

國立交通大學

電機與控制工程學系

碩士論文

移動式機械臂之抓物控制設計

Grasping Control of a Mobile Manipulator



研究生：葉忠憲

指導教授：宋開泰 博士

中華民國九十五年八月

移動式機械臂之抓物控制設計

Grasping Control of a Mobile Manipulator

研究生：葉忠憲

Student: Chung-Hsien Yeh

指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song



**A Thesis**

**Submitted to Department of Electrical and 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 and Control Engineering**

**July 2006**

**Hsinchu, Taiwan, Republic of China**

中華民國九十五年八月

# 移動式機械臂之抓物控制設計

學生:葉忠憲

指導教授:宋開泰 博士

國立交通大學電機與控制工程學系

## 摘要

本論文的目的是在設計一移動式機械臂的抓物控制系統，藉由機械臂上攝影機之影像資訊，迴授控制移動式機器人本體與機械臂的速度，以達成自動抓取的功能。本文基於 Image Jacobian 及機器人與攝影機間的 Jacobian 關係，設計出一個基於影像的視覺伺服控制器。此控制器可同時產生機器人本體與手臂的速度命令以追蹤空間中之物體特徵。在影像特徵擷取方面，本文利用一套嵌入式影像處理平台，採用 Eye-in-hand 的攝影機架設方式，以 Pattern matching 的方法尋找目標物體的角點以作為伺服控制器的輸入影像特徵點。文中以目標影像中兩種不同的角點作為視覺伺服控制器的輸入，設計機器人接近目標物體的行為。在控制設計上採用 Behavior-based 控制架構來結合兩種接近行為與一個抓取行為，可引導移動式機械臂到達目標物所在地點，進而以單自由度之手爪抓取目標物。透過實驗驗證，此設計可導引一部擁有 4 自由度的移動式機械臂成功抓取特定之目標物。

# Grasping Control of a Mobile Manipulator

Student: Chung-Hsien Yeh

Advisor: Dr. Kai-Tai Song

Department of Electrical and Control Engineering  
National Chiao Tung University

## ABSTRACT

The objective of this study is to design a grasping controller for a mobile manipulator. By controlling the velocity of the mobile robot and the 2-DOF robot arm simultaneously, the grasping of an object is realized. A visual servo controller has been realized by adopting the concept of image Jacobian, and the Jacobian of mobile manipulator and camera. This controller determines the robot body and robot arm's velocity commands. We define the corners of the target image as the feature points which will be the input of visual servo controller. An embedded vision system is utilized to extract these feature points by pattern matching method. With two kinds of feature points on the target image and the visual servo controller, we design two approaching behaviors which will be activated when target is near or far away from camera, respectively. Combining two approaching behaviors and a grasping behavior, we use a behavior-based control architecture to realize an autonomous grasping behavior of a mobile manipulator. With this design, the 4-DOF mobile manipulator can find and grasp a target successfully.

# 目錄

中文摘要.....	i
英文摘要.....	ii
目錄.....	iii
圖例.....	v
表例.....	vii
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 相關研究回顧.....	4
1.3 問題描述.....	7
1.4 章節說明.....	7
第二章 影像伺服設計.....	8
2.1 視覺伺服控制.....	8
2.2 特徵空間控制法則設計.....	11
2.2.1. 設計概念.....	11
2.2.2. Image Jacobian.....	12
2.2.3. 機器人(Mobile manipulator) Jacobian.....	16
2.2.4. 座標轉換.....	22
2.2.5. 分解運動速率控制(Resolved motion rate control).....	23
2.2.6. 控制器設計.....	24
2.2.7. 奇異點分析.....	25
2.3 本章總結.....	25
第三章 影像特徵點擷取.....	26
3.1 對抓取目標物體的基本假設.....	26
3.2 影像特徵定義.....	27

3.3	影像特徵擷取.....	30
3.3.1	遠端特徵點.....	31
3.3.2	近端特徵點.....	32
3.4	目標距離估測.....	34
3.5	攝影機參數的校正.....	35
第四章	模擬與實驗結果.....	38
4.1	硬體介紹.....	38
4.2	視覺伺服模擬與實驗.....	41
4.2.1	模擬結果.....	41
4.2.2	實驗結果.....	44
4.3	抓取靜止目標物實驗.....	45
4.4	抓取移動目標物實驗.....	49
第五章	結論與未來展望.....	54
5.1	結論.....	54
5.2	未來展望.....	54
	參考文獻.....	56



## 圖例

圖 1.1	交通大學電控系實驗室之機器人: Easybot.....	3
圖 1.2	Pan-tilt camera and 1-DOF robot arm .....	5
圖 1.3	Mobile robot with dual arm .....	5
圖 2.1	Position-based visual servo.....	8
圖 2.2	Image-based visual servo.....	8
圖 2.3	(a) 機器人側視圖；(b) Eye-in-hand camera mounting；(c) 機器人上視圖	10
圖 2.4	Perspective projection.....	13
圖 2.5	機器人上的座標系與各項參數.....	17
圖 2.6	手臂上 D-H model 與各個座標系 .....	17
圖 2.7	機器人之奇異點.....	25
圖 3.1	不同視角下的目標物體.....	27
圖 3.2	目標物體非垂直地面：(a) 物體相對於手爪有滾動(Roll)；(b) 物體相對於手爪有傾斜(Tilt) .....	28
圖 3.3	目標影像樣板.....	29
圖 3.4	(a) 目標物體； (b) 目標影像 .....	29
圖 3.5	影像特徵點：(a) 遠端特徵點； (b) 近端特徵點 .....	30
圖 3.6	(a) Top-left pattern； (b) Bottom-right pattern.....	31
圖 3.7	遠端特徵點檢驗：(a) 垂直掃描； (b) 長寬比 .....	32
圖 3.8	近端特徵點檢驗：(a) 垂直掃描； (b) 長寬比 .....	33
圖 3.9	影像特徵擷取流程圖.....	34
圖 3.10	距離估測原理.....	35
圖 4.1	嵌入式影像處理平台.....	39
圖 4.2	機器人硬體架構圖.....	40
圖 4.3	IBVS 伺服架構 .....	41

圖 4.4	視覺伺服模擬結果 1:(a) 影像特徵點移動軌跡;(b) 機器人本體角速度、 手臂兩軸之角速度;(c) 機器人移動軌跡;(d) 機器人本體朝向角、手臂 兩軸角度.....	42
圖 4.5	視覺伺服模擬結果 2:(a) 影像特徵點移動軌跡;(b) 機器人本體角速度、 手臂兩軸之角速度;(c) 機器人移動軌跡;(d) 機器人本體朝向角、手臂 兩軸角度.....	43
圖 4.6	機器人初始位置.....	44
圖 4.7	視覺伺服實驗結果:(a) 影像特徵點移動軌跡;(b) 特徵點距離估測結 果;(c) 機器人移動軌跡;(d) 手臂兩軸角度、攝影機 tilt 角度.....	45
圖 4.8	抓取行為控制架構.....	46
圖 4.9	特徵擷取過程的影像與所得結果 – 只擷取到遠端特徵點.....	47
圖 4.10	特徵擷取過程的影像與所得結果 – 同時擷取到遠端與近端特徵點....	47
圖 4.11	特徵擷取過程的影像與所得結果 – 只擷取到近端特徵點.....	48
圖 4.12	抓取靜止目標實驗.....	50
圖 4.13	抓取靜止目標物實驗結果:(a) 特徵點移動軌跡;(b) 目標距離估測結 果;(c) 機器人移動軌跡;(d) 攝影機移動軌跡;(e) 手臂兩軸角度、攝 影機 tilt 角度.....	51
圖 4.14	抓取移動目標物實驗.....	52
圖 4.15	抓取移動目標物實驗結果:(a) 目標距離估測結果;(b) 機器人移動軌 跡;(c) 攝影機移動軌跡;(d) 手臂兩軸角度、攝影機 tilt 角度.....	53
圖 5.1	整合避障與導航功能.....	54

## 表例

表 2.1	Kinematics parameters of robot and arm.....	18
表 3.1	遠端特徵點之距離估測結果.....	36
表 3.2	近端特徵點之距離估測結果.....	36
表 3.3	攝影機參數.....	37



# 第一章 緒論

## 1.1 研究動機

近幾年來，隨著各種感測器與電腦運算能力的快速進步，過去因為需要許多複雜運算而無法達到即時性的控制方法，現在已能藉由一般的電腦硬體來完成。相較於過去傳統工業用的固定式機械臂，移動式機器人相關的研究也越來越多，各種服務型機器人產品也逐漸出現在市場上，較常見的有吸塵機器人(如 iRobot 所出產的 Roomba)、寵物機器人(如 Sony 出產的 AIBO)與家用機器人(如 Mitsubishi 出產的 Wakamaru)。另外人形機器人如 ASIMO(由日本 Honda 所研發)，能夠和人類一樣的步行、閃避障礙物、拿取物品，並配備視覺系統能與人類互動，讓人了解到機器人領域的發展已進入新的世代。

現今由於醫療科技的進步，人類壽命的延長，造成老年人口比例上升，使得家用與看護機器人等服務型機器人更形重要。對於服務型的機器人而言，其工作環境通常就是一般人的生活環境，且無可避免地，服務型機器人必須能和人類互動，這也是其最主要的功能之一。要讓機器人與人類互動，一般常見的方法有語音介面及在機器人上設置螢幕來提供人機介面的功能。另一方面也可以透過在機器人上之機械手臂，使得機器人能有抓取物品或與人類合作搬運物品的能力。

基於以上所述，裝置在移動式平台上的移動式機械臂相較於固定式機械臂有更多的應用機會，但衍生出許多控制問題。傳統固定式機械臂的工作環境通常都在工廠中的固定位置，並且經過特殊規劃，因此環境的因素較為單純；但是加上移動平台後工作環境範圍變大，環境因素也相對變得較為複雜。尤其是服務型機器人等需要與一般人互動的應用，工作環境中不但有家具等固定式的障礙物，在環境中走動的人類也是一種移動式的障礙物。

除了障礙物的問題，將機械臂裝置在移動式平台上通常會遇到贅餘自由度(Redundancy)的問題，當移動平台和機械臂的自由度總合超過所要執行的工作所

需的自由度時，就會遇到贅餘自由度的問題。而對移動式機械臂控制最重要的問題莫過於如何協調手臂與移動平台的動作，以到達可抓取到目標的姿勢(Posture)。這方面的控制主要可以分為兩種，一種是先估測目標的位置，再規劃出 End-effector 到達目標的空間路徑(Path planning)，最後讓移動式機械臂追蹤路徑(Path tracking)到達目標的所在位置[3]；另一種是直接經由感測器所測得的目標資訊，隨時調整移動式機械臂的姿態，最後到達可抓取目標的位置[1, 2, 4]。第一種控制方式在移動前事先規劃好路徑，因此其優點在於可以針對各種需要規劃最佳的路徑，例如機器人動作最少的路徑、可維持機械臂最佳可操作性(Manipulability)的路徑等；第二種方式的優點是可以針對環境的變化來即時修正機器人的動作。對於需要與人互動的機器人應用，其工作環境通常是會變動的，因此第二種控制方式會較為合適。

對於服務性機器人而言，需要抓取的物品種類相對於工業用的機器人增加許多。一般狀況下，攝影機(CMOS 或 CCD)所取得的影像具有高密度資訊，配合各種影像處理方法，可以從影像中取得各種資訊，例如用影像辨識技術來識別欲抓取的目標，或利用光流法估測與前方障礙物的距離[1]等。也因此，攝影機常被用來做為移動式機械臂所使用的感測器，衍生出很多相關研究。如何利用影像資訊來控制移動式機器人到達並抓取特定的目標，是機器人領域重要的研究議題之一，也是本篇論文所要探討的主要對象。

交通大學電控所的 Henry Andrian 針對機器人應用開發了一套嵌入式影像處理系統[1]，並利用此系統在實驗室的機器人 Easybot 上實現過物品抓取的功能。Easybot 是交通大學實驗室中專為個人助理方面的應用所開發的機器人之一，其主要硬體構造如圖 1.1 所示，Easybot 本體為一部由 2 輪驅動的移動式平台，並搭載 3-DOF(含 Gripper)機械臂，用來做為與人或環境互動的工具。在 Andrian 的研究中，主要是設計一套嵌入式影像平台，並希望在此平台上實作自主抓取功能，讓機器人能在環境中搜尋固定目標，同時閃避前方的障礙物，當找到目標後能準確抓取物品。Andrian 利用 Behavior-based subsumption 架構來設計機器人抓



圖 1.1 交通大學電控系實驗室之機器人: Easybot

取功能，但是在設計上，原系統中並沒有考慮手臂兩軸的控制，也就是說，機器人的手臂兩軸的角度是固定的，只能抓取固定高度的物體。這對於抓取的功能來說是相當大的限制，也降低了抓取功能的實用性。

為了讓機器人的手臂能抓取不同高度的物體，我們需要能控制手臂移動的抓取行為控制系統。由於 Andrian 所設計的抓取和避障功能都是利用同一套嵌入式影像平台來完成，在避障部分是應用混合式光流法(Mixed optical flow)計算機器人與障礙物的碰撞時間(Time-To-Collision)，以此時間和機器人當時前進速度來估算與障礙物間的距離，建立安全分布直方圖(Safe distribution histogram)，判斷前方是否有障礙物存在，並決定閃避方向。此避障功能在影像平台上每秒鐘約可處理 5 張影像，若加上抓取功能所需的影像辨識系統，將會使處理速度更慢。且在方法上，利用此種避障方法是假設攝影機只有平行地面水平速度，若手臂會上下移動，攝影機將同時具有水平和垂直方向的速度，不適用此種避障方式。因此，本論文將針對實驗室的移動式機械臂設計一個視覺伺服控制器，利用目標物體在影像中的特徵點位置作為控制器的輸入可導引機器人到達可抓到目標的姿態，達成抓取功能的設計。機器人避障的功能目前實驗室已有利用超音波避障的設計，在抓取功能完成後，可以和超音波避障做整合。

在抓取功能的設計上，我們將採用 Eye-in-hand 的攝影機設置，主要考量是

能避免目標影像被本身的手臂遮蔽，無法拍攝到完整的目標影像。基於 Eye-in-hand 設置，當機器人前進時，目標影像可能會離開攝影機視野範圍，因此攝影機也要跟著移動，將目標保持在視野範圍內，這表示手臂部分也要跟著移動。因此，在設計抓取功能時，我們必須同時控制機器人本體與手臂兩者的動作，才能讓機器人在接近目標的同時，也能讓目標保持在視野範圍內，並且最後手臂會位於能抓取的姿態。

在抓取行為的設計中，為了協調機器人本身的各種基本行為以應付環境的變動，並增加與其他機器人功能的整合性(如避障功能)，我們需要一個能利用感測資訊對環境做快速反應，並且具有和其他機器人行為整合彈性的控制架構，因此本論文將使用 Behavior-based 架構來協調機器人的行為。

## 1.2 相關研究回顧



在移動式機械臂的運動控制領域，近年來大部分的研究主要集中在如何控制移動式機械臂在一個指定的路徑上移動；其次是如何讓移動式機械臂到達特定的地點並維持某個指定的 pose 或維持機械臂的 Manipulability，使機械臂到達定點後能有足夠的靈活度執行所需的工作；另外還有許多相關研究。這裡僅對於利用影像導引的方式來完成抓取物品工作的研究做回顧。

在文獻[2]中，Martinez-Marin 和 Duckett 使用一個移動式平台搭載一個自由度的機械臂與一個 pan-tilt 攝影機(如圖 1.2)，其所要完成的工作為：抓取一個放置在桌子邊緣的鐵罐。他們將此工作分成兩個部分，docking 和 grasping，並將重點放在 docking 部分。文中作者利用 Reinforcement learning 的方法來控制移動式機械臂，當機械臂成功到達目標位置，則給予正獎賞(Positive reward)，若失敗則給予負獎賞。當學習完成後，可以依據感測器得到的資訊找到對應最大獎賞的機器人動作，也就是最適合當時環境的動作。作者利用機器人上裝設的 pan-tilt 攝影機追蹤目標影像並將影像維持在畫面中央，此時由攝影機的角度和



圖 1.2 Pan-tilt camera and 1-DOF robot arm [2]

桌面邊緣在畫面中的角度，可以推算出目標與機器人之間的相對位置。

Takahama 等利用移動式機器臂來完成特定動作：抓取放置在桌上的書本，並放置到書架上(如圖 1.3)[3]。此篇文章使用一個配備全向輪的移動平台，上面裝設了兩支機械臂，分別裝設 CCD 攝影機來取得環境資訊，並將所要完成的工作分成八個部分工作，採用循序執行的方式，利用兩支機械臂合作來完成歸還書本的動作。在此研究中，作者將重點放在機械臂的動作控制上，在移動平台到達抓取地點後，首先估測出書本的位置，再計算出此時兩支機械臂工作範圍判斷機械臂是否能抓取到書本，並考慮機械臂的可操作性(Manipulability)，規劃機械臂在空間中的移動軌跡(Path planning)，再讓機械臂依照此軌跡動作。

MacKenzie 和 Arkin 利用移動式機械臂完成對汽油筒內液體的取樣[4]。機器人會移動到一個汽油筒旁，利用裝設有攝影機的機械臂伸入油桶上方的孔內，

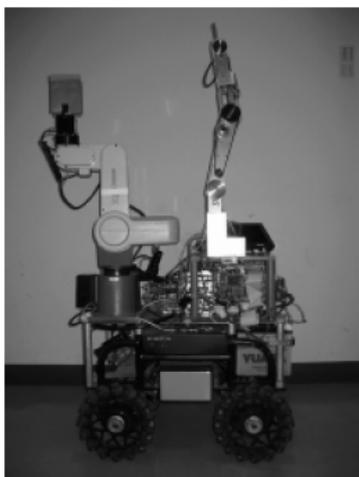


圖 1.3 Mobile robot with dual arm [3]

採樣後再移動到另一個汽油筒將樣本注入。機器人一開始先利用攝影機在影像中尋找事先定義好的汽油筒上的影像特徵，以確定汽油筒的方向，到達筒邊後機械臂則由上方尋找符合特徵的汽油筒開口。MacKenzie 及 Arkin 採用 Motor schema-based 的行為模式架構來協調機械臂與移動平台的行為，因此移動平台靠近汽油筒的行為，與機械臂對準汽油筒開口的行為可同時執行，達到 preshaping 的動作以縮短完成工作所需的時間。在他們的研究中，行為的協調是用 Finite state acceptor(FSA) 的方式循序執行。文獻[5]採用另外一種行為模式架構，即 subsumption 架構。其機器人所要完成的工作為在房間中遊走並尋找可樂罐，找到後將它撿起並帶到一個指定地點。文獻[6]提出一套行為協調的方法來控制在不同環境下，機器人的行為控制。作者透過 Bayesian belief network(BBN)來訓練不同的環境和目標擺放位置下，機器人行為之間的協調方式。使得訓練完成之後，機器人對於各種環境能有較合適的行為輸出。作者在訓練階段是在虛擬環境中實行，對於實際環境中存在的各種變化與雜訊是否依然適用，是需要進一步探討的問題。

在移動式機械臂抓取控制的相關文獻中，機器人行為的設計方式主要可分為兩種：一種是藉由分析機器人結構與所要執行的工作特性，藉以設計專用的控制器來執行所需的工作；另一種是利用類神經網路，透過訓練來得到能完成特定工作的控制器。由於第二種設計需要透過多次實驗累積資料，才能訓練出合適的控制器，訓練過程需要花費大量時間，且因為訓練用的資料直接影響到最後的控制器，資料的正確性相當重要，因此需要做大量的實驗，以免最後的控制器受到某些訓練資料的影像太大。相對之下，第一種設計方式需要對機器人的結構與環境做分析，如果在機器人相對上複雜度不高，此種設計方式將會比較有效率。由於本論文中所使用的機器人只有四個自由度需要控制，其控制維度相較於其他使用五~六自由度為手臂的移動式機械臂，結構較為簡單，因此可以透過第一種設計方式，藉由分析機器人的結構去設計適當的控制器。

### 1.3 問題描述

對移動式機械臂而言，最重要的就是經由機器人視覺導引其抓取物品的能力。在機器人找到目標物後，必須能準確的朝目標前進並到達能夠抓取目標物的位置。這部分同時牽涉到移動平台與機械臂的控制，因為移動平台與機械臂的移動均會影響到攝影機所取得的影像，如何同時控制機械臂與移動平台的動作，使機械臂能到達抓取位置是抓取行為設計中最重要問題。有些研究中將移動平台與機械臂的控制分開，如[2, 3, 11]，先讓移動平台移動到目標附近，再讓機械臂進行抓取，這樣做一方面是為了避開 Redundant 問題，另一方面是可將移動式機械臂視為傳統的固定式機械臂來控制，但是如此一來，移動平台和機械臂不能同時動作也使得抓取動作要花較多時間。另外，因為考慮機器人要在非固定的環境下工作，因此不使用估測物體姿態後規劃路徑的方式，而採用基於影像輸入隨時修正機器人動作的方式，以對環境或物體的變化能有快速的反應。因此，本篇論文希望能設計一套抓取控制系統，能利用當時攝影機取得的影像，在影像空間中控制機器人的動作，使移動平台和機械臂能互相協調，在抓取過程中同時動作，使移動平台最終能到達目標所在地點，同時機械臂也到達可抓取目標的姿態。

### 1.4 章節說明

本論文共分 5 章，第一章為緒論，介紹機器人抓取行為設計上可能遇到的問題。第二章介紹本論文所使用的特徵空間控制器，包含 Image Jacobian 的推導、Mobile manipulator Jacobian 的推導，與控制器設計。第三章介紹如何擷取控制器所需的影像特徵點。第四章為實驗與模擬結果，以驗證控制器的設計。第五章為結論與未來展望。

## 第二章 影像伺服設計

### 2.1 視覺伺服控制

抓取功能的目的是在於如何以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，也就是一個視覺伺服(Visual servo)問題。所謂的視覺伺服泛指所有利用視覺控制的機器人系統，然依照控制器輸入資訊的不同，目前較為常見的視覺伺服架構可分為兩類[7]，分別是基於位置的視覺伺服(Position-based visual servo, PBVS)(圖 2.1)和基於影像的視覺伺服(Image-based visual servo, IBVS)(圖 2.2)。

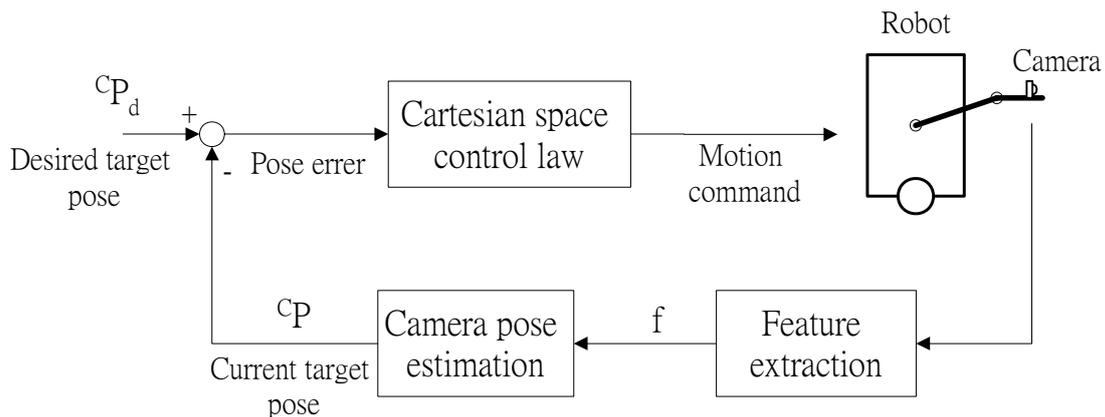


圖 2.1 Position-based visual servo

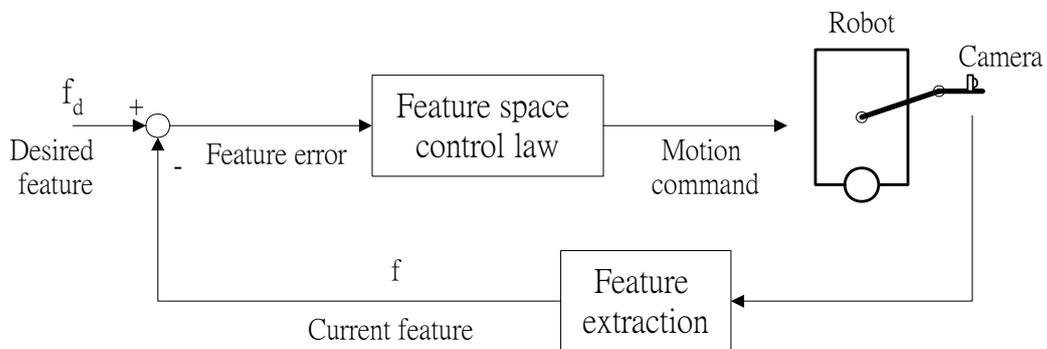


圖 2.2 Image-based visual servo

如圖 2.1 所示，在 PBVS 架構中，由攝影機取得影像後，先經過特徵擷取得到目標的特徵點，再利用特徵點套用在事先定義好的目標幾何模型來估測目標物體目前的姿態(Current target pose)。所謂的姿態(Pose)，包含有物體在卡氏座標系(如攝影機的座標系)中的位置與朝向角。有了目前的目標物體的姿態後，將目前的姿態和預定到達的姿態(Desired target pose)相減會得到目前的姿態誤差；利用此誤差和卡氏空間控制法則(Cartesian space control law)來產生機器人的控制命令，修正攝影機和目標物體的相對位置，使末端操作器(End-effector)最後到達預定的姿態。

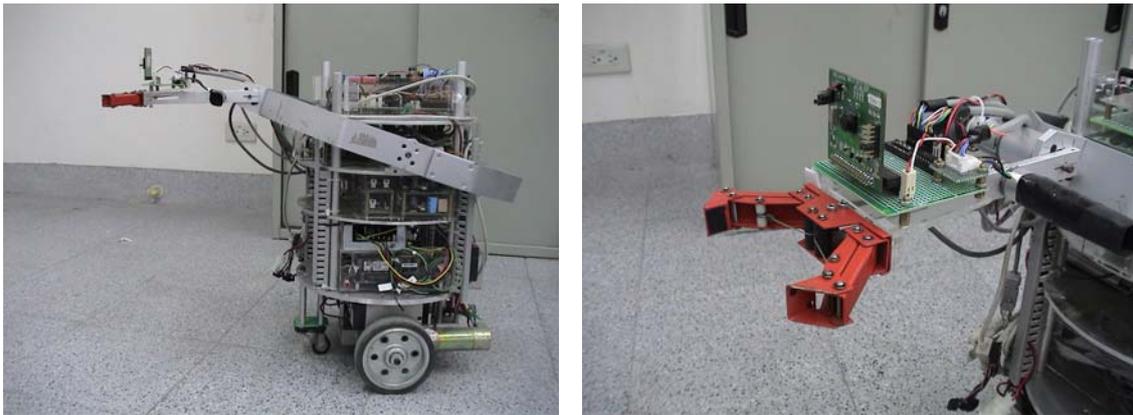
而 IBVS 架構，如圖 2.2 所示，在經過特徵擷取得到目前目標影像的特徵點位置後，直接利用目前的特徵點位置和預定到達的特徵點位置相減，得到兩者間的誤差；利用此誤差作為以特徵空間為輸入的控制器，產生所需的控制訊號，以修正攝影機與目標物體的相對位置。

在 PBVS 架構中，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，因此若用來估測的模型和實際的目標物體有誤差，可能會使系統無法到達預定的姿態。對 IBVS 架構來說，因為直接在影像特徵空間中做控制，沒有經過姿態估測的過程，因此不需要精確的目標幾何模型，也就沒有上述的問題。對於機器人抓取行為的設計來說，若所有要讓機器人抓取的目標物體都要事先建立好精確的幾何模型，會讓機器人顯得很不實用，因此在本論文中將採用基於影像的視覺伺服架構來設計移動式機器人的抓取行為。

在 IBVS 架構中，預定到達的特徵點位置(Desired feature position)的取得，通常是將機器人的末端操作器(End-effector)姿態設定為我們希望機器人最後到達的姿態時所擷取到的影像特徵點位置。在視覺伺服控制過程中，若影像的特徵點能到達預定的位置，理論上此時末端操作器也會到達我們先前取得預定特徵點位置時的姿態。

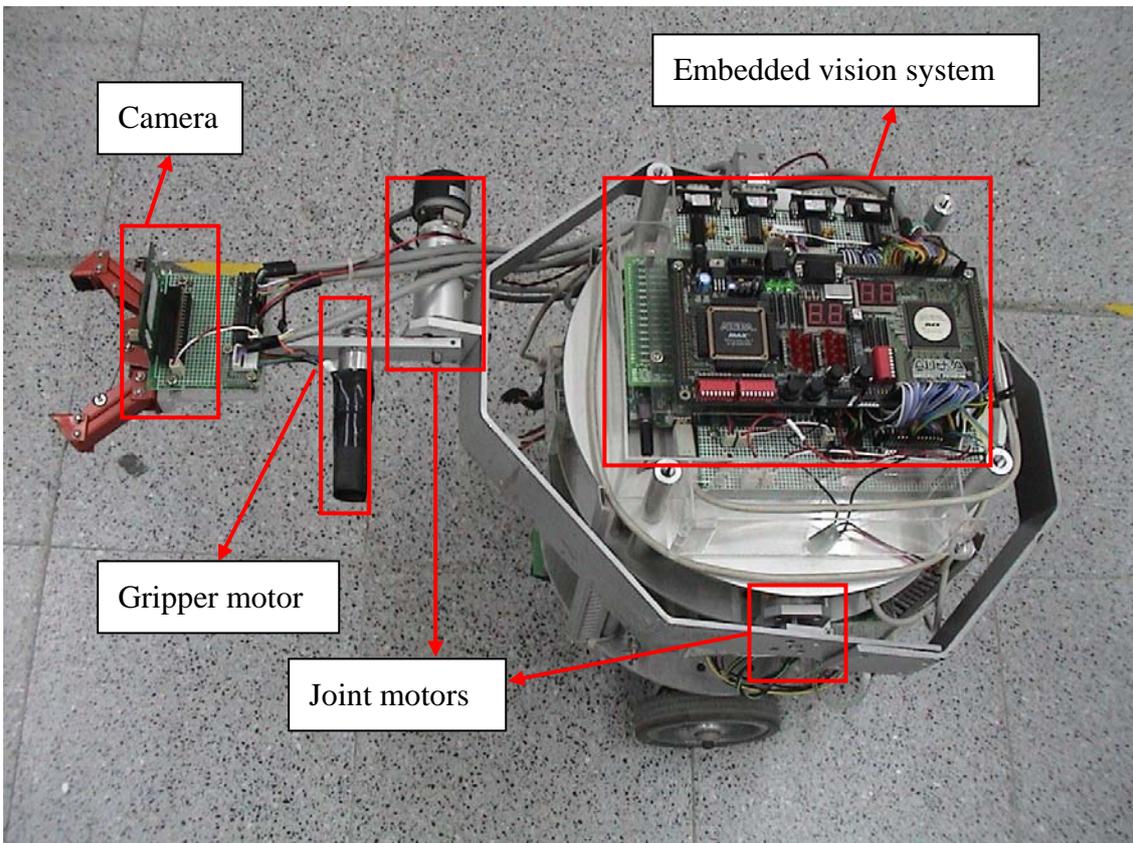
對於一個抓取工作，最後機器人的姿態應該位於可以抓到目標物體的位置，因此若我們採用 Eye-in-hand 的攝影機架設方式(圖 2.3(b))，並取得機器人手爪可

抓取到目標時的影像特徵點，則根據 IBVS 架構的特性，當我們在影像平面上到達此特徵點時，機器人手爪會到達當時取得特徵點的位置，也就是機器人可抓取到目標的位置。圖 2.3(a)(c)分別為本論文所使用的移動式機械臂外觀與其搭載的影像處理系統，在往後的章節中，我們將針對此移動式機械臂設計視覺伺服控制器，實現抓取目標物體的功能。



(a)

(b)



(c)

圖 2.3 (a) 機器人側視圖；(b) Eye-in-hand camera mounting；(c) 機器人上視圖

## 2.2 特徵空間控制法則設計

本章將介紹特徵空間控制法則的設計，所謂的特徵空間，意指由影像特徵點向量所構成的空間，而影像的特徵點可以是影像中的某些像素座標，或是影像中某兩像素間的長度等可以用來描述物體在影像中所形成的幾何形狀的參數。

### 2.2.1. 設計概念

如 2.1 節所述，在 IBVS 架構中，特徵空間控制法則藉由將目前的特徵點位置移到預定的特徵點位置來達成視覺伺服的目的，因此其輸入端為目前特徵點位置與預定特徵點位置的誤差，輸出為修正特徵點位置誤差所需機器人控制命令。由於我們的機器人所用的控制命令為馬達速度，因此我們利用分解運動速率控制 (Resolved motion rate control) 方法來設計控制器。分解運動速率控制為 IBVS 架構中，常被用來設計控制器的方法[7]，2.2.5 節將介紹分解運動速率控制與控制器的設計。

在本論文中，我們將用目標在影像平面的像素座標來作為特徵點，並利用分解運動速率控制的概念來設計控制。因為我們的機器人共有 4 個自由度(Degree of freedom, DOF)(不包含手爪部分)，即控制器輸出 4 軸速度，輸入端的資料維度至少要大於等於 4，才能完整利用到機器人上的所有自由度，而影像平面上一個像素座標(u, v)維度為 2，所以我們使用兩個特徵點，即目標的左上角與右下角作為控制器的輸入，如此輸入端的特徵空間維度為 4，和機器人的自由度相同。

由於我們所要設計的控制需要將特徵空間的誤差訊號轉換為機器人的速度命令，因此我們要先建立特徵空間與機器人控制命令之間的轉換關係。在分解運動速率控制中，此關係意指特徵點在影像平面上的移動速度和機器人各軸角速度間的轉換。以下將此關係的推導分成三個部分：2.2.2 節說明特徵點速度和攝影機速度間的轉換關係；2.2.3 節說明攝影機速度和機器人各軸角速度間的關

係；2.2.4 節說明連結 2.2.2 和 2.2.3 節所得結果所需要的座標系轉換關係。

## 2.2.2. Image Jacobian

我們首先推導特徵點速度和攝影機速度間的轉換關係，此轉換可用一個矩陣來表示，在文獻中此矩陣稱為 Image Jacobian，即特徵點和攝影機間的速度轉換矩陣。

特徵點是影像平面上的像素座標，而攝影機的位置座標是定義在依附於攝影機上的卡氏座標上，兩者間的座標轉換關係稱為投影模型(Projection model)，比較常見的投影模型有[7]: 透視投影(Perspective projection)、正射投影(Orthographic projection)、仿射投影(Affine projection)等。其中，正射投影適用於場景中，投影物體的景深差距遠比投影物體與攝影機間的距離小的情況；仿射投影則是將透視投影用線性的方法近似，可以簡化攝影機的校正。本論文中，我們將用透視投影作為攝影機的投影模型。

圖 2.4 為透視投影示意圖，圖中的  $\lambda$  為攝影機焦距，特徵點 P 的空間座標為  $(x,y,z)$ ，投影在影像平面上的特徵點座標為  $(u,v)$ ，圖上的空間座標軸為依附於攝影機上的座標軸  $\{F_c\}$ ， $\{F_c\}$  隨著攝影機移動。

在透視投影中，影像平面的特徵點座標和空間座標的關係為：

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{z} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.1)$$

假設攝影機位於一個靜止的環境中，相對於攝影機座標軸  $\{F_c\}$ ，以平移速度  $T = [T_x, T_y, T_z]^T$ 、旋轉速度  $\Omega = [\omega_x, \omega_y, \omega_z]^T$  移動。則 P 點在  $\{F_c\}$  中的速度為：

$$\frac{dP}{dt} = -\Omega \times P - T \quad (2.2)$$

(2.3)式為矩陣形式，其中各分量分別為：

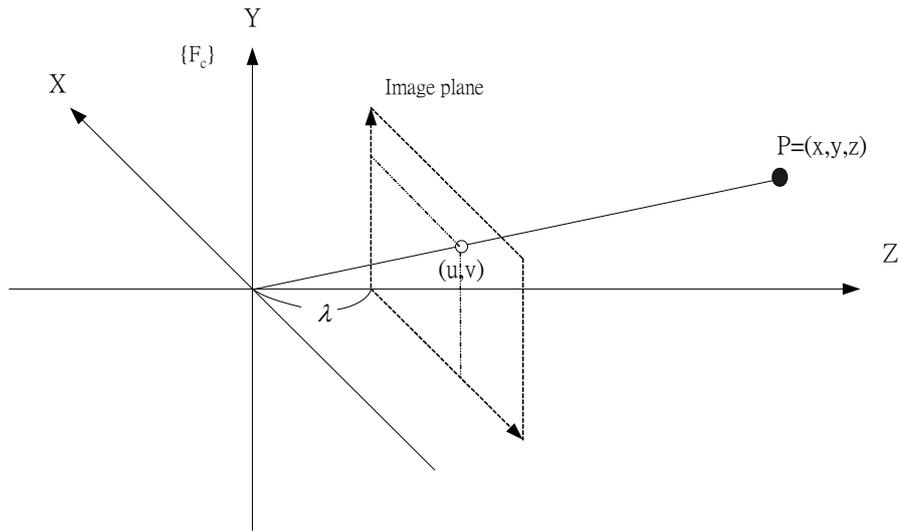


圖 2.4 Perspective projection[7]

$$\begin{aligned}
 \dot{x} &= -z\omega_y + y\omega_z - T_x \\
 \dot{y} &= -x\omega_z + z\omega_x - T_y \\
 \dot{z} &= -y\omega_x + x\omega_y - T_z
 \end{aligned} \tag{2.3}$$

將(2.1)式代入(2.3)式：

$$\begin{aligned}
 \dot{x} &= -z\omega_y + \frac{zv}{\lambda}\omega_z - T_x \\
 \dot{y} &= -\frac{zu}{\lambda}\omega_z + z\omega_x - T_y \\
 \dot{z} &= -\frac{zv}{\lambda}\omega_x + \frac{zu}{\lambda}\omega_y - T_z
 \end{aligned} \tag{2.4}$$

將(2.1)式微分後，再將(2.4)代入可得到：

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uv}{\lambda} & -\frac{\lambda^2 + u^2}{\lambda} \\ 0 & -\frac{\lambda}{z} & \frac{v}{z} & \frac{\lambda^2 + v^2}{\lambda} & -\frac{uv}{\lambda} \end{bmatrix} \begin{bmatrix} v \\ -u \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.5}$$

$$\text{令 } J = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uv}{\lambda} & -\frac{\lambda^2 + u^2}{\lambda} & v \\ 0 & -\frac{\lambda}{z} & \frac{v}{z} & \frac{\lambda^2 + v^2}{\lambda} & -\frac{uv}{\lambda} & -u \end{bmatrix}, \dot{r} = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \text{ 則(2.5)式可表}$$

示為：

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = J\dot{r} \quad (2.6)$$

(2.6)式中， $J$ 即為 Image Jacobian， $\dot{r}$ 為攝影機在 $\{F_c\}$ 中的速度， $[u, v]^T$ 為單一特徵點的像素座標。因此(2.6)式即是攝影機速度與影像特徵點的速度關係。當選取的特徵點增加時，Image Jacobian 可以由個別特徵點的 Image Jacobian 堆疊而成，如本論文中的視覺伺服系統使用兩個特徵點 $(u_1, v_1), (u_2, v_2)$ ，其速度轉換關係為：

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{z_1} & 0 & \frac{u_1}{z_1} & \frac{u_1 v_1}{\lambda} & -\frac{\lambda^2 + u_1^2}{\lambda} & v_1 \\ 0 & -\frac{\lambda}{z_1} & \frac{v_1}{z_1} & \frac{\lambda^2 + v_1^2}{\lambda} & -\frac{u_1 v_1}{\lambda} & -u_1 \\ -\frac{\lambda}{z_2} & 0 & \frac{u_2}{z_2} & \frac{u_2 v_2}{\lambda} & -\frac{\lambda^2 + u_2^2}{\lambda} & v_2 \\ 0 & -\frac{\lambda}{z_2} & \frac{v_2}{z_2} & \frac{\lambda^2 + v_2^2}{\lambda} & -\frac{u_2 v_2}{\lambda} & -u_2 \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.7)$$

在(2.1)式中，影像平面座標 $(u, v)$ 所使用的單位是長度單位，和空間座標的單位相同。但是，實際上攝影機所取得的影像是一連串的像素，因此在實際使用時需要將 $(u, v)$ 的單位轉換成像素。將(2.1)式中的 $(u, v)$ 座標單位換成像素後，可得到(2.8)式[8]：

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{z} \begin{bmatrix} x \\ s_x \\ y \\ s_y \end{bmatrix} \quad (2.8)$$

其中， $s_x$  為影像平面  $x$  軸的轉換係數， $s_y$  為影像平面  $y$  軸的轉換係數。

(2.8)式中和攝影機有關的三個參數  $\lambda$ 、 $s_x$ 、 $s_y$  均為未知數，需要經由實際的量測來取得正確數值。為了簡化量測過程，我們將三個參數簡化成兩個，分別為：

$$\begin{aligned}\lambda_x &= \frac{\lambda}{s_x} \\ \lambda_y &= \frac{\lambda}{s_y}\end{aligned}\quad (2.9)$$

以(2.8)、(2.9)式取代(2.1)式代入(2.7)式中，可以得到實際使用時的 Image Jacobian 為：

$$J = \begin{bmatrix} -\frac{\lambda_x}{z} & 0 & \frac{u}{z} & \frac{uv}{\lambda_y} & -\frac{\lambda_x^2 + u^2}{\lambda_x} & \frac{\lambda_x}{\lambda_y}v \\ 0 & -\frac{\lambda_y}{z} & \frac{v}{z} & \frac{\lambda_y^2 + v^2}{\lambda_y} & -\frac{uv}{\lambda_x} & -\frac{\lambda_y}{\lambda_x}u \end{bmatrix}\quad (2.10)$$

因此，修正後的(2.7)式為：

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -\frac{\lambda_x}{z_1} & 0 & \frac{u_1}{z_1} & \frac{u_1v_1}{\lambda_y} & -\frac{\lambda_x^2 + u_1^2}{\lambda_x} & \frac{\lambda_x}{\lambda_y}v_1 \\ 0 & -\frac{\lambda_y}{z_1} & \frac{v_1}{z_1} & \frac{\lambda_y^2 + v_1^2}{\lambda_y} & -\frac{u_1v_1}{\lambda_x} & -\frac{\lambda_y}{\lambda_x}u_1 \\ -\frac{\lambda_x}{z_2} & 0 & \frac{u_2}{z_2} & \frac{u_2v_2}{\lambda_y} & -\frac{\lambda_x^2 + u_2^2}{\lambda_x} & \frac{\lambda_x}{\lambda_y}v_2 \\ 0 & -\frac{\lambda_y}{z_2} & \frac{v_2}{z_2} & \frac{\lambda_y^2 + v_2^2}{\lambda_y} & -\frac{u_2v_2}{\lambda_x} & -\frac{\lambda_y}{\lambda_x}u_2 \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}\quad (2.11)$$

$$\text{令 } \dot{f} = \begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \end{bmatrix}, \quad J_{im} = \begin{bmatrix} -\frac{\lambda_x}{z_1} & 0 & \frac{u_1}{z_1} & \frac{u_1v_1}{\lambda_y} & -\frac{\lambda_x^2 + u_1^2}{\lambda_x} & \frac{\lambda_x}{\lambda_y}v_1 \\ 0 & -\frac{\lambda_y}{z_1} & \frac{v_1}{z_1} & \frac{\lambda_y^2 + v_1^2}{\lambda_y} & -\frac{u_1v_1}{\lambda_x} & -\frac{\lambda_y}{\lambda_x}u_1 \\ -\frac{\lambda_x}{z_2} & 0 & \frac{u_2}{z_2} & \frac{u_2v_2}{\lambda_y} & -\frac{\lambda_x^2 + u_2^2}{\lambda_x} & \frac{\lambda_x}{\lambda_y}v_2 \\ 0 & -\frac{\lambda_y}{z_2} & \frac{v_2}{z_2} & \frac{\lambda_y^2 + v_2^2}{\lambda_y} & -\frac{u_2v_2}{\lambda_x} & -\frac{\lambda_y}{\lambda_x}u_2 \end{bmatrix},$$

$${}^cV_C = \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \text{ 則(2.11)式可表示成:}$$

$$\dot{f} = J_{im} {}^cV_C \quad (2.12)$$

$J_{im}$  即為實際使用在控制器設計中的 Image Jacobian。

### 2.2.3. 機器人(Mobile manipulator) Jacobian

由前一節的推導，我們已有了影像速度與攝影機速度間的轉換矩陣  $J_{im}$ 。接下來我們需要將攝影機速度和機器人各軸速度的轉換，以便使控制器能由影像平面的輸入直接控制機器人的速度。

首先我們先定義機器人上的各個座標系，圖 2.5 標示了機器人上的座標系  $\{F_B\}$  與各項參數，包括手臂各段長度： $L_1, L_2$ ；手臂二軸的轉動角度： $\theta_1, \theta_2$ ；攝影機距手臂高度： $L_3$ ；機器人本體的旋轉角度： $\theta_B$ ；機器人的線速度： $v_B$ 。

圖 2.6 利用 Denavit-Hartenberg model (D-H model)[15]表示手臂上每個關節與依附其上的各個座標系，其中座標系  $\{B\}$ 、 $\{B1\}$ 、 $\{0\}$  的原點相同， $\{4\}$ 、 $\{C1\}$  的原點相同，為了能更清楚地標示，將其以虛線連結標示在所依附的關節旁。手臂的各項 D-H model 參數列表於表 2.1。由於機器人總共有 4 個自由度，其中和手臂有關的有 3 個： $\theta_B, \theta_1, \theta_2$ ；因此，表 2.1 中只有三個為可動關節，其他的關節 B1、0 為虛擬關節，由於 D-H model 只使用 4 個參數來表示兩個關節間的轉換，因此無法直接表示關節 B 到 1 的參數，因此多用這兩個關節來表示；同樣地，4 也是為了用 D-H model 表示關節 3 到 C1 的虛擬關節；只有關節 0、1、2 為實際可動的關節。

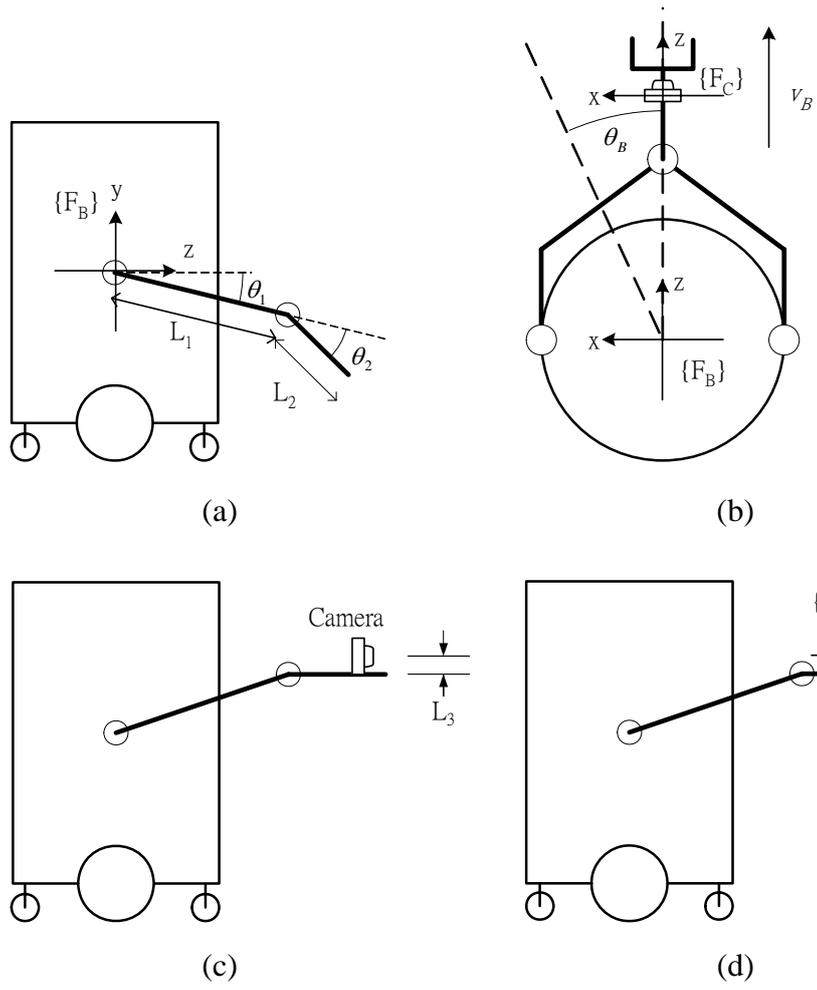


圖 2.5 機器人上的座標系與各項參數

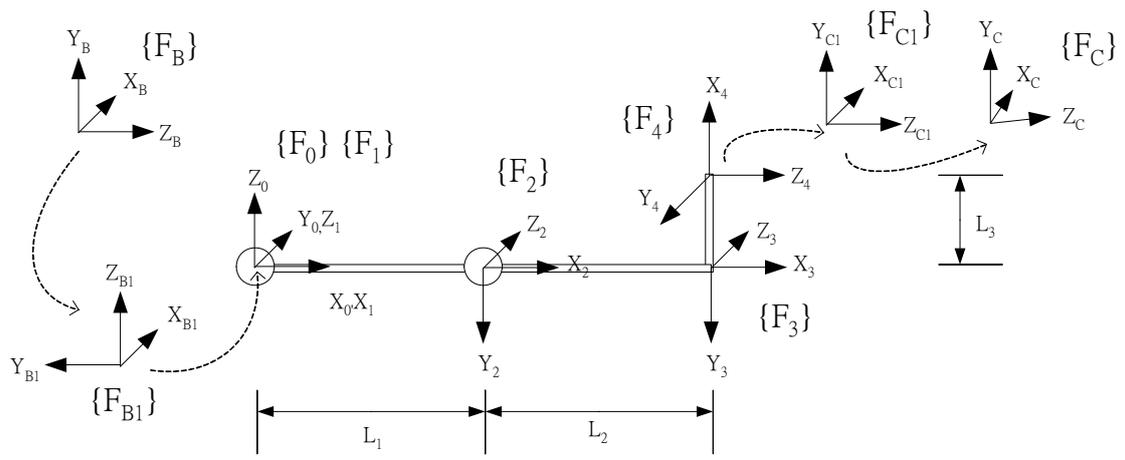


圖 2.6 手臂上 D-H model 與各個座標系

表 2.1 Kinematics parameters of robot and arm

Joint	$\theta$	d	a	$\alpha$
$B_1$	0	0	0	$-\pi/2$
0	$-\pi/2$	0	0	0
1	$\theta_B$	0	0	$-\pi/2$
2	$\theta_1$	0	$L_1$	0
3	$\theta_2$	0	$L_2$	0
4	$-\pi/2$	0	$L_3$	$-\pi/2$
C1	$-\pi/2$	0	0	0

圖 2.6 上的座標軸 $\{F_{C1}\}$ 並非實際依附在攝影機上的座標軸。因為攝影機架設上的誤差，鏡頭軸心並非朝向機器人的正前方，而是對  $Y_{C1}$  軸旋轉了一個小角度 ( $\theta_x$ )，因此在 D-H model 之後再加上 $\{F_C\}$ 來修正這項誤差。

首先，我們先不考慮機器人的線速度  $v_B$ ，只看機器人本體的旋轉，和手臂兩軸的旋轉，則機器人可以視為一個固定式的機械臂，共有三個自由度。利用 D-H model 參數所表示的座標系之間，其轉換矩陣(Transform matrix)為：

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

利用(2.13)式與表 2.1，可以得到各座標系的轉換矩陣如下：

$${}^B A_{B1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{B1} A_0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^0 A_1 = \begin{bmatrix} C_B & 0 & -S_B & 0 \\ S_B & 0 & C_B & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^1 A_2 = \begin{bmatrix} C_1 & -S_1 & 0 & L_1 C_1 \\ S_1 & C_1 & 0 & L_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^2 A_3 = \begin{bmatrix} C_2 & -S_2 & 0 & L_2 C_2 \\ S_2 & C_2 & 0 & L_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^3 A_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -L_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^4 A_{C1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

另外，由 $\{F_{C1}\}$ 到 $\{F_C\}$ 為對 Y 軸旋轉，角度為 $\theta_x$ ，其轉換矩陣為：

$${}^{C1}A_C = \begin{bmatrix} C_x & 0 & S_x & 0 \\ 0 & 1 & 0 & 0 \\ -S_x & 0 & C_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

因此，機器人座標系 $\{F_B\}$ 到攝影機座標系 $\{F_C\}$ 的轉換矩陣：

$$\begin{aligned} {}^B A_C &= {}^B A_{B1} {}^{B1} A_0 {}^0 A_1 {}^1 A_2 {}^2 A_3 {}^3 A_4 {}^4 A_{C1} {}^{C1} A_C \\ &= \begin{bmatrix} C_B C_x - S_B C_{12} S_x & S_B S_{12} & C_B S_x + S_B C_{12} C_x & S_B (L_3 S_{12} + L_2 C_{12} + L_1 C_1) \\ S_{12} S_x & C_{12} & -S_{12} C_x & L_3 C_{12} - L_2 S_{12} - L_1 S_1 \\ -S_B C_x - C_B C_{12} S_x & C_B S_{12} & -S_B S_x + C_B C_{12} C_x & C_B (L_3 S_{12} + L_2 C_{12} + L_1 C_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.16)$$

其中， $C_B = \cos(\theta_B)$ 、 $S_B = \sin(\theta_B)$ 、 $S_{12} = \sin(\theta_1 + \theta_2)$ ，依此類推。

因此，若不考慮機器人之線速度 $v_B$ ，則由(2.16)式可直接得到攝影機座標系原點在機器人座標系中的平移向量為

$${}^B r_C = \begin{bmatrix} S_B (L_3 S_{12} + L_2 C_{12} + L_1 C_1) \\ L_3 C_{12} - L_2 S_{12} - L_1 S_1 \\ C_B (L_3 S_{12} + L_2 C_{12} + L_1 C_1) \end{bmatrix} \quad (2.17)$$

旋轉向量為

$${}^B R_C = \begin{bmatrix} C_B C_x - S_B C_{12} S_x & S_B S_{12} & C_B S_x + S_B C_{12} C_x \\ S_{12} S_x & C_{12} & -S_{12} C_x \\ -S_B C_x - C_B C_{12} S_x & C_B S_{12} & -S_B S_x + C_B C_{12} C_x \end{bmatrix} \quad (2.18)$$

對(2.17)式微分可得到攝影機的原點速度

$${}^B T_C = {}^B \dot{r}_C = \begin{bmatrix} C_B (L_3 S_{12} + L_2 C_{12} + L_1 C_1) & S_B (L_3 C_{12} - L_2 S_{12} - L_1 S_1) & S_B (L_3 C_{12} - L_2 S_{12}) \\ 0 & -L_3 S_{12} - L_2 C_{12} - L_1 C_1 & -L_3 S_{12} - L_2 C_{12} \\ -S_B (L_3 S_{12} + L_2 C_{12} + L_1 C_1) & C_B (L_3 C_{12} - L_2 S_{12} - L_1 S_1) & C_B (L_3 C_{12} - L_2 S_{12}) \end{bmatrix} \begin{bmatrix} \dot{\theta}_B \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (2.19)$$

(2.19)式是在假設機器人為一個固定式的機械臂的前提下所得到的攝影機平移速度。對一個固定式的機械臂來說，基座座標系 $\{F_B\}$ 是固定不動的，但是實際

上機器人並非固定式，因此座標系 $\{F_B\}$ 會隨機器人而旋轉，在任何時刻下的 $\theta_B$ 均為0。將 $\theta_B = 0$ 代入(2.19)式，並加上機器人線速度，可得到完整的攝影機平移速度：

$${}^B T_C = \begin{bmatrix} L_3 S_{12} + L_2 C_{12} + L_1 C_1 & 0 & 0 \\ 0 & -L_3 S_{12} - L_2 C_{12} - L_1 C_1 & -L_3 S_{12} - L_2 C_{12} \\ 0 & L_3 C_{12} - L_2 S_{12} - L_1 S_1 & L_3 C_{12} - L_2 S_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_B \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ v_B \end{bmatrix} \quad (2.20)$$

有了攝影機的平移速度後，接下來我們需要知道攝影機的旋轉速度，以得到完整的攝影機和機器人速度間的轉換關係。

在 D-H model 中，一個座標系的旋轉速度( $\Omega$ )可由前一個座標系的旋轉速度加上本身的旋轉速度來求得[9]：

$${}^i \Omega_i = {}^i R_{i-1} {}^{i-1} \Omega_{i-1} + \dot{\theta}_i \hat{Z}_i, \text{ 其中 } \hat{Z}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.21)$$

(2.21)式中， ${}^i \Omega_i$ 為 $i$ 座標系在本身座標系中的旋轉速度， ${}^i R_{i-1}$ 為 $i-1$ 座標系到 $i$ 座標系的旋轉矩陣。

由(2.14)(2.15)，可直接得知各座標系間的旋轉矩陣( ${}^i R_{i-1}$ )如下：

$${}^1 R_0 = \begin{bmatrix} C_B & S_B & 0 \\ 0 & 0 & -1 \\ -S_B & C_B & 0 \end{bmatrix}, {}^2 R_1 = \begin{bmatrix} C_1 & S_1 & 0 \\ -S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, {}^3 R_2 = \begin{bmatrix} C_2 & S_2 & 0 \\ -S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$${}^4 R_3 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}, {}^{c1} R_4 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, {}^C R_{c1} = \begin{bmatrix} C_x & 0 & -S_x \\ 0 & 1 & 0 \\ S_x & 0 & C_x \end{bmatrix} \quad (2.22)$$

利用(2.21)式可依序求得：

$${}^0\Omega_0 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_B \end{bmatrix}$$

$${}^1\Omega_1 = {}^1R_0 {}^0\Omega_0 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} C_B & S_B & 0 \\ 0 & 0 & -1 \\ -S_B & C_B & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_B \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -\dot{\theta}_B \\ \dot{\theta}_1 \end{bmatrix}$$

$${}^2\Omega_2 = {}^2R_1 {}^1\Omega_1 + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} C_1 & S_1 & 0 \\ -S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -\dot{\theta}_B \\ \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -S_1\dot{\theta}_B \\ -C_1\dot{\theta}_B \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$${}^3\Omega_3 = {}^3R_2 {}^2\Omega_2 = \begin{bmatrix} C_2 & S_2 & 0 \\ -S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -S_1\dot{\theta}_B \\ -C_1\dot{\theta}_B \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -S_{12}\dot{\theta}_B \\ -C_{12}\dot{\theta}_B \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}$$

$${}^4\Omega_4 = {}^4R_3 {}^3\Omega_3 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -S_{12}\dot{\theta}_B \\ -C_{12}\dot{\theta}_B \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} C_{12}\dot{\theta}_B \\ -(\dot{\theta}_1 + \dot{\theta}_2) \\ -S_{12}\dot{\theta}_B \end{bmatrix}$$

$${}^{c1}\Omega_{c1} = {}^{c1}R_4 {}^4\Omega_4 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_{12}\dot{\theta}_B \\ -(\dot{\theta}_1 + \dot{\theta}_2) \\ -S_{12}\dot{\theta}_B \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 + \dot{\theta}_2 \\ C_{12}\dot{\theta}_B \\ -S_{12}\dot{\theta}_B \end{bmatrix}$$

$${}^c\Omega_C = {}^cR_{c1} {}^{c1}\Omega_{c1} = \begin{bmatrix} C_x & 0 & -S_x \\ 0 & 1 & 0 \\ S_x & 0 & C_x \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 + \dot{\theta}_2 \\ C_{12}\dot{\theta}_B \\ -S_{12}\dot{\theta}_B \end{bmatrix} = \begin{bmatrix} C_x(\dot{\theta}_1 + \dot{\theta}_2) + S_{12}S_x\dot{\theta}_B \\ C_{12}\dot{\theta}_B \\ S_x(\dot{\theta}_1 + \dot{\theta}_2) - S_{12}C_x\dot{\theta}_B \end{bmatrix} \quad (2.23)$$

再將 ${}^c\Omega_C$ 轉換到機器人座標系，並將 $\theta_B = 0$ 條件代入：

$$\begin{aligned} {}^B\Omega_C &= {}^B R_C {}^c\Omega_C = \begin{bmatrix} C_B C_x - S_B C_{12} S_x & S_B S_{12} & C_B S_x + S_B C_{12} C_x \\ S_{12} S_x & C_{12} & -S_{12} C_x \\ -S_B C_x - C_B C_{12} S_x & C_B S_{12} & -S_B S_x + C_B C_{12} C_x \end{bmatrix} \begin{bmatrix} C_x(\dot{\theta}_1 + \dot{\theta}_2) + S_{12} S_x \dot{\theta}_B \\ C_{12} \dot{\theta}_B \\ S_x(\dot{\theta}_1 + \dot{\theta}_2) - S_{12} C_x \dot{\theta}_B \end{bmatrix} \\ &= \begin{bmatrix} \dot{\theta}_1 + \dot{\theta}_2 \\ \dot{\theta}_B \\ 0 \end{bmatrix} \end{aligned} \quad (2.24)$$

結合(2.20)(2.24)式，可得到攝影機速度與機器人速度間的關係：

$${}^B V_C = \begin{bmatrix} {}^B T_C \\ {}^B \Omega_C \end{bmatrix} = \begin{bmatrix} 0 & L_3 S_{12} + L_2 C_{12} + L_1 C_1 & 0 & 0 \\ 0 & 0 & -L_3 S_{12} - L_2 C_{12} - L_1 C_1 & -L_3 S_{12} - L_2 C_{12} \\ 1 & 0 & L_3 C_{12} - L_2 S_{12} - L_1 S_1 & L_3 C_{12} - L_2 S_{12} \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_B \\ \dot{\theta}_B \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (2.25)$$

$$\text{令 } J_r = \begin{bmatrix} 0 & L_3 S_{12} + L_2 C_{12} + L_1 C_1 & 0 & 0 \\ 0 & 0 & -L_3 S_{12} - L_2 C_{12} - L_1 C_1 & -L_3 S_{12} - L_2 C_{12} \\ 1 & 0 & L_3 C_{12} - L_2 S_{12} - L_1 S_1 & L_3 C_{12} - L_2 S_{12} \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$u = \begin{bmatrix} v_B \\ \dot{\theta}_B \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \text{ 可將(2.25)式表示為：}$$



$${}^B V_C = J_r u \quad (2.26)$$

其中， $J_r$  即為機器人的 Jacobian， $u$  為機器人的速度控制命令。

#### 2.2.4. 座標轉換

(2.12)中的攝影機速度為在攝影機座標系 $\{F_C\}$ 中的速度，(2.26)中的攝影機速度為機器人座標系 $\{F_B\}$ 中的速度，兩者參考的座標系不同，因此要結合(2.12)和(2.26)時，需要先經過座標轉換。

兩個座標系間的速度轉換，可用下列式子來表示[7]：

$$\begin{aligned}
{}^B V_C &= \begin{bmatrix} {}^B T_C \\ {}^B \Omega_C \end{bmatrix} = \begin{bmatrix} {}^B R_C {}^C T_C - {}^B R_C {}^C \Omega_C \times {}^B r_C \\ {}^B R_C {}^C \Omega_C \end{bmatrix} = \begin{bmatrix} {}^B R_C {}^C T_C + {}^B r_C \times {}^B R_C {}^C \Omega_C \\ {}^B R_C {}^C \Omega_C \end{bmatrix} \\
&= \begin{bmatrix} {}^B R_C & sk({}^B r_C) {}^B R_C \\ 0 & {}^B R_C \end{bmatrix} \begin{bmatrix} {}^C T_C \\ {}^C \Omega_C \end{bmatrix}
\end{aligned} \tag{2.27}$$

$$\text{令 } {}^B M_C = \begin{bmatrix} {}^B R_C & sk({}^B r_C) {}^B R_C \\ 0 & {}^B R_C \end{bmatrix}, \quad {}^C V_C = \begin{bmatrix} {}^C T_C \\ {}^C \Omega_C \end{bmatrix}, \text{ 則(2.27)可寫成:}$$

$${}^B V_C = {}^B M_C {}^C V_C \tag{2.28}$$

因此， ${}^B M_C$  即為攝影機座標系  $\{F_C\}$  和機器人座標系  $\{F_B\}$  間的速度轉換矩陣。

## 2.2.5. 分解運動速率控制(Resolved motion rate control)

分解運動速率控制是常被利用來設計 IBVS 架構控制器的方法[7]，此方法原本是在固定式的機械臂上用來設計運動控制器的方法，由於多軸的機械臂其 Inverse kinematics 通常都很複雜，因此可以利用其各軸與末端操作器間的速度關係矩陣(Jacobian matrix)來設計速度控制器，若需要末端操作器的位置資訊，則可將其速度積分。

分解運動速率控制設計的方法是[14]，假設  $x$  是位於機械臂工作空間中的座標，即末端控制器可到達的座標，若我們可以得到機器人的 Jacobian 關係

$$\dot{x} = J_a \cdot \dot{q},$$

其中， $\dot{x}$  為末端操作器的速度， $\dot{q}$  為各軸的速度， $J_a$  是機械臂的 Jacobian 矩陣，大小為  $n \times m$ ，則若  $n = m$ ，我們可以直接求  $J_a$  的反矩陣，並求得某個末端操作器的速度來下，機器人各軸的速度：

$$\dot{q} = J_a^{-1} \cdot \dot{x}$$

若  $n < m$ ，則可以求  $J_a$  的 pseudo inverse，同樣可得到末端操作器與各軸速度的關係：

$$\dot{q} = (J_a^T J_a)^{-1} J_a^T \cdot \dot{x} \quad (2.29)$$

## 2.2.6. 控制器設計

結合(2.12)(2.26)(2.28)三式，可得到影像平面上的特徵點速度與機器人速度間的關係：

$$\begin{aligned} \dot{f} &= J_{im}^C V_C = J_{im} ({}^B M_C^{-1} V_C) = J_{im} {}^B M_C^{-1} (J_r u) \\ \rightarrow \dot{f} &= J u \end{aligned} \quad (2.30)$$

其中， $J = J_{im} {}^B M_C^{-1} J_r$  即為特徵點速度與機器人速度的轉換矩陣。

接下來我們利用分解運動速率控制(Resolved motion rate control)來設計控制器，讓機器人能在影像平面上到達預定的影像特徵點位置。利用分解運動速率控制的設計概念，可以將其運用在 IBVS 控制器的設計上。利用(2.30)式，我們可以反過來由特徵點的速度來求得機器人的速度：

$$u = K \cdot J^{-1} \cdot \dot{f} \quad (2.31)$$

其中  $u = [v_B \quad \dot{\theta}_B \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T$  為機器人的速度控制命令， $\dot{f}$  為影像特徵點速度， $K$  為常數增益矩陣(Gain matrix)。由於  $u$  和  $\dot{f}$  均為四個自由度，因此  $J$  為大小 4x4 的方陣，可直接求得反矩陣。

假設影像平面上，預定的影像特徵點位置(Desired feature position)為  $f_d$ ，目前的特徵點位置為  $f$ 。我們定義特徵點位置誤差訊號為  $e = f_d - f$ ，並將其視為特徵點速度  $\dot{f}$  代入(2.31)式，可得到[7]

$$u = K \cdot J^{-1} \cdot e = K \cdot J^{-1} (f_d - f) \quad (2.32)$$

將  $e$  視為  $\dot{f}$  是假設若要在下一個時間點到達  $f_d$ ，所需要的機器人速度。 $K$  的數值

可以視馬達控制訊號的大小範圍調整。(2.32)式就是我們將用在 IBVS 中的特徵空間控制器。

### 2.2.7. 奇異點分析

在利用(2.32)式時，我們需要計算  $J^{-1}$ ，當  $J^{-1}$  不存在時，控制器將不能使用，形成一個奇異點的問題。機器人的奇異點通常發生在其姿態位於比平常自由度少的點，對本論文中的機器人而言，手臂第一軸  $\theta_1 = \pm 90^\circ$  時(如圖 2.7)，矩陣  $J$  將為奇異矩陣，因此若機器人的手臂到達此姿態，例如目標物體高度高於或低於手臂第一段長度，手臂的速度將無法控制。對於此一已知奇異點，在本論文中我們以限制第一軸動作角度在  $\pm 30^\circ$  間來避開。

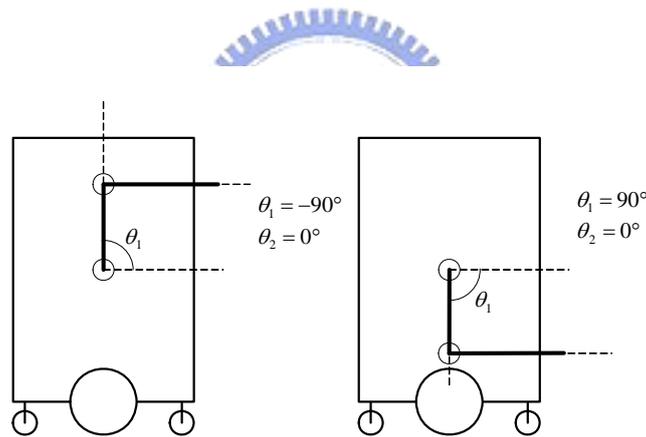


圖 2.7 機器人之奇異點

## 2.3 本章總結

本章利用 Image Jacobian 與自走勢機器人(Mobile manipulator)Jacobian 得到特徵空間與機器人速度控制命令間的轉換關係，並利用分解運動速率控制方法設計了一個 IBVS 控制器。接下去將利用此控制器來設計機器人對目標物體的靠近行為，藉此到達機器人能抓取到目標的姿態，進而使其能抓取到目標。

## 第三章 影像特徵點擷取

由於本論文中所使用的機器人自由度的限制，無法抓取任意姿態與形狀的目標，因此本章將對機器人可抓取的目標做一些基本假設。若目標可被機器人抓取，在第二章所設計的視覺伺服控制器中，我們需要兩個影像平面上的特徵點來作為控制器的輸入，以產生機器人的速度控制命令。本章將介紹影像特徵點的定義，並介紹辨識目標物體與取得此兩個特徵點的影像處理流程。同時，為了簡化擷取特徵點，本論文中利用類條碼樣板(Barcode-like pattern)黏貼在目標上，作為辨識的基準。

除了影像特徵點之外，計算(2.32)式中的 Jacobian( $J$ )時，同時也需要兩個特徵點對攝影機的景深(Depth)，和兩個攝影機模型參數。因此在本章最後將針對景深估測與攝影機參數的校正做說明。



### 3.1 對抓取目標物體的基本假設

在本論文中，對機器人所能抓取的物體有下列兩項假設：

(1) 目標物體的形狀為圓柱體：

做此假設主要是因為我們所選取的特徵點為目標影像的左上與右下角點在影像平面上的座標(目標影像將在 3.2 節說明)。如圖 3.1 所示，若目標物體不是圓柱體，則若攝影機接近目標時的角度和取得預定特徵點位置時攝影機對目標的角度不同，則攝影機所看到的目標影像幾何形狀也會改變。此幾何形狀上的改變將會使得兩個特徵點的相對位置改變，導致機器人無法讓影像平面上的特徵點到達預定的位置。

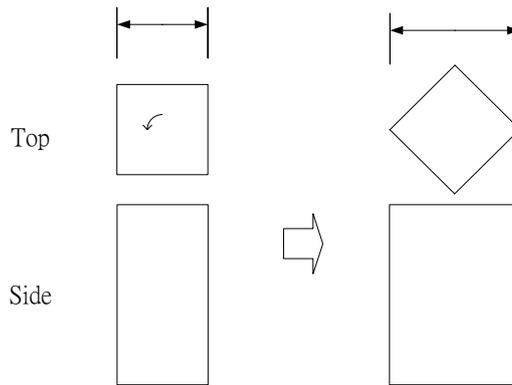


圖 3.1 不同視角下的目標物體

(2) 目標物體的擺設垂直於地面：

做此假設的原因是由於機器人本身的自由度限制，裝設於機器人上的機械臂只有兩個自由度，可以使末端操作器做上下移動，但是無法做滾動(Roll)的動作。此限制將會使裝設在末端操作器上的手爪無法抓取非垂直於地面擺設的目標物體。如圖 3.2 所示，在 (a)情況下，物體相對於手爪有滾動，機械臂無法做出此種動作，因此無法抓取此類目標；在(b)情況下，目標物體往後傾斜，此時若攝影機從側面拍攝，將會看到和(a)情況下相同的目標影像。排除圖 3.2 中的兩種情況，則目標物體只能垂直於地面擺設。

### 3.2 影像特徵定義

因為在 IBVS 中，我們選擇影像平面上的像素座標來做特徵點，因此需要選擇能完整表示物體姿態的特徵，且此特徵點不會因為攝影機接近物體的方向不同而改變。由於我們使用的目標物體為圓柱形，對一個圓柱體來說，從側面的各種角度觀察都會看到一個矩形輪廓，且在前一節的第(2)項假設之下，此矩形的長寬比也將是固定的數值，可以用來作為辨識目標影像階段的判斷條件。因此，我

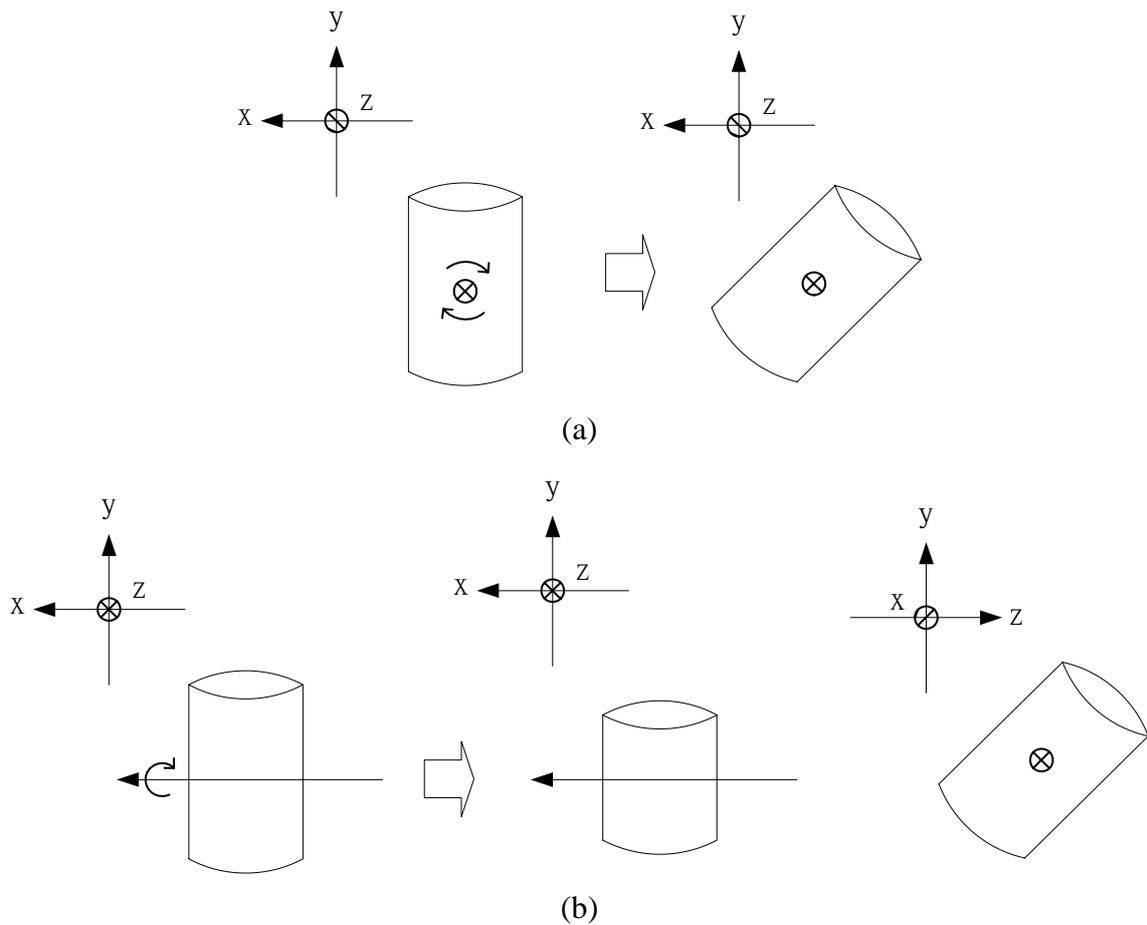


圖 3.2 目標物體非垂直地面：(a) 物體相對於手爪有滾動(Roll)；(b) 物體相對於手爪有傾斜(Tilt)

們定義影像的特徵點為此矩形的左上和右下角點，並且為了方便辨識目標物體，我們設計了一個目標影像的樣板，並將其圍繞在物體的側面，如圖 3.3。

如圖 3.3 所示，目標影像樣板為類條碼式之設計，將其圍繞在物體上時，由攝影機可得到如圖 3.4(b)的目標影像。使用類條碼式的設計主要考量是要讓影像系統能從環境中辨識出目標物體，並且當有多個不同的物體可能被抓取的情況下，能夠由影像中識別出不同的目標物體。如圖 3.3 所示，若我們在影像中做垂直方向的掃描，當掃到目標影像時，可以得到圖 3.3 中三條黑色水平條碼( $V_1, V_3, V_5$ )，與兩條白色水平條碼的長度( $V_2, V_4$ )。在設計上，我們讓  $V_1=V_5$ ，利用  $V_2, V_3, V_4$  三段長度的改變，作為辨識不同目標的依據。

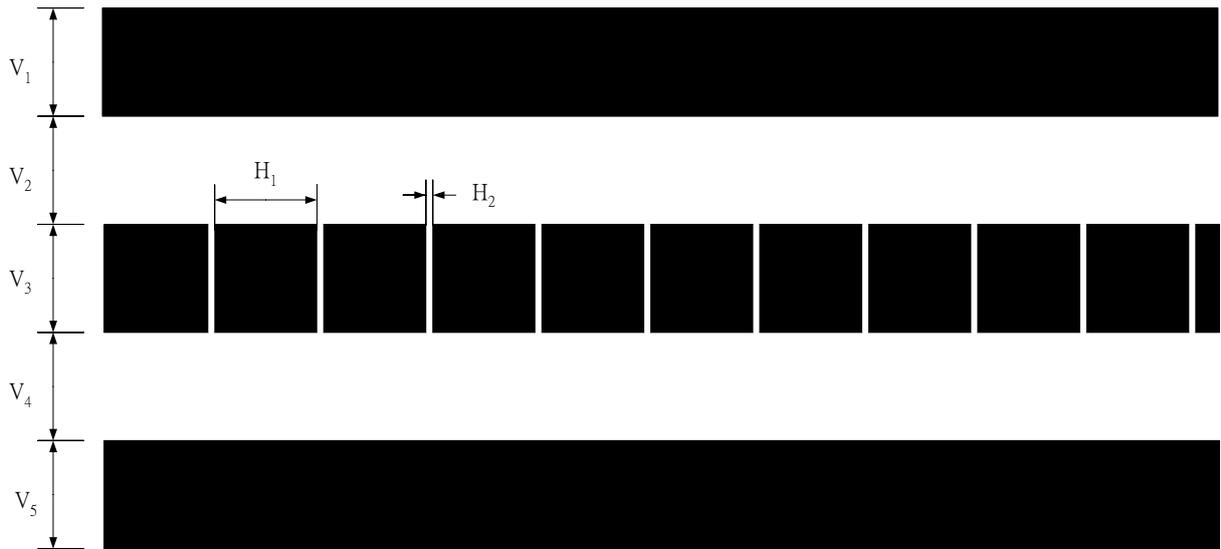


圖 3.3 目標影像樣板

由於機器人上的攝影機採用 Eye-in-hand 的方式設置，在手爪很接近目標物體時，攝影機與目標物體距離將會很短，在此情況下，由於目標影像太大，攝影機將不能看到如圖 3.4(b)中完整的目標影像，當中的三條黑色條碼無法同時在影像上出現。因此，在攝影機很靠近目標物體時，辨識的方法改為掃描中央黑色水平條碼上的垂直白色條碼，若在影像中找到規律的垂直條碼，則視為找到目標影像。



(a) 目標物體； (b) 目標影像

找到目標影像後，我們取目標影像的左上和右下角點作為影像特徵點(圖 3.5(a))，若我們只看到中間的水平條碼，則取中間黑色長條的左上和右下角為影像特徵點(圖 3.5(b))。

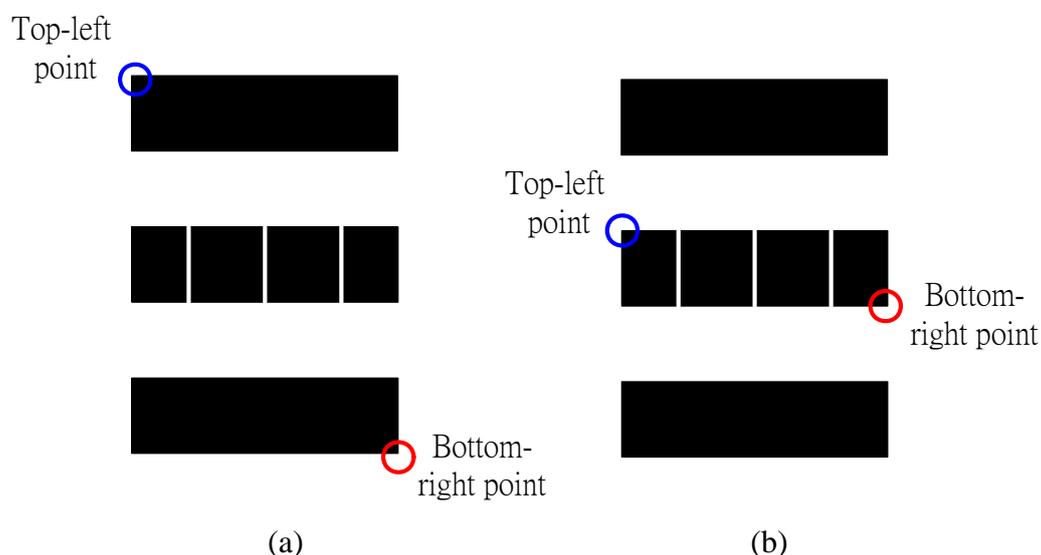


圖 3.5 影像特徵點：(a) 遠端特徵點； (b) 近端特徵點



### 3.3 影像特徵擷取

在本論文中，影像特徵擷取將由一個嵌入式的影像處理平台來完成，詳細硬體說明將在 4.1 節中介紹。

影像系統所取得的原始影像為 640x480 的灰階影像，為了加快特徵擷取的速度，我們將影像解析度降到 160x120 再作處理，首先將影像二值化，接著找出影像中可能為左上和右下角點像素座標，找角點的方法是左上與右下角的影像樣板 (image pattern)(圖 3.6)來和二值化後的影像做 SSD(Sum-of-Square) Correlation 運算，找出可能為目標影像左上與右下角點的座標，SSD Correlation 計算式如下：

$$Cor(x, y) = \sum_{j=-2}^{+2} \sum_{i=-2}^{+2} (I(x+i, y+j) - P(i, j))^2 \quad (3.1)$$

SSD Correlation 所得的數值愈小，則該點愈符合影像樣板，完全符合時數值為 0。

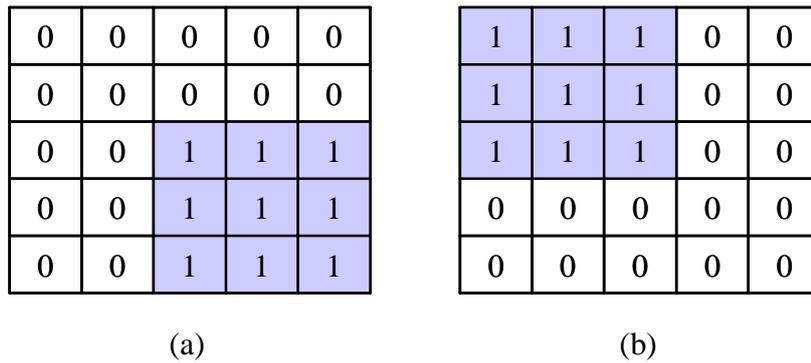


圖 3.6 (a) Top-left pattern ; (b) Bottom-right pattern

找出影像中的可能的角點之後，我們需要從中選出正確的影像特徵點，因為根據機器人距離目標物體遠近的不同，我們選用不同的特徵點，因此檢驗角點的方法也分為兩種：遠端特徵點與近端特徵點。

### 3.3.1 遠端特徵點



遠端特徵點的檢驗分為兩個步驟：

- (1) 檢查角點是否位於目標影像上：

此部份的檢驗方式是對所有找出的左上角點向下做垂直掃描，依序掃出圖 3.3 中的五段長度  $V_1 \sim V_5$ ，此五段長度為目標影像的垂直條碼，如圖 3.7(a) 中掃描出的每段長度都為 5 個像素，因此條碼為 {5, 5, 5, 5, 5}。

當攝影機和目標物體的距離不同，解出的垂直條碼也會跟著改變，但是條碼中的五個元素的大小比例會維持固定值。另外，由於光源與影像二值化所使用的灰階閾值(Threshold)的影響，黑色與白色交界處的影像可能會在二值化時誤判，不過通常若其中一個黑色或白色長條的寬度被誤判，則其他同顏色的長條寬度也會有同樣的誤判。例如，若圖 3.7 中第一個條碼值可能因為二值化閾值的關係而變成 6，第二個條碼值變為 4，通常最後掃描出來的條碼會變成 {6, 4, 6, 4, 6}，也就是相同顏色的長條寬度會相同。因此，基於

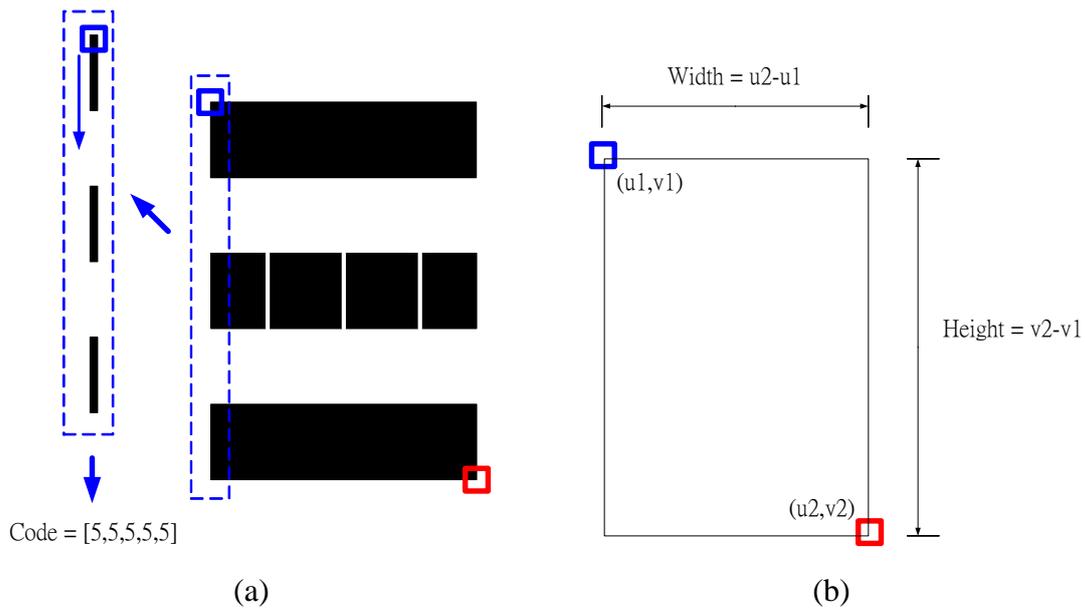


圖 3.7 遠端特徵點檢驗：(a) 垂直掃描；(b) 長寬比

以上兩個原因，檢查垂直條碼的方式以判斷第 1, 2 個條碼元素和與第 3, 4 個元素和的比例是否正確來決定此角點是否落在目標影像上。

藉由檢查垂直條碼，可以確定此左上角點是否位於類條碼目標影像上。右下角點的檢驗和左上角點相同，不過掃描方向為垂直向上。

(2) 選擇長寬比最正確的一對左上與右下角點：

當所有落於目標影像上的左上與右下角點找到之後，再找出所有左上右下點組合所形成的長寬比(圖 3.7(b))中和預定的特徵點(Desired feature)之長寬比最相近的一組，並檢查此組長寬比和預定特徵點之長寬比間的誤差是否小於一固定閾值，若在閾值範圍內即認定為正確的遠端特徵點。

### 3.3.2 近端特徵點

近端特徵點的檢驗方式同樣分為兩個步驟：

(1) 檢查角點是否位於目標影像上：

由於在近端看不到完整的目標影像，因此這部分的檢驗方式針對目標影像中間的水平條碼。此部份的角點檢驗一次需要一對左上與右下角點，因為在水平條碼的部分無法確定條碼的長度為何，其條碼長度會因為觀察的角度不同或光源影響(例如因為光源太弱，使其中一個垂直白色條紋在二值化時被判定為黑色)而改變，在不知道條碼長度的情況下，如果只有單一個左上角點，則我們只能知道掃描的起點位置，無法得知何時該結束掃描。因此，我們一次取一對左上右下角點來做檢驗，在兩個角點間的三個不同水平位置掃描，得到三個條碼值分別為 Code\_up, Code\_center, Code\_down(如圖 3.8(a)所示)，並以 Code\_center 條碼為基準，檢查 Code\_up 或 Code\_down 條碼中是否有一組和 Code\_center 相同，若兩組中有一組和 Code\_center 相同，則判定目前的左上右下角點落在目標影像的中央長條上。

(2) 選擇長寬比最正確的一對左上與右下角點：

若在前一個步驟檢查到一組左上右下角點位於中央長條上，則比較此組角點所形成的長寬比(圖 3.8(b))和預定特徵點的長寬比間的誤差。最後找出所有位於中央長條上的角點對中，長寬比誤差最小的一對，若此誤差小於一固定閾值，則判定此角點對為正確的影像特徵點。

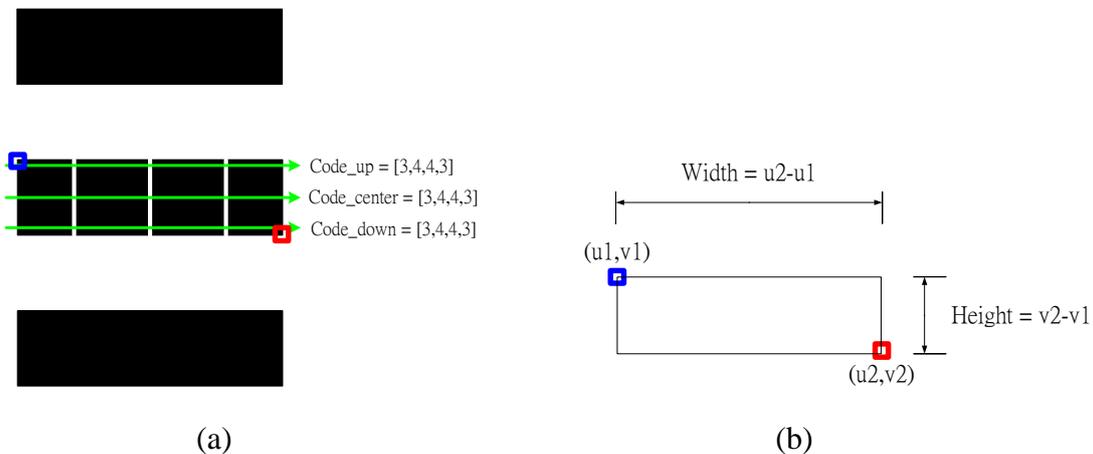


圖 3.8 近端特徵點檢驗：(a) 水平掃描；(b) 長寬比

圖 3.9 為擷取影像特徵的整體流程圖，包含以上所介紹的所有步驟。

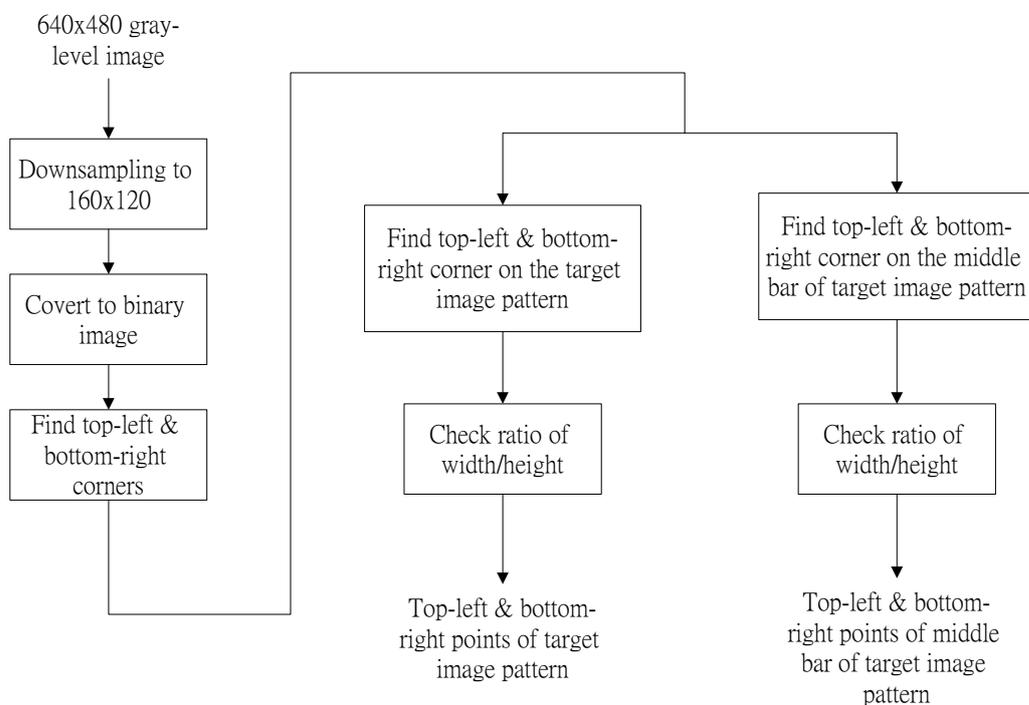


圖 3.9 影像特徵擷取流程圖



### 3.4 目標距離估測

在計算 Image Jacobian 時，需要有每個特徵點與攝影機的景深(Depth)資訊，即(2.11)式中的  $z_1, z_2$ 。在本論文中主要利用透視投影(Perspective projection)的特性來估測特徵點的景深。

如圖 3.10 所示， $D$  為目標影像在空間中的實際寬度， $\lambda$  為攝影機焦距，當物體與鏡頭距離為  $x$  和  $x'$  時，在影像平面上的成像長度分別為  $d$  和  $d'$ 。由相似三角形的比例關係，我們可得到：

$$\begin{cases} \frac{d}{\lambda} = \frac{D}{x} \\ \frac{d'}{\lambda} = \frac{D}{x'} \end{cases} \Rightarrow \frac{d'}{\lambda} = \frac{dx/\lambda}{x'} \Rightarrow x' = \frac{d}{d'}x \quad (3.2)$$

若我們知道當目標物體與鏡頭的距離為  $x$  時，影像平面上的寬度為  $d$ ，則可

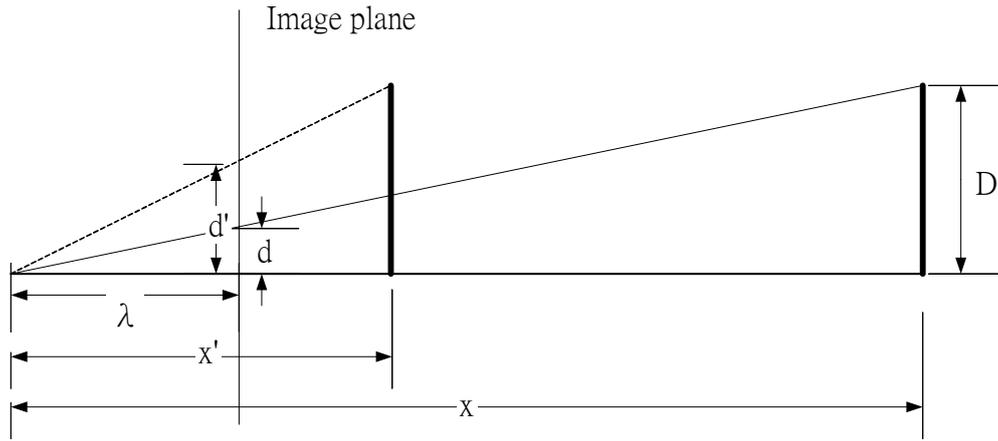


圖 3.10 距離估側原理

以利用(3.2)式和當時目標物體在影像平面上的寬度  $d'$ ，來推得當時物體與鏡頭的距離  $x'$ 。表 3.1 與表 3.2 是以  $x = 21.6cm$ ， $d = 47 pixel$  作為參考點，在各距離下利用遠端與近端特徵點求得物體寬度，代入(3.2)式所得到的估測距離。

由於我們假設實驗時目標物體為垂直地面，且攝影機初始狀態為與地面平行，兩個特徵點對鏡頭的距離可視為近似相等，因此在 Image Jacobian 計算中，我們將兩個特徵點與鏡頭的距離  $z_1, z_2$  視為相等來計算。這個假設在 4.2.1 節的模擬結果中驗證是可行的。

### 3.5 攝影機參數的校正

在 Image Jacobian 運算時，需要知道兩個攝影機參數，即(2.9)式中的  $\lambda_x$  和  $\lambda_y$ 。要取得這兩個參數，我們先將(2.9)式代入(2.8)式，可得到

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\lambda_x x}{z} \\ \frac{\lambda_y y}{z} \end{bmatrix} \quad (3.3)$$

假設特徵點在影像平面上的座標為左上角  $(u_1, v_1)$ ，右下角  $(u_2, v_2)$ ，實際在空間中的座標為左上角  $(x_1, y_1, z_1)$ ，右下角  $(x_2, y_2, z_2)$ ，則目標影像實際在空間中的寬

表 3.1 遠端特徵點之距離估測結果

遠端特徵點		
實際距離(cm)	估測距離(cm)	絕對誤差百分比(%)
84	92.3	9.88
80	84.6	5.75
75.5	78.1	3.44
68.5	72.5	5.84
65	67.7	4.15
61.5	63.5	3.25
58.5	59.7	2.05
54.5	53.4	2.02
51.5	50.8	1.36
48	46.1	3.96
44.5	42.3	4.94
41.5	40.6	2.17
37.5	35	6.67
33.5	31.7	5.37
28	27.4	2.14
25	24.1	3.60
22	21.2	3.64

表 3.2 近端特徵點之距離估測結果

近端特徵點		
實際距離(cm)	估測距離(cm)	絕對誤差百分比(%)
44	46.1	4.77
40.5	40.61	0.27
34.5	36.3	5.22
30.5	29.9	1.97
27	27.4	1.48
24.5	24.8	1.22
22	21.6	1.82
20.5	19.2	6.34
18.5	18.1	2.16
16.5	15.6	5.45
15.8	14.7	6.96
14	12.9	7.86

度  $W = x_2 - x_1$ ，高度  $H = y_2 - y_1$ 。若在取得預定特徵點時，目標為垂直地面放置，攝影機與地面保持水平且沒有滾動(Roll)角度，則此時目標左上與右下角點距離攝影機的景深均為  $z$ ，即  $z_1 = z_2 = z$ 。由(3.3)式，可推得

$$\begin{cases} u_1 = \frac{\lambda_x x_1}{z} \\ u_2 = \frac{\lambda_x x_2}{z} \end{cases} \Rightarrow (u_2 - u_1) = \frac{\lambda_x}{z} (x_2 - x_1) \Rightarrow \lambda_x = \frac{z}{(x_2 - x_1)} (u_2 - u_1) = \frac{z}{W} (u_2 - u_1) \quad (3.4)$$

同理可得

$$\lambda_y = \frac{z}{(y_2 - y_1)} (v_2 - v_1) = \frac{z}{H} (v_2 - v_1) \quad (3.5)$$

因此在校正時，我們只要知道黏貼在目標物體上的影像樣版在空間中實際的寬度與高度、目標物體距離攝影機的景深，加上此時所取得的影像特徵點座標，即可得到  $\lambda_x$  和  $\lambda_y$  的數值。經由實驗校正後所得到的參數數值如表 3.3。



表 3.3 攝影機參數

$\lambda_x$	145.03
$\lambda_y$	149.04

## 第四章 模擬與實驗結果

本章主要目的是為了驗證第二章所設計的伺服控制器，並整合第三章所介紹的影像特徵點擷取方法來設計機器人的抓取行為。由於本論文中，特徵點擷取的部分是利用實驗室所開發的一套嵌入式影像處理平台來完成，因此在實驗前將先介紹此平台與整個機器人的硬體架構。第一個實驗與模擬為引導機器人接近目標物體，藉以驗證(2.32)式的控制器為有效。由於攝影機架設方式的限制，使用固定的影像特徵點無法讓機器人到達可以抓取到目標的距離，因此本論文使用兩種影像特徵點，分別套用到(2.32)式的控制器，成為兩個引導機器人接近目標的行為，並利用 Subsumption 架構機器人結合接近與抓取行為，以第二個實驗驗證此架構能讓機器人成功抓取到目標。第三個實驗為測試機器人在物體移動時，是否能有效追蹤目標。



### 4.1 硬體介紹

本論文中的影像特徵點擷取部分採用實驗室學長所開發的嵌入式影像處理平台(圖 4.1)[1]來完成。此平台由以下四個主要元件構成：

#### (1) DSP 板：

DSP 板的核心為 TI TMS320C6416，時脈為 600MHz，具有 64KB 的內部 L2 快取記憶體與 256KWords 的外部記憶體(SDRAM)。C64xx 系列為定點 DSP 處理器，特別適合用來作影像處理方面的應用。

DSP 板在平台中主要負責處理影像資料，在本論文中，影像特徵點擷取的程式即是在此 DSP 板上運作。

#### (2) FPGA 板：

晶片型號為 Altera FLEX10K70RC240，其邏輯閘數目最高可達 70000。FPGA 板的主要負責初始化 CMOS 影像感測器，設定其中的各

項參數，包括：自動曝光、自動白平衡、影像銳利化、Gamma 校正、亮度校正等。另外也負責產生各項時脈與控制訊號，例如，frame buffer 的 read, write enable 訊號，觸發 DSP 板接收影像資料的同步訊號等。

(3) CMOS 影像感測板[12]：

此感測板上的 CMOS 影像感測器為 IC-Media 公司的 ICM - 205B 影像感測器。感測板輸出的影像格式為 16bit YCrCb 彩色影像，解析度為 640x480，畫面更新率每秒 30 張。

(4) 子電路板：

子電路板包含一顆 FIFO 暫存記憶體 AL422B[13]，容量為 384KB，負責暫時儲存 CMOS 影像感測器所取得的影像資料，等待 FPGA 讀取。板上另一部分為四個利用 DSP 上的 GPIO 模擬出的 RS-232 介面，作為嵌入式影像處理平台與其他裝置(如 PC)的連結介面。

此嵌入式影像平台取得的影像規格為 640x480 的灰階影像，畫面更新率最高為 30 frames/second。將 3.3 節所介紹的影像特徵點擷取程式在影像平台上運作時，畫面更新率為 7~8 frames/second，即 IPC 每秒鐘可收到 7~8 次影像特徵點資訊，其中包含沒找到特徵點的資訊。

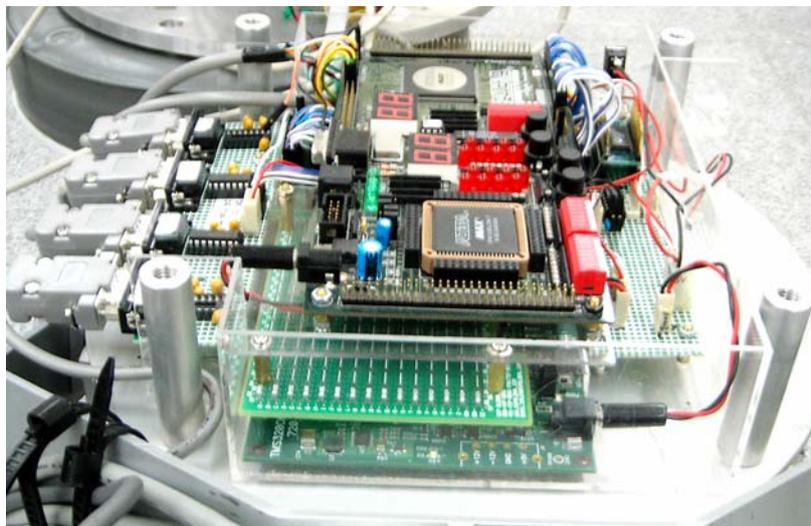


圖 4.1 嵌入式影像處理平台

圖 4.2 為機器人的硬體架構圖，主要的控制主體為 IPC，透過四個 RS-232 介面分別和三張 DSP 馬達控制卡與嵌入式影像平台連接。影像資料經過嵌入式影像處理平台取得影像特徵點座標後，透過 RS-232 介面傳給工業電腦(IPC)，由 IPC 上的控制程式產生各馬達所需的速度命令，並傳送給 DSP 馬達控制卡，由控制卡對馬達做速度回授控制，同時，控制卡會將馬達的軸編碼器訊號回傳給 IPC，以便紀錄機器人的行進軌跡與計算 Image Jacobian 所需的機械臂兩軸角度。

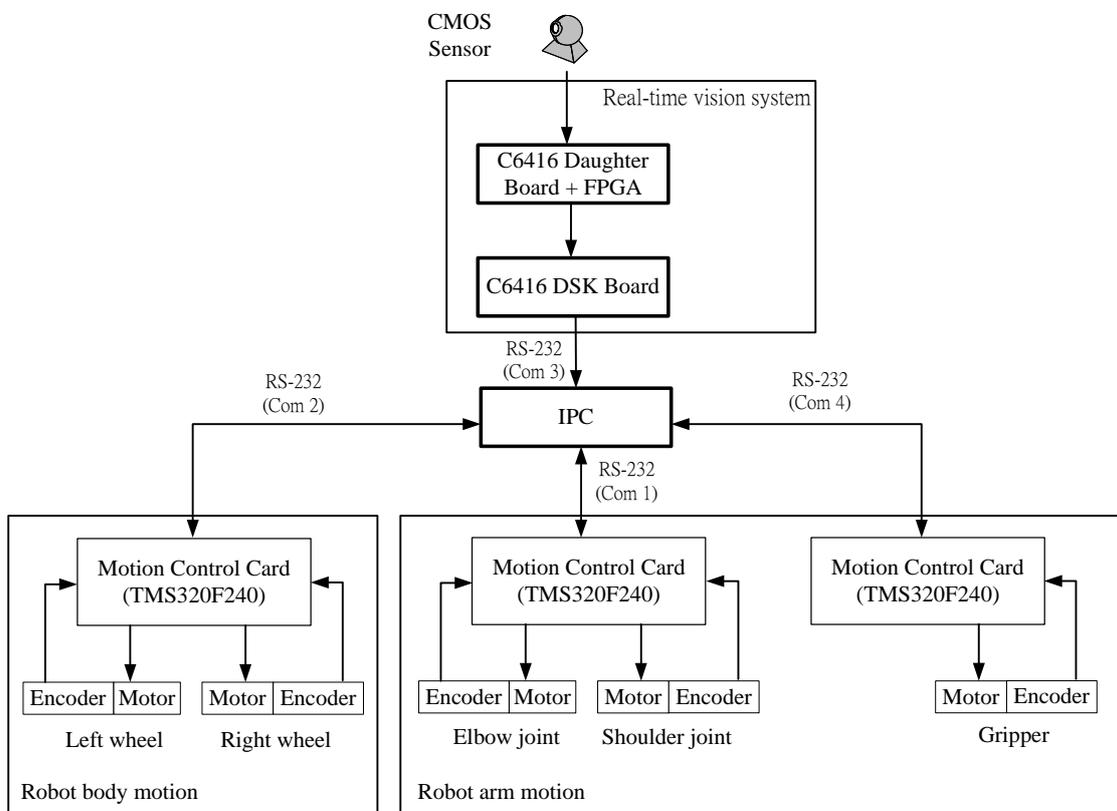


圖 4.2 機器人硬體架構圖

## 4.2 視覺伺服模擬與實驗

本節的模擬與實驗為驗證第二章所提出的控制器為有效之設計。其中，視覺伺服控制器為(2.32)式，整個 IBVS 伺服架構如圖 4.3 所示。

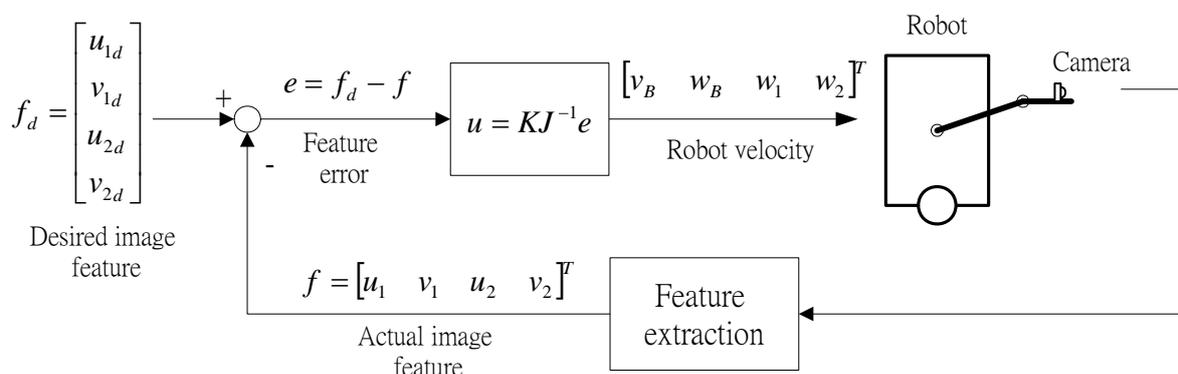


圖 4.3 IBVS 伺服架構



### 4.2.1 模擬結果

在第一個模擬中，預定到達的影像特徵點位置(Desired feature position)為將目標影像放置在機器人本體前方 1 公尺處取得影像特徵點。假設世界座標系為機器人初始狀態時的機器人座標系，則機器人初始位置座標為(0, 0, 0)，目標物體座標為(1, 0.05, 5)m，機器人手臂長度參數和實體相同(不考慮攝影機架設誤差  $\theta_x$ )，影像取樣頻率設定為理想的 30 frames/sec，即取樣週期為 0.033sec，並假設每次都能成功取到目標特徵點，距離估測採用 3.4 節所介紹的方法，假設兩個特徵點和鏡頭的距離相等。

模擬的結果如圖 4.4(a)~(d)所示。控制器有效地在影像平面上的特徵點移到預定到達的位置(圖 4.4(a))，且機器人也同時向目標物體靠近(圖 4.4(c))。

在第二個模擬中，所有的模擬條件都和前一次模擬相同，唯有特徵點和鏡頭的距離採用真正空間中的距離資訊，而非利用 3.4 節中的方法估測出的資訊，同

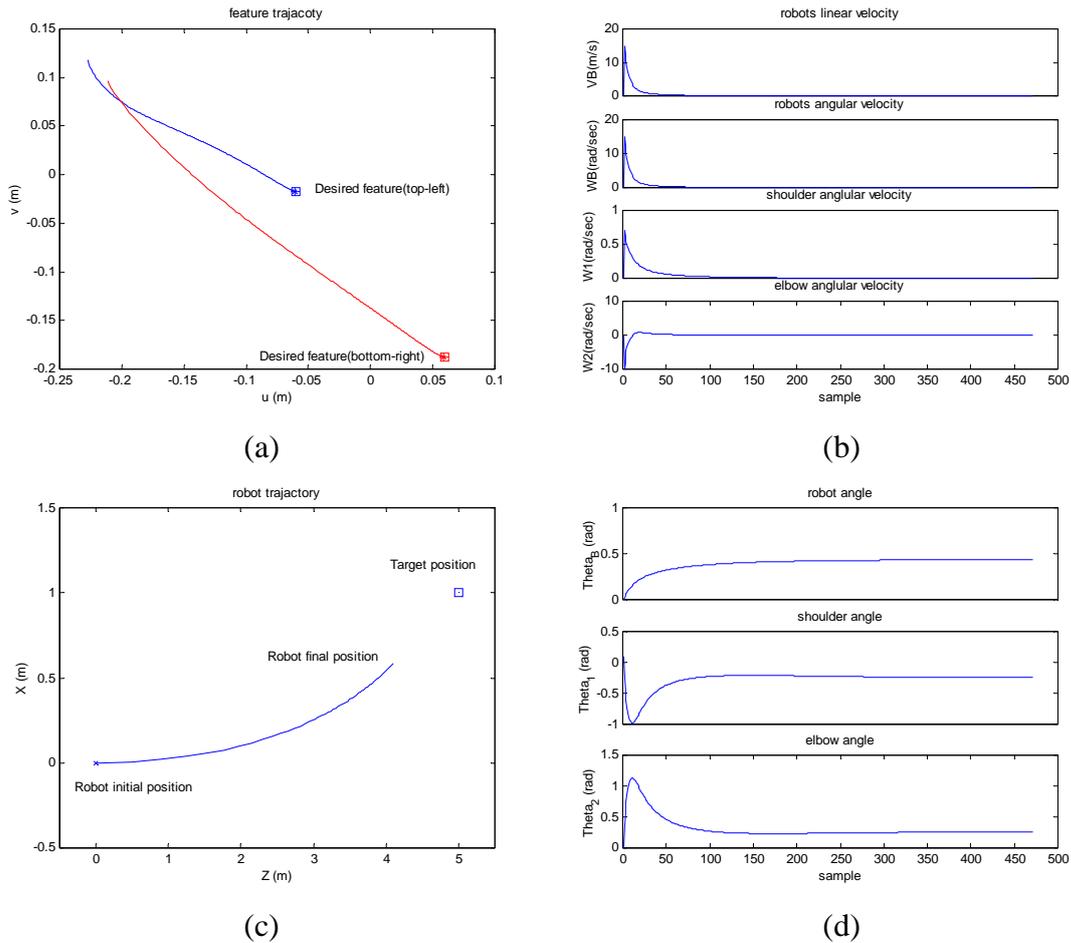


圖 4.4 視覺伺服模擬結果 1：(a) 影像特徵點移動軌跡；(b) 機器人本體角速度、手臂兩軸之角速度；(c) 機器人移動軌跡；(d) 機器人本體朝向角、手臂兩軸角度

時兩特徵點和鏡頭的距離也不再視為相等。其模擬結果如圖 4.5(a)~(d)所示，其中，影像特徵點軌跡和前次模擬稍微不同，但是由圖 4.4(d)與圖 4.5(d)比較可發現，手臂兩軸的角度響應則比前次模擬更好，第二個模擬中角度響應的 overshoot 和 undershoot 幅度都比第一次模擬小。

由以上兩個模擬中，我們知道不將兩特徵點與鏡頭的距離視為相等時，機械臂的角度響應會比視為相等時更好，但是因為在擷取影像特徵點時，尋找角點所做的 SSD Correlation 運算需要大量的運算時間，若利用 3.4 節所介紹的距離估測方法，我們需要額外找出目標影像的另外兩個角點(右上、左下)，並利用左上和右上角點計算出的目標距離作為左上角距離，利用左下和右下角點計算出的目標

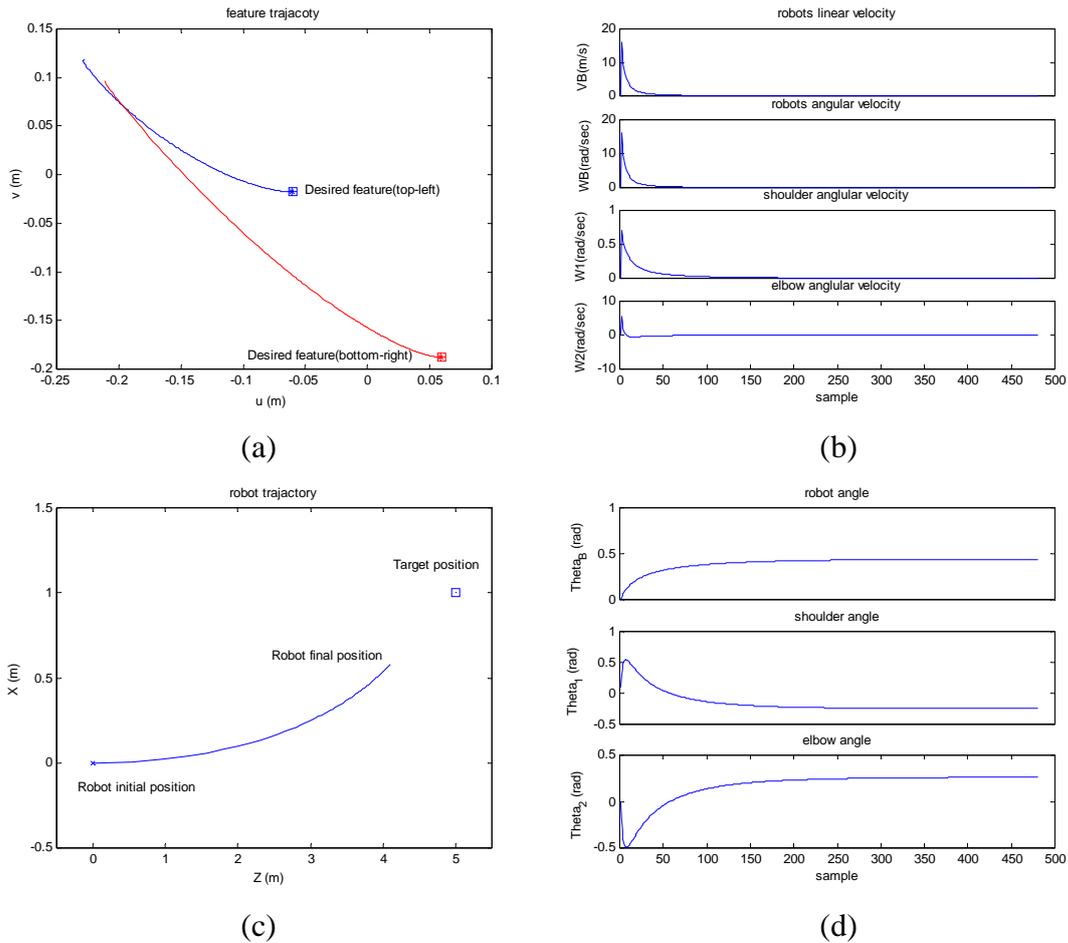


圖 4.5 視覺伺服模擬結果 2：(a) 影像特徵點移動軌跡；(b) 機器人本體角速度、手臂兩軸之角速度；(c) 機器人移動軌跡；(d) 機器人本體朝向角、手臂兩軸角度

距離作為右下角距離，經過實際測試，這樣做會使得影像處理部分的 Frame rate 降低到 5 frames/sec 以下。因此，為了維持影像特徵點擷取的速度，之後的實驗中還是採用將兩個特徵點與攝影機的距離視為相等的做法。

另外，由圖 4.4(d)與圖 4.5(d)中，可發現兩次模擬的手臂兩軸角度都有很大的 overshoot 和 undershoot，這似乎是控制器的特性，雖然可以藉由降低 Gain matrix 中的數值來消除，但是這麼做當特徵點靠近預定位置(Desired feature position)時所產生的速度命令會變得太小，無法控制機器人靠近目標。

## 4.2.2 實驗結果

實驗時，機器人起始位置為(0, 0, 0)m，目標物體位置約在(0.32, 0.37, 1.26)m，如圖 4.6。圖 4.7(a)~(d)為實驗結果，由圖(a)的特徵點軌跡圖中，藍色與紅色軌跡分別是左上與右下角點的運動軌跡，可看出特徵點確實往預定特徵點位置(方形點)靠近。在本實驗的程式中，有設定目前特徵點與預定特徵點位置誤差小於某個臨界值內便使機器人停止，因此最後軌跡沒有完全到達預定特徵點。由圖 4.7(a)中可發現特徵點軌跡有跳動的現象，推測可能的原因有兩個，第一是因為機器人的前後兩個萬向輪(Caster)與地面並非完全貼齊，因此在降速與加速時，將會使機器人稍微往前傾或往後傾，造成攝影機搖晃；第二個原因可能是不穩定的光源或不完美的二值化過程，造成目標影像的角落不平整，使得有多組不同的角點都能被辨識為特徵點，造成特徵點軌跡的跳動。

在此實驗中，機器人成功地由初始位置往目標物體移動，並在攝影機如我們所預期的在距離目標約 30 公分處停止(圖 4.7(b))，因為實驗中所使用的預定特徵點是將目標放在攝影機前 30 公分處時所取得的特徵點。圖 4.7(d)由上而下為機械臂兩軸的角度與攝影機傾斜角，由圖中可看出攝影機大致上會保持水平。



圖 4.6 機器人初始位置

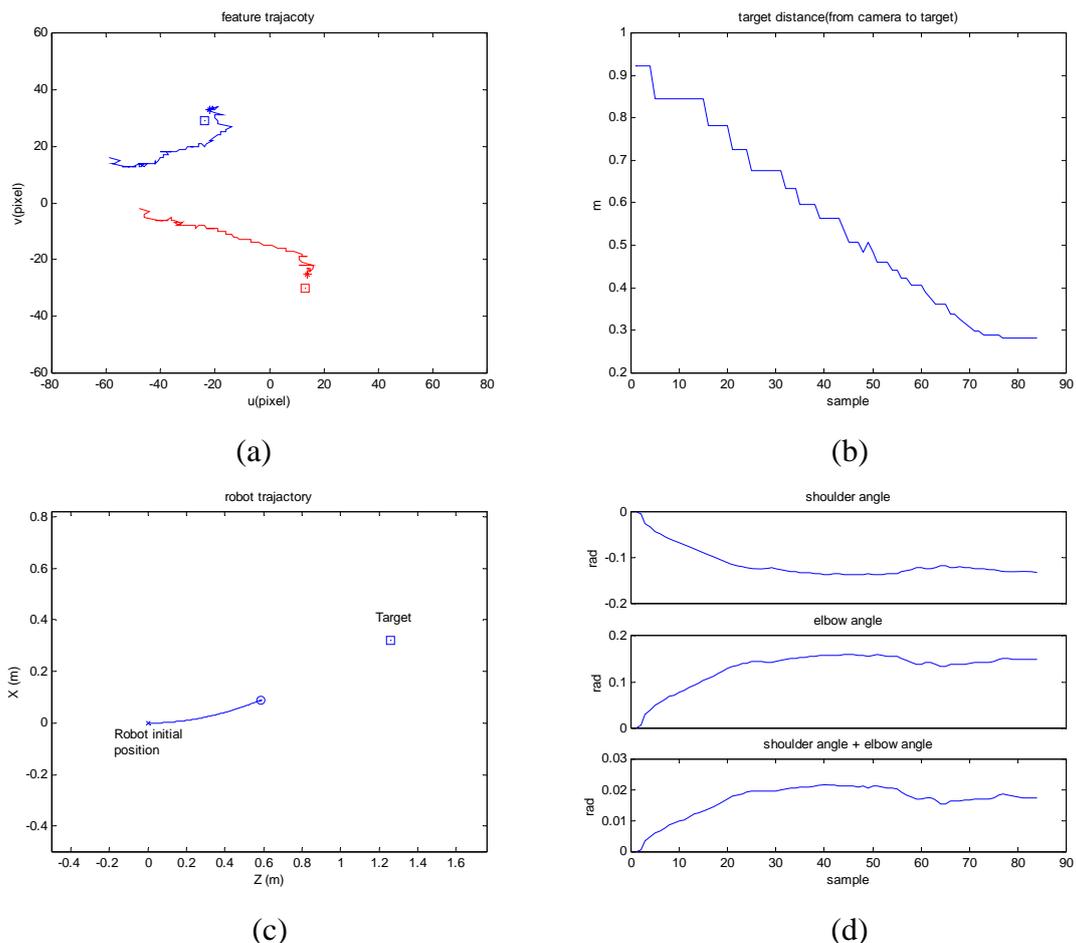


圖 4.7 視覺伺服實驗結果：(a) 影像特徵點移動軌跡；(b) 特徵點距離估測結果；(c) 機器人移動軌跡；(d) 手臂兩軸角度、攝影機 tilt 角度

### 4.3 抓取靜止目標物實驗

由前一節的實驗與模擬結果，我們得知(2.32)式為有效的控制器設計，可以引導機器人接近目標物體，接下來可以利用此控制器進一步讓機器人能抓取到目標。

因為攝影機使用 Eye-in-hand 的架設方式，遠端特徵點在攝影機太靠近目標時無法使用，而近端特徵點因為影像系統解析度的限制，在離攝影機距離太遠時無法被辨識出來。為了協調近端與遠端兩個控制器，我們利用 Behavior-based 架構中的 Subsumption 架構[10]來設計機器人的抓取行為：由於機器人距離目標遠

近會決定視覺伺服系統所用的影像特徵點，若抓取目標為移動式的物體，我們無法預測機器人會在何種距離下偵測到目標物體。利用 Subsumption 架構，可以讓機器人在不同的目標距離下快速切換適當的控制器行為，提高成功抓取的機會。

抓取行為的 Subsumption 控制架構圖如圖 4.8 所示，其中機器人利用三種行為控制器的協調來達成抓取的目的：Approach\_far, Approach\_near, Grasp。Approach\_far 和 Approach\_near 分別為機器人與目標物體距離較遠與較近時，控制機器人速度的行為，兩者同樣是利用(2.32)式所設計的控制器，只是兩者所用的影像特徵點不同，Approach\_far 用的是遠端特徵點(如圖 3.5(a))，Approach\_near 則是用近端特徵點(如圖 3.5(b))，另外，兩者的 Gain matrix 也不相同。我們利用近端特徵點需要在攝影機與機器人距離約為 30~40 公分左右時才能被辨識出來的特點，設計出如圖 4.8 中的架構。當物體距離較遠時，影像系統只能擷取到遠端的特徵點(圖 4.9(a)~(c))，因此由 Approach\_far 行為控制機器人往目標靠近；當接近到與目標的距離進入 30~40 公分內時，影像系統將開始辨識出近端特徵點(圖 4.10(a)~(c))，此時 Approach\_near 開始作用，並切斷原本 Approach\_far 的控制輸出。當機器人繼續往目標靠近時，因為攝影機與目標距離很短，因此將只能擷取到近端特徵點(圖 4.11(a)~(c))。當 Approach\_near 引導機器人移動到可抓取範圍內時，Grasp 行為開始動作，讓機器人往前行進一小段距離並閉合手爪以拿取目標物體。其中可抓取範圍的判定是依據近端特徵點與預定近端特徵點的 1-norm

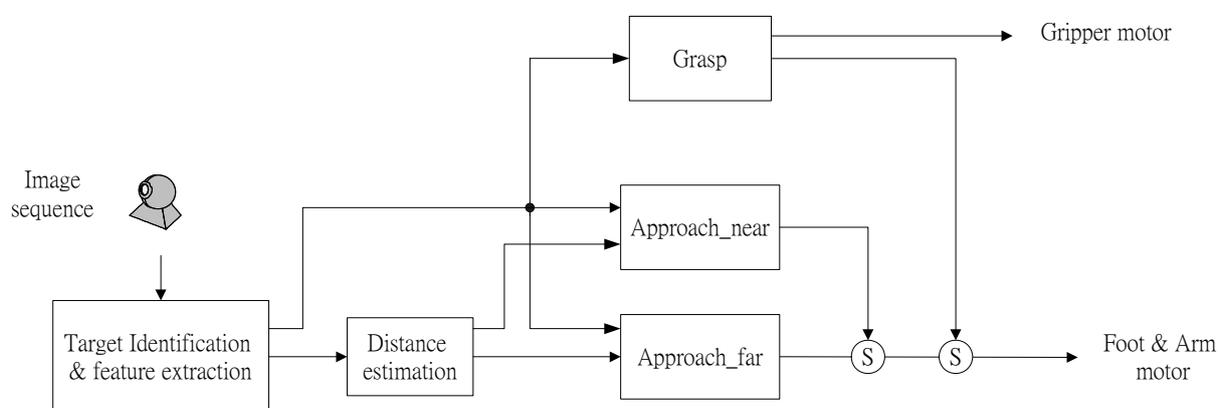


圖 4.8 抓取行為控制架構

值大小，當兩個特徵點的 1-norm 值均小於一固定閾值時，代表手爪已經極為靠近目標物體，因此可讓 Grasp 動作，控制機器人進行抓取。

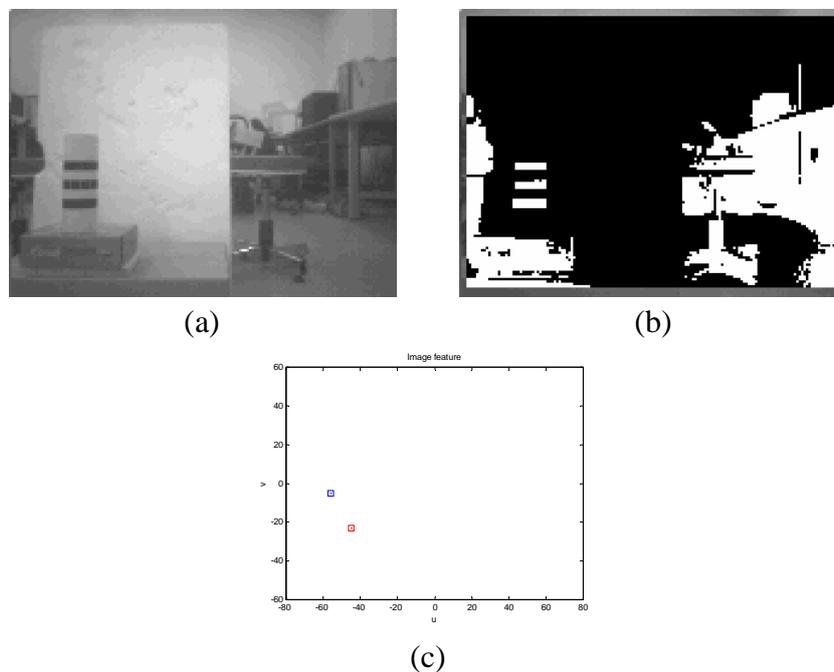


圖 4.9 特徵擷取過程的影像與所得結果 – 只擷取到遠端特徵點

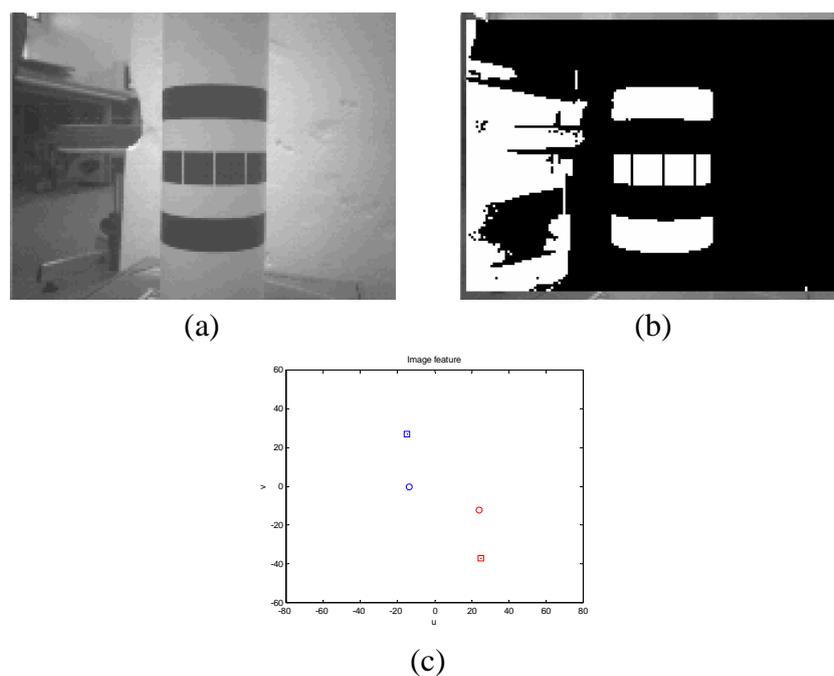


圖 4.10 特徵擷取過程的影像與所得結果 – 同時擷取到遠端與近端特徵點

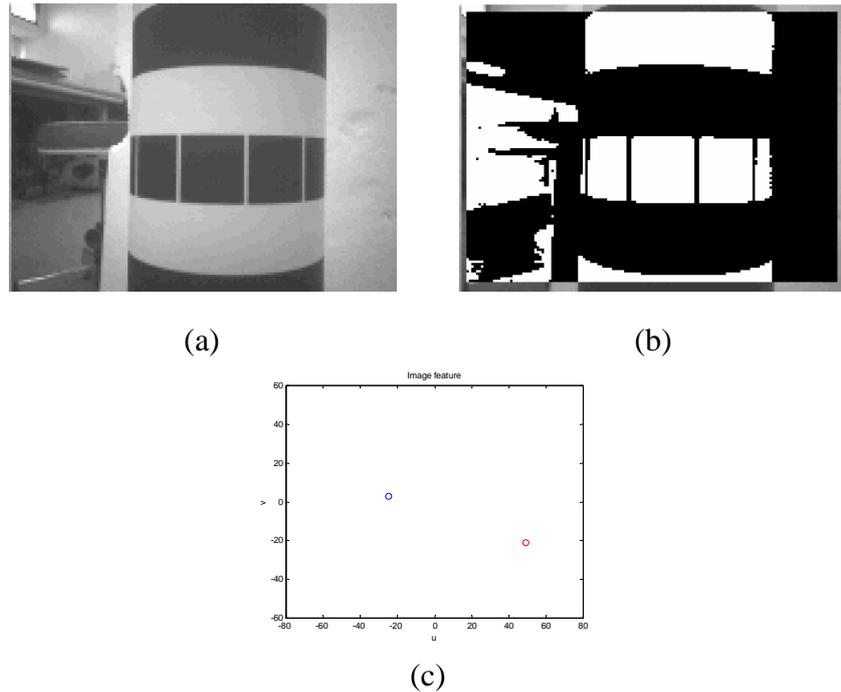


圖 4.11 特徵擷取過程的影像與所得結果 – 只擷取到近端特徵點

圖 4.12 為抓取靜止目標物之實驗流程，圖 4.12(a)為機器人與目標物體的初始位置，以此時機器人座標系作為世界座標系，則機器人位於(0, 0, 0)m，目標物體放置點約在(0.32, 0.37, 1.26)m。手臂初始狀態為水平，兩軸角度為0，如圖 4.12(b)所示。如圖 4.12(c)~(f)，一開始由 Approach\_far 動作，控制手臂與機器人本體行動。當機器人靠近到可看見近端特徵點時，改由 Approach\_near 行為控制機器人動作，如圖 4.12(g)(h)。當影像特徵點與近端特徵點小於閾值時(圖 4.12(i))，機器人由 Grasp 控制，前進一段距離，並抓取目標。

由圖 4.13(a)為影像特徵點的移動軌跡，藍色與紅色軌跡分別為遠端特徵點的左上和右下角點，此軌跡是由 Approach\_far 行為控制機器人行動時所造成；綠色和紫色軌跡分別為近端特徵點的左上和右下角點，此軌跡是由 Approach\_near 行為控制機器人行動時所造成。由圖中可以明顯看到 Approach\_far 和 Approach\_near 兩種行為所造成的特徵點軌跡都隨著機器人的行進，逐漸靠近各自的預定特徵點位置(方形點為遠端的預定特徵點，圓形點為近端的預定特徵點)。同樣在距離估測部分(圖 4.13(b))，藍色為 Approach\_far 控制機器人時所估

測出的距離，紅色為 Approach\_far 控制機器人時所估測出的距離，可以看到由兩種行為的切換，成功地將攝影機引導至距離目標物體只有約 12 公分處，此時手爪與目標之間大約只有 7~8 公分的距離，接著由 Grasp 行為控制機器人前進一段距離並抓取物體。圖 4.13(c)為機器人本體的移動軌跡，藍色軌跡由 Approach\_far 造成，紅色軌跡由 Approach\_near 造成，綠色軌跡由 Grasp 造成。圖 4.13(d)為攝影機在空間中的移動軌跡，藍色軌跡由 Approach\_far 造成，紅色軌跡由 Approach\_near 造成，綠色軌跡由 Grasp 造成。圖 4.13(e)由上而下分別是手臂兩軸的角度和攝影機的傾斜角(tilt angle)，由圖中可以看到攝影機大致上會維持水平。

#### 4.4 抓取移動目標物實驗

成功抓取靜態目標物之後，我們進行抓取移動目標的實驗，以測試控制架構對移動物體的追蹤能力。圖 4.14(a)~(h)為實驗進行流程，首先讓機器人辨識到目標物體(圖 4.14 (a))，並將目標稍微遠離機器人(圖 4.14 (b))，之後再將目標往機器人左前方移動(圖 4.14 (c)~(e))。從圖 4.14 中可以看到機器人能成功地追蹤並抓取移動的物體。

圖 4.15 為實驗結果，由圖 4.15(a)的距離估測結果可以看出機器人藉由 Approach\_far 和 Approach\_near 兩種行為成功導引攝影機到距離目標約 12 公分處，接著由 Grasp 行為控制機器人進行抓取。



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

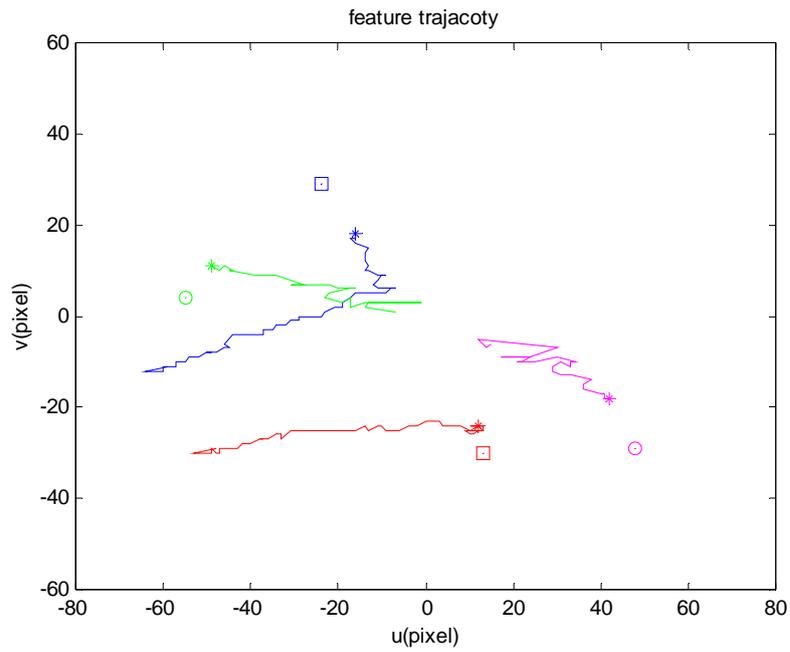


(i)

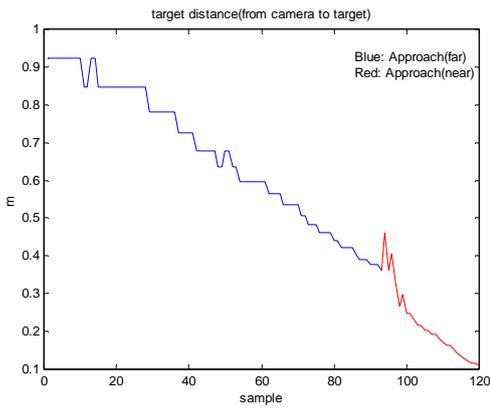


(j)

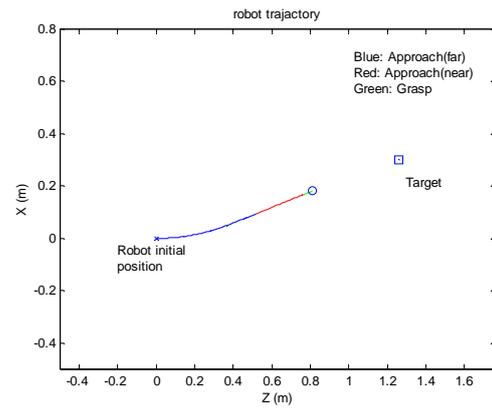
圖 4.12 抓取靜止目標實驗



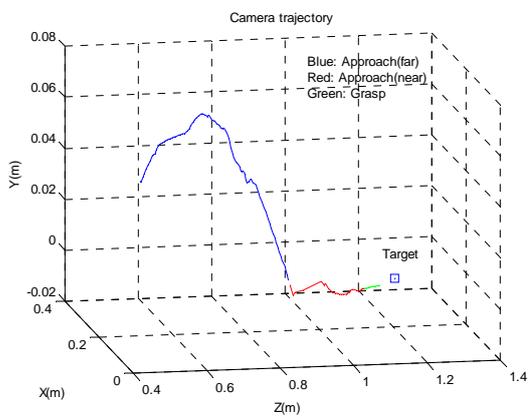
(a)



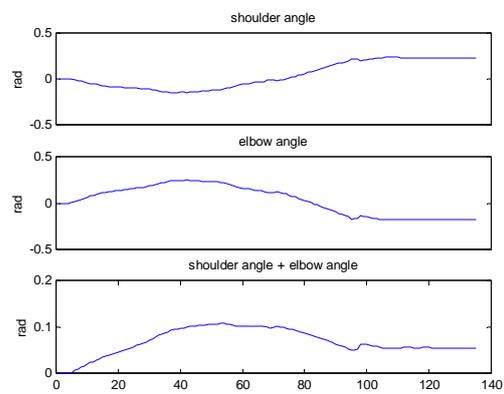
(b)



(c)



(d)



(e)

圖 4.13 抓取靜止目標物實驗結果：(a) 特徵點移動軌跡；(b) 目標距離估測結果；(c) 機器人移動軌跡；(d) 攝影機移動軌跡；(e) 手臂兩軸角度、攝影機 tilt 角度



(a)



(b)



(c)



(d)



(e)



(f)

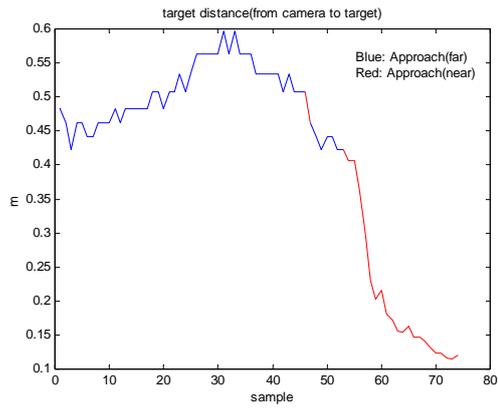


(g)

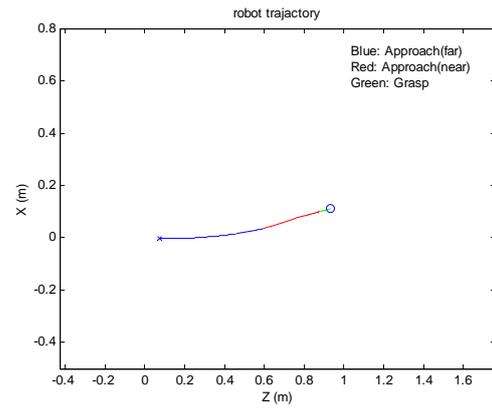


(h)

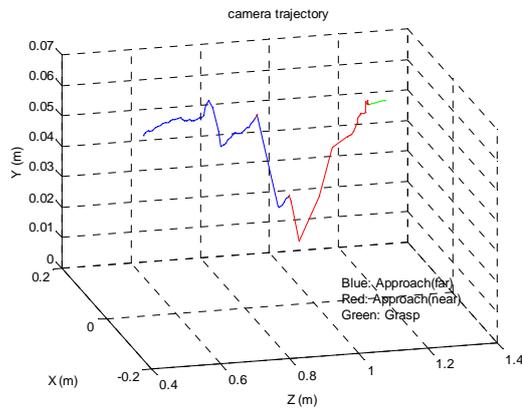
圖 4.14 抓取移動目標物實驗



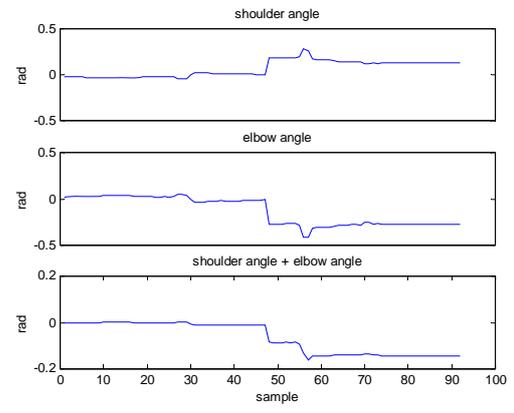
(a)



(b)



(c)



(d)

圖 4.15 抓取移動目標物實驗結果：(a) 目標距離估測結果；(b) 機器人移動軌跡；  
(c) 攝影機移動軌跡；(d) 手臂兩軸角度、攝影機 tilt 角度

## 第五章 結論與未來展望

### 5.1 結論

本論文利用 Image Jacobian 的概念結合機器人與攝影機間的 Jacobian 關係，發展出一個基於影像的視覺伺服控制器。此控制器藉由修正在影像平面上特徵點位置，能夠同時產生機器人移動平台與機械臂的速度控制命令，協調機器人上的機械臂到達與目標相對的特定位置。但是由於 Eye-in-hand 攝影機架設的限制，使得機械臂靠近目標的過程中無法使用單一影像特徵作為控制器輸入，因此本論文中利用兩個使用不同影像特徵的控制器，藉由 Behavior-based subsumption 架構協調兩個控制器的輸出，使得機械臂能到達極為靠近目標的位置，成功抓取到目標。由實驗結果中可看出，藉由兩個控制器的協調，可以讓手爪到達距離目標只有 7~8 公分的位置，在此位置下，機器人只需稍微前進即可抓取到目標。另外，在移動物體的實驗中也可看到，機器人對於移動的物體也能追蹤目標並成功抓取。而在影像系統方面，特徵點擷取頻率為每秒 7~8 次，能穩定擷取到影像特徵點的目標距離為與攝影機距離 80~100 公分的範圍內，大於此距離時因為物體在畫面上顯得太小，將會無法辨識與擷取特徵點。

### 5.2 未來展望

本論文在影像辨識時，為了加快辨識速度而將影像解析度降低到 160x120，這造成影像系統在攝影機與目標距離 1 公尺以上時無法辨識出物體，因為在 160x120 的解析度中，目標距離攝影機 1 公尺時的影像會變得很小，目標上的垂直條碼寬度大約只有 2~3 個像素寬度，這使得辨識第一階段中的角點偵測發生困難，因為我們使用的影像樣板為 5x5，已經大於垂直條碼的寬度了。因此若未來能改善影像辨識的方法，讓影像階析度不需要降到 160x120，便能讓影像系統有

更遠的視野。

在第 4 章的實驗與模擬中，在機器人剛開始抓取目標時，手臂的角度響應都有很大的 overshoot 或 undershoot，這點會限制機械臂能抓取的物體放置範圍，如果目標物體放在較高或較低的位置，手臂的 overshoot 與 undershoot 可能會使角度超過限制，間接縮小了機械臂可抓取物品的高度範圍。因此若能改善控制器，降低 overshoot 與 undershoot 的問題，可以擴大可抓取目標高度範圍。

在實用上，抓取物品的行為通常需要配合避障或導航的功能。例如要讓機器人幫忙拿物品，並指定目標所在的房間，機器人在到達該房間途中可能會遇到障礙物，因此需要有避障的能力。由於本論文所設計的抓取行為採用 Subsumption 架構，可以直接在底層整合避障與導航行為，如圖 5.1，即可讓機器人同時具備抓取與避障的能力。在抓取目標物的辨識方法上，本論文中的做法是利用貼在目標物體上的特殊類條碼影像樣板來作為辨識的依據，此辨識方法在實用上不夠一般化，如果能夠在不改變目標物體外觀的情況下擷取出所需影像特徵點，例如以物體的形狀與外觀影像來辨識，將會使抓取功能較為貼近實用。

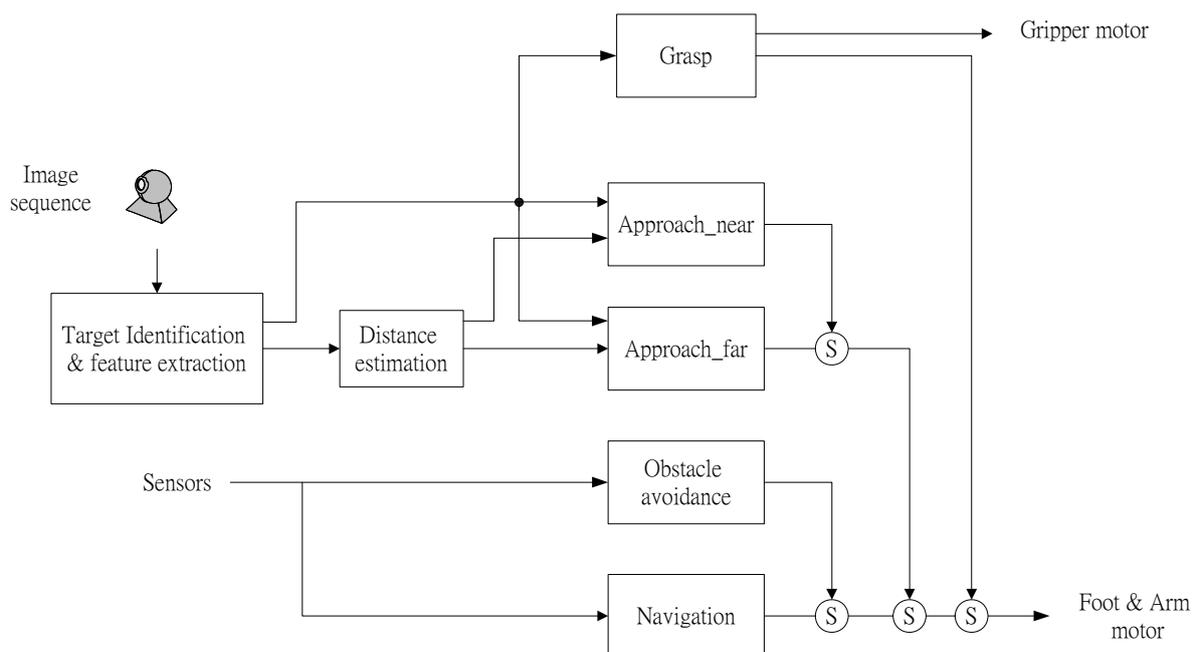


圖 5.1 整合避障與導航功能

## 參考文獻

- [1] Henry Andrian, "An embedded image processing platform for a mobile manipulator," Master's Thesis, National Chiao Tung University, Taiwan, Republic of China, July, 2004.
- [2] T. Martinez-Marin and T. Duckett, "Robot docking by reinforcement learning in a visual servoing framework," in *Proceedings of IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, 2004, pp. 159-164.
- [3] T. Takahama, K. Nagatani and Y. Tanaka, "Motion planning for dual-arm mobile manipulator - realization of "tidying a room motion", " in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004, pp. 4338-4343.
- [4] D.C. MacKenzie and R.C. Arkin, "Behavior-based mobile manipulation for drum sampling," in *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, USA, 1996, pp. 2389-2395.
- [5] J.H. Connell, "A behavior-based arm controller," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 6, pp. 784-791, 1989.
- [6] A. Matsikis, F. Schulte, F. Broicher and K.-F. Kraiss, "A behaviour coordination manager for a mobile manipulator," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, USA, 2003, pp. 174-181.
- [7] S. Hutchinson, G.D. Hager and P.I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670, 1996.
- [8] N.P. Papanikolopoulos and P.K. Khosla, "Adaptive robotic visual tracking: theory and experiments," *IEEE Transactions on Automatic Control*, Vol. 38, No.

3, pp. 429-445, 1993.

- [9] John J. Craig, *Introduction to Robotics: Mechanics & Control*, Reading, MA: Addison-Wesley, 1989.
- [10] Ronald C. Arkin, *Behavior-Based Robotics*, Cambridge, MA: MIT Press, 1998.
- [11] T. I. J. Tsay and C.J. Chang, "Pose control of mobile manipulators with an uncalibrated eye-in-hand vision system," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 3033-3038.
- [12] ICM205B, Datasheet, "VGA/QVGA CMOS image sensor with digital YUV output," Oct 2002, IC-MEDIA Corp.
- [13] AL422B Data sheet, Data sheet, 2004, Aver Logic Inc.
- [14] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transaction of Man-Machine Systems*, Vol. 10, pp. 47-53, 1969.
- [15] Richard P. Paul, *Robot Manipulators: Mathematics Programming and Control*, Cambridge, MA: MIT Press, 1981.