

# 國立交通大學

電控工程研究所

碩士論文

基於立體視覺之移動式機械臂  
影像伺服設計

Visual Servo Design of a Mobile Manipulator  
Based on Stereo Vision

研究生：張哲豪

指導教授：宋開泰 博士  
林清安 博士

中華民國九十九年七月

基於立體視覺之移動式機械臂影像伺服設計

Visual Servo Design of a Mobile Manipulator

Based on Stereo Vision

研究生：張哲豪

Student: Che-Hao Chang

指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song

林清安 博士

Dr. Ching-An Lin

國立交通大學



**Submitted to Institute of Electrical Control Engineering  
College of Electrical and Computer Engineering  
National Chiao Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of Master  
in  
Electrical Control Engineering  
July 2010  
Hsinchu, Taiwan, Republic of China**

中華民國九十九年七月

# 基於立體視覺之移動式機械臂影像伺服設計

學生：張哲豪

指導教授：宋開泰 博士  
林清安 博士

國立交通大學 電控工程研究所

## 摘要

本論文之主旨在設計一移動式機械臂的抓物控制系統。藉由安裝於機器人頭部之立體攝影機擷取影像資訊，使機器人得以在環境中找尋目標物體，並自主導引機器手臂至適當的抓取位置，完成抓取之動作。本文採用加速強健特徵點演算法(Speed up robust feature, SURF)來定義目標物體的特徵點，並且藉由比對當前畫面中的特徵點來判斷目標物是否存在於影像中。為了強化特徵比對結果並算出抓物控制所需的參考點，本論文採用隨機取樣篩選演算法(RANdom Sample Consensus, RANSAC)來估測平面轉換矩陣(Homography matrix)以準確的標出目標物的中心點。本論文並發展出一套座標估測的校正法，提高對於目標物座標估測之準確度。在影像伺服的控制設計上，針對機械手臂及移動平台設計了以座標估測結果產生控制命令的控制方法，導引移動式機械臂自主抓物。在本論文中，以實驗室自行設計的多自由度雙臂機器人搭配全向式移動平台，並裝置立體攝影機作為影像資訊輸入，驗證整套方法的可行性。經過實驗驗證，此系統可以導引機器人順利的移動並拿取所設定的目標物。

# Visual Servo Design of a Mobile Manipulator

## Based on Stereo Vision

Student: Che-Hao Chang

Advisor: Dr. Kai-Tai Song

Dr. Ching-An Lin

Institute of Electrical Control Engineering  
National Chiao Tung University

### ABSTRACT

The objective of this study is to design a grasping system for a mobile manipulator, such that it can find and grasp a target in the environment. Speed up robust feature (SURF) algorithm is used to define the feature of a target object and to match features between current image and object database to confirm the target. To strengthen the feature matching results and calculate the necessary reference control point, we adopt RANSAC(RANdom Sample Consensus) algorithm to estimate the planar transformation matrix (homography matrix) in order to accurately mark the center of target. A set of coordinate estimation correction method is developed to improve the accuracy of target location estimation. A control design is developed based on coordinate estimation for the mobile manipulator for visual servoing for object grasping. Experiments on an self-constructed mobile manipulator reveal that the proposed method can find and grasp a target object successfully.

## 誌謝

謹向我的指導教授宋開泰博士致上感謝之意，感謝他兩年來在專業上和生活上待人處事的指導，以他豐富的學識與經驗，配合理論的應用，使得本論文得以順利完成。

也要感謝學長孟儒、嘉豪、格豪、學姊巧敏在理論與實驗過程中的指導，以及感謝同學信毅、宗暘、維峻、傑巽、奕玚在研究過程互相勉勵與打氣，同時要感謝學弟仕晟、建宏、上峻、碩成、家昌、章宏在生活上與課業中帶來的樂趣。

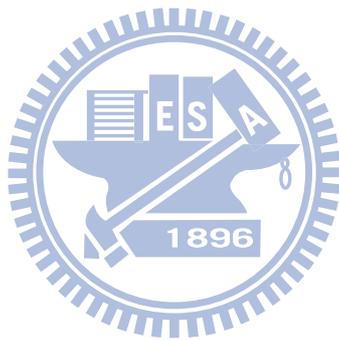
最後，特別感謝我的父母以及所有家人，由於他們的辛苦栽培與鼓勵，在生活上給予我細心地關愛與照料，並支持我就讀碩士學位，使得我才得以順利完成此論文，在此我願以此論文獻給我最感激及最偉大的父母親。



# 目錄

摘要.....	i
ABSTRACT.....	i
誌謝.....	i
目錄.....	i
圖目錄.....	i
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 相關研究.....	1
1.3 問題描述.....	5
1.4 系統架構.....	6
1.5 章節說明.....	7
第二章 目標物辨識.....	8
2.1 特徵點擷取.....	9
2.1.1. SURF 工作原理.....	9
2.1.2. 影像特徵點比對.....	13
2.2 錯誤比對剔除.....	14
2.2.1. Homography.....	14
2.2.2. RANSAC.....	16
2.3 定義中心點.....	19
第三章 目標物姿態估測.....	21
3.1 立體視覺原理.....	21
3.2 3D 座標公式校正.....	23
3.2.1. Z 軸座標值估測.....	24
3.2.2. X 軸座標值估測.....	28
3.2.3. Y 軸座標值估測.....	32
3.3 三軸旋轉角.....	36
第四章 移動式機械臂之影像伺服控制設計.....	39
4.1 手臂控制.....	39
4.1.1. 硬體構造.....	39
4.1.2. 座標轉換.....	39
4.1.3. 手腕逆運動學推導.....	44
4.1.4. 從假定腕部點逼近實際腕部點的方法.....	46
4.1.5. 抓取動作路徑規劃.....	48
4.2 移動平台控制.....	49
4.2.1. 硬體構造.....	49
4.2.2. 平台運動模型.....	51

4.2.3.	姿態估測 (Odometry).....	54
4.2.4.	平台之運動控制.....	55
4.3	影像伺服控制.....	57
4.3.1.	視覺伺服控制架構.....	57
4.3.2.	控制設計.....	58
第五章	實驗結果.....	62
5.1	實驗用機器人構造.....	62
5.2	平台靜止抓取實驗.....	65
5.3	平台移動抓取實驗.....	69
第六章	結論與未來展望.....	74
6.1	結論.....	74
6.2	未來展望.....	74
參考文獻	.....	76



## 圖目錄

圖 1.1 姿態(pose)追蹤的結果[10].....	3
圖 1.2 物體模型建立的基本流程[11].....	3
圖 1.3 影像辨識流程[12].....	4
圖 1.4 目標物辨識的 3D 點群結果[16].....	5
圖 1.5 Database 在目前影像中的比對結果，藍色的框框表示 2D 的定位結果 [17].....	6
圖 1.6 系統流程圖.....	7
圖 2.1 直接對左右影像做對應點計算的結果.....	8
圖 2.2 計算 D 區域像素總和.....	10
圖 2.3 從左到右分別是在 y 及 xy 方向高斯二階偏微分，以及其中利用 box 濾波器的近似。而這些灰色的區域近似於零[2]。.....	11
圖 2.4 區域極值計算[1].....	12
圖 2.5 這種場景清楚的顯示出從 Hessian-based 偵測器所得到特徵的性質[2] .....	12
圖 2.6 一個塗鴉場景的細節顯示出在不同尺度下描述器視窗的大小[2].....	13
圖 2.7 哈爾小波響應轉換結果統計圖[2].....	13
圖 2.8 透過轉換矩陣 H 得到資料庫物件在目前影像平面中之位置.....	16
圖 2.9 左圖為一群分布不均勻的點集合，需要從這裡面決定出一條直線。 右圖為經過 RANSAC 計算後，得到符合條件的直線，不符合的點則被 剔除在外。[22].....	18
圖 2.10 用 SURF 做特徵比對的結果.....	18
圖 2.11 用 RANSAC 做 homography estimation 的結果.....	18
圖 2.12 實際左影像跟資料庫影像做轉換矩陣計算的結果.....	19
圖 2.13 實際右影像跟資料庫影像做轉換矩陣計算的結果.....	19
圖 2.14 目標物在影像平面的區域及中心點位置的定義.....	20
圖 3.1 立體影像處理的模型[23].....	21
圖 3.2 攝影機跟世界座標關係的上視圖[23].....	22
圖 3.3 校正版圖示.....	24
圖 3.4 距離為 19 格時(532mm)左圖為左攝影機之影像，右圖為右攝影機之 影像.....	24
圖 3.5 Z 座標估測值.....	26
圖 3.6 校正後 Z 座標誤差值.....	27
圖 3.7 Xl 座標估測值.....	28
圖 3.8 Xr 座標估測值.....	29
圖 3.9 校正後 X 座標誤差值.....	31

圖 3.10 Yl 座標估測值.....	33
圖 3.11 Yr 座標估測值.....	34
圖 3.12 校正後 Y 座標誤差值.....	36
圖 3.13 目標物的三軸向量與世界座標的旋轉關係.....	37
圖 4.1 機械手臂設計圖.....	40
圖 4.2 機器人各部位及目標物之間的座標關係.....	41
圖 4.3 base 跟 arm 座標軸的關係.....	42
圖 4.4 機器人頭部跟目標物的座標關係.....	43
圖 4.5 移動平台跟目標物的座標關係.....	43
圖 4.6 (a)機械手爪設計圖(b)手臂各點代號位置.....	45
圖 4.7 機械手臂各關節點之間的關係圖.....	45
圖 4.8 空間中向量及平面的夾角關係.....	46
圖 4.9 給定抓取點 b', 假定手腕點 c', 對應 c 的抓取點 a', 對應 b 的手腕點 c', 四點之間的相對關係圖。.....	47
圖 4.10 逼近抓取點的示意圖, 隨著 b' 漸漸接近 b, 我們可以得到對應 b 的腕部點 c'.....	48
圖 4.11 夾取之動作規劃示意圖.....	49
圖 4.12 全向輪實體圖[24].....	50
圖 4.13 平台四輪配置圖[24].....	51
圖 4.14 全向輪轉動示意圖[24].....	52
圖 4.15 全向輪之幾何關係[24].....	53
圖 4.16 平台座標旋轉示意圖[24].....	53
圖 4.17 平台運動模型圖[24].....	56
圖 4.18 平台控制方塊圖[24].....	57
圖 4.19 移動式機械臂工作模式流程圖.....	59
圖 4.20 整體系統架構圖.....	60
圖 5.1 實驗用機器人.....	62
圖 5.2 Bumblebee2 立體攝影機.....	63
圖 5.3 機器人硬體架構圖.....	64
圖 5.4 (a)資料庫物件及(b)其特徵點位置.....	66
圖 5.5 平台靜止時, 攝影機所擷取的畫面及目標物位置的計算結果.....	67
圖 5.6 量測出之目標物在影像平面上的中心點.....	67
圖 5.7 攝影機量測到目標物的座標位置.....	67
圖 5.8 抓取過程中手臂各軸軌跡圖.....	68
圖 5.9 平台靜止抓取實驗.....	68
圖 5.10 移動抓取實驗中, 起點時攝影機所擷取的畫面及目標物位置.....	69
圖 5.11 攝影機在起點所量測出之目標物在影像平面上的中心點.....	70
圖 5.12 移動抓取實驗中, 到達工作範圍時攝影機所擷取的畫面及目標物位	

置.....	70
圖 5.13 攝影機在工作範圍所量測出之目標物在影像平面上的中心點.....	70
圖 5.14 攝影機量測到的目標物座標位置.....	71
圖 5.15 抓取過程中手臂各軸軌跡圖.....	71
圖 5.16 平台移動抓取實驗.....	72
圖 5.17 平台移動軌跡紀錄.....	73



## 表目錄

表 3.1 不同距離時目標點在影像平面上的座標值.....	25
表 3.2 Z 座標未校正前的估測值 .....	26
表 3.3 Z 座標校正後的結果及誤差值 .....	27
表 3.4 Xl 座標未校正前的估測值.....	29
表 3.5 Xr 座標未校正前的估測值 .....	30
表 3.6 X 座標校正後的結果及誤差值.....	31
表 3.7 Yl 座標未校正前的估測值.....	32
表 3.8 Yr 座標未校正前的估測值 .....	33
表 3.9 Y 座標校正後的結果及誤差值.....	35
表 3.10 三軸旋轉角實際測試結果.....	38
表 5.1 Bumblebee2 立體攝影機之規格[29].....	63
表 5.2 使用 Matlab Toolbox 得到的左攝影機內部參數.....	65
表 5.3 使用 Matlab Toolbox 得到的右攝影機內部參數.....	65
表 5.4 使用 Matlab Toolbox 得到的攝影機外部參數.....	65



# 第一章 緒論

## 1.1 研究動機

由於近年來人類不斷的創新研發，致使科技的發展日益提升，電腦運算能力大幅提高，可以透過各種不同的感測器以及電腦高速的運算能力來完成複雜的工作。因此，相較於過去傳統僅在工廠中固定式的機械手臂，現在已經有越來越多的機器人開始走進人類的生活周遭，並且有許多也已朝商品化的方向發展，例如吸塵機器人或寵物機器人等等。尤其近年來時常會有智慧型機器人的公開發表，由這個趨勢來看，機器人與人類的生活之間將會有更密切的關係。

所有科技發展的目的都是為了使人類的生活更加便利，而機器人的研究當然也不例外，如何讓機器人能夠幫助人類甚至是代替人類做一些繁複瑣碎或是危險的事，都是所有在這個領域努力的人們所追求的目標。如同人類一般，視覺掌握了人類大多數的資訊來源，因此一個具有影像系統，機械手臂加上移動功能的機器人，將會對我們的生活有相當大的幫助。以往的機器人大多是存在於工廠中，在單純的環境中工作，重複著同樣的動作，但是現今的發展中，機器人與人互動漸趨重要，因應這個需求，勢必需要一個穩定又安全的移動式抓取系統，才能讓人機互動時具有安全性的保障。

此研究的主要目的在於實現一台具有手臂的移動式機器人，使其能夠辨識出指定的物品後達成拿取物體的功能。透過裝置於頭部的攝影機，辨識出一般日常生活中的用品，藉此估測出目標物體所在的位置，並且能由此位置資訊導引機器人前進並抓取目標物。

## 1.2 相關研究

對於建立機器人的自主抓取能力，有許多方面的問題需要解決，例如：偵測環境的能力、搜尋物體的能力、移動路徑規劃的能力以及抓取物體的能力，有了這些能力的綜合運用，才能確保機器人在環境中能搜尋到目標物，並且能夠透過適當的運動控制到達目標物所在位置，進而達到抓取該目標的目的。

近年來在辨識物體方面最常見的方法為 SIFT[1]和 SURF[2]，許多的方法都以此為基礎延伸下去。文獻[3-5]都是用立體攝影機擷取影像資訊將目標物周邊的環境建立起來，知道了整體環境情況之後就能針對抓取最適當的路徑規劃。文獻[3]除了利用 SIFT 的特徵點資訊之外還加上了幾何的特徵，將物體的邊緣都描繪出來，如此一來目前影像中的環境資訊就能更清楚的顯示出來，此外作者們還提出了一個方法能夠根據目前影像來決定下一時刻最好的視角位置。文獻[4]所用的方法是將所有物體的以方塊的形式描繪出來，再將目標物標色，模擬出機械手臂抓取的路徑，找出最適當的抓取角度。文獻[5]能夠做到即時的在整個工作區域當中辨識出目標物並將其替換成三維模型。文獻[6]則是以 SIFT 作為其能夠在複雜環境中辨識出物體的方法，以三個目的建立其視覺系統，分別是辨識目標物，追蹤目標物及克服複雜環境，最後透過一個六軸的機械手臂來完成抓取任務。文獻[7]發展出一套結合反投影直方圖及 SURF 的辨識方法，讓救援機器人能夠在一個未知的環境中辨識出危險標誌，自我定位，劃出環境地圖讓搜救隊在急難救助時能對災害現場有一個參考的依據。

為了讓機器人在一個智慧型的居家環境中辨識出目標物，文獻[8]發展出一套方法，將射頻辨識技術(RFID)加入辨識架構中，目的是讓未來的服務型機器人能夠達到便宜且強健的目的。為了讓機器人在居家服務更有效率，M. Aiguo, et al 在文章中提出一個方便且有效率的訊息融合系統包含 RFID、立體攝影機、雷射掃描儀在其中[9]。Changhyun Choi, et al 提出一套可以同時用在單眼視覺跟立體視覺之中的即時目標物姿態估測跟追蹤方法[10]，他用的是目標物上的自然特徵，以這些特徵當作追蹤點，攝影機在不同角度的情況下，都能將目標物的姿態清楚的描繪出來，如圖 1.1 所示。

由 Tomono, et al.提出的研究結果[11]，係利用一個移動式機器人建立整體環境的物體地圖，其首先藉由雷射掃描整個環境，建立出整體二維環境的地圖，接著透過 dense shape model 的方式建立物體的模型再加上 SIFT 的特徵點擷取，便可以判斷出該目標物以及所在位置。其建立物體模型的流程如圖 1.2 所示。

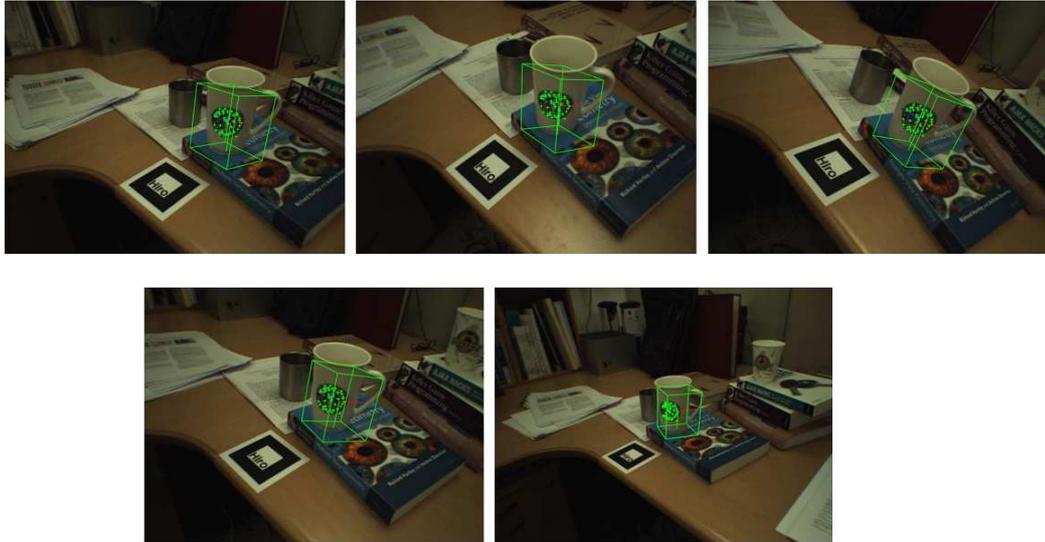


圖 1.1 姿態(pose)追蹤的結果[10]

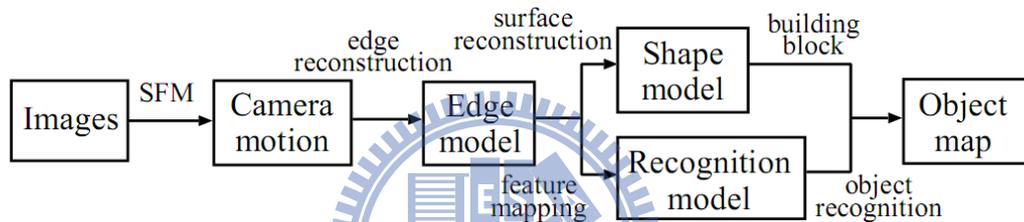


圖 1.2 物體模型建立的基本流程[11]

在抓取的規劃上，Yamazaki, et al.則利用 SFM(Structure from motion)的方式描繪物體的 3D model[12]，並描述物體每個像素上的 Oriented points，但亦由於資料量較大，因此透過 voxel model 將資料量降低。並且計算在滿足所設定的條件下，所有可能的抓取方式，如圖 1.3 所示，並從中找尋最佳的姿態(pose)，利用 Oriented points 描繪出物體，以及所有可能的抓取方式，並且最後計算出夾爪最適當的抓取位置跟角度。

除了單一機器人抓取之外，Hirata, et al 提出了一個多機器人合作來拿同一個物體的運動控制演算法[13]，克服了操作時手臂動作及輪子移動之間的時間延遲而衍生出的問題，這個問題會使人在操作機械臂拿取物體時產生困難，但是在這個新的控制方法之下，人們就能操控機器人合作達成任務而不用考慮這些控制的問題。

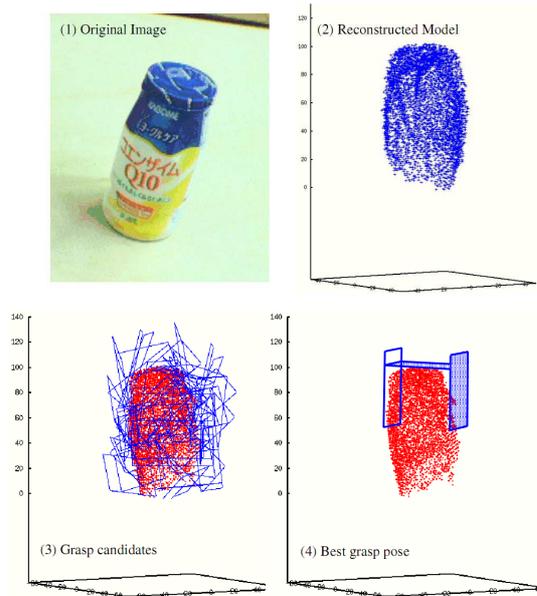


圖 1.3 影像辨識流程[12]

在整體的移動式抓取設計方面，文獻[14-17]所提出的方法之中有許多值得參考的地方。由 Christensen, et al.所提出的研究結果[14]，發展出一隻具備定位、導航、物體辨認、追蹤、抓取規劃的機器人，此機器人為 eye-in-hand 的構造。擁有一套演算法可以從特徵擷取，目標物位置及姿態(pose)估測，到最後完成抓取的功能。由 Lee, et al.提出來的研究結果[15]中，發展出一套任務導向的抓取演算法，用在規劃對於不同的任務在物體上做適當的抓取。在運作的同時，利用一個外部攝影機，透過不同角度的影像資訊，可以使機器人同時觀察手及物體來進行操作。

在攝影機的擺放位置方面，若是能放置於頭部的位置，如[16][17]，就能夠改善在 eye-in-hand 的構造中，抓取到目標物後，攝影機就失去繼續接受資訊的功能，無法繼續新的任務。[16]提出的研究結果，發展出一套演算法，可以做到物體的辨識跟定位，藉由 SIFT 特徵點擷取建立 3D 的點群資訊，如圖 1.4 所示，描述出物體的 3D pose，利用眼手座標關係，完成抓取的任務，並藉由精準的校正達到抓取過程中的避障功能。



圖 1.4 目標物辨識的 3D 點群結果[16]

由 Azad, et al.等人提出的研究結果[17]，發展出一套用在抓取的基於立體影像的物體 6D 定位演算法，應用在人形機器人 ARMAR。由五個步驟完成抓取的任務，一開始分別用左右眼取特徵點，把每個比對成功的轉換成 3D 座標，再將這些 3D 座標集合相稱的放到 3D 平面上，最後得到目標物的四個角落框出表面，如圖 1.5，並給定他們的世界座標。

近期的研究中，將 RANSAC 及 Homography 加入 3D 姿態估測中也越來越常見，[18]所提出來的座標估測方法中就有將這兩個方法用在其中，應用在無人遙控直升機在飛行時的定位，藉由於地面上的參考目標或停機坪，定義無人飛機上的攝影機座標系，此文獻中所提出的演算法對於直升機的 3D 位置及朝向的估測方面都具有很高的品質，可以將結果應用在複雜的控制如自主降落及低空準確定位和下降等任務。

### 1.3 問題描述

在一個具有抓取功能的移動式機械手臂的研究中，首先會遇到的問題就是如何在環境中辨識出欲抓取的目標物。我們期望能在較少的先期資訊之下，不必耗費很多時間在建立資料庫影像。利用物體本身所具有特徵點的辨識方法上，其優點在於可以運用簡單的辨識方法達成，以利於機器人在即時的环境下進行反應。

尤其對於辨識環境中 3D 的物體而言，還需考量如何可以盡量達到不受物體旋轉、視角改變以及遮蔽物的影響，正確的辨識出物體。而最近使用的辨識方法中以 SIFT 及 SURF 演算法較常見，他們具有抗旋轉、尺度不變的特性，可以針

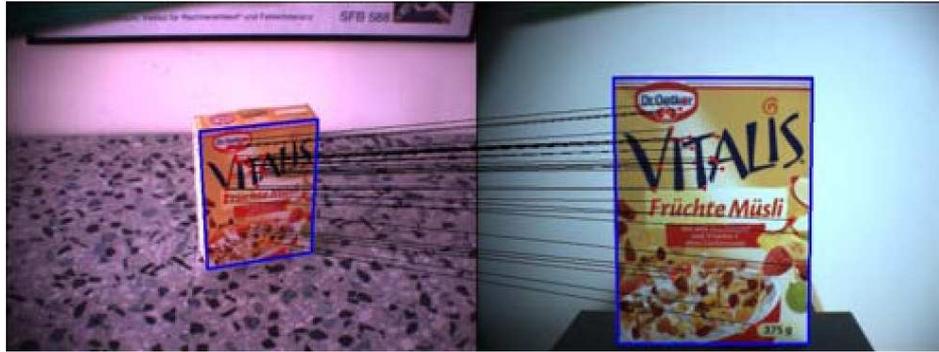


圖 1.5 Database 在目前影像中的比對結果，藍色的框框表示 2D 的定位結果[17] 對不同的目標先行建立 database，不過在運算量較為龐大且可能出現錯誤比對的情況，所以需要有其他輔助篩選特徵點的演算法，提高辨識的準確性跟穩定度。

當順利辨識出物體後，其下一步驟便是估測目標物的所在位置，其中深度資訊決定了機器人在影像伺服過程中需要移動的距離，於是在本論文中採用立體視覺系統，藉由同一時刻的兩張影像，擷取其中的資訊，期望能提升在目標物位置估測上的準確度。

在移動式抓取的研究上，必須將機器人本身所有的部件彼此間的座標轉換關係弄清楚，且在控制上必須有一個穩定的參考點當作控制命令輸入，以利於導引機器人能夠順利的朝向目標物來移動。如何得到這個穩定的參考點就是本論文中一個重要的環節，如果能經由一般性的影像辨識方法就能找出這個參考點，而不用透過許多複雜的演算法，想必對解決問題會有很大的幫助。

如何運用這個參考點為中心，轉換成機器人在移動抓取過程中的控制命令，整體的影像伺服控制設計就顯得格外重要，且必須考慮到如何做出機器人各種行為之間的決策，如此一來，就能夠順利的完成移動抓取的任務。

## 1.4 系統架構

本論文使用實驗室研製的全向式移動平台，上面裝置自行設計的雙臂機械手臂，以及兩自由度的頭部機構，組合成一具完整的移動式機械臂(Mobile manipulator)。本系統所使用的主要感測器為立體攝影機，系統流程圖如圖 1.6 所示，藉由影像辨識演算法擷取目標物及目前影像之間成對的特徵點，算出目標

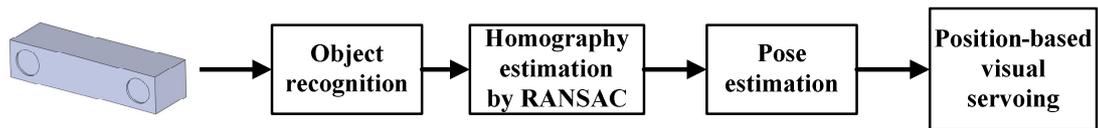


圖 1.6 系統流程圖

物在空間中的座標位置，作為導引平台移動及控制機械手臂抓取目標物的命令。藉由不斷的更新目標物的即時位置，以及讀取手臂跟平台所回傳的位置，比較兩者之間的誤差關係，來做為是否完成任務的判斷依據。

## 1.5 章節說明

本論文一共分成六章，第一章介紹研究動機與目的，並且概略介紹所設計之系統架構。第二章說明 SURF 特徵點擷取演算法及剔除錯誤比對的方法。第三章則是介紹目標物的姿態估測及座標量測的校正。第四章介紹本論文所用之機器人的影像伺服設計以及各部位控制設計的介紹。第五章為實驗結果，驗證整體設計上的可行性。第六章為結論與未來展望。



## 第二章目標物辨識

在本論文中所提出的影像伺服方法之中，最重要的一個環節，就是估測出目標物的三維座標。估測目標物的三維座標時，資訊來源就是目標物在左右攝影機影像中成對的特徵點，雖然我們可以直接將取到的左右攝影機影像直接做對應點的計算(correspondence)但是這樣得到的是整個影像平面上的對應點，在複雜背景的狀況下，整個影像平面都會是對應點，而單純背景但是存在數個物體時，如圖 2.1 所示，則會在不同物體上出現多組對應點，如何從複雜背景的情況下針對單一目標物找出我們所需的對應點，就是一個需要解決的問題。

為了解決這個問題，也就是如何找出目標物在左影像平面跟右影像平面之中的位置，我們將特徵比對設計成一開始將資料庫影像分別跟左右攝影機取到的影像做特徵比對，這樣就能得到兩組比對成功的特徵點，之後再將這兩組特徵點去做交叉比對，就可以得到目標物在左右攝影機影像平面上成對的特徵點。

另一方面，在本論文的控制器輸入需要一個在目標物上穩定的參考點，我們希望這個參考點能在目標物的中心位置，以利於在整個影像伺服控制的過程中，給予機器人一個準確的控制命令，使其能夠正確的移動到目標物前方，並穩定的抓取目標物，而不會因為控制命令的不正確，而使機器人無法移動到適當的抓取位置，導致抓取任務失敗。

雖然說我們可以將上述方法中所得到的成對特徵點取平均，當作控制用的參



圖 2.1 直接對左右影像做對應點計算的結果

考點，但是我們不能保證，這個平均值是位於目標物的中心，因為目標物的特徵點分布可能在任何位置。於是為了能夠得到穩定的參考點，我們將 homography 加入我們的特徵比對過程之中[19]，在資料庫影像分別跟左右攝影機影像做特徵比對之後，分別將這些成對的特徵點拿去算平面轉換矩陣，因為這個平面轉換矩陣包含著縮放及旋轉因子，所以我們可以由這個平面轉換矩陣準確的框出目標物在左右攝影機影像中的區域，接著再取其四個角點的平均，就可以得到一組穩定的對應點，也就是目標物的中心點，由這組中心點就能算出控制所需要的穩定參考點。

本章將說明算出這組對應點的方法。首先，2.1 節介紹特徵點擷取及比對的方法，2.2 節說明如何得到平面轉換矩陣(Homography matrix)，至於如何求得這組中心點的三維座標作為控制所需的參考點將在第三章會有詳細的說明。

## 2.1 特徵點擷取

要從目前攝影機所擷取到的影像中找出我們所要的目標物，需要有一個辨識物體的演算法。在本論文中，使用的是特徵點擷取。現存的特徵點擷取方法之中，以 SIFT 跟 SURF 較常見，經常在論文中被引用。由於立體攝影機需要同時處理兩張影像，於是我們採用了在運算上較為快速的 SURF 來作為本論文的特徵點擷取演算法。

### 2.1.1. SURF 工作原理

加速強健特徵點演算法(Speed Up Robust Feature, SURF)[2]為 Bay *et al.*所提出的一種新的尺度不變且抗旋轉的特徵點偵測及描述法。這個方法的設計概念在於發展出一套重複性、獨特性以及強健性優於現存方案的特徵點擷取演算法，且能夠有更快的運算速度。在計算過程中，首先採用 Integral image，定義為

$$\mathbf{I}_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} \mathbf{I}(i, j)$$

它是讓 SURF 能快速計算的關鍵[20]，概念為累積影像矩形區域的像素總和，並以此為基礎計算不同區塊代表的特徵值。例如：如圖 2.2 所示。當要計算 D 區域的像數總和時，首先要取出積分影像 1、2、3、4 的值，

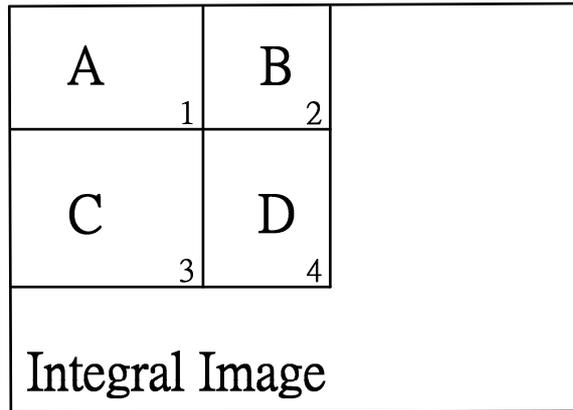


圖 2.2 計算 D 區域像素總和

接著根據定理  $D=4-2-3+1$ ，即可快速算出 D 區域像素總和。本章節將描述如何透過 SURF 所得到的特徵點資訊來描述目標物體，使機器人能夠從影像中辨識出物體存在的位置。

以下為 SURF 在運算過程中主要的步驟：

### 2.1.1.1 Fast-Hessian Detector

SURF 尋找特徵點的方式是基於 Hessian 矩陣的行列式，找出二次微分的極大跟極小值。在 SURF 演算法裡將 Hessian matrix 當作偵測器(Detector)，因為它在計算時間及精確度上具有很好的表現。給定一個在影像平面上的點

$X = (x, y)$ ，則 Hessian matrix  $H(x, \sigma)$  以尺度  $\sigma$ ，在  $X$  中被定義如下：

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2-1)$$

其中  $L_{xx}(x, \sigma)$ 、 $L_{xy}(x, \sigma)$  及  $L_{yy}(x, \sigma)$  是 Laplacian of Gaussians(LoG)。

在圖 2.3 中的 9x9 遮罩是  $\sigma=1.2$  高斯二階微分的近似值且代表我們的最低尺度，也就是最高空間的解。我們將  $D_{xx}$ 、 $D_{yy}$ 、 $D_{xy}$  標記為我們的近似值。而為了計算的效率應用，在長方形區域的權重被保持為簡單的形式，但是為了

Hessian's 行列式  $\frac{|L_{xy}(1.2)|_F |D_{xx}(9)|_F}{|L_{xx}(1.2)|_F |D_{xy}(9)|_F} = 0.912... \approx 0.9$ ，其中  $|x|_F$  是 Frobenius norm。

我們需要去進一步的平衡在表達強度上的權重。產生的結果如下：

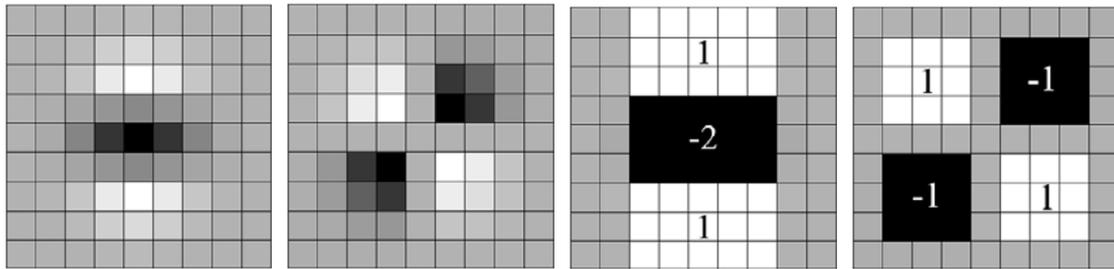


圖 2.3 從左到右分別是在  $y$  及  $xy$  方向高斯二階偏微分，以及其中利用 box 濾波器的近似。而這些灰色的區域近似於零[2]。

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (2-2)$$

此外，應該要將濾波器的遮罩大小正規化。因此可以保證對任何的濾波器大小都能有一個常數的 Frobenius norm。經過測試  $9 \times 9$  的遮罩效果符合在  $\sigma = 1.2$  時的高斯微分，而  $27 \times 27$  則符合於  $\sigma = 3.6$  時。

### 2.1.1.2 找出區域極值

算出近似值之後的步驟就是要找出區域極值，參考圖 2.4，檢查中間 X 位置點周圍 8 點與相鄰取樣尺度空間的 18 個點，也就是綠色點的位置，如果 X 點是這些點中的極值，則將 X 選為關鍵點。

### 2.1.1.3 計算特徵點主要方向

由於特徵點對四周的描述向量會跟方向有關，所以首先要找到主要的方向，才能抗旋轉。為了能夠抗旋轉，將所有找到的特徵點確認一個重複性地朝向。要達成這個目的，首先計算在  $x$  跟  $y$  方向的哈爾小波響應(Haar-wavelet responses)，接著將這些特徵點以 6 倍的  $s$  畫圓，如圖 2.5 所示。 $s$  為這些特徵點被偵測到的尺度。

### 2.1.1.4 計算出 64 個特徵向量

這些圓經過哈爾小波響應可以得到一個方向，接著再利用這個方向切出 20 倍  $\sigma$  為邊長的正方形，如圖 2.6。將每個正方形再切割成  $4 \times 4$  個小方塊，為了簡化，我們將  $d_x$  稱為水平方向的哈爾小波響應， $d_y$  稱為垂直方向的哈爾小波響應。

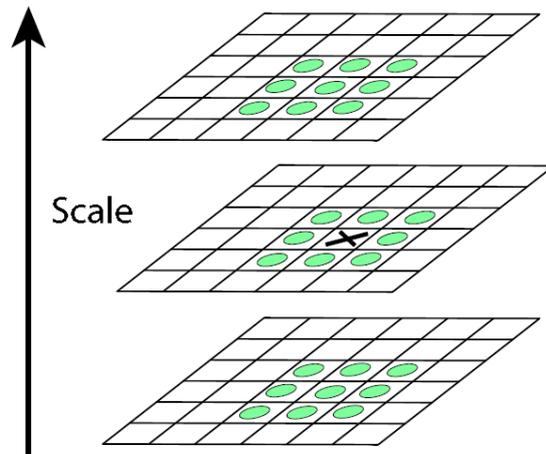


圖 2.4 區域極值計算[1]

每個子區域具有一個四維的描述向量  $v$  當作它基本強度的結構。將結果的水平、垂直變化量分別做累加與絕對後再累加， $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ ，如圖 2.7，便可得到在所有  $4 \times 4$  的子區域中維度是 64 的特徵向量。另一方面，切成  $3 \times 3$  方塊(SURF-36)的較短描述效果就稍差，不過他允許快速的比對，跟其他描述子比較之下在文獻中依然有相當的被接受度。

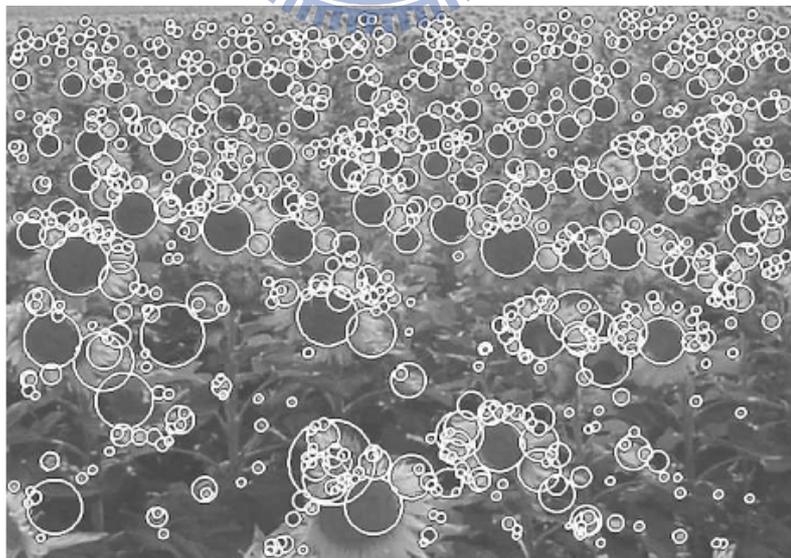


圖 2.5 這種場景清楚的顯示出從 Hessian-based 偵測器所得到特徵的性質[2]

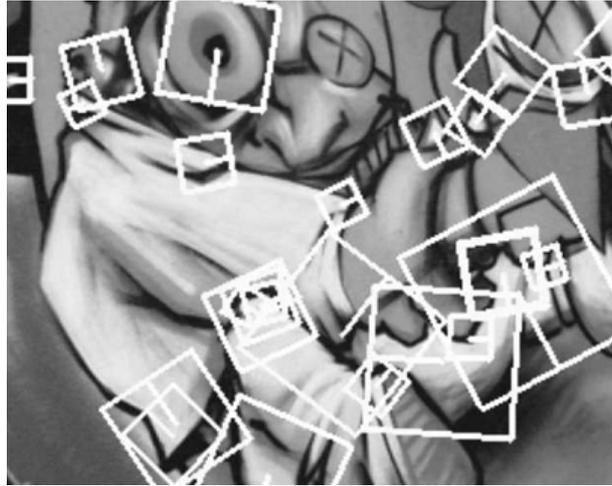


圖 2.6 一個塗鴉場景的細節顯示出在不同尺度下描述器視窗的大小[2]

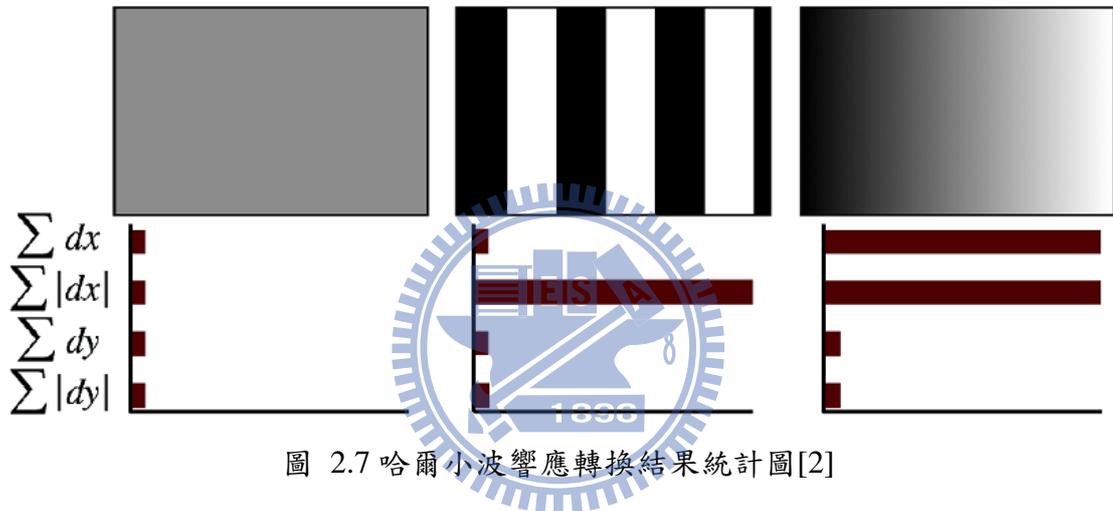


圖 2.7 哈爾小波響應轉換結果統計圖[2]

### 2.1.2. 影像特徵點比對

當計算出影像平面中每一個特徵點的特徵描述向量後，便可以透過與資料庫中所儲存的特徵點資訊進行比對，藉此得以判斷當前影像中是否存在機器人所需抓取的目標物。

本論文利用最近鄰居演算法 (Nearest neighborhood algorithm)，如 (2-3)式，其中  $Des_c(i)$  代表當前影像中該特徵描述向量的第  $i$  個元素， $Des_d(i)$  代表資料庫影像中該特徵描述向量的第  $i$  個元素。藉由比較當前影像中所有特徵點描述向量與資料庫中現存的特徵點描述向量，尋找其兩點距離最短的，則其本身則可能為相同的特徵點。

$$d = \left( \sum_{i=1}^{128} (Des_c(i) - Des_d(i))^2 \right)^{1/2} \quad (2-3)$$

雖然經過 SURF 處理後的特徵向量具有高度的差異，但其亦有可能因為來自於背景或者是雜訊恰巧相同，因此比對上建議至少有 3 點以上的特徵點相同，以增加比對的正確性。而且對於一般使用固定門檻值來限制是否為正確結果的效果將受限於所選定的門檻值，因此根據文獻[1]中所提的建議可以採取比較最近鄰居與次近鄰居的比值，當比值越接近 1 時，其越有可能是來自於背景或雜訊的干擾，因此可以視為錯誤的配對。

## 2.2 錯誤比對剔除

在做完特徵點比對之後，經常會出現一些錯誤比對的特徵點。由於成對的特徵點在之後 3D 座標計算上扮演著非常重要的角色，所以我們必須提高這些特徵點的品質，追求更加穩定跟準確的結果。本論文採用 RANSAC 跟 homography[19] 來做特徵點的二次篩選加強最後辨識的結果。

### 2.2.1. Homography

Homography 的功用在於找到兩個影像平面中，點跟點之間的對應關係。在影像平面上，一組對應的特徵點之間，存在著一種線性變換的關係，而對應矩陣 (Homography) 定義為其中一個影像平面上的點  $P_a$  轉換到另一個影像平面上的點  $P_b$  之間的線性轉換。Homography 是由一 3\*3 的非奇異矩陣 (Non-singular) 矩陣所決定，因為具有縮放因子  $\omega'$  的關係，能夠反映出目標物在影像平面上，跟資料庫影像相比，尺度大小的變化倍率。它具有 8 個自由度 (degree of freedom)，因為平面上一個點  $(x, y)$  具有兩個自由度，所以決定一個 Homography 至少需要四點以上的對應關係。

$$p'_b = H_{ab} p_a, \text{ 其中 } p_a, p_b \in P^2, \text{ } H \text{ 為 } 3*3 \text{ 的 Homography matrix}$$

$$H_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (2-4)$$

$$p'_b = H_{ab} p_a = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix} = \begin{bmatrix} \omega' x_b \\ \omega' y_b \\ \omega' \end{bmatrix} \quad (2-5)$$

$$x_b = \frac{h_{11}x_a + h_{12}y_a + h_{13}}{h_{31}x_a + h_{32}y_a + h_{33}}, \quad y_b = \frac{h_{21}x_a + h_{22}y_a + h_{23}}{h_{31}x_a + h_{32}y_a + h_{33}}, \quad \omega' = h_{31}x_a + h_{32}y_a + h_{33} \quad (2-6)$$

轉換矩陣  $H$  共有 9 個未知數，由於齊次座標轉換有比例相等的關係，於是假設轉換矩陣  $H$  中的某一個未知數固定(如:  $h_{33}=1$ )，如此一來待解的參數變成 8 個，而平面上一組對應點可提供 2 個線性獨立的方程式，因此 8 個未知數需要 4 組對應點提供 8 個線性獨立的方程式，而這 4 組對應點是從所有的對應點中任意選出的，所以要得到一個轉換矩陣至少要 4 組對應點。當求得轉換矩陣後，可算出每個對應點的誤差，因此符合最小誤差的對應矩陣極為我們所需要的對應矩陣。我們可以將 (2-6) 改寫成(2-7)。

當我們算出轉換矩陣之後，就可以經由(2-8)來驗證特徵點之間的轉換關係跟這個轉換矩陣的關係是否一致。將資料庫影像中的特徵點代入(2-8)中，可以得到資料庫中的特徵點對應在目前影像平面上的特徵點位置，這個步驟在 2.2.2 節 RANSAC 演算法的步驟中，是相當重要的一環，我們必須檢查這個對應關係的誤差來決定所算出來的轉換矩陣是否符合大部分的特徵對應關係。

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2 x_2 & -x'_2 y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2 x_2 & -y'_2 y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3 x_3 & -x'_3 y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3 x_3 & -y'_3 y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4 x_4 & -x'_4 y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4 x_4 & -y'_4 y_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix} \quad (2-7)$$

$$p_b = p'_b / \omega' = \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} \quad (2-8)$$

因為 homography 具有對應旋轉及尺度變化的特性，所以即使目標物在不同距離或是不同角度，我們都可以準確的框出目標物在目前影像平面上的位置。如圖 2.8 所示，左邊是資料庫影像，首先藉由所擷取到的特徵點，求得資料庫影像跟目前影像平面之間的轉換矩陣，接著透過轉換矩陣  $H$ ，我們可以得到目標物在目前影像平面(image plane)中的位置。

### 2.2.2. RANSAC

從 2.2.1 節中得知，要算出平面轉換矩陣，需要四個對應點，雖然從前面的特徵點擷取及最近鄰居演算法 (Nearest neighborhood algorithm) 運算過後，可以得到一群對應點，但是要算出更精確的轉換矩陣，就需要更準確的特徵對應點，所在這邊我們採用 RANSAC(Random sample consensus)[21]此種隨機取樣方式。RANSAC 演算法，其主要精神在於從一群分布不均勻或是參雜雜訊的資料中，隨機挑選數個樣本，算出參考的模型，之後再把其他資料逐次拿進來運算，留住符合模型的大多數，排除少數不符合者，以符合大眾資料的模型為最後結果。舉例來說[22]，如圖 2.9 所示，運用此演算法能夠從一群分布不均勻的點集合之中找出一條直線，符合大多數的點，而少數不符合的點則被剔除在外。

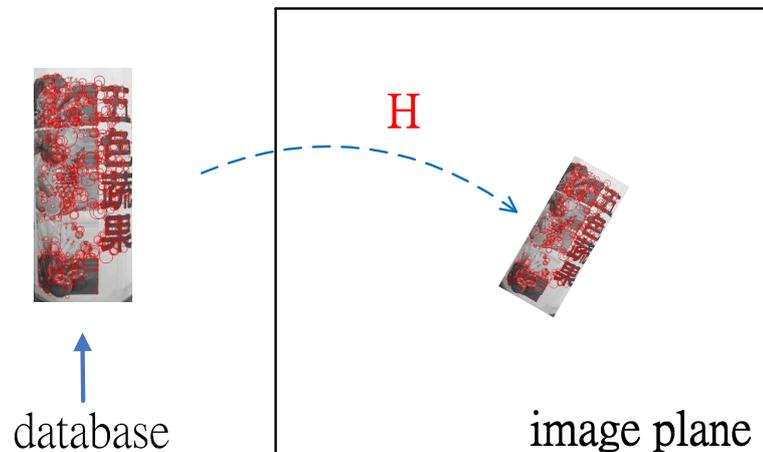


圖 2.8 透過轉換矩陣  $H$  得到資料庫物件在目前影像平面中之位置

RANSAC 演算法，為一個條件中止式的隨機演算法，當算出來的結果合乎我們所設定的閾值時，表示此結果已經合乎大部分的特徵對應關係了。

其演算步驟簡述如下：

1. 隨機挑選 4 對以上在第一階段比對成功的對應點，求出目前影像跟資料庫影像之間的平面轉換矩陣  $H$ 。
2. 將所有對應點使用步驟 1 求得的平面轉換矩陣進行位置轉換，將資料庫影像的對應點代入轉換矩陣得到另一組配對點，之後計算這些經過轉換的配對點座標與原本目前影像中的配對點座標之間的距離。
3. 統計步驟 2 中所有配對點間的距離，其配對點的距離若小於某個自訂的臨界距離  $D_c$ ，則表示此組配對點符合這個平面轉換矩陣，然後統計這個平面轉換矩陣的正確配對數量。

重複上述步驟  $N$  次之後，若某次的正確配對點數量最多，且大於某個自訂的臨界數量  $S_c$  時，則採用此次的平面轉換矩陣，再將這些配對點拿去求得最終的平面轉換矩陣。

在本論文中，參雜雜訊的資料樣本就是第一次特徵比對的結果，如圖 2.10 所示，可以看出當中存在在錯誤比對的特徵點，而模型就是轉換矩陣(homography matrix)，將 RANSAC 作為計算轉換矩陣的演算法，如此一來就可以得到對應正確比例最高的一群特徵點，用 homography 來框出在目前影像平面上的目標物時，位置也能更加準確，如圖 2.11，剔除錯誤比對特徵點之後，再將剩餘比對成功的特徵點拿來算轉換矩陣，就可以準確的框出目標物的位置，在分別對左右攝影機影像目標物的四個角點取平均後，就能得到中心點，在平面轉換矩陣計算的相當準確的情形下，目標物的範圍可以清楚的被框出來，所以用這個範圍的平均位置來當作參考點是比較穩定的結果。將這對中心點代入三維座標估測公式，如此一來就能夠得到穩定的控制參考點，將會對整個移動式抓取的控制上有很大的助益。

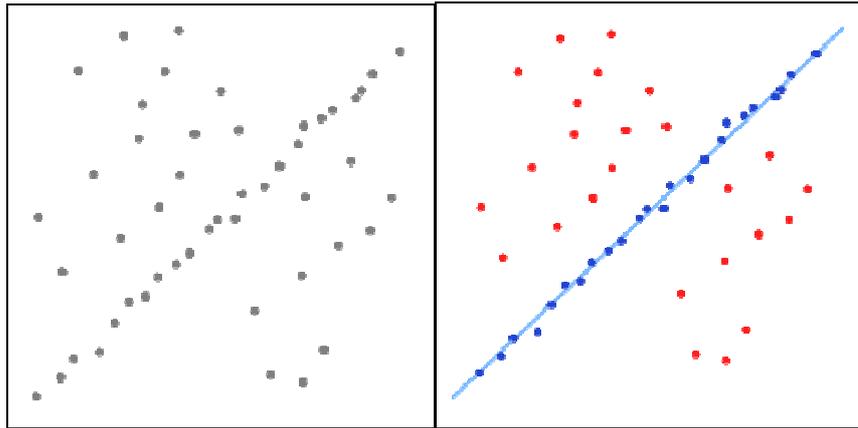


圖 2.9 左圖為一群分布不均勻的點集合，需要從這裡面決定出一條直線。右圖為經過 RANSAC 計算後，得到符合條件的直線，不符合的點則被剔除在外。[22]

實際的運算結果如圖 2.12 及圖 2.13 所示，我們將立體攝影機所讀取到的左右兩個影像，分別跟資料庫影像做前述的處理，框出目標物後，分別求得目標物在左右攝影機影像的中心點，接著將這對中心點代入第三章將會提到的三維座標估測公式，就可以知道目標物在世界座標中的位置，做為導引機器人往前抓取目標物的控制命令輸入。

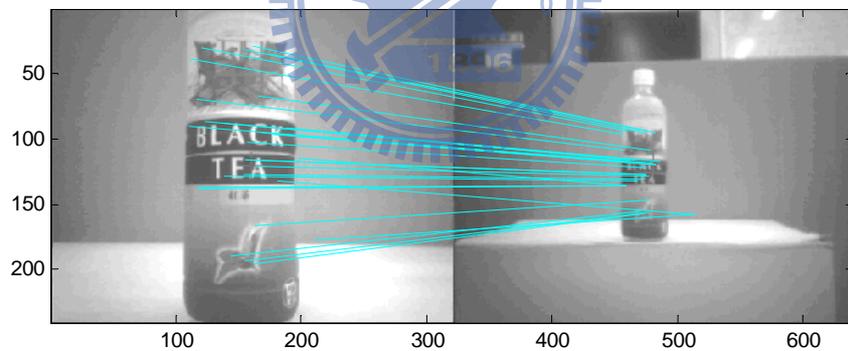


圖 2.10 用 SURF 做特徵比對的結果

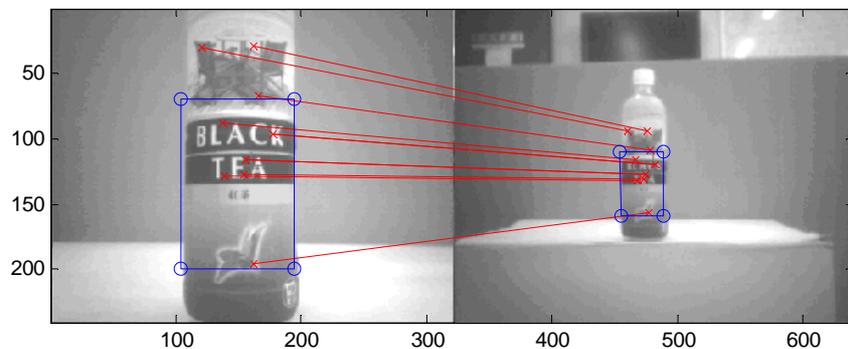


圖 2.11 用 RANSAC 做 homography estimation 的結果

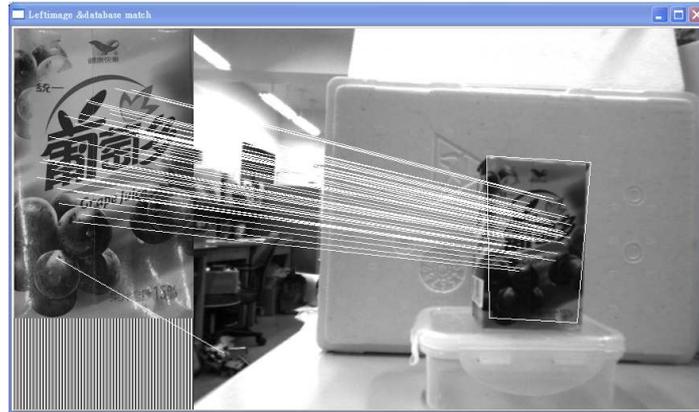


圖 2.12 實際左影像跟資料庫影像做轉換矩陣計算的結果

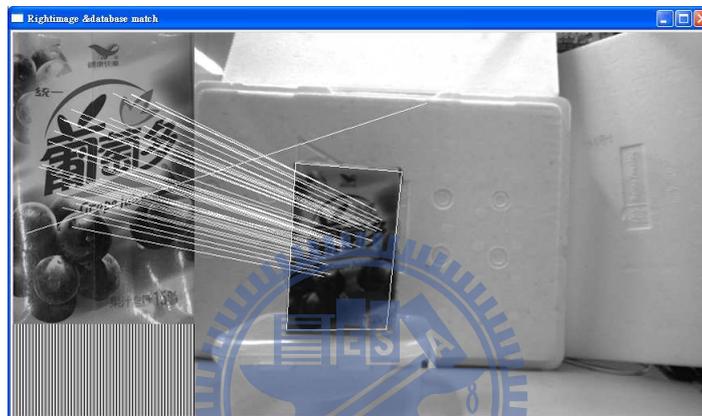


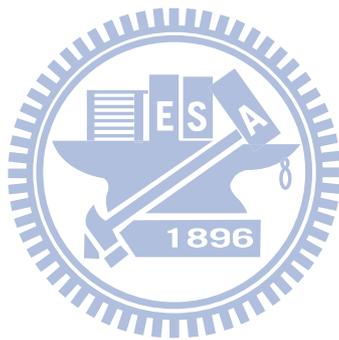
圖 2.13 實際右影像跟資料庫影像做轉換矩陣計算的結果

### 2.3 定義中心點

當我們算出 homography 之後，我們可以得到目標物跟目前影像之間的平面轉換關係。接著我們可以算出目標物在左右影像平面上，四個角點的位置，將這四個角點連接起來框出目標物在目前影像平面的位置。如圖 2.14 所示，我們將目標物在左攝影機影像平面的四個角點取平均，定為左中心點(left center)，目標物在右攝影機影像平面的四個角點取平均，定為右中心點(right center)，將這組中心點當作控制用的參考點，以這點的三維座標表示目標物的位置，藉此導引機器人完成抓取任務。



圖 2.14 目標物在影像平面的區域及中心點位置的定義



### 第三章目標物姿態估測

本章節的主要目的在說明如何將第二章所算出的特徵對應點，轉換成在空間中實際的三維座標，當作控制器輸入的重要資訊來源。

#### 3.1 立體視覺原理

要如何從空間中一組成對的影像中找出互相匹配的特徵配對，也就是在世界座標中的某個點分別在左右眼影像中成像的位置是立體視覺的形成很重要的一個關鍵點。這組對應點稱為(Corresponding points)，而對應點在影像中的位置差稱為視差(Disparity)。圖 3.1 中，世界座標中的一點  $w = (x, y, z)$  分別投影到左影像平面中的  $p_l(u_l, v_l)$  以及右影像平面中的  $p_r(u_r, v_r)$ ，兩個攝影機之間的鏡心距離設為  $B$ (baseline)，透鏡的焦距為  $\lambda$ ，兩個成像點的橫軸誤差為  $d = u_l - u_r$ ，最後可由這組對應點求回世界座標[23]。

圖 3.2 為圖 3.1 投影到  $XZ$  平面的示意圖，由其中的幾何關係可以得到，如何從對應點  $p_l(u_l, v_l)$  和  $p_r(u_r, v_r)$  推導回世界座標  $w = (x, y, z)$  的公式。

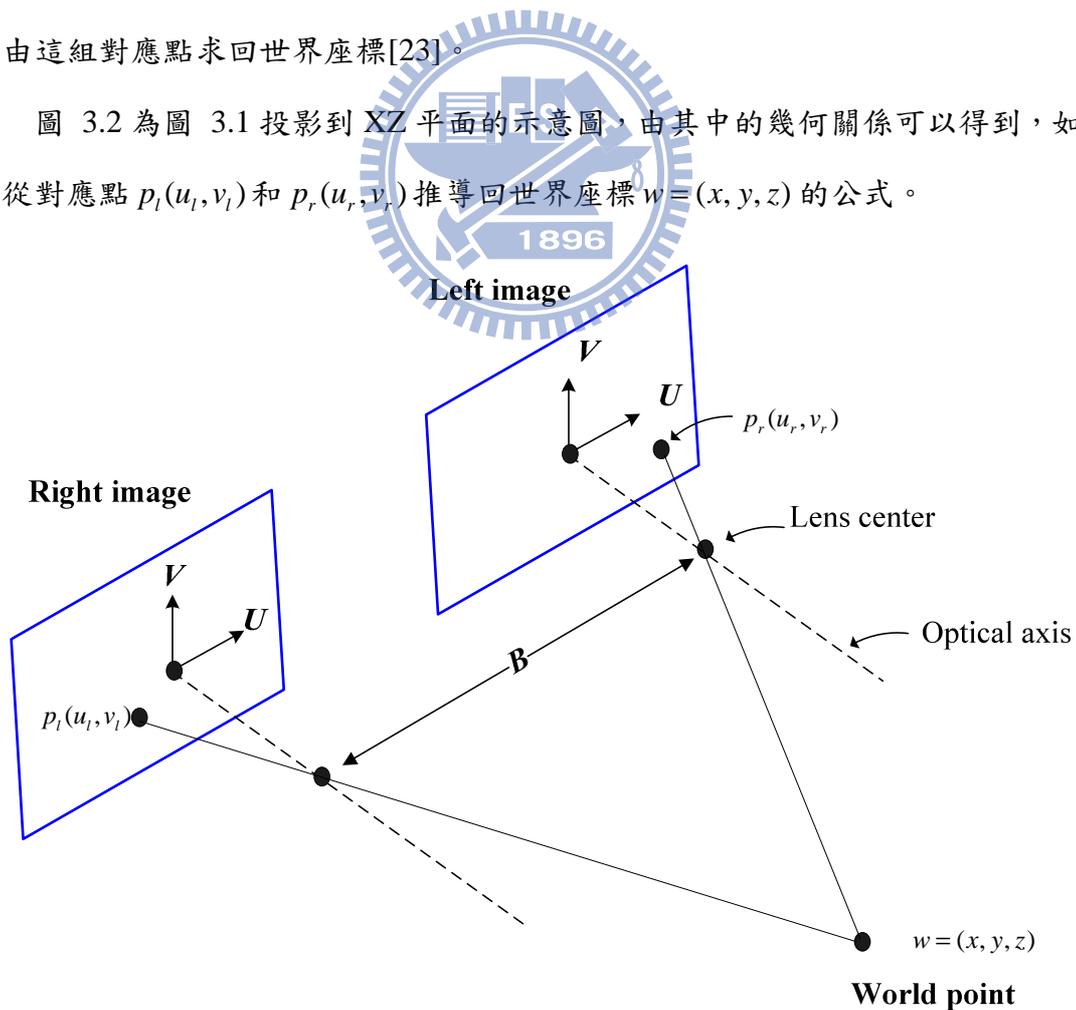


圖 3.1 立體影像處理的模型[23]

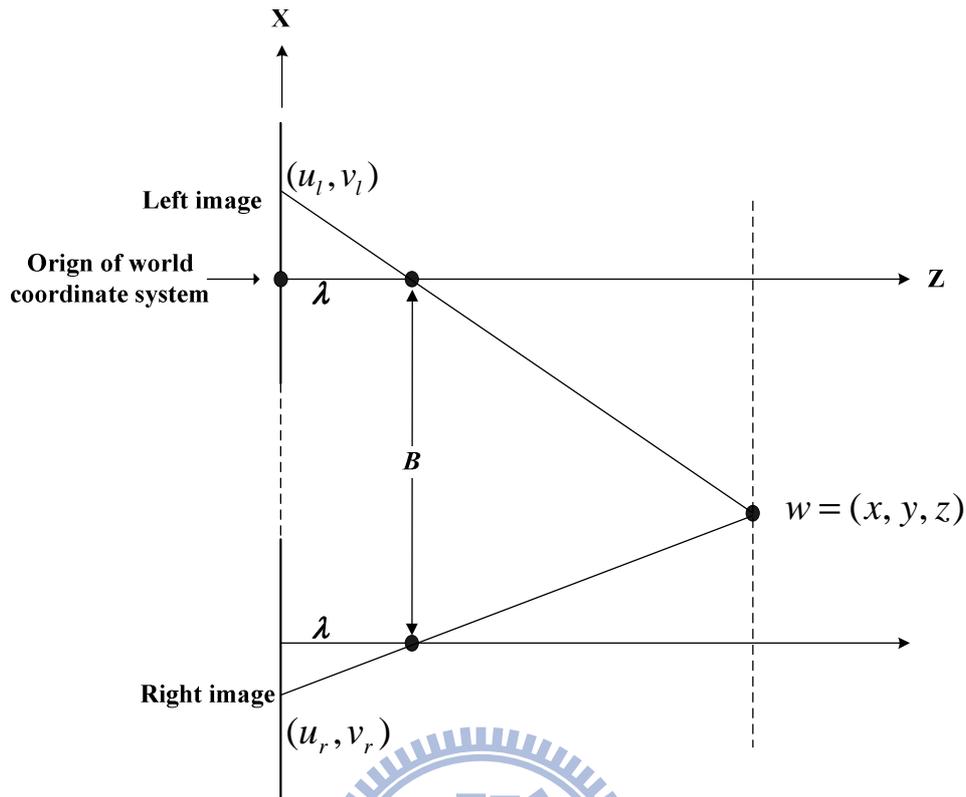


圖 3.2 攝影機跟世界座標關係的上視圖[23]

首先由圖 3.2 中相似三角形的關係得到

$$\frac{u}{\lambda} = \frac{x}{z-\lambda} = \frac{x}{\lambda-z} \quad (3-1)$$

$$\frac{v}{\lambda} = -\frac{y}{z-\lambda} = \frac{y}{\lambda-z} \quad (3-2)$$

接著整理過後得到

$$u = \frac{\lambda x}{\lambda - z} \quad (3-3)$$

$$v = \frac{\lambda y}{\lambda - z} \quad (3-4)$$

將(3-3)及(3-4)的 x 及 y 整理到左項，得到 XY 座標的公式

$$x = \frac{u}{\lambda}(\lambda - z) \quad (3-5)$$

$$y = \frac{v}{\lambda}(\lambda - z) \quad (3-6)$$

將影像平面上的 U 座標數值，代入(3-5)，得到

$$x_1 = \frac{u_1}{\lambda}(\lambda - z_1) \quad (3-7)$$

$$x_2 = \frac{u_2}{\lambda}(\lambda - z_2) \quad (3-8)$$

$x_1$ 、 $x_2$ 之間的座標差為兩攝影機間鏡心距離  $B$  (Baseline)

$$x_2 = x_1 + B \quad (3-9)$$

左右攝影機看到的  $Z$  座標值皆相同

$$z_2 = z_1 = z \quad (3-10)$$

將(3-9)代入(3-8)得到

$$x_1 + B = \frac{u_2}{\lambda}(\lambda - z_2) \quad (3-11)$$

整理之後，就得到  $Z$  座標的公式

$$z = \lambda - \frac{\lambda * B}{u_l - u_r} \quad (3-12)$$

### 3.2 3D 座標公式校正

雖然可以從 3.1 節中所述的公式求出一組對應點的世界座標，但是從實際計算的結果來看，仍存在著不小的誤差，假設結果跟實際座標值存在著一個轉換關係，於是我們需要求出這個轉換關係，來對 3D 座標公式做出修正，以期待能夠影像平面上的座標值，得到目標物在空間中實際的三維座標。

為了得到公式值跟實際座標值之間的轉換關係，我們需要一群參考的資料。在本論文中，採用的方法是針對世界座標中固定的一點跟攝影機的相對距離越來越遠時，紀錄其投影到左右攝影機影像平面上的座標值，將這些座標值代入公式中，找出不同距離時，公式值跟實際值之間的關係。求出這個轉換關係後，我們就可以由空間中任何一點的在左右攝影機影像平面上的二維座標位置，求出其在空間中的三維座標。

在公式的校正方面，用的是每個方格 28mm\*28mm 的校正板，如圖 3.3 所示。校正板每次往後退一個方格的距離( $Z$  方向)，針對校正板上的同一個點做紀錄(在本次校正中是針對  $C(0,27,Z)$  這個點)，如圖 3.4 所示，把  $C$  在左右眼影像平面的座標記錄下來，如表 3.1，直到某個距離  $Z$  時， $p_l$ (左視窗中的目標點)及  $p_r$ (右

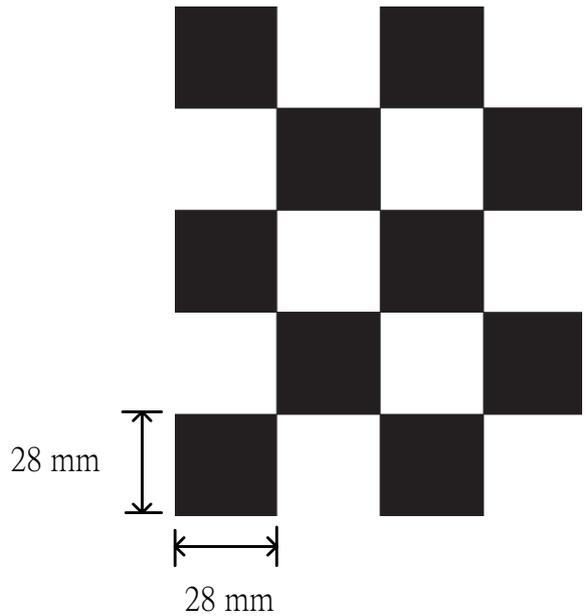


圖 3.3 校正版圖示

視窗中的目標點)的座標值跟上一個距離相比幾乎沒有變動時，表示到達攝影機辨識物體的極限距離。把這過程中記錄下來的所有座標對，用來求出二維座標到三維座標之間的轉換公式。

### 3.2.1. Z 軸座標值估測

由於 X、Y、Z 都需要 Z(深度)的估測值來做計算，所以一開始我們先將上面得到的所有對應點代入(3-12)得到深度的估測值( $\tilde{Z}$ )。從圖 3.5 及表 3.2 可以發現， $\tilde{Z}$  跟實際距離( $Z_a$ )的關係呈現出一個二元一次直線的關係，因此，只要找出兩者之間的關係就能從  $\tilde{Z}$  得到實際的 Z 座標。

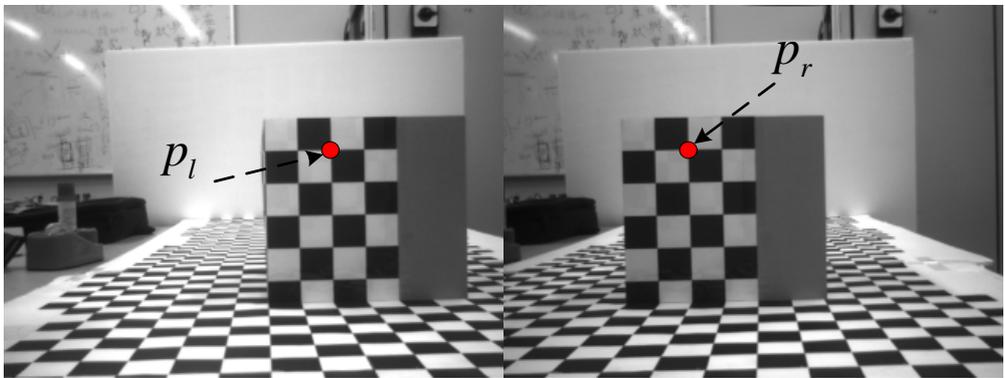


圖 3.4 距離為 19 格時(532mm)左圖為左攝影機之影像，右圖為右攝影機之影像

表 3.1 不同距離時目標點在影像平面上的座標值

Z <sub>a</sub> (mm)	center_lx	center_ly	center_rx	center_ry	Z <sub>a</sub> (mm)	center_lx	center_ly	center_rx	center_ry
280.000	493	150	157	150	868.000	389	208	277	208
308.000	479	158	173	158	896.000	387	209	279	209
336.000	466	164	185	164	924.000	385	210	281	210
364.000	457	170	198	170	952.000	383	211	282	211
392.000	449	175	207	175	980.000	382	212	283	212
420.000	443	179	217	179	1008.000	380	212	284	212
448.000	435	183	223	183	1036.000	379	213	286	213
476.000	430	186	230	186	1064.000	378	213	287	213
504.000	425	190	236	190	1092.000	377	213	288	213
532.000	421	193	241	193	1120.000	376	214	289	214
560.000	416	195	245	195	1148.000	375	214	290	214
588.000	412	197	249	197	1176.000	374	215	291	215
616.000	409	199	253	199	1204.000	373	215	292	215
644.000	406	201	257	201	1232.000	372	216	294	216
672.000	404	202	261	202	1260.000	371	216	295	216
700.000	402	203	264	203	1288.000	370	216	296	216
728.000	399	203	267	203	1316.000	370	217	297	217
756.000	397	204	270	204	1344.000	369	217	297	217
784.000	395	205	272	205	1372.000	369	218	298	218
812.000	393	206	274	206	1400.000	368	218	299	218
840.000	390	207	276	207					

為了得到這個轉換關係，首先要求出一條逼近 $\tilde{Z}$ 的直線，於是我們將 $Z_a$ 跟 $\tilde{Z}$ 之間的對應數值，運用最小迴歸曲線的原理，就能夠得到逼近這條直線的方程式。

(3-13)是由 MATLAB 所計算出來的就是 $\tilde{Z}$ 跟 $Z$ 座標之間的轉換公式。

$$pose\_Z = 802.6 - 1.9701 * \tilde{Z} \quad (3-13)$$

根據圖 3.6 可看出，校正後的 $Z$ 座標值在 1 公尺的範圍內誤差都在正負 10mm 之間，但是隨著距離變遠，對應點的差距越來越不明顯而導致誤差略為上升，大概在 1.5 公尺之後，目標點在左右影像平面上的位置幾乎不再變動，沒辦法再用視差的方法來算深度。如果直接使用文獻中[23]的公式來算深度會隨著距

離增加，誤差也跟著增加，但是結合公式值及本論文所提出的轉換關係尋找方法之後，Z 座標的估測精確度跟未修正前之結果相比，可以說是具有相當顯著的提升。

表 3.2 Z 座標未校正前的估測值

$Z_a$ (mm)	$\tilde{Z}$	$Z_a$ (mm)	$\tilde{Z}$
280.000	265.119	868.000	-29.458
308.000	250.679	896.000	-45.823
336.000	236.290	924.000	-63.447
364.000	221.331	952.000	-77.582
392.000	207.908	980.000	-87.480
420.000	193.430	1008.000	-103.102
448.000	178.969	1036.000	-119.731
476.000	164.963	1064.000	-131.427
504.000	150.561	1092.000	-143.648
532.000	137.469	1120.000	-156.430
560.000	122.999	1148.000	-169.815
588.000	108.795	1176.000	-183.844
616.000	95.171	1204.000	-198.567
644.000	80.267	1232.000	-222.066
672.000	66.331	1260.000	-238.762
700.000	53.792	1288.000	-256.361
728.000	37.492	1316.000	-265.523
756.000	22.731	1344.000	-274.938
784.000	10.059	1372.000	-284.619
812.000	-3.466	1400.000	-304.823
840.000	-21.706		

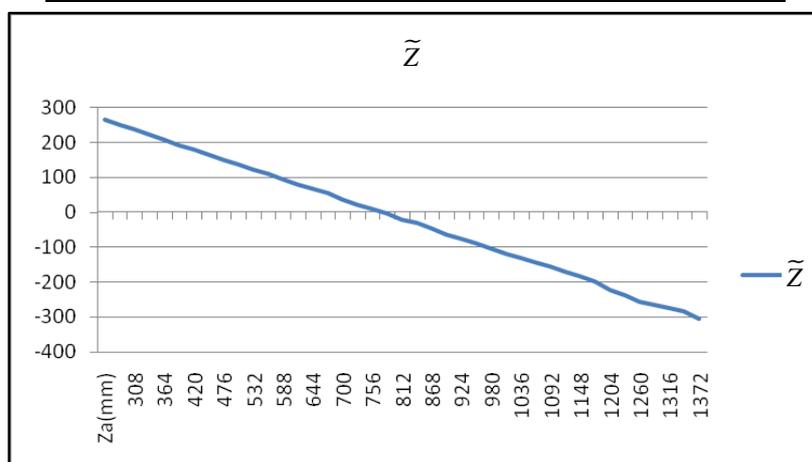


圖 3.5 Z 座標估測值

表 3.3 Z 座標校正後的結果及誤差值

$Z_a$ (mm)	pose_Z	pose_Z_error	$Z_a$ (mm)	pose_Z	pose_Z_error
280.000	280.289	0.289	868.000	860.635	-7.365
308.000	308.737	0.737	896.000	892.876	-3.124
336.000	337.084	1.084	924.000	927.598	3.598
364.000	366.556	2.556	952.000	955.444	3.444
392.000	393.001	1.001	980.000	974.945	-5.055
420.000	421.523	1.523	1008.000	1005.721	-2.279
448.000	450.013	2.013	1036.000	1038.482	2.482
476.000	477.606	1.606	1064.000	1061.524	-2.476
504.000	505.979	1.979	1092.000	1085.600	-6.400
532.000	531.772	-0.228	1120.000	1110.784	-9.216
560.000	560.280	0.280	1148.000	1137.152	-10.848
588.000	588.264	0.264	1176.000	1164.792	-11.208
616.000	615.104	-0.896	1204.000	1193.796	-10.204
644.000	644.466	0.466	1232.000	1240.091	8.091
672.000	671.921	-0.079	1260.000	1272.986	12.986
700.000	696.624	-3.376	1288.000	1307.658	19.658
728.000	728.738	0.738	1316.000	1325.706	9.706
756.000	757.817	1.817	1344.000	1344.256	0.256
784.000	782.783	-1.217	1372.000	1363.329	-8.671
812.000	809.428	-2.572	1400.000	1403.132	3.132
840.000	845.362	5.362			

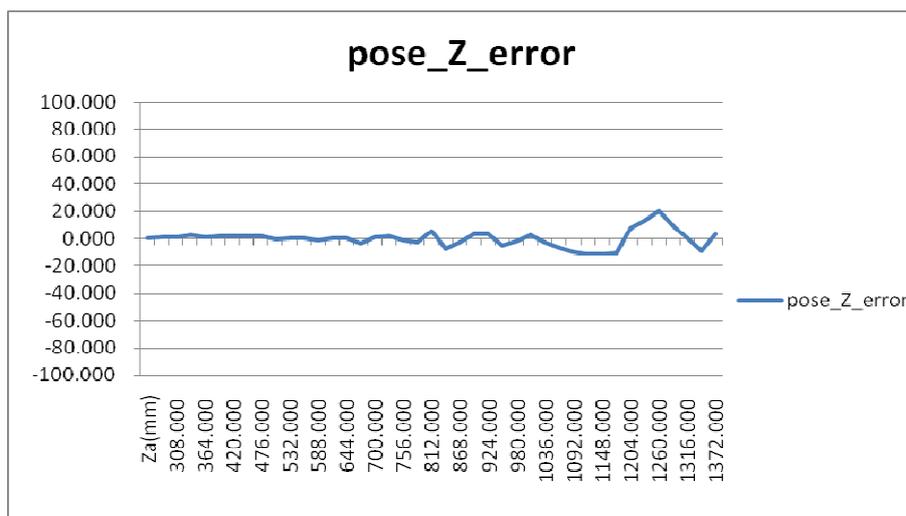


圖 3.6 校正後 Z 座標誤差值

### 3.2.2. X 軸座標值估測

在 X 座標方面，一樣要先把對應點代入(3-5)式中，求出各個距離的  $\tilde{X}$ ，在本論文中，攝影機座標系的原點是訂在左右攝影機的中間，加上在本次校正實驗中是針對 X 座標等於 0 時，所以對於左攝影機來說，實際的 X 座標值  $Xl_a$  是 60，對於右攝影機，實際的 X 座標值  $Xr_a$  則是 -60，由此可得到左右攝影機的估測值跟實際 X 座標值之間的誤差。

從左攝影機跟右攝影機的  $\tilde{X}$  值統計圖表可以看出來，隨著距離增加，X 值的誤差逐漸增加，但是從圖 3.7 及圖 3.8 可以看出來誤差大致上是呈線性的增加，所以在這邊我們需要找出來，隨著距離的增加我們所要減掉的估測誤差，也就是誤差方程式。

$\tilde{X} = \frac{x_0}{\lambda}(\lambda - Z)$  為估測出來的 X 座標，將 pose\_Z 跟算出來的 X 座標誤差  $x\_error$  的所有數值，算出誤差方程式，(3-14)及(3-15)是由 MATLAB 所求出來的誤差方程式。

$$\overline{x_l\_error} = 1.1557 + 0.41236 * pose\_Z \quad (3-14)$$

$$\overline{x_r\_error} = 4.3684 + 0.40812 * pose\_Z \quad (3-15)$$

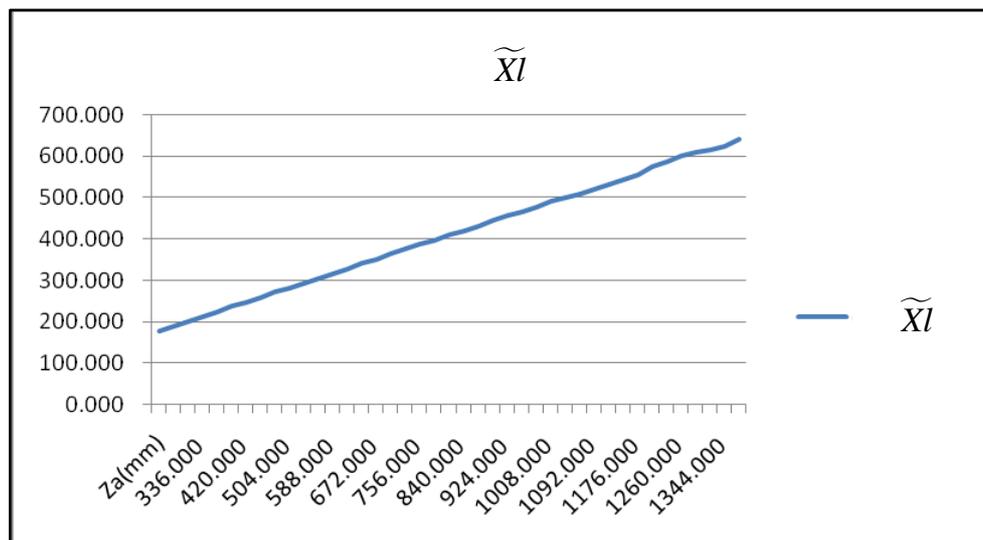


圖 3.7 Xl 座標估測值

表 3.4 XI 座標未校正前的估測值

Z <sub>a</sub> (mm)	$\widetilde{Xl}$	Xl <sub>a</sub>	Xl_error	Z <sub>a</sub> (mm)	$\widetilde{Xl}$	Xl <sub>a</sub>	Xl_error
280.000	176.071	60	116.071	868.000	416.786	60	356.786
308.000	187.843	60	127.843	896.000	430.000	60	370.000
336.000	199.004	60	139.004	924.000	444.231	60	384.231
364.000	211.737	60	151.737	952.000	455.050	60	395.050
392.000	222.645	60	162.645	980.000	463.030	60	403.030
420.000	235.221	60	175.221	1008.000	475.000	60	415.000
448.000	246.226	60	186.226	1036.000	489.032	60	429.032
476.000	258.000	60	198.000	1064.000	498.462	60	438.462
504.000	269.841	60	209.841	1092.000	508.315	60	448.315
532.000	280.667	60	220.667	1120.000	518.621	60	458.621
560.000	291.930	60	231.930	1148.000	529.412	60	469.412
588.000	303.313	60	243.313	1176.000	540.723	60	480.723
616.000	314.615	60	254.615	1204.000	552.593	60	492.593
644.000	326.980	60	266.980	1232.000	572.308	60	512.308
672.000	339.021	60	279.021	1260.000	585.789	60	525.789
700.000	349.565	60	289.565	1288.000	600.000	60	540.000
728.000	362.727	60	302.727	1316.000	608.219	60	548.219
756.000	375.118	60	315.118	1344.000	615.000	60	555.000
784.000	385.366	60	325.366	1372.000	623.662	60	563.662
812.000	396.303	60	336.303	1400.000	640.000	60	580.000
840.000	410.526	60	350.526				

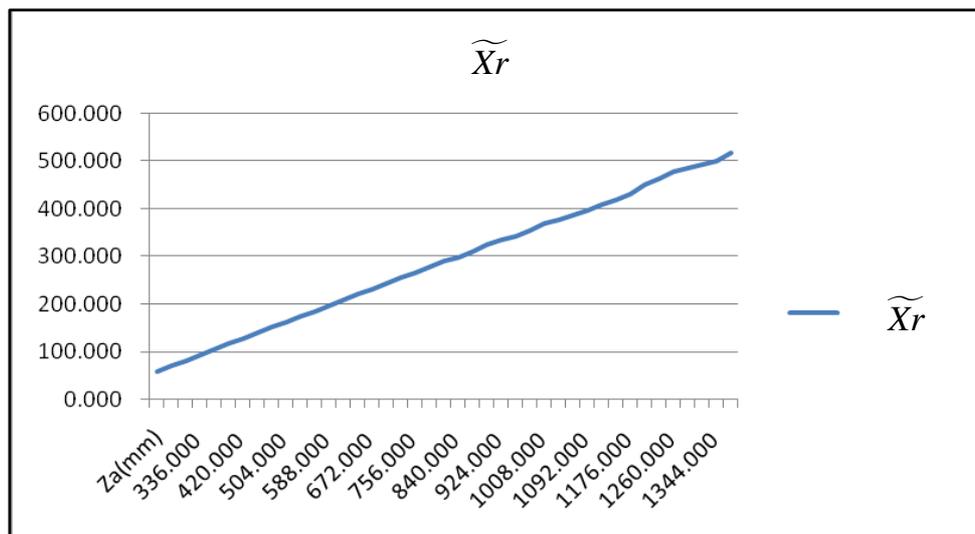


圖 3.8 Xr 座標估測值

表 3.5 Xr 座標未校正前的估測值

Z <sub>a</sub> (mm)	$\widetilde{X}_r$	X <sub>r<sub>a</sub></sub>	Xr_Slope	Z <sub>a</sub> (mm)	$\widetilde{X}_r$	X <sub>r<sub>a</sub></sub>	Xr_error
280.000	57.441	-60	0.422	868.000	296.517	-60	356.517
308.000	69.271	-60	0.399	896.000	309.579	-60	369.579
336.000	80.442	-60	0.455	924.000	323.644	-60	383.644
364.000	93.180	-60	0.389	952.000	334.329	-60	394.329
392.000	104.061	-60	0.448	980.000	342.215	-60	402.215
420.000	116.603	-60	0.391	1008.000	354.036	-60	414.036
448.000	127.540	-60	0.418	1036.000	367.905	-60	427.905
476.000	139.249	-60	0.420	1064.000	377.220	-60	437.220
504.000	151.011	-60	0.384	1092.000	386.953	-60	446.953
532.000	161.757	-60	0.399	1120.000	397.133	-60	457.133
560.000	172.922	-60	0.403	1148.000	407.791	-60	467.791
588.000	184.205	-60	0.400	1176.000	418.962	-60	478.962
616.000	195.408	-60	0.438	1204.000	430.684	-60	490.684
644.000	207.659	-60	0.426	1232.000	450.159	-60	510.159
672.000	219.591	-60	0.373	1260.000	463.471	-60	523.471
700.000	230.033	-60	0.465	1288.000	477.502	-60	537.502
728.000	243.057	-60	0.438	1316.000	485.623	-60	545.623
756.000	255.320	-60	0.362	1344.000	492.312	-60	552.312
784.000	265.456	-60	0.386	1372.000	500.870	-60	560.870
812.000	276.271	-60	0.502	1400.000	517.000	-60	577.000
840.000	290.329	-60	0.221				

將估測值減去對應的誤差值，就可以得到左右攝影機對應的 X 座標值。

$$pose\_x = est\_x - x\_error \quad (3-16)$$

(3-16)式所代表的意義是，隨著距離的不同，在 x 座標產生的誤差量也不同，由誤差方程式我們可以算出在不同距離時產生的誤差量，得到此誤差量後，將當時 x 座標的估測值減掉誤差就能得到正確的 x 座標。

分別將左影像跟右影像的結果代入得到 pose\_xl 跟 pose\_xr，將兩個值相加取平均，就能得到目標點的 X 座標值

$$pose\_X = (pose\_xl + pose\_xr) / 2 \quad (3-17)$$

表 3.6 X 座標校正後的結果及誤差值

$Z_a$ (mm)	pose_X	pose_X_error	$Z_a$ (mm)	pose_X	pose_X_error
280.000	-0.991	-0.991	868.000	0.822	0.822
308.000	-0.861	-0.861	896.000	0.734	0.734
336.000	-1.324	-1.324	924.000	0.637	0.637
364.000	-0.679	-0.679	952.000	-0.034	-0.034
392.000	-0.634	-0.634	980.000	-0.101	-0.101
420.000	0.224	0.224	1008.000	-0.831	-0.831
448.000	-0.492	-0.492	1036.000	-0.320	-0.320
476.000	-0.071	-0.071	1064.000	-0.401	-0.401
504.000	0.091	0.091	1092.000	-0.485	-0.485
532.000	0.296	0.296	1120.000	-0.573	-0.573
560.000	-0.186	-0.186	1148.000	-0.666	-0.666
588.000	-0.333	-0.333	1176.000	-0.764	-0.764
616.000	-0.091	-0.091	1204.000	-0.867	-0.867
644.000	0.172	0.172	1232.000	-0.264	-0.264
672.000	0.895	0.895	1260.000	-0.362	-0.362
700.000	1.254	1.254	1288.000	-0.465	-0.465
728.000	1.172	1.172	1316.000	0.302	0.302
756.000	1.570	1.570	1344.000	-0.574	-0.574
784.000	1.520	1.520	1372.000	0.212	0.212
812.000	1.465	1.465	1400.000	0.117	0.117
840.000	0.864	0.864			

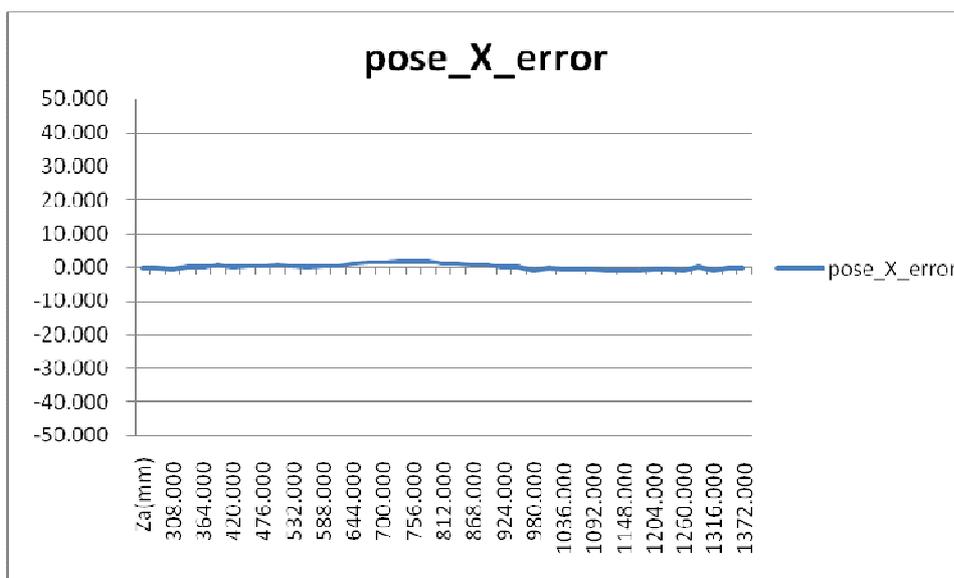


圖 3.9 校正後 X 座標誤差值

我們希望在影像伺服的過程中所觀測到目標物的 X 座標應該是一致的，不能隨著距離的增加而有所變動，否則將會導致機器人在移動的過程中，無法朝著目標物前進，而出現左右偏移的情況。由圖 3.9 可看出，X 座標的估測公式經過校正後，可以讓誤差縮減到將近 5mm 之內，大大增加 X 座標的精確度。

### 3.2.3. Y 軸座標值估測

在 Y 座標方面，一樣將對對應點代入(3-6)式中，求出各距離時的  $\tilde{Y}$ ，在本實驗中校正給定的 Y 座標  $Y_a$  是 27mm，而因為攝影機的配置是左右眼平行，所以在攝影機平行於地面的情況下，量測出來的 Y 座標應是相同的值。

表 3.7 Y1 座標未校正前的估測值

$Z_a(\text{mm})$	$\tilde{Y}$	$Y_a$	Y1_error	$Z_a(\text{mm})$	$\tilde{Y}$	$Y_a$	Y1_error
280.000	54.777	27	27.777	868.000	222.671	27	195.671
308.000	63.161	27	36.161	896.000	231.932	27	204.932
336.000	71.210	27	44.210	924.000	241.904	27	214.904
364.000	79.905	27	52.905	952.000	250.197	27	223.197
392.000	87.879	27	60.879	980.000	256.408	27	229.408
420.000	96.093	27	69.093	1008.000	264.338	27	237.338
448.000	104.577	27	77.577	1036.000	274.066	27	247.066
476.000	112.530	27	85.530	1064.000	280.031	27	253.031
504.000	121.502	27	94.502	1092.000	286.264	27	259.264
532.000	129.471	27	102.471	1120.000	294.158	27	267.158
560.000	137.569	27	110.569	1148.000	301.016	27	274.016
588.000	145.680	27	118.680	1176.000	309.646	27	282.646
616.000	153.651	27	126.651	1204.000	317.225	27	290.225
644.000	162.368	27	135.368	1232.000	330.854	27	303.854
672.000	169.917	27	142.917	1260.000	339.490	27	312.490
700.000	176.853	27	149.853	1288.000	348.592	27	321.592
728.000	184.776	27	157.776	1316.000	354.966	27	327.966
756.000	192.896	27	165.896	1344.000	359.859	27	332.859
784.000	200.062	27	173.062	1372.000	366.570	27	339.570
812.000	207.709	27	180.709	1400.000	377.117	27	350.117
840.000	217.759	27	190.759				

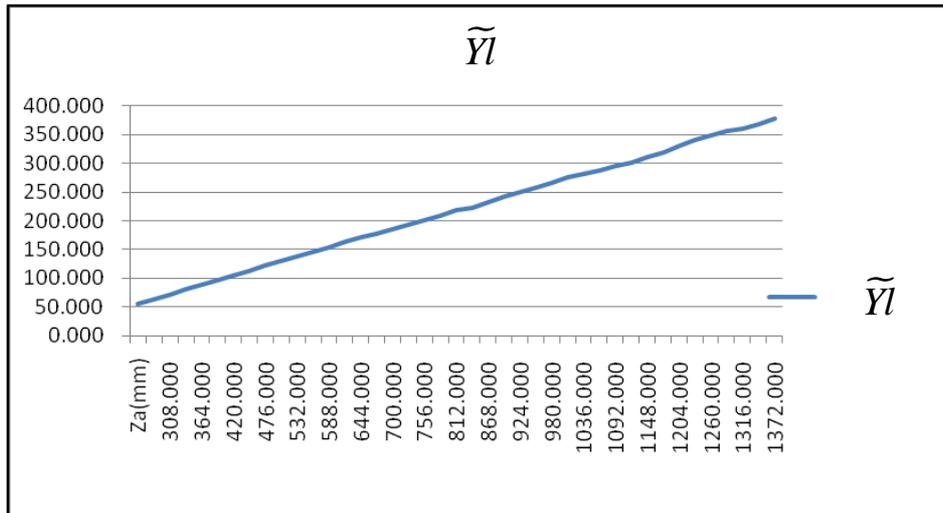


圖 3.10 Yl 座標估測值

表 3.8 Yr 座標未校正前的估測值

$Z_a$ (mm)	$\bar{Y}_r$	$Y_r$	Yr_error	$Z_a$ (mm)	$\bar{Y}_r$	$Y_r$	Yr_error
280.000	55.151	27	28.151	868.000	222.614	27	195.614
308.000	63.534	27	36.534	896.000	231.842	27	204.842
336.000	71.574	27	44.574	924.000	241.779	27	214.779
364.000	80.259	27	53.259	952.000	250.043	27	223.043
392.000	88.222	27	61.222	980.000	256.233	27	229.233
420.000	96.419	27	69.419	1008.000	264.132	27	237.132
448.000	104.885	27	77.885	1036.000	273.826	27	246.826
476.000	112.818	27	85.818	1064.000	279.767	27	252.767
504.000	121.771	27	94.771	1092.000	285.976	27	258.976
532.000	129.720	27	102.720	1120.000	293.843	27	266.843
560.000	137.795	27	110.795	1148.000	300.675	27	273.675
588.000	145.882	27	118.882	1176.000	309.274	27	282.274
616.000	153.829	27	126.829	1204.000	316.823	27	289.823
644.000	162.520	27	135.520	1232.000	330.403	27	303.403
672.000	170.043	27	143.043	1260.000	339.005	27	312.005
700.000	176.955	27	149.955	1288.000	348.071	27	321.071
728.000	184.848	27	157.848	1316.000	354.424	27	327.424
756.000	192.940	27	165.940	1344.000	359.297	27	332.297
784.000	200.082	27	173.082	1372.000	365.987	27	338.987
812.000	207.703	27	180.703	1400.000	376.492	27	349.492
840.000	217.716	27	190.716				

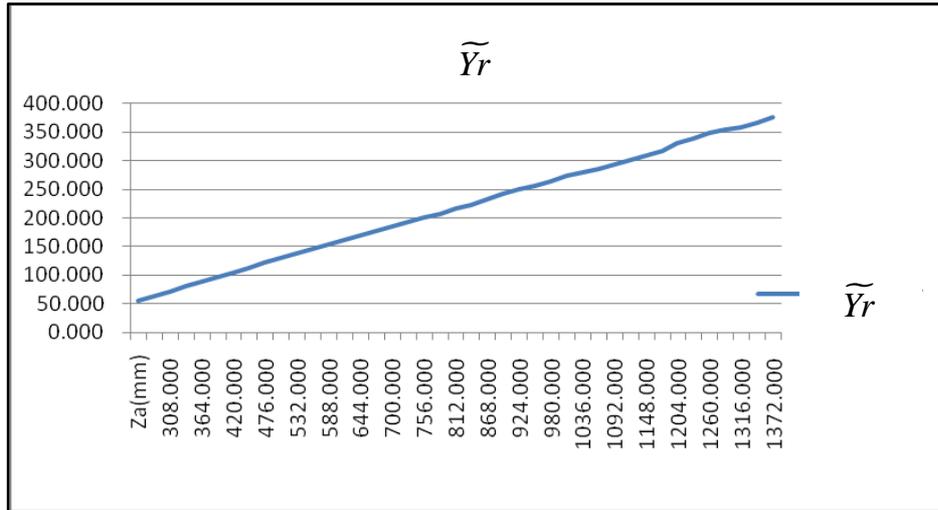


圖 3.11 Yr 座標估測值

從  $\tilde{Y}_r$  的統計圖表可看出，大致上跟距離成正比，誤差隨距離逐漸增加，所以我們要求其誤差方程式，針對不同距離減去對應的誤差值，得到正確的 Y 座標。

$\tilde{Y} = \frac{y_0}{\lambda}(\lambda - Z)$  為估測出來的 Y 座標，接著算出 pose\_Z 跟 Y 座標誤差 y\_error 之間的誤差方程式，(3-18)及(3-19)是透過 MATLAB 所求出來的誤差方程式。

$$\overline{yl\_error} = -50.722 + 0.28601 * \overline{pose\_Z} \quad (3-18)$$

$$\overline{yr\_error} = -49.993 + 0.28507 * \overline{pose\_Z} \quad (3-19)$$

將估測值減去對應的誤差值，就可以得到左右攝影機對應的 Y 座標值。

$$\overline{pose\_y} = \overline{est\_y} - \overline{y\_error} \quad (3-20)$$

(3-20)式所代表的意義是，隨著距離的不同，在 y 座標產生的誤差量也不同，由誤差方程式我們可以算出在不同距離時產生的誤差量，得到此誤差量後，將當時 y 座標的估測值減掉誤差就能得到正確的 y 座標。

分別將左影像跟右影像的結果代入得到 pose\_y1 跟 pose\_yr，將兩個值相加取平均，就能得到目標點的 Y 座標值

$$\overline{pose\_Y} = (\overline{pose\_y1} + \overline{pose\_yr}) / 2 \quad (3-21)$$

Y 座標在我們所設計的影像伺服架構中的角色是算出目標物在空間中的高

度是在哪個位置，也就是手臂在抓取時所要移動到的垂直位置，所以 Y 座標必須要準確，機器人才會確實的抓到目標物。

由校正後的 Y 座標誤差統計圖表可看出，經過以上求出來的誤差方程式，可以精確的修正 Y 座標的誤差達到 5mm 以內，確實增加量測精確度。

在完成 3.2 節中 3D 座標估測公式的校正後，我們可以由左右影像平面上的對應點算出相對的 3D 座標，並且可以將誤差縮小到 10mm 之內，對於之後手臂的控制，平台的控制，甚至是整個移動式機械臂(Mobile Manipulator)所需的控制命令都有相當大的助益。

表 3.9 Y 座標校正後的結果及誤差值

$Z_a$ (mm)	pose_Y	error_Y	$Z_a$ (mm)	pose_Y	error_Y
280.000	24.787	-2.213	868.000	26.754	-0.246
308.000	25.048	-1.952	896.000	26.793	-0.207
336.000	24.999	-2.001	924.000	26.832	-0.168
364.000	25.273	-1.727	952.000	27.160	0.160
392.000	25.691	-1.309	980.000	27.792	0.792
420.000	25.752	-1.248	1008.000	26.919	-0.081
448.000	26.092	-0.908	1036.000	27.275	0.275
476.000	26.156	-0.844	1064.000	26.649	-0.351
504.000	27.017	0.017	1092.000	25.995	-1.005
532.000	27.611	0.611	1120.000	26.685	-0.315
560.000	27.557	0.557	1148.000	26.001	-0.999
588.000	27.666	0.666	1176.000	26.722	-0.278
616.000	27.961	0.961	1204.000	26.005	-0.995
644.000	28.281	1.281	1232.000	26.390	-0.610
672.000	27.977	0.977	1260.000	25.616	-1.384
700.000	27.847	0.847	1288.000	24.801	-2.199
728.000	26.586	-0.414	1316.000	26.011	-0.989
756.000	26.389	-0.611	1344.000	25.597	-1.403
784.000	26.414	-0.586	1372.000	26.851	-0.149
812.000	26.440	-0.560	1400.000	26.011	-0.989
840.000	26.210	-0.790			

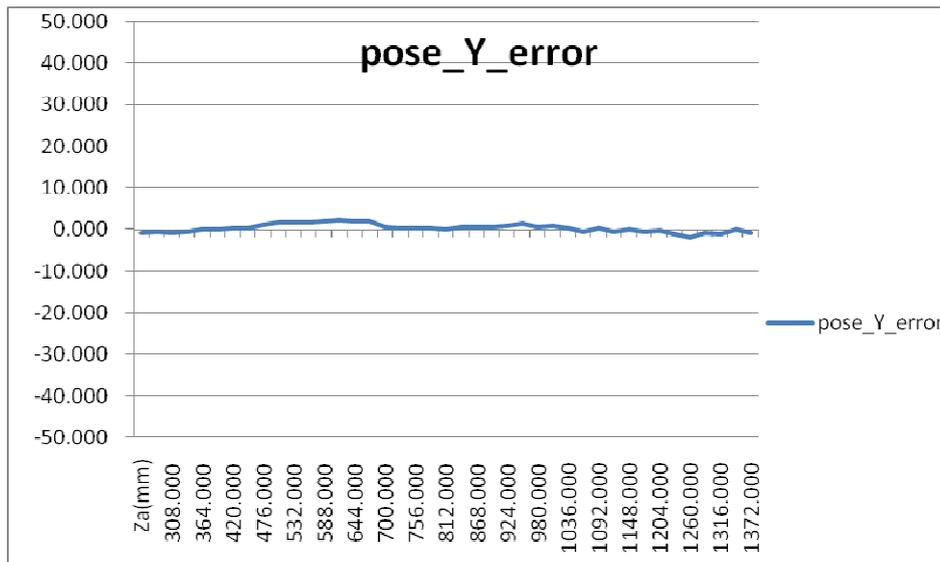


圖 3.12 校正後 Y 座標誤差值

### 3.3 三軸旋轉角

一個物體在  $\mathbb{R}^3$  姿態(pose) (位置及朝向) 通常是由六自由度表示：三個給位置，三個給朝向。因此，目標物的姿態(pose)通常被更簡潔地寫為 pose vector  $p = (X, Y, Z, \phi, \theta, \varphi)^T$ ，其中  $X, Y, Z$  代表位置， $\phi, \theta, \varphi$  是尤拉角(Euler angles) 代表朝向。

在做完 3D 座標估測後，我們可以利用 Homography 框出目標物在影像平面上的位置，如圖 2.12 跟圖 2.13，我們可以將這四個角點，當成 4 組匹配的特徵點，算出四個角點的 3D 座標，選定其中三個座標，組成  $V_x$  跟  $V_y$ ，再用  $V_x$  跟  $V_y$  外積得到  $V_z$ ，如圖 3.13 所標示，利用這三個向量跟 XYZ 三軸的夾角，就可以知道目標物相對於世界座標軸，分別旋轉了幾度如 (3-22)、(3-23)、(3-24)。

$$\phi = 90 - \cos^{-1} \frac{\vec{v}_y \cdot \vec{Z}}{|\vec{v}_y| |\vec{Z}|} \quad (3-22)$$

$$\theta = 90 - \cos^{-1} \frac{\vec{v}_z \cdot \vec{X}}{|\vec{v}_z| |\vec{X}|} \quad (3-23)$$

$$\varphi = 90 - \cos^{-1} \frac{\vec{v}_x \cdot \vec{Y}}{|\vec{v}_x| |\vec{Y}|} \quad (3-24)$$

這個設計的目的是在於得到目標物相對於機器人本體的角度變化，因為在更進一步的抓取規劃上，目標物不見得是直立於桌面，且正面朝著攝影機，他可能有其他角度的變化，例如：左右傾斜，前後傾斜，正面向左偏轉幾度等情況。若是我們能夠知道這些狀態，就可以調整我們抓取的姿態，以最適合的方式抓取目標物[12]。

為了驗證這個三軸角度計算方法的正確性，我們任意選取幾個角度，對每一筆選取的角度做十次估測取其平均值，觀察其實際估測結果所呈現出的誤差情況，如表 3.10 所示，當各軸旋轉的角度加大時，雖然是對單一軸旋轉，其他軸也會被算出轉動的角度，整體來說，各軸在旋轉 30 度之內時，誤差大致上在 10 度之內，但是隨著角度上升，誤差就更大，所以目前來說，這個角度估測方法適用在小角度變化時的計算，若是出現大角度的旋轉時，需要再輔以其他的演算法，以便能算出更精確的角度變化。

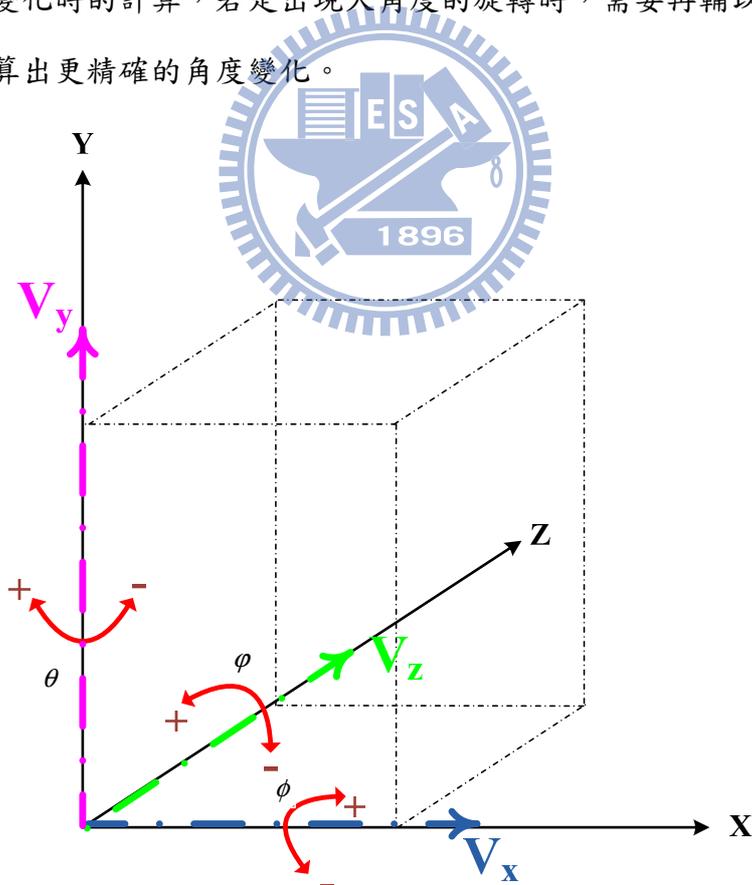


圖 3.13 目標物的三軸向量與世界座標的旋轉關係

表 3.10 三軸旋轉角實際測試結果

Assigned angle			Experiment angle			error		
X	Y	Z	X	Y	Z	X	Y	Z
0	0	0	-1.752	-1.771	0.204	1.752	1.771	0.204
-20.000	0.000	0.000	-19.365	-2.159	1.449	0.635	2.159	1.449
30.000	0.000	0.000	30.743	-16.348	-7.143	0.743	16.348	7.143
0.000	20.000	0.000	1.817	21.084	1.007	1.817	1.084	1.007
0.000	-30.000	0.000	7.481	-29.488	-2.536	7.481	0.512	2.536
0.000	0.000	30.000	0.946	-1.273	33.264	0.946	1.273	3.264
0.000	0.000	-20.000	8.475	-9.083	-21.417	8.475	9.083	1.417
0.000	0.000	-40.000	7.929	-10.085	-38.624	7.929	10.085	1.376



## 第四章移動式機械臂之影像伺服控制設計

經由第二章跟第三章所介紹的方法之後，我們可以得到目標物在空間中的三維座標。本章節的目的在說明我們如何使用這個估測出來的座標當作控制器的輸入，導引機器人向前移動並抓取目標物。首先在 4.1 節跟 4.2 節之中，我們分別介紹機械手臂跟全向式移動平台的控制設計。在 4.3 節中，將會介紹移動式機械臂整體的系統架構，說明移動式抓取的方法跟設計理念。

### 4.1 手臂控制

整體的控制透過工業電腦(IPC)，而利用 CANbus 作為傳輸介面，控制各馬達的運作。手臂的馬達控制器是由 Maxon 公司所生產的 EPOS Positioning Controller，我們可以用其內建的指令集，針對我們控制上的需要去撰寫程式。在本論文中，主要用到的是位置控制，當我們經由逆運動學得到各軸旋轉的角度後，將角度轉為馬達轉動所需要的資料型態，將這些資訊送給馬達，就可以讓手臂轉到我們所指定的座標位置，而在手臂從起點移動到終點的過程中，我們可以藉由讀回編碼器(Encoder)的數值，判斷各軸馬達是否已經轉到我們給定的轉動量，以便確認手臂是否已經移動至終點。

#### 4.1.1. 硬體構造

本論文所使用的機械手臂是由實驗室所自行設計的雙臂構造。為一具有 6 自由度的機械手臂，其中在肩膀及手肘關節的部分使用的是 Harmonic driver，這種設計可以讓機械手臂在運作時動作更加的平順。如圖 4.1 所示，單邊肩長 198mm，肩膀到手軸長 285mm，手軸到手腕長 287mm，手掌長度約為 200mm。

#### 4.1.2. 座標轉換

本論文中所使用的實驗平台由數個部分組合起來，包括立體攝影機、機器人頭部、機械手臂，移動平台，為了整合機器人的各個部分需要知道每個座標間的轉換關係，以利於整體的控制設計。機器人的座標原點定在雙臂肩關節的連線，

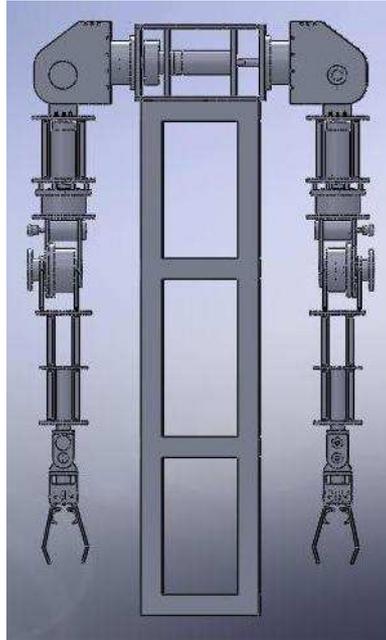


圖 4.1 機械手臂設計圖

跟頭部旋轉軸的交叉點，其中攝影機是主要的資訊來源，所以必須知道攝影機座標到手臂及平台間的座標轉換。整個移動式機械臂之間的座標關係，如圖 4.2 所示，其中的符號定義如下，C：立體視覺攝影機、B：機械手臂本體中心、E：機械手臂的手腕點、G：機械手臂的夾爪末端、P：全向式移動平台、O：欲抓取的目標物體，而 ${}^A H_B$  矩陣代表的意義是從 A 座標系轉換到 B 座標系之轉換矩陣。我們得知了機器人各個部位的座標轉換關係之後，就能夠針對控制的需求，將命令轉換到適當的座標系，例如：當機器人位於機械手臂的工作範圍之外時，我們就必須將攝影機所觀測到的座標轉換到移動平台的座標系，給出適當的命令，導引平台移動到手臂的工作範圍之內。

在圖 4.3 中所表達的是機器人本體的座標系(ver.new0)跟右機械手臂座標系(ver.0)之間的座標轉換關係，我們希望在手臂控制方面能夠先轉換到一個統一的座標系，之後再根據逆運動學的需要，將座標系轉換到左手或是右手，以符合手臂逆運動學所需的座標系。機器人本體的座標系(ver.new0)跟右機械手臂座標系(ver.0)之間的座標轉換關係之中，在 X 軸跟 Z 軸的座標原點及方向是一致的，不同的地方在於 Y 軸，右手臂的座標系式以肩關節為座標原點，但是我們希望整

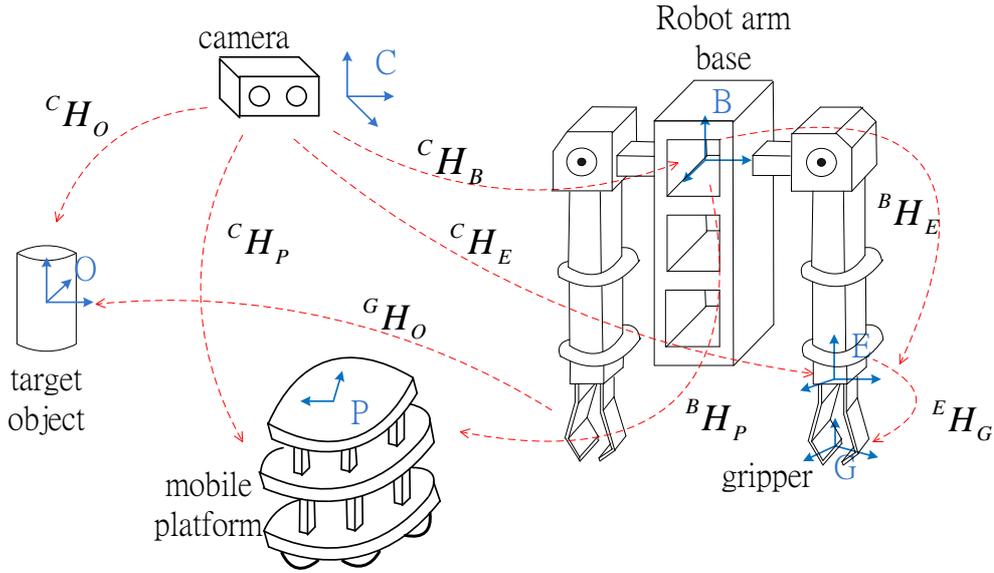


圖 4.2 機器人各部位及目標物之間的座標關係

體機器人的座標系原點是在中間，所以兩個座標系在 Y 軸會有相差 198cm(手臂肩膀長度)的情形出現。(4-1)式是機器人整體座標原點到右手臂座標原點的座標轉換。

$$ver.new0 \rightarrow ver.0: \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 198 \\ 0 \end{bmatrix} \quad (4-1)$$

在頭部座標系方面，由於機器人頭部具有兩個自由度，前後旋轉 $\theta_{16}$ 跟左右旋轉 $\theta_{15}$ ，在本論文所提出的控制設計中，在尋找目標物方面，我們希望攝影機跟機器人本體是正對著目標物。因此若是機器人正對著目標物，但是目標物不在攝影機水平視角的可見範圍時，要轉動 $\theta_{16}$ 才能看到目標物，如圖 4.4 所示，若是要轉動 $\theta_{15}$ 才能看到目標物，表示目標物在機器人的左側或右側，如圖 4.5 所示，我們會先給平台命令，使機器人移動到正對著目標物的位置。(4-2)式是攝影機座標系(camera)到機器人頭部座標系(head)的座標轉換，其相對關係如圖 4.4 所示。

$$camera \rightarrow head: \begin{bmatrix} pose\_head\_x \\ pose\_head\_y \\ pose\_head\_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\sin \theta_{16} \\ 0 & 0 & \cos \theta_{16} \end{bmatrix} * \begin{bmatrix} pose\_x \\ pose\_y \\ pose\_z \end{bmatrix} + \begin{bmatrix} 0 \\ 54 * \cos \theta_{16} \\ 54 * \sin \theta_{16} \end{bmatrix} \quad (4-2)$$

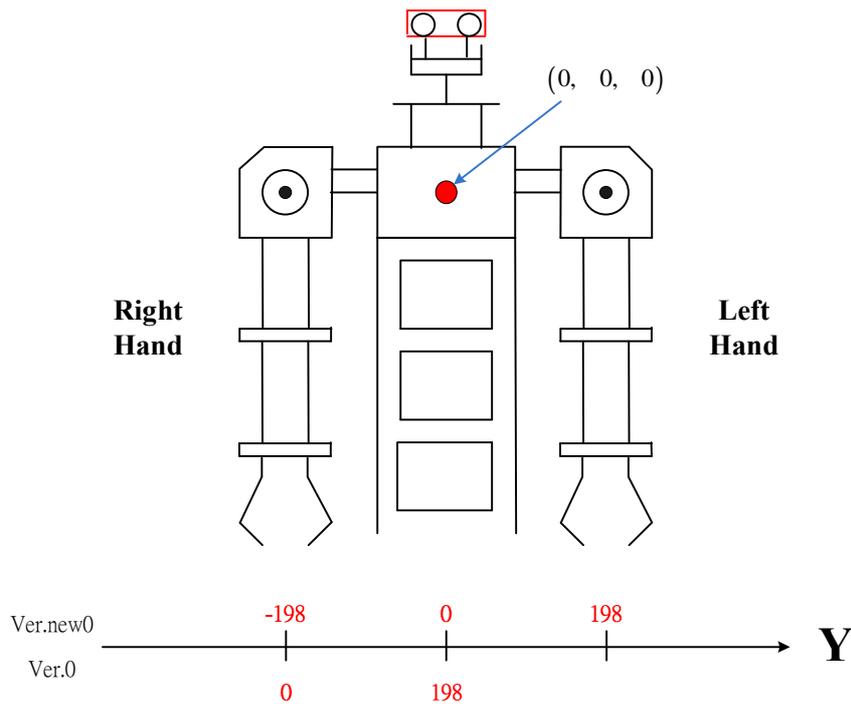


圖 4.3 base 跟 arm 座標軸的關係

而(4-3)式是機器人頭部座標系(head)到手臂肩關節座標系(ver.0)的座標轉換，表現出如何從攝影機量測到的座標，轉換到手臂逆運動學輸入的座標系。

$$head \rightarrow ver.0: \begin{bmatrix} pwx\_end \\ pwy\_end \\ pwz\_end \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} pose\_head\_x \\ pose\_head\_y \\ pose\_head\_z \end{bmatrix} + \begin{bmatrix} 0 \\ 198 \\ 187+55 \end{bmatrix} \quad (4-3)$$

圖 4.4 是攝影機跟目標物之間的座標轉換關係圖，攝影機裝設的位置是在一個具有二自由度的機器人頭結構，當目標物低於機器人水平視角時，頭部必須向下傾斜到一個能夠將目標物放在影像中間的角度，根據其中的幾何關係，可以得到機器人本體跟目標物之間的座標關係，作為導引控制器往前抓取目標物的控制命令。

當目標物不在機器人正前方時，移動平台或是機器人頭部需要旋轉一個角度  $\theta_{15}$ ，使攝影機能夠看到目標物以便計算其位置，如圖 4.5。當機器人出現旋轉的情況時，根據旋轉的角度及當時攝影機量測出來的座標位置，我們可以先由(4-2)將攝影機的座標系轉到機器人頭部的座標系，接著推導出在 XY 平面上，機器人跟目標物之間的相對關係。

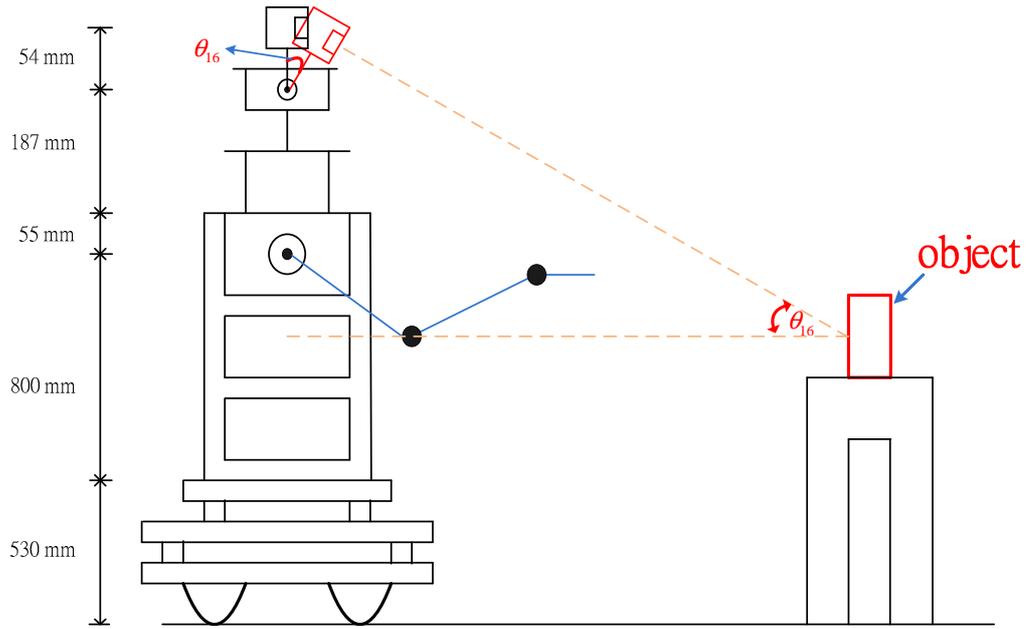


圖 4.4 機器人頭部跟目標物的座標關係

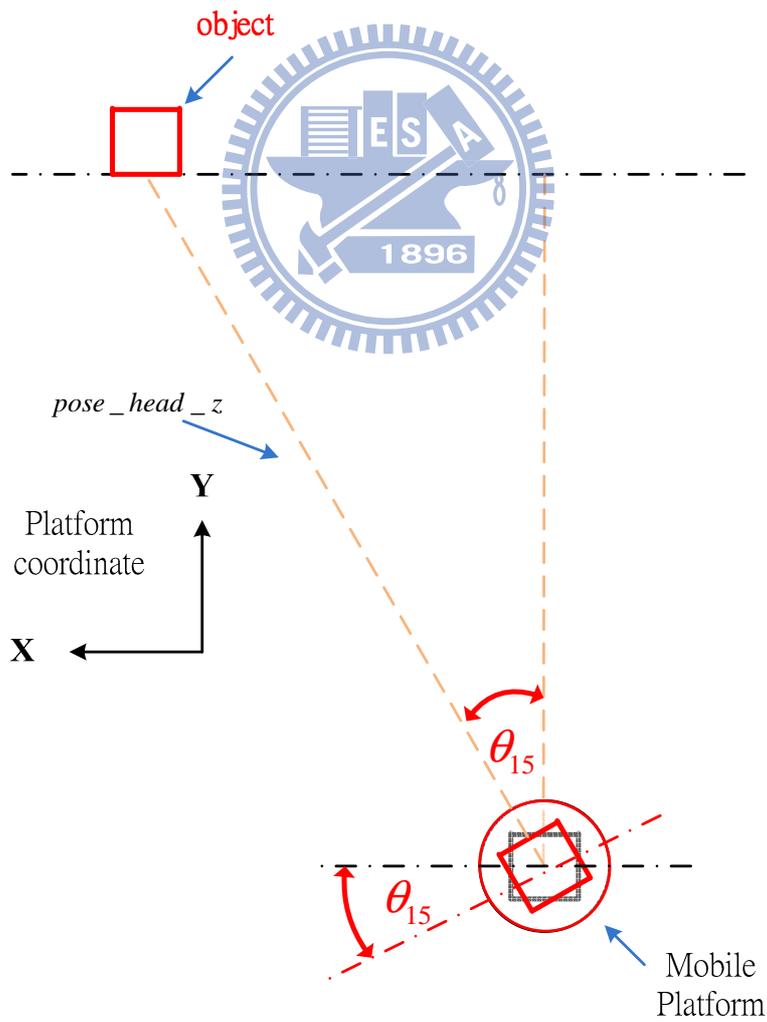


圖 4.5 移動平台跟目標物的座標關係

由(4-4)可以得到 X 方向的距離，也就是將機器人移動到目標物正前方的左右距離，而(4-5)可以得到機器人跟目標物在 Y 座標的距離，根據所量測出來的目標物三維座標，我們可以知道機器人抓取時跟目標物最適當的距離是多少，將 Y 值減掉這個距離，就是機器人到達到工作範圍所需移動的距離。

$$X = pose\_head\_z * \sin \theta_{15} \quad (4-4)$$

$$Y = pose\_head\_z * \cos \theta_{15} \quad (4-5)$$

### 4.1.3. 手腕逆運動學推導

在手臂運動學的方面，我們一開始推導到手腕的部分(4 軸)，針對手掌的部分，根據前面四軸的結果來推導手腕的逆運動學，可以讓整個手臂更加活用，不用因為手掌的不同，重新推導整隻手臂的逆運動學，增加工作效率，而且可以針對目標物的姿態不同來調整。本實驗中所使用的機械手臂一共是六個自由度，肩膀到手腕是四個，手掌部分是兩個，如圖 4.6(a)所示。手掌的設計上，具有旋轉跟上下轉動兩個自由度，利用這兩個自由度在抓取時讓手腕保持跟地面平行，如此一來就能確實的抓住放在桌面上的目標物。

在推導手腕的逆運動學方面，手臂各點位置定義如圖 4.6(b)所示，所需之各向量間的關係如圖 4.7 所示，首先由肩膀當作原點  $a$ ，加上已知的座標有手軸位置  $b$  跟手腕位置  $c$ ，將此三點組成一個平面  $\overline{E}_1$ ，算出此平面的法向量

$\overline{n}_1 = \overline{ba} \times \overline{ca}$ 。接著由空間中向量跟平面的夾角關係，如圖 4.8 所示，求出手腕兩自由度所需旋轉的角度。在抓取的任務當中，目前所設計的方法是假設瓶子直立於桌面的情況下來抓取，所以我們希望手掌部分保持跟地面平行，也就是跟 XY 平面平行，以便於在抓取目標物的時候能夠穩定的夾住物體。

在求得  $\overline{V}_1$  跟  $\overline{n}_1$  之後，分別求出與 XY 平面法向量 ( $\overline{Z} = (0, 0, 1)$ ) 的夾角，就可以得到手腕的兩自由度在使手掌平行 XY 平面時，所需要的角度。(4-6)是手腕旋轉的角度，(4-7)是手腕上下轉動的角度。

$$\theta_5 = \left| \sin^{-1} \frac{\bar{n}_1 \cdot \bar{Z}}{|\bar{n}_1| * |\bar{Z}|} \right| \quad (4-6)$$

$$\theta_6 = \left| \sin^{-1} \frac{\bar{V}_1 \cdot \bar{Z}}{|\bar{V}_1| * |\bar{Z}|} \right| \quad (4-7)$$

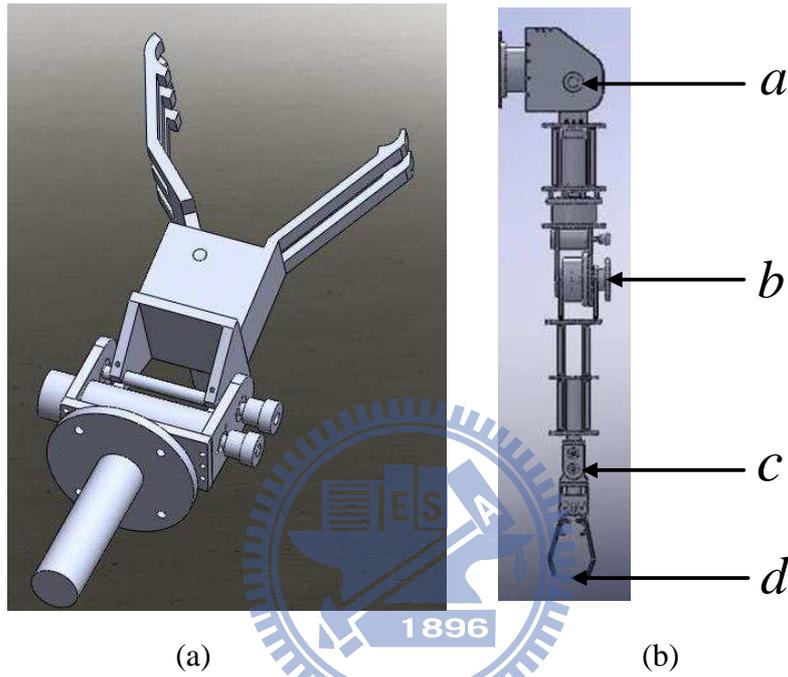


圖 4.6 (a)機械手爪設計圖(b)手臂各點代號位置

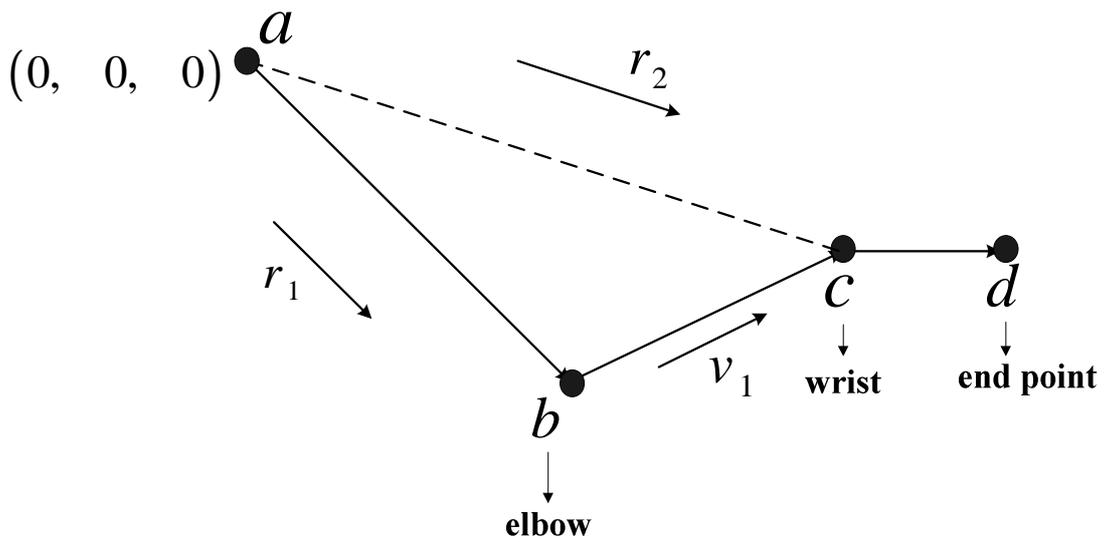


圖 4.7 機械手臂各關節點之間的關係圖

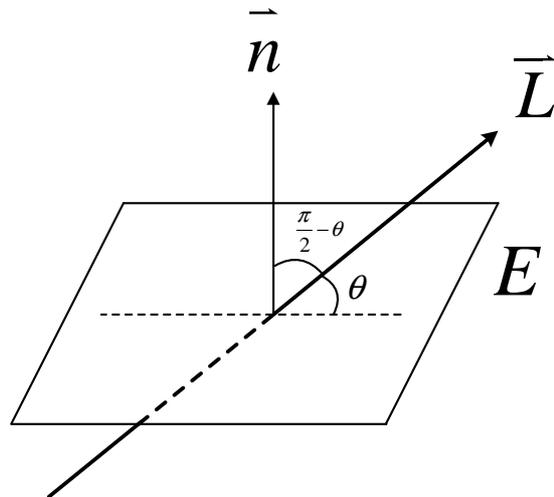


圖 4.8 空間中向量及平面的夾角關係

#### 4.1.4. 從假定腕部點逼近實際腕部點的方法

手臂逆向運動學的推導分為兩個階段，第一階段是肩關節到手腕的部分，第二階段則是利用前四軸的結果，推導出手掌的逆向運動學。讓手掌也就是夾爪開口保持水平的結果，使手掌能夠順利抓起目標物，但是預抓取位置(pre-grasp point)並不是我們當初所給的那點，因為本質上逆運動學的輸入還是手腕的位置，順向運動學所算回來的夾爪位置並不是我們所給定抓取目標物時，所需達到的預抓取位置(pre-grasp point)，所以為了讓逆向運動學的結果能夠是我們所給定的 pre-grasp point，我們需要得到一個最適合的手腕位置當作逆運動學的輸入，讓對應的夾爪位置是我們所給定的點。為了達到這個結果我們設計了一個逼近抓取點估測法。

由圖 4.9 我們可以看出，當我們給定抓取點  $b(pw\_end)$  時，減掉手掌長度約 20cm，會得到手腕位置  $c(pw\_rwst)$ ，但是將  $c$  代入逆向運動學之後，得到的抓取點位置是  $a(end\_point)$ ，跟我們所給定的抓取位置不同，於是根據  $a$ 、 $b$ 、 $c$  三點間的關係，我們假設對應  $b$  的腕部點是在  $c'$  的位置，劃出一個平行四邊形的區域。但是將  $c'$  代入逆運動學之後，得到的位置不一定是  $b$ ，於是我們繼續在  $c$  跟  $c'$  的連線上尋找，將算出來的抓取點最接近  $b$  的令為抓取時的腕部點位置，而對應此腕部點的就是逼近抓取點的結果  $b'$ 。

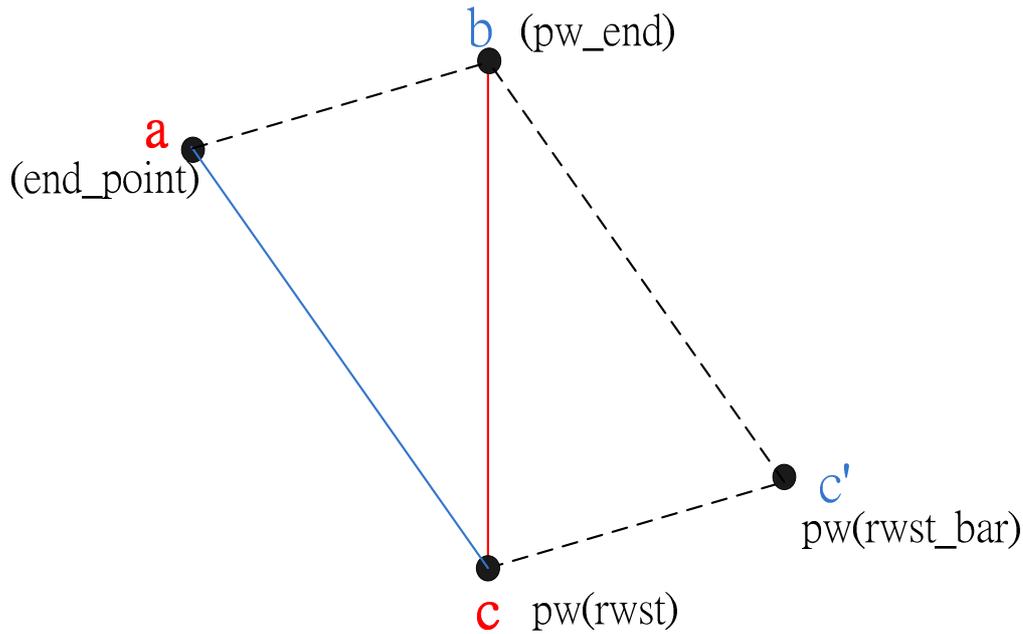


圖 4.9 給定抓取點  $b$ ，假定手腕點  $c$ ，對應  $c$  的抓取點  $a$ ，對應  $b$  的手腕點  $c'$ ，四點之間的相對關係圖。

以下是逼近抓取點方法的運算步驟：

- a. 運用平行四邊形原理求出參考點  $c'$  ( $rwst\_bar$ )

$$\overrightarrow{ba} = \overrightarrow{c'c}$$

- b. 將  $c'$  帶入逆向運動學，得到  $current\_end(b')$  做為跟給定抓取點  $pw\_end(b)$  比較的依據
- c. 算出目前手臂的終端點 ( $current\_end$ ) 跟給定要到達的終端點 ( $pw\_end$ ) 兩者間的距離 ( $dis\_end$ )，根據這個結果來判斷逆向運動學的結果是否合乎我們的要求
- d. 根據腕部點跟終端點的變化趨勢，可以看出符合給定點的對應腕部點是沿著  $\overrightarrow{cc'}$  來做變化，所以基本上就是搜尋這條線上的腕部點中，算出來最接近給定抓取點的，即是我們需要的結果
- e. 持續步驟 c 跟 d，直到  $dis\_end$  小於某個值 (在這邊是設定為 5mm)，或是  $b'$  已經到達  $b$  的邊界。此時的  $b'$  所對應的逆向運動學結果，就是我們所需要的最終結果。如圖 4.10 所示。

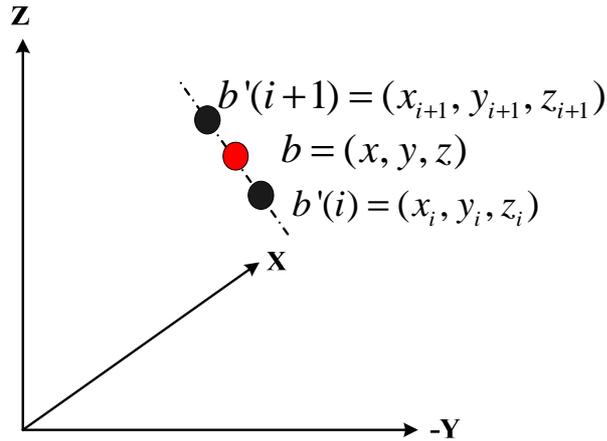


圖 4.10 逼近抓取點的示意圖，隨著  $b'$  漸漸接近  $b$ ，我們可以得到對應  $b$  的腕部點  $c'$

目前來說，這個逼近抓取點的方法能夠讓機械手臂真正的達到我們的給定抓取點，成功到達的機率在 90% 以上，誤差約在 1cm 之內，其餘產生較大誤差的情況在於目標物距離身體太近，或是工作範圍之邊界的比較極端的位置，但是由於整個移動式抓取的設計，能夠讓機器人移動到適當的抓取位置，所以整體而言這個方法在應用上是相當有效的。

#### 4.1.5. 抓取動作路徑規劃

根據機械手臂的機構，無法在每次抓取目標物時都保持夾爪正對著目標物，也就是手掌平行 X 軸垂直 Y 軸的方向，因此在抓取方面我們需要在手爪到達 pre-grasp 位置之後，做最後夾取的路徑規劃，而 pre-grasp point 可能在目標物的任何角度方向，沒辦法固定路徑抓取，所以在本論文中提出了一個針對各種情況來規畫路徑的方法。首先，在到達 pre-grasp point 之後，我們可以得到腕部 (pw\_rwst) 跟手爪末端 (pw\_end) 的座標點，這兩點的直線方程式必定會通過目標物的中心 (object\_center)，如圖 4.11 所示。

所以在求出此方程式之後，再加上夾爪中心 (pw\_end) 到目標物中心的距離算出目標物中心的座標 (Object\_center)。我們把夾爪跟目標物之間的位置關係放大，如圖 4.11 右上角所示，得到這兩點之後，就是讓手爪沿著這個路徑，到達終點後，把目標物夾起來，完成任務。

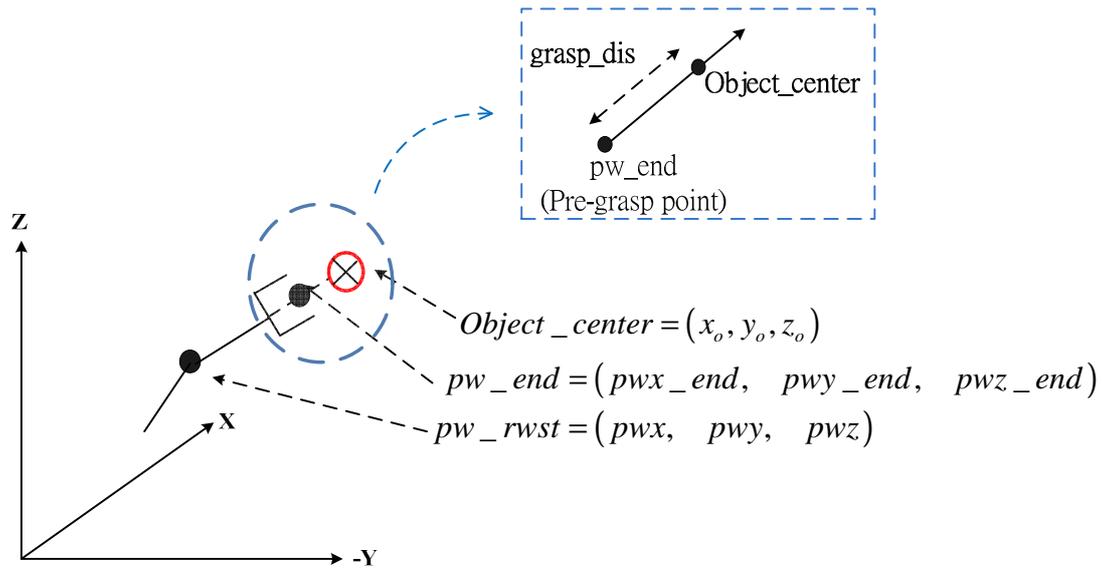


圖 4.11 夾取之動作規劃示意圖

## 4.2 移動平台控制

移動平台在本論文所提出的移動式抓取方法中扮演的相當重要的角色，當目標物須在機械手臂的工作範圍之外時，我們需要移動平台前進到達手臂工作範圍，讓機械手臂能夠順利的抓取目標物。本論文的影像服架構是沿用[24]中所推導出的平台移動模型及控制架構，本節中有詳細的說明。至於如何給定平台的控制命令，給予方向及距離在 4.1 節中有說明。

### 4.2.1. 硬體構造

本論文之全向式機器人採用全方向性的輪子[24]，如圖 4.12 所示，於輪子的周圍排列了六個滾軸，同時擁有橫向及縱向相互垂直的輪子，可以透過輪子間合力分力的關係，決定任何移動方向；並於輪軸附近，安插了彈簧式的機構，增加輪子與地面的接觸性。比起差動輪平台，全向式移動平台有更好的移動性，其優點為：(1)改變行進方向時，輪軸不需移動。(2)可在原地旋轉平台，不需要旋轉半徑。(3)可同時自旋與曲線運動。此類的移動平台，已被廣泛的討論。目前常見的全向式移動平台，大多以三軸到五軸的平台為主，在不同的考量下，這三類平台各有本身的優缺點，三軸在以 60 度為倍數的移動角度上，可以達到較好



圖 4.12 全向輪實體圖[24]

的移動效率，四軸則是在 90 度為倍數的角度上，能獲得遠大於其他兩者的效果，至於五軸的效能，則是在各種角度效能都相當平均。此外，在負重能力上，軸數越多負重能力越好。

本論文之全向式移動平台其主要規格如下：

- 外型(長×寬×高)：450×450×530(mm)
- 線速度：最高約 1m/sec
- 旋轉速度：最高約 4.44rad/sec
- 輪徑：約 60mm
- 電池規格：12V 2.2Ah / 7.2Ah 鉛酸電池

此全向式移動平台系統是以四個全方位輪所組成，以四個馬達驅動，採用 Faulhaber 公司所提供之馬達，其額定轉速為 6500 rpm，搭配減速齒輪組，其減速比為 14:1。此全向式平台除了能提供與其主軸垂直方向上轉動的方向外，還能提供沿著其主軸平行方向上移動的方向，因此我們可以藉由適當的設計移動機器人的底座，讓其具有任意方向平移或在移動時可以進行旋轉等動作。

馬達之伺服控制是由實驗室研製之雙軸運動控制卡完成，此運動控制卡採用 TI 之 DSP 運動控制晶片，DSP 晶片接收到速度命令後傳送給四輪馬達控制卡，經由伺服驅動器驅動馬達的運轉，馬達上之軸編碼器(Shaft encoder)可回傳機器人之移動資訊，經由里程計計算機器人姿態，其取樣時間為 51ms。

## 4.2.2. 平台運動模型

根據本實驗室學長之論文[24]，圖 4.13 為全向式移動平台四輪配置圖，其中標號 1~4 的方框表示四個相隔 90 度輪子的排列位置， $L$  為從全方位輪至全方位底盤中心點的距離，而  $v_i$  表示第  $i$  個輪子方向的速度。這樣的設計安排將使得四個全向輪具有對稱性，使得無論是在機構重心上的考量或是馬達運轉控制上都較為穩定與容易。在輪子轉動時，我們可得知每一個輪子方向轉動速度與其半徑、輪子自身角速度的關係如下：

$$v_i = r\dot{\phi}_i, \quad i=1 \sim 4 \quad (4-8)$$

其中  $r$  表示輪子半徑， $\dot{\phi}_i$  表示輪子自身角速度，如圖 4.14 所示。

現假設機器人前進的方向速度為  $V$ ，同時要機器人之角速度為  $\dot{\theta}$ ，此時首先把向量  $V$  拆解成所定義之座標平面上的二向量合  $\dot{x}$  和  $\dot{y}$ ：

$$V = \dot{x} + \dot{y} \quad (4-9)$$

然後藉由如圖 4.15 的幾何關係將可分別計算出  $k$  時刻  $\dot{x}$  與  $\dot{y}$  二向量在  $v_1$  方向的速度而得到如下式：

$$v_{1k} = -\sin\left(\frac{\pi}{4}\right)\dot{x}_k + \cos\left(\frac{\pi}{4}\right)\dot{y}_k + L\dot{\theta}_k \quad (4-10)$$

同理由以上方式，藉由四個輪子在空間中排列的對稱性，推得如下於  $k$  時刻之平

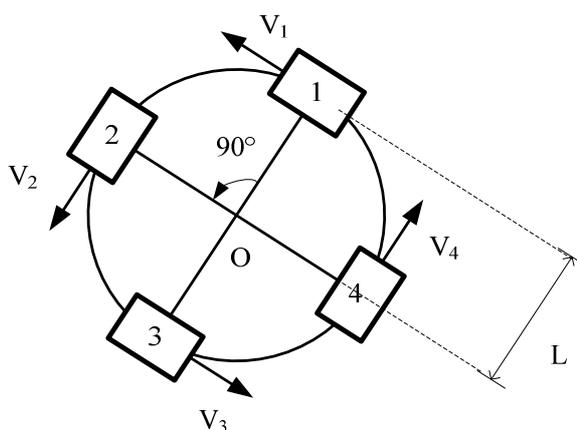


圖 4.13 平台四輪配置圖[24]

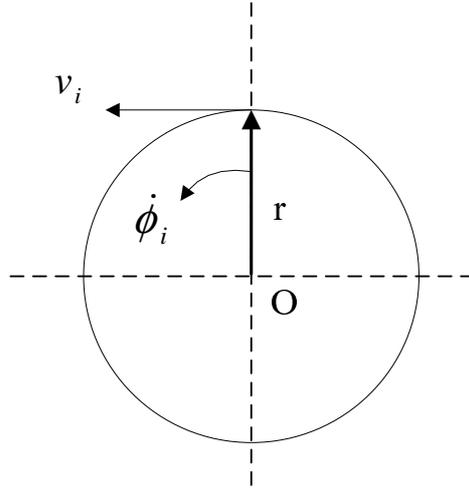


圖 4.14 全向輪轉動示意圖[24]

台速度  $\begin{bmatrix} \dot{x}_k & \dot{y}_k & \dot{\theta}_k \end{bmatrix}^T$  轉換成四輪速度  $v_{1k} \sim v_{4k}$  的轉換方程式：

$$\begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} = \begin{bmatrix} -\sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & L \\ -\sin(\frac{\pi}{4}) & -\cos(\frac{\pi}{4}) & L \\ \sin(\frac{\pi}{4}) & -\cos(\frac{\pi}{4}) & L \\ \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & L \end{bmatrix} \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} \quad (4-11)$$

在這裡等號右邊的第一個矩陣為常數，因此由(4-11)式我們將可用來對此全向式平台做任意方向速度移動的控制。

為了之後平台的定位控制，要對平台本身座標和世界座標作清楚的定義，以便估測平台於世界座標的位置。本論文將世界座標(Global)軸定義為  $X$  和  $Y$ ，平台局部座標(Local)軸為  $X_R$  和  $Y_R$ 。當平台以旋轉角速度  $\dot{\theta}$  旋轉之後，平台本身的座標軸  $X_R$  和  $Y_R$  相對於世界座標軸  $X$  和  $Y$  會產生如圖 4.16 所示一朝向角  $\theta$ ，若機器人朝著  $\theta$  方向以同一速度  $\begin{bmatrix} \dot{x}_k & \dot{y}_k & \dot{\theta}_k \end{bmatrix}^T$  繼續移動，根據先前運動方程式的推導方法，同樣可以推導出此時全方位移動機器人的運動方程式如下，藉由此式可由平台速度  $\begin{bmatrix} \dot{x}_k & \dot{y}_k & \dot{\theta}_k \end{bmatrix}^T$  推得四輪速度  $v_{1k} \sim v_{4k}$ ：

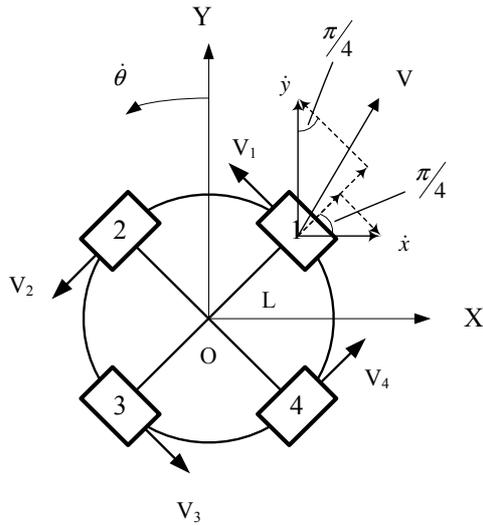


圖 4.15 全向輪之幾何關係[24]

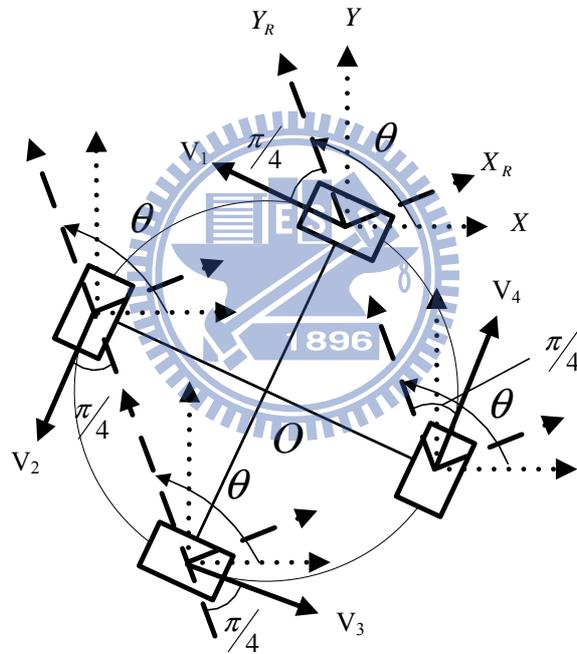


圖 4.16 平台座標旋轉示意圖[24]

$$\begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} = \begin{bmatrix} -\sin(-\frac{\pi}{4} + \theta) & \cos(-\frac{\pi}{4} + \theta) & L \\ -\sin(\frac{3\pi}{4} - \theta) & -\cos(\frac{3\pi}{4} - \theta) & L \\ \sin(-\frac{\pi}{4} + \theta) & -\cos(-\frac{\pi}{4} + \theta) & L \\ \sin(\frac{3\pi}{4} - \theta) & \cos(\frac{3\pi}{4} - \theta) & L \end{bmatrix} \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = T(\theta) \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} \quad (4-12)$$

令速度轉換矩陣為 $T(\theta)$ ，為了估測平台於 $k$ 時刻的姿態，需由四輪速度 $v_{1k} \sim v_{4k}$ 先推得平台本身速度 $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ ，將(4-12)式之 $T(\theta)$ 作 Jacobian 反矩陣運算，得到以 $v_{1k} \sim v_{4k}$ 為輸入的運動模型：

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} -0.5 \sin(-\frac{\pi}{4} + \theta) & -0.5 \sin(\frac{3\pi}{4} - \theta) & 0.5 \sin(-\frac{\pi}{4} + \theta) & 0.5 \sin(\frac{3\pi}{4} - \theta) \\ 0.5 \cos(-\frac{\pi}{4} + \theta) & -0.5 \cos(\frac{3\pi}{4} - \theta) & -0.5 \cos(-\frac{\pi}{4} + \theta) & 0.5 \cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} \quad (4-13)$$

### 4.2.3. 姿態估測 (Odometry)

得到(4-13)式後，便可由此基於平台四個馬達之速度估測，預測機器人姿態。如圖 4.17 所示， $(x, y)$  為平台所在座標位置，定義機器人於世界座標之朝向角 $\theta$ 右手算起為 $0$ 度，前方為 $90$ 度。假設機器人在 $\Delta t$ 時間內是以固定速度朝 $\theta$ 方向沿一直線軌跡移動，若機器人在 $k-1$ 時刻之姿態為 $[x_{k-1} \quad y_{k-1} \quad \theta_{k-1}]^T$ ，且保持移動速度經過 $\Delta t$ 時間後，其 $k$ 時刻姿態 $[x_k \quad y_k \quad \theta_k]^T$ 可由 $T(\theta)$ 作 Jacobian 反矩陣運算得到並表示成如(4-14)式：

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} &= \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} -0.5 \sin(-\frac{\pi}{4} + \theta) & -0.5 \sin(\frac{3\pi}{4} - \theta) & 0.5 \sin(-\frac{\pi}{4} + \theta) & 0.5 \sin(\frac{3\pi}{4} - \theta) \\ 0.5 \cos(-\frac{\pi}{4} + \theta) & -0.5 \cos(\frac{3\pi}{4} - \theta) & -0.5 \cos(-\frac{\pi}{4} + \theta) & 0.5 \cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \\ v_{4k} \end{bmatrix} \Delta t \\ &= \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} 0.5[-\sin(-\frac{\pi}{4} + \theta)v_{1k} - \sin(\frac{3\pi}{4} - \theta)v_{2k} + \sin(-\frac{\pi}{4} + \theta)v_{3k} + \sin(\frac{3\pi}{4} - \theta)v_{4k}] \Delta t \\ 0.5[\cos(-\frac{\pi}{4} + \theta)v_{1k} - \cos(\frac{3\pi}{4} - \theta)v_{2k} - \cos(-\frac{\pi}{4} + \theta)v_{3k} + \cos(\frac{3\pi}{4} - \theta)v_{4k}] \Delta t \\ \frac{1}{4L}(v_{1k} + v_{2k} + v_{3k} + v_{4k}) \Delta t \end{bmatrix} \end{aligned} \quad (4-14)$$

其中  $\Delta t$  為 Sampling time，當機器人移動時，藉由馬達上之 Encoder 回傳四輪 Pulse 數，得到機器人四輪之速度命令  $v_{1k} \sim v_{4k}$ ，進而推算機器人於  $\Delta t$  時間內的位移。此即為全向式機器人於世界座標上的姿態估測。

#### 4.2.4. 平台之運動控制

有了機器人姿態估測後，就可對其進行運動控制。根據[24]之內容其中最簡單的設計是利用一 P 控制器對機器人四輪下控制命令，圖 4.18 為控制方塊圖。

設定目標點為  $[x_d \ y_d \ \theta_d]^T$ ，Odometer 得到機器人於 k 時刻的姿態為  $[x_k \ y_k \ \theta_k]^T$ ，將兩者相減得到機器人姿態誤差：

$$\begin{bmatrix} x_{ek} \\ y_{ek} \\ \theta_{ek} \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (4-15)$$

透過控制器(Controller)將姿態誤差  $x_{ek}$ 、 $y_{ek}$ 、 $\theta_{ek}$  各乘上  $K_x$ 、 $K_y$ 、 $K_\theta$ ，得到

控制命令  $V_{cmd\_k}$ ：

$$V_{cmd\_k} = \begin{bmatrix} \dot{x}_{cmd\_k} \\ \dot{y}_{cmd\_k} \\ \dot{\theta}_{cmd\_k} \end{bmatrix} = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_\theta \end{bmatrix} \begin{bmatrix} x_{ek} \\ y_{ek} \\ \theta_{ek} \end{bmatrix}, \text{ 且 } K_x, K_y, K_\theta > 0 \quad (4-16)$$

控制命令  $V_{cmd\_k}$  經由四輪轉換函數  $T(\theta)$  可得對機器人控制之四輪速度命令：

$$\begin{bmatrix} v_{1cmd\_k} \\ v_{2cmd\_k} \\ v_{3cmd\_k} \\ v_{4cmd\_k} \end{bmatrix} = \begin{bmatrix} -\sin(-\frac{\pi}{4} + \theta) & \cos(-\frac{\pi}{4} + \theta) & L \\ -\sin(\frac{3\pi}{4} - \theta) & -\cos(\frac{3\pi}{4} - \theta) & L \\ \sin(-\frac{\pi}{4} + \theta) & -\cos(-\frac{\pi}{4} + \theta) & L \\ \sin(\frac{3\pi}{4} - \theta) & \cos(\frac{3\pi}{4} - \theta) & L \end{bmatrix} \begin{bmatrix} \dot{x}_{cmd\_k} \\ \dot{y}_{cmd\_k} \\ \dot{\theta}_{cmd\_k} \end{bmatrix} \quad (4-17)$$

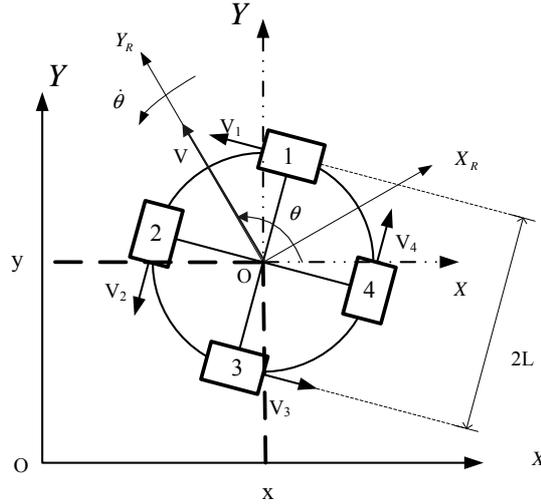


圖 4.17 平台運動模型圖[24]

機器人移動後，接收到馬達上之 Encoder 回傳值  $v_{1\_pulse} \sim v_{4\_pulse}$ ，經由(4-18)式轉

換成速度  $v_{1\_actual} \sim v_{4\_actual}$ ：

$$v_{i\_actual} = (v_{i\_pulse} / 2000) \cdot (0.12\pi) / 14 / \Delta t, \quad i = 1 \sim 4 \quad (4-18)$$

其中馬達轉一圈所需的 Pulse 數為 2000，故  $v_{i\_pulse} / 2000$  即為輪子所轉的圈數。輪子之半徑為 0.06m， $0.12\pi$  即為輪子週長。齒輪減速比為 14， $\Delta t$  為 Sampling time。如(4-19)式，接著將四輪實際速度  $v_{1\_actual} \sim v_{4\_actual}$  經由  $T(\theta)^{-1}$  反求出機器人

於 k 時刻之速度  $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$ ，透過 (4-20) 式，將  $[\dot{x}_k \quad \dot{y}_k \quad \dot{\theta}_k]^T$  乘上  $\Delta t$ ，加上

機器人於 k-1 時刻的姿態  $[x_{k-1} \quad y_{k-1} \quad \theta_{k-1}]^T$  後，便得到機器人於 k 時刻之姿態

$[x_k \quad y_k \quad \theta_k]^T$  作為速度控制的輸入，完成一輪控制迴圈。

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} -0.5 \sin(-\frac{\pi}{4} + \theta) & -0.5 \sin(\frac{3\pi}{4} - \theta) & 0.5 \sin(-\frac{\pi}{4} + \theta) & 0.5 \sin(\frac{3\pi}{4} - \theta) \\ 0.5 \cos(-\frac{\pi}{4} + \theta) & -0.5 \cos(\frac{3\pi}{4} - \theta) & -0.5 \cos(-\frac{\pi}{4} + \theta) & 0.5 \cos(\frac{3\pi}{4} - \theta) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_{1\_actual} \\ v_{2\_actual} \\ v_{3\_actual} \\ v_{4\_actual} \end{bmatrix} \quad (4-19)$$

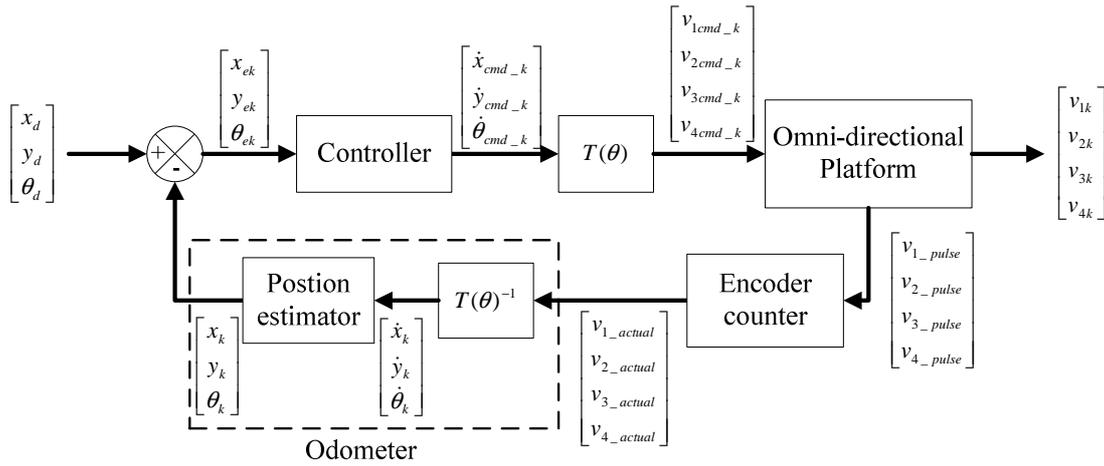


圖 4.18 平台控制方塊圖[24]

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \dot{x}_k \Delta t \\ \dot{y}_k \Delta t \\ \dot{\theta}_k \Delta t \end{bmatrix} \quad (4-20)$$

### 4.3 影像伺服控制

在 4.1 跟 4.2 節之中，我們知道了在整個移動式抓取的架構之下，如何得到手臂跟平台的控制命令。本節將介紹整個完整的影像伺服移動式抓取的方法。

#### 4.3.1. 視覺伺服控制架構

機器人自主抓取之功能在於如何利用攝影機所取得的影像來導引移動式機械臂到達目標所在位置，此為一個視覺伺服(Visual servo)問題[25]。所謂的視覺伺服泛指所有利用視覺控制的機器人系統，依照控制器輸入資訊的不同，可分為兩類，分別是基於位置的視覺伺服(Position-based visual servo, PBVS)和基於影像的視覺伺服(Image-based visual servo, IBVS)[26][[27]。

在 IBVS 的架構中，是藉由攝影機取得影像後，經過特徵點擷取得到目前目標的特徵點座標(Current feature)，將其直接與事先所定義好的目標特徵點座標(Desired feature)進行相減，便可以得到兩者之間的誤差；利用此誤差作為特徵空間(Feature space)中的輸入訊號，藉此產生適當的控制訊號，以修正攝影機與目標物之間的位置，達成將機器人導引至最後預定到達的位置。

反之在 PBVS 架構中，其需要一個事先建立的目標物體幾何模型來估測目標

物體的姿態，利用估測的模型和實際的目標物體的誤差，當作控制器的輸入，回傳實際目標物的姿態，持續做修正，直到姿態之間的誤差小於設定的閾值，表示完成任務。

### 4.3.2. 控制設計

本論文採用視覺伺服為 PBVS 架構，藉由計算出空間中實際的座標關係來對機器人做控制，避免在 IBVS 架構中，雖然影像平面上的參考點已經到達所需的位置，但是實際上機器人卻尚未移動到所需的位置，這種情況的發生。對於整個移動式機械臂來說，攝影機所獲取的資訊是所有控制器的控制命令來源，包括頭部、手臂、平台。從第三章我們得知，如何由對應的特徵點算出 3D 座標，接著由 4.1 節跟 4.2 節得知機器人跟目標物之間的關係，來決定機器人的移動，導引平台前進，並控制手臂抓取目標物。

與一般 PBVS 架構的不同地方在於本論文針對欲抓取的目標物並不是先建立其模型，而是建立一張目標物的影像於資料庫中，作為我們一開始在第二章所提到特徵點比對的依據，藉由整個目標物辨識的過程接續著第三章的座標估測，可以算出所有對應點的 3D 座標。至於要如何從這群座標中挑選出抓取所需要的參考點，我們引入了[28]所提到中心點(Center point)的概念，運用 Homography 可以框出在左右攝影機影像平面中目標物的位置，取這兩個框的中心點，當作我們抓取所需的參考點，由這組對應的中心點所得到的 3D 座標，來當作機器人跟目標物位置關係的依據，藉由 encoder 回傳手臂位置跟攝影機量測目標物的位置之間的誤差是否已經小於某一個誤差值，來判斷是否完成抓取的任務。

控制器法則(control law)裡的行為模式如圖 4.19 所示，由於機器人並非一開始就位於機械手臂的工作範圍之內，目標物也可能會移動，所以針對這些情況在影像伺服的過程中設計了不同的模式，機器人根據目前辨識的結果選擇最適當的模式來操作，增加機器人的智慧性。

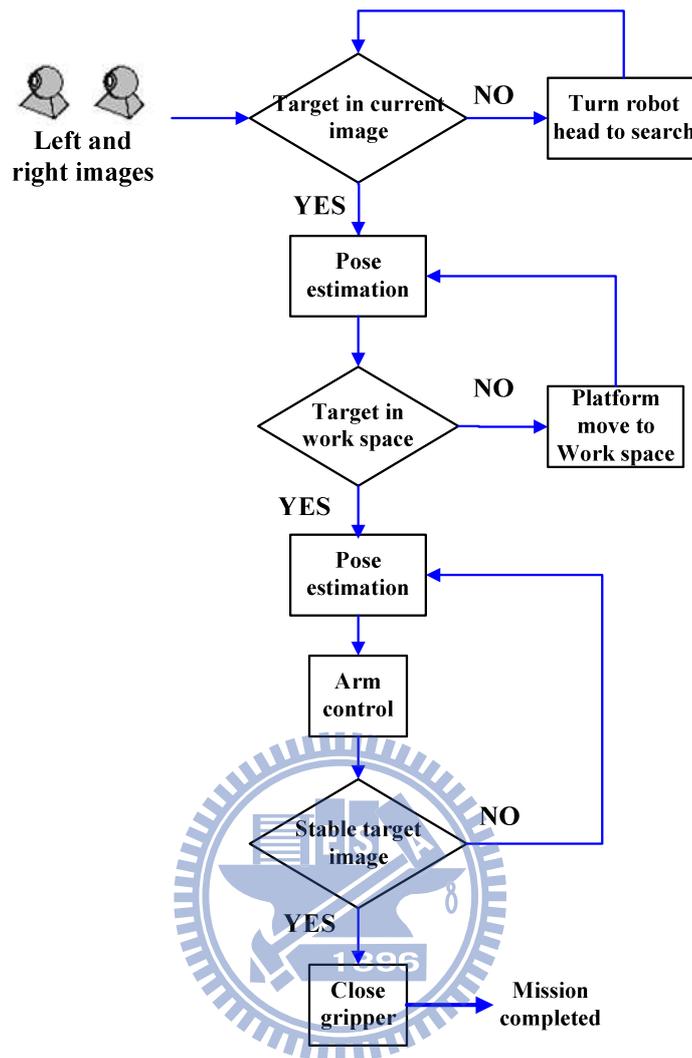


圖 4.19 移動式機械臂工作模式流程圖

整體的系統架構，如圖 4.20 所示。將輸入的影像做特徵點擷取之後得到兩群特徵點  $[x_{il} \ y_{il}]^T$  以及  $[x_{ir} \ y_{ir}]^T$ ，接著和資料庫特徵點  $[x_{id} \ y_{id}]^T$  做第一次的特徵比對，得到兩組比對成功的特徵點  $[x_{il} \ y_{il} \ x_{id} \ y_{id}]^T$  以及  $[x_{ir} \ y_{ir} \ x_{id} \ y_{id}]^T$ ，分別計算這兩組的特徵點的平面轉換矩陣，運用 RANSAC 將第一次比對錯誤的特徵點剔除，得到比對成功率更高的一組特徵點  $[\overline{x_{il}} \ \overline{y_{il}} \ \overline{x_{id}} \ \overline{y_{id}}]^T$  以及  $[\overline{x_{ir}} \ \overline{y_{ir}} \ \overline{x_{id}} \ \overline{y_{id}}]^T$ ，由這組特徵點算出平面轉換矩陣之後，就能夠將目標物在左右攝影機影像平面的位置框出來，並以此區域的中心當作計算控制參考點的那組特徵點  $[x_l \ y_l \ x_r \ y_r]^T$ ，由這組對應點可以算出目

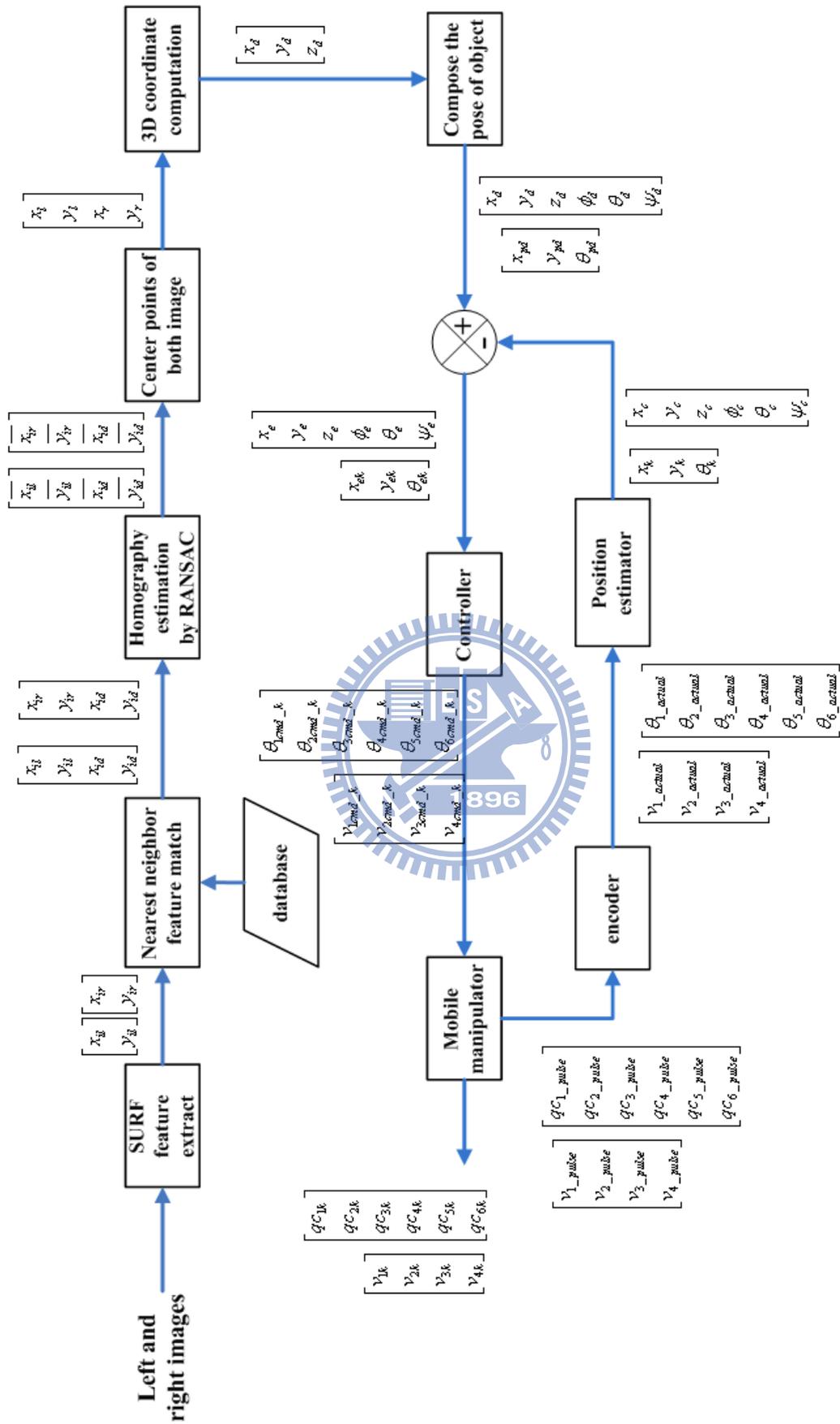


圖 4.20 整體系統架構圖

標物的中心點在空間中的三維座標 $[x_d \ y_d \ z_d]^T$ ，將這個座標加上目標物的三軸旋轉角就可以組成一個完整的目標物 6D 姿態 $[x_d \ y_d \ z_d \ \phi_d \ \theta_d \ \varphi_d]^T$ ，此外由目標物座標可以算出平台的位置 $[x_{pd} \ y_{pd} \ z_{pd}]^T$ ，作為平台移動的控制命令。在控制迴路中，將手臂末端位置及平台位置的誤差輸入，轉換成修正誤差所需給的命令，手臂方面是 $\theta_1 \sim \theta_6$ ，移動平台是 $[x_d \ y_d \ z_d]^T$ ，控制器法則(control law)裡的功能在於判斷機器人目前的狀態，選擇機器人要完成抓取任務目前所需的行為模式，例如:要繼續移動平台，還是尋找目標物，手臂是否開始執行抓取的動作等等。當移動抓取完成時，所有的位置都跟我們的估測結果一致，誤差達到最小，任務完成。



## 第五章實驗結果

本章節的主要目的在驗證第二章及第三章介紹之特徵點擷取方法及座標估測方法的確可使機器人辨識出目標物在空間中的位置，並且整合第四章所設計的控制架構，達成移動機器人抓取功能的實現。

### 5.1 實驗用機器人構造

圖 5.1 為本實驗所使用之機器人，機器人的運算核心為工業電腦，攝影機在擷取影像之後，辨識目標物及下達控制命令都是由電腦處理。機器人最上方為立體攝

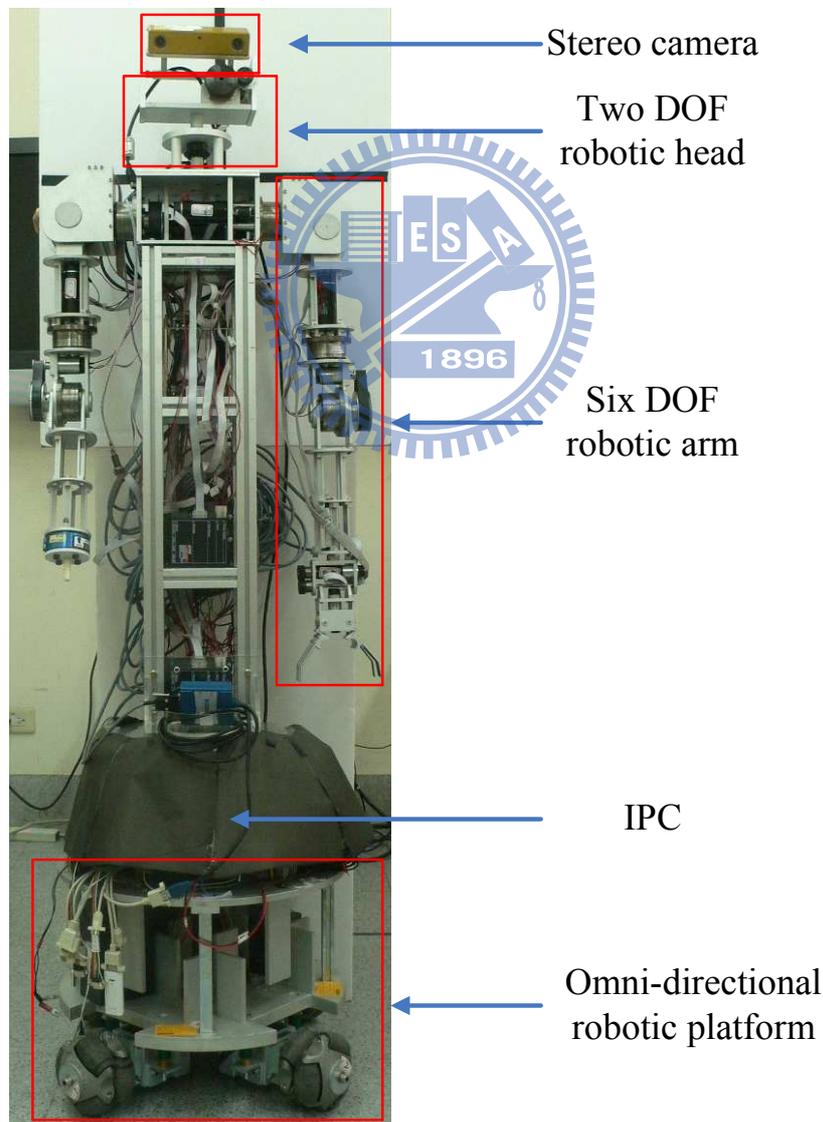


圖 5.1 實驗用機器人



圖 5.2 Bumblebee2 立體攝影機

影機，實驗所使用的攝影機為 Point Grey Research, Inc.所生產的 Bumblebee2 立體攝影機，他是透過 IEEE-1394 的傳輸介面跟電腦進行連結，左右眼的間距為 12cm。

攝影機裝置在實驗室自行設計的二自由度機械頭部上面，其下是機械手臂本體，構造如 4.1.1 節所介紹，最下面為全向式移動平台，構造如 4.2.1 節所介紹。

機器人的硬體架構圖 5.3 如所示，以工業電腦(IPC)為運算核心，透過 IEEE-1394 跟立體攝影機傳輸擷取到的影像，而移動平台方面，則是透過 RS-232 傳輸，在機械手臂及頭部馬達的部分，則是使用 CANbus 傳輸控制命令。

在做實驗之前，我們需要得到攝影機參數，表 5.1 是原廠所附的攝影機參數列表，而表 5.2 則是使用 Matlab Toolbox 所得到的攝影機參數列表。在本實驗中，我們所需要用到的參數只有焦距(Focal Lengths)跟鏡心距離(Baseline)，這兩個參數為計算目標物中心點之三維座標所需要的數值，而其他數值則是列出來讓讀者參考。

表 5.1 Bumblebee2 立體攝影機之規格[29]

Specification	BB2-03S2
Image Sensor Type	Sony® 1/3" progressive scan CCD ICX424 (648x488 max pixels) 7.4µm square pixels
Baseline	12 cm
Focal Lengths	2.5mm with 97° HFOV or 3.8mm with 66° HFOV or 6mm with 43° HFOV
A/D Converter	12-bit analog-to-digital converter
White Balance	Automatic / Manual (Color model)
Frame Rates	48 FPS
Interfaces	6-pin IEEE-1394a for camera control and video data transmission
Voltage Requirements	8-30V via IEEE-1394 interface or GPIO connector

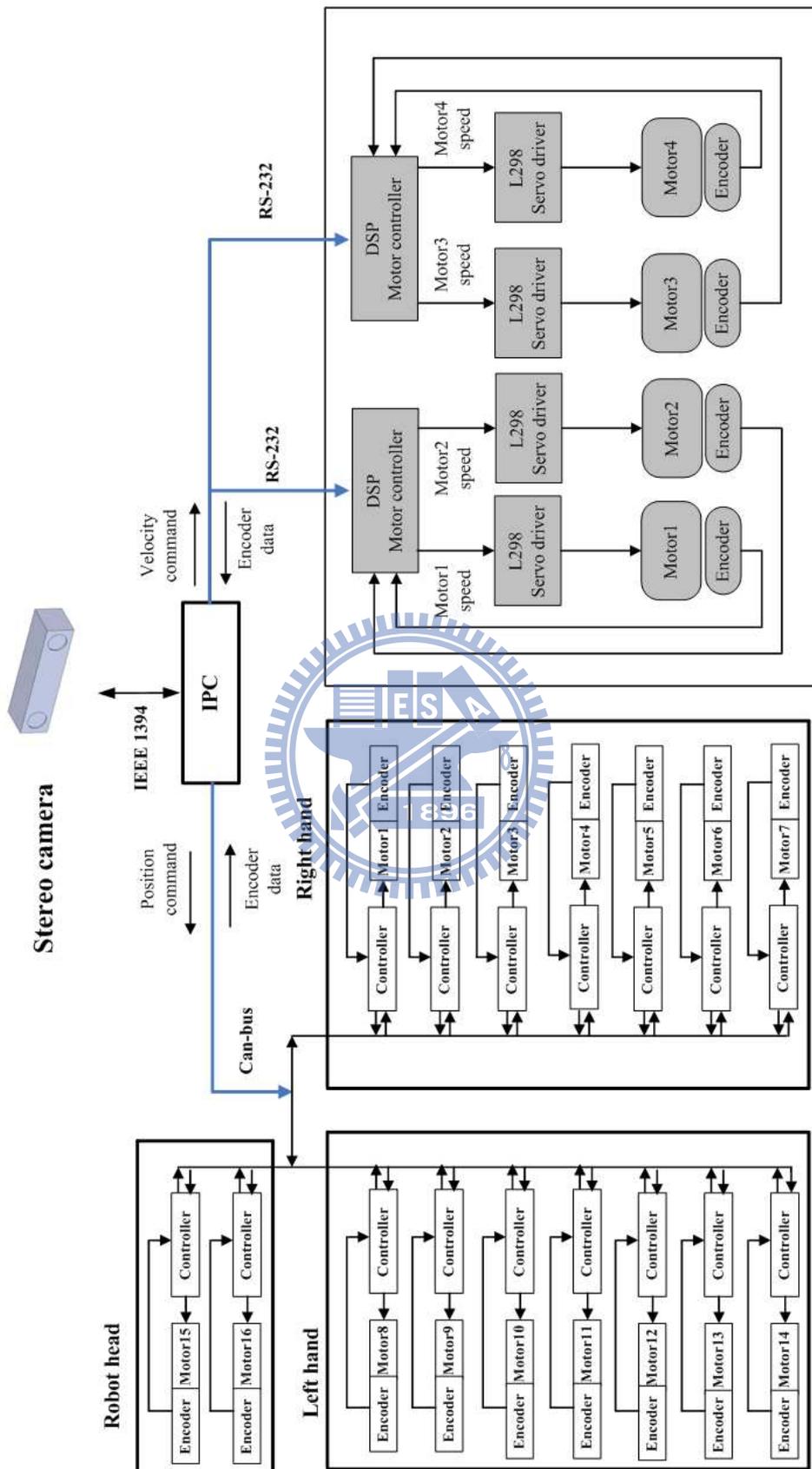


圖 5.3 機器人硬體架構圖

表 5.2 使用 Matlab Toolbox 得到的左攝影機內部參數

Intrinsic parameters of left camera	
Focal Length	[ 412.40757 417.62677 ]± [ 9.67967 9.58434 ]
Principal point	[ 198.34122 108.10353 ]± [ 10.94986 11.51069 ]
Skew	[ 0.00000 ]± [ 0.00000 ] => angle of pixel axes = 90.00000±0.00000 degrees
Distortion	[ 0.26032 -0.26006 -0.02099 0.03091 0.00000 ] ± [ 0.07935 0.17035 0.01609 0.01570 0.00000 ]

表 5.3 使用 Matlab Toolbox 得到的右攝影機內部參數

Intrinsic parameters of right camera	
Focal Length	[ 418.08286 419.27400 ]± [ 9.92160 9.51373 ]
Principal point	[ 170.60282 126.22348 ]± [ 13.37103 8.93098 ]
Skew	[ 0.00000 ]± [ 0.00000 ] => angle of pixel axes = 90.00000±0.00000 degrees
Distortion	[ 0.15381 -0.11568 0.00172 0.03264 0.00000 ] ± [ 0.08507 0.20307 0.00865 0.01523 0.00000 ]

表 5.4 使用 Matlab Toolbox 得到的攝影機外部參數

Extrinsic parameters (position of right camera wrt left camera)	
Rotation vector	[ 0.04255 0.07063 0.00280 ] ± [ 0.02921 0.03697 0.00346 ]
Translation vector	[ -121.01237 -0.74298 -4.41426 ] ± [ 1.57676 1.30451 7.80259 ]

## 5.2 平台靜止抓取實驗

本節實驗中，將驗證先前各章所述方法是否能夠達成抓取目標物的目的。首先是針對機械手臂已經到達工作範圍之內，平台靜止的情況。實驗環境方面，目標物離地約 105 公分，機器人距離目標物約為 60cm。圖 5.4 為建立目標物資料庫之特徵點影像。

在實驗的過程中透過攝影機讀取目前的影像跟資料庫的影像做比對，算出對應的特徵點之後，由 RANSAC 得到平面轉換矩陣，接著框出目標物在影像平面上的位置，以此框框的平均位置當作中心點，將左右眼的中心點代入



圖 5.4 (a)資料庫物件及(b)其特徵點位置

座標估測公式，求出目標物在空間中的座標，重複上述步驟，不斷的更新，當辨識的結果穩定，小於我們所設定的閾值時，進入到抓取的模式，移動機械臂到目標物前方，並做抓取路徑規劃，手爪沿路徑前進之後，闔上夾爪，完成任務。

第一個實驗的情境是機器人已經位於工作範圍內，此時藉由攝影機擷取目前影像，算出目標物的三維座標，控制機械手臂抓取目標物。圖 5.5 所顯示的是攝影機所觀察到目標物在影像平面的位置，透過本論文設計的目標物辨識方法，能夠準確的框出目標物的位置。我們將啟動程式到抓取成功過程中，所計算出的目標物中心點紀錄下來，若是有前後兩筆資料變動太大時，系統會將其視為無效的資料，由圖 5.6 可以看出當機器人在固定位置時，所量測出來的中心點幾乎保持不變，顯現出本論文所設計的目標物辨識方法，在實際運作時，有優良的表現。由於量測出來的中心點有穩定的結果，圖 5.7 中，估測出來的座標呈穩定的結果，沒有太大的變動。將上面所量測到穩定的座標位置，當作機械手臂的控制器輸入，移動手爪到目標物前方，將目標物抓起，過程中並透過即時的估測，修正手爪應該移動到的位置。圖 5.8 可以看出手臂抓取過程的軌跡，紅色為起點到預抓取點間的路徑，而藍色則為最後抓取路徑的規劃。

實際機器人移動抓取時的情形則如圖 5.9(a)-(e)所示。當攝影機估測出目標物的位置之後，算出手臂起點到預抓取點的位置之間的路徑，手臂順著這個軌跡

移動到預抓取點(a)-(d)，(d)-(e)則是將手爪移動到讓目標物位在夾爪中心的地方，最後將夾爪闔上舉起目標物。



圖 5.5 平台靜止時，攝影機所擷取的畫面及目標物位置的計算結果

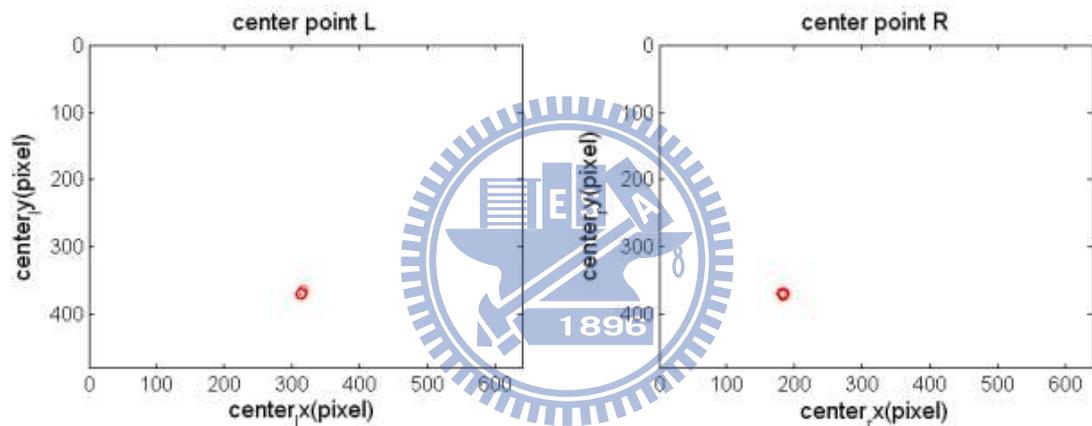


圖 5.6 量測出之目標物在影像平面上的中心點

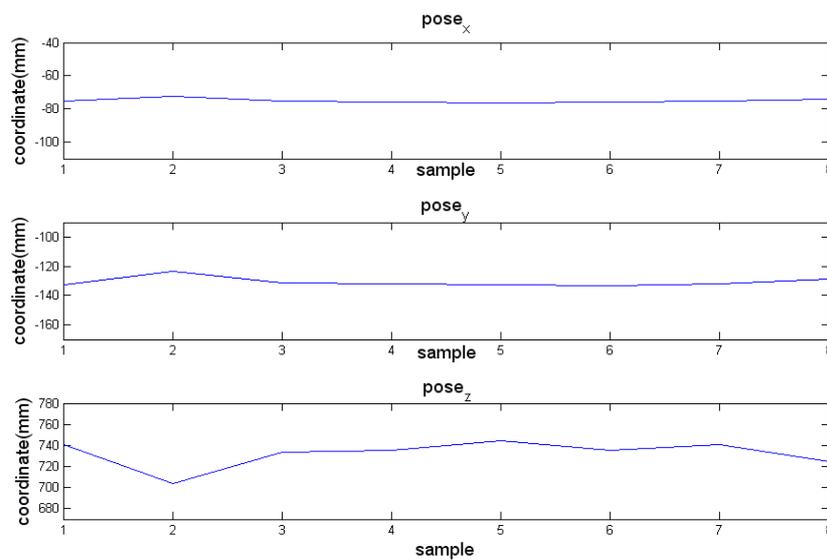


圖 5.7 攝影機量測到目標物的座標位置

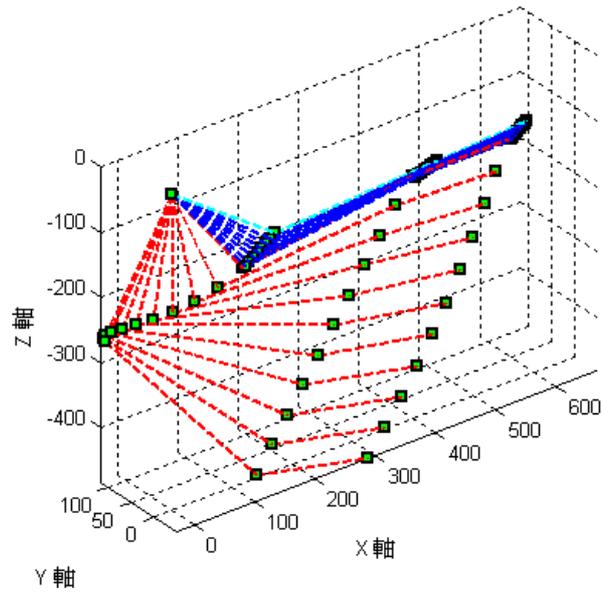


圖 5.8 抓取過程中手臂各軸軌跡圖

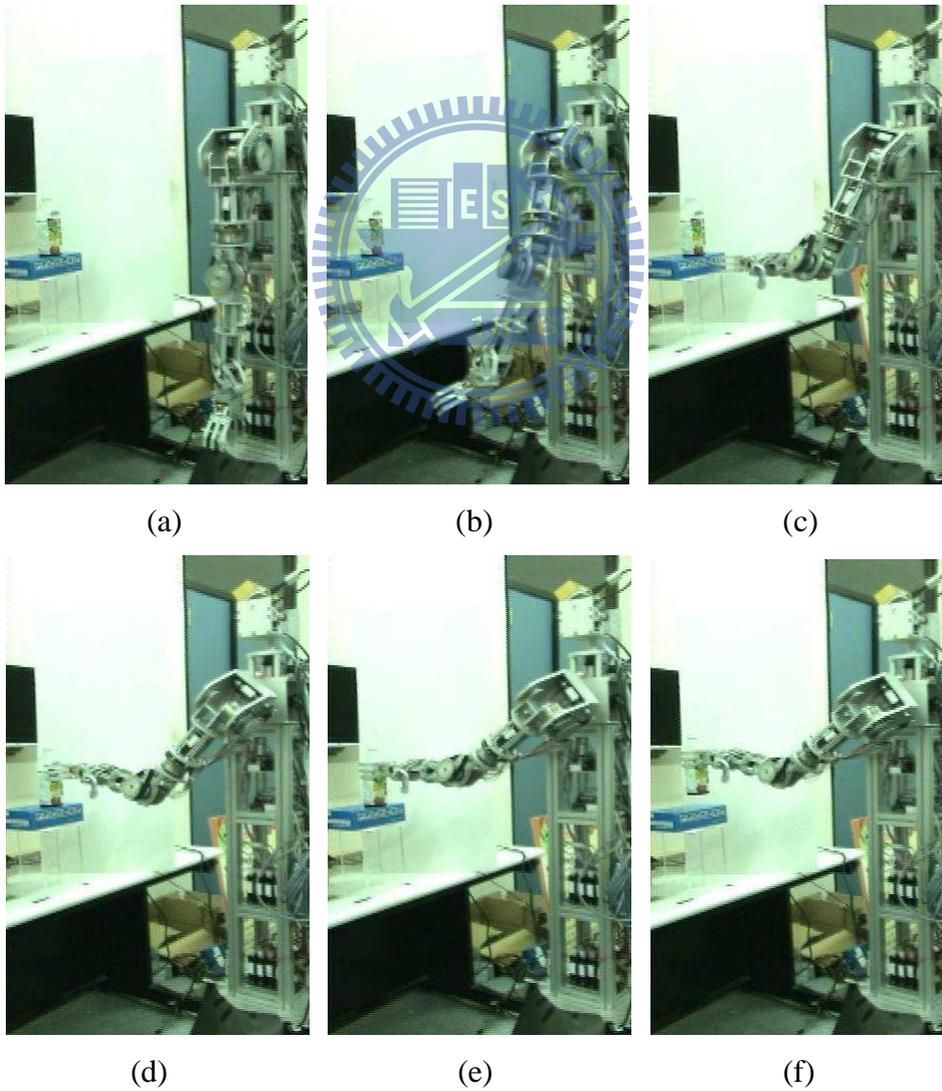


圖 5.9 平台靜止抓取實驗

### 5.3 平台移動抓取實驗

在本實驗中，我們一樣在機器人前方放置一個瓶子，作為抓取的目標物，而這次瓶子放置在離機器人約 93 公分處，離地高度為 105 公分，這次的實驗主要目的在於驗證，當機器人判斷目標物所在位置超出機械手臂工作範圍時，他會控制移動平台向前移動到機械手臂的工作範圍之內，再進行最後的抓取任務。

將整個實驗過程中，從開始到完成抓取動作，所計算出目標物的中心點在影像平面上的位置記錄下來。在起點時攝影機擷取到的畫面如圖 5.10，此時計算出目標物的中心點位置如圖 5.11，是在整個影像平面上半部的位置。由於此時算出機器人跟目標物的距離超出手臂工作範圍，於是計算出要移動的距離之後，控制平台往前移動。到達工作範圍後，攝影機所截取到的畫面如圖 5.12，此時紀錄的中心點位置如圖 5.13，中心點在影像平面上的位置向下移動。由於攝影機傾斜的角度固定，機器人在移動的過程中，中心點在影像平面上的位置也有所變化，目標物在影像平面上的位置隨著距離的減少，y 值有所變動，而 x 值則變化不大，顯示出機器人是往正前方移動。在座標量測方面，隨著機器人的移動，目標物在影像平面上的位置有所變動，大致分為兩個區間，分別是在起點時，及到達工作範圍之後，所以在圖 5.14 中數據有一個跳動，就是對應到當時所計算出的目標物中心點位置。圖 5.15 則是手臂移動的軌跡。



圖 5.10 移動抓取實驗中，起點時攝影機所擷取的畫面及目標物位置

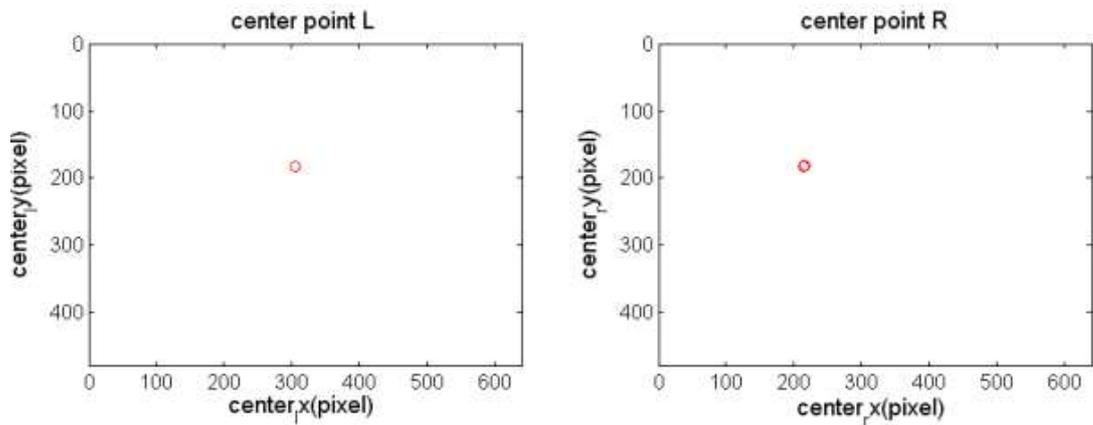


圖 5.11 攝影機在起點所量測出之目標物在影像平面上的中心點



圖 5.12 移動抓取實驗中，到達工作範圍時攝影機所擷取的畫面及目標物位置

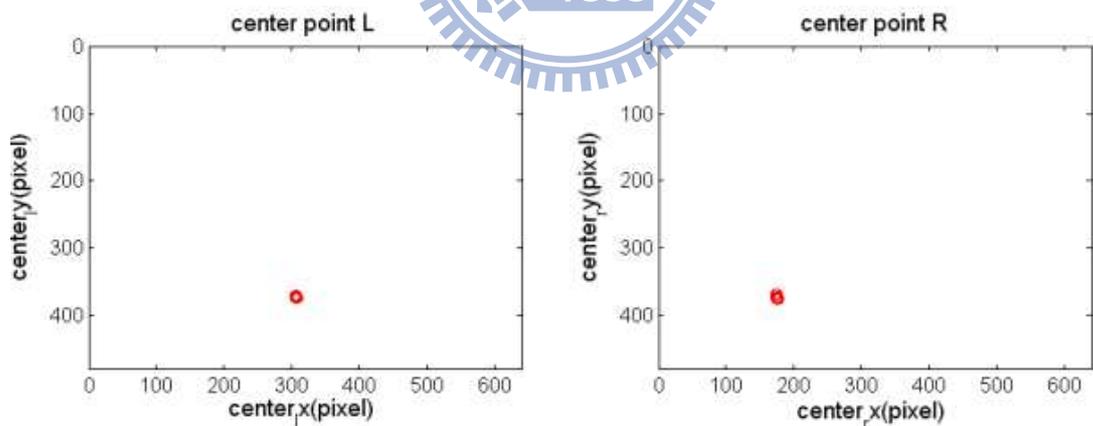


圖 5.13 攝影機在工作範圍所量測出之目標物在影像平面上的中心點

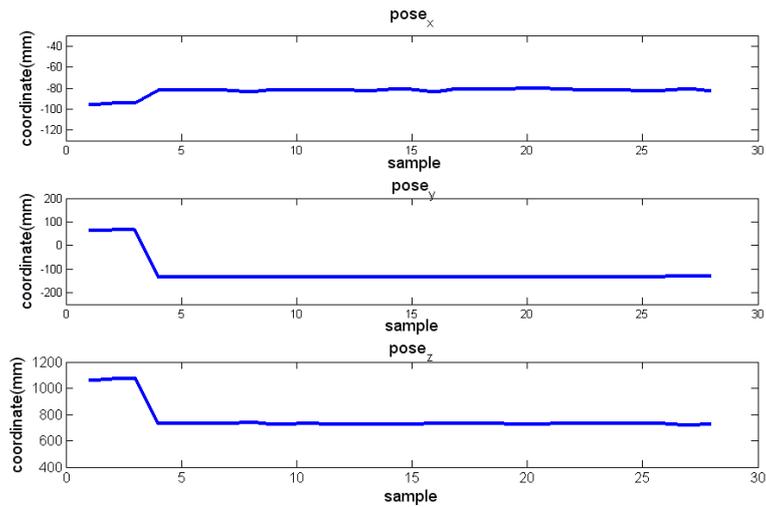


圖 5.14 攝影機量測到的目標物座標位置

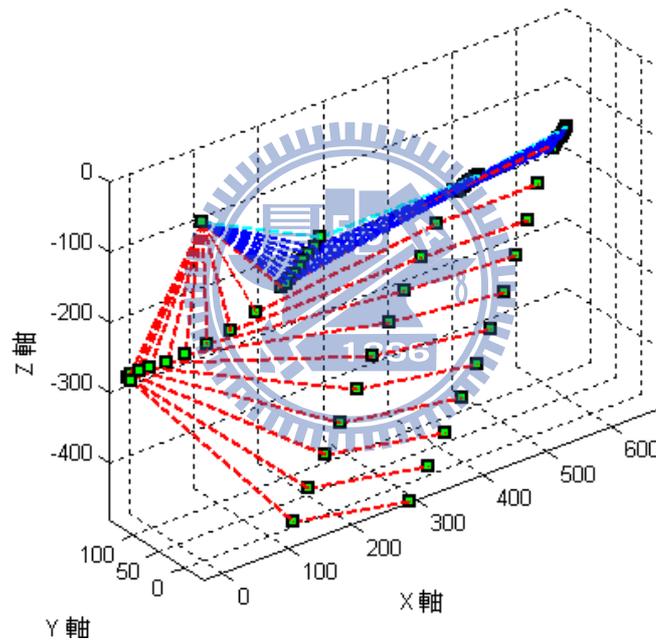


圖 5.15 抓取過程中手臂各軸軌跡圖

實際機器人移動抓取時的情形則如圖 5.16 (a)-(i)所示，(a)-(d)為機器人移動的過程，在機器人判讀尚未到達手臂的工作範圍之前，控制器會持續的下命令給平台，使其移動，透過影像資訊的輸入，再即時的計算是否到達工作範圍，到達後平台才會停止。根據估測出來的目標物位置，來判斷機器人移動的距離，在本實驗中，目標物所對應的適當工作範圍大致在深度為 600mm 的距離之內，一開始攝影機估測出距離大於 600mm 所以根據目前位置到工作範圍之間的距離，下命令給平台讓他移動這個距離，使機器人到達手臂的工作範圍之內。



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)

圖 5.16 平台移動抓取實驗

其移動的軌跡紀錄如圖 5.17 所示，由於我們將世界座標原點定在機器人本體，所以畫出來的移動軌跡圖，為一個相對的座標關係。由距離紀錄可以得知，機器人在出發點所量測到的深度是 932mm，於是機器人應該向前移動的距離應該是 332mm，其中移動的誤差可以根據攝影機計算的結果再作調整，根據移動軌跡顯示機器人最後停在距離目標物 631mm 的地方，表示已經到達工作範圍可以執行最後抓取的任務，而 X 方向在前進的過程中略有偏差，不過由於機器人已經判斷在工作範圍內，所以不必再進行左右的修正。

圖 5.17 中，藍色的圓點是目標物的位置，紅色的線是移動的軌跡，因為目標物跟機器人彼此的座標位置為一相對的關係，所以在目標物不動的情況之下，隨著機器人移動，攝影機觀測到的目標物位置會有所變動。在移動過程中 Z 座標值會漸漸變小，而 X 跟 Y 座標則是根據中心點位置算出對應的結果，當機器人到達工作範圍後，所觀測到的中心點位置就會趨於穩定，算出來的座標值就不會再有劇烈的變動。

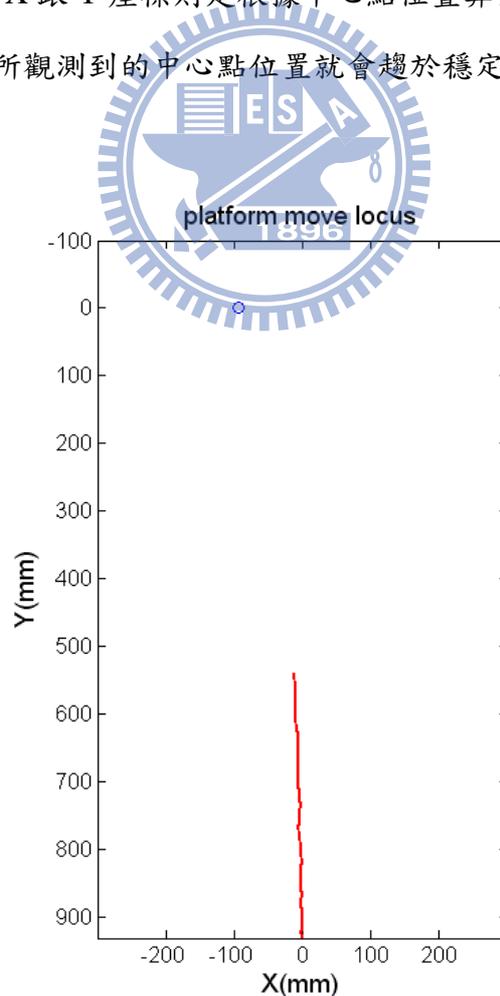


圖 5.17 平台移動軌跡紀錄

## 第六章結論與未來展望

### 6.1 結論

本論文藉由 SURF 演算法，使機器人能透過目標物上的自然特徵點資訊來定義該物體，並且加入了平面轉換矩陣(homography)的觀念及 RANSAC 剔除錯誤比對，經由這幾種方法的結合讓我們能夠得到更穩定的對應特徵點。在此基礎之下，搭配本論文提出的座標估測方法，經由推導出在不同距離時座標估測的誤差，找出其誤差修正方程式，以便於機器人在不同距離時都能準確量測出目標物的位置。這些方法的結合讓機器人在任務規劃上更加的實用，針對目前的座標位置相對關係，來決定機器人的運動策略，是否需要移動，或是已經可以移動手臂過去抓取，都能夠靈活的規畫。在機器人影像伺服控制方面，採用 PBVS 的架構，由裝置在機器人頭部的立體攝影機擷取環境中的資訊作為機器來移動的資訊來源，相對於 Eye-in-hand 的攝影機架構，在實際應用上更加的廣泛，在完成抓取之後，攝影機還能繼續擷取資訊，作為規劃接續任務的依據。

在實驗的驗證之下，我們確認此架構能夠順利的導引機器人至目標物前方，並藉由最後手爪跟目標物之間的幾何關係，求出抓取的軌跡，完成抓取的任務。

### 6.2 未來展望

本論文在影像辨識方面，雖然設計了一套由特徵點辨識乃至於錯誤特徵點剔除及求解平面轉換矩陣，完整的一套辨識環境中物體的方法，但是其中還有幾個可以改良的地方。在運算速度方面，由於是採用雙眼視覺，所以必須同時做兩張影像的特徵點比對，如此大幅增加了運算量，導致運算時間增加，另外，因為環境中光線的因素，常常會導致擷取的影像亮度不穩定，而使特徵點的計算及轉換矩陣的求解上出現較大的誤差。因此若能針對這些問題做出改良，將可提高整體的實用性，增加機器人對環境的反應能力。

在機器人的抓取方面，目前是針對直立的物體去做夾爪的控制，若是能夠針對手掌這部分設計出一個擬人化的手掌，在抓取的目標物方面就能夠更多樣化，

目標物的姿態也不必拘泥在直立的狀況，可以針對不同的姿態，改變手掌抓取的角度，用途更加廣泛。

在機器人的應用方面，目前可以完成到自主抓取環境中指定物品的任務，若能規劃在拿到物品後的任務，例如使用者在指定機器人抓取物品之後，再拿給使用者。如此一來，將可以大大提升人類生活的便利性及互動性，使其成為家中的好幫手。此外，由於頭部機構的設置以及雙臂的設計，可以研究兩手臂間的協調合作，完成更多的任務。



## 參考文獻

- [1] D. G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, 2004, pp.91-110.
- [2] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," in *Proc. of the 9th European Conf. on Computer Vision*, Graz Austria, 2006, pp. 404-417.
- [3] Sukhan Lee, Eunyoung Kim and Yeonchool Park, "3D Object Recognition Using Multiple Features for Robotic Manipulation," in *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, 2006, pp. 3768-3774.
- [4] Han-Young Jang, Moradi Hadi, Suyeon Hong, Sukhan Lee, and JungHyun Hong, "Spatial Reasoning for Real-time Robotic Manipulation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2632-2637.
- [5] Sukhan Lee, Daesik Jang, Eunyoung Kim, Suyeon Hong and JungHyun Hong, "A Real-Time 3D Workspace Modeling with Stereo Camera," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, 2005, pp. 2140-2147.
- [6] S. C. Hung and M. F. Li, "A SIFT Method Application for a Six DOF Manipulator Grasping an Object Against Complicated Environment," in *the 33rd Annual Conference of the IEEE Industrial Electronics Society*, Taipei, Taiwan, 2007, pp. 465-470.
- [7] David Gossow, Johannes PellenZ and Dietrich Paulus, "Danger Sign Detection Using Color Histograms and SURF Matching," in *Proceedings of IEEE International Workshop on Safety, Security and Rescue Robotics*, Sendai,

- Japan, 2008, pp. 13-18.
- [8] S. H. Baeg, J. H. Park, J. Koh, K. W. Park and M. H. Baeg, "An Object Recognition System for a Smart Home Environment on the Basis of Color and Texture Descriptors," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 901-906.
- [9] M. Aiguo, X. Zhaoxian, T. Yoshida, M. Yamashiro, T. Chao and M. Shimojo, "Home Service by a Mobile Manipulator System -System Configuration and Basic Experiments," in *Proceedings of IEEE International Conference on Information and Automation*, Zhangjiajie, China, 2008, pp. 464-469.
- [10] Changhyun Choi, Seung-Min Baek and Sukhan Lee, "Real-time 3D Object Pose Estimation and Tracking for Natural Landmark Based Visual Servo," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 3983-3989.
- [11] M. Tomono, "3-D Object Map Building Using Dense Object Models with SIFT-based Recognition Features," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 1885-1890.
- [12] K. Yamazaki, M. Tomono, T. Tsubouchi and S. Yuta, "A Grasp Planning for Picking up an Unknown Object for a Mobile Manipulator," in *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, 2006, pp. 2143-2149.
- [13] Y. Hirata, Y. Matsuda and K. Kosuge, "Handling of an Object in 3-D Space by Multiple Mobile Manipulators Based on Intentional Force/Moment Applied by Human," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Zurich, 2007, pp. 1-6.

- [14] H. I. Christensen and P. Case, "Mobile Manipulation for Everyday Environments," in *Proceedings of IEEE International Conference on Control, Automation and Systems*, COEX, Seoul, Korea, 2008, pp. xlii-xlvi.
- [15] M. Prats, P. Martinet, A. P. del Pobil and Sukhan Lee, "VIision Force Control in Task-Oriented Grasping and Manipulation," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 1320-1325.
- [16] T. Grundmann, R. Eidenberger, R. D. Zoellner, X. Zhixing, S. Ruehl, J. M. Zoellner, R. Dillmann, J. Kuehnle and A. Verl, "Integration of 6D Object Localization and Obstacle Detection for Collision Free Robotic Manipulation," in *Proceedings of IEEE International Symposium on System Integration*, Japan, 2008, pp. 66-71.
- [17] P. Azad, T. Asfour and R. Dillmann, "Stereo-Based 6D Object Localization for Grasping with Humanoid Robot Systems," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 919-924.
- [18] Iván F. Mondragón, Pascual Campoy, Carol Martínez and Miguel A. Olivares-Méndez, "3D Pose Estimation Based on Planar Object Tracking for UAVs Control," in *Proceedings of IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, 2010, pp. 35-41.
- [19] S. Benhimane and E. Malis, "Homography-based 2D Visual Tracking and Servoing," *International Journal of Robotics Research*, 2007, pp.661-676.
- [20] Viola, P., Jones, M., "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, USA, 2001, vol. 1, pp.511-518.

- [21] M. Fischler and R. Bolles, "Random Sampling Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Commun. Assoc. Comp. Mach.*, 1981, vol. 24, pp. 381-395.
- [22] <http://en.wikipedia.org/wiki/RANSAC>
- [23] K.S. Fu, R.C. Gonzalez, C.S.G. Lee, *Robotics Control, Sensing, Vision, and Intelligence*, McGraw-Hill Book Company, 1987..
- [24] 王兆戊, "全向式移動機器人之同步定位與環境地圖建立" 國立交通大學電機與控制工程學系碩士論文, 2008.
- [25] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing," in *IEEE Transactions on Robotics and Automation*, 1996, Vol. 12, NO. 5, pp. 684-696.
- [26] Geoffrey Taylor and Lindsay Kleeman, *Visual Perception and Robotic Manipulation*, Springer, 2006.
- [27] S. Hutchinson, G.D. Hager and P.I. Corke, "A Tutorial on Visual Servo Control," in *IEEE Transactions on Robotics and Automation*, 1996, Vol. 12, No. 5, pp. 651-670.
- [28] 陳弘哲, "基於影像辨識之移動式機械臂抓物控制設計" 國立交通大學電機與控制工程學系碩士論文, 2008.
- [29] <http://www.ptgrey.com/products/bumblebee2/>