

國立交通大學

電機與控制工程學系

碩士論文

基於全向式影像之機器人同步定位
與環境地圖建立

Simultaneous Localization and Mapping Using
Omni-Directional Images

The logo of National Tsing Hua University is a circular emblem with a gear-like outer border. Inside the circle, there are stylized Chinese characters and the letters 'NSA'.

研究生：黃富聖

指導教授：宋開泰 博士

中華民國九十七年一月

基於全向式影像之機器人同時定位 與環境地圖建立

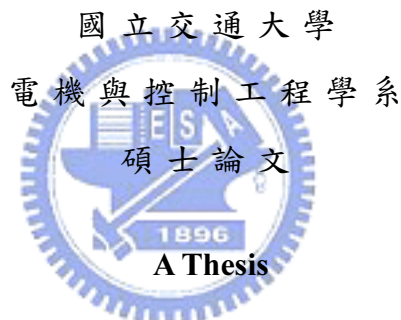
Simultaneous Localization and Mapping Using
Omni-Directional Images

研究生：黃富聖

Student: Fu-Sheng Huang

指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song



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

January 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年一月

基於全向式影像之機器人同時定位 與環境地圖建立

學生:黃富聖

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

本論文提出一使用全向式影像之機器人定位方法。以全向式攝影機為感測器，結合基於 Extended Kalman Filter(EKF)之同時定位與環境地圖建立(Simultaneous Localization and Mapping, SLAM)演算法，讓機器人在移動的同時，能夠建立起環境特徵地圖並定位出機器人本身的位置。全向式攝影機具有 360 度的視角，除了能取得更多的環境特徵外，亦能增加持續追蹤到 landmark 的時間，讓 SLAM 的運作更為穩定。配合攝影機的特性，本論文參考 Scale Invariant Feature Transform(SIFT)演算法發展出一有效之特徵點辨識演算法，用以辨識兩張相鄰的影像中相同的環境特徵點，此方法對於影像經過旋轉及大小縮放後依然擁有相當穩健的辨識。本論文提出一視覺參考點建立與轉換的策略，讓機器人進入新的環境時能建立新的參考點與地圖，當走回舊地區時則從資料庫中取回舊有的參考點資訊使用，減少參考點總數，降低 EKF 濾波器的運算負擔。論文中以實驗室之機器人進行導航實驗來驗證所提出之定位演算法，實驗結果顯示特徵點比對之正確率為 90%，行經 30 公尺後之定位誤差為 0.1 公尺。實驗的結果證實機器人能依定位系統的幫助在走廊上長距離的移動，並且同時建立起走廊環境的特徵點地圖，達成機器人室內導航的功能。

Simultaneous Localization and Mapping

Using Omni-Directional Images

Student: Fu-Sheng Huang

Advisor: Dr. Kai-Tai Song

Department of Electrical and Control Engineering
National Chiao Tung University

ABSTRACT

This study investigates robot localization and mapping using omni-directional images. A method is proposed to use an omni-directional camera to realize simultaneous localization and mapping (SLAM) algorithm based on extended Kalman filter (EKF). Because of the 360° field of view, an omni-directional camera is suitable for simultaneous localization and mapping (SLAM) for detecting and tracking environmental features. A new algorithm is developed adopting scale invariant feature transform (SIFT) method to match features in environment between two images. This thesis also presents a switching method of visual reference scans. In this method, reference scan can be added to a database or switched automatically among reference scans. These scans can be used repeatedly to reduce the complexity of extended Kalman filter (EKF). Experiments results show that the matching rate of landmark features is 90%. A long range indoor navigation experiment revealed that the proposed localization algorithm can help robot to navigate in indoor environment and build the features map simultaneously.

誌謝

謹向我的指導教授宋開泰博士致上感謝之意，感謝他兩年半來在專業研究上的指導，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。

感謝學長奇謚、嘉豪及孟儒在實做及理論上的指導，以及感謝與我共同奮鬥的同學振暘、俊瑋、濬尉及志昇的相互鼓勵及提攜，同時也感謝學弟弘哲、兆戊、煥坤、仕傑及裕宏在研究途中的協助及在生活上帶來的樂趣。

最後，特別感謝我的父母，由於他們的辛苦栽培，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

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
圖例	vi
表例	xiii
第一章 緒論	1
1.1 研究動機與目的	1
1.2 相關研究回顧	2
1.3 問題描述	4
1.4 系統架構	5
1.5 章節說明	5
第二章 環境特徵偵測	7
2.1 全向式攝影機概述	7
2.2 環境特徵點辨識	9
2.2.1 SIFT 特徵點擷取	11
2.2.2 基於全向式攝影機之特徵點擷取與比對演算法	12
2.2.2.1 特徵點擷取法則	13
2.2.2.2 特徵點描述	15
2.2.2.3 特徵點比對	18
2.2.2.4 Outlier 之判定	19
第三章 機器人定位	21
3.1 v-SLAM 簡介	21
3.2 特徵點初始化	23

3.2.1	水平位置估測.....	23
3.2.2	垂直高度估測.....	26
3.3	EKF-SLAM 演算法.....	28
3.3.1	系統設計.....	28
3.3.2	系統模型.....	29
3.3.3	演算法流程.....	33
3.3.3.1	Prediction step.....	34
3.3.3.2	Update step.....	36
第四章	參考點轉換.....	40
4.1	參考點比對.....	40
4.2	參考點轉換演算法.....	40
第五章	實驗結果.....	45
5.1	機器人硬體介紹.....	45
5.2	環境特徵點辨識實驗.....	47
5.3	EKF-SLAM 初步測試實驗.....	50
5.4	定位參考點轉換之測試實驗.....	52
5.5	機器人室內導航之情境展示實驗.....	56
5.6	討論.....	58
第六章	結論與未來展望.....	60
	參考文獻.....	61



圖例

圖 1-1	機器人定位系統演算法流程圖.....	6
圖 2-1	全向式攝影機示意圖(擷取水平面以下影像).....	8
圖 2-2	全向式攝影機示意圖(擷取水平面以上影像).....	8
圖 2-3	Vstone VS-C450N-TR 全向式攝影機.....	8
圖 2-4	全向式攝影機所擷取之影像.....	9
圖 2-5	基於全向式攝影機之特徵點擷取與比對演算法流程圖.....	11
圖 2-6	不同尺度空間之 DoG(Difference of Gaussian)示意圖[8].....	12
圖 2-7	相鄰影像中之物體大小變化比較.....	13
圖 2-8	機器人平移後影像中的物體大小變化比較.....	14
圖 2-9	Sobel 運算子.....	14
圖 2-10	角點偵測結果.....	15
圖 2-11	特徵區域旋轉示意圖.....	17
圖 2-12	特徵點描述向量之建立.....	18
圖 2-13	Outlier detector 示意圖[7].....	19
圖 3-1	Online SLAM 之圖示模型[14].....	21
圖 3-2	SLAM 演算法概念示意圖[13].....	22
圖 3-3	Landmark 位置估測之座標關係圖.....	24
圖 3-4	特徵點初始化流程圖.....	25
圖 3-5	Feature 點的平面座標估測圖.....	25
圖 3-6	機器人與 Landmark 位置關係側視圖.....	26
圖 3-7	視角校正之實驗影像.....	27
圖 3-8	視角-pixel 實驗結果及近似關係圖.....	27
圖 3-9	特徵點高度估測結果.....	28

圖 3-10	EKF SLAM 設計概念圖.....	29
圖 3-11	機器人運動模型表示圖.....	30
圖 3-12	機器人及 Landmark 之水平位置關係圖.....	33
圖 4-1	參考點影像負責區域示意圖.....	42
圖 4-2	參考點轉換策略之流程圖.....	43
圖 4-3	資料庫之結構示意圖.....	43
圖 5-1	機器人實體圖.....	46
圖 5-2	機器人硬體架構.....	46
圖 5-3	Vstone VS-C450N-TR 全向式攝影機.....	47
圖 5-4	特徵點比點結果.....	48
圖 5-5	特徵點比對結果(加入 Outlier detection).....	49
圖 5-6	特徵點比對結果(旋轉影像).....	49
圖 5-7	實驗場景.....	51
圖 5-8	繞行移動之實驗結果.....	52
圖 5-9	來回行走之實驗結果.....	53
圖 5-10	來回行走實驗中參考轉換之結果.....	54
圖 5-11	機器人 8 字型移動之實驗結果.....	55
圖 5-12	走廊移動實驗結果.....	57
圖 5-13	實驗擷圖.....	58

表例

表 3-1	EKF SLAM 演算法之步驟流程.....	39
表 5-1	特徵點比對實驗結果.....	50
表 5-2	來回行走之實驗結果誤差.....	55



第一章 緒論

1.1 研究動機與目的

移動式機器人所要擁有的主要功能之一即為自主式導航，其主要目的就是要讓機器人能自動的移動到設定的目標地點。為了要完成這項功能，機器人需具有能夠藉由偵測周圍環境的相關資訊知道本身的位置。知道自己本身的位置後才知道目標在哪個方向、距離多遠，機器人才能往目的地前進；在前往目的地的路途中，環境中不可能空無一物，因此機器需要具有對環境偵測的能力，才能順利的走到目標地點。

近年來，有許多研究者專注在同時定位與地圖建立(Simultaneous localization and mapping, SLAM)的研究，SLAM 主要的內容就是機器人之同時自我定位及環境地圖建立，解決了 SLAM 的問題，實際上也幾乎解決了機器人導航的問題，也因為 SLAM 能同時解決了機器人導航的兩大問題，故才受到了許多人的重視。SLAM 主要的方法是利用機器人對本身運動的預測及感測器對環境的估測融合所完成的，最常用的環境感測器即是距離感測器，如超音波、雷射掃描儀等。近年來攝影機這種視覺感測器也常被使用，雖然單一攝影機無法測得距離資訊，但影像中的環境資訊更為豐富，可擷取到更多的環境特徵。基於視覺的 SLAM，也被稱作為 vSLAM (Vision-based SLAM)，不同於基於環境距離特徵，vSLAM 則是以影像平面中的特徵點為 Landmark。

在攝影機當中，全向式攝影機(Omni-directional camera)常被用於多機器人系統中，此種攝影機系統以一凸面鏡及單眼攝影機所組成，周圍 360 度的環境影像會透過凸面鏡的反射，而全被攝影機所拍攝到，因為具有 360 度的 Field of view (FOV)，可在第一時間擷取到週遭的影像資訊，所以在機器人的導航上也更有幫助。也因為全方位的視角，能同時取得更多的 Landmark，讓系統更穩健，且增加持續追蹤到 Landmark 的時間，讓打滑及訊號漂移的影響降到最小。

本論文的研究目的即是希望能夠藉由全向式攝影機豐富的影像資訊及 360°

的視野，幫助機器人在室內環境完成自我定位的功能。配合自主移動的行為，讓機器人能移動至設定的地點，完成機器人在室內走廊環境下的巡航及建立環境地圖之功能。

1.2 相關研究回顧

SLAM 是同時定位出機器人位置及建立環境地圖的一種演算法，定位及地圖建立這兩種功能亦是自主式機器人最重要的功能之一，故在機器人導航的研究中 SLAM 亦受到許多研究者的重視。

在 SLAM 的初期研究中，距離感測器是最常被使用的，如超音波(Ultrasonic)[16]或是雷射掃描儀(Laser scanner)[15]等等，近年來最常被使用的就屬雷射掃描儀；雷射掃描儀之類的距離感測器最大的優點就是能夠直接取得周圍環境的距離資訊，這對於建立地圖或是定位來說都是非方便且直接的感測器資訊，但缺點就是此類的感測器通常都較為昂貴。

在近期的 SLAM 研究中，開始有許多研究所使用的感測器從雷射轉換成視覺感測器[2]-[7]，稱之為 vSLAM。使用攝影機感測器其價格較為便宜，沒有了雷射掃描儀昂貴的問題，並且能在影像平面中取得更豐富的环境資訊與特徵，但是相較之下使用視訊的話，需要的較複雜的影像處理演算法，消耗較多的運算能力。而近代電腦運算能力的快速進步也較能負擔影像處理的能力，使得基於影像為基礎的 SLAM 演算法興盛了起來。

在影像平面中要找出環境特徵的方式有許多，像是[1]中即是直接對環境中已有的特定物體或是圖樣作偵測，像是日光燈、門...等，此種方式雖然較為直覺簡易，但卻只能適用於某些特定場地，不具有一般性；使用環境中自然的特徵(Nature landmark)則是近來的趨勢，像是環境中物體的角落、邊緣均是不錯的特徵，現在亦有許多特徵點的偵測辨識方可使用，而 Lowe[8]所提出的 Scale invariant feature transform (SIFT)演算法即是最近較受重視的主題之一，如[3][6]均是以 SIFT 特徵點為基礎來發展機器人定位的 SLAM 演算法。SIFT 演算法主

要是以 Difference of gaussian (DoG)法來擷取環境中穩健的特徵點，並對多種不同大小解析度的影像也擷取出特徵點，特徵點的描述亦具有其方向性，讓特徵點的辨識具有對抗影像大小變化及旋轉的能力，使特徵點的辨識較為穩健，即是此方法受到重視的主因之一。

而使用電腦視覺最大的問題就是在於無法即時的偵測環境的距離，[4]就是利用 Structure from motion (SFM) algorithm 來對環境距離作估測，SFM 主要是結合相鄰數張的影像，重建出攝影機的移動及環境結構，利用特徵點在影像 2D 平面上的位置，結合運動及攝影機本身的校正參數，推測出 3D 環境中的相對關係，重建出 3D 景物，在 SFM 中串列影像的間隔時間通常較短，因此也具有讓特徵點比對、追蹤演算法能較為簡易的優點；[8]中雖然是使用一般的單眼攝影機，但是導入了 Inverse depth 的概念，在一般的特徵點估測及初始化方法中，較遠距離特徵點的距離估測的準確度較低，通常都會捨棄不列入定位演算法的運算中，但遠距離特徵點卻是對機器人朝向角估測最有代表性的，Inverse depth 的概念能在一個影像 Frame 中定義出特徵點的相對距離及角度，並對環境特徵做 State 及 Covariance 的初始化，即使相對距離的不確定性很高，但相對角度卻有一定的可靠度，在 SLAM 中能對機器人的朝向角修正有所貢獻。

相對於單眼攝影機，擁有兩台攝影機的立體視覺系統，即能在第一時間利用兩個攝影機所拍攝到影像決定出特徵點的距離資訊。[6]就是使立體視覺的例子，其中利用了立體視覺能估測畫面景深的特性，對影像中的特徵距離估測，並近似成像 Laser scanner 那樣的距離 Scan，並套用了雷射定位中常見的 Scan-matching 的觀念，機器人每隔一段距離便建立 Reference scan，利用 Reference scan 與即時畫面中的 SIFT 特徵比較，辨識環境特徵點，定位出機器人與參考點的相對位置及建立環境地圖。

在使用視覺資訊來建立環境物理位置資訊時，由於攝影機視角有限的關係，如何整合並 Match 各畫面中的特徵是個重要的問題。全向式攝影機具有 360 度全方向的視角，能同時取得更多的環境特徵，且增加追蹤到特徵的時間，且在

特徵點的描述上亦不會受到旋轉的影響，增加比對的成功率，因此對於基於視覺的 SLAM 來說是個很好的感測器。[1]是在影像中偵測出走廊環境中最常見的物體“門”當做 Landmark，以門的色彩資訊做為偵測的依據，利用影像平面中門的底部與影像中的距離依實驗所求得之轉換式估測其位置作為 SLAM 演算法的輸入。[3]則是使用 SIFT 演算法來作特徵點的比對，在每隔一小段距離便將影像記錄下來，之後移動時即從資料庫中找出最相近的影像，利用 SIFT 特徵與資料庫影像中 Local feature 比對，並使用 Particle filter 完成機器人定位的功能。[4]先是利用基於全向式攝影機的 SFM 來對環境結構作估測，利用結合數張影像整合得知，最後使用 EKF 濾波器完成 SLAM 演算法。以上數篇文獻其特色均是善用全向式攝影機的優點，並修改故有的演算法，除了是要配合感測器的不同，更能發揮出全向式攝影機的優勢，增加機器人定位的穩健度，如[4]使用基於全向式攝影機的 SFM 能建立更大視角範圍的環境結構，增加可視特徵點數目，更能持續追蹤特徵點。



1.3 問題描述

由文獻中可得知，基於視覺的 SLAM 研究，首先最重要的問題就是如何在影像平面中偵測出環境特徵點並作位置的估測，但對於全向式攝影機來說，其視覺模型與傳統的單眼攝影機不同，因此勢必要對舊有文獻中的方法修正，以適合全向式攝影機的使用，再者，更要對全向式攝影機做有效的利用，以展示其優勢所在。在全向式影像中要如何對故有的演算法做修正，有效的偵測辨識出環境特徵點，接著估測出其空間位置，使其能有效的幫助機器人定位，即是在基於全向式影像的機器人定位議題中首先要解決的問題。

在使用影像感測器的 SLAM 研究中，如[6]中使 EKF 演算法，會因所觀測到的特徵持續增多，而造成運算的複雜度增加、運算的效能低下。[6]中因為其攝影機的視角有限，故旋轉角度超過限定值時則需要再建立一個新的參考點，若機器人走回之前走過的地點時，仍需再次建立新的參考點，隨著參考點的增加，

EKF 中 State vector 及 Covariance matrix 會一直持續的增大，增加運算的複雜度。為此本論文希望能設計出基於全向式攝影機之 EKF-SLAM 機器人定位方法，藉由全向式攝影機 360°的視角，減少參考點建立的數量，並建立參考點轉換及重覆利用的機制，降低運算複雜度持續增加的問題之影響。

1.4 系統架構

本研究使用實驗室自製的機器人作為測試平台，機器人本體建構在一輪式運動平台上，此移動平台採用雙獨立驅動輪式之運動機構，利用對兩馬達的速度控制實現機器人在平面上之運動；移動平台後方另有一被動式自由輪，為三輪式的運動平台設計。所使用的主要感測器為全向式攝影機，使用 SIFT 演算法將即時影像與參考點影像比對，藉以估測出同一特徵點的位置資訊；對已估測出之特徵點判別，若是第一次見到的則加入環境地圖中，若是已存在的特徵點，則將特徵點的觀測資訊與 Odometer 的訊號結合，利用 Extended Kalman filter 將資料融合，同時更新機器人本體座標及環境特徵點地圖的位置；最後利用定位出的機器人位置及環境地圖，配合移動控制法則，即可讓機器人走向目標地點，完成機器人的導航功能。

為了延長機器人定位的距離範圍，本論文設計一參考點建立及轉換的流程，當機器人移動到新地點時則建立此處的參考點影像，回程時則從資料庫中取回之前已存入的參考點資訊，如此將環境中不同地點的全向式影像特徵整合，讓機器人能在大範圍環境中持續重覆行走，達成機器人長距離導航的功能。圖 1-1 即為本研究之機器人定位演算法流程圖。

1.5 章節說明

本論文一共分為六章，第一章為介紹研究動機目的及說明機器人定位系統之架構。第二章說明全向式影像中對環境特徵點的偵測辨識方法。第三章推導 EKF based SLAM 演算法。第四章則是對定位參考點建立及轉換的策略設計作說明。第五章為實驗結果，驗證各演算法的正確性。第六章為結論及未來展望。

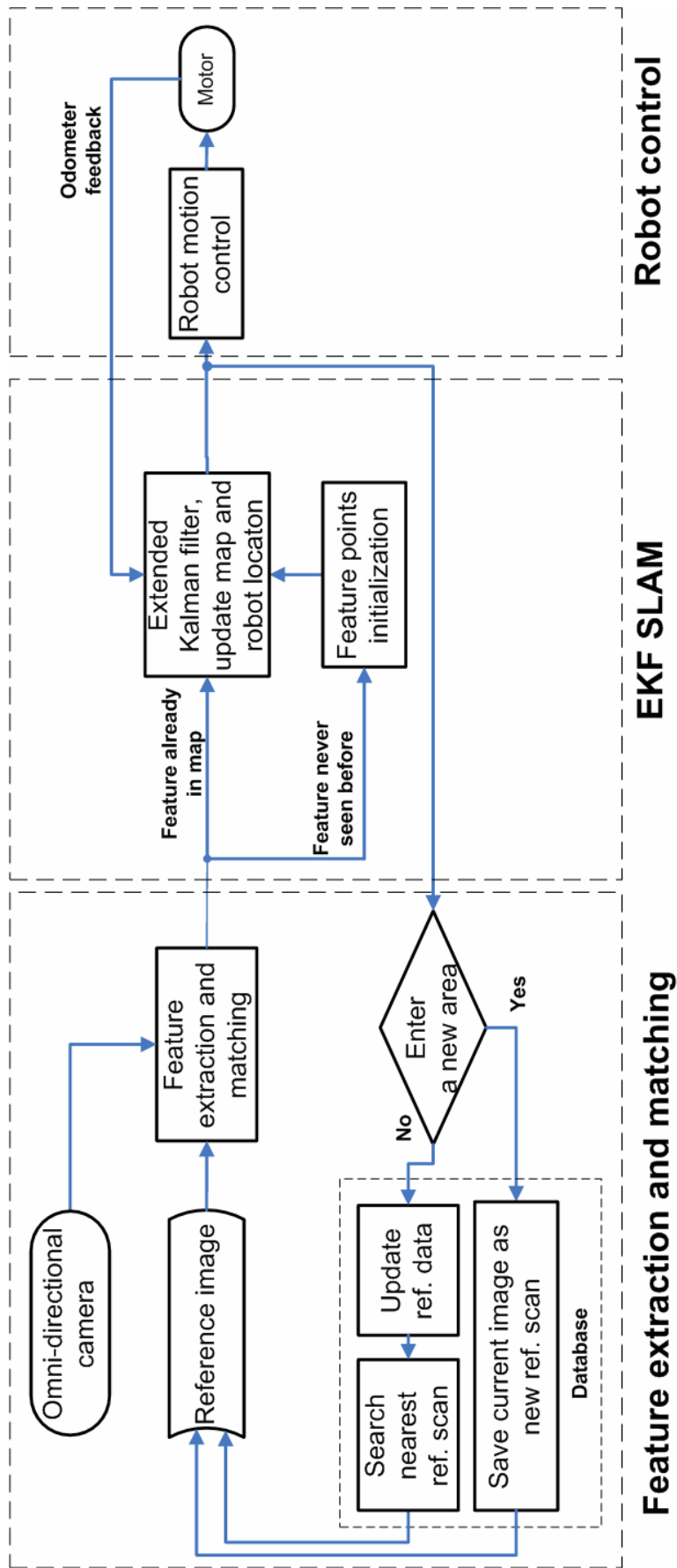


圖 1-1 機器人定位系統演算法流程圖

第二章 環境特徵偵測

為了機器人定位，需要從環境中取得 Landmark 作為定位演算法的依據，而在影像平面中即以影像特徵點作為 Landmark，在本章中說明了在影像中擷取特徵點並加以辨識的方法，以達成環境特徵點之偵測。

2.1 全向式攝影機概述

一般的單眼攝影機所能拍攝到的影像僅限於攝影機之視角範圍，用於機器人導航時，會因其較狹窄的視角而產生限制，如當機器人移動時，對於同一個 Landmark 無法追蹤較長的時間，或是當有一些障礙物遮蔽視線時，無法取得足夠的環境資訊。有別於一般單眼攝影機的有限視角，全向式影像系統能夠即時取得週圍 360° 的影像資訊，用於機器人導航時將會更有幫助。

全方位視覺系統有很多類型，像是僅使用一個攝影機與特殊透鏡的魚眼鏡頭攝影機[17]，或是使用多台攝影機的立體視覺[18]，還有一種常被使用的則是使用一台攝影機及反射鏡所組合而成的全向式攝影機[19]。僅使用反射鏡與攝影機所組成的全向式攝影機系統，因為其機構簡單，所以常用在機器人系統上；此種全向式攝影有兩種類型，如圖 2-1、圖 2-2：圖 2-1 此種攝影機的可視角度大致上為水平零度以下，除了攝影機本身以外，整個地板平面都可以是可見範圍，Winters *et al.*[9]就是利用此種影像來判斷出牆角，使機器人能在走廊環境中行走。圖 2-2 此種類型的可視角度則大致為水平零度以上，除了自己本身以外，整個天花板都是在可見範圍之內，[10]就是利用天花板上長方型的燈管來定位，判別機器人在走道上的何處。

本論文實驗中所使用全向式攝影機即是圖 2-1 的這種類型，圖 2-3 為攝影機照片，圖 2-4 為此類型攝影機所擷取到的影像。

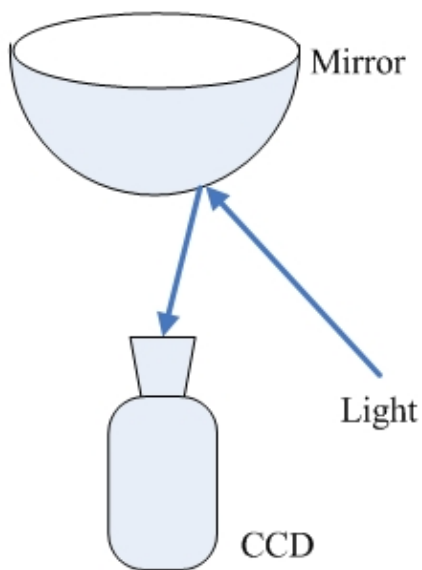


圖 2-1 全向式攝影機示意圖
(擷取水平面以下影像)

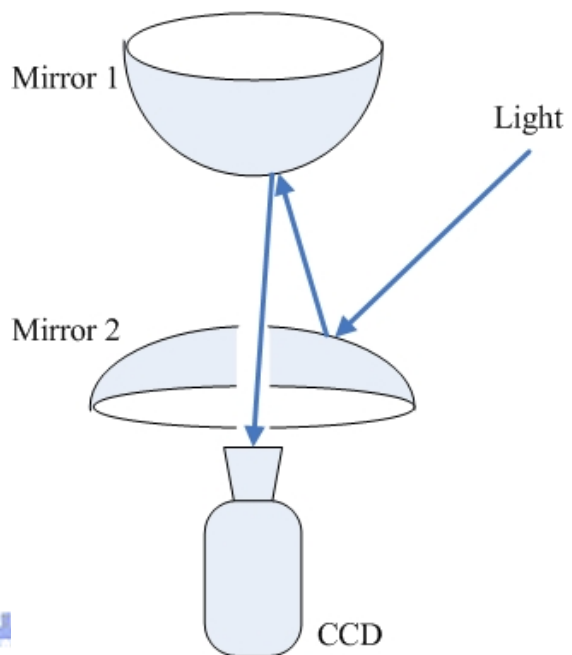


圖 2-2 全向式攝影機示意圖
(擷取水平面以上影像)



圖 2-3 Vstone VS-C450N-TR 全向式攝影機

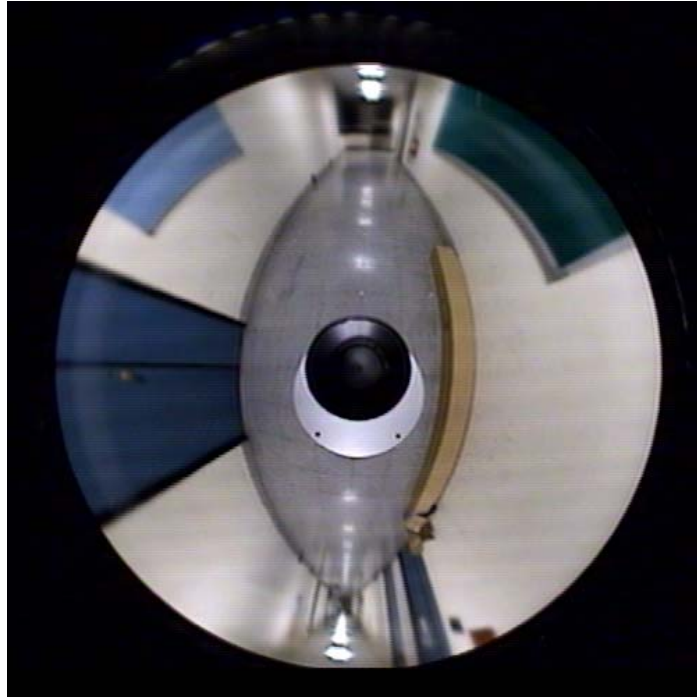


圖 2-4 全向式攝影機所擷取之影像

2.2 環境特徵點辨識

在使用影像定位的方法中，主要是利用影像來偵測環境中的 Landmark，利用環境中固定位置的 Landmark 與自己本身的相對位置來估測機器人在環境中的位置。因此，首要的步驟就是要能從影像中偵測到周遭環境中的 Landmark。

因為是使用 Landmark 來定位，因此 Landmark 的判別要能夠穩定且正確，最簡單的方法就是在環境中放置人為的 Landmark(Artificial landmark)[20]，將特製的 Landmark 放置於環境中，因為 Landmark 的圖樣已知，故只要在影像中搜尋特定的圖樣即可輕易的得知 Landmark 的位置，但使用人為的 Landmark 需要事前就放置好 Landmark，當機器人到達一個新的環境時還要重新的佈置 Landmark，讓這個方法不具有一般性，現在的研究也較少使用人為的 Landmark 來做定位。

另一種方法則是不使用人為特製的 Landmark，先觀察原始環境中較常出現的圖樣，將較常出現的圖樣預先存在資料庫中，Landmark 的判別就是依照這幾種圖樣來偵測：如[10]就是偵測天花板中的燈管來判別機器人的位置，[1]則是利

用走廊環境中最常出現的”門”來做為 Landmark，在資料庫中預先設定好燈管、門...等特別物體的圖樣，接著在影像偵測相同圖樣的物體，即可估測出特定物體的位置。

使用非人為特製，而是原本就存在於環境中的 Natural landmark 來定位是最一般化的做法，不需依賴人為的 Landmark，不用事先儲存 Landmark 的圖樣，直接從環境擷取出 Landmark 來使用，讓機器人不管移動到舊或是新的環境中時，都能直接開始工作，增加使用上的效率及便利性。

偵測環境中的 Natural landmark 有許多方法，其中之一就是以環境中的特徵點(Features)做為 Landmark，像是物體的邊緣、角落、或是花樣等色彩分明的位置點都可以當作是特徵點，像是角點偵測或是邊緣偵測等方法，都可找出環境中的特徵點，接著再對這些點做辨識與追蹤，使機器人能在影像平面中持續穩定的追蹤到環境的特徵點，以做為機器人定位的重要依據。

在本論文中所使用的特徵點擷取及辨識方法為 SIFT 演算法，此方法之特徵點辨識擁有”對抗影像尺度大小改變的影響”、”特徵點描述具有對抗旋轉的特性”、”對於視角的改變有一定容忍性”...等特性，因此對於使用在移動式機器人的影像特徵點辨識上有許多的優勢；配合用於機器人導航的目的及全向式攝影機的特性，在本研究中修改了原始的 SIFT 演算法來使用，使其更有效率。

圖 2-5 為本論文中所設計使用的基於全向式攝影機之特徵點擷取與比對演算法流程圖，主要目的為比較相鄰連續兩張的影像，在這兩張影像中找出相對應、相同的特徵點，以此計算出特徵點相對於機器人本身的位置變化，並且依據特徵點來定位出機器人本身的位置。主要方法為先將兩張影像圖片做角點偵測，找出影像中的特徵點；接著對特徵點及其周圍的區域影像旋轉至同一個方向後，記錄此特徵點的描述向量(Descriptor vector)；利用最小距離法(Nearest neighbor method)比對出兩張影像中最相似的兩個點，最後使用一 Outlier detector 來消除明顯比對錯誤的點，增加比對的正確性。用此方法可得到相鄰兩張影像中的特徵點對應關係，而此演算法的詳細內容也將在接下來的幾個小節中說明。

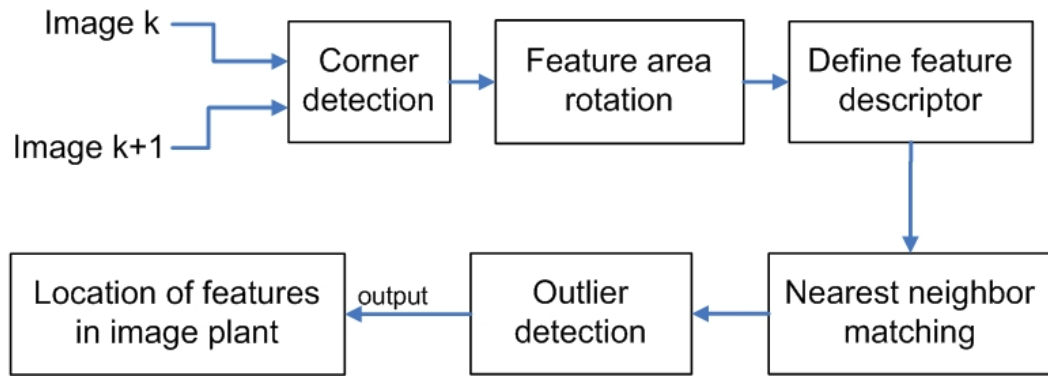


圖 2-5 基於全向式攝影機之特徵點擷取與比對演算法流程圖

2.2.1 SIFT 特徵點擷取

在 Lowe 的 SIFT 演算法中是使用高斯差值函數(Difference of gaussian)來擷取圖形的特徵點[8]，選取出影像的邊緣為候選的特徵點，接著與候選特徵點周遭的區域作比較，判斷是否為區域中的極值，以此挑選出不易受雜訊干擾且穩定的特徵點；為了對抗影像尺度大小改變的影響，則對不同尺度空間大小的影像也做特徵點的擷取，讓在欲辨識的影像有大小的變化時，亦有不同尺度的特徵點可供比對，增加對抗尺度大小改變的特性。

圖 2-6 為以 Difference of gaussian 為基礎之特徵點擷取方法之示意圖，首先求出不同程度的尺寸空間影像，作法如(1)式，其中 $G(x,y,\sigma)$ 為高斯函數， $I(x,y)$ 為輸入之影像：

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (1)$$

$$\text{where } G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2)$$

$L(x,y,\sigma)$ 除了是由如圖 2-6 中不同的解析度影像中求得外，更依參數 σ 求得不同層次的尺度空間影像，做法如(3)式，其中 k 為 σ 之比例係數，每層影像的 σ 差距為 k 倍：

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \quad (3)$$

將不同層的 Gaussian 影像相減後即為 Difference of gaussian， $D(x,y,\sigma)$ ，在

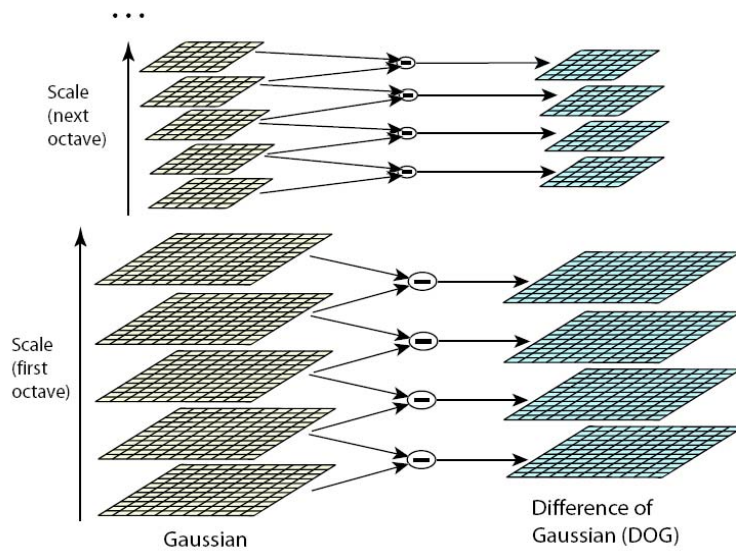


圖 2-6 不同尺度空間之 DoG(Difference of gaussian)示意圖[8]

$D(x, y, \sigma)$ 中判別出各個 Pixel 點是否為區域極值，以此方法擷取出影像中之特徵點。

2.2.2 基於全向式攝影機之特徵點擷取與比對演算法

當使用的 Sensor 為全向式的攝影機，具有 360 度的視角範圍，即使某一方向的影像因為機器人的移動而改變了大小，讓特徵點的 Scale 改變了，但其他方向的影像改變卻不一定會很大，因此仍有許多 Scale 變化不大的特徵點留在影像當中。例如圖 2-7 為全向式攝影機所拍攝的兩張影像，此影像為在走廊環境下所拍攝，左圖為機器人在走廊中心所拍攝之影像，右圖為向影像上方的門(牆面方向)移動 0.7m 後所拍攝的影像。可發現到，上方的門(圈出部份)因為機器人攝影機的移動有著較明顯的大小變化，其周圍區域不僅有大小的變化，形狀更是有扭曲的現象發生，特徵點因此而不易辨識。下方的佈告欄及門(圈出部份)，因為位於非機器人的移動方向上且距離有一段距離，因此在影像中的大小較無明顯的變化。因此，使用全向式攝影機時，即使因為機器人的移動使一部份的特徵點變的不易辨識，但因為攝影機的 360 度視角，仍有其他方向的特徵點可以清楚的辨識使用，讓定位演算法可以繼續運行。

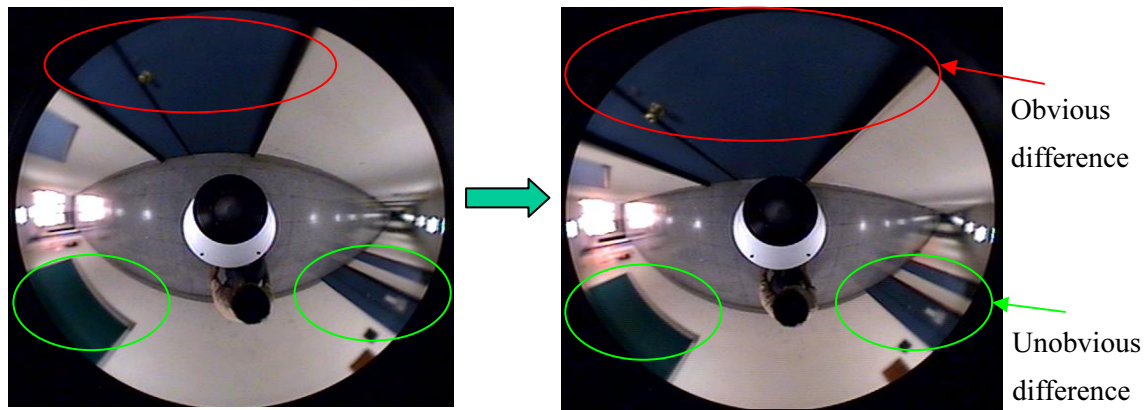


圖 2-7 相鄰影像中之物體大小變化比較

另一方面，機器人工作環境是在室內走廊上移動的情境時，所擷取的特徵點幾乎都是在兩側的牆面上，機器人在此環境中移動時，大部份都是沿著走道行走，也就是平行牆面的方向行走，對牆面平行方向的移動來說，所觀測到牆面上的特徵點大小不會變化的很嚴重。例如圖 2-8 為兩張全向式攝影機所拍攝到的影像，圖右為圖左的拍攝位置向右(沿走廊方向)移動一公尺後所拍攝，可發現到，影像上方的門(圈出部份)，並未因為移動一公尺後而有很大的形變量，也就是說，在圖左所能偵測到的特徵點，在圖右中亦有很大的機會能夠辨識的出來，特徵點的辨識能力在移動一小段距離後仍有一定的成功率。

為了減少特徵點擷取之運算時間，本論文之特徵擷取不使用對不同尺度大小的影像做 DoG 來擷取特徵點，基於上述全向式攝影機及實驗環境的特性，使用較為簡易的角點偵測[11]來擷取環境中的特徵點。本論文所使用的角點偵測方法是採用文獻[11]的方法，其原理與 OpenCV 函式庫中的 cvGoodFeaturesToTrack 函式相同：計算各個 Pixel 點的最小特徵值，之後再判斷是否為區域極值，最後求得特徵角點的位置。

2.2.2.1 特徵點擷取法則

為了取得環境中的 Landmark 作為定位的依據，故從影像畫面中擷取影像特徵點作為 Landmark，本論文使用文獻[11]的方法來擷取影像中的特徵角點，下述為此角點偵測法之演算法步驟：

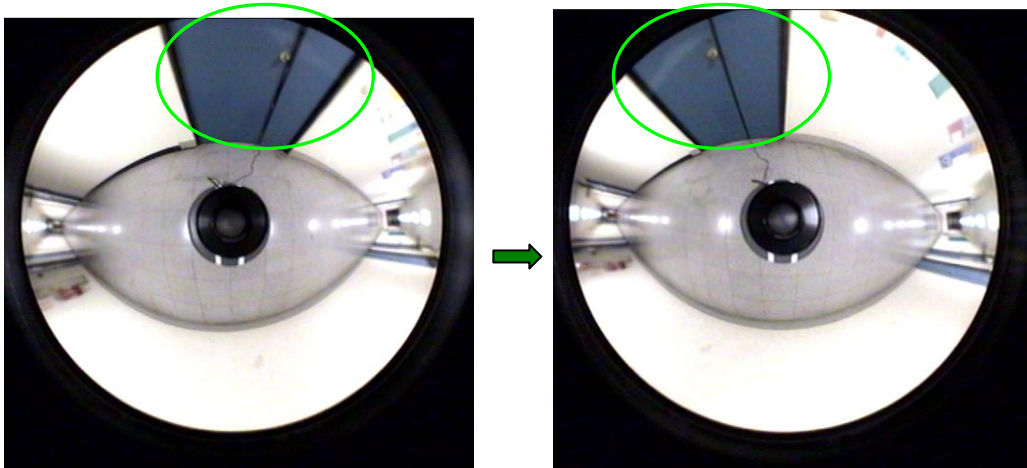


圖 2-8 機器人平移後影像中的物體大小變化比較

Step 1 :

首先使用 Sobel 運算子[27]求得影像中各 Pixel 點的 x、y 方向的偏微值 D_x 、 D_y ，如圖 2-9 所示。

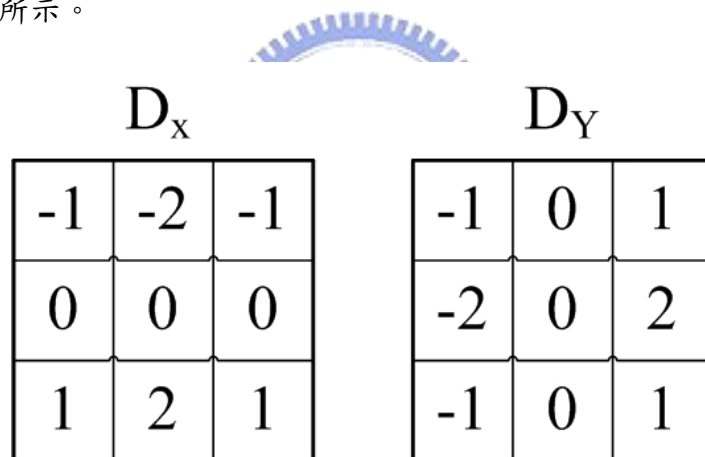


圖 2-9 Sobel 運算子

Step 2 :

對於影像中的每個 Pixel 點均建立一矩陣 C ，矩陣 C 如下(4)式，是以各 Pixel 點為中心的 3x3 Pixel 遮罩區域 Q 的 D_x 、 D_y 的運算累積，並且計算出此矩陣 C 的特徵值(Eigenvalue)。

$$C = \begin{bmatrix} \sum D_x^2 & \sum D_x D_y \\ \sum D_x D_y & \sum D_y^2 \end{bmatrix} \quad (4)$$

特徵值的算法如下(5)式，取兩個特徵值中較小的值代表此 Pixel 的特徵值

$$\lambda = \frac{\Sigma D_x^2 + \Sigma D_y^2 \pm \sqrt{(\Sigma D_x^2 + \Sigma D_y^2)^2 - 4 \cdot (\Sigma D_x^2 \cdot \Sigma D_y^2 - (\Sigma D_x D_y)^2)}}{2} \quad (5)$$

Step 3 :

對各點的特徵值做 3x3 Pixel 遮罩區域的極大值判斷，過濾出為區域極大特徵值的點；最後設定一特徵值之 Threshold，各區域極大值之點若大於此 Threshold 即為所求得的特徵角點。

為了之後特徵點辨識的需求，特徵角點要能愈特殊愈好，故距離太相近的兩個特徵角點對以後的辨識不利，因此最後再將間隔距離太近的特徵角點刪除，留下的角點均具有一定的距離。圖 2-10 為一張全向式影像圖片所擷取出的特徵角點結果。

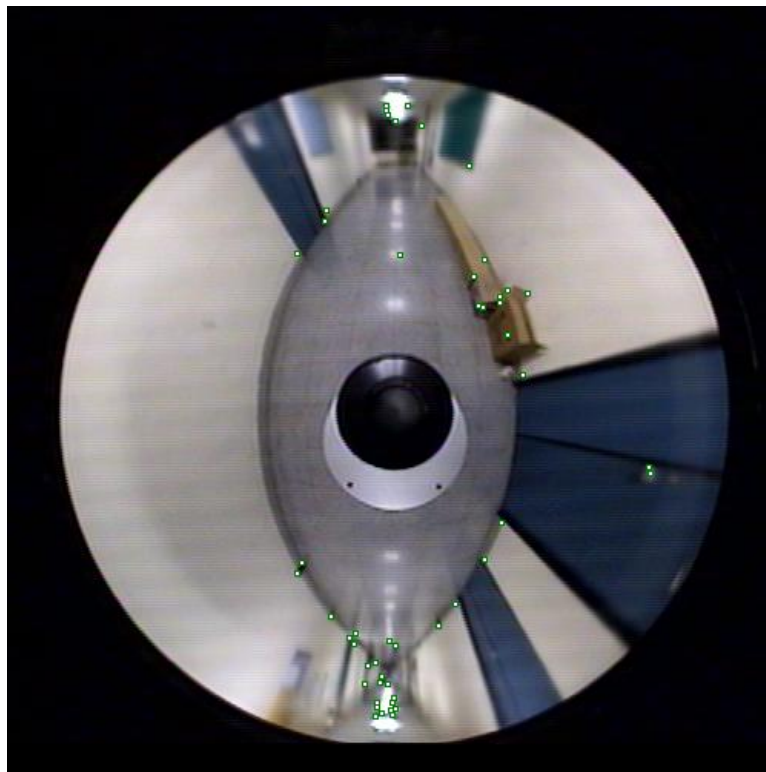


圖 2-10 角點偵測結果

2.2.2.2 特徵點描述

在擷取出影像中的特徵點之後，接著就要對每個特徵點建立其描述向量(Key point descriptor vector)，描述向量可說是特徵點的表示法，是由特徵點本身及其周圍的像素所組成。但是單只有特徵點本身的資訊是無法完全代表特徵點的所

在，基於環境是較不易變動的情況下，以周圍環境像素資訊所組成的特徵描述向量來代表此特徵點是較為完善的。基於特徵點的描述向量，之後在不同影像中要辨識出同一個特徵點時，此描述向量就是判別的主要資料。

因為描述向量關係著之後特徵點辨識的成功與正確性，故特徵點描述的方法是非常重要的。在 SIFT 演算法中，為了使特徵點的描述向量具有抗旋轉的特性，所以在特徵點周圍像素的描述上，其方向的描述均以特徵點本身的梯度方向為基礎，所以若影像畫面有旋轉的現象時，特徵點描述的方向亦會隨著特徵點旋轉的角度而修正，使方向具有均一性，即使影像旋轉了，描述向量依舊是不會改變，使特徵點的描述具有抗旋轉的特性。

在全向式影像中，若是機器人旋轉時，影像中的物體不會像一般攝影畫面一樣水平的移動，而是以鏡心為圓心，繞著鏡心而旋轉，為了使旋轉後影像中的特徵點仍然是正確的，在本研究中參考了[3]所使用的特徵點描述抗旋轉方法，利用全向式影像的特性，在影像平面中直接移動特徵點及周圍像素後，再開始特徵點的描述工作，此方法簡易且直接，利用全向式攝影機的特性，完成了特徵點描述抗旋轉的功能。

如圖 2-11，P 代表的是其中一個特徵點及其周圍的描述區域，此特徵點距離鏡心的長度為 r ，統一將所有的特徵點周圍區域影像旋轉至影像中的正上方，將 P 區域中的像素以鏡心為圓心， r 為半徑，以 Bilinear interpolation 的方式，將所有的像素旋轉至 P' 處，P' 即代表特徵點描述區域旋轉至影像正上方後的所在，其轉換方式如(6)式。其中 (x,y) 代表影像平面上 P 區域中的 Pixel 點位標， (x',y') 代表 P 轉換至 P' 區域後的 Pixel 點位標， r 為 P 區域中心特徵點距鏡心的距離， θ 為與影像正上方的角度差。

$$\underbrace{\begin{bmatrix} x' \\ y' \end{bmatrix}}_{P'} = \begin{bmatrix} x - r \cos \theta \\ y + r \sin \theta \end{bmatrix} \quad (6)$$

將所有的特徵點均旋轉至同一方向後，再作特徵點的描述，使描述向量不受畫面旋轉的影響。

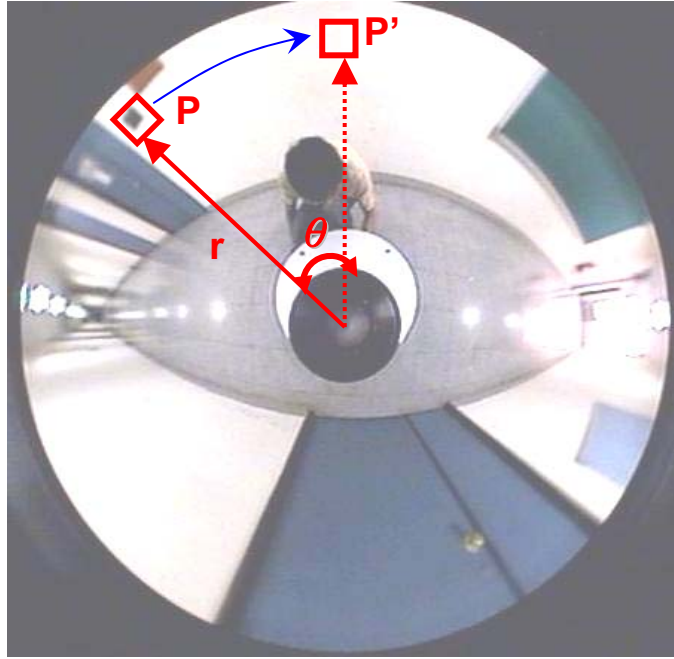


圖 2-11 特徵區域旋轉示意圖

經過上述的旋轉過程後，每個特徵點及其周圍描述區域影像都已旋轉至同一方向，接下就是開始對描述區域內的每個像素計算其所代表的特徵向量值：以各個像素點的梯度大小 $m(x,y)$ 及梯度方向 $\theta(x,y)$ ，其計算方式分別如(7)式、(8)式， (x,y) 代表該像素點在影像平面上的座標， $L(x,y)$ 則是該點像素的灰階值。

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (7)$$

$$\theta(x,y) = \tan^{-1} \left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \right) \quad (8)$$

圖 2-12 為特徵點描述向量建立的示意圖，圖左中每個方形都代表描述區域中的一個像素點，方形中的向量方向為該點的梯度方向，方形中向量的長度為該點的梯度大小乘以其高斯權重(Gaussian weight)，如(9)式，其中 $m(x,y)$ 為該點的梯度大小、 (x_c, y_c) 為描述區域的中心座標，因為愈靠近特徵點本身的位置(描述區域中心)，代表對於此特徵點的描述重要性愈高，故以描述區域中心開始對各向量長度作高斯的權重分佈，以突顯各點所代表的重要性。

$$\text{向量長度} = m(x,y) \cdot \frac{1}{2\pi\sigma^2} e^{-((x-x_c)^2 + (y-y_c)^2) / 2\sigma^2} \quad (9)$$

在本論文中，對於特徵點的描述區域大小均設定為 16 Pixel x 16 Pixel，並將此

區域分割成 4x4 的子區域，每個子區域中的 16 個點均以固定的八個方向累積其向量值，以此代表此子區域的八個特徵向量值，如圖 2-12 右；全部的描述區域內共有 4x4 個子區域，每個子區域有代表其特徵的 8 個向量值，故此描述區域的描述向量共有 4x4x8=128 維的向量值，此 128 維的特徵向量即是此特徵點的描述向量，也是特徵點比對的主要資訊。

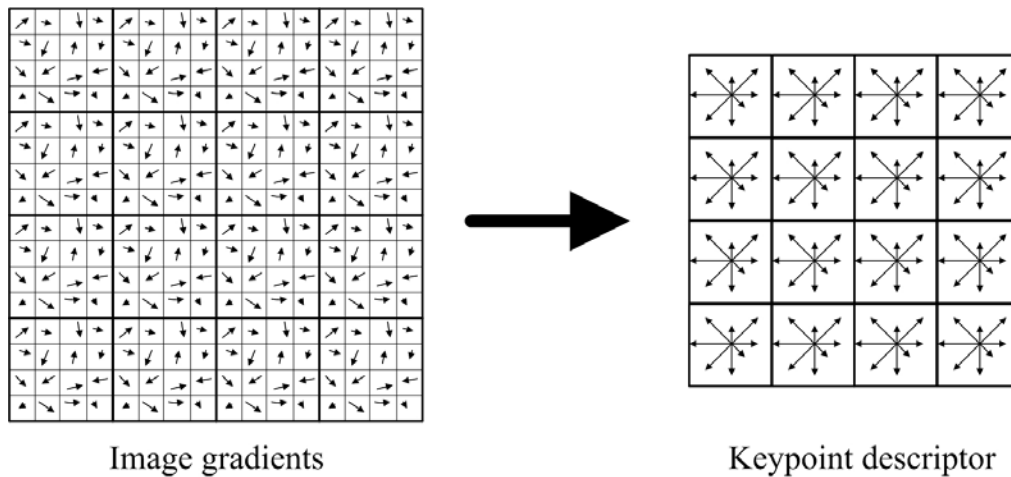


圖 2-12 特徵點描述向量之建立

2.2.2.3 特徵點比對

計算出特徵點的特徵描述向量後，影像中的每個點均有其代表的向量，在兩張影像中，就能以此特徵描述向量來比對出兩張影像中相同特徵點的相互對應關係，得知不同的時間機器人對同一個特徵點的觀測值，反推出機器人的相對位置，達到機器人定位的功能。

本論文採用最小距離法(Nearest neighbor method)作為特徵點的比對方法：因為每個特徵點均已用 128 維的特徵向量表示，因此與此向量最相近的另一個特徵點向量極有可能就是同一個點，(10)式即為 Nearest neighbor method，其中 L 代表兩個特徵向量的距離， $Des_1(i)$ 代第一張影像之特徵向量的第 i 個元素， $Des_2(i)$ 代第二張影像之特徵向量的第 i 個元素，利用此方法找出距離最相近的兩個點，配對出兩張影像中特徵點的對應關係。

$$L = \left(\sum_{i=1}^N (Des_1(i) - Des_2(i))^2 \right)^{\frac{1}{2}} \quad (10)$$

2.2.2.4 Outlier 之判定

因為最小距離法是依特徵描述向量來配對的，若是影像中有數個特徵點之描述向量內容很相近，使用此配對法時就有可能配對到另外的點，造成配對錯誤的情況產生。因為特徵描述向量是以該點周圍的區域像素所產生的，若有數個特徵點其周圍較為單純或是圖樣相近，其描述就會較為相似，造成比對錯誤的情況產生。對錯的比對結果，不僅會對特徵點的位置估測產生偏差，更會造成定位系統的不穩定，本研究在 SIFT 演算法比對結束後，再增加一比對錯誤點判斷式(Outlier detector)，用以檢測各特徵點比對的正確性，刪除可能錯誤的點，增加比對的正確率。

採用 Kim *et al.*[7]所使用的 Outlier 判別方法，利用各組已 Match 的特徵點位置關係來判別 Outlier，剔除在已 Match 的特徵點中明顯錯誤的特徵點。

Step 1 :

如圖 2-13，左右各表示在兩張影像中幾個特徵點配對的結果與其相對位置(i 與 i', j 與 j' 配對)，首先求出 i 點與 j 點在 Current frame 及 Previous frame 中的最小相對距離比 D_t^{ij} ，算法如(11)式，其中 R_t^{ij} 代表 Current frame 中 ij 的距離， $D_{t-1}^{i'j'}$ 代表 Previous frame 中 i'j' 的距離。

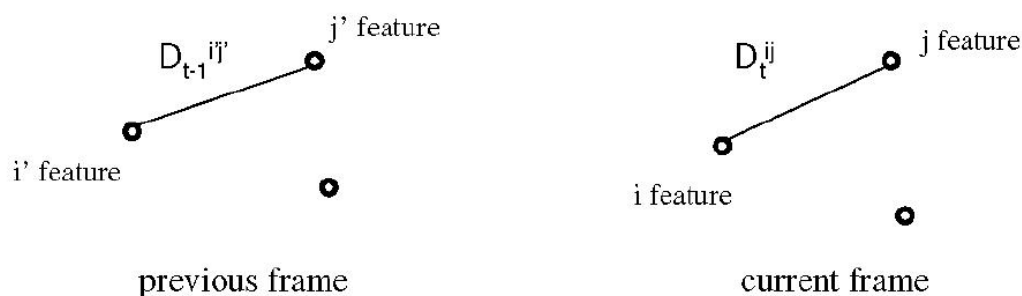


圖 2-13 Outlier detector 示意圖[7]

若 i 與 j 點的配對是正確的，ij 的距離在兩張相鄰的影像中差距是不會有很大

大的變化的，因此 R_{ij} 會接近 1，反之若比對錯誤時， R_{ij} 則會較小於 1。

Step 2 :

計算出 i 點與其他已配對點的距離比例並累計相乘，如(12)式，若 i 點為 Outlier 點， R_i^i 勢必會愈乘愈小，

$$R_i^i = \prod_j R_{ij} \quad (12)$$

Step 3 :

最後將 R_i^i 代入判斷式(13)式中，若其值小於臨界值，則代表 i 點為 Outlier，式中 N 為配對點之總數， T 為 Threshold。

$$\log_{10}(R_i^i) / N < \log_{10}(T) \quad (13)$$

將每個配對完成個特徵點均代入 Outlier detector 中，找出其中較為可能錯誤的點，並刪除之，增加特徵點比對的正確率。

本章中說明了本論文對環境特徵點偵測與辨識的方法，利用角點偵測法擷取出影像中的特徵點，並將特徵點描述區域旋轉至影像的正上方，讓描述區域具有一致的方向性，統計出描述區域的描述向量後，利用 nearest neighbor 法作比對，以辨識出兩張影像中相對應的特徵點，最後為了增加特徵點比對的正確率，使用一 outlier 偵測法，刪除比對錯誤的點，即完成了特徵點的偵測與辨識演算法。

第三章 機器人定位

本章主要說明機器人定位的演算法，以第二章演算法所偵測出之特徵點位置為輸入，使用 Extended Kalman filter based SLAM 演算法同步達成機器人定位及環境特徵圖地圖的建立。

3.1 vSLAM 簡介

機器人定位是處理機器人本體相對於環境地圖位置的過程，是一種決定機器人位在地圖中的什麼地方的方法。在地圖已知的條件下，藉由觀測機器人周遭的環境再與地圖做比較，即可得知機器人在地圖上的哪個位置，這是機器人定位中基本的情境。但是機器人若是移動至不在地圖中的位置、或是一開就沒有地圖的話，機器人就無法定位了，而 SLAM 演算法即是解決了在未知地圖、或機器人的位置也未知的情況下的定位問題。

在 SLAM 演算法中，當機器人依所觀測的的環境資訊而定位出本身相對位置的同時，也能夠逐漸建立起周遭的環境地圖。圖 3-1 中的 x_t 代表機器人在 t 時刻的位置資訊，運動命令 u_{t+1} 控制機器人往目標點前進，在 $t+1$ 時刻位置，機器人能得到對環境的觀測資訊 z_{t+1} ，此觀測值不僅能幫助機器人定位出本體的相對位置 x_{t+1} ，亦顯示出周遭的環境地圖 m ；同時定位出機器人的位置 x_{t+1} 及建立起地圖 m ，即是 SLAM 最重要的精髓。

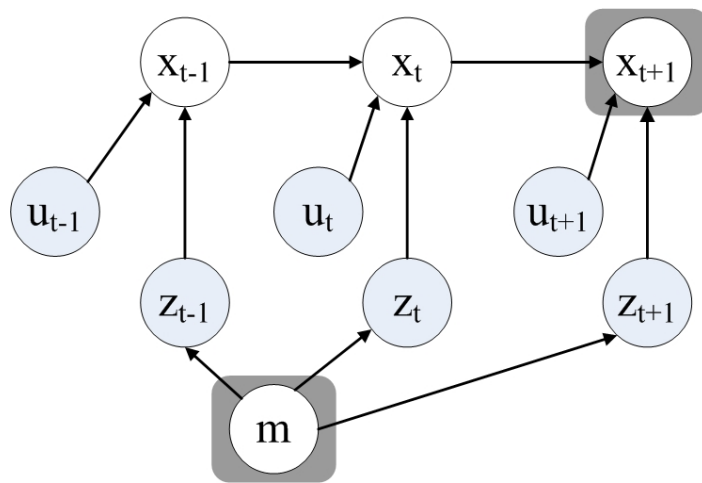


圖 3-1 Online SLAM 之圖示模型[14]

示意圖 3-2 為 Durrant-Whyte[13]所說明的 SLAM 中所要解決的問題：因為無法得知機器人及環境 Landmark 確實的位置，所以要藉由 Sensor 來測量環境的資訊，以此估測出機器人的位置及環境地圖。例如當機器人位於 X_{k-1} 時，觀測到 m_i 等兩個 Landmark，將兩個沒見過的 Landmark 建立在地圖上後，亦估測出機器人本身的相對位置。因為機器人移動的距離及對 Landmark 的觀測值不可能是完全準確的，因此 Landmark 及機器人位置均有一不確定性(Uncertainty) 存在；在 X_k 時，同時觀測的到 m_i 、 m_j ...等三個 Landmark，除了將之前沒見過的 m_j 等兩個 Landmark 加入地圖後，因為 m_i 是第二次觀測到了，不僅更新了 m_i 的位置，更能縮小其位置所在的不確定性，增加 Landmark 的正確度，同時依據觀測到的 Landmark，也能再次估測出機器人的位置；以此循環，機器人便能完成同時定位且建立環境地圖的功能。

在環境地圖的表示法中，最常見的 Grid-based、Feature-based...等方法[21]，

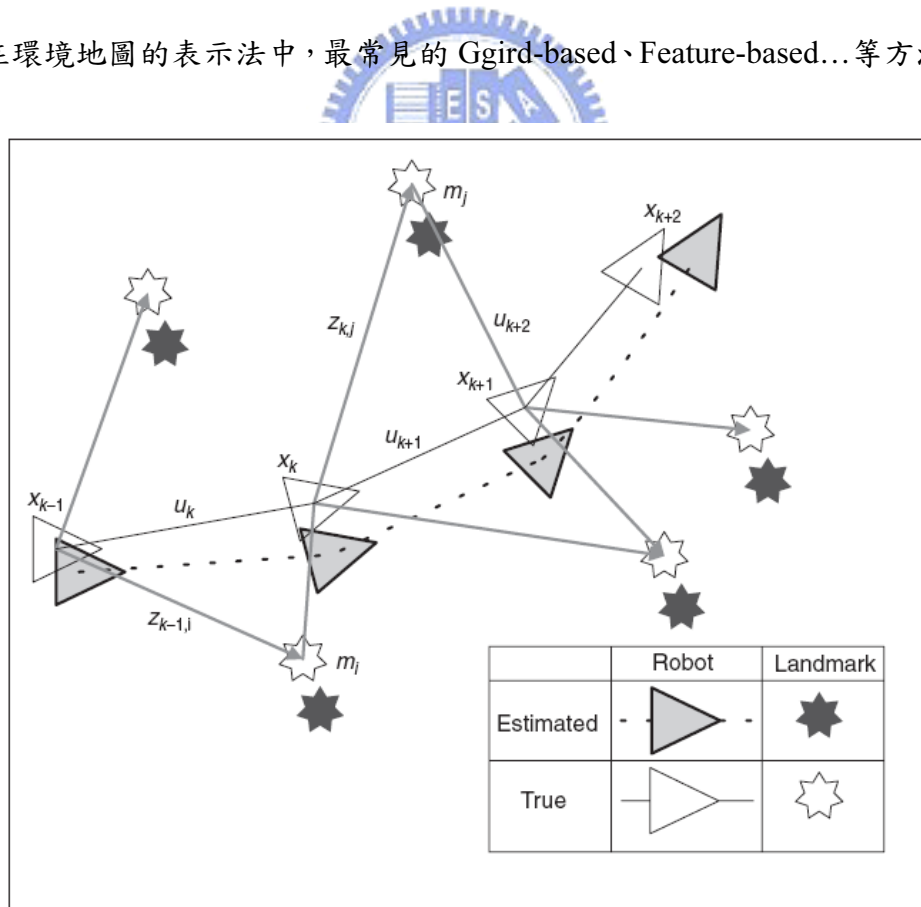


圖 3-2 SLAM 演算法概念示意圖[13]

Feature-based 表示法不像 Grid-based 一樣將所有地圖中的位置資料均顯示出來，而是以特徵點或是直線來代表地圖，這種特徵也被稱為 Landmark。不同於基於距離特徵為 Landmark 的 SLAM，vSLAM 則是基於影像平面中的特徵 Landmark，在影像中能夠看到更多的環境資訊，雖然可以擷取出更多的特徵，但在特徵點距離估測及影像處理方面則要更加投入。

在 SLAM 中，解決機器人位置與地圖不確定性的機率問題是主要的議題之一，而基於 Feature 之 Extended Kalman filter 則是常用的解決方法[22]，EKF 具有收斂速度快、運算精度高、穩定性佳等優點，其演算法較為直接且容易實現，並適用於非線性系統中，可應用在即時系統，有效克服數據的不確定性，保持系統狀態估測的正確性。在本論文中即是結合了全向式攝影機的觀測資訊與 EKF 來解決 SLAM 的問題。

3.2 特徵點初始化

在 EKF-based SLAM 中，發現新的特徵 Landmark 時，除了要將估測出 Landmark 的位置加入地圖外，也要同時設定此點的 Covariance matrix。特徵點的初始化設定 (Feature initialization) 在 EKF 中是很重要的，好的初始化設定值能讓往後 EKF 運算時，EKF 的 State 估測值能更新到較準確的數值。在 vSLAM 中，由於使用的感測器是攝影機，無法從單一畫面中得知特徵點的景深，只能得到相對於機器人本身的視角，要得到景深的資訊至少需要對同一個特徵點進行兩次的觀測後才能得到，因此在對特徵點的初始化上要特別的注意。

在本論文中的特徵點初始化是使用一非即時性的方法，即是利用數個影像 frame 後再定義出特徵點。因為無法在一個影像 Frame 中得知 Landmark 的深度資訊，因此利用在不同的時刻多次對同一 Landmark 的觀測，與機器人本身的移動，整合數次的觀測值後才在地圖中新增此 Landmark 並定義其初始化設定。

3.2.1 水平位置估測

圖3-3為Landmark深度估測的方法示意圖，使用三角定位法，由機器人在(t)

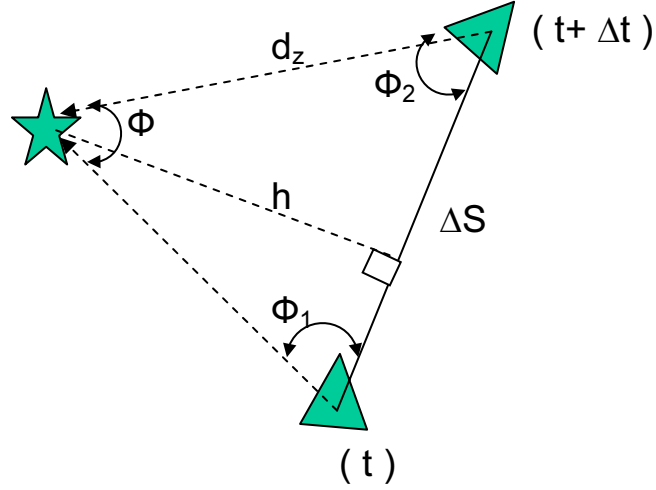


圖 3-3 Landmark 位置估測之座標關係圖

及 $(t + \Delta t)$ 時間對特徵點的視角 Φ_1 、 Φ_2 ，及兩個時間點機器人所在位置的直線距離 Δs ，利用此三角形計算出 Landmark 相對於機器人的距離 d_z ：

$$\begin{aligned}
 h \cot(\phi_1) + h \cot(\phi_2) &= \Delta S \\
 h &= \frac{\Delta S}{\cot(\phi_1) + \cot(\phi_2)} \\
 \Rightarrow d_z &= \frac{\Delta S}{\cot(\phi_1) + \cot(\phi_2)} \cdot \frac{1}{\sin(\phi_2)}
 \end{aligned} \tag{14}$$

因此由(14)式可以推得 Landmark 的座標位置：

$$\begin{cases} f_x = x_{t+\Delta t} + d_z \cos(\phi_2) \\ f_y = y_{t+\Delta t} + d_z \sin(\phi_2) \end{cases} \tag{15}$$

圖 3-4 為特徵點初始化之流程圖，其中 n 為連續偵測的最少次數， ϕ 為估測特徵點之最小視差， S 為估測特徵點位置的最大距離。首先要先判別特徵點是否能被偵測到多次，此條件不只是為了讓特徵點的位置能被估測的出來，更能夠確定此特徵點是否夠穩定而能被影像系統持續的追蹤。接下來依上述的特徵點位置估測方法估測出特徵點的位置，並將數次的觀測值取平均值後成為此特徵點之位置座標。當兩個觀測點的距離很短，使視差過小時(圖 3-3 之視差角 Φ)，這樣估測出來 Landmark 的距離誤差是會很大的，且在三角定位法中，對於愈遠的物體估測，誤差會隨之增大，因此在一個特徵點位置估測完畢後，還要對其視差的下限、距離的上限設定 Threshold，確保特徵點的估測誤差不會過大；若不符合條

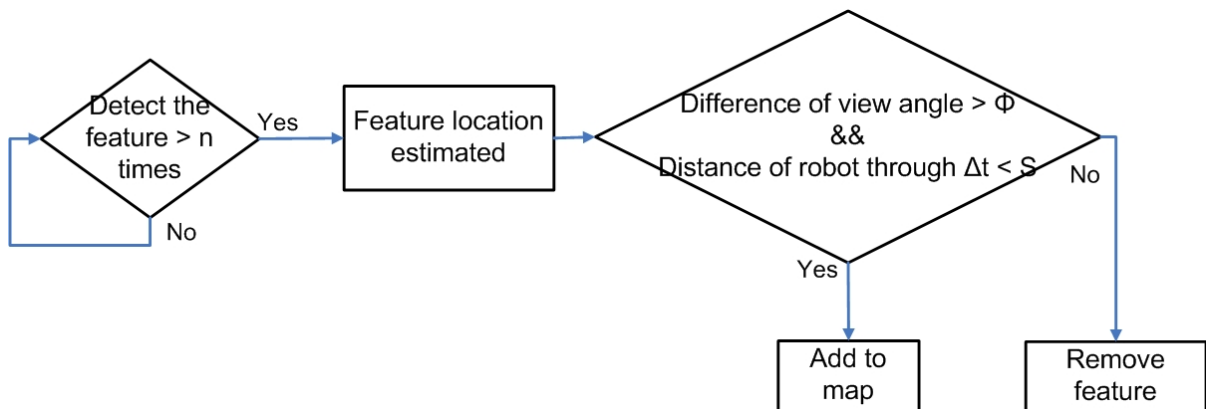


圖 3-4 特徵點初始化流程圖

件限制的特徵點則將其移除不用，符合者則加入地圖，雖然不符合的特徵點無法即時加入地圖中，但仍可在機器人持續移動後，待這些特徵點變得符合條件，即可重新加入地圖中，完成特徵點初始化的步驟。

圖 3-5 為特徵點位置估測的一次實驗結果，機器人從原點出發朝(0,1)前進，過程中所擷取的影像均與原點影像做特徵點的比對與位置估測，菱形點即為所估

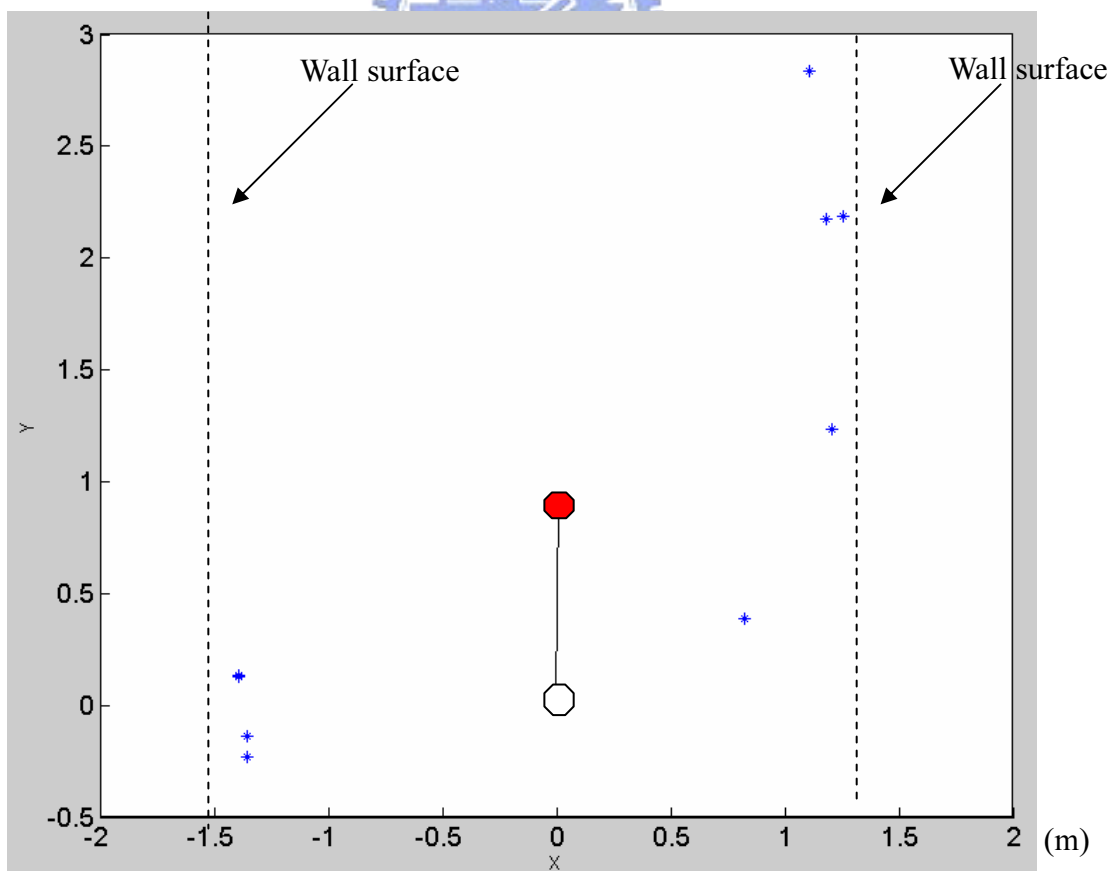


圖3-5 Feature點的平面座標估測圖

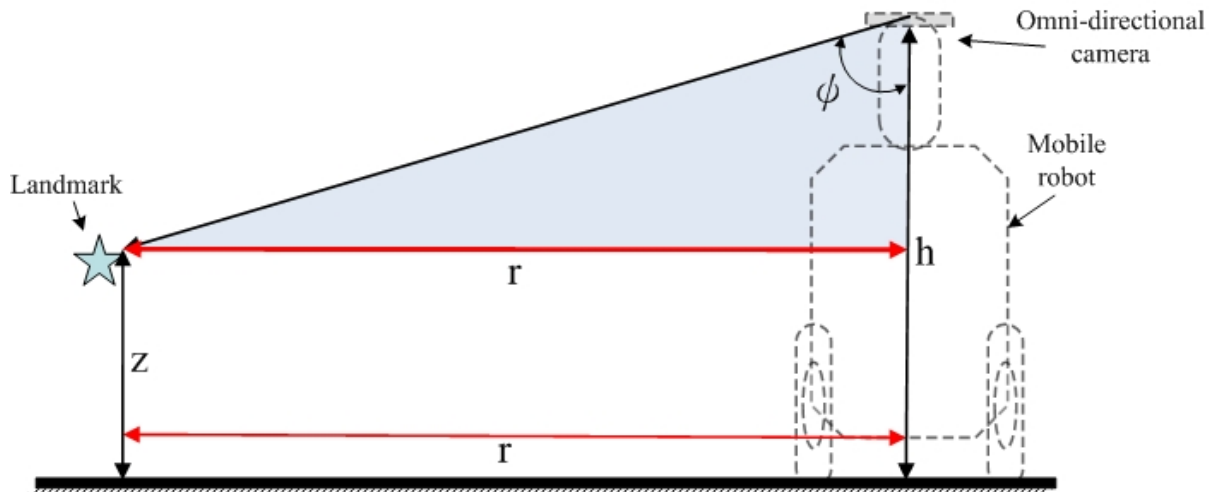


圖 3-6 機器人與 Landmark 位置關係側視圖

測之特徵點所在位置，因為實驗環境是在走廊上，特徵點都是在牆壁上，所以估測出之特徵點都是分佈在虛線(牆面位置)附近。多次實驗後，計算出對特徵點位置估測之平均誤差為 0.415m，此數據也是特徵點 Covariance 值設定之主要依據。

3.2.2 垂直高度估測

計算出特徵點的水平位置之後，即以此為基礎估測出特徵點對於地圖之高度，如圖 3-6 之側視圖，其中攝影機之高度 h 已知，相對距離 r 也已估測出來，若攝影機對特徵點的垂直視角 ϕ 也能得知的话，特徵點高度 z 就能以(16)式計算出來。

$$(h - z) \tan \phi = r \Rightarrow z = \frac{h \tan \phi - r}{\tan \phi} \quad (16)$$

在全向式影像平面上，距離鏡心愈遠的位置，代表相對於鏡心的視角愈大，本論文用校正實驗來近似出影像平面上距鏡心的距離(Pixel)對於垂直視角之對應關係。圖 3-7 即為校正實驗之影像，在牆面上放置 300(15*20)個已知高度的格子點做為樣本，並將攝影機對每個格子點之垂直視角與在影像平面中相對於鏡心之距離記錄下來(如圖 3-8)，在圖 3-8 中將各個記錄值以一條二階曲線近似出來，此二階函數即為影像平面上距鏡心的距離(Pixel)對於垂直視角之轉換公式，如(17)



圖 3-7 視角校正之實驗影像

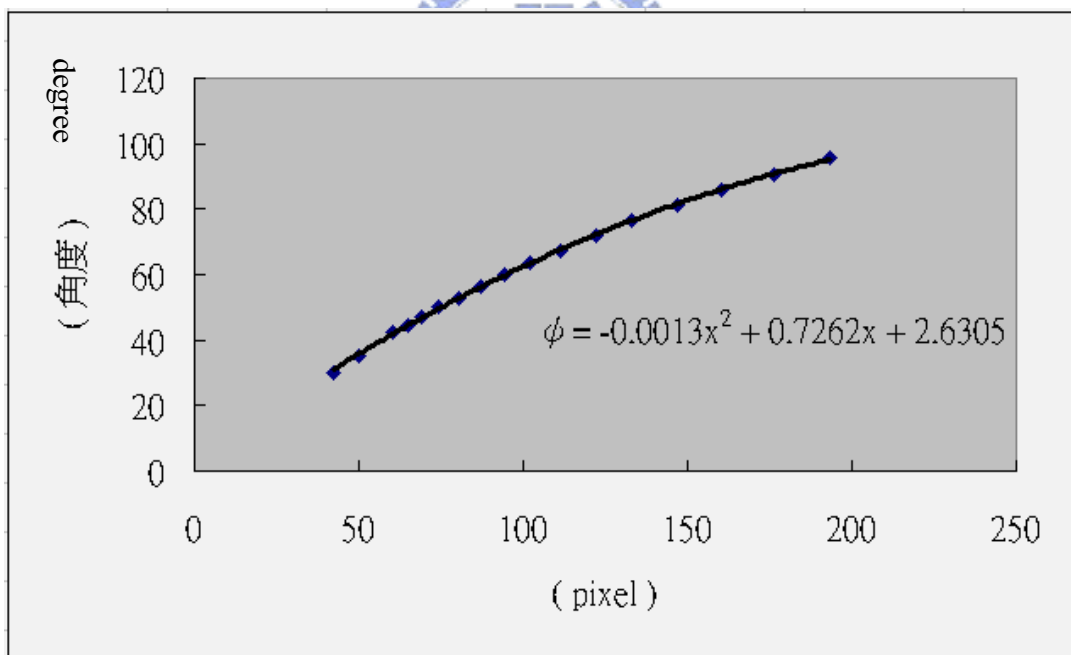


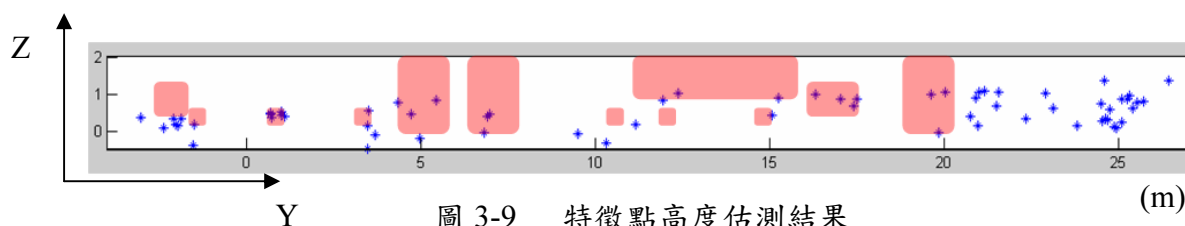
圖 3-8 視角-Pixel 實驗結果及近似關係圖

式，其中 ϕ 為垂直視角($^{\circ}$)， x 為影像平面上距鏡心的距離(Pixel)。

$$\phi = -0.0013x^2 + 0.7262x + 2.6305 \quad (17)$$

利用特徵點在影像平面上的位置距離代入轉換公式，即可得知攝影機對此特徵點的垂直視角，將 Φ 套入(16)式中，就可以求出我們對特徵點估測的另一個資

訊，高度 z 。圖 3-9 為特徵點高度估測之實驗，Y 軸為水平座標，Z 軸為垂直座標，此結果顯示一牆面上之特徵點位置分佈，其中淺色塊代表牆面上較可能存在特徵點的區塊，如門、佈告欄等，圖中特徵點大都依這些區塊所分佈，利影像中的各個特徵點找出實際環境中正確對應的位置，並與估測結果做比較，Z 方向之平均誤差為 0.355m。



3.3 EKF-SLAM 演算法

3.3.1 系統設計

Kalman filter 是一種利用系統模型與系統狀態量測值，對系統的狀態作最佳估測的方法；利用系統的狀態方程式預測當時的狀態值，使用感測器測量系統狀態之量測值，而狀態預測及感測器量測值都有其不確定性存在，Kalman filter 就是依據兩者資訊之可靠程度，以最佳比例融合兩種數值，得到系統之最佳化狀態估測值。

本研究之目的即是要對機器人姿態及環境地圖的狀態作估測，使用全向式攝影機為系統量測裝置。因為本系統之模型為非線性系統，因此使用 Extended Kalman filter 來處理非線性之系統，以達成系統狀態之最佳估測。本系統之系統狀態包含了機器人本身的姿態及環境地圖的位置資訊。其狀態向量(State vector) y_t 如(18)式所示； x_t 代表機器人的姿態，包含有水平位置座標(x,y)、朝向角度 θ ， m 代表環境地圖資訊也就是所有 Landmark 的位置資訊，每一個 Landmark 以 $(m_{n,x}, m_{n,y}, m_{n,z})$ 表示其位置資訊，下標 n 代表第 n 個 Landmark，下標 x,y,z 分別為平面座標(x,y)及高度 z 。

State vector :

$$\mathbf{y}_t = \begin{pmatrix} \mathbf{x}_t \\ \mathbf{m} \end{pmatrix}^T$$

$$= (x \quad y \quad \theta \quad m_{1,x} \quad m_{1,y} \quad m_{1,z} \quad \dots \quad m_{n,x} \quad m_{n,y} \quad m_{n,z}) \quad (18)$$

不同於一般僅估測機器人狀態之定位設計，在 SLAM 的演算法中亦要對所有的 Landmark 作估測：除了用感測器量測環境資訊以定位出機器人的位置外，也要同時建立環境地圖，這是 SLAM 的主要概念。因此在 State vector 中也要增加 Landmark 的資訊，達到同時估測機器人姿態與環境 Landmark 的功能。

圖 3-10 為 EKF-SLAM 之設計概念，主要分為兩個階段：首先在 Prediction step 中利用機器人的 Motion model 預測出系統此時的狀態，接著在 Update step 中整合預測出的系統狀態及感測器量測出的測量值，計算出 Kalman gain 以更新對系統狀態的估測，更新後的狀態資訊再回傳到 Prediction step，作為下一時刻狀態預測依據；整個系統即是在這兩個階段中循環運作，完成機器人定位及地圖建立的功能。

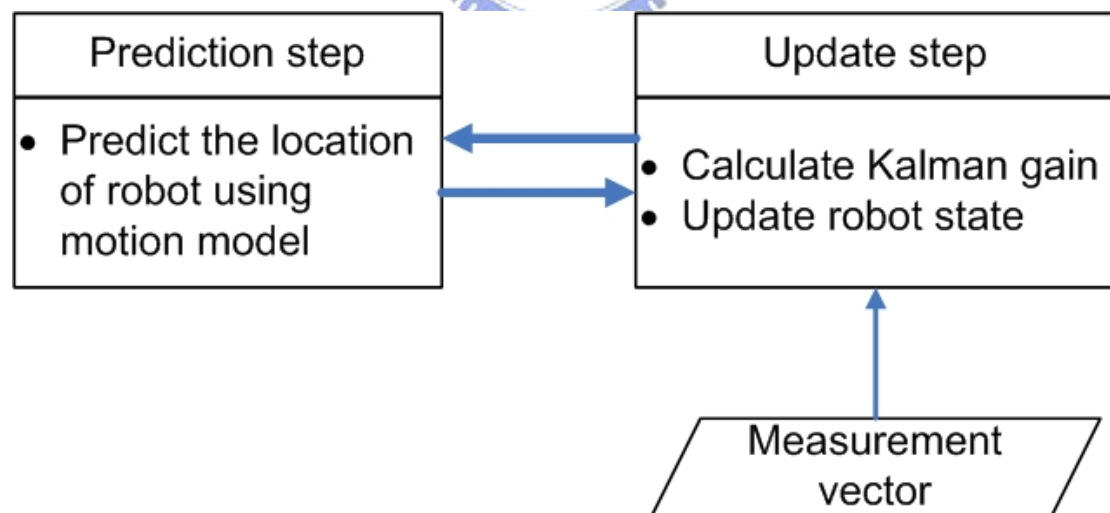


圖 3-10 EKF SLAM 設計概念圖

3.3.2 系統模型

運動模型：

由於本研究是使用速度控制來操作機器人之運動，因此使用基於速度之機器

人移動模型(Velocity motion model)。機器人的速度命令主要有兩個：線速度 (Linear velocity) v 及角速度(Angular velocity) ω ，速度控制模型即是以此兩種速度為輸入之模型。假設機器人在 Δt 時間內是以固定速度沿一圓弧形軌跡移動，此圓弧之半徑為 r 。圖 3-11 為運動模型之示意圖， (x,y) 為機器人所在位置座標、 θ 為其朝向角，箭號代表其移動之圓弧方向， (x_c,y_c) 則為此圓之圓心，已知在半徑 r 之圓弧軌跡上移動之速度關係如(19)式：

$$v = \omega \cdot r \quad (19)$$

因此可得在已知速度的條件下機器人的移動半徑

$$r = \left| \frac{v}{\omega} \right| \quad (20)$$

由(20)式及示意圖 3-11 可推得圓心座標：

$$\begin{cases} x_c = x - \frac{v}{\omega} \sin \theta \\ y_c = y + \frac{v}{\omega} \cos \theta \end{cases} \quad (21)$$

若機器人在(t-1)時刻之姿態 $x_{t-1}=(x, y, \theta)$ ，且保持移動速度經過 Δt 時間後，其理想之姿態 $x_t=(x', y', \theta')$ 可表示成：

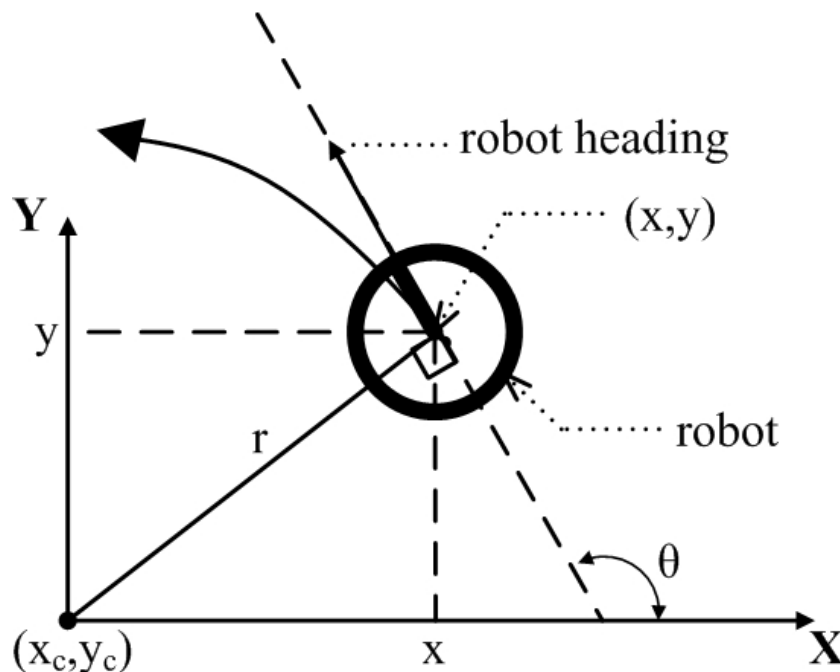


圖 3-11 機器人運動模型表示圖

$$\begin{aligned}
x_t &= \begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x_c + \frac{v}{\omega} \sin(\theta + \omega\Delta t) \\ y_c - \frac{v}{\omega} \cos(\theta + \omega\Delta t) \\ \theta + \omega\Delta t \end{pmatrix} \\
&= \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega} \sin \theta + \frac{v}{\omega} \sin(\theta + \omega\Delta t) \\ \frac{v}{\omega} \cos \theta - \frac{v}{\omega} \cos(\theta + \omega\Delta t) \\ \omega\Delta t \end{pmatrix} \quad (22)
\end{aligned}$$

此即為機器人之速度運動模型。

但實際上機器移動時並不會如運動模型那樣的完美，必定會有誤差存在，故以(23)式表示實際之機器人速度

$$u_t = \begin{pmatrix} \hat{v}_t \\ \hat{\omega}_t \end{pmatrix} = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix} + \begin{pmatrix} \mathcal{E}_{\alpha_1|v_t|+\alpha_2|\omega_t|} \\ \mathcal{E}_{\alpha_3|v_t|+\alpha_4|\omega_t|} \end{pmatrix} \quad (23)$$

左式代表機器人實際的速度，右式則為理想速度值加上雜訊(在 EKF 中，均假設雜訊為高斯雜訊)，雜訊大小依 v 及 ω 改變，參數 α_{1-4} 決定 v 、 w 影響雜訊的比例，移動情況愈不準確的機器人其 α 值愈高。在 EKF 中，對於速度控制雜訊之 Covariance matrix 設定為 M_t ：

$$M_t = \begin{pmatrix} (\alpha_1|v_t| + \alpha_2|\omega_t|)^2 & 0 \\ 0 & (\alpha_3|v_t| + \alpha_4|\omega_t|)^2 \end{pmatrix} \quad (24)$$

在本論文中以實驗測試的結果來設定參數 α_{1-4} ，讓機器人以一設定好之定速度向前移動一段時間，量測移動距離與時間並計算出實際速度，以此方式來算出速度命令與實際速度之誤差，經數次實驗得到速度誤差約在控制命令的 5% 左右，故在往後的實驗中即以此值來設定運動模型的雜訊。

但此模型有一特殊情況，也就是當機器人以直線前進時，也就是 $\omega=0$ ，其移動半徑為無限大，因此在此情況時則不能使用原本的運動模型，改以直線移動的模型代替：

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} v \cos \theta \\ v \sin \theta \\ 0 \end{pmatrix} \quad (25)$$

往後之演算法流程推導表示仍以圓弧運動模型為基礎。

觀測模型；

在本研究中使用的感測器為全向式攝影機，因此量測模型(Measurement model)即是基於全向式攝影機器可量測的內容為依據來設定。量測模型包含了觀測之量測向量(Observed measurement vector)及預測之量測向量(Predicted measurement vector)，Observed measurement vector 即是感測器測量到的狀態值向量，將其定義為 z_t^i ：

Observed measurement vector :

$$z_t^i = \begin{pmatrix} r_t^i \\ \theta_t^i \\ \phi_t^i \end{pmatrix} \quad (26)$$

其中 r_t^i 代表在 t 時刻對第 i 個 Landmark 所觀測到的相對距離，量測方法如 3.2 節中所述，利用一非即時性之方法所求得； θ_t^i 代表對 Landmark 的水平視角，在全向式影像平面上，Landmark 所出現的方向即是實際相對於機器人所在的方向，因此利用 Landmark 在影像平面上對於鏡心的角度即可得知 θ_t^i (如圖 3-12)； ϕ_t^i 為對於 Landmark 的垂直視角(如圖 3-6 之 ϕ)，利用圖 3-9 中之垂直視角轉換公式，以及影像平面中 Landmark 相距鏡心的距離即可求得。

Predicted measurement vector 不同於 Observed measurement vector 由感測器所量測到的狀態值，而是由機器人所預測出來的預測值。基於前一時刻所估測出的 Landmark 位置(m_x, m_y, m_z)及預測出之機器人位置(x, y, θ)，算出預測之量測向量。圖 3-12 為機器人及 Landmark 之水平位置示意圖，可推得預測之 Landmark 相對距離 \hat{r}_t^i 及角度 $\hat{\theta}_t^i$ ，參考圖 3-7 機器人及 Landmark 之垂直位置示意圖，利用

已知固定之鏡心高度 h 可推得對 Landmark 預測之垂直視角 $\hat{\phi}_t^i$ ，Predicted measurement vector 如下(27)式：

$$\hat{z}_t^i = \begin{pmatrix} \hat{r}_t^i \\ \hat{\theta}_t^i \\ \hat{\phi}_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2} \\ a \tan 2(m_{i,y} - y, m_{i,x} - x) - \theta \\ a \tan 2(\sqrt{(m_{i,x} - x)^2 + (m_{i,y} - y)^2}, h - m_{i,z}) \end{pmatrix} \quad (27)$$

至此可得知一個重要的向量：

$$\text{Innovation vector} = z_t^i - \hat{z}_t^i$$

此向量代表著觀測值與預測值差異：此向量愈大代表實際觀測值與系統預測值差距很大，系統狀態值需作大幅度的更新修正，反之則代表實際值與預測值差異不大，只需小幅修正即可，故在 EKF 的 Update step 中有著重要的意義，結合 Kalman gain 之後，就是決定系統狀態估測值更新的主要依據。

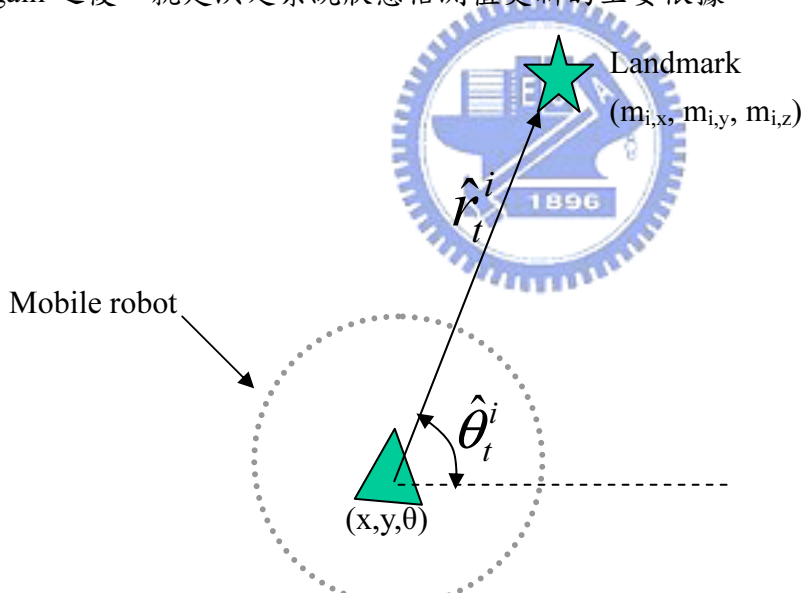


圖 3-12 機器人及 Landmark 之水平位置關係圖

3.3.3 演算法流程

本演算法參考 Thrun *et al.*[14]所提之 EKF SLAM 流程：

在 SLAM 中，Landmark 的狀態也包含在系統狀態之內，故系統狀態向量(state vector) y_t 中有 $3N+3$ 個元素，前三個為機器人的姿態資訊，之後為 N 個 Landmark 的狀態，每個 Landmark 擁有三個狀態元素；而 Covariance matrix Σ_t 則為 $3N+3 \times$

3N+3 維矩陣。

3.3.3.1 Prediction step

當機器人移動時，其狀態會依運動模型，(22)式而改變，而在 SLAM 中，移動模型要隨著 SLAM 的 State vector 而改變，新的狀態預測模型：

$$y_t = y_{t-1} + \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (28)$$

其中右式第二項的前三個元素即(22)式之運動模型，代表機器人的姿態依運動模型的預測而改變；其他元素則全為零，代表機器人的移動不會影響 Landmark 的狀態，其狀態均保留與前一刻相同。

將(28)式簡化：

$$y_t = y_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix} \quad (29)$$

$$F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & \underbrace{0 \cdots 0}_{3N \text{ columns}} \end{pmatrix} \quad (30)$$

F 為一個維度轉換矩陣，將 3 維矩陣擴展成 3N+3 維。

如(22)式機器人姿態向量 $x_t = g(u_t, x_{t-1})$ ，為了將此模型使用在 EKF 中，故將 $g(u_t, x_{t-1})$ 在 μ_{t-1} 處以一階泰勒展開式線性近似：

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \quad (31)$$

其中 μ_{t-1} 為 EKF 所估測出的 t-1 時刻的 state，Jacobian matrix G_t 即為 $g(u_t, x_{t-1})$ 對


x_{t-1} 之偏導數：

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mu_{t-1,x}} & \frac{\partial x'}{\partial \mu_{t-1,y}} & \frac{\partial x'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial y'}{\partial \mu_{t-1,x}} & \frac{\partial y'}{\partial \mu_{t-1,y}} & \frac{\partial y'}{\partial \mu_{t-1,\theta}} \\ \frac{\partial \theta'}{\partial \mu_{t-1,x}} & \frac{\partial \theta'}{\partial \mu_{t-1,y}} & \frac{\partial \theta'}{\partial \mu_{t-1,\theta}} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & \frac{v_t}{\omega_t} (-\cos \mu_{t-1,\theta} + \cos(\mu_{t-1,\theta} + \omega_t \Delta t)) \\ 0 & 1 & \frac{v_t}{\omega_t} (-\sin \mu_{t-1,\theta} + \sin(\mu_{t-1,\theta} + \omega_t \Delta t)) \\ 0 & 0 & 1 \end{pmatrix} \quad (32)$$

另求出 Jacobian matrix V_t ， $g(u_t, x_{t-1})$ 對 u_t 之偏導數：

$$V_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial u_t}$$

$$= \begin{pmatrix} \frac{\partial x'}{\partial v_t} & \frac{\partial x'}{\partial \omega_t} \\ \frac{\partial y'}{\partial v_t} & \frac{\partial y'}{\partial \omega_t} \\ \frac{\partial \theta'}{\partial v_t} & \frac{\partial \theta'}{\partial \omega_t} \end{pmatrix}$$


$$= \begin{pmatrix} -\frac{\sin \theta + \sin(\theta + \omega_t \Delta t)}{\omega_t} & \frac{v_t (\sin \theta - \sin(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \cos(\theta + \omega_t \Delta t) \Delta t}{\omega_t} \\ \frac{\cos \theta - \cos(\theta + \omega_t \Delta t)}{\omega_t} & -\frac{v_t (\cos \theta - \cos(\theta + \omega_t \Delta t))}{\omega_t^2} + \frac{v_t \sin(\theta + \omega_t \Delta t) \Delta t}{\omega_t} \\ 0 & \Delta t \end{pmatrix} \quad (33)$$

V_t 為控制空間雜訊與狀態空間雜訊之近似轉換，基於(24)式速度控制空間之 Covariance matrix M_t ，此刻新增的移動雜訊之 Covariance matrix R_t 可由(34)式轉換式得到：

$$R_t = V_t \cdot M_t \cdot V_t^T \quad (34)$$

而在 EKF SLAM 中，包含 Noise 之完整的移動模型可表示為：

$$y_t = y_{t-1} + F_x^T \underbrace{\begin{pmatrix} -\frac{v_t}{\omega_t} \sin \theta + \frac{v_t}{\omega_t} \sin(\theta + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \theta - \frac{v_t}{\omega_t} \cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}}_{g(u_t, y_{t-1})} + N(0, F_x^T R_t F_x) \quad (35)$$

與 EKF 同理，將 $g(u_t, y_{t-1})$ 以一階泰勒展開線性近似可得 Jacobian matrix G_t ：

$$G_t = I + F_x^T \begin{pmatrix} 0 & 0 & \frac{v_t}{\omega_t} (-\cos \mu_{t-1, \theta} + \cos(\mu_{t-1, \theta} + \omega_t \Delta t)) \\ 0 & 0 & \frac{v_t}{\omega_t} (-\sin \mu_{t-1, \theta} + \sin(\mu_{t-1, \theta} + \omega_t \Delta t)) \\ 0 & 0 & 0 \end{pmatrix} F_x \quad (36)$$

而新增的移動雜訊 Covariance R_t 則以 F_x 擴展成： $F_x^T R_t F_x$ (37)

系統 State 之 Covariance matrix $\bar{\Sigma}_t$ 以(34)式、(35)式、(37)式整合為：

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x \quad (38)$$

(其中 Σ_{t-1} 為前一時刻所估測出之系統 Covariance)

3.3.3.2 Update step

由(25)式，包含 Noise 之完整的 Measurement model 表示如下式：

$$\hat{z}_t^j = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ a \tan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ a \tan 2(\sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2}, h - m_{j,z}) \end{pmatrix}}_{h(y_t, j)} + N \left(0, \underbrace{\begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\theta & 0 \\ 0 & 0 & \sigma_\phi \end{pmatrix}}_{Q_t} \right) \quad (39)$$

此式代表對第 j 個 Landmark 之觀測模型，其中 Q_t 為 Observation noise 之 Covariance matrix， σ_r 、 σ_θ 、 σ_ϕ 分別代表 sensor 對 landmark 的觀測值 r 、 θ 、 ϕ 的標準差，利用如圖 3-5、圖 3-9 之實驗，讓機器人在環境中行走一段距離後，將機器人對環境特徵點的估測值記錄下來，並計算出其標準差，經數次實驗後得到

$\sigma_r=0.41\text{m}$ 、 $\sigma_\theta=3.1^\circ$ 、 $\sigma_\phi=6.4^\circ$ ，此數值即代表 sensor 的 noise，並將之用在 Kalman gain 的計算中。

將 Observation function $h(y_t, j)$ 近似為線性方程式：

$$h(y_t, j) \approx h(\bar{\mu}_t, j) + H_t^j (y_t - \bar{\mu}_t) \quad (40)$$

其中 H_t^j 為 $h(y_t, j)$ 對於系統狀態 y_t 之偏導數，而此時 $h(y_t, j)$ 僅對機器人的姿態

x_t 及第 j 個 Landmark 的資訊 m_j 偏微有意義，故將 H_t^j 改寫：

$$H_t^j = h_t^j F_{x, j} \quad (41)$$

h_t^j 代表對 x_t 及 m_j 的偏導數，而 $F_{x, j}$ 則是將 3×6 維之 h_t^j 擴展成 $3 \times (3N+3)$ 之轉換矩陣。

$$h_t^j = \frac{\partial h(\bar{\mu}_t, j)}{\partial y_t} = \begin{pmatrix} \frac{\partial r_t^j}{\partial \bar{\mu}_{t,x}} & \frac{\partial r_t^j}{\partial \bar{\mu}_{t,y}} & \frac{\partial r_t^j}{\partial \bar{\mu}_{t,\theta}} & \frac{\partial r_t^j}{\partial \bar{\mu}_{t,m_{jx}}} & \frac{\partial r_t^j}{\partial \bar{\mu}_{t,m_{jy}}} & \frac{\partial r_t^j}{\partial \bar{\mu}_{t,m_{jz}}} \\ \frac{\partial \theta_t^j}{\partial \bar{\mu}_{t,x}} & \frac{\partial \theta_t^j}{\partial \bar{\mu}_{t,y}} & \frac{\partial \theta_t^j}{\partial \bar{\mu}_{t,\theta}} & \frac{\partial \theta_t^j}{\partial \bar{\mu}_{t,m_{jx}}} & \frac{\partial \theta_t^j}{\partial \bar{\mu}_{t,m_{jy}}} & \frac{\partial \theta_t^j}{\partial \bar{\mu}_{t,m_{jz}}} \\ \frac{\partial \phi_t^j}{\partial \bar{\mu}_{t,x}} & \frac{\partial \phi_t^j}{\partial \bar{\mu}_{t,y}} & \frac{\partial \phi_t^j}{\partial \bar{\mu}_{t,\theta}} & \frac{\partial \phi_t^j}{\partial \bar{\mu}_{t,m_{jx}}} & \frac{\partial \phi_t^j}{\partial \bar{\mu}_{t,m_{jy}}} & \frac{\partial \phi_t^j}{\partial \bar{\mu}_{t,m_{jz}}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & \frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 & \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & \frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 \\ \frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q_t}} & \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & -1 & \frac{m_{j,y} - \bar{\mu}_{t,y}}{\sqrt{q_t}} & \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & 0 \\ \frac{q_t}{(m_{j,x} - \bar{\mu}_{t,x})(h - m_{j,z})} & \frac{q_t}{(m_{j,y} - \bar{\mu}_{t,y})(h - m_{j,z})} & 0 & \frac{q_t}{(m_{j,x} - \bar{\mu}_{t,x})(h - m_{j,z})} & \frac{q_t}{(m_{j,y} - \bar{\mu}_{t,y})(h - m_{j,z})} & \frac{\sqrt{q_t}}{q_t + (h - m_{j,z})^2} \\ \frac{q_t}{(q_t + (h - m_{j,z})^2) \cdot \sqrt{q_t}} & \frac{q_t}{(q_t + (h - m_{j,z})^2) \cdot \sqrt{q_t}} & 0 & \frac{q_t}{(q_t + (h - m_{j,z})^2) \cdot \sqrt{q_t}} & \frac{q_t}{(q_t + (h - m_{j,z})^2) \cdot \sqrt{q_t}} & \frac{\sqrt{q_t}}{q_t + (h - m_{j,z})^2} \end{pmatrix} \quad (42)$$

其中 $q_t = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2$

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & \underbrace{0 \cdots 0}_{3j-3} & 0 & 0 & 1 & \underbrace{0 \cdots 0}_{3N-3j} \end{pmatrix} \quad (43)$$

由上述推導結果，可求得

$$\text{Kalman gain :} \quad K_t^j = \bar{\Sigma}_t H_t^{jT} (H_t^j \bar{\Sigma}_t H_t^{jT} + Q_t)^{-1} \quad (44)$$

利用 Kalman gain 及 Innovation vector 更新系統狀態估測向量 $\bar{\mu}_t$:

$$\bar{\mu}_t = \bar{\mu}_t + K_t^j (z_t^j - \hat{z}_t^j) \quad (45)$$

並更新系統之 Covariance matrix $\bar{\Sigma}_t$:

$$\bar{\Sigma}_t = (I - K_t^j H_t^j) \cdot \bar{\Sigma}_t \quad (46)$$

至此，EKF SLAM 已完成一個循環，重覆循環步驟完成機器人同時定位且建立環境地圖的功能。表 3-1 即為 EKF SLAM 演算法之流程步驟。

表 3-1 EKF SLAM 演算法之步驟流程

EKF SLAM Algorithm ($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)	
Prediction Step :	
1.	$\mu_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{\omega_t} \sin \mu_{t-1, \theta} + \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \frac{v_t}{\omega_t} \cos \mu_{t-1, \theta} - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) \\ \omega_t \Delta t \end{pmatrix}$
2.	$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T R_t F_x$
Update Step :	
3.	for all observed features z_t^j
4.	if Landmark j never seen before
5.	add Landmark j to map
6.	endif
7.	$\hat{z}_t^j = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ a \tan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ a \tan 2(\sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2}, h - m_{j,z}) \end{pmatrix}$
8.	$H_t^j = h_t^j F_{x,j}$
9.	$K_t^j = \bar{\Sigma}_t H_t^{jT} (H_t^j \bar{\Sigma}_t H_t^{jT} + Q_t)^{-1}$
10.	$\bar{\mu}_t = \bar{\mu}_t + K_t^j (z_t^j - \hat{z}_t^j)$
11.	$\bar{\Sigma}_t = (I - K_t^j H_t^j) \cdot \bar{\Sigma}_t$
12.	endfor
13.	$\mu_t = \bar{\mu}_t, \Sigma_t = \bar{\Sigma}_t$
14.	return μ_t, Σ_t

第四章 參考點轉換

4.1 參考點比對

在使用 Laser scanner 之定位與地圖建立的研究中，scan-matching 是個常用的方法，利用兩組雷射資訊的比對以估測出與 reference scan 的差異，藉此判定與 reference scan 相對位置關係，整個地圖就是以一連串的 reference scan 所組成，利用機器人即時的雷射掃描資訊與 reference scan 比對，即可得知機器人在地圖中的位置。[28]即是利用 Laser scanner 與 scan-matching 的概念來完成定位與地圖建立的功能，將 current scan 與 reference scan 做比對，可得到與參考點的位置差異，地圖中建立了數個已知位置 reference scan，便能利用這些參考點所掃描到的雷射資訊，來決定出機器人的正確位置。

在[6]中亦是使用 scan-matching 的概念，不同的是使用的感測器資訊不是 Laser 的距離資訊而是攝影機的影像資訊，利用擷取到的影像完成 visual scan-matching，並影像畫面做為視覺參考點。機器人以即時影像與地圖中的參考點影像比對，即可判定出目前影像中所看到的特徵點與地圖中特徵點的對應關係，以此對地圖中的特徵點位置作更新的動作並定位出機器人本身的位置。

在基本的 SLAM 演算法中，當感測器觀測到一個特徵點時，要以此點之資訊與地圖中所有的特徵點做比對，以找出現在觀測到的這個特徵點是地圖上的哪個特徵點，才能更新此點的位置資訊，若不存在於地圖中，則視為新的特徵點加入地圖中。在 visual scan-matching 的方法中則不同，擷取到的影像特徵點不需與地圖中所有的特徵點比對以找出對應關係，利用在地圖上已建立的視覺參考點資訊，特徵點只需與參考點影像比對即能找出現在觀測到的這個特徵點是地圖上的哪個特徵點，增加了特徵點與地圖對應的速度。

4.2 參考點轉換演算法

本研究之主要目的就是希望能夠藉由全向式攝影機豐富的影像資訊及廣闊的視野，幫助機器人在室內走廊的環境下完成自我定位的功能，配合自主移動的

行為，讓機器人能移動至目的地，完成機器人在室內走廊環境下的巡航功能。在第三章中已說明了利用 EKF SLAM 來達成機器人自我定位的方法，由於使用的感測器是攝影機，對於偵測辨識環境中的 Landmark 需要兩張影像方可比對出相同的特徵點；在本文中所使用的比對策略為以起始點影像為參考點，機器人移動後所擷取到的畫面均與起始點影像作比較，以此方法定位出機器人自身的位置並可建立出起始點周遭環境特徵點之地圖。

但此方法並不能讓機器人走出較遠的距離，因為機器人移動的愈遠，與起始點影像所能比對到的特徵點愈少，因此若要讓機器人走出較長的距離，且要能比對出足夠的特徵點來定位，則不能再以起點之單一參考點影像來做為比對的基準，勢必要改變參考點的影像。為了解決上述問題，本論文設計一參考點建立及轉換的流程。當機器人移動到新地點時則建立此處的參考點影像；每到一個新的區域時，重新設定參考點影像，並將先前的參考點資訊存入資料庫中，讓各區域都各有做為基準的參考點影像，機器人即可持續行走更長的距離，並依據各參考點影像比對出特徵點來定位出機器人的位置。判斷參考點建立的時機決定於移動的距離及特徵點的數目，距離是一個最大的因素，因此若機器人移動超過設定的距離時，則重新設定此時的位置為新的參考點；除了距離之外特徵點的數目也是判斷的標準之一，當比對成功的特徵點愈來愈少時，就表示當前的參考點影像已不敷使用，亦是要替換參考點的時機之一。

在替換參考點的同時，首先要先將原本的參考點區域相關資訊存入資料庫，如參考點影像、特徵點資訊、環境地圖(特徵點位置)、各個特徵點之 Covariance... 等，並且將新舊兩個參考點影像作比較，將共有的特徵點及其 Covariance 相關資料留下，作為新參考點區域已知的環境特徵點，也因此新舊兩區域有了共同的連繫，不會互不相關。

在[6]中因為使用的攝影機視角有限，故當旋轉超過一定角度時，會無法與原參考點影像成功比對，要建立新的參考點，例如在回到已走過的地點時，因為與之前建立的參考點影像視角不同無法比對，而再次建立參考點，相同的特徵點

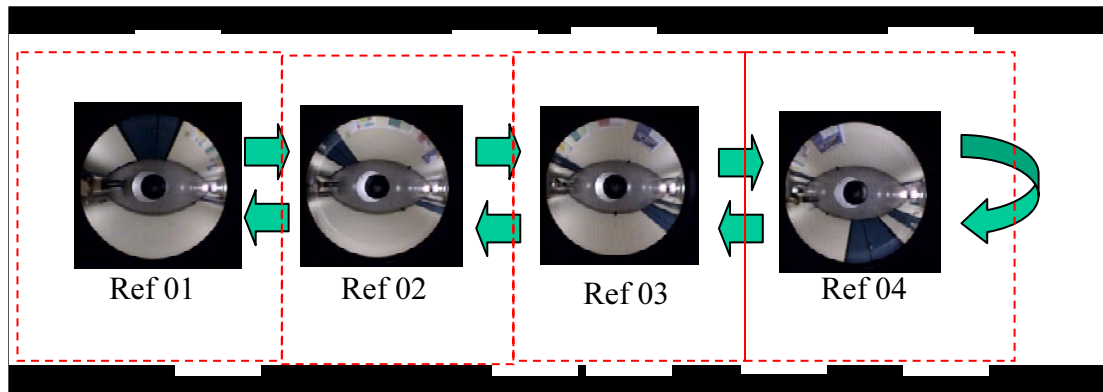


圖 4-1 參考點影像負責區域示意圖

會因為不同的視角而重覆定義成不同的點。在本論文中，因為使用的全向式攝影機擁有 360 度視角，即使機器人旋轉了 180 度，在影像平面中仍能辨識出是同一區域，故在機器人回頭走向原本已走過的路徑時，仍可在影像中辨識出之已經估測好位置的特徵點，此種情況下不需執行新參考點建立的步驟，而是將之前舊的參考點資訊取出才是最好的選擇。這個情況就好像圖 4-1，每個參考點影像都有其負責的區域範圍，在各區域(Ref01、Ref02、Ref03、Ref04)內均以各區的參考點影像作比較來定位出機器人的位置，到達了新的地點則建立新的參考點，回到了走過的地方，則從資料庫中取回舊有的資料幫助機器人定位。

以距離參考點設定位置的遠近來判別是否進入舊有的參考點區域，當要進入舊的參考點區域時，則將原本的參考點區域資料存入資料庫，以備以後使用，並從資料庫中找出舊有的參考點資料，以舊有的參考點影像為基準，持續的運行機器人定位的演算法。如此一來不僅可以減少參考點建立的數量，更可以降低同一特徵點被重覆定義的情況發生，降低地圖與定位系統的複雜度。

圖 4-2 即參考點轉換之流程圖，其中 Δs 代表參考點區域範圍的最大半徑， N 代表特徵點比對成功數目下限。首先以移動的距離及特徵點比對數目作為判斷的基準，當距離大於設定的上限值或是特徵點數目小於設定的下限，則要開始進入參考點轉換的步驟；搜尋資料庫中距離機器人此時位置最近的參考點，判斷機器人是否在那參考點的區域範圍之內；若是，則代表進入了舊有的參考點區域，並取出資料庫中舊有的資訊來使用；若否，則表示機器人此時已進入了一個新的

地點，將原本的資訊存入資料庫後，建立新的參考點，並開始建立出新的參考點區域環境地圖。

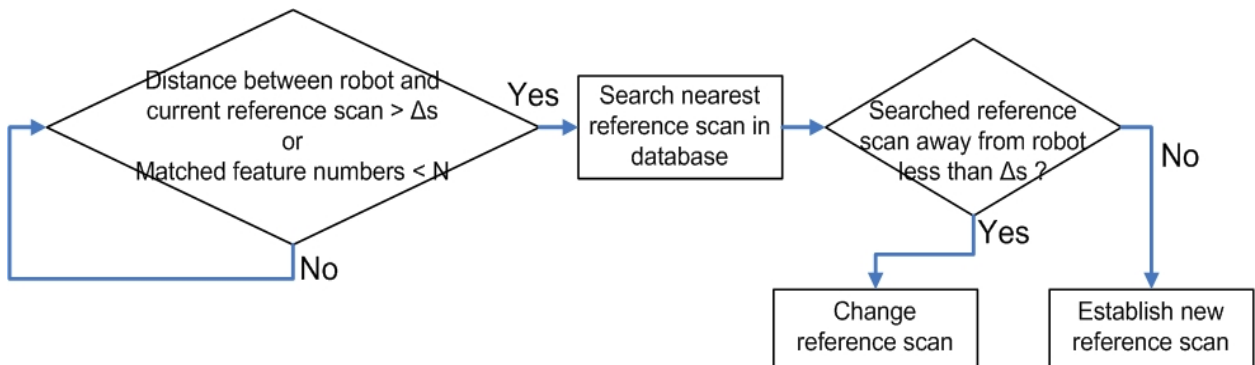


圖 4-2 參考點轉換策略之流程圖

使用此參考點建立及轉換的流程，當機器人移動到新地點時則建立此處的參考點影像，回程時則從資料庫中取回之前已存入的參考點資訊，如此將環境中不同地點的全向式影像特徵整合，讓機器人不僅能建立出環境特徵地圖，並可在大範圍環境中持續重覆行走，達成機器人長距離導航的功能。

圖 4-3 為系統資料庫之結構示意圖，主要以各個參考點影像(Ref_01、Ref_02...)為群組，每個參考點下包含其擁有的特徵點資訊，如影像特徵、位置資訊...等，每個參考點間亦具有共同的特徵點，在前一個參考點區域中所得到的估測值亦能被下一個參考點所使用。以下步驟為參考點轉換與資料庫存取之流程。

database

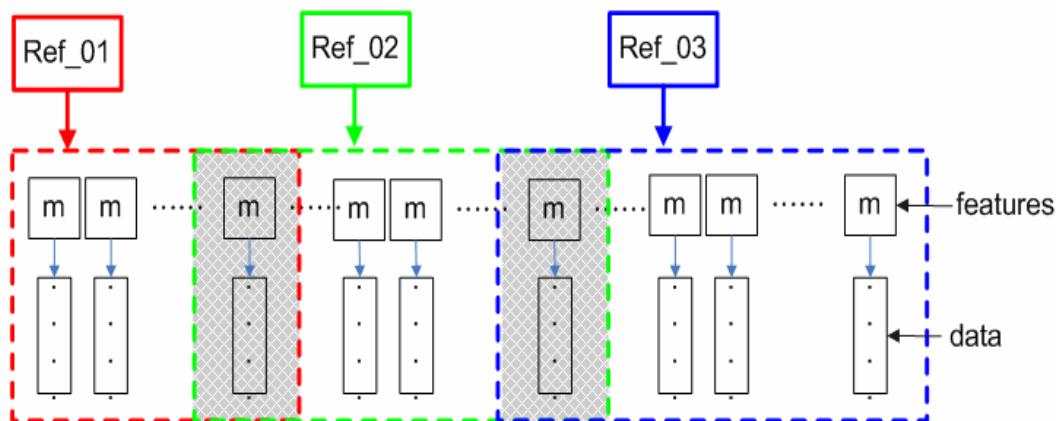


圖 4-3 資料庫之結構示意圖

Step1 :

即時影像與參考點影像比對，使用 EKF-SLAM 完成對機器人位置與特徵點的估測。

Step2 :

判斷機器人此刻與參考點的距離是否超過參考點區域最大半徑，若否，代表機器人仍在同一參考點區域內，回到 Step1 繼續下一個影像 frame 的運算。若是，代表機器人已離開參考點區域，即開始轉換參考點的步驟。

Step3 :

搜尋 database 中區域半徑內是否有其他的參考點，若是，則將原參考點影像中的特徵點資訊存入並更新 database，並將參考點轉換成已搜尋到的舊有參考點。若否，將原參考點影像中的特徵點資訊存入並更新 database，此刻的即時影像則新增存入 database，視為一個新的參考點，與原參考點之特徵點對應關係，即為兩個參考點所共有的特徵，並在 database 中標記。切換參考點後再次回到 Step1 繼續下一個影像 frame 的運算。



第五章 實驗結果

經過上述章節的演算法設計描述後，在本章中設計了幾個實驗來驗證各章節中所述演算法之性能。最後以一個機器人室內導航的情境展示實驗來驗證本論文所設計之方法能實現機器人室內導航的功能。

在 5.2 節中將介紹實驗用之機器人，包括了機器人本體、運算核心、及攝影機系統，並說明機器人的硬體控制架構。5.3 節將介紹環境特徵偵測之實驗，此實驗主要用以驗證第二章中所述之演算法的可行性，利用不同位置、不同角度所拍攝下之影像畫面，偵測、辨識出相同的特徵點，確定機器人在移動或是旋轉的情況下，影像處理系統仍能偵測出足夠的特徵點供定位演算法使用。5.4 節將介紹 EKF-SLAM 之初步測試實驗，此實驗主要是用以驗證第三章所述之 EKF 演算法之正確性，利用手動遙控機器人移動的實驗，比較 Odometer 與 EKF-SLAM 演算法所定位出的移動軌跡，並確定環境特徵點地圖是否已被正確記錄下來。5.5 節中以前後來回行走實驗來測試第四章所述參考點轉換策略，利用往前行走來測試參考點是否能逐一被建立出來，利用往回行走來測試是否能正確轉換回資料庫中已存在的參考點，以此實驗來證實參考點轉換之策略的正確性。5.6 節為機器人室內導航的情境展示實驗，實驗中模擬機器人從一房間內為起點，能夠主動走出房間並在走廊中移動到遠處的目標點，接著迴轉後沿原路回到原本的房間內，此實驗展示了機器人能夠自主的移動到所要求的目標位置，並具有重複性，能回到原來的地點，達成機器人室內導航的功能。

5.1 機器人硬體介紹

實驗所使用的機器人外觀如圖 5-1，其主要規格：外型(長×寬×高)為 450×480×1030(mm)，移動速度最高約 1m/sec，輪徑 150mm，電池規格 12V 7.2Ah 鉛酸電池。

機器人本體建構在一輪式運動平台上，此移動平台採用雙獨立驅動輪式之運動機構，利用對兩馬達的速度控制實現機器人在平面上之運動；移動平台後方另

有一被動式自由輪，三輪式的設計使機器人能夠平順穩定的在室內環境中移動。
機器人運算核心為威強公司出產之工業電腦主機板(NOVA-6612)，搭載 2GHz 之 CPU，主要用以影像處理及機器人之運動控制。



圖 5-1 機器人實體圖

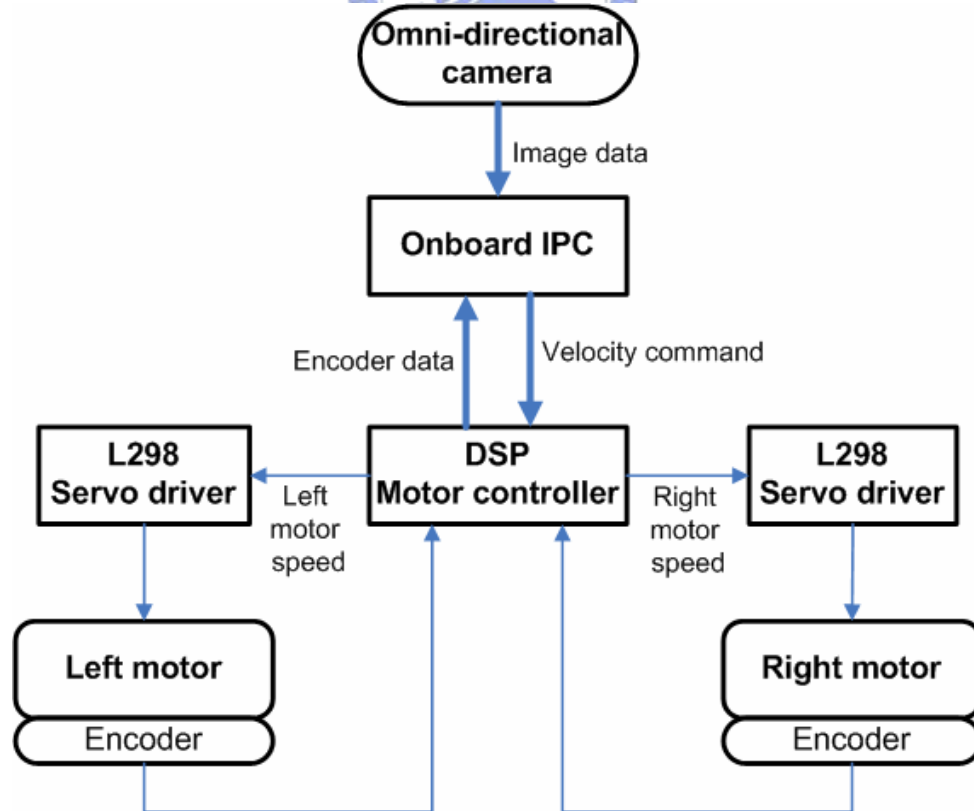


圖 5-2 機器人硬體架構

圖 5-2 為機器人硬體架構，以工業電腦為運算核心，攝影機為主要的感測器，馬達上有 Odometer 回授機器人移動的資訊，本機器人的 Odometer 之解析度為 500 lines，取樣時間為 51 ms，利用此資訊及式(3-8)之運動模型即可推算出機器人之移動軌跡。以電腦發出控制命令給馬達控制卡，經由馬達驅動器驅動馬達的運轉，完成機器人的移動控制。

機器人頂端之攝影機為日本 Vstone 公司所製之全向式攝影機，如圖 5-3，型號為 VS-C450N-TR，其主要規格如下：尺寸為 59(Φ)x199(H) mm，可視範圍：水平以上 15 度以下 60 度，有效影像最大像素 768*494，重量 228g。

機器人上裝置有 Imagenation 公司之影像擷取卡 PXC-200，用以擷取全向式影像系統之 CCD 所拍攝到的畫面，利用此影像擷取卡可以每秒 30 張 Frame 的速度擷取 640*480 Pixel² 大小的影像。

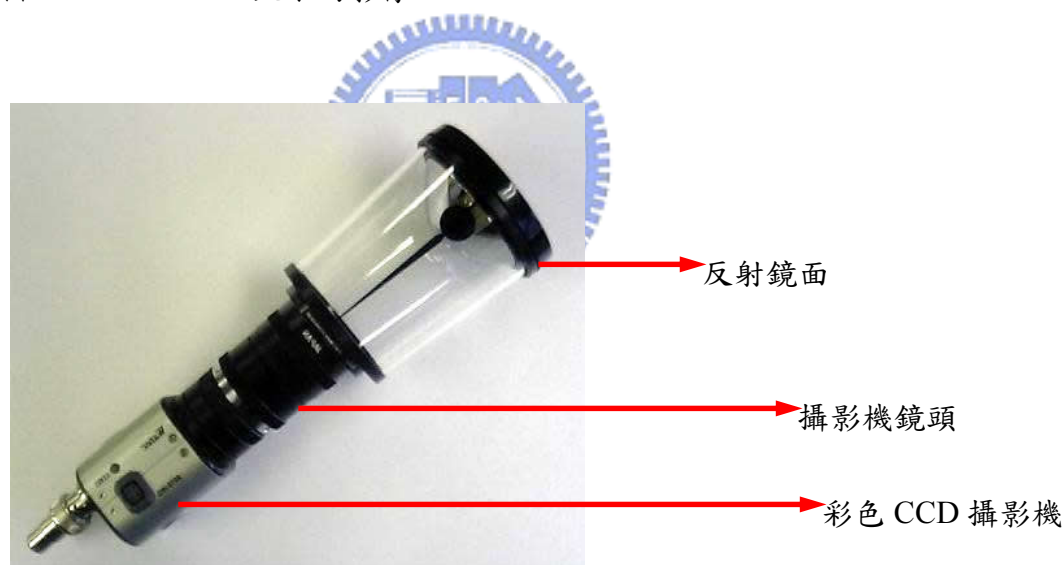


圖 5-3 Vstone VS-C450N-TR 全向式攝影機

5.2 環境特徵點辨識實驗

使用第二章所述之特徵點辨識演算法，實現在全向式影像中擷取出環境特徵點及特徵點比對的方法。利用全向式攝影機在走廊環境中拍下畫面來測試演算法的正確性，將不同位置、不同角度所拍下的數個畫面作比對，以驗證演算法的穩定性。測試過程中機器人平移一段距離後，再與原始位置的影像作比對，測試機器人移動後是否還能辨識到特徵點；另原地旋轉後，再與原始位置的影像作比

對，測試是否當機器人旋轉時依舊能偵測得到相同的環境特徵點。

實驗中擷取 $500 \times 500 \text{ pixel}^2$ 大小的影像來作處理，圖 5-4 為兩張拍攝地點相差 1 公尺之全向式影像，圖中之之方點代表所偵測到特徵點之位置，以綠線連接表示兩張影像中的特徵點對應關係，可發現到大部份的特徵點都對應到另一張影像中相同的環境位置中，驗證了特徵點比對之演算法對於移動一段距離後的影像仍能有效的比對出相同特徵點之所在位置。

在圖 5-4 中仍可發現有數個特徵點還是有比對錯誤的情況發生，再套用 Outlier detector 後，其結果如圖 5-5 所示，經過錯誤比對點刪除的演算法後，共有 63 個特徵點比對成功，那些明顯比對錯誤的點都被刪除了，使特徵點比對的正確率更為提高。

雖然證明了在機器人移動情況下能有效的比對出特徵點，但經多次不同的實驗下發現，機器人移動超過 1.5 公尺後，與原始位置影像比對成功的特徵點數量會大幅的減少，因此欲比較的兩張影像其相對距離是有一定的限制的。圖 5-6 為兩張相同位置的影像，其中一張為機器人原始旋轉 45 度後才拍攝，在影像中可注意到，所有的物體均以鏡心旋轉了 45 度，連接之線條即為經過特徵點比對後的配對結果，共有 47 個特徵點比對成功。即使影像旋轉了，仍可比對出相對應的特徵點所在，證實了此特徵點比對具有對抗全向式影像旋轉的特性。

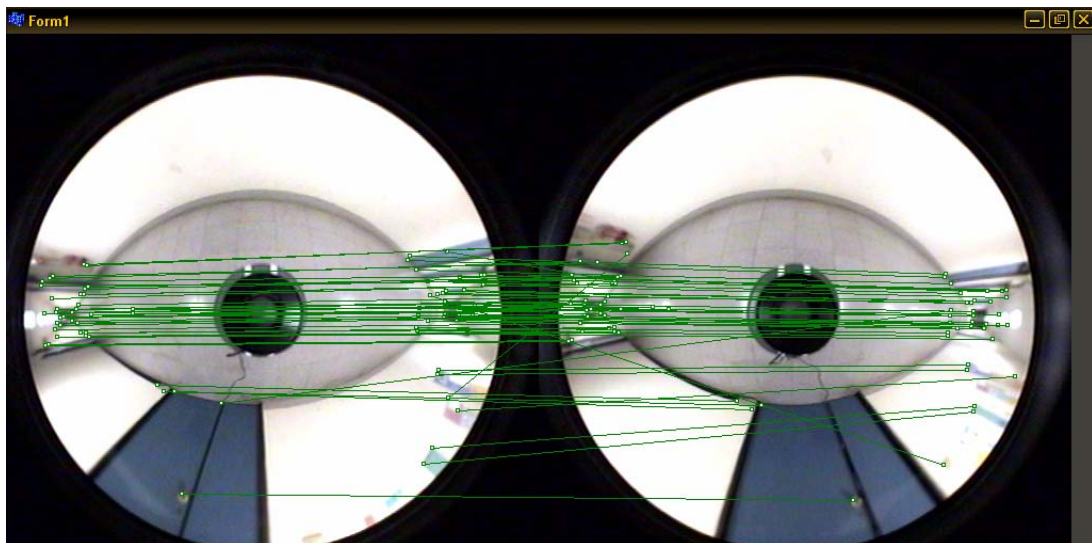


圖 5-4 特徵點比對結果

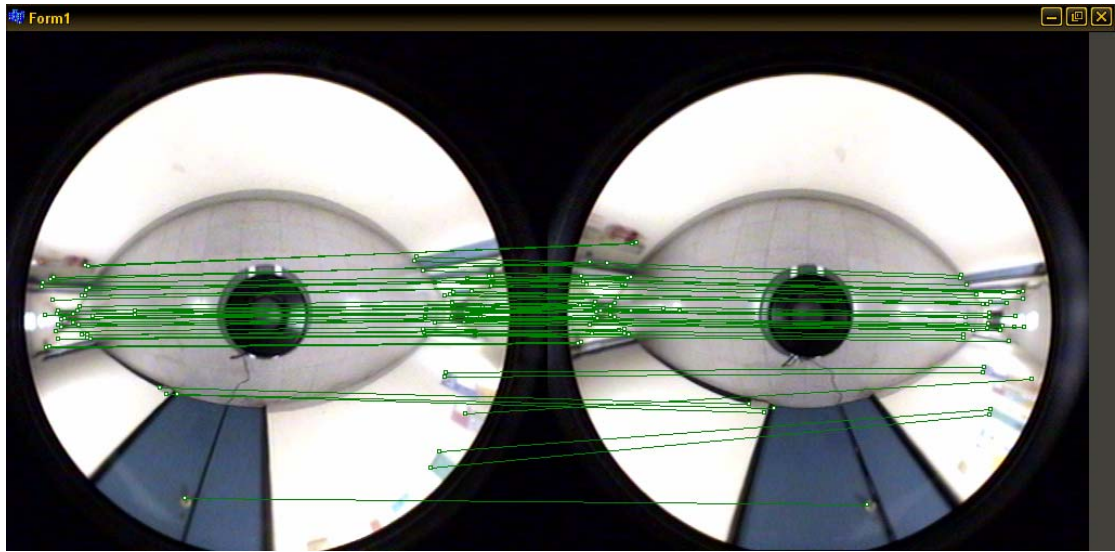


圖 5-5 特徵點比對結果(加入 Outlier detection)

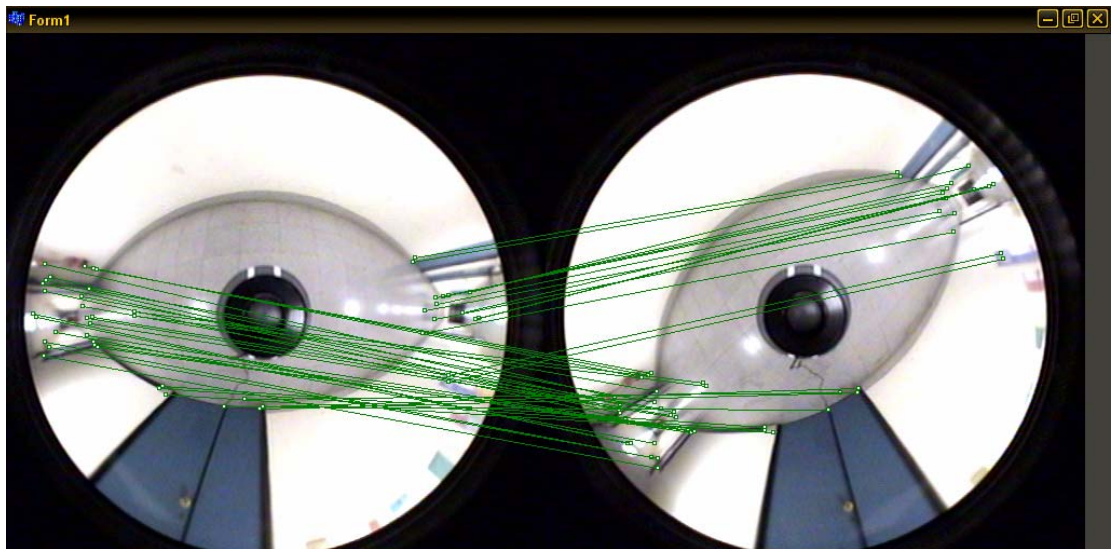


圖 5-6 特徵點比對結果(旋轉影像)

在實際測試中特徵點擷取及比對的流程所耗費的時間約為一秒。表 5-1 為數張距離、角度不同之全向式影像中特徵點比對及正確率的結果，其中 Matched point 為此影像有幾個比對成功的特徵點，Correct matched point 為比對正確無誤的特徵點，Rate 代表特徵點比對的成功率；測試內容為在實驗環境中取其中一地點之影像為比對的基準影像，之後移動一段距離或是旋轉角度，再次擷取影像後與基準影像做特徵點的比對。

從實驗結果中可發現當兩張影像相距到 1.5 公尺左右時，比對正確率開始有較顯著的下降，平移後之整體平均比對正確率為 87%；在影像旋轉後，正確率的

差異不大，平均的比對正確率為 93%。經上述實驗結果驗證，本特徵點比對之演算法在機器人移動或是旋轉的情況下，均能有效的辨識出環境特徵點，對特徵點的辨識平均有 90% 的正確率，這些特徵點即為機器人定位演算法所需要的 Landmark，以此結果作為定位演算法的 Input，幫助完成機器人定位的功能。

表 5-1 特徵點比對實驗結果

	Case 1 (-1m)	Case 2 (-0.5m)	Case 3 (0.5m)	Case 4 (1m)	Case 5 (1.5m)	Case 6 (2m)	Case 7 (2.5m)	Average correction rate
Matched point	29	30	30	23	17	12	16	
Correct matched point	28	29	30	22	13	8	12	
Rate	96%	96%	100%	95%	76%	67%	75%	87%
	CaseA (45°)	Case B (90°)	Case C (135°)	CaseD (180°)	Case E (225°)	Case F (270°)	CaseG (315°)	
Matched point	28	25	28	29	28	26	26	
Correct matched point	25	25	26	29	26	23	23	
Rate	89%	100%	93%	100%	93%	88%	88%	93%
total								90%

5.3 EKF SLAM 初步測試實驗

為了驗證機器人於走廊環境中移動時之定位功能，實驗過程以手動遙控機器人沿 2.8x1.2 公尺的矩形路徑，繞行五圈，實驗場景及行走範圍如圖 5-7，圖中虛線為行走之矩形範圍。使用以第三章為基礎之 EFK-SLAM 演算法，建立出環境特徵地圖及定位出機器人的座標。(47)式為 odometer 計算機器人位置之算式， Δx 代表在取樣時間 Δt 內，x 方向所移動的距離， Δy 代表在取樣時間 Δt 內，y 方向所移動的距離， $\Delta\theta$ 代表在取樣時間 Δt 內，機器人所旋轉的角度， v_l 、 v_r 分別為左、右輪的轉速， d 為兩個輪子之間距， θ 為機器人的朝向角。所估測到之特徵

點位置及機器人行走軌跡結果如圖 5-8(b)，並將 odometer 所記錄的軌跡繪製如圖 5-8(a)，圖中(0,0)位置為起點，而圖中的三角形位置代表繞行五圈後的終點位置所在。

$$\begin{aligned}\Delta x &= \Delta t * \frac{(vl + vr)}{2} * \cos \theta \\ \Delta y &= \Delta t * \frac{(vl + vr)}{2} * \sin \theta \\ \Delta \theta &= \Delta t * \frac{(vr - vl)}{d}\end{aligned}\quad (47)$$

在圖 5-8(b)，虛線代表牆面位置，所估測到的特徵點均落在虛線附近，由於在實際環境中，能擷取到的環境特徵點亦是分佈在牆面上，代表演算法中對特徵點的位置估測正確。因為在實驗過程中是以人工手動的方式遙控機器人沿矩形行走，因此繞行軌跡應是近似相同位置的矩形軌跡，圖 5-8(a)是直接記錄 Odometer 的數值所繪製的軌跡，由於機器人行走可能產生的誤差影響，記錄出來的矩形軌跡逐漸偏離初始的矩形位置，繞行五圈後的終點位置與起始點有 0.5 公尺以上的誤差，且會愈來愈大。而圖 5-8(b)EKF SLAM 所估測出的軌跡即較為正確，在初始矩形位置附近，繞行五圈後的終點位置與起始點有 0.25 公尺左右的誤差。此結果證明出定位演算法是可行的，並有一定程度的修正僅用 Odometer 定位所出現的誤差。



圖 5-7 實驗場景

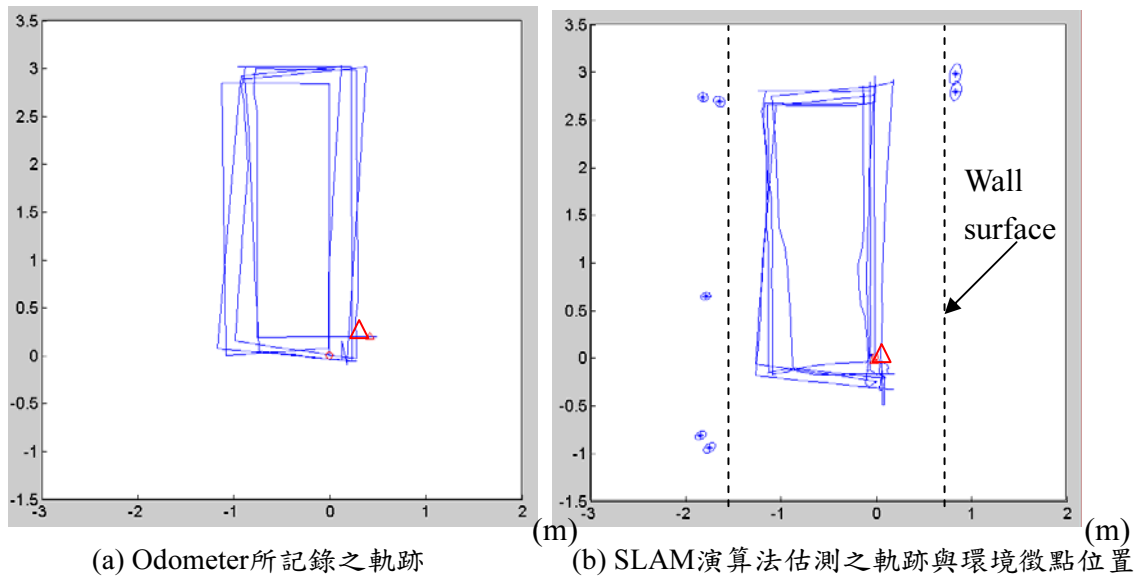


圖 5-8 繞行移動之實驗結果

5.4 定位參考點轉換之測試實驗

使用第四章之參考點轉換策略，讓機器人在走廊環境下自主運動較長距離，主要內容為在走廊上來回行走：機器人往前行走 5 公尺處，旋轉 180 度後往回走 10 公尺，再旋轉 180 度後行走 5 公尺，最後回到起始點，用此實驗來驗證參考點能順利的轉換，並同時定位出機器人的位置及建立環境特徵點地圖。

圖 5-9 為實驗結果，圖 5-9(a)~(c)圖為實驗過程中移動軌跡及特徵點估測之演進，起點為(0,0)，三角形代表特徵點建立的位置，數個橢圓形代表各特徵點的 Uncertainty 分佈。(a)圖代表機器人走至 2 公尺處的位置，此時較近的特徵點開始被建立在地圖上，但 Covariance 還很大；(b)圖代表已繞過 5 公尺處回到 0.5 公尺左右的位置，此時不僅遠處的特徵點被建立了出來，特徵點的 covariance 也因為多次的估測而收斂變小；(c)圖代表機器人已走回原點，並將路徑中的所觀測到的特徵點地圖建立出來；(d)圖中的虛線代表牆面的位置，可確認特徵點均落在牆面附近。

此實驗利用了參考點建立與轉換的策略，到了新的地點則建立新的參考點，回到舊的區域則利用舊的參考點，圖 5-9 之(d)圖中的五個紅色三角形即代表了在此實驗中共建立了 5 個參考點。

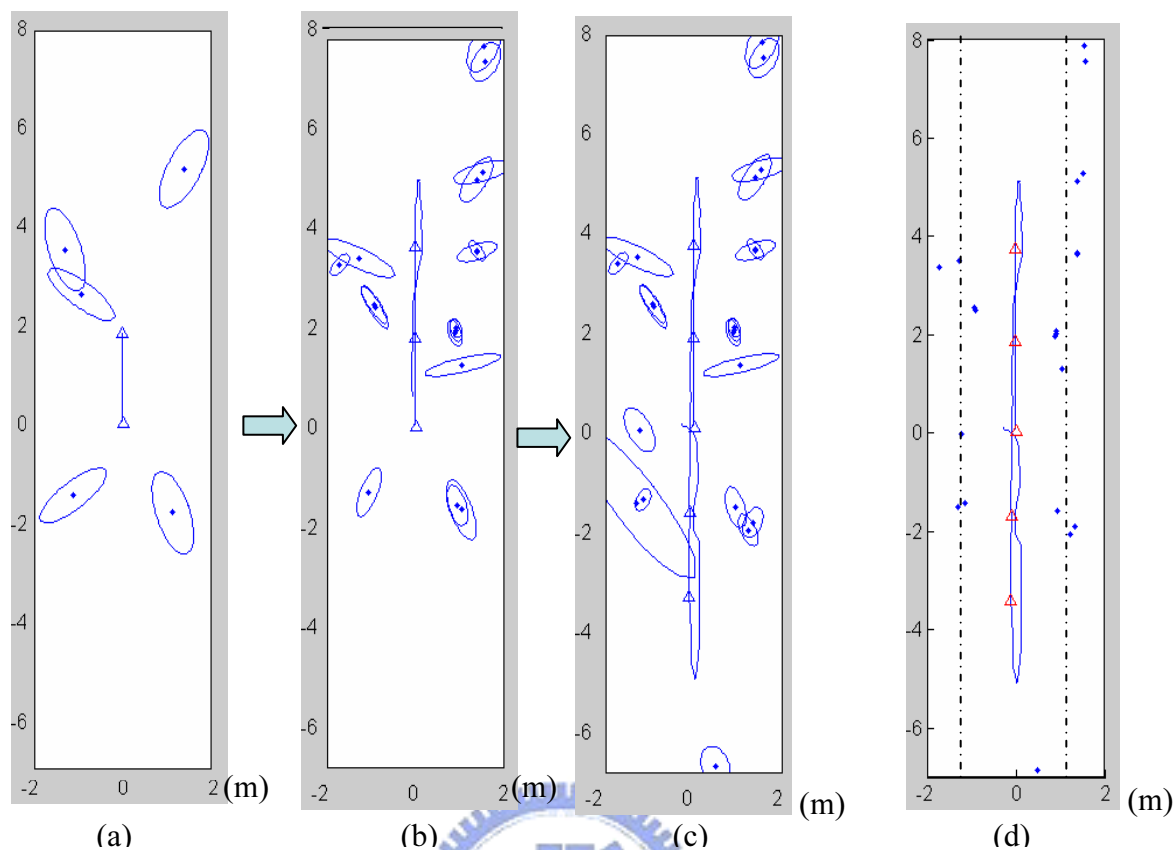


圖 5-9 來回行走之實驗結果

圖 5-10 為其中一次來回行走實驗中在四個位置之特徵點比對的擷圖，圖中 (a) 為在 0.5 公尺處之比對結果，此時的參考點影像即為起點 0 公尺處之影像 Ref_01；圖中 (b) 為在 2.5 公尺之結果，此時因距離 Ref_01 超過 2 公尺了，故新增一參考點 Ref_02 作為新的比對基準參考點；圖中 (c) 為移動到 5 公尺處後沿原路繞回 -0.5 公尺處時之擷圖，此時因為附近已有之前已建立過的參考點 Ref_01，故不需再建立新的參考點，而從資料庫中取出 Ref_01 做為比對的基準；圖中 (d) 為移動至 -2 公尺處之擷圖，因此地區尚未來過，故新增參考點 Ref_04 為比對的基準。

表 5-2 為五種不同距離之來回行走實驗之誤差結果，如 case 5 為機器人從起點出發，向前行走 15m 後原地旋轉 180 度，再走 15m 回到起點，移動共 30m 的距離後即記錄機器人此刻所估測出的位置座標，並與機器人之 ground truth 座標比對，計算其位置誤差。

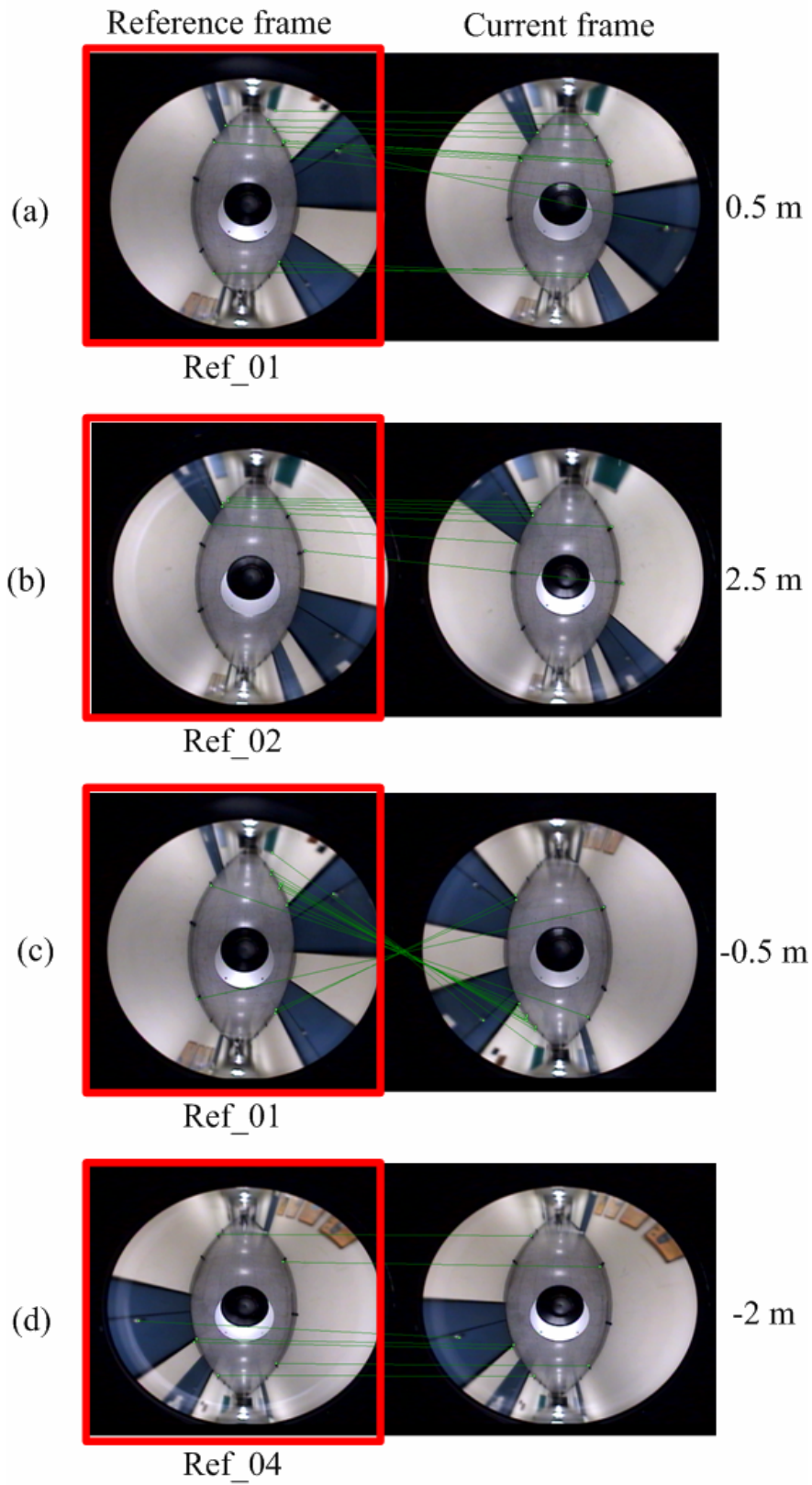


圖 5-10 Ref_01 等在 caption 中說明來回行走實驗中參考轉換之結果

表 5-2 來回行走之實驗結果誤差

Error	Case 1			Case 2			Case 3			Case 4			Case 5		
	10m			15m			20m			25m			30m		
	x	y	θ	x	y	θ	x	y	θ	x	y	θ	x	y	θ
Exp1	0.01	0.03	2	0.06	0.01	0	0.08	0.08	2	0.1	0.01	1	0.07	0.03	3
Exp2	0.09	0.01	3	0.09	0.01	2	0.12	0.02	2	0.12	0.03	2	0.09	0.04	1
Exp3	0.08	0	1	0.08	0.02	2	0.1	0.06	0	0.07	0.03	2	0.1	0.05	1
Exp4	0.07	0	2	0.1	0.03	1	0.15	0.04	3	0.06	0.03	2	0.15	0.05	8
Exp5	0.05	0.01	4	0.11	0.02	3	0.1	0.02	1	0.1	0.05	2	0.09	0.01	5
Average	0.06	0.01	3	0.9	0.02	2	0.11	0.04	2	0.09	0.03	2	0.1	0.04	4

上述實驗結果證實了各參考點間均能正確的轉換，經 30 公尺的行走距離後位置的平均誤差為 0.11m，驗證了本參考轉換之演算法能讓機器人定位行走更長的距離並建立出環境特徵點地圖。

圖 5-11 為機器人之 looping 實驗結果，於 8*10 公尺大小空間中以一 8 字型之封閉路徑移動，用以證實機器人對環境地圖的建立，在起點之地圖與繞行一圈後回到起點附近之地圖能夠銜接的上。

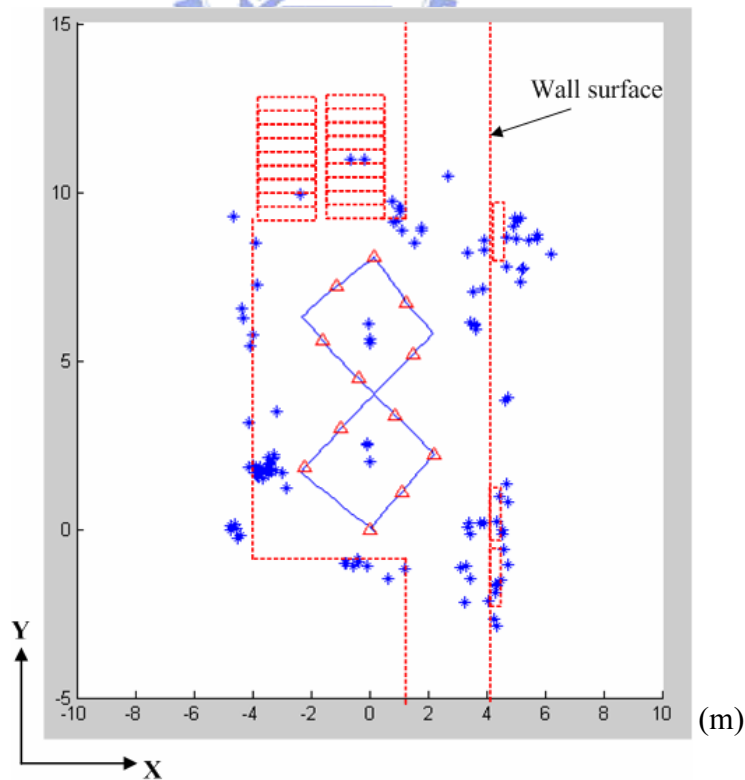


圖 5-11 機器人 8 字型移動之實驗結果

此實驗共建立了 12 個參考點，並有 145 個特徵點被建立於地圖上。經 5 次相同的實驗，量測機器人回到原點後的距離誤差，x 方向平均誤差為 0.16m、y 方向平均誤差為 0.06m。

5.5 機器人室內導航之情境展示實驗

本實驗之目的為驗證機器人在室內之長距離移動時，能夠同時定位出機器人本身的位置，建立出環境地圖，並證實機器人能藉由定位系統的幫助，達成自主式移動之巡航能力。實驗中機器人從一房間內為起點，開始主動走出房間並在走廊中移動到遠處 23 公尺處走廊盡頭事前已設定好的目標點，接著以一三角形軌跡迴轉後，沿原路回到原本的房間內。

圖 5-12 為實驗過程之記錄結果，其中菱形點代表特徵點的位置，藍線為定位系統所記錄下的移動軌跡，紅色三角形為移動過程中所建立的參考點位置，而紫線則為實際上房間及走廊上的牆壁位置。在機器人開始移動之前，先預設好五個目的地座標點作為機器人導航時依序要經過的地方：座標(0,-4)為房間內的起點，機器人首先開始主動的朝向第一個目標點(0,0)移動，穿越房後到達位於走廊上的第一個目標點(0,0)；接著旋轉 90 度後沿著走廊上行走，朝向第二個目標點(0,21)前進；在走廊盡頭有較寬的空間，機器人依序走向三個目標點(0,21),(2,22),(0,23)，沿一個三角形軌跡移動迴轉；轉向，走回先前的目標點(0,0)，最後穿越房門走回起點。全程除了預先設定好的目標點座標外，均是機器人主動的定位本身的位置，並自主式的移動到目標地點。

在本次實驗中，機器人從房間內開始移動後，即一直重覆的運行 SLAM 演算法，即時的定位出機器人的所在座標，讓機器人可得知如何移動到目標點，使其能成功的走出房間並到達遠處的目標點位置；而環境特徵點地圖亦同時能建立出來，在圖 5-12 中可發現特徵點大都落在兩側的牆面上；在移動的過程中，依照參考點建立及轉換的策略，成功的延長機器人的定位範圍，本次實驗共建立了 17 個參考點，如圖 5-11 中的三角形位置。圖 5-13 顯示實驗過程擷取之照片，其

中(a)為機器人開始出發之位置，(b)為機器人正走出房門，(c)為走出房間後轉向朝走廊移動，(d)為走廊上行走之途中，(h)為機器人走至最遠之 23 公尺處，(g)為機器人從走廊回程，正轉向房間方向，(f)為機器人正走進房門，(e)為回到起點，完成一次的巡航任務。

沿著設定好的數個目的地依序前進，是機器人導航中的基本情境，本實驗完成了此情境的展示，並且能建立環境地圖，讓機器人就算是到了未知的地點，亦能定位出自身的位置，證實了本論文所設計的機器人定位方法能夠幫助機器人達成室內導航的功能。

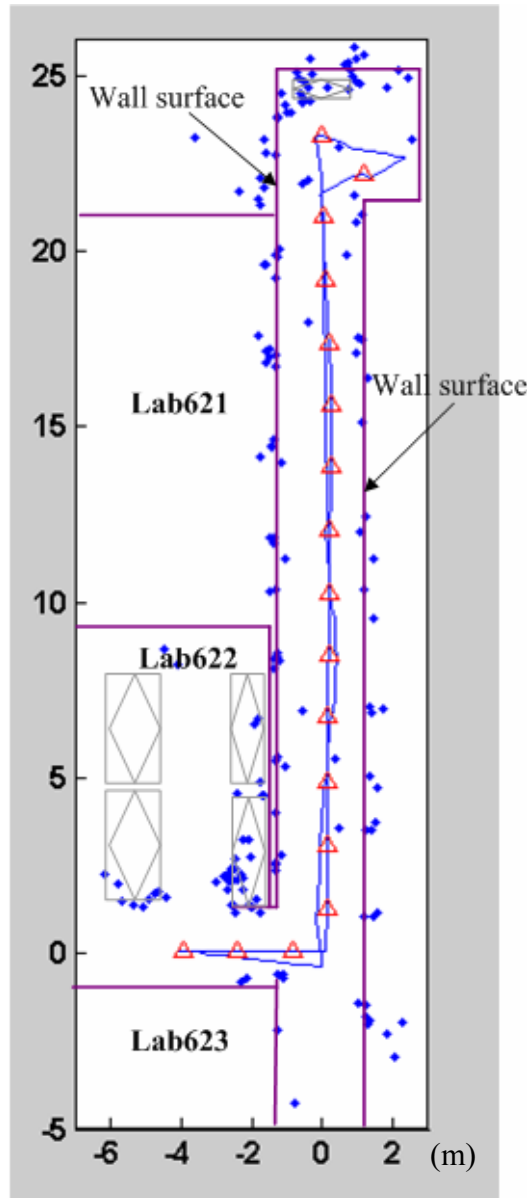


圖 5-12 走廊移動實驗結果

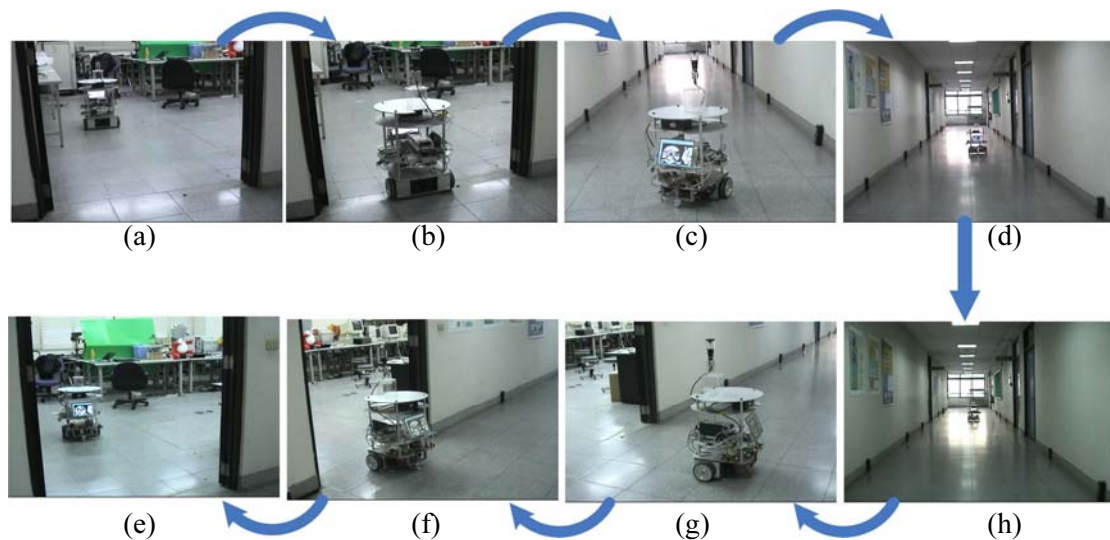


圖 5-13 實驗擷圖

5.6 討論

本論文主要是使用基於全向式影像及參考點轉換的定位方法：使用一套基於全向式影像特性的特徵點辨識方法，偵測環境中的特徵；定位方式是設定一參考點影像為基準，機器人移動後所擷取到的畫面與參考點影像作比較，以此方法定位出機器人自身的位置並建立出參考點周遭環境特徵點之地圖。

其中定位參考點之建立與轉換策略為最主要的特色，此方法能讓機器人的定位範圍距離延長，解決了因為機器人移動的愈遠，與參考點影像所能比對到的特徵點變少的問題。本方法與[6]所使用的 Visual reference scan matching 方法類似，利用建立新的參考點來延長可定位的距離，但本論文與[6]不同的地方在於使用的攝影機為全向式攝影機，利用全向式影像的 360 度視角，可對參考點影像作重覆的使用。

[6]中因為其攝影機的視角有限，故旋轉角度超過限定值時則需要再建立一個新的參考點，若機器人走回之前走過的地點時，仍需再次建立新的參考點。本研究中因使用的是全向式的影像，即使機器人旋轉了 180 度，仍可與參考點影像成功的比對出相同的特徵點，故以此特性設計一參考點轉換的策略，將所有已建立的參考點相關資料均存入資料庫中，當機器人走回到舊地區時不用再建立新的參考點，而從資料庫中取出已有的資訊來使用即可，就好像每個參考點都有其監

測區域那樣。本論文所使用之方法可減少參考點建立的數量，並能重覆使用已存在資料庫中的參考點影像及其特徵點資訊，也因此能讓EKF-SLAM中State vector及Covariance matrix不會一直持續的增大，降低EKF-SLAM最常見之複雜度持續增加的問題所造成的影響。

上述之實驗證實了影像處理及定位演算法的可行性，亦驗證了參考點轉換的策略無誤，並以最後情境模擬實驗證實本論文所設計的機器人定位方法能夠幫助機器人達成室內導航的功能。



第六章 結論與未來展望

本論文完成了使用全向式攝影機之機器人定位設計與實驗，基於 EKF SLAM 之演算法，以全向式攝影機對環境做觀測，在機器人移動的同時，能夠建立出環境地圖並定位出機器人的位置。

在環境的觀測方面，使用全向式攝影機為感測器，利用 SIFT 特徵點辨識演算法配合攝影機的特性，成功的擷取辨識出在影像平面上的環境特徵點，並使用一除錯法，實驗結果特徵點辨識的正確率平均為 90%。

基於攝影機對環境特徵點的觀測，及不同地點對特徵點的觀測視角，推算出特徵點相對於機器人本身的距離關係，並以此資訊做為定位系統的輸入。使用 SLAM 演算法解決機器人之定位及地圖建立的問題，以機器人之移動模型、攝影機之觀測模型，達成 EKF-based SLAM 演算法。結合影像處理與 SLAM 演算法，實現在機器人平台上，目前實驗的測試上，一次的定位系統演算法運算耗時 2 秒以內。

最後並以實際的實驗來驗證定位演算法：在來回行走共 30 公尺後之定位誤差平均為 0.11 公尺，而情境模擬實驗的結果證實機器人能依定位系統的幫助，從房間內走出到走廊，沿走廊移動近 50 公尺的距離後回到房間內，並且同時建立出走廊環境的特徵點地圖，達成機器人室內導航的功能。

在本論文中之定位演算法需耗時 2 秒左右，此運算時間在較為慢速的測試實驗中雖然可行，但若在實際的應用中對於環境可能的複雜性或是改變則不能有及時的反應，因此在影像處理及定位演算法的設計上可以再加以改善，提升整體定位系統的效能。

全向式攝影機常被用於多機器人系統上，而本論文發展出的基於全向攝影機之機器人定位方法亦可與之整合，讓攝影機的功能不僅能夠觀測機器人隊伍中的隊友，也讓機器人能對周遭整體的環境做估測，並得知所在之位置，增加多機器人隊伍的實用與功能。

参考文献

- [1] A. Sharma, D. Morales, G. Kantor and H. Choset, "Towards removing artificial landmarks for autonomous exploration in structured environments," Master's thesis, Carnegie Mellon University, 2005.
- [2] S. Thompson and A. Zelinsky, "Accurate local positioning using visual landmarks from a panoramic sensor," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, 2002, pp. 2656-2661.
- [3] H. Andreasson, A. Treptow and T. Duckett, "Localization for Mobile Robots using Panoramic Vision, Local Features and Particle Filter," in *Proceedings of IEEE International Conference on Robotics and Automation*, Apr. 2005, pp. 3348-3353.
- [4] J.H. Kim and M.J. Chung, "SLAM with Omni-directional Stereo Vision Sensor," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, October 2003, pp. 442-447
- [5] J.M.M. Montiel, J. Civera and A. J. Davison, "Unified Inverse Depth Parametrization for Monocular SLAM," in *Robotics Science and Systems Conference*, Philadelphia, 2006.
- [6] F. Bertolli, P. Jensfelt and H. I. Chistensen, "SLAM using Visual Scan-Matching with Distinguishable 3D Points," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, October 2006, pp. 4042-4047.
- [7] J. Kim, K.J. Yoon, J.S. Kim and I. Kweon, "Visual SLAM by Single-Camera Catadioptric Stereo," in *Proceedings of SICE-ICASE International Joint Conference*, Oct. 2006, pp. 2005-2009.
- [8] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [9] N. Winters, J. Gaspar, G. Lacey and J. Santos-Victor, "Omni-directional vision for robot navigation," in *Proceedings of IEEE Workshop on Omni-directional Vision*, 2000, pp. 21-28.
- [10] S. Panzieri, F. Pascucci, R. Setola and G. Ulivi, "A Low Cost Vision Based Localization System for Mobile Robots," *9th Mediterranean Conference on Control and Automation*, Jun. 2001.
- [11] J. Shi and C. Tomasi, "Good features to track." in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.
- [12] H. Durrant-Whyte and T. Bailey, "TUTORIAL : Simultaneous Localization and Mapping (SLAM): Part II", *IEEE Robotics & Automation Magazine*, Sep. 2006, pp. 108-117.

- [13] H. Durrant-Whyte and T. Bailey, "TUTORIAL : Simultaneous Localization and Mapping: Part I", *IEEE Robotics & Automation Magazine*, Jun. 2006, pp. 99-108.
- [14] S. Thrun, D. Fox and W. Burgard, *Probabilistic Robotics*, Massachusetts Institute of Technology, 2005.
- [15] S. Thrun, D. Fox and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots," in *Machine Learning and Autonomous Robots*, vol.31, pp. 29-53, 1998.
- [16] K. S. Chong and L. Kleeman, "Feature-based mapping in real, large scale environments using an ultrasonic array," in *International Journal of Robotics Research*, vol. 18, no. 2, pp. 3-19, Jan. 1999.
- [17] Z. Huaibin and J. K. Aggarwal, "3D Reconstruction of an Urban Scene from Synthetic Fish-eye Images," in *Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 219-223, 2000.
- [18] F. Pedemini, A. Sarti and S. Tuburo, "Multi-Camera Systems," *IEEE Signal Processing Magazine*, vol. 16, pp. 55-65, 1999.
- [19] F. Bruckstein and T. Richardson, "Omniview Cameras with Curved Surface Mirrors," in *Proceedings of IEEE Workshop on Omnidirectional Vision*, pp. 79-84, 2000.
- [20] G. Jang, S. Kim, J. Kim and I. Kweon, "Metric localization using a single artificial landmark for indoor mobile robots," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 2857-2862, Aug. 2005.
- [21] S. Thrun, "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," in *Journal of Artificial Intelligence*, vol. 99, pp. 21-71, Feb. 1998.
- [22] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Mar. 2002.
- [23] C.Y. Wu, "A Novel Robotic vSLAM System to Realize Exploration in Larger Environment," M.S. thesis, Department of Graduate Institute of Electrical Engineering National Taiwan University, Jun. 2007.
- [24] N. Winters and D. Hons, "A Holistic Approach to Mobile Robot Navigation using Omnidirectional Vision," University of Dublin, Trinity College, Oct. 2001.
- [25] L. Wang, Q. Zhu and Z. Liu, "Location Research of Mobile Robot with an Omni-directional Camera," in *Proceedings of International Conference on Intelligent Mechatronics and Automation*, 2004, pp. 662-666.
- [26] J. V. Miro, G. Dissanayake and W. Xhou, "Vision-based SLAM using natural features in indoor environments," in *Proceedings of International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2005, pp. 151-156.

- [27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Ed. Prentice Hall, 2002.
- [28] A. Diosi and L. Kleeman, "Laser Scan Matching in Polar Coordinates with Application to SLAM," in *Proceedings of International Conference on Intelligent Robots and Systems*, 2005, pp. 3317-3322.

