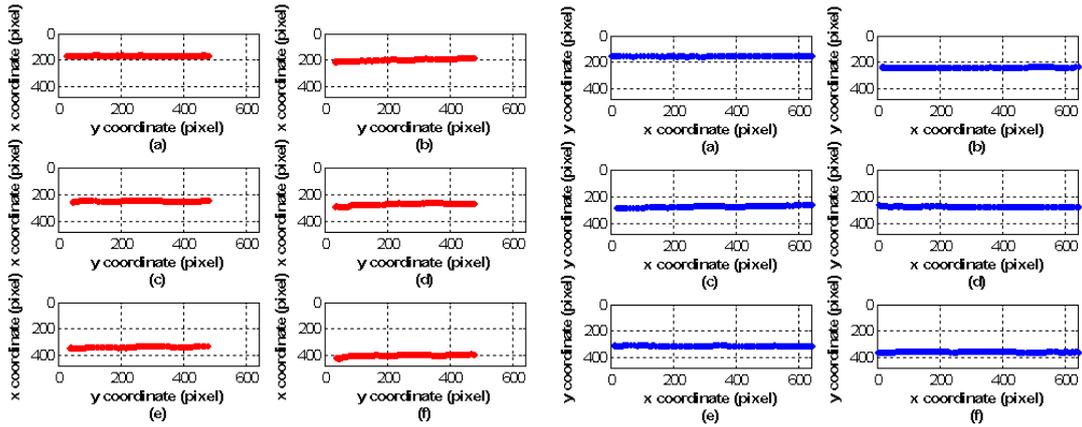


圖 5.16 利用 5.2.3 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。

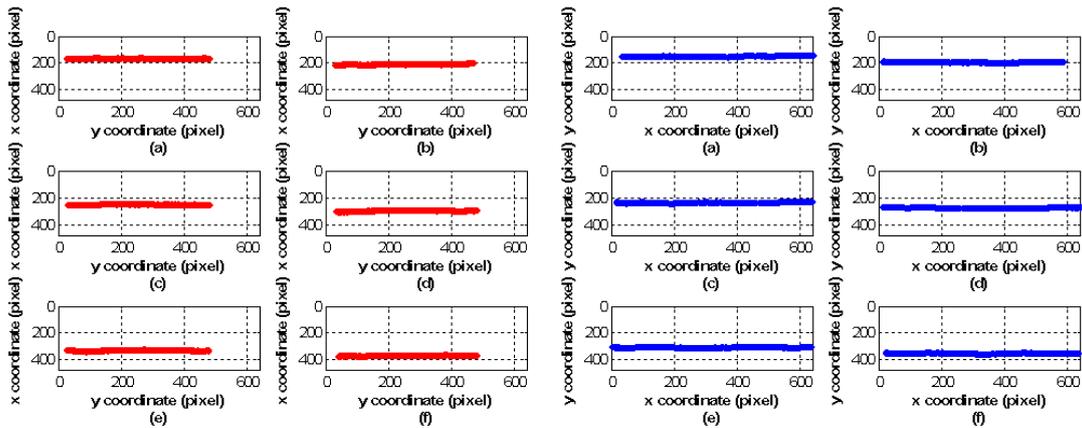


圖 5.17 利用 5.2.4 節的方法，沿著鉛直及水平的方向，各比劃六次的結果。

從圖 5.14 與圖 5.15 可以發現，方法一與方法二中所呈現的點數較為零散，也較為凌亂。這是因為此兩方法是累積五點 $t_{I_L} = t_{I_R}$ 的資料點才作運算。所以在每一滑鼠游標之間的時間間隔會較大，看起來就會較為零散，也因此會導致 Latency 較長。由於是選取 $t_{I_L} = t_{I_R}$ 的資料點，但因為 t_{I_L} 與 t_{I_R} 時間點常會有差 1 的情況，導致我們選取作平均的資料點時間間隔太長，平均點間的數值差距就會較大，以致於重建結果會較為凌亂，重建分佈範圍較廣，但約略可以感覺每次所指的是一直線。

而從圖 5.16 與圖 5.17 中可以發現，方法三與方法四重建點數較多且較密，表示滑鼠游標一直在改變，其 Latency 也就會較短。方法三的跳動情形比方法四

明顯，這是因為每次作重建時，都是取佇列中的前九個資料點作平均，所以如果使用者在進行指向的過程中，有過於劇烈的晃動，則有劇烈晃動的幾張畫格產生的資料點，將會很明顯地影響到之後的指向點重建位置，就會造成指向錯誤。而方法四因為是選取在系統重建當下時間點的前一秒資料點，所以只與重建當下時間點的前一秒的資料點有關，故不管是因為雜訊所造成之前的資料點偏差，或之前的指向情況，將不會影響此次的重建，所以系統也會較為穩定。

從這次的動態指向實驗中，可以看出，整體移動的趨勢是存在的，即滑鼠游標會跟隨著使用者所指的路徑移動。而水平移動的穩定性較鉛直移動的穩定性高，我們預測一方面是使用者在指鉛直線時，手抖動的情況比在指水平線時嚴重。另一方面，則是可能平面交角的關係，誤差對鉛直方向的影响較水平方向的影响大。故當使用者在鉛直方向比劃時，誤差跳動的情況會較水平方向比劃時嚴重。



5.5 結論

本章中，我們在系統流程的不同步驟中，利用簡單的平均觀念來達到降低雜訊對重建點造成的誤差，讓每次重建時，指向物的兩端點座標變化幅度小，使得重建點的跳動幅度小，進而達到穩定的重建效果。在我們提出的四種簡單的方法中，取重建時一秒內的指向物端點平均作重建，所得到的效果最好，誤差範圍最小。以此方法，在靜態指向中，由於多數點作平均，且不受時間點的影響，所以在每一次重建時，指向物端點的平均值差距很小，故重建點較為穩定；而在動態指向中，由於是取重建時一秒內的資料作平均，所以計算出來的重建點座標，約為重建前 0.5 秒的位置，對於人的感覺來說，應該是可以接受的延遲時間。所以在系統中，最後我們採用重建時一秒內的指向物端點平均作重建的方法。

第六章 結論與未來展望

6.1 結論

在本篇論文中，我們參考[8][9]的作法，提出並實作了不同於以往方法的指向系統。我們基於立體視覺，先擷取出在影像上的指向物端點座標，以平面投影轉換和三維幾何的觀念為基礎，計算出指向物的朝向，最後正確地重建出重建點，並控制滑鼠游標位置，藉以完成一套簡單的指向系統。我們在影像端與滑鼠端作分析，以簡單的平均概念，來降低雜訊、量測及計算上對系統穩定度的影響。我們也透過分析在影像端的誤差，與重建誤差間的關係，瞭解到在影像端的小誤差，可能會對系統造成極大的影響，且誤差對指向線在鉛直方向的影響較水平方向的影響大。從實驗結果可以知道，滑鼠游標雖然會有些許跳動，但其穩定度是可以接受的，而滑鼠游標會循著使用者所指的路徑移動，使得整個系統能正確的運作。



6.2 未來展望

我們在系統中，是先經由連接物體標記，再利用主成份分析法，來判斷指向物的朝向，但主成份分析法會把所有判斷為指向物的點皆計算，所以只要一有雜訊的影響，計算出的朝向就會有所偏差，導致系統的不穩定。故可以嘗試以別種演算法取代主成份分析法，降低雜訊對系統的影響。而在系統中，相機中心、參考平面、指向平面的座標，皆是以量測的方式取得，若系統是固定在某一地方，則可以先以電腦視覺的理論先求出相機中心的精確位置，而參考平面的角點則可以影像處理的演算法進行偵測，相信對於系統的精確度會有所助益。

目前攝影機是固定在實驗場景，利用已知的三維空間資訊，系統才能正確地運作。若能將這些已知的資訊去除，將手動量測的部分改以自動偵測的方式進行，則可以發展成在任何地點，只要使用兩台攝影機，從影像端能夠正確地判斷

出指向物，並利用三維幾何運算，就可以利用本論文所提出的方法進行指向，提高系統的可攜性。

目前僅對系統進行誤差分析，在未來則可以根據誤差的特性，如鉛直線的變化比水平線的變化大，設計一套使用者介面，根據使用者比劃至不同的區域給予不同的指令。而目前系統僅支援一位使用者，未來則可以發展成多人可同時使用的指向系統。



參考文獻

- [1] R. Sukthankar, R. Stockton, and M. Mullin, "Smarter Presentations: Exploiting Homography in Camera-Projector Systems," *Proceedings of International Conference on Computer Vision*, pp. 247-253, 2001.
- [2] J.-F. Lapointe, and G. Godin, "On-Screen Laser Spot Detection for Large Display Interaction," *Proceedings of the IEEE International Workshop on Haptic Audio Environments and their Applications (HAVE)*, pp. 72-76, 2005.
- [3] D. Laberge, and J.-F. Lapointe, "An Auto-Calibrated Laser-Pointing Interface for Collaborative Environments," *Proceedings of the VSMM 2003 – Ninth International Conference on Virtual Systems and Multimedia*, pp. 501-508, 2003.
- [4] A. A. Argyros, and M. I.A. Lourakis, "Vision-based Interpretation of Hand Gestures for Remote Control of a Computer Mouse," *Proceedings of the HCI'06 workshop (in conjunction with ECCV'06)*, LNCS 3979, pp. 40-51, 2006.
- [5] S. Sato, and S. Sakane, "Interactive hand pointer that projects a mark in the real work space," *Transactions of the Institute of Electrical Engineers of Japan*, Vol. 121-C, No. 9, pp. 1464-1470, 2001.
- [6] Y. P. Hung, Y. S. Yang, Y. S. Chen, I. B. Hsieh, and C. S. Fuh, "Free-Hand Pointer by Use of an Active Stereo Vision System," *Proceedings of 14th International Conference on Pattern Recognition*, pp. 1244-1246, 1998.
- [7] E. Hosoya, H. Sato, M. Kitabata, I. Harada, H. Nojima, and A. Onozawa, "Arm-Pointer: 3D Pointing Interface for Real-World Interaction," *Processings of the European Conference on Computer Vision Workshop on Human Computer Interaction*, pp. 72-82, 2004.
- [8] J. C. Hsu, "An Implement of a Rel-time Human Computer Interaction Application under Stereo Vision-based Environment," Master Thesis, National Chiao Tung University, 2006.
- [9] J. S. Huang, "Accuracy Analysis of a Stereo Vision-based Pointing System," Master Thesis, National Chiao Tung University, 2006.

- [10] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Elsevier Academic Press, 2003.
- [11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [12] D. Forsyth and J. Ponce, *Computer Vision A Modern Approach*, Prentice Hall, 2002.

