

國立交通大學

電控工程研究所

碩士論文

基於擴展卡門濾波同時定位與地圖建立之地圖

接合研究

Study of Map Joining in EKF-SLAM



研究生：劉建宏

指導教授：宋開泰 博士

中華民國一百零一年七月

基於擴展卡門濾波同時定位與地圖建立之地圖
接合研究

Study of Map Joining in EKF-SLAM

研究生：劉建宏

Student: Chien-Hung Liu

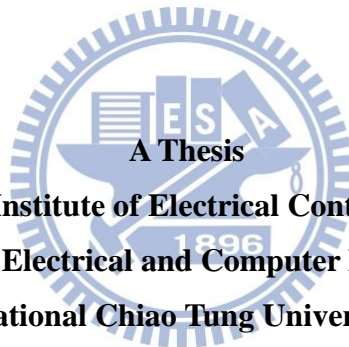
指導教授：宋開泰 博士

Advisor: Dr. Kai-Tai Song

國立交通大學

電控工程研究所

碩士論文



A Thesis

Submitted to Institute of Electrical Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical Control Engineering

July, 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年七月

基於擴展卡門濾波同時定位與地圖建立之地圖 接合研究

學生:劉建宏

指導教授:宋開泰 博士

國立交通大學電控工程研究所

摘要

本論文提出一使用 Kinect 深度攝影機之機器人定位方法。以 Kinect 為感測器取得環境資訊，結合 Extended Kalman Filter(EKF) 之同時定位與環境地圖建立 (Simultaneous Localization and Mapping, SLAM) 演算法，並以地圖接合之方式降低定位系統運算複雜度。Kinect 深度影像攝影機同時提供彩色影像與距離資訊，本研究以擷取 SURF 特徵點對應感測器所取得之深度關係，快速且精確的取得特徵點之環境資訊，接著以 EKF 修正機器人狀態與特徵點三維座標。為了避免 EKF 隨環境增長而使矩陣運算複雜度快速增加，本論文提出以區域路徑範圍判斷之作法將環境分為數個子區域，機器人僅需要使用區域內之特徵點訊息，而無須使用整個環境資訊做定位系統演算，如此提高定位系統於機器人應用之即時運算之性能，最後鄰近的子區域以地圖融合演算法修正其間之差異，以維持整個環境地圖之完整性。實驗結果顯示機器人運行於一 16mX7m 之室內環境行走約 83 公尺，當機器人回到原點附近時實際位置與估測之間的二維座標平均誤差小於 0.1 公尺。實驗結果證實機器人能以 EKF 之定位系統藉地圖接合之作法，達成機器人室內導航之功能。

Study of Map Joining in EKF-SLAM

Student: Chien-Hung Liu

Advisor: Dr. Kai-Tai Song

Institute of Electrical Control Engineering
National Chiao Tung University

ABSTRACT

This study investigates simultaneous localization and mapping(SLAM) of a mobile robot using a Kinect depth camera. Depth and image information from Kinect are utilized to realize SLAM algorithms based on extended Kalman filter(EKF). In this thesis, visual landmarks are extracted by SURF algorithm, then three dimensional location of feature points are calculated from Kinect depth image data. A map joining method is proposed to reduce computational complexity of EKF-SLAM, and to correct the deviations of adjacent local maps. A global map of the environment is constructed by the map joining procedure. Navigation experiments show that the accuracy of robot localization for a travel about 83m path is within 0.1m. It is verified that the developed algorithm of simultaneous localization and mapping with map joining can allow robot to navigate in an indoor environment ◦

誌謝

僅向我的指導教授宋開泰博士至上感謝之意，感謝他三年來在專業研究上之指導以及於學習態度與方法上之教誨，並以他豐富的學識與經驗，配合理論之應用使得本論文得以順利完成。

感謝學長孟儒、嘉豪、格豪、信毅、允智、學姊巧敏在理論以及實驗過程中的指導，以及感謝同學上峻、仕晟和碩成在學習過程中互相勉勵與成長，同時感謝學弟家昌、章宏、昭宇、Carlos、俊儒、京叡、明翰、瑋哲、政輝、佑霖、奕夫以及學妹依穎在生活與課業帶來的樂趣。

最後，特別感謝我的父母以及所有家人，由於他們時刻的關心與鼓勵，在生活與精神上的支持與照料使我能順利完成論文，在此以這篇論文獻給我摯愛的家人們。



目錄

摘要.....	i
ABSTRACT.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	viii
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 相關研究回顧.....	1
1.3 問題描述.....	5
1.4 系統架構.....	7
1.5 章節說明.....	8
第二章 特徵點偵測.....	9
2.1 Kinect 感測器.....	9
2.2 SURF 特徵點擷取.....	11
2.2.1 積分影像.....	11
2.2.2 Fast-Hessian Detector.....	12
2.2.3 描述元.....	14
2.2.3.1 關鍵點的主方向.....	15
2.2.3.2 特徵點描述元.....	15
2.3 特徵點比對.....	16
2.4 Kinect 距離量測精度測試與校正.....	17
2.5 特徵點於環境的三維座標資訊.....	20
2.6 特徵點資料庫的建立.....	25
第三章 機器人定位系統.....	27
3.1 機器人運動模型.....	28
3.2 特徵點觀測模型.....	31
3.3 EKF 演算法.....	34
3.4 討論.....	39
第四章 地圖接合.....	40
4.1 區域地圖.....	40
4.2 全域地圖.....	41
4.3 重合事件偵測.....	43
4.4 地圖選擇.....	45
4.5 地圖融合演算法.....	47
4.6 討論.....	57

第五章	實驗結果.....	58
5.1	機器人硬體介紹	58
5.2	機器人運動控制	60
5.3	EKF-SLAM 定位系統實驗-三公尺直線運動.....	60
5.4	EKF-SLAM 定位系統實驗-機器人直線來回實驗	62
5.4.1	無使用控制器之機器人直線來回實驗	64
5.4.2	使用控制器的機器人直線來回運動	68
5.5	EKF-SLAM 定位以及地圖接合 機器人方型軌跡運動	71
5.6	EKF-SLAM 機器人室內長距離移動實驗	80
5.7	討論.....	85
第六章	結論與未來展望	87
6.1	結論.....	87
6.2	未來展望.....	87
參考文獻	89



圖目錄

圖 1.1 影像分界設計之示意圖[3].....	4
圖 1.2 D&C SLAM 地圖融合設計架構[18].....	4
圖 1.3 以影像資訊找尋重合之地圖資料庫特徵[20].....	5
圖 1.4 利用攝影機姿態之行進軌跡做匹配[20].....	6
圖 1.5 資料庫影像與當前影像字串化訊息統計[21].....	6
圖 1.6 系統架構圖.....	8
圖 2.1 Kinect 感測器.....	10
圖 2.2 Kinect 彩色影像與深度影像.....	11
圖 2.3 影像關注區域之示意圖.....	12
圖 2.4 Box Filter 近似 Hessian matrix 之影像遮罩[17].....	13
圖 2.5 SURF 遮罩與尺度關係示意[24].....	14
圖 2.6 影像關鍵點計算之示意圖[16].....	15
圖 2.7 哈爾小波響應之影像遮罩[24].....	16
圖 2.8 描述元計算之示意圖.....	16
圖 2.9 哈爾小波響應轉換結果統計圖[17].....	17
圖 2.10 SURF 影像特徵點擷取之範例.....	17
圖 2.11 校正板與 Kinect 相距 180cm 的取點結果.....	18
圖 2.12 Kinect 量測 Y 方向實際位置與平均誤差之關係圖.....	19
圖 2.13 Y 方向 60cm-720cm 經線性曲線轉換後與誤差之關係圖.....	20
圖 2.14 特徵點 X 方向量測位置與平均誤差關係.....	21
圖 2.15 特徵點 Y 方向量測位置與平均誤差關係.....	22
圖 2.16 特徵點 Z 方向量測位置與平均誤差關係.....	22
圖 2.17 影像特徵點三維座標之計算流程圖.....	23
圖 2.18 座標系特徵點資訊修正之示意圖.....	24
圖 2.19 資料庫統合影像特徵點訊息之步驟.....	26
圖 3.1 EKF SLAM 演算示意圖.....	28
圖 3.2 機器人運動模型表示圖.....	29
圖 3.3 機器人中心與特徵點在二維座標直線距離 r 和垂直夾角 ϕ	32
圖 3.4 機器人量測模型圖示.....	32
圖 4.1 區域地圖之示意圖.....	42
圖 4.2 區域地圖融合策略.....	43
圖 4.3 重合區域偵測示意圖.....	44
圖 4.4 區域路徑重疊範圍之示意圖.....	46
圖 4.5 地圖建構與選擇之流程圖.....	46
圖 4.6 區域一區域二與共同特徵點的關係圖.....	50

圖 5.1 機器人實體圖.....	59
圖 5.2 機器人硬體架構圖.....	59
圖 5.3 路徑控制器架構圖.....	60
圖 5.4 路徑控制器之架構圖.....	61
圖 5.5 直線移動實驗結果與實際環境配置對照圖.....	62
圖 5.6 三公尺定點實驗之誤差直條圖.....	63
圖 5.7 機器人來回實驗之過程變化圖.....	66
圖 5.8 無控制器直線來回實驗結果與實際環境配置對照圖.....	66
圖 5.9 無控制器直線來回實驗之誤差直條圖.....	67
圖 5.10 無路徑控制器直線來回實驗之運動軌跡.....	68
圖 5.11 具控制器直線來回實驗結果與環境配置對照圖.....	69
圖 5.12 具路徑控制器直線來回實驗之誤差直條圖.....	70
圖 5.13 具路徑控制器直線來回實驗之運動軌跡.....	71
圖 5.14 機器人角度響應圖.....	71
圖 5.15 實驗室方形路徑實驗結果與實際環境對照圖.....	72
圖 5.16 實驗室方形路徑實驗之區域地圖.....	73
圖 5.17 方形路徑實驗之運算時間.....	74
圖 5.18 方形路徑實驗之地圖建構運算時間(不包括地圖融合).....	75
圖 5.19 方形路徑實驗地圖建構之運算時間(包括地圖融合).....	76
圖 5.20 方形路徑實驗之重複路徑運算時間.....	77
圖 5.21 實驗室方形路徑實驗之誤差直條圖.....	78
圖 5.22 方形路徑實驗無地圖接合演算之地圖建構結果.....	79
圖 5.23 方形路徑實驗無地圖接合演算之地圖建構運算時間.....	80
圖 5.24 長距離實驗環境配置圖.....	81
圖 5.25 室內長距離實驗區間 1 建立之區域地圖.....	82
圖 5.26 室內長距離實驗區域 2 建立之區域地圖.....	83
圖 5.27 長距離運行機器人建立之環境地圖.....	84
圖 5.28 長距離機器人運行實驗結果與實際環境配置之對照圖.....	84
圖 5.29 室內長距離實驗座標誤差之直條圖.....	85
圖 5.30 標記影像特徵點對實驗結果影響之不穩定部分.....	86

表目錄

表 2.1 Kinect 規格[22]	10
表 2.2 Kinect 量測 Y 方向實際位置與平均誤差之數值	19
表 2.3 X 方向實際距離與平均誤差之數值	21
表 2.4 Y 方向實際距離與平均誤差之數值	21
表 2.5 Z 方向實際距離與平均誤差之數值	22
表 2.6 直線移動實驗之機器人位置估測結果	26
表 3.1 EKF SLAM 演算流程	38
表 4.1 地圖融合演算法之流程	56
表 5.1 運動控制計算流程	61
表 5.2 直線移動實驗之機器人位置估測結果	63
表 5.3 無控制器直線來回實驗之機器人位置估測結果	67
表 5.4 具路徑控制器直線來回實驗之機器人狀態估測結果	70
表 5.5 建構地圖階段區域最大運算時間	76
表 5.6 建構地圖階段地圖接合運算時間	76
表 5.7 實驗室方形路徑實驗之機器人位置估測結果	78
表 5.8 長距離實驗機器人實際與估測之狀態資訊	85



第一章 緒論

1.1 研究動機

近幾年來機器人的發展逐漸從工業用途拓展到公共服務及居家照護等服務型機器人的應用。服務型機器人的種類相當多元，例如它可以做為攙扶行動不便者的照料者，或是在發生緊急狀況時通知救助人員的守護者，機器人將逐步地融入人們的各種生活環境裡。

當機器人步入人類的生活，就必須與人們協調互動，而其應用情境都必然有其對應的任務區域，例如機器人要拿飲料，可能要到廚房開冰箱，當有人跌倒摔傷等緊急事件發生時，它可能要趕到客廳或浴室了解狀況；但無論機器人有多麼強大複雜的功能，其基本的面向是要如何正確地到達執行任務的地點，這種自主導航的功能便是本論文所要探討的議題。

同時定位與建立地圖(Simultaneous localization and mapping, SLAM)，是機器人定位與導航很重要的一項技術。其主要的方法是藉由感測裝置取得環境中的參考資訊來修正機器人的定位狀態。常被使用的感測裝置有雷射掃描儀、超音波等的距離感測裝置，或是單眼攝影機以及立體攝影機等視覺感測器。距離感測裝置能夠精確的取得環境的深度資訊，然而其所取得的特徵彼此獨特性較為不足，而不容易分辨環境的差異。而視覺感測裝置，則沒有提供距離的資訊，必須經過兩張影像或是利用兩個以上的攝影機，才能間接計算特徵點在環境中的三維座標。

Kinect感測器是一種結合影像與深度感測器的裝置，在SLAM的運用上，以影像作特徵點擷取，並藉由深度的資訊與影像的結合，而能快速的將影像上的特徵資訊，轉換為環境中的三維座標。本論文即是研究使用Kinect影像深度感測裝置，藉由影像的豐富資訊以及深度訊息，幫助機器人在室內環境完成自我定位的功能。

1.2 相關研究回顧

近十年來，基於影像之同時定位與建立地圖(SLAM)，是移動式機器人技術

發展上一項重要的領域，其基本概念是透過環境中的參考資訊對機器人狀態進行修正[1][2]，且整個系統是由多個不同的部分所組成，包括特徵點的擷取、特徵點的比對、定位系統的演算法、地圖的建立以及地圖資訊的應用等。在SLAM的研究上所使用之感測裝置主要以視覺感測裝置以及距離感測裝置為主。 [3]以單眼攝影機在機器人上實現EKF SLAM，[4]則估測手持攝影機之姿態並建立環境地圖，為了使機器人更詳盡的擷取環境資訊，[5][6]以全向鏡做感測裝置實現於雙輪以及全向式移動平台，而[7][8]則以不同角度之四個單眼攝影機取得環境影像，並且實現自我外部攝影機參數設定之功能，[9]則藉由Rao-Blackwellized Particle Filter (RBPF)以立體視覺感測器實現機器人定位，並以區域和整體環境中的權重修正地圖建立後的結果。在使用距離感測器上則有雷射掃描儀[10]和超音波[11]等之使用；距離感測裝置能取得準確的環境深度資訊，但距離資訊所得到的特徵通常具有獨特性不足之問題。視覺感測裝置帶有豐富訊息之影像，然通常需要數次的影像或兩個以上的攝影機才能計算特徵點在環境中的空間資訊。近年來結合影像和距離感測裝置Kinect受到許多研究的重視，[12]以Kinect作為感測裝置架設於四輪移動平台，並以不同色環作為環境地標(Landmark)實現EKF和PF (Particle filter)定位系統演算法。

在以視覺感測器取得影像作為環境的感測裝置的研究上，關於特徵點的擷取的作法，[13][14]以環境中對稱物體的性質做地標(Landmark)，提出 Multi-scale Symmetry Transform(MUST)的做法，[15]提出具快速運算特性的角點偵測方法，而近年來相當被重視的則為Lowe[16]所提出的Scale invariant feature transform (SIFT)演算法，其將影像以Difference of Gaussian(DoG)擷取影像中穩健的關鍵點，計算不同尺度下的變化，並以具方向性的特徵點描述，使得提取出來的影像特徵點具備有抗尺度變化、抗旋轉、以及抵抗在一定的改變量下的視角和光源變化。同樣藉由影像取得環境資訊，Speed up robust feature(SURF)演算法以積分影像的使用簡化影像摺積運算，並透過Hessian matrix的應用以簡化的影像遮罩找尋影像特徵，除了有著穩健且具獨特性的影像特徵外，同時具備較快的運算速度[17]。

定位與導航是移動式機器人基本技術之一，機器人能得知自身於環境中之位置才能自由移動。然為了使定位系統持續修正機器人狀態並降低對整體機器人效能的影響，定位與導航的即時運算性能是機器人應用之關鍵要素。定位與導航是由數個不同的部分所組成，提高其即時運算的性能可透過降低各部分運算成本而達成，諸如特徵點擷取的效率，特徵點比對的做法和修改定位系統的演算法等，其中定位系統計算所耗費的運算成本通常是最主要的部分，而影響定位系統所耗費的運算成本上主要是使用的特徵點數量。以Particle Filter(PF)的演算法而言，隨著散佈的粒子數量越高，其估測的機器人狀態也就越準確，但所需花費的運算時間也就越久。而Kalman Filter(KF)的演算法是一種以疊代的方式估測機器人狀態，系統狀態向量的大小影響運算矩陣的維度，同樣的受到所使用的特徵點數量影響，於是為了解決定位與導航在運算成本上的問題，最直觀的作法便是降低所取樣的特徵點數量，在前段中所述的特徵點擷取方法MUST [13] [14]，其目的除了取得影像中更加具代表性且更穩定的特徵外，同時也能降低特徵點擷取的數量；除了直接降低擷取的特徵點數量外。另一個做法便是對做為定位系統的特徵點進行額外的處理，[3]以單眼攝影機為感測裝置，其將機器人行進間所取得之影像分為數個群組，每一個群組以開頭的影像作為參考影像，接續的影像特徵與參考影像的特徵進行比對，取得的相同特徵點作為EKF的輸入，修正系統狀態向量，藉此達成同時定位與建立地圖，圖1.1說明此方法的設計方式。圖中每個區間由四個影像所組成，其中第一個影像為參考影像，以固定數量的影像資料做分界降低定位系統矩陣複雜度的增長。

[18]提出一種處理特徵點地圖資訊的做法，此做法用以降低定位系統的運算成本，在該文章中其所使用的資料庫為特徵點地圖，在定位與地圖建立使用EKF演算法，降低建構地圖時的運算成本是以限制單一區域內資料總數的方式將地圖分為數個部分，藉以降低定位系統所使用的系統狀態向量維度，而最終分離的區域地圖則以二分法的地圖融合設計將區域地圖資訊整合。圖1.2為該文所設計之

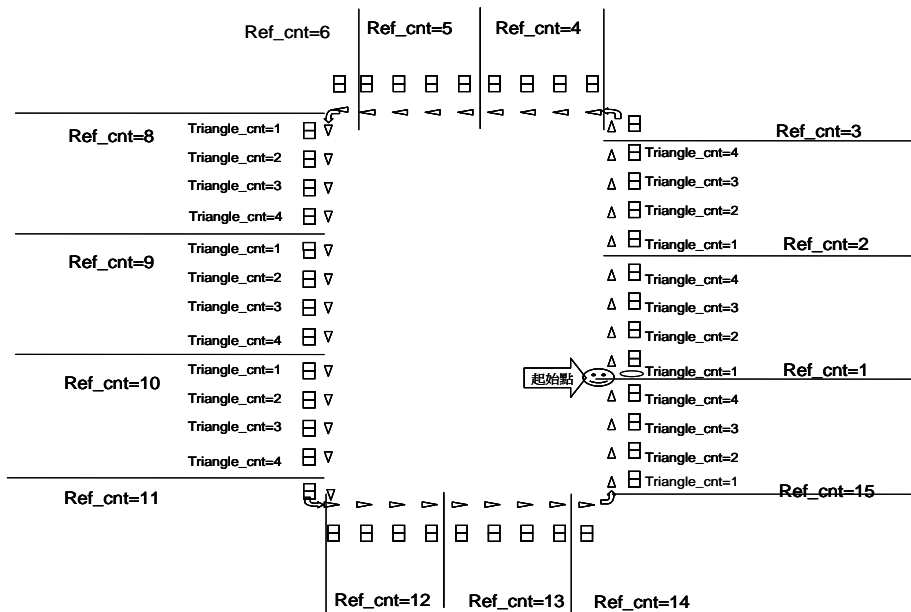


圖 1.1 影像分界設計之示意圖[3]

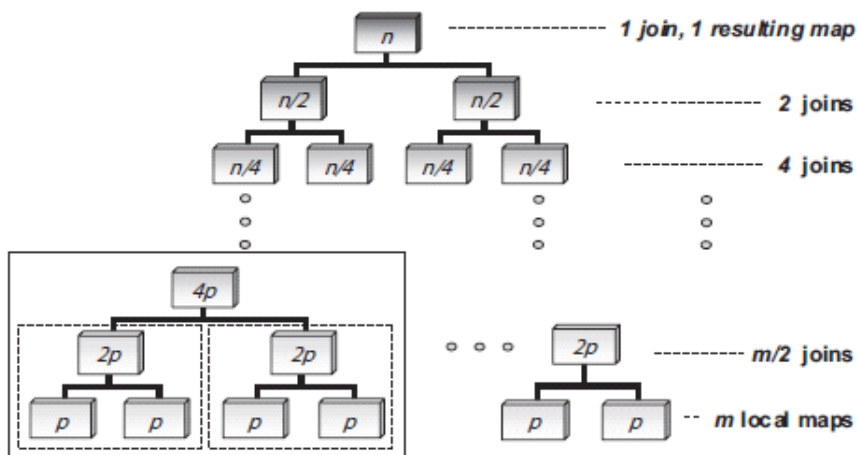


圖 1.2 D&C SLAM地圖融合設計架構[18]

二分式地圖融合設計，假設機器人完成運行共生成了 m 個區域地圖，過程中 m 個區域地圖融合成總數為 $m/2$ 個資訊，再將 $m/2$ 個資訊結合為 $m/4$ 個，以此類推直到所有資訊被融合成一個完整的世界地圖。

在SLAM中，機器人應用已建立之地圖的一個典型例子是對重合路徑發生 (Loop closure) 之辨認。其實質的意義是使機器人在建立地圖後能使用已建立之地圖資訊，避免系統重複建圖並使機器人能使用已建立之地圖資料庫。且當機器人再次行進於重複路徑時，透過使用已建立的地圖資料庫加強定位結果的正確性。其中最典型的情境是當機器人行進的路徑為方型封閉路線，方形封閉路徑使機器

人最終將從起點位置重複繞行。[20]將影像特徵點整合為特徵點地圖，重合事件 (Loop closing) 的偵測是以運行時的影像特徵點比對地圖資料庫訊息(圖1.3)以及攝影機移動姿態的關聯性(圖1.4)來達成。其影像資訊與資料庫比對的方式是利用隨機搜尋樹演算法(Randomised tree)先將資料庫特徵點訊息分類，影像特徵資訊同樣以隨機搜尋樹演算法計算歸屬，藉以加速影像特徵比對之效率。[21]所使用的資料類型為單張影像的特徵點訊息，每個影像特徵點資訊被轉化為字串化訊息 (bag of word)，以圖1.5為例，在 t 時刻所得到的影像 I_t 其透過與資料庫的字串化訊息比對而得知當前影像事件 S_t ，事件 S_t 的結果代表當前影像是否為重合影像，資料庫中的字串化訊息代表不同影像的資料，透過統計的方式計算與資料庫各個影像的相似程度，也就是當兩時段的影像字串訊息有越多的相同部分其統計數值也就越高。以[20]建構地圖的方式是將影像資訊整合為特徵點地圖資料庫，與以影像作為資料庫的做法[21]比較，整合特徵點地圖可以避免相同的特徵點重複存放，減輕資料庫所需容量降低比對時所需的運算，在重合事件的偵測上，以影像作為資料庫的做法較易準確的偵測，但所需比對的影像特徵點數量較大。

1.3 問題描述

過去對於機器人在同時定位以及建立地圖上的研究，主要以距離量測裝置或是視覺感測裝置提供環境資訊，距離量測裝置能夠精確的得到二維平面上機器人與物體的相對距離關係，但以二維平面所取得的距離資訊，易因地標(Landmark)

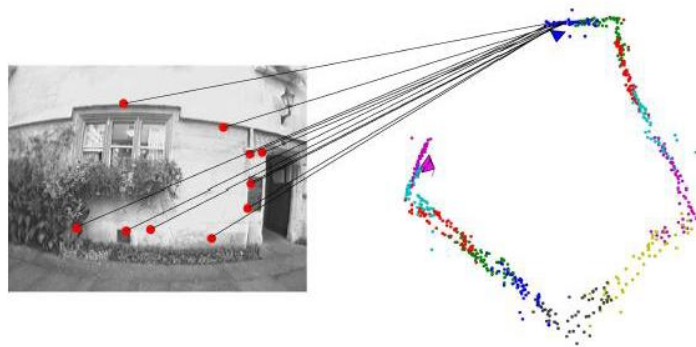


圖 1.3以影像資訊找尋重合之地圖資料庫特徵[20]

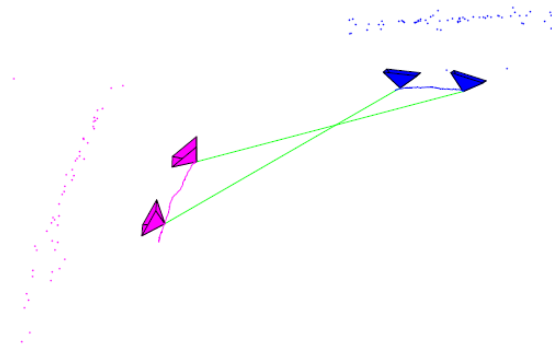


圖 1.4 利用攝影機姿態之行進軌跡做匹配[20]

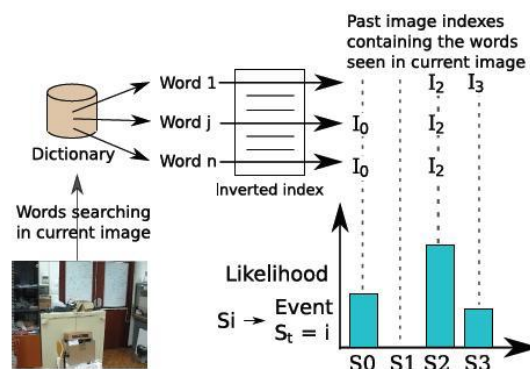


圖 1.5 資料庫影像與當前影像字串化訊息統計[21]

獨特性不足而難以區分不同的環境；而以視覺做為感測裝置，則因無法直接取得特徵的三維座標關係，通常必須透過兩張以上的連續影像，或是多個視覺感測裝置計算特徵的三維關係，而影響機器人的即時運算，因此，環境資料的選擇上希望以影像作為基礎，同時降低取得距離資訊的所需運算。

機器人在同時定位與建立地圖之應用，定位系統需要不斷的修正機器人狀態以降低誤差的產生，然而隨著機器人運行的範圍越大，所建立之環境亦隨之擴大，對定位系統而言，越多的環境訊息其所需運算時間便會加長，修正的間隔時間越長，機器人狀態修正的次數也就越少，若以降低環境資料量的方式來減輕定位系統之運算成本，又可能會造成整體環境資訊變的不完整。在[3]中以數個影像分組的方式進行機器人同時定位與地圖建立，該做法限制影像張數降低所使用之特徵點數量而減輕運算成本，然影像群組間彼此獨立而沒有任何修正，可能使得區域與區域間存在差異，而使得誤差逐漸累積，且單一攝影機取樣的方式，受限於

特徵點尺度與特徵點訊息計算的方式，使得影像群組內相同特徵用於定位系統的次數受到限制，對於即時修正機器人定位上較為不利(單一攝影機於特徵點三維座標的計算需要兩張以上的影像，又以群組的第一張影像作為參考影像，受特徵點尺度的變化，能使用的特徵點數量會越來越少)。而[18]環境中的特徵點訊息分為數個區域，以二分法的方式在特定的時機將地圖資訊融合，其將整體環境分為數個部分的作法同樣能有效降低運算成本的消耗，然其區域地圖必須建立到特定數量才將地圖資訊作結合，使得部分區段間的資訊要等到機器人到達特定步驟次數下才能夠做關聯性的計算(其以二分法作地圖資訊融合的選擇，一個完整的地圖資訊融合發生在區域地圖建構數量在 $2、4、8、16...2^j$ 等二的次方數量)，過程中部分機器人狀態無法即時修正，而將所有特徵點地圖加入地圖融合演算，在該次步驟仍需耗費相當的運算時間。因此，將SLAM的功能實現於機器人上，提高即時運算能力又必須保持整個環境資訊的完整性，亦是我們所探討之問題。

1.4 系統架構

本論文之系統架構如圖1.6所示，由Kinect所得到的彩色影像以SURF演算法擷取影像中的特徵點，這些特徵與對應的區域地圖特徵進行比對，接著以Kinect取得之深度資訊計算特徵點三維座標，這些帶有距離且與對應的區域地圖特徵相同的特徵點資訊做為EKF定位演算法之輸入(在架構圖中，t時刻與地圖特徵相同的影像特徵點數量假設為 n_{t_HD} 個，故作為EKF定位系統的輸入如架構圖中所示的 $[m_{1,x} m_{1,y} m_{1,z} \dots m_{n_{t_HD},x} m_{n_{t_HD},y} m_{n_{t_HD},z}]$)，結合里程計(Odometry)計算之機器人座標與朝向角，修正估測之機器人狀態向量並更新區域地圖特徵點三維座標資訊，接著，由當前的機器人狀態資訊與區域地圖的路徑範圍，判斷機器人是否完成地圖建構。當區域地圖完成建構，當前區域地圖以地圖融合演算法將相鄰的區域地圖資訊結合，修正地圖特徵點的三維座標並且再一次修正機器人姿態，而當前區域地圖結束的位置會決定新區域的路徑範圍，這個新區域地圖的路徑範圍以重合事件偵測的方式判斷是否與其他已建立的區域重合，藉由判定後的結果

得知與新區域地圖相鄰的區域。這個結果影響新區域地圖完成建構後地圖融合時所使用的地圖資訊。最終，以地圖選擇決定當前所使用的區域地圖，並以修正的機器人狀態對機器人做路徑控制，整個SLAM系統便是在此系統架構中不斷運行，達成機器人同時定位與建立地圖以及機器人路徑導航之功能。

1.5 章節說明

本論文總共分為6章，第一章介紹研究動機及說明機器人定位系統架構，第二章說明本論文所使用之特徵點擷取的做法以及特徵點環境資訊的計算，第三章為本論文所使用的定位系統演算法，第四章為地圖接合的演算方式，第五章則透過實驗驗證定位系統在機器人上運行的實際結果，第六章為結論及未來展望。

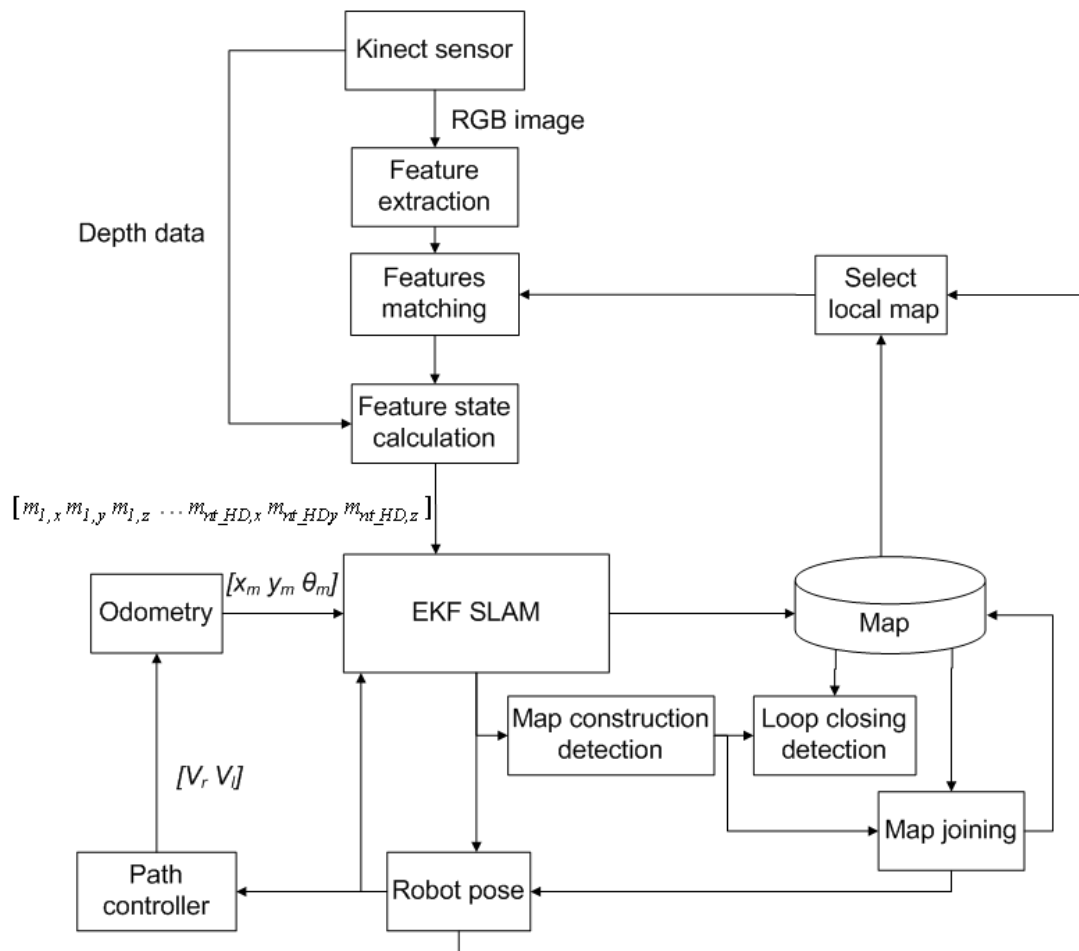


圖 1.6系統架構圖

第二章 特徵點偵測

現實世界中當我們回憶一個曾經到過的場所，會藉由該場所中顯著的建築或景物來敘述，像是台北 101 大樓。反過來說，當我們講起這些建築或奇景時，便能夠知道對應的地點。SLAM 以感測裝置取得環境中的特徵，這個特徵就像是環境中的地標(landmark)，提供機器人辨別環境的依據，為了有效的分辨不同的場所或地點，截取的特徵點必須具備獨特性；但是對於機器人而言，環境並非一成不變的畫面，隨機器人應用情境的不同機器人會在不同的區域行進，即使機器人回到相同的環境，取得環境資訊時的機器人姿態或是環境狀態(如光源強度，地面傾角等)亦不可能完全相同，然而在這些變化下要能夠成功的辨識相同的特徵，對定位系統而言才能有效應用，因此特徵在能夠抵抗各種環境變因的穩定性亦是必須考量的重要因素。最終，這些特徵將作為定位系統計算機器人姿態的依據，特徵點在環境中的三維座標必須由感測裝置取得或計算。滿足這些條件的環境特徵點訊息，是 SLAM 中最初且重要的一環。

本論文透過視覺感測裝置取得影像特徵點做為代表環境之資訊，特徵點擷取的方法是使用 SURF 演算法。SURF 演算法具備有抗旋轉，抗尺度，抗角度以及抗光源變化等的特性[17]，且其即時運算的效能適合於機器人應用，結合 Kinect 深度感測裝置能快速並且精確的計算特徵點於環境之三維座標資訊。

2.1 Kinect 感測器

圖 2.1 為 Microsoft 之 Kinect 彩色影像及深度感測裝置，其能夠提供 640X480 32-bit 每秒 30 個 frame 的彩色影像，以及 320X240 16-bit 每秒 30 個 frame 的深度資訊，其規格如表 2.1 所示，本論文以此感測器應用於 SLAM 之開發，能快速有效的取得環境資訊。

Kinect 為具備多種感測器的裝置，其有 RGB 攝影機、紅外線深度感應器以及陣列式麥克風，且於基座的地方設有馬達可使感測器上下擺動，本論文主要使用其 RGB 攝影機所取得之彩色影像以及紅外線深度感測器所測量之距離資訊。



圖 2.1 Kinect 感測器

表 2.1 Kinect 規格[22]

感應器	<ul style="list-style-type: none"> • RGB 攝影機鏡頭 • 深度感應鏡頭 • 陣列式麥克風
視野角度	<ul style="list-style-type: none"> • 水平視野：57 度 • 垂直視野：43 度 • 實體傾斜範圍：± 27 度 • 深度感測範圍：1.2m – 3.5m
資料串流	<ul style="list-style-type: none"> • 深度感應器：320×240 16-bit 30 frames/sec • 彩色攝影機：640×480 32-bit 30 frames/sec • 聲音規格：16-bit 16 kHz

Kinect 的深度感測原理是以紅外線發出人眼所看不見的雷射光，經過鏡頭前的光柵均勻投射到量測空間中，接收器得到反射回來的散斑，晶片對其進行解碼後得到深度資訊[23]。所謂的散斑是一種當雷射照射到物體，物體的粗糙表面反射相干子波互相疊加，反射回接收端時形成具高度隨機性的圖樣，這種隨機強度分布圖樣可以用統計的方法來描述，經過解碼後便可以得到相對應的深度資訊，利用此種測量距離的方法為一種光編碼(Light coding)的技術。

Kinect 由 Microsoft 推出之時，並沒有提供使用者相對應的 API，PrimeSense 這家公司響應了 Kinect 這項產品，並且釋出 OpenNI(Open Natural Interaction) [25]。OpenNI 這個 API 的目的在於希望利用感測器在任何系統下建立一套“自然互動”

的標準，所謂“自然互動”的標準主要是指使用者身上不需要裝配任何感測裝置，因此像是「手勢」、「身體動作」等等，都是其主要定義的對象。OpenNI 是硬體端與應用端的中間架構，本研究主要藉 OpenNI 取得 Kinect 的影像及深度資訊，以及將深度資訊轉換為 Kinect 座標系下的三維座標等函式。

圖 2.2(a)是用 Kinect 取得實驗室一處的 RGB 彩色影像，而圖 2.2(b)則是對應的深度影像，深度影像是利用在影像平面不同位置的深度資訊轉化為對應的色階。其中顏色越淺表示目標與 Kinect 感測裝置的位置越近，相反的，顏色越深則表示目標與 Kinect 感測裝置的位置越遠，而部分顏色為純黑色的部分，則為超過 Kinect 深度感測裝置所能感測的範圍，或是量測的物體為自發光源，像是日光燈或是電腦螢幕。另外，某些物體後方像陰影的部分，則是由於 Kinect 感測器的硬體配置所造成的現象，因 Kinect 的紅外線發送端與接收端的間隔距離較長，接收與發送端之間的角度差造成物體後方部分受到遮蔽而無法得知深度資訊。

2.2 SURF 特徵點擷取

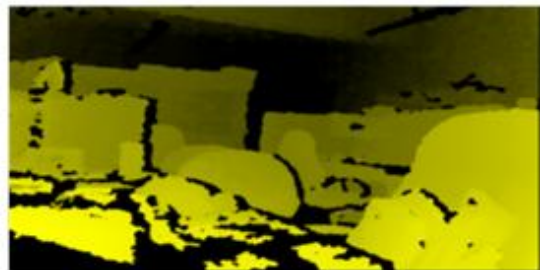
SURF 演算法以積分影像的使用簡化影像摺積運算，並透過 Hessian matrix 的應用以簡化的影像遮罩找尋影像特徵，除具備抗旋轉，抗尺度，抗角度以及抗光源變化的特性外，其較佳的即時運算能力使其更適合於機器人應用，本論文即以此種方法取得環境影像之資訊，以下各分節詳述其演算方法。

2.2.1 積分影像

積分影像對於 SURF 而言可以說是一種前置步驟，由於 SURF 演算法對影像



(a)RGB影像



(b)對應RGB影像之深度影像

圖 2.2 Kinect 彩色影像與深度影像

大量使用遮罩的演算，如計算區域間之關鍵點時以 Box Filter 做 Hessian matrix 的近似計算，這類的運算量大且重複性高，積分影像之運用則能有效降低此問題，其透過關注區域的四點運算，即可得到區域內的像素值總合。

積分影像 I_{Σ} 的形成，是將原影像 I 從原點的像素值累加至對應的影像座標，令一積分影像上座標 (u,v) ， $I(u,v)$ 為該影像座標的像素值，則該點積分影像的數值 $I_{\Sigma(u,v)}$ 如式 2.1:

$$I_{\Sigma(u,v)} = \sum_{i=0}^{i \leq u} \sum_{j=0}^{j \leq v} I(i,j) \quad (2.1)$$

當需要計算原影像上某區塊的像素值總和時，只要以積分影像計算關注之區域的邊界點即可，如圖 2.3 中藍色區域的邊界點為影像座標 A、B、C、D 四點，則對應原影像的藍色區域像素總值 K 可由積分影像 A、B、C、D 四點求得:

$$K = I_{\Sigma(A)} - I_{\Sigma(B)} - I_{\Sigma(C)} + I_{\Sigma(D)} \quad (2.2)$$

2.2.2 Fast-Hessian Detector

計算 SURF 的方式是基於 Hessian 矩陣的行列式，找出二次微分的極大與極小值，Hessian matrix 在 SURF 演算法中被當作偵測器，其在計算時間及精確度上具有很好的表現，定義於影像平面某點位置 $X=(u,v)$ ，尺度 σ 的 Hessian matrix $H(X,\sigma)$ 如式 2.3

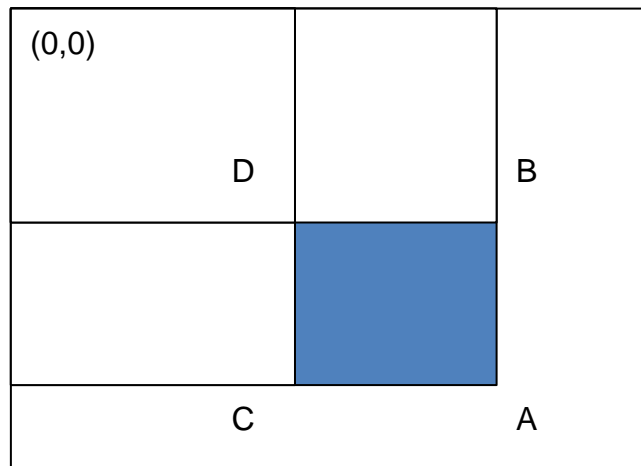
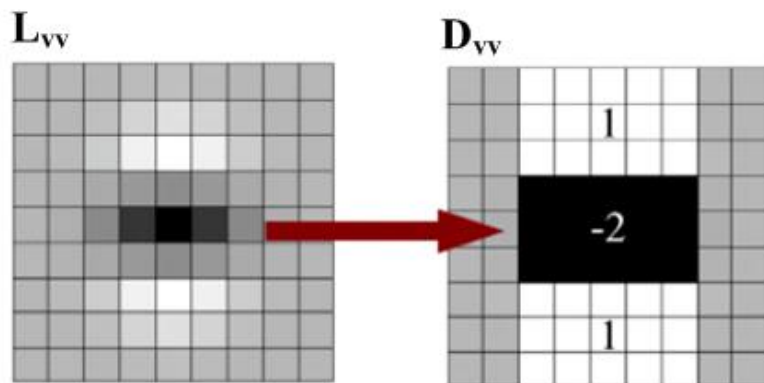


圖 2.3 影像關注區域之示意圖

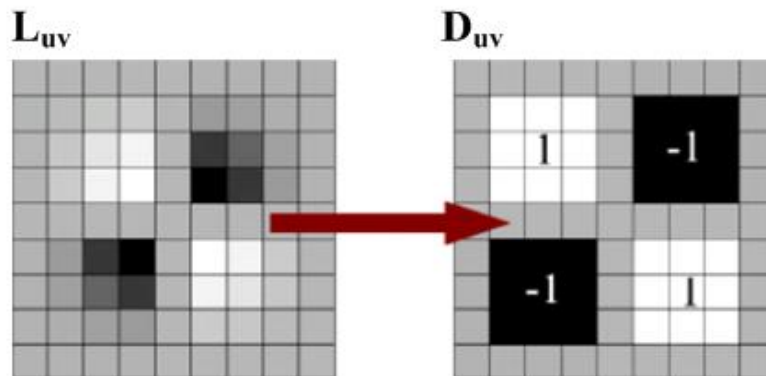
$$H(X, \sigma) = \begin{bmatrix} L_{uu}(X, \sigma) & L_{uv}(X, \sigma) \\ L_{vu}(X, \sigma) & L_{vv}(X, \sigma) \end{bmatrix} \quad (2.3)$$

其中 $L_{uu}(X, \sigma)$ 、 $L_{uv}(X, \sigma)$ 、 $L_{vu}(X, \sigma)$ 、 $L_{vv}(X, \sigma)$ 為 Laplacian of Gaussians (LoG)，是在影像座標 X 對不同方向以高斯二次偏微分之卷積運算的結果。

影像上對不同方向以高斯二次偏微分運算需要被離散化以及裁剪，如圖 2.4(a)左半邊是對影像在 V 方向做高斯二次偏微分，以 L_{vv} 表示，而圖 2.4(b)左半邊則是對 UV 方向做高斯二次偏微分，以 L_{uv} 表示，這樣離散和裁減的結果使得運算的結果與實際有所偏差，但以 Low 以 LoG 近似運算在特徵點取樣上的成功應用[16]為參考，[17]將 Hessian matrix 利用 Box Filter 做近似並配合積分影像加速運算。圖 2.4 中(a)右半邊是以 Box filter 近似 Hessian matrix 對 V 方向之二階偏微，圖 2.4 中(b)則為對 UV 方向之二階偏微，其中灰色的區域值為零，在不同方向以 Boxfilter 近似 Hessian matrix 的方式分別以 D_{vv} 和 D_{uv} 來表示。



(a)Box Filter近似Hessian matrix 之V方向二階偏導



(b)Box Filter近似Hessian matrix 之UV方向二階偏導

圖 2.4 Box Filter 近似 Hessian matrix 之影像遮罩[17]

不同濾波器的遮罩大小其對應 Hessian matrix 中不同的尺度(σ)，一個 9x9 的遮罩效果符合尺度 $\sigma = 1.2$ 時的高斯微分同時也是最低的尺度，為了使方形遮罩的權重保持運算效率上的簡化，以 Hessian 的行列式 $\frac{|L_{uv}(1.2)|_F |D_{uu}(9)|_F}{|L_{uu}(1.2)|_F |D_{uv}(9)|_F} = 0.912 \approx 0.9$ 進一步的去平衡權重，其中 $|x|_F$ 是 Frobenius norm，並且產生式(2.4)的結果。此外，遮罩響應對應不同的遮罩大小要被正規化，如此以確保在不同遮罩大小下具有相同的 Frobenius norm。因此，經過測試，遮罩大小為 27x27 的 Boxfilter 符合尺度 σ 為 3.6 時的情況，以這樣的方式檢測相同影像在不同尺度下的特徵，圖 2.5 表示以不同大小的 Boxfilter 可以用來檢測相同影像在不同尺度下的特徵。

$$\det(H_{approx}) = D_{xx} D_{yy} - (0.9 D_{xy})^2 \quad (2.4)$$

將影像上每個位置 $X(u,v)$ 經過遮罩計算後，每個影像平面上座標皆對應一個經 Hessian matrix 運算之近似值，將影像平面上每一個點與其周圍 8 個點以及相鄰取樣尺度空間的 18 個點比較(圖 2.6)，判斷其是否為極大或極小值，若是，則我們稱該點為本影像中的一個關鍵點。

2.2.3 描述元

特徵點在連續的影像中要能夠被找到，依賴的是對關鍵點的描述方式，SURF 先透過哈爾小波響應(Haar-wavelet responses)[17]找出代表關鍵點的主方向，再利用關鍵點周圍像素的哈爾小波響應統計為關鍵點的描述元。



圖 2.5 SURF 遮罩與尺度關係示意[24]

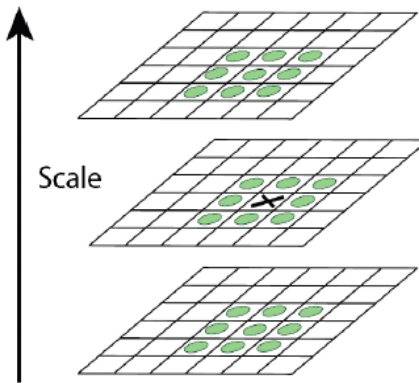


圖 2.6 影像關鍵點計算之示意圖[16]

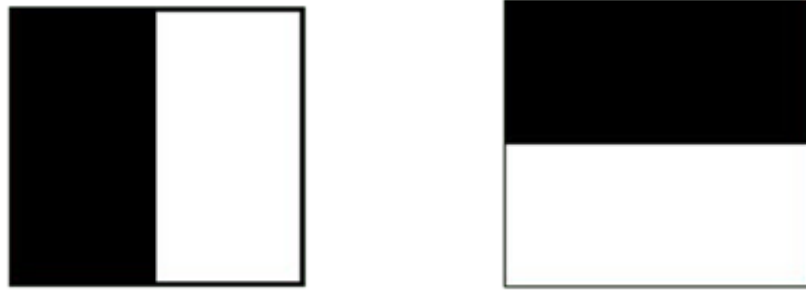
2.2.3.1 關鍵點的主方向

首先，SURF 以關鍵點為中心半徑為 $6s$ 的圓型區域計算在 u 以及 v 方向的哈爾小波響應，其於影像平面之應用是以圖 2.7 做遮罩運算，遮罩的大小則為 $4s$ ，其中 s 代表的是所要檢測的關鍵點所在的尺度大小，如此得到區域內每個點在 u 以及 v 方向的分量，接著再以大小為 $\pi/3$ 圓的區域做統計，每個區域統計得到一個代表這 $\pi/3$ 圓區域的向量長度以及方向，滑動 $\pi/3$ 圓的位置得到多組區域的向量，最終，找出一個向量長度最長者作為代表這關鍵點的描述方向，透過關鍵點主方向的計算，使得 SURF 特徵點具備抗旋轉之特性

2.2.3.2 特徵點描述元

具備描述元之關鍵點即為最終 SURF 演算法所得到之特徵點。SURF 以關鍵點為中心，以代表其關鍵點的主方向為基準，設立一個邊長大小為 20σ (尺度) 方框如圖 2.8，該方框內的區域再區分為 $16 (4*4)$ 個子區域，計算區域內的哈爾小波響應是以關鍵點的主方向為基準計算 U (水平) 以及 V (垂直) 方向之哈爾小波響應(圖 2.8)。其中 d_u 稱為水平方向的哈爾小波響應， d_v 稱為垂直方向的哈爾小波響應，每個子區域將其水平、垂直、水平的絕對值以及垂直的絕對值變化量個別累加的總值作為子區域的描述向量 P 。

$$P = (\sum d_u, \sum d_v, \sum |d_u|, \sum |d_v|) \quad (2.5)$$



(a)U方向哈爾小波響應之影像遮罩 (b)V方向哈爾小波響應之影像遮罩

圖 2.7 哈爾小波響應之影像遮罩[24]

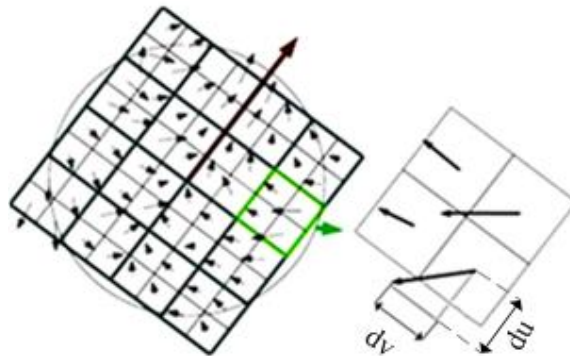


圖 2.8 描述元計算之示意圖

圖 2.9 顯示以不同影像做哈爾小波響應後各分量得到的統計之結果，最左邊的取樣影像為單調色彩的同質影像，其經過哈爾小波響應做描述元的所有分量之數值都相對的低；而圖 2.9 中間為 U 方向間隔一段距離變換色彩的影像，因此水平方向的絕對值總合 $\sum |d_u|$ 最為突出；而圖 2.9 最右邊的圖為 U 方向亮度逐漸加強的影像，因此水平方向以及水平方向 $\sum d_u$ 的絕對值總合的哈爾小波響應 $\sum |d_u|$ 的數值較大。

描述元的建立以關鍵點為中心 20 倍尺度長的方框劃分 16 個子區域，每個子區域計算四個維度的資訊，故每個描述元是由維度為 64 之特徵向量所構成。

圖 2.10 顯示實驗室一處以 SURF 特徵點擷取後的結果，其中圓的中心代表特徵點在影像平面上的位置，而圓的直徑大小代表該特徵點尺度的不同，越大的圓表示該特徵點是在較大的尺度下所計算出來的結果。

2.3 特徵點比對

本論文以影像特徵點做為定位系統之環境資訊，當擷取出影像中的特徵點

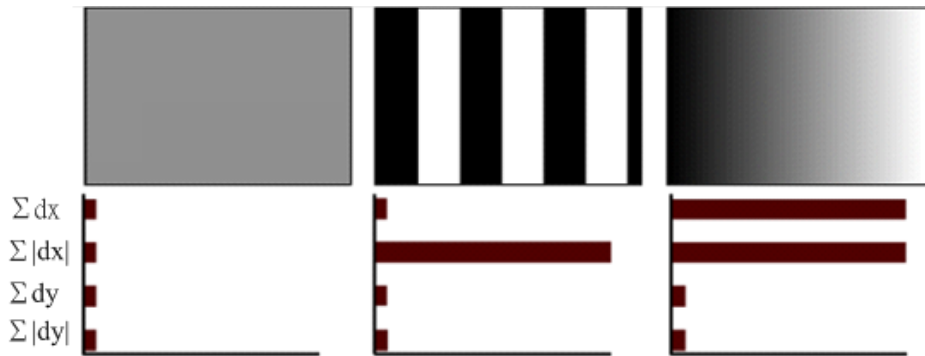


圖 2.9 哈爾小波響應轉換結果統計圖[17]



圖 2.10 SURF 影像特徵點擷取之範例

後，特徵點與資料庫特徵進行比對，與資料庫相同的影像特徵點藉由 Kinect 深度感測裝置所計算而得的三維座標，作為定位系統的輸入進而修正機器人狀態資訊。

本論文於特徵點比對上之作法是採用最鄰近演算法(Nearest Neighbor Algorithm)來計算，式 2.6 是用以計算兩不同特徵點之間的距離資訊，特徵點之描述向量為 64 維度之特徵向量，令 $Des_c(i)$ 表示為當前影像取得之特徵點的第 i 個元素，而 $Des_d(i)$ 則為資料庫中特徵點之第 i 個元素，計算兩特徵點在該維度空間下之距離，其距離 d 越短，則越有可能為相同之特徵點。

$$d = \left(\sum_{i=1}^{64} (Des_c(i) - Des_d(i))^2 \right)^{\frac{1}{2}} \quad (2.6)$$

2.4 Kinect 距離量測精度測試與校正

經過 SURF 特徵點擷取，特徵點在影像上的位置與 Kinect 深度資訊結合，透過 OpenNI 將深度資訊轉換為 Kinect 座標系下的三維座標位置，這些資訊是經

由感測器所得到的觀測量，而觀測值與實際位置的關係並不完全相同，因此在這裡我們透過實驗將由 Kinect 感測裝置所得到之量測資訊，與實際目標之位置比較並以線性的漸進曲線降低誤差量，提高特徵點資訊的精確度。

為了瞭解 Kinect 深度資訊在經過 OpenNI 轉換為三維座標資訊，其量測資訊與實際位置間的關係，我們透過實驗的方式記錄在不同位置的量測結果，此實驗是以間隔為 30mm×30mm 方格的校正板作為量測的工具，將 Kinect 感測裝置放在距離目標前從 60cm、90cm 每次隔 30 公分，到距離為 720cm 為止每個位置取 72 個已知點作量測全部共 1656 個目標點，紀錄深度資訊轉換為 Kinect 座標系下的 Y 方向座標與實際位置的差異，圖 2.11 為校正板在 180cm 時的實際影像。表 2.2 紀錄 Kinect 深度資訊在 Y 方向上不同位置實際與量測位置的平均誤差，圖 2.12 則以折線圖表示本實驗的結果，其中橫軸方向為目標點的實際位置，縱軸方向為平均的誤差值。

從 60cm 到 720cm Y 方向上的量測結果可以知道經過轉換為三維座標資訊，Y 方向量測距離的誤差隨著距離越遠其誤差值也就越大。為了降低量測的誤差以式 2.7 一階線性曲線對量測資訊作轉換，其中式 2.7 中 y 為 Kinect 座標系下的量測資訊，而 y_c 為經過漸進曲線修正後的座標值。圖 2.13 為轉換後在不同位置上

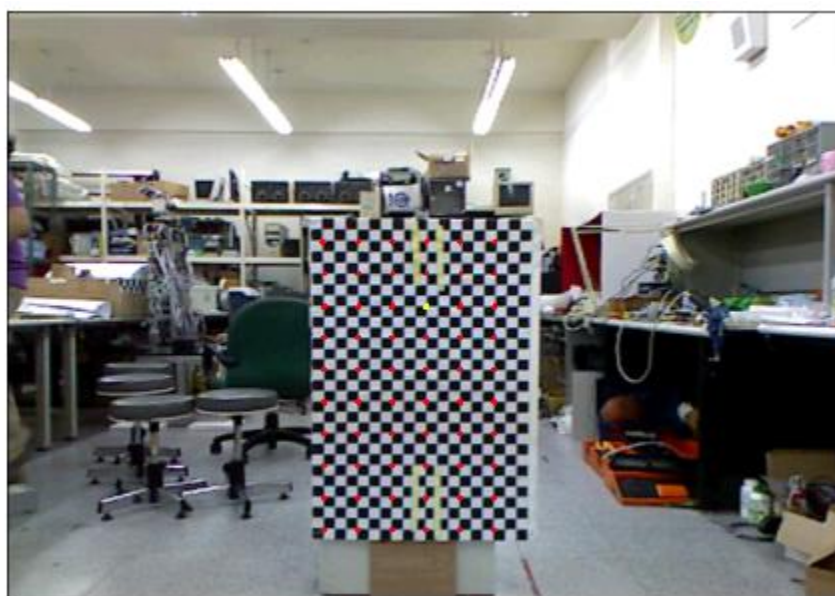


圖 2.11 校正板與 Kinect 相距 180cm 的取點結果

的平均誤差。

由 60cm-720cm 的量測結果經過漸進曲線式 2.8 的轉換，其誤差值坐落在 10cm 內如圖 2.13，但對於本論文而言，特徵點在環境中位置量測的準確度會影響機器人定位的結果，因此我們希望得到的三維座標資訊能夠更為精確，同時以之後應用所需的感測範圍做為考量，因此我們選擇使用的感測距離範圍以 Kinect 深度資訊從 70cm-350cm 作為有效範圍，而做為校正用的資訊是校正板距離 Kinect 感測 裝置 60cm、90cm 每次隔 30 公分，到距離為 330cm 為止共 10 個位

表 2.2 Kinect 量測 Y 方向實際位置與平均誤差之數值

Y 實際位置(cm)	平均誤差(cm)	Y 實際位置(cm)	平均誤差(cm)
60	0.58	420	11.51
90	0.79	450	14.21
120	1.03	480	17.86
150	1.13	510	20.95
180	1.18	540	23.57
210	1.49	570	26.39
240	2.52	600	28.63
270	3.45	630	34.55
300	5.60	660	43.57
330	6.49	690	42.93
360	8.84	720	50.54
390	10.61		

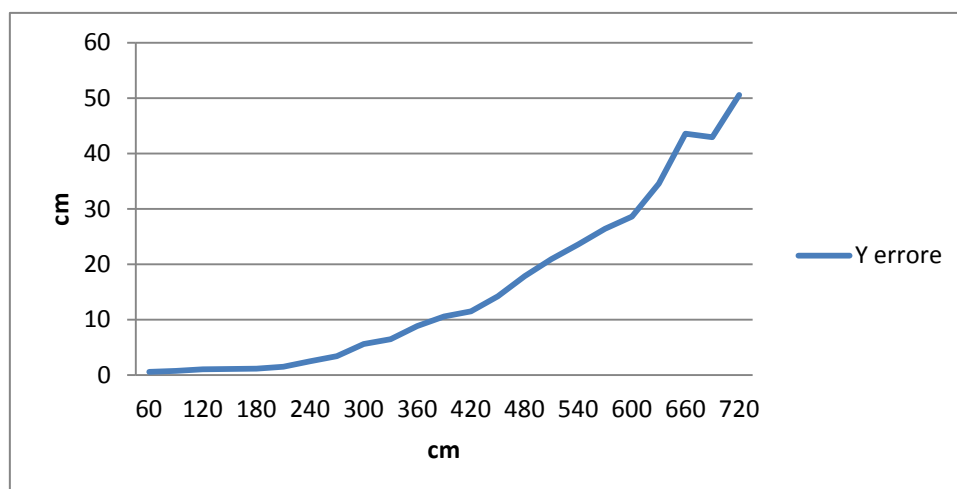


圖 2.12 Kinect 量測 Y 方向實際位置與平均誤差之關係圖

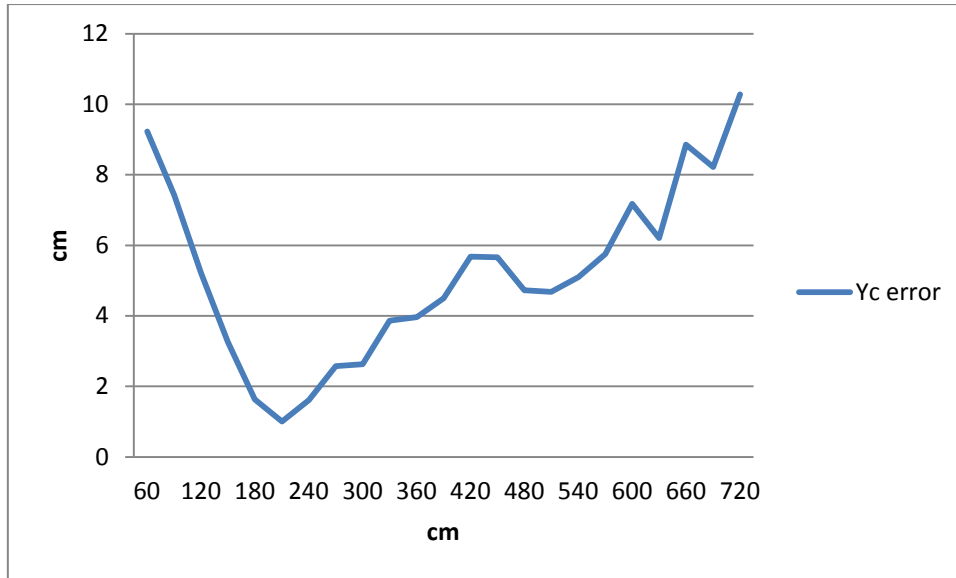


圖 2.13 Y 方向 60cm-720cm 經線性曲線轉換後與誤差之關係圖

$$y_c = 0.9292y + 13.471 \quad (2.7)$$

置每個位置取 72 個點總共 720 個目標點做統計，這段距離的資訊以一階線性曲線做校正，三個維度的資訊分別以式 2.8 做轉換，其中 x 、 y 和 z 為 Kinect 所取得的座標資訊，而 X_c 、 Y_c 和 Z_c 則為經過漸進曲線修正後的座標值。

$$X_c(x) = 1.083x - 0.529$$

$$Y_c(y) = 0.9754y + 2.7816$$

$$Z_c(z) = 1.0941z - 0.3133 \quad (2.8)$$

各分項之平均誤差與實際目標位置的結果如圖 2.14、2.15 以及 2.16 所示，分別為各方向與對應之平均誤差量之關係圖，而表 2.3、表 2.4 以及表 2.5 則記錄目標點實際位置與量測資訊間的平均誤差數值，由圖顯示統計結果，Y 方向之平均誤差約為 1 公分到 2 公分，X 方向和 Z 方向之平均誤差則在 1 公分以內。X 方向經過轉換後的座標資訊其實際距離與平均誤差的關係如圖 2.14，而圖 2.15 以及圖 2.16 則分別表示 Y 方向以及 Z 方向實際距離與平均誤差之間的關係。

2.5 特徵點於環境的三維座標資訊

本論文以視覺感測裝置實現機器人定位並同時建立地圖，以影像中的特徵點作為環境的參考資訊，特徵點於環境中的三維座標由感測裝置取得影像開始到計

表 2.3 X 方向實際距離與平均誤差之數值

X 方向 (cm)	平均誤差(cm)	X 方向 (cm)	平均誤差(cm)
-87	0.72	0	0.53
-78	0.36	9	0.76
-69	0.21	18	0.91
-60	0.29	33	0.27
-51	0.29	42	0.31
-42	0.31	51	0.23
-27	0.23	60	0.23
-18	0.18	69	0.60
-9	0.35	78	0.97

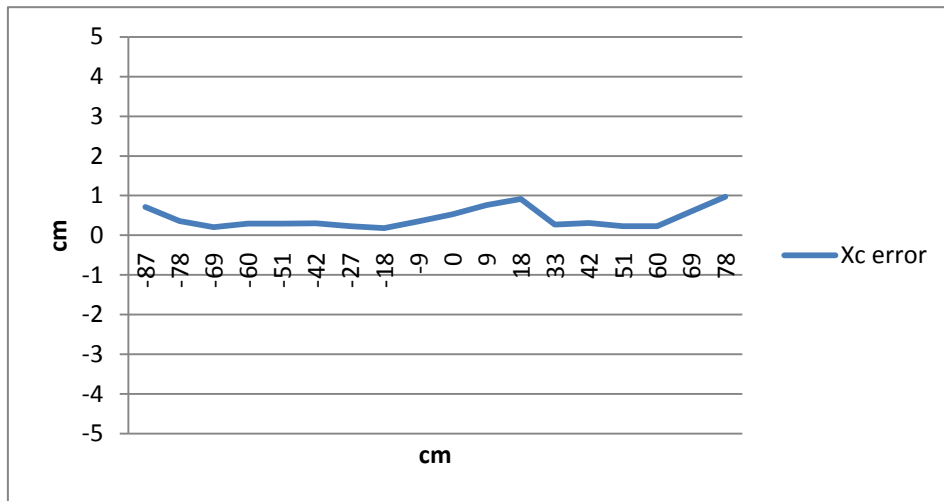


圖 2.14 特徵點 X 方向量測位置與平均誤差關係

表 2.4 Y 方向實際距離與平均誤差之數值

Y 方向 (cm)	平均誤差(cm)	Y 方向 (cm)	平均誤差(cm)
60	1.31	210	1.44
90	1.01	240	1.47
120	0.97	270	1.69
150	1.10	300	1.43
180	1.05	330	2.14

算出其座標資訊的流程如圖 2.17 所示，影像從感測裝置取得，透過 SURF 演算法擷取影像特徵點，這些影像特徵點與資料庫進行比對而能得知影像特徵中與資料庫相同的特徵點，（在這裡影像特徵與資料庫比對的結果是將影像中的特徵點分為與資料庫不相同以及與資料庫相同的兩個部分），影像中每個特徵點皆有其

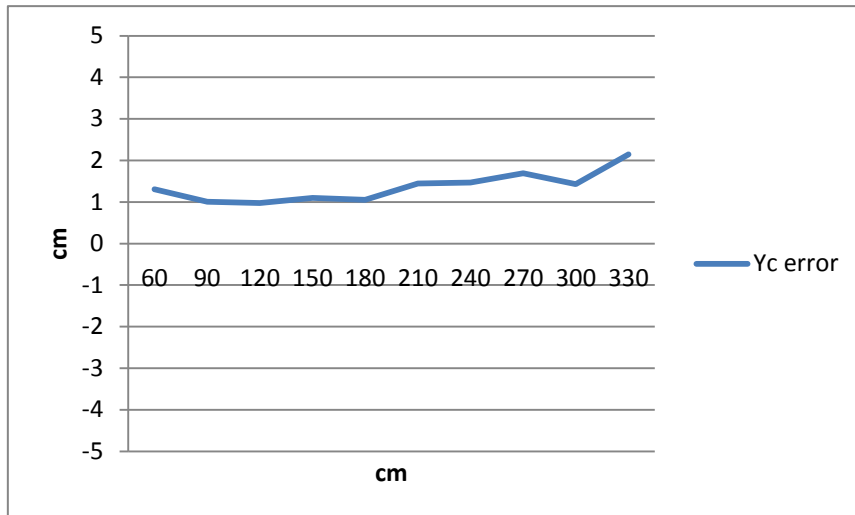


圖 2.15 特徵點 Y 方向量測位置與平均誤差關係

表 2.5 Z 方向實際距離與平均誤差之數值

Z 方向實際數值(cm)	平均誤差	Z 方向實際數值(cm)	平均誤差
18	0.32	-27	0.27
9	0.18	-36	0.31
0	0.08	-45	0.31
-9	0.15	-54	0.48
-18	0.20	-63	0.61

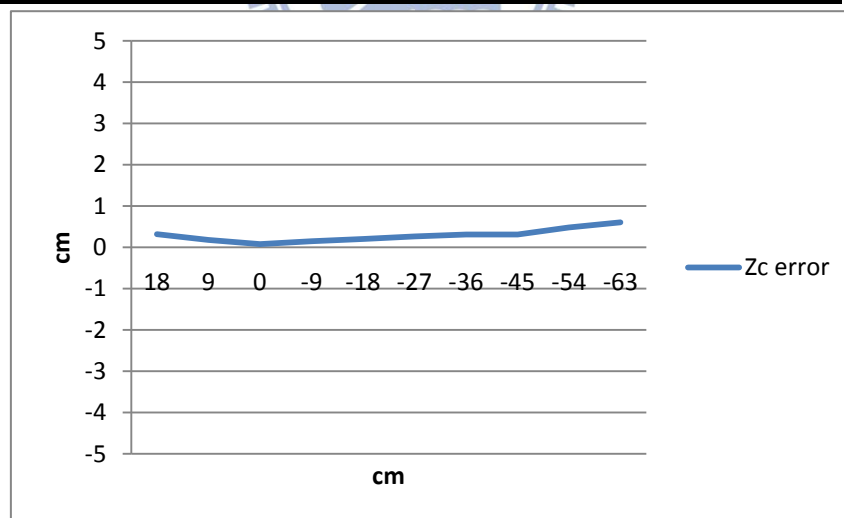


圖 2.16 特徵點 Z 方向量測位置與平均誤差關係

對應的影像平面座標，假設當次影像有 All_t 個特徵點且其對應的影像座標可用 $[Im_{1,row} \ Im_{1,col} \ \dots \ Im_{All_t,row} \ Im_{All_t,col}]$ 表示，接著，由 Kinect 所取得的深度資訊，藉由 OpenNI 將影像平面與深度資訊結合，使得每個影像座標都對應一個深度資訊，這個深度資訊以 OpenNI 轉換為 Kinect 座標系下的三維座標，於是可以得到

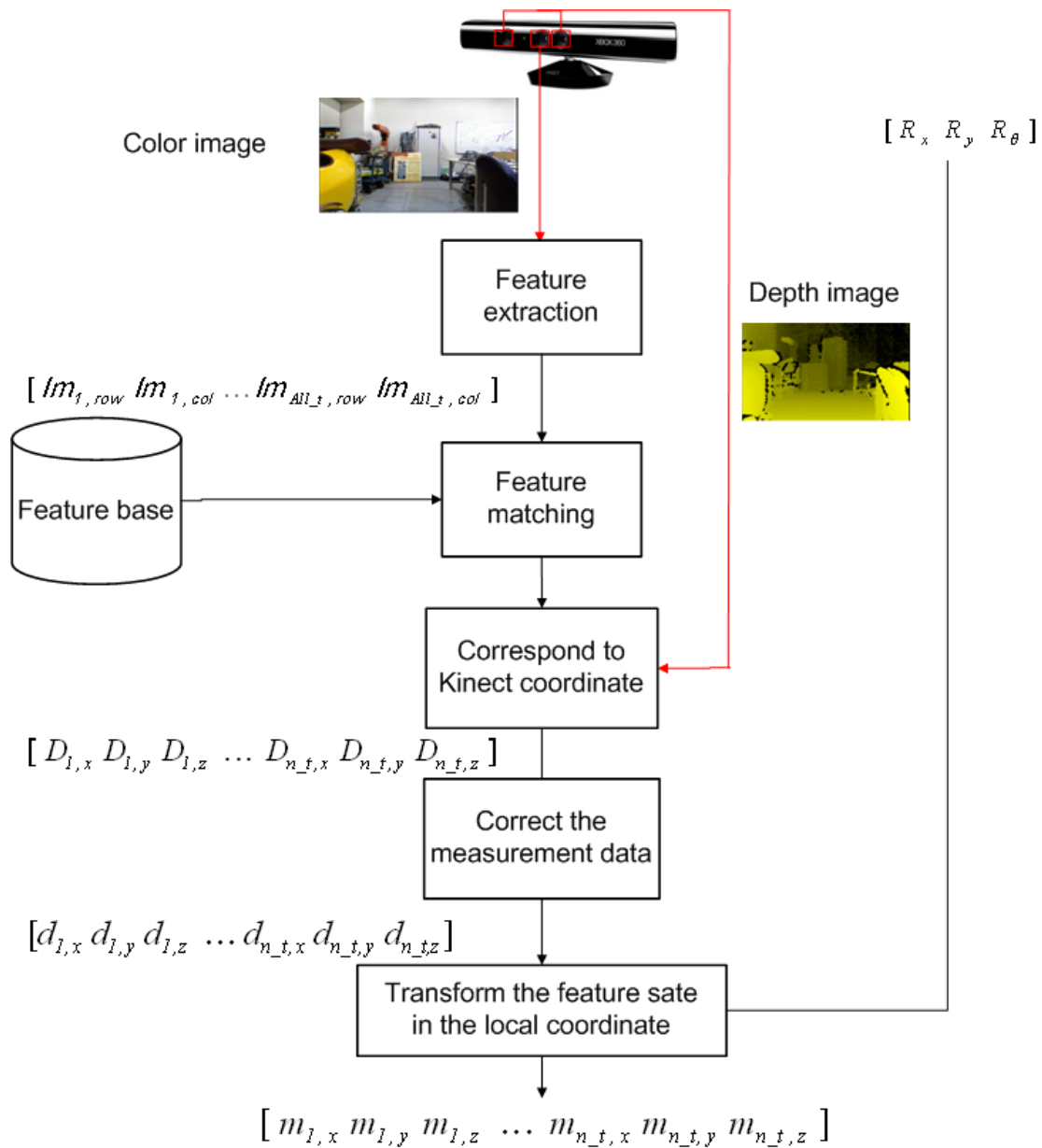


圖 2.17 影像特徵點三維座標之計算流程圖

當次影像上特徵點在 Kinect 座標系下的三維座標資訊，但因為感測裝置有其能感測的範圍限制，假設在感測範圍內的影像特徵點的總數為 n_t ，則這些特徵點在 Kinect 座標系下的三維座標以 $[D_{1,x} \quad D_{1,y} \quad D_{1,z} \quad \dots \quad D_{n_t,x} \quad D_{n_t,y} \quad D_{n_t,z}]$ 表示，接著，由上節所敘述的方式透過漸進曲線降低感測器量測數值的誤差，修正後的三維座標資訊為 $[d_{i,x} \quad d_{i,y} \quad d_{i,z}]$ (以其中第 i 個特徵點為例)，最後，結合機器人狀態資訊 $[R_x \quad R_y \quad R_\theta]$ ，轉換特徵點的座標至世界座標系，完成特徵點三維座標的計算。

在計算特徵點於世界座標的三維資訊時，根據本論文所使用之機器人移動平

台而必須對特徵點在機器人座標系下之資訊添加一修正項；因實現本論文之機器人為實驗室自製的二輪差動式移動平台，其以雙獨立驅動輪為前置，一輔助輪為後，而 Kinect 則架設於機器人正中心的位置(圖 2.18 中 R_0 的位置)，然機器人運動是控制雙獨立驅動輪做速度控制的結果，而雙輪之架構偏機器人前方，而非整體機器人的中心位置，也就是非 Kinect 所架設之中心位置，為了特徵點轉換座標系之一致性，而應以輪軸中心(圖 2.18 中 COR 的位置)為機器人中心，因此在轉換 Kinect 座標系下的特徵點三維座標到世界座標的步驟，對每個 Y 方向的座標資訊皆會扣除一個位移距離 D_{wo} 如圖 2.18，假設當前計算影像中第 i 個特徵點在環境中的三維座標，則該特徵點在 Kinect 座標系下的 Y 方向座標資訊修正如式 2.9。

$$\hat{d}_{i,y} = d_{i,y} - D_{wo} \quad (2.9)$$

得到機器人座標系下的特徵點三維座標後，透過座標轉換的關係式，結合機器人在世界座標狀態資訊，透過式 2.10 便可得到特徵點位於世界座標的三維座標 $[m_{i,x} \ m_{i,y} \ m_{i,z}]$

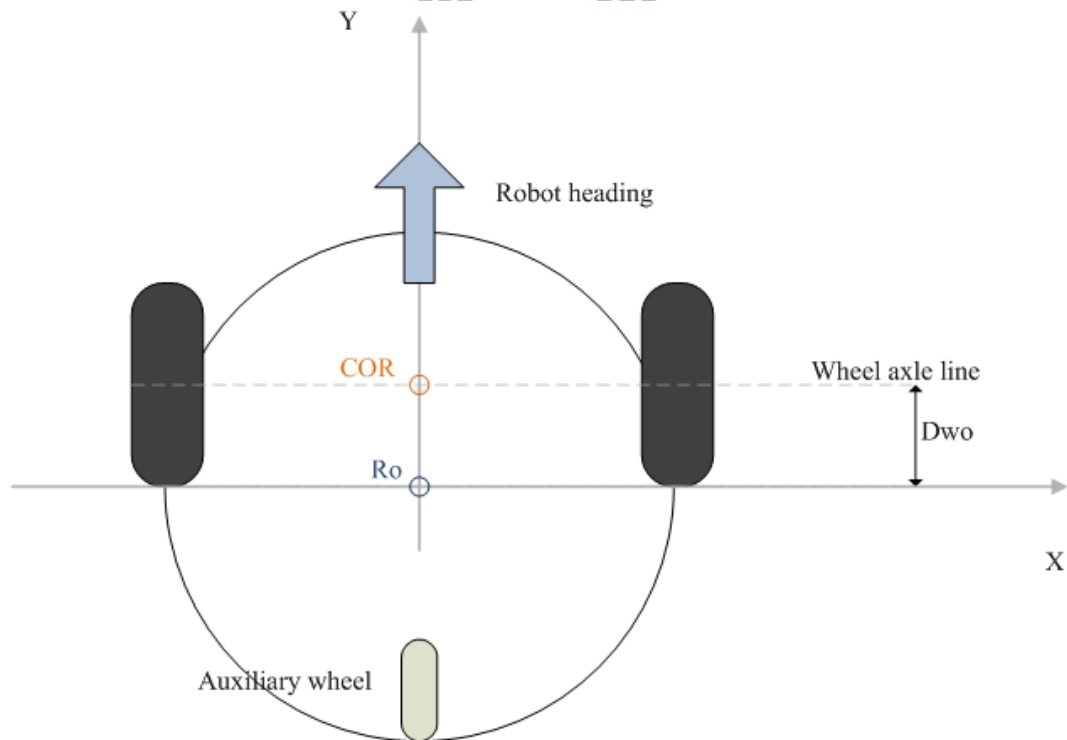


圖 2.18 座標系特徵點資訊修正之示意圖

$$\begin{bmatrix} m_{i,x} \\ m_{i,y} \end{bmatrix} = \begin{bmatrix} R_x \\ R_y \end{bmatrix} + \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} d_{i,x} \\ \hat{d}_{i,y} \end{bmatrix} \quad (2.10)$$

其中 ϕ 指的是取樣影像與深度資訊時的機器人朝向角與世界座標縱軸的夾角，因為我們是以機器人初始位置的機器人朝向角為世界座標縱軸 Y，以其水平垂直方向為橫軸 X，假設當前機器人於世界座標之朝向角為 R_θ ，則定義 ϕ ：

$$\phi = \frac{\pi}{2} - R_\theta \quad (2.11)$$

機器人狀態向量以 $[R_x R_y R_\theta]$ 代表機器人在世界座標二維座標位置和朝向角， R_H 為 Kinect 感測裝置裝設在機器人的高度，因此實際特徵點於世界座標的高度為：

$$m_{i,z} = R_H + d_{i,z} \quad (2.12)$$

2.6 特徵點資料庫的建立

機器人同時定位並建立地圖，每一次取得新的影像，經過前面數節的流程，藉由影像取得特徵點然後透過 Kinect 感測裝置得到的深度資訊並轉換為三維座標資訊，接著以漸進曲線修正量測誤差，最後經過座標轉換與機器人狀態結合得到特徵點的三維資訊，本論文地圖資料庫所使用的特徵點，皆是帶有三維座標資訊的特徵點。

每一次取得的新影像所擷取出來的特徵點，依據特徵點比對的結果以及特徵點是否帶有距離資訊，可將影像中的特徵點分為捨棄的部分、無作用的部分、新增為資料庫的部份以及作為定位系統輸入的部分四種，表 2.6 表示對應的各個狀態下的資料類型。

影像中帶有三維資訊的特徵點，會被整合成為定位系統所需的參考資訊，也就是做為地圖的特徵點資料庫，每次從影像中取得的特徵點，有些與資料庫的內容相同有些則為新得到的特徵點，本論文透過下述的方式，建立一組定位系統所需的地圖特徵點資料庫。

表 2.6 直線移動實驗之機器人位置估測結果

	與資料庫特徵點比對成功	與資料庫特徵點比對失敗
帶有距離資訊	作為定位系統的輸入	做為新增的地圖資料
無距離資訊	無作用的影像特徵點	捨棄的影像特徵點

建立資料庫的流程如圖 2.19 所示，圖 2.19(a)右側的綠色部分，表示當前影像中帶有三維資訊的特徵點，影像中取得的特徵點與資料庫的特徵以特徵點比對的方式找尋相匹配的特徵，於是影像中的特徵點可分為成功比對與沒有成功比對的兩部分如圖 2.19(b)，比對成功的部分，會被用在定位系統的使用上，在圖 2.19 上標記為 EKF input 的部分，而沒有與資料庫特徵相同的新影像特徵點，則做為擴增資料庫的特徵點如圖 2.19(c)，從(a)到(c)完成一次資料庫的更新，更新而得一組新的特徵點資料庫(d)。

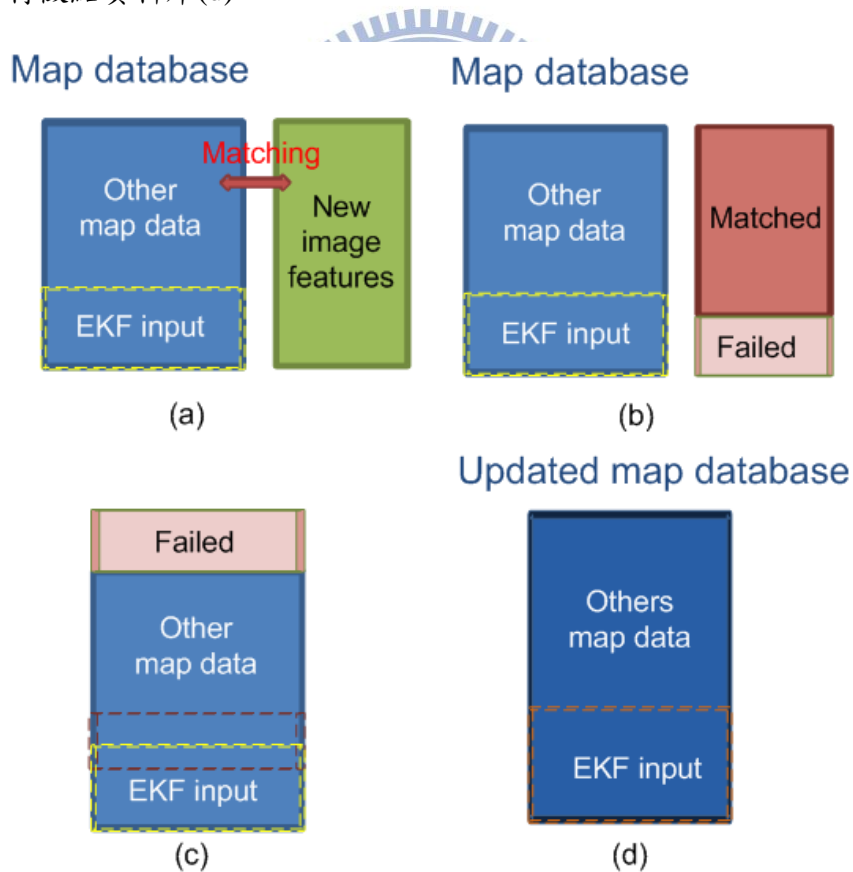


圖 2.19 資料庫統合影像特徵點訊息之步驟

第三章 機器人定位系統

機器人如要能夠得知自己在環境中的位置，最常使用的方式應該是透過馬達的軸編碼器計數，藉由里程計的計算得知自身行進的距離以及旋轉的角度。然因地表光滑或地面崎嶇甚至是受到機器人硬體結構的影響造成機器人運行時發生打滑或抖動等現象，在這些情況下當機器人行走越長的距離，里程計所產生的累積誤差也就越嚴重。研究者以環境提供相對或絕對的參考資訊計算機器人於環境中的狀態，其參考資訊依感測裝置的不同，可由影像取得特徵點的三維座標或以距離感測裝置得到環境的二維資訊，又或是以無線網路的網路節點得到機器人與節點的相對位置關係，假設所得到的環境資訊與實際情形相符，則我們可以計算單一的結果代表機器人的狀態資訊，然而，實際上環境充滿各種不確定性因素，觀測資訊會因相似環境或相同物品而導致計算結果與實際不符，相較於以觀測資訊取得單一的結果的做法，概率性機器人位置估測在定位演算法上以機率之方式計算機器人在整個空間位置的可能性分布，以程度之方式合理計算機器人狀態資訊，並以可能性最高的位置做為機器人位置估測的結果。

概率性機器人位置估測於定位系統上的使用，貝式濾波(Bayes filter)以感測器所取得的環境資訊，以遞歸的方式計算當前機器人在整個空間的機率分佈，而 Kalman Filter 為實現 Bayes Filter 中極具代表性的方法，主要是針對線性高斯系統下濾波以及預測的技術，然而，實際上機器人運動多為非線性模型，因此衍生 Kalman Filter 演算法而有 EKF(Extended Kalman Filter 拓展式卡曼濾波器)，EKF 將非線性的機器人運動模型以及特徵點觀測模型線性化並以最佳化融合的方式實現機器人狀態估測，本論文即是以 EKF 做為機器人定位的演算法。

EKF 的整體流程可分為三個主要部分，觀測步驟取得環境中機器人相對於特徵點之間的關係，而預測步驟則以前一時刻之機器人狀態向量與運動命令計算當前機器人位置，預測之機器人狀態向量與特徵點觀測資訊做為更新步驟之輸入，定位系統將兩者資訊融合得到更新的系統狀態向量(系統狀態向量為機器人狀態

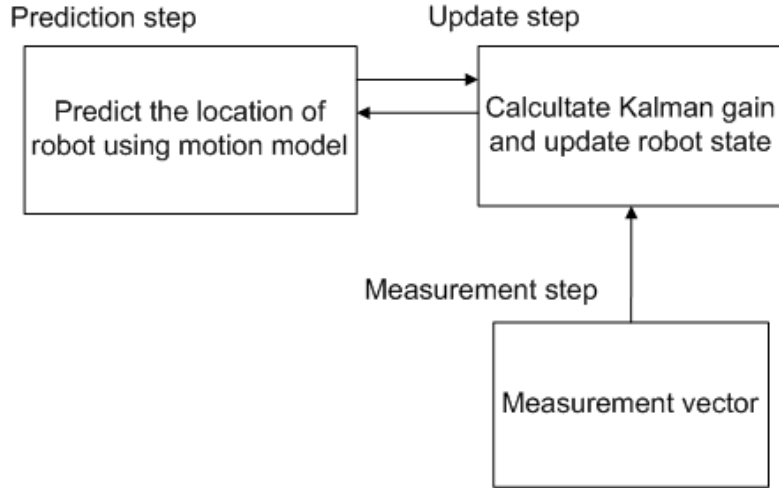


圖 3.1 EKF SLAM 演算示意圖

向量與特徵點狀態向量合併之資訊如 3.1 式), 而更新後的機器人狀態向量傳回預測步驟用以計算下一時刻狀態, 整個 EKF 即是在這三個步驟下不斷循環。

本論文即以 Extended Kalman Filter 作為機器人定位的演算法, 透過觀測資訊與估測資訊以最佳化融合的方式, 找出最有可能的機器人位置, 並且持續更新系統狀態向量, 達成機器人同時定位以及建立地圖。

3.1 式代表 t 時刻的系統狀態向量 f_t 其中包含機器人狀態向量 q_t 以及資料庫特徵點狀態向量 M_t , 機器人狀態向量 q_t 包含二維座標 (x_t, y_t) 以及朝向角 θ_t , 而 M_t 是由 N 個特徵點所組成的向量, 每個特徵點皆有一組特徵點狀態向量, 其資訊為特徵點於環境中的三維座標, 以第 i 個特徵點為例, 其三維座標表示為 $(M_{i,x}, M_{i,y}, M_{i,z})$,

$$\begin{aligned}
 f_t &= \begin{bmatrix} q_t \\ M_t \end{bmatrix} \\
 &= [x_t \ y_t \ \theta_t \ M_{1,x} \ M_{1,y} \ M_{1,z} \ \cdots \ M_{N,x} \ M_{N,y} \ M_{N,z}]^T
 \end{aligned} \tag{3.1}$$

在這裡的系統狀態向量 f_t 表示未知的變數, 而之後更新步驟下所使用的系統狀態向量 $\bar{\mu}_t$ 則是已知的估測狀態, 而 μ_t 則是經過 EKF 演算法後最終的定位結果。

3.1 機器人運動模型

本論文所使用的機器人為二輪差動式移動平台, 其由兩直流馬達做為主要驅動, 後方則為一被動式輔助輪, 其運動控制命令上以速度控制來操作, 機器人之

狀態向量 $(x_t \ y_t \ \theta_t)$ 可由線速度 v 以及角速度 w 推導。

令機器人運動控制命令為 $u_t = (v_t \ w_t)^T$ ，假設機器人在 Δt 時間內，皆保持相同的運動控制命令，則機器人會以一半徑為 r 的弧線繞圓，如圖 3.2 所示定義該半徑：

$$r = \frac{v_t}{w_t} \quad (3.2)$$

假設 $t-1$ 時刻機器人狀態向量為 $(x_{t-1} \ y_{t-1} \ \theta_{t-1})$ ，由 3.2 式半徑與速度和角速度的關係，可以推得機器人所繞之圓心座標為：

$$\begin{aligned} x_c &= x_{t-1} - \frac{v_t}{w_t} \sin \theta_{t-1} \\ y_c &= y_{t-1} + \frac{v_t}{w_t} \cos \theta_{t-1} \end{aligned} \quad (3.3)$$

當機器人以運動控制命令 $u_t = (v_t \ w_t)^T$ 經過 Δt 的時間後機器人狀態向量變成 q_t ，且其繞著同樣的旋轉中心 $(x_c \ y_c)$ ，藉由兩式相等的結果可以推得機器人狀態向量 q_t 如式 3.4：

$$q_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} -\frac{v_t}{w_t} \sin \theta_{t-1} + \frac{v_t}{w_t} \sin(\theta_{t-1} + w_t \Delta t) \\ \frac{v_t}{w_t} \cos \theta_{t-1} - \frac{v_t}{w_t} \cos(\theta_{t-1} + w_t \Delta t) \\ w_t \Delta t \end{bmatrix} \quad (3.4)$$

若機器人以直線運動的方式前進時，上述的運動模型並不能正確的表示其運

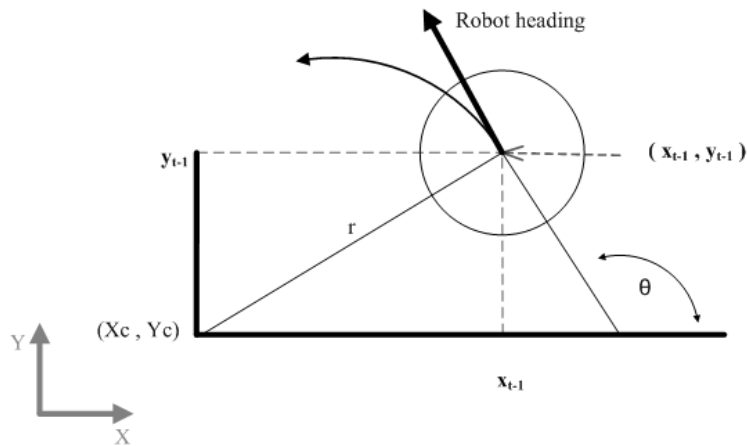


圖 3.2 機器人運動模型表示圖

動狀態，因直線運動時的角速度為 0 且旋轉半徑為無限大，故機器人直線運動是以式 3.5 表示之。

$$q_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} v_t \cos \theta_{t-1} \\ v_t \sin \theta_{t-1} \\ \theta_{t-1} \end{bmatrix} \quad (3.5)$$

理想上，機器人以 $u_t = (v_t \ w_t)^T$ 作為運動控制命令，行進了 Δt 的時間而得到下一時刻的機器人狀態資訊，然而，在實際的情況下，機器人的運動會受到雜訊影響，因此實際的機器人運動除了以理想的速度與角速度前進外必須加上雜訊項以符合實際情形，所以我們將速度運動控制的命令項改寫為式 3.6， $\varepsilon_{\alpha_1|v_t|+\alpha_2|w_t|}$ 代表平均值為 0 且標準差為 $\alpha_1|v_t|+\alpha_2|w_t|$ 的高斯雜訊，而 $\alpha_1 \sim \alpha_4 \geq 0$ 作為描述機器人誤差大小的參數，當機器人存在越大的誤差，則該四個參數的值也就越大。

$$\begin{bmatrix} \hat{v}_t \\ \hat{w}_t \end{bmatrix} = \begin{bmatrix} v_t \\ w_t \end{bmatrix} + \begin{bmatrix} \varepsilon_{\alpha_1|v_t|+\alpha_2|w_t|} \\ \varepsilon_{\alpha_3|v_t|+\alpha_4|w_t|} \end{bmatrix} \quad (3.6)$$

EKF 演算法假設雜訊為高斯模型，因此速度運動控制 (\hat{v}_t, \hat{w}_t) 可寫為理想的速度控制命令 u_t 加上雜訊 $N(0, M_t)$ ，其中 M_t 為運動控制命令中雜訊模型之共變異數矩陣如式 3.7。

$$M_t = \begin{bmatrix} (\alpha_1|v_t|+\alpha_2|w_t|)^2 & 0 \\ 0 & (\alpha_3|v_t|+\alpha_4|w_t|)^2 \end{bmatrix} \quad (3.7)$$

雜訊項影響運動控制命令的結果同時也影響機器人狀態估測，故機器人狀態可由一個理想的狀態估測後加上位移雜訊量來表示，如 3.8 示。

$$q_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \underbrace{\begin{bmatrix} -\frac{v_t}{w_t} \sin \theta_{t-1} + \frac{v_t}{w_t} \sin \theta_{t-1} (+ w_t \Delta t) \\ \frac{v_t}{w_t} \cos \theta_{t-1} - \frac{v_t}{w_t} \cos \theta_{t-1} (+ w_t \Delta t) \\ w_t \Delta t \end{bmatrix}}_{g(u_t, q_{t-1})} + N(\mathbf{0}_t), \quad (3.8)$$

其中 E_t 為狀態向量雜訊的共變異數矩陣，其是由運動控制命令雜訊 M_t 轉換至

狀態向量空間而得：

$$E_t = V_t \cdot M_t \cdot V_t^T \quad (3.9)$$

其中 V_t 為將控制命令雜訊 M_t 轉換到狀態空間的轉換矩陣

$$V_t = \frac{\partial g(u_t, q_{t-1})}{\partial u_t} = \begin{bmatrix} \frac{\partial x_t}{\partial v_t} & \frac{\partial x_t}{\partial w_t} \\ \frac{\partial y_t}{\partial v_t} & \frac{\partial y_t}{\partial w_t} \\ \frac{\partial \theta_t}{\partial v_t} & \frac{\partial \theta_t}{\partial w_t} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-\sin \theta_{t-1} + \sin(\theta_{t-1} + w_t \Delta t)}{w_t} & \frac{v_t (\sin \theta_{t-1} - \sin(\theta_{t-1} + w_t \Delta t))}{w_t^2} + \frac{v_t \cos(\theta_{t-1} + w_t \Delta t) \Delta t}{w_t} \\ \frac{\cos \theta_{t-1} - \cos(\theta_{t-1} + w_t \Delta t)}{w_t} & -\frac{v_t (\cos \theta_{t-1} - \cos(\theta_{t-1} + w_t \Delta t))}{w_t^2} + \frac{v_t \sin(\theta_{t-1} + w_t \Delta t) \Delta t}{w_t} \\ 0 & \Delta t \end{bmatrix} \quad (3.10)$$

3.2 特徵點觀測模型

特徵點觀測模型指的是特徵點與機器人之間的相對座標關係，其包括由感測裝置所量測到的特徵點訊息以及資料庫特徵點與機器人之間的關係，其分別被定義為觀測之量測向量(Observed measurement vector)與預測之量測向量(Predicted measurement vector)，在本論文中量測向量包括特徵點與機器人中心點在二維平面上之直線距離 r ，特徵點至機器人中心與機器人朝向間的水平夾角 θ_d ，以及特徵點和感測裝置中心垂直夾角 ϕ 三個資訊所構成，假設 t 時刻的機器人狀態向量為 $(x_t \ y_t \ \theta_t)$ ，其高度為感測器距離水平面的垂直距離 R_H ，而空間中某個特徵點的三維座標資訊為 $(M_{i,x} \ M_{i,y} \ M_{i,z})$ ，圖 3.3 與圖 3.4 以一個點代表機器人的中心位置，說明機器人中心與特徵點在各分量的關係，並以 $h(q_t, i)$ 表示為機器人狀態向量 q_t 與空間中第 i 個特徵點間的觀測模型 (Observation model)：

$$h(q_t, i) = \begin{bmatrix} r \\ \theta_d \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(M_{i,x} - x_t)^2 + (M_{i,y} - y_t)^2} \\ \text{atan2}(M_{i,y} - y_t, M_{i,x} - x_t) - \theta_t \\ \text{atan}((M_{i,z} - R_H) / \sqrt{(M_{i,x} - x_t)^2 + (M_{i,y} - y_t)^2}) \end{bmatrix} \quad (3.11)$$

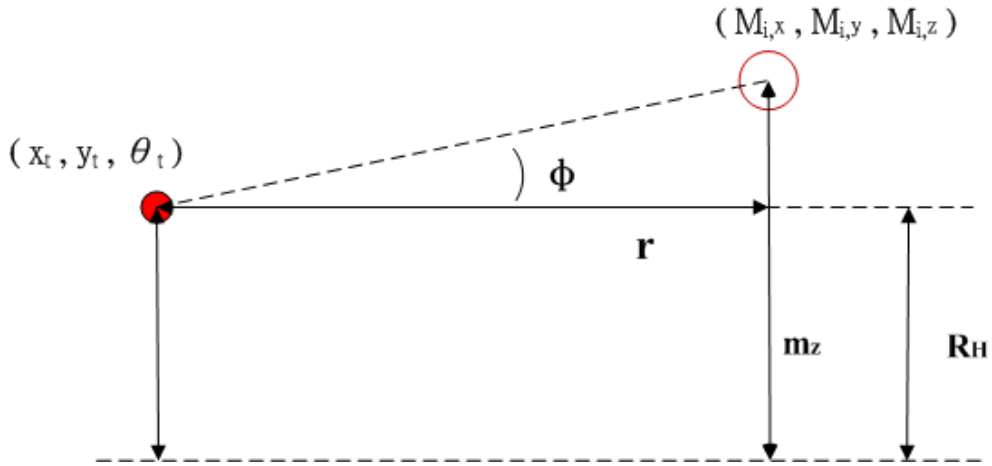


圖 3.3 機器人中心與特徵點在二維座標直線距離 r 和垂直夾角 ϕ

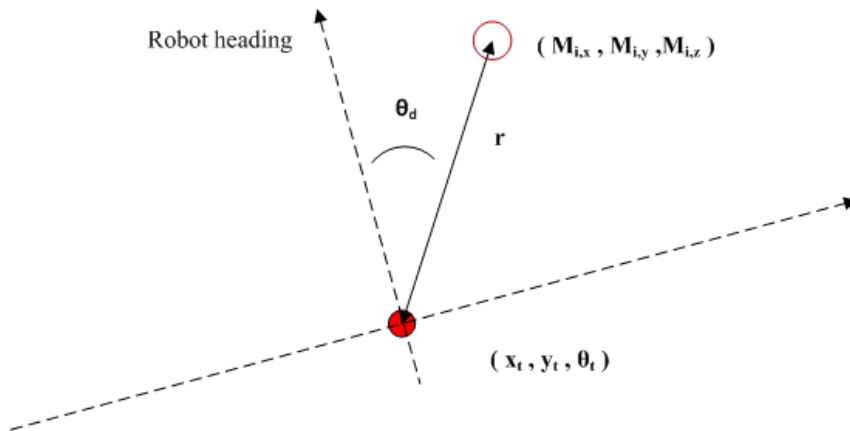


圖 3.4 機器人量測模型圖示

在 2.4 節中，藉由 Kinect 感測裝置所得到之特徵點三維座標，在本論文中，所有在當前影像中帶有距離資訊的特徵點都會進入資料庫中，依據 2.5 節中資料庫的分類方式，特徵點被分為僅為帶有距離的資訊，以及被重複比對的特徵點訊息，這些被重複比對到的特徵點訊息在 EKF 定位演算法中是很重要的要素，以 t 時刻所取得的影像中特徵點，依 2.4 節中距離資訊與特徵點的計算得到特徵點的三維座標資訊 $(m_{i,x} \ m_{i,y} \ m_{i,z})$ ，定義下標 i 為對應當前比對成功的第 i 個特徵點，假設 t 時刻估測的機器人狀態向量為 $(\hat{R}_{t,x} \ \hat{R}_{t,y} \ \hat{R}_{t,\theta})$ ，則其量測向量資訊為 $(r_t^i \ \theta_{d_t}^i \ \phi_t^i)$ 如式 3.12，由於其為 t 時刻擷取影像中特徵與機器人的相對關係式，故又稱其為觀測之量測向量 (Observed measurement vector) Z_t^i ，然而，在實際的量測中感測裝置所得之量測資訊並非完全準確，其資訊包含雜訊存在，在 EKF 中以高斯雜訊定義其誤差，令 $N(0, Q_t)$ 為平均值為 0 且共變異數矩陣為 Q_t 之高斯雜訊。

$$Z_t^i = \begin{bmatrix} r_t^i \\ \theta_{d_t}^i \\ \phi_t^i \end{bmatrix} + N(0, Q_t) \quad (3.12)$$

$$= \begin{bmatrix} \sqrt{(m_{i,x} - \hat{R}_{t,x})^2 + (m_{i,y} - \hat{R}_{t,y})^2} \\ \text{atan2}(m_{i,y} - \hat{R}_{t,y}, m_{i,x} - \hat{R}_{t,x}) - \hat{R}_{t,\theta} \\ \text{atan}((m_{i,z} - R_H) / \sqrt{(m_{i,x} - \hat{R}_{t,x})^2 + (m_{i,y} - \hat{R}_{t,y})^2}) \end{bmatrix} + N(0, Q_t)$$

$$Q_t = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix} \quad (3.13)$$

與 t 時刻影像中取得的特徵點相同之資料庫特徵點，其資訊依不同設計方式可用事先建立之環境固定座標，又或是隨機器人運行所建立之環境資訊，本論文為機器人同時定位與建立地圖之應用，所採用之作法是隨機器人運行而建立之特徵點訊息，影像中的特徵點被整合至地圖資料庫，當前影像所比對到的資料庫特徵點是過去影像所整合的環境特徵點，假設 t 時刻估測的機器人狀態向量為 $(\hat{R}_{t,x}, \hat{R}_{t,y}, \hat{R}_{t,\theta})$ ，對應 t 時刻影像中所得之第 i 個共同特徵點，其對應系統狀態向量內的資料庫特徵點序列為 $k=C_t^i$ ，其特徵點狀態向量 \hat{Z}_t^i 可由 3.14 式計算而得，這個資訊是由資料庫特徵計算而得，故又稱之為預測之量測向量(Predicted measurement vector)：

$$\hat{Z}_t^i = \begin{bmatrix} \hat{r}_t^i \\ \hat{\theta}_{d_t}^i \\ \hat{\phi}_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{(\hat{m}_{k,x} - \hat{R}_{t,x})^2 + (\hat{m}_{k,y} - \hat{R}_{t,y})^2} \\ a \tan 2(\hat{m}_{k,y} - \hat{R}_{t,y}, \hat{m}_{k,x} - \hat{R}_{t,x}) - \hat{R}_{t,\theta} \\ \text{atan}((\hat{m}_{k,z} - R_H) / \sqrt{(\hat{m}_{k,x} - \hat{R}_{t,x})^2 + (\hat{m}_{k,y} - \hat{R}_{t,y})^2}) \end{bmatrix} \quad (3.14)$$

觀測之量測向量(Observed measurement vector) Z_t 為 t 時刻所得之機器人與特徵點相對關係，而資料庫中對應的共同特徵點與機器人間的相對關係 \hat{Z}_t ，則代表理想上這些共同特徵點與機器人的相對資訊，兩者之間的差值代表著 t 時刻機器人位置實際與理想上的差異程度，其在 EKF 演算法為非常重要的一項資訊，稱之為 Innovation vector：

$$\text{Innovation vector} = Z_t - \hat{Z}_t \quad (3.15)$$

3.3 EKF 演算法

本演算法主要是參考自 Thrun et al.[26]所提之 EKF SLAM 流程。EKF 演算法是用前一時刻所估測得到的機器人狀態向量與特徵點狀態向量以及觀測而得的特徵點狀態向量作為輸入，修正機器人狀態並更新特徵點資料庫；因此，演算法需要前一時刻的機器人狀態 R_{t-1} 以及特徵點狀態向量 \hat{Z}_{t-1} ，這兩個狀態向量組成系統狀態向量 μ_{t-1} ，而 Σ_{t-1} 則代表這個系統狀態向量的共變異數矩陣，而 t 時刻的機器人是從狀態 R_{t-1} 以運動控制 u_t 經過 Δt 後所到達，此時透過感測器擷取影像中的特徵點，這些特徵點與資料庫進行比對後而得到觀測之量測特徵點向量 Z_t ，整個演算法即以上述資訊做為開端，藉由疊代的方式對系統狀態向量進行修正。假設特徵點資料庫的數量為 N 個，則系統狀態向量是由 $3+3N$ 個元素所構成如式 3.1，而共變異數矩陣 Σ_t 則為 $(3+3N) \times (3+3N)$ 之矩陣。

式 3.16 為系統狀態向量 f_{t-1} 以運動控制 u_t 經過 Δt 的時間，由 3.1 的推導可估測 t 時刻的系統狀態向量：

$$f_t = f_{t-1} + \begin{bmatrix} -\frac{v_t}{w_t} \sin \theta_{t-1} + \frac{v_t}{w_t} \sin(\theta_{t-1} + w_t \Delta t) \\ \frac{v_t}{w_t} \cos \theta_{t-1} - \frac{v_t}{w_t} \cos(\theta_{t-1} + w_t \Delta t) \\ w_t \Delta t \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3.16)$$

由於實際上機器人的運動結果受到雜訊所影響，演算法以平均值為 0 共變異數矩陣為 M_t 之雜訊作為運動控制的雜訊，而影響機器人狀態向量的結果，則是將該雜訊從運動控制轉換至機器人狀態空間，如同 3.1 節所述，以 E_t 代表因運動命令雜訊對機器人狀態影響之雜訊共變異數矩陣，式 3.17 所得到的便是由前一時刻系統狀態向量 f_{t-1} 以帶有雜訊 $N(0, M_t)$ 的運動控制 u_t 經過 Δt 後所估測之系統狀態向量 f_t ，而 F_x 則用以簡化矩陣表示式，其作用為將矩陣由維度 3 轉換至 $3+3N$ ：

$$f_t = f_{t-1} + F_x^T \underbrace{\begin{bmatrix} -\frac{v_t}{w_t} \sin \theta_{t-1} + \frac{v_t}{w_t} \sin(\theta_{t-1} + w_t \Delta t) \\ \frac{v_t}{w_t} \cos \theta_{t-1} - \frac{v_t}{w_t} \cos(\theta_{t-1} + w_t \Delta t) \\ w_t \Delta t \end{bmatrix}}_{g(u_t, f_{t-1})} + N(0, F_x^T E_t F_x) \quad (3.17)$$

$$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ & & & \underbrace{\hspace{2cm}}_{3N} & & \end{bmatrix} \quad (3.18)$$

EKF 將非線性的運動模型線性化，將運動模型 $g(u_t, f_{t-1})$ 於 $t-1$ 時刻的系統狀態向量 μ_{t-1} 以一階泰勒展開：

$$g(u_t, f_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t (f_{t-1} - \mu_{t-1}) \quad (3.19)$$

Jacobian matrix G_t 為運動模型 $g(u_t, f_{t-1})$ 對系統狀態變數之偏導數，因運動模型 $g(u_t, f_{t-1})$ 對機器人狀態變數以及相同的特徵點資訊有意義，因此運動模型於 $t-1$ 時刻的系統狀態向量 μ_{t-1} 對系統狀態偏導數的結果如式 3.20，其中 $\frac{\partial x_t}{\partial R_{t-1,x}}$ 為系統運動模型中的機器人狀態分量 x_t 在系統狀態為 μ_{t-1} 時對 x_{t-1} 偏導數的簡寫：

$$\begin{aligned} G_t &= \frac{\partial g(u_t, \mu_{t-1})}{\partial f_{t-1}} \\ &= I + F_x^T \begin{bmatrix} \frac{\partial x_t}{\partial R_{t-1,x}} & \frac{\partial x_t}{\partial R_{t-1,y}} & \frac{\partial x_t}{\partial R_{t-1,\theta}} \\ \frac{\partial y_t}{\partial R_{t-1,x}} & \frac{\partial y_t}{\partial R_{t-1,y}} & \frac{\partial y_t}{\partial R_{t-1,\theta}} \\ \frac{\partial \theta_t}{\partial R_{t-1,x}} & \frac{\partial \theta_t}{\partial R_{t-1,y}} & \frac{\partial \theta_t}{\partial R_{t-1,\theta}} \end{bmatrix} F_x \\ &= I + F_x^T \begin{bmatrix} 0 & 0 & \frac{v_t}{w_t} (-\cos R_{t-1,\theta} + \cos(R_{t-1,\theta} + w_t \Delta t)) \\ 0 & 0 & \frac{v_t}{w_t} (-\sin R_{t-1,\theta} + \sin(R_{t-1,\theta} + w_t \Delta t)) \\ 0 & 0 & 0 \end{bmatrix} F_x \quad (3.20) \end{aligned}$$

運動控制命令 u_t 以及 t-1 時刻的系統狀態向量 μ_{t-1} 可以估測 t 時刻的系統狀態向量 $\bar{\mu}_t$ ，而對應 t 時刻系統狀態向量的共變異數矩陣以式 3.21 的方式計算，其中 E_t 是由運動控制命令雜訊轉換至狀態空間的共變異數矩陣。

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T E_t F_x \quad (3.21)$$

預測步驟下給予更新步驟的資訊為透過運動控制命令及 t-1 時刻的系統狀態向量所估測的系統狀態向量 $\bar{\mu}_t$ 、共變異數矩陣 $\bar{\Sigma}_t$ 以及特徵點與機器人間的預測之量測向量 \hat{Z}_t ，而觀測步驟下則是提供當前影像中與資料庫特徵點相同的觀測之量測向量 Z_t 。

於 t 時刻所得到的影像特徵點中，與特徵點資料庫比對成功的共同特徵點作為 EKF 演算法的輸入，而特徵點與機器人狀態的關係如同 3.2 節所描述，是以觀測模型敘述機器人中心位置與特徵點的相對關係，在這裡以系統狀態向量進行運算，觀測模型 $h(f_t, i)$ 表示當前影像中所得到的觀測之量測向量 Z_t 中第 i 個特徵點與機器人之間的相對關係，EKF 演算法假設觀測模型為線性，因此觀測模型近似為線性方程式如式 3.22，其中 H_t^i 為觀測模型於系統狀態向量為 $\bar{\mu}_t$ 時對系統狀態變數偏導數的結果：

$$h(f_t, i) \approx h(\bar{\mu}_t, i) + H_t^i (f_t - \bar{\mu}_t) \quad (3.22)$$

H_t^i 由觀測模型 $h(f_t, i)$ 在系統狀態向量為 $\bar{\mu}_t$ 時對系統狀態向量之偏導數而得，其僅與機器人狀態變數和對應之第 k 個資料庫特徵點有意義：

$$H_t^i = \frac{\partial h(\bar{\mu}_t, j)}{\partial f_t} = \begin{bmatrix} \frac{\partial r_t^i}{\partial \hat{R}_{t,x}} & \frac{\partial r_t^i}{\partial \hat{R}_{t,y}} & \frac{\partial r_t^i}{\partial \hat{R}_{t,\theta}} & \cdots & \frac{\partial r_t^i}{\partial \hat{m}_{k,x}} & \frac{\partial r_t^i}{\partial \hat{m}_{k,y}} & \frac{\partial r_t^i}{\partial \hat{m}_{k,z}} & \cdots \\ \frac{\partial \theta_{d_t}^i}{\partial \hat{R}_{t,x}} & \frac{\partial \theta_{d_t}^i}{\partial \hat{R}_{t,y}} & \frac{\partial \theta_{d_t}^i}{\partial \hat{R}_{t,\theta}} & \cdots & \frac{\partial \theta_{d_t}^i}{\partial \hat{m}_{k,x}} & \frac{\partial \theta_{d_t}^i}{\partial \hat{m}_{k,y}} & \frac{\partial \theta_{d_t}^i}{\partial \hat{m}_{k,z}} & \cdots \\ \frac{\partial \phi_t^i}{\partial \hat{R}_{t,x}} & \frac{\partial \phi_t^i}{\partial \hat{R}_{t,y}} & \frac{\partial \phi_t^i}{\partial \hat{R}_{t,\theta}} & \cdots & \frac{\partial \phi_t^i}{\partial \hat{m}_{k,x}} & \frac{\partial \phi_t^i}{\partial \hat{m}_{k,y}} & \frac{\partial \phi_t^i}{\partial \hat{m}_{k,z}} & \cdots \end{bmatrix} = h_t^i F_{x,k} \quad (3.23)$$

其中 h_t^i 代表對機器人狀態向量以及特徵點狀態向量的偏導數，而 $F_{x,k}$ 則是用以簡化矩陣表示式，其主要的用意在於表示觀測模型對系統狀態的偏導數僅與機器人狀態向量以及對應第 k 個特徵點資料庫有意義。

$$h_t^i = \frac{\partial h(\bar{\mu}_t, i)}{\partial f_t} = \begin{bmatrix} \frac{\partial r_t^i}{\partial \hat{R}_{t,x}} & \frac{\partial r_t^i}{\partial \hat{R}_{t,y}} & \frac{\partial r_t^i}{\partial \hat{R}_{t,\theta}} & \frac{\partial r_t^i}{\partial \hat{m}_{k,x}} & \frac{\partial r_t^i}{\partial \hat{m}_{k,y}} & \frac{\partial r_t^i}{\partial \hat{m}_{k,z}} \\ \frac{\partial \theta_{dt}^i}{\partial \hat{R}_{t,x}} & \frac{\partial \theta_{dt}^i}{\partial \hat{R}_{t,y}} & \frac{\partial \theta_{dt}^i}{\partial \hat{R}_{t,\theta}} & \frac{\partial \theta_{dt}^i}{\partial \hat{m}_{k,x}} & \frac{\partial \theta_{dt}^i}{\partial \hat{m}_{k,y}} & \frac{\partial \theta_{dt}^i}{\partial \hat{m}_{k,z}} \\ \frac{\partial \phi_t^i}{\partial \hat{R}_{t,x}} & \frac{\partial \phi_t^i}{\partial \hat{R}_{t,y}} & \frac{\partial \phi_t^i}{\partial \hat{R}_{t,\theta}} & \frac{\partial \phi_t^i}{\partial \hat{m}_{k,x}} & \frac{\partial \phi_t^i}{\partial \hat{m}_{k,y}} & \frac{\partial \phi_t^i}{\partial \hat{m}_{k,z}} \end{bmatrix} \quad (3.24)$$

$$= \begin{bmatrix} \frac{-(m_{i,x} - \hat{R}_{t,x})}{\sqrt{\hat{R}_{t,r}}} & \frac{-(m_{i,y} - \hat{R}_{t,y})}{\sqrt{\hat{R}_{t,r}}} & 0 & \frac{(m_{i,x} - \hat{R}_{t,x})}{\sqrt{\hat{R}_{t,r}}} & \frac{(m_{i,y} - \hat{R}_{t,y})}{\sqrt{\hat{R}_{t,r}}} & 0 \\ \frac{(m_{i,x} - \hat{R}_{t,x})}{\hat{R}_{t,r}} & \frac{-(m_{i,y} - \hat{R}_{t,y})}{\hat{R}_{t,r}} & -1 & \frac{-(m_{i,y} - \hat{R}_{t,y})}{\hat{R}_{t,r}} & \frac{(m_{i,x} - \hat{R}_{t,x})}{\hat{R}_{t,r}} & 0 \\ \frac{(m_{i,x} - \hat{R}_{t,x}) * (m_{i,z} - R_H)}{\sqrt{\hat{R}_{t,r}}(\hat{R}_{t,r} + (m_{i,z} - R_H)^2)} & \frac{(m_{i,y} - \hat{R}_{t,y}) * (m_{i,z} - R_H)}{\sqrt{\hat{R}_{t,r}}(\hat{R}_{t,r} + (m_{i,z} - R_H)^2)} & 0 & \frac{-(m_{i,x} - \hat{R}_{t,x}) * (m_{i,z} - R_H)}{\sqrt{\hat{R}_{t,r}}(\hat{R}_{t,r} + (m_{i,z} - R_H)^2)} & \frac{-(m_{i,y} - \hat{R}_{t,y}) * (m_{i,z} - R_H)}{\sqrt{\hat{R}_{t,r}}(\hat{R}_{t,r} + (m_{i,z} - R_H)^2)} & \frac{\sqrt{\hat{R}_{t,r}}}{\hat{R}_{t,r} + (m_{i,z} - R_H)^2} \end{bmatrix}$$

其中 $\hat{R}_{t,r} = (m_{i,x} - \hat{R}_{t,x})^2 + (m_{i,y} - \hat{R}_{t,y})^2$

$$F_{x,k} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \quad (3.25)$$

藉由上述各分量，計算 Kalman gain，其中 Q_t 是觀測之量測向量雜訊的共變異數矩陣：

$$K_t^i = \bar{\Sigma}_t H_t^{iT} (H_t^i \bar{\Sigma}_t H_t^{iT} + Q_t)^{-1} \quad (3.26)$$

以 Kalman gain 及 Innovation vector ($Z_t - \hat{Z}_t$) 更新系統狀態估測向量 $\bar{\mu}_t$

$$\bar{\mu}_t = \bar{\mu}_t + K_t^i (Z_t^i - \hat{Z}_t^i) \quad (3.27)$$

並更新系統之共變異數矩陣，

$$\bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t \quad (3.28)$$

至此，完成一次 EKF 演算法的計算，EKF 演算法是以疊代的方式計算估測之系統狀態向量，其中疊代的次數依 t 時刻影像特徵與資料庫特徵之共同特徵點

的總數來決定，最後輸出的資訊為系統狀態向量以及共變異數矩陣，整個流程圖可用表 3.1 呈現。

表 3.1 EKF SLAM 演算流程

EKF SLAM Algorithm ($\mu_{t-1}, \Sigma_{t-1}, u_t, Z_t$)	
Prediction step:	
1.	$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{bmatrix} -\frac{v_t}{w_t} \sin R_{t-1,\theta} + \frac{v_t}{w_t} \sin(R_{t-1,\theta} + w_t \Delta t) \\ \frac{v_t}{w_t} \cos R_{t-1,\theta} - \frac{v_t}{w_t} \cos(R_{t-1,\theta} + w_t \Delta t) \\ w_t \Delta t \end{bmatrix}$
2.	$G_t = I + F_x^T \begin{bmatrix} 0 & 0 & \frac{v_t}{w_t} (-\cos R_{t-1,\theta} + \cos(R_{t-1,\theta} + w_t \Delta t)) \\ 0 & 0 & \frac{v_t}{w_t} (-\sin R_{t-1,\theta} + \sin(R_{t-1,\theta} + w_t \Delta t)) \\ 0 & 0 & 0 \end{bmatrix} F_x$
3.	$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + F_x^T E_t F_x$
Update Step:	
4.	for all observed feature , Z_t^i
5.	if landmark i never seen before
6.	Add landmark to map
7.	endif
8.	$\hat{Z}_t^i = \begin{bmatrix} \sqrt{(\hat{m}_{k,x} - \hat{R}_{t,x})^2 + (\hat{m}_{k,y} - \hat{R}_{t,y})^2} \\ \text{atan2}(\hat{m}_{k,y} - \hat{R}_{t,y}, \hat{m}_{k,x} - \hat{R}_{t,x}) - \hat{R}_{t,\theta} \\ \text{atan}((\hat{m}_{k,z} - R_H) / \sqrt{(\hat{m}_{k,x} - \hat{R}_{t,x})^2 + (\hat{m}_{k,y} - \hat{R}_{t,y})^2}) \end{bmatrix}, \quad k = C_t^i$
9.	$H_t^i = h_t^i F_{x,k}$
10.	$K_t^i = \bar{\Sigma}_t H_t^{iT} (H_t^i \bar{\Sigma}_t H_t^{iT} + Q_t)^{-1}$
11.	$\bar{\mu}_t = \bar{\mu}_t + K_t^i (Z_t^i - \hat{Z}_t^i)$
12.	$\bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t$
13.	endfor
14.	$\mu_t = \bar{\mu}_t; \Sigma_t = \bar{\Sigma}_t$
15.	return μ_t, Σ_t

3.4 討論

在本論文中，計算影像特徵點在空間座標的資訊是以第二章的方式運行，其中在對於環境中距離較遠的特徵點上的處理方式，是將這些特徵點與以捨棄不使用，在這裡考慮的因素是因為這些特徵點其準確度較低，由 2.4 節中的實驗可以知道特徵點的資訊隨著距離越遠而越不準確，但是影像中距離較遠的特徵點在與機器人間的角度關係較為穩定，因此可以使用距離較遠的特徵點與機器人間的角度關係做定位系統演算對機器人朝向進行修正，對機器人定位的結果將有所幫助。



第四章 地圖接合

SLAM 的技術以環境資訊修正機器人狀態資訊，定位系統的演算需要盡可能快速才能即時修正誤差，標準之 EKF 演算法其運算複雜度約為 $O(n^2)$ ，其中 n 代表的是系統狀態向量的維度，運算複雜度實際的數值意義代表演算法運算次數的總合。因此，EKF 演算法中隨著資料庫越來越龐大，矩陣運算複雜度會嚴重影響定位系統在即時運算上的表現。

為了實現定位系統於機器人的即時應用，以降低系統向量維度大小以減輕運算，[3]以單眼攝影機作為感測裝置，其將機器人行進間所取得之影像分為數個群組，每一個群組以開頭的影像作為參考影像，接續者與參考影像比對，藉此取得特徵點於環境中的距離資訊，並以 EKF 定位系統修正機器人狀態，此做法以固定數量的影像做分界，確實的降低矩陣複雜度的增長。[18]提出以地圖的資訊做區界的概念，將世界地圖分為數個較小的部分，同樣能達到降低系統狀態向量之維度。本論文以將整體地圖劃分為數個較小部分的概念為基準，並參考[18][11]的做法，將所有區域地圖之資訊透過地圖融合的演算法結合成世界地圖，修正區域地圖間差異所產生之誤差，使 SLAM 於機器人應用上達成即時運算並保持地圖完整性。

4.1 區域地圖

本論文以機器人運行時所取得之影像特徵點作為環境資訊，這些特徵點如第二章所求，以影像與深度感測裝置 Kinect 的使用取得環境影像，計算影像中特徵點的三維座標資訊，所有接續的影像透過最鄰近演算法與地圖資料庫之特徵點進行比對，計算影像間相同的特徵點，能被重複比對的特徵點會做為定位系統的輸入，修正機器人狀態資訊，其餘的影像特徵點帶有距離資訊但不被定位系統所使用，此兩類訊息皆被統合入資料庫內，這個資料庫代表整個環境資訊，而因其代表整個環境資訊，故資料庫實質的意義即有“地圖”本質，因此本論文中，這些被整合的特徵點資料庫訊息即代表環境地圖。

而區域地圖的產生，其目的是為了解解決定位系統在運算複雜度上的問題，因定位系統所使用之環境資訊會隨著機器人行進而不斷增長，所需的運算成本也就越大，這造成機器人即時運算難以實行。與使用整個環境資訊不同，區域地圖的資訊雖僅代表整個環境中的一部分訊息，但其仍能給予機器人足夠的參考資料，使定位系統能正確修正機器人狀態，同時能達成機器人即時運算之應用。

區域地圖的形成，本論文是以機器人行徑之路徑範圍做為判斷依據，而設定範圍大小的依據是以 Kinect 感測裝置的感測範圍做考量，Kinect 的深度感測裝置依第二章距離感測精度的測試結果，我們可以知道感測裝置所量得的資訊隨著距離越遠而存在越大的誤差，為了得到較好量測結果我們將感測器所得到的有效量測距離限制在 0.7m-3.5m。並且區域地圖作為整個環境的部分資訊，兩兩地圖間的資訊須有所聯集，但超過兩區以上的地圖資訊應無關連，而又取樣區間的大小必須考量定位系統所需運算成本，因此，我們定義地圖分界的範圍，以機器人開啟區域地圖的位置為該區之原點，以該點機器人朝向為 Y 軸方向，與之水平垂直的座標軸為 X 軸，定義 3mX3m 之方形區域為區域地圖的建立範圍。圖 4.1 為建立區域地圖範圍之示意圖，假設其為整個世界地圖中第 i 個被設立的區域地圖，且此區域之原點於世界座標中的位置是 $(G_{i,x} \ G_{i,y} \ G_{i,\theta})$ ，橘色的方框為設定之區域邊界，當機器人超過邊界範圍，則結束此區域並開啟新的區域地圖，其中包覆橘色方框之灰色圓形，在本論文中其為判斷機器人選擇區域地圖之依據，稱其為區域路徑範圍，地圖選擇的方式於 4.4 節說明。

4.2 全域地圖

區域地圖是將環境資訊分為數個較小的部分，降低系統狀態向量維度，使定位系統能達到機器人應用之即時運算，而全域地圖則是統一特徵點資料庫訊息於相同座標系並連結各區域地圖之間的關聯性。

地圖融合是將兩張以上的區域地圖結合而成，參考論文[18]在地圖融合的設計上是依地圖數量以二分法的方式逐步將地圖資訊結合，如第二章述，圖 1.2

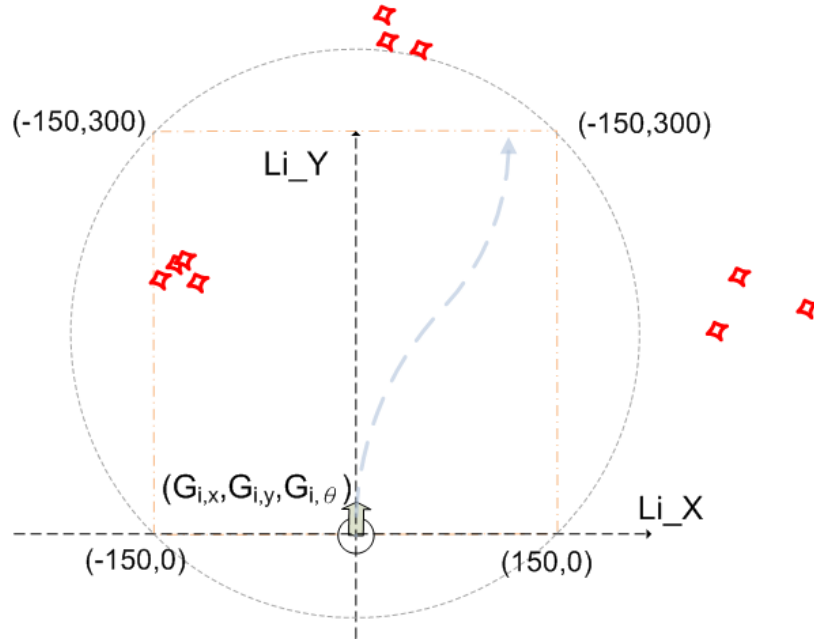


圖 4.1 區域地圖之示意圖

假設機器人完成運行時總共生成了 m 個區域地圖，接著以二分法的方式將 m 個區域地圖融合成總數為 $m/2$ 個資訊，再將 $m/2$ 個資訊結合為 $m/4$ 個，以此類推直到所有資訊備融合成一個完整的世界地圖。

而本論文所提出的做法，則假設當前區域地圖與相鄰地圖最具關聯性，其餘地圖資訊對當前區域地圖的影響極小，故每次僅以相鄰之區域地圖作融合。假設機器人完成運行共建立了 k 個區域地圖，則期間地圖融合演算的方式如圖 4.2 所示，機器人一開始所建立之區域地圖 M_1 其座標系與世界地圖座標系一致，其既是區域地圖也是世界地圖，當區域地圖完成建立(圖 4.2 中實線方框的部分)，每個區域建立完成時的機器人狀態是當前區域的結束位置亦是定義下一個區域座標系的起點位置，由這個新區域的起點資訊，以 4.1 節所描述的方式可以知道下一個區域所包含的範圍(圖 4.2 中虛線方框的部分)，根據新區域包含的範圍與地圖資料庫其他區域地圖的關聯性，判斷新的區域是否為重合區域(判斷區域是否為重合區域的方法在 4.3 節中描述)。根據重合區域偵測的結果記錄與新區域之相鄰的區域地圖，這些相鄰的區域地圖資訊會以地圖融合演算修正其間的差異(地圖融合演算法在 4.5 節中被描述)，整個地圖融合便是以這樣的方式不斷重複運行，直到第 k 個區域地圖完成整個地圖資訊的融合。

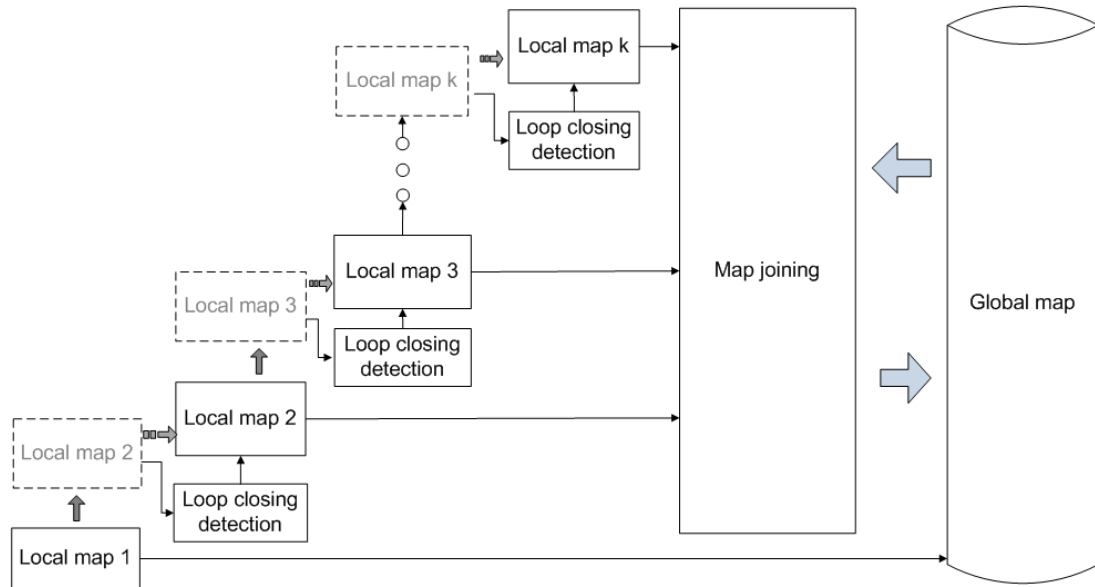


圖 4.2 區域地圖融合策略

4.3 重合事件偵測

區域地圖隨著機器人運行不斷建立，機器人若非以直線前進之方式運動，則機器人可能進入到較早時刻所建立之區域而進入區域重合的狀態(loop closure)，若系統持續建立地圖而無法使用已建立的地圖資訊，將會造成系統資源的浪費，並使整體地圖資訊缺乏完整性，因此，本論文以區域所構成的路徑範圍作為判斷依據，偵測重合事件(Loop closing)的發生。

區域所構成的路徑範圍是以 4.1 節所敘述的做法，這個範圍是以區域分界的邊界決定圓的大小，偵測重合區域的時間點是在系統完成區域地圖與建立下一區域的時刻，當機器人跨過所設定的區域邊界，系統將區域地圖完結並以地圖融合演算將之與更早的地圖資訊做結合，接著依機器人位置與朝向角定義下一區域地圖的範圍，重合區域偵測定義若新區域的中心位置落在其他已建立之地圖路徑範圍裡，則新建立的區域地圖為發生重合事件的區域，其影響是決定新區域在地圖融合演算法所使用的相鄰區域地圖，並在新區域完成融合演算法後，改變機器人使用地圖資料庫的方式（從建構地圖的模式改為使用已建立之地圖資訊），以圖 4.3 為範例，機器人以方形路徑移動，當機器人返回起始位置的時候，其創建之區域 Map6 的區域中心落在初始地圖 Map1 的路徑範圍內，故 Map6 為一重合區

域(圖 4.3 之路型範圍與重合區域的發生即為 5.5 節所展示之實驗)。

重合事件偵測對於定位與地圖建立的系統是轉換地圖建立模式與使用舊有地圖資料庫的關鍵點，而其影響地圖完整性與定位的準確性；本論文在重合事件偵測會判斷下一個區域是否與其他區域重合，當重合事件發生系統便能記錄與新區域相鄰的地圖序別，當新區域完成建構後地圖融合演算法才能使用正確的相鄰區域地圖資料計算，計算後的結果能夠使區域間的特徵點訊息得到更為正確的修正結果（與沒有重合事件偵測，不能得知除了前一次建立的區域地圖外其他相鄰的舊有地圖的地圖融合相比）；概率型機器人位置估測的方式，當相同的特徵點能夠被重複比對的次數越多時，越能確定機器人位置估測的正確性，相較於接近機器人起始的位置，機器人行走的距離越遠其位置估測的不確定性會逐漸提高，重合事件偵測判斷機器人是否走回以建立的區域中，當確定機器人在已建立的環境區域中便能使用舊有的環境資訊，定位系統能使用舊有的環境資訊能降低位置估測的不確定性。

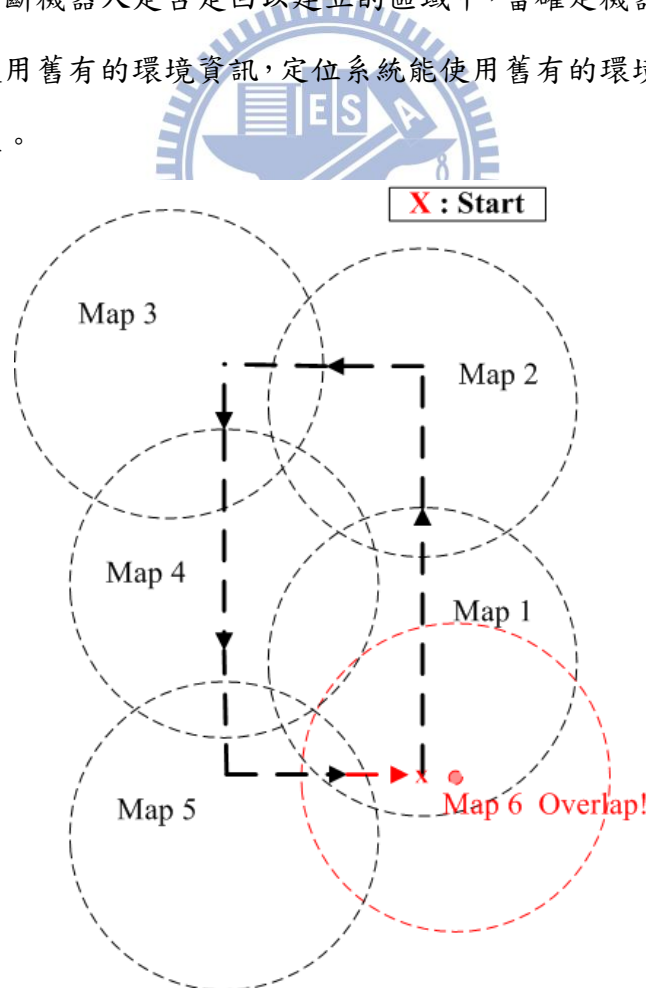


圖 4.3 重合區域偵測示意圖

4.4 地圖選擇

重合事件偵測的結果使機器人能得知新區域與已建立之區域有較大的範圍重疊，當新區域地圖完成建構後以地圖融合演算法統合區域間的訊息，此時我們認定機器人在已建立的環境中運行，機器人將使用已建立之區域地圖資料庫做定位系統的計算，直到機器人離開已建立之地圖區域範圍系統才會重新回到地圖建構的模式。

當機器人在已建立的環境中運行時所會面臨的一種情況是當進入區域重疊的範圍，在一般的情況下（機器人在非區域重疊的範圍），定位系統演算法使用與機器人位置對應之區域地圖，而選擇區域地圖的方式，其基本原則是依機器人位置與區域地圖的路徑範圍來決定，當機器人位置落於哪一個區域地圖的路徑範圍，即用該區域之地圖特徵點資訊，而當機器人位於區域地圖之間重疊範圍時如圖 4.4，則以當前影像特徵對區域特徵比對之數量作為區域轉換之條件，假設機器人位置由 Map2 進入 Map2 與 Map3 的重疊區域，則機器人在區域地圖上的選擇原則上是使用 Map2，但若特徵點比對數量小於設定的閾值，則會藉由比較當前影像特徵與 Map3 的共同特徵數量和當前影像特徵與 Map2 的共同特徵數量，若與 Map3 的共同特徵點數量較多，則使用的區域地圖轉換為 Map3，反之，則繼續使用 Map2。

區域地圖的選擇主要是根據系統在不同的階段而有所不同，圖 4.5 以流程圖的方式說明本論文所設計的系統在地圖選擇上的轉換，圖從定位以及地圖建構的部分出發，如果當前系統處於建構地圖的階段，根據機器人位置與區域地圖路徑範圍判定該區域是否已建立完成，如果未達到完成地圖所設定的地圖邊界，則系統持續擴建環境資訊並同時修正機器人狀態；若此時完成區域地圖建立，當前完成的區域地圖以地圖融合與相鄰的地圖資訊融合，而完成當前區域地圖時的機器人姿態可以得知下一新區域的路徑範圍，新區域的路徑範圍用來做重合區域的偵測，根據 4.3 節說明，重合區域偵測判斷新區域是否與已建立的區域地圖重合，

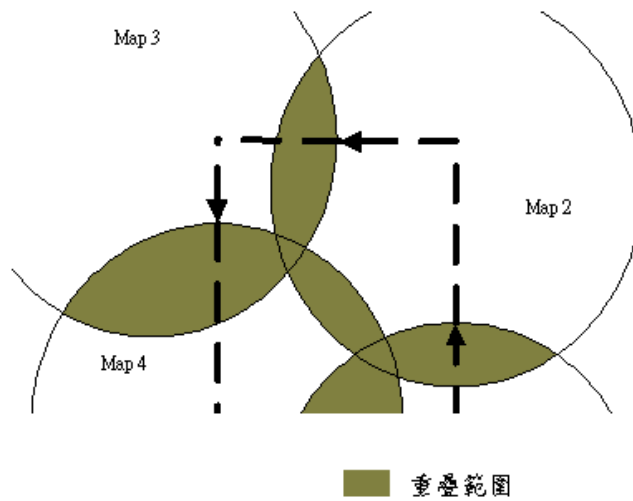


圖 4.4 區域路徑重疊範圍之示意圖

根據判斷的結果影響新區域在地圖融合演算法所使用的相鄰地圖。接著，若當前地圖為重合區域，則可以假設機器人將進入已建立之區域地圖的區域範圍，根據機器人當前的位置選擇對應的區域地圖。若當前位置為數個區域地圖重疊的地方，則依前段所敘述的方式，透過成功比對的特徵點數量來轉換區域地圖的使用，另外，在這個使用已建立之區域地圖的階段，若機器人運行間離開了已建立之區域地圖的範圍，則必須跳離使用舊有地圖的階段，重新回到建構地圖模式。若當前

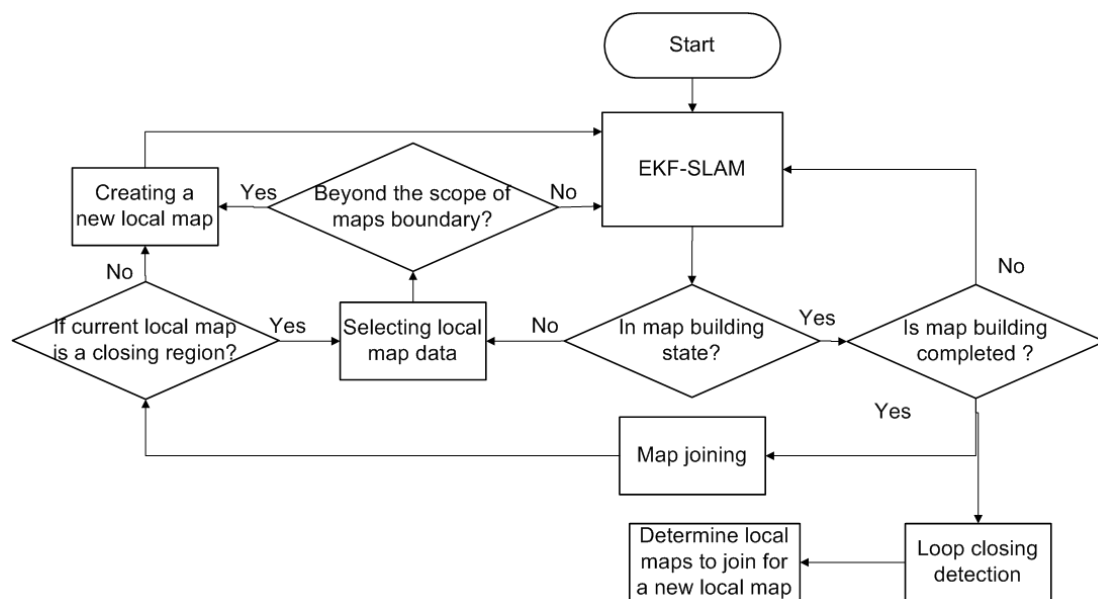


圖 4.5 地圖建構與選擇之流程圖

的區域地圖並非與已建立的區域地圖重合，則直接開啟下一個區域地圖，以定位

與地圖建構系統完成該地圖的建立。整個地圖建構的流程便是以這樣的方式不斷運行，而區域地圖的選擇的也依不同階段而必須選擇相對應的區域地圖。

4.5 地圖融合演算法

地圖融合演算法最主要的目的在於將分離的地圖資訊藉由區域地圖間共同的特徵點訊息使機器人狀態向量得到進一步的修正，並且修正區域地圖特徵之訊息，最終得到完整的地圖資訊。地圖融合之策略已於 4.2 節中說明，而 4.3 節中藉由區域路徑範圍的設定找尋重合之區域，並透過地圖融合演算法將相鄰之區域地圖結合，本節敘述地圖融合演算法，此做法參考自[11][18]之演算流程。

令兩相鄰的地圖資訊，在第一個地圖中機器人從世界座標 A 點走至座標 B 點，接著機器人開啟第二張區域地圖，而第二張區域地圖開始與結束的位置分別為世界座標 B 點以及座標 C 點，並令兩區域地圖之座標系皆以起始位置為座標系原點 (0,0)，以起始位置之機器人朝向角為座標系 Y 軸，與之水平垂直的座標系為 X 軸，機器人感測裝置高度為 R_H ，且機器人在區域地圖的初始狀態皆為 $R_{initial}$ 。

$$R_{initial} = \begin{bmatrix} R_{initial_X} \\ R_{initial_Y} \\ R_{initial_theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 90^\circ \end{bmatrix} \quad (4.1)$$

令第一張地圖的資訊為 $\hat{X}_{A..B}$ 該資訊包含從世界座標 A 點走到世界座標 B 點之機器人狀態向量 $\hat{X}_{R_A R_B}$ 以及區域內所有特徵點狀態向量量 $\hat{X}_{R_A N}$ ，且區域內特徵點總數為 k 個，而對應的系統共變異數矩陣為 $\hat{P}_{A..B}$

$$\hat{X}_{A..B} = \begin{bmatrix} \hat{X}_{R_A R_B} \\ \hat{X}_{R_A N} \end{bmatrix} = \begin{bmatrix} \hat{X}_{R_A R_B} \\ \hat{X}_{R_A N_1} \\ \hat{X}_{R_A N_2} \\ \vdots \\ \hat{X}_{R_A N_k} \end{bmatrix} \quad (4.2)$$

其中機器人狀態向量 $\hat{X}_{R_A R_B}$ ，其資訊包括二維座標向量以及朝向角，這個機器人狀態向量的資訊主要是標明機器人在區域一結束位置(B 點)的狀態：

$$\hat{X}_{R_A R_B} = \begin{bmatrix} \hat{X}_{R_A R_B, x} \\ \hat{X}_{R_A R_B, y} \\ \hat{X}_{R_A R_B, \theta} \end{bmatrix} \quad (4.3)$$

其中 $\hat{X}_{R_A N_i}$ 為第一個區域地圖中第 i 個特徵點的狀態向量，下標 R_A 代表所使用的座標系為以 A 為起點的第一個區域地圖，而此狀態向量包括特徵點於環境中的三維座標資訊：

$$\hat{X}_{R_A N_i} = \begin{bmatrix} \hat{Z}_{R_A N_i, x} \\ \hat{Z}_{R_A N_i, y} \\ \hat{Z}_{R_A N_i, z} \end{bmatrix} \quad (4.4)$$

令接續的第二張地圖資訊，其系統狀態向量為 $\hat{X}_{B...C}$ (式 4.5)，同樣的，該區域以 B 點為區域原點，系統狀態資訊包含了機器人狀態向量 $\hat{X}_{R_B R_C}$ 以及區域內所有特徵點狀態向量 $\hat{X}_{R_B M}$ ，且令特徵點總數為 g 個，而與此系統狀態向量對應的共變異數矩陣為 $\hat{P}_{B...C}$ 。

$$\hat{X}_{B...C} = \begin{bmatrix} \hat{X}_{R_B R_C} \\ \hat{X}_{R_B M_1} \\ \hat{X}_{R_B M_2} \\ \vdots \\ \hat{X}_{R_B M_g} \end{bmatrix} \quad (4.5)$$

機器人從世界座標 A 移動到位置 B 再從位置 B 走到位置 C ，兩區域間必然會有感測器共同觀測到的區域，假設兩區域具有 w 個相同的特徵點，對應系統狀態向量 $\hat{X}_{A...B}$ 與狀態向量 $\hat{X}_{B...C}$ 中，相同的特徵分別以 $n_1 \dots n_w$ 以及特徵點 $m_1 \dots m_w$ 表示，並且這些相同特徵皆與對應各自區域內的特徵點的數值相同，以第 i 個相同特徵點 n_i 而言，對應其區域資料庫特徵點的序列為 $k_n = C_N^i$ ，而特徵點 m_i 對應其區域資料庫特徵點的序列為 $k_m = C_M^i$ 。

地圖融合主要分為三個步驟：1.合併狀態向量，2.更新步驟和 3.座標系轉換。分別說明如下。

1. 合併狀態向量：

地圖融合後之資訊為機器人從 A 點走至 C 點的資訊，令 $\hat{X}_{A...C}^-$ 為結合 $\hat{X}_{A...B}$ 以及 $\hat{X}_{B...C}$ 之系統狀態向量， $\hat{P}_{A...C}^-$ 為對應之共變異數矩陣，上標負號表達的意思是當前結合的系統狀態向量與系統共變異數矩陣未完成地圖融合演算，而共變異數矩陣在兩地圖間的關聯性為 0 是假設兩區域地圖為獨立，在建立區域地圖的階段兩區域地圖的資訊是以各自的座標系完成地圖建構，此假設會使合併結果略與實際有所差異，但能降低融合所需運算。

$$\hat{X}_{A...C}^- = \begin{bmatrix} \hat{X}_{A...B} \\ \hat{X}_{B...C} \end{bmatrix} \quad (4.6)$$

$$\hat{P}_{A...C}^- = \begin{bmatrix} \hat{P}_{A...B} & 0 \\ 0 & \hat{P}_{A...C} \end{bmatrix} \quad (4.7)$$

2. 更新步驟：

首先定義不同區域的特徵點轉換座標系的運算，其運算符號以及轉換關係式依[11]所定義，假設區域二內第 i 個特徵點 M_i 轉換至區域一的座標系，令該行為所使用的運算符號為 $\hat{X}_{R_A R_B} \oplus \hat{X}_{R_B M_i}$ 而此運算的轉換關係式為 4.8 式，其間的關係可由圖 4.6 表示。

$$\begin{bmatrix} \hat{Z}_{R_A M_i, x} \\ \hat{Z}_{R_A M_i, y} \\ \hat{Z}_{R_A M_i, z} \end{bmatrix} = \hat{X}_{R_A R_B} \oplus \hat{X}_{R_B M_i}$$

$$= \begin{bmatrix} \hat{X}_{R_A R_B, x} + \hat{Z}_{R_B M_i, x} * \cos(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) - \hat{Z}_{R_B M_i, y} * \sin(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) \\ \hat{X}_{R_A R_B, y} + \hat{Z}_{R_B M_i, x} * \sin(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) + \hat{Z}_{R_B M_i, y} * \cos(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) \\ \hat{Z}_{R_B M_i, z} \end{bmatrix} \quad (4.8)$$

因為區域地圖的座標軸設定是以開啟區域地圖時的機器人朝向角為 Y 座標軸，因此將區域二中的特徵點轉換至區域一的座標系下的座標軸旋轉角度，是以區域一結束時的機器人朝向角(亦為區域二的座標系 Y 軸)與區域一座標系 Y 軸的夾角，因此 $\hat{X}_{R_A, \theta}$ 定義為機器人在區域一(以 A 點為初始的座標系)之初始機器

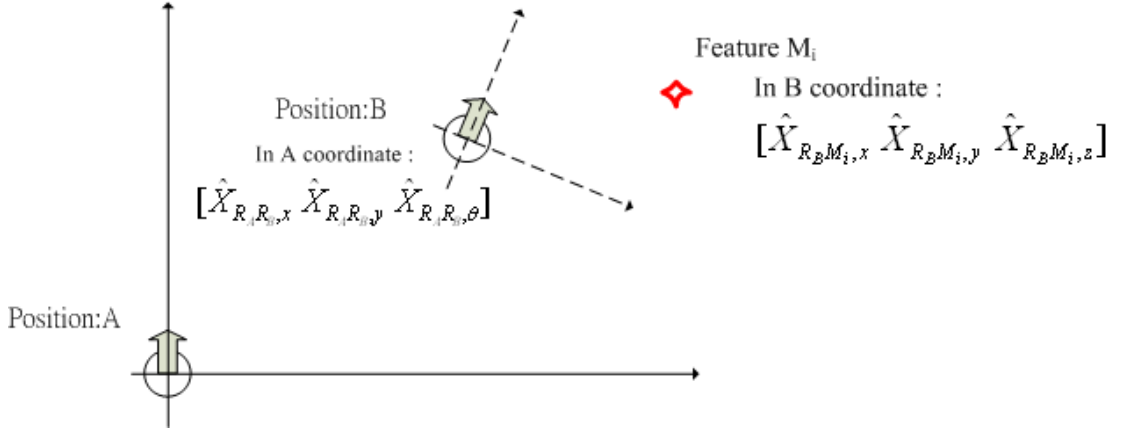


圖 4.6 區域一區域二與共同特徵點的關係圖

人朝向角定義為 90° 。

區域 1 和區域 2 為相鄰的兩區域地圖，其有 w 個共同特徵點，定義 $h_H(\hat{X}_{A...C}^-)$ 為兩區域共同特徵在區域一座標系下的差值向量如式 4.9

$$h_H(\hat{X}_{A...C}^-) = \begin{bmatrix} \hat{h}_{n_1, m_1} \\ \vdots \\ \hat{h}_{n_w, m_w} \end{bmatrix} \quad (4.9)$$

差值向量 $h_H(\hat{X}_{A...C}^-)$ 裡的元素是共同特徵在同一座標系下所計算得的差值資訊，令 $\hat{h}_{n_i m_i}$ 為第 i 個相同特徵點在區域一座標系下計算而得的差值向量，該向量包括在二維平面上特徵點至座標起點的直線距離差值 $\hat{h}_{n_i m_i, r}$ ，二維平面上特徵點至原點與 Y 軸的夾角差值 $\hat{h}_{n_i m_i, \theta}$ 以及特徵點至原點的垂直夾角差值 $\hat{h}_{n_i m_i, \phi}$ 。

$$h_{n_i, m_i} = \hat{X}_{R_A n_i} - \hat{X}_{R_A R_B} \oplus \hat{X}_{R_B m_i} = \begin{bmatrix} \hat{h}_{n_i, m_i, r} \\ \hat{h}_{n_i, m_i, \theta} \\ \hat{h}_{n_i, m_i, \phi} \end{bmatrix}$$

$$= \begin{bmatrix} (\hat{Z}_{R_A n_i, x}^2 + \hat{Z}_{R_A n_i, y}^2)^{1/2} - (\hat{Z}_{R_A m_i, x}^2 + \hat{Z}_{R_A m_i, y}^2)^{1/2} \\ -a \tan(\hat{Z}_{R_A n_i, x} / \hat{Z}_{R_A n_i, y}) - (-a \tan(\hat{Z}_{R_A m_i, x} / \hat{Z}_{R_A m_i, y})) \\ a \tan \hat{Z}_{R_A n_i, z} / \hat{Z}_{R_A n_i, r} - a \tan \hat{Z}_{R_A m_i, z} / \hat{Z}_{R_A m_i, r} \end{bmatrix} \quad (4.10)$$

其中，為了簡化運算式定義 $\hat{Z}_{R_A n_i, r}$ 以及 $\hat{Z}_{R_A m_i, r}$ 如下：

$$\hat{Z}_{R_A n_i, r} = (\hat{Z}_{R_A n_i, x}^2 + \hat{Z}_{R_A n_i, y}^2)^{1/2} \quad (4.11)$$

$$\hat{Z}_{R_A m_i, r} = (\hat{Z}_{R_A m_i, x}^2 + \hat{Z}_{R_A m_i, y}^2)^{1/2} \quad (4.12)$$

理想上，兩區域地圖共同特徵點之三維座標應完全相同：

$$h_H(\hat{X}_{A..C}^-) = 0 \quad (4.13)$$

將 $h_H(\hat{X}_{A..C}^-)$ 以一階泰勒展開於 $\hat{X}_{A..C}^-$ ，得到近似線性方程 H_h ，其中 $X_{A..C}^-$ 表示為未完成地圖融合演算的系統狀態向量變數，與 $\hat{X}_{A..C}^-$ 為已知的系統狀態不同。

$$h_H(X_{A..C}^-) \cong h_H(\hat{X}_{A..C}^-) + H_h(X_{A..C}^- - \hat{X}_{A..C}^-) \quad (4.14)$$

H_h 為差值向量對系統狀態向量偏導數之結果，其中 $\frac{\partial h_{n_i m_i}}{\partial x_{R_A R_B}}$ 表示第 i 個差值向量 $h_{n_i m_i}$ 在系統狀態 $\hat{X}_{A..C}^-$ 下對機器人狀態向量變數 $x_{R_A R_B}$ 偏導數的簡化表示式，其他的依此類推：

$$H_h = \left. \frac{\partial h_H}{\partial X_{A..C}^-} \right|_{(\hat{X}_{A..C}^-)}$$

$$= \begin{bmatrix} \frac{\partial h_{n_1 m_1}}{\partial \hat{X}_{R_A R_B}} & \cdots & \frac{\partial h_{n_1 m_1}}{\partial \hat{X}_{R_A N_k}} & \frac{\partial h_{n_1 m_1}}{\partial \hat{X}_{R_B R_C}} & \frac{\partial h_{n_1 m_1}}{\partial \hat{X}_{R_B M_1}} & \cdots & \frac{\partial h_{n_1 m_1}}{\partial \hat{X}_{R_B M_g}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_{n_w m_w}}{\partial \hat{X}_{R_A R_B}} & \cdots & \frac{\partial h_{n_w m_w}}{\partial \hat{X}_{R_A N_k}} & \frac{\partial h_{n_w m_w}}{\partial \hat{X}_{R_B R_C}} & \frac{\partial h_{n_w m_w}}{\partial \hat{X}_{R_B M_1}} & \cdots & \frac{\partial h_{n_w m_w}}{\partial \hat{X}_{R_B M_g}} \end{bmatrix} \quad (4.15)$$

共同特徵點差值向量對系統狀態向量做偏導數，其中具有意義的分項主要有三個部分，其餘的結果則為0矩陣。

特徵點狀態向量與機器人狀態向量 $x_{R_A R_B}$ 做偏導數的結果如4.16式。(以第 i 個差值向量 $h_{n_i m_i}$ 對 $\hat{X}_{R_A R_B}$ 偏微分為例)

$$\left. \frac{\partial h_{n_i m_i}}{\partial x_{R_A R_B}} \right|_{(\hat{X}_{A..C}^-)} = \begin{bmatrix} \frac{\partial h_{n_i m_i, r}}{\partial \hat{X}_{R_A R_B, x}} & \frac{\partial h_{n_i m_i, r}}{\partial \hat{X}_{R_A R_B, y}} & \frac{\partial h_{n_i m_i, r}}{\partial \hat{X}_{R_A R_B, \theta}} \\ \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{X}_{R_A R_B, x}} & \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{X}_{R_A R_B, y}} & \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{X}_{R_A R_B, \theta}} \\ \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{X}_{R_A R_B, x}} & \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{X}_{R_A R_B, y}} & \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{X}_{R_A R_B, \theta}} \end{bmatrix} \quad (4.16)$$

$$= \begin{bmatrix} -\frac{\hat{Z}_{R_A m_i, x}}{\hat{Z}_{R_A m_i, r}} & -\frac{\hat{Z}_{R_A m_i, y}}{\hat{Z}_{R_A m_i, r}} & -\frac{\hat{Z}_{R_A m_i, x} * \hat{Z}_{R_A R_B, y} - \hat{Z}_{R_A m_i, y} * \hat{Z}_{R_A R_B, x}}{\hat{Z}_{R_A m_i, r}} \\ \frac{\hat{Z}_{R_A m_i, y}}{\hat{Z}_{R_A m_i, r}^2} & -\frac{\hat{Z}_{R_A m_i, x}}{\hat{Z}_{R_A m_i, r}^2} & \frac{\hat{Z}_{R_A R_B, y} * \hat{Z}_{R_A m_i, y} + \hat{Z}_{R_A R_B, x} * \hat{Z}_{R_A m_i, x} - \hat{Z}_{R_A m_i, r}^2}{\hat{Z}_{R_A m_i, r}^2} \\ \frac{\hat{Z}_{R_A m_i, x} * (\hat{Z}_{R_A m_i, z} - R_H)}{\hat{Z}_{R_A m_i, D}^2 * \hat{Z}_{R_A m_i, r}} & \frac{\hat{Z}_{R_A m_i, y} * (\hat{Z}_{R_A m_i, z} - R_H)}{\hat{Z}_{R_A m_i, D}^2 * \hat{Z}_{R_A m_i, r}} & \frac{(\hat{Z}_{R_A m_i, z} - R_H) * (\hat{Z}_{R_A m_i, x} * \hat{Z}_{R_A R_B, y} - \hat{Z}_{R_A m_i, y} * \hat{Z}_{R_A R_B, x})}{\hat{Z}_{R_A m_i, D}^2 * \hat{Z}_{R_A m_i, r}} \end{bmatrix}$$

其中，為了簡化運算式定義 $\hat{Z}_{R_A m_i, D}$ 如下：

$$\hat{Z}_{R_A m_i, D} = \sqrt{(\hat{Z}_{R_A m_i, z} - R_H)^2 + \hat{Z}_{R_A m_i, x}^2 + \hat{Z}_{R_A m_i, y}^2} \quad (4.17)$$

第二項為差值向量 $h_{n_i m_i}$ 對區域一的特徵狀態向量 $x_{R_A N}$ 做偏微分的結果，其僅與相同特徵之偏導數有意義，令變數 k_n 為共同特徵差值 $h_{n_i m_i}$ 對應於區域一中相同的特徵點序列， $k_n = C_N^i$ 。

$$\frac{\partial h_{n_i m_i}}{\partial x_{R_A N k_n}} \Big|_{(\hat{x}_{A \dots C})} = \begin{bmatrix} \frac{\partial h_{n_i m_i, r}}{\partial \hat{Z}_{R_A N k_n, x}} & \frac{\partial h_{n_i m_i, r}}{\partial \hat{Z}_{R_A N k_n, y}} & \frac{\partial h_{n_i m_i, r}}{\partial \hat{Z}_{R_A N k_n, z}} \\ \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{Z}_{R_A N k_n, x}} & \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{Z}_{R_A N k_n, y}} & \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{Z}_{R_A N k_n, z}} \\ \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{Z}_{R_A N k_n, x}} & \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{Z}_{R_A N k_n, y}} & \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{Z}_{R_A N k_n, z}} \end{bmatrix} = \begin{bmatrix} \frac{\hat{Z}_{R_A n_i, x}}{\hat{Z}_{R_A n_i, r}} & \frac{\hat{Z}_{R_A n_i, y}}{\hat{Z}_{R_A n_i, r}} & 0 \\ -\frac{\hat{Z}_{R_A n_i, y}}{\hat{Z}_{R_A n_i, r}^2} & \frac{\hat{Z}_{R_A n_i, x}}{\hat{Z}_{R_A n_i, r}^2} & 0 \\ -\frac{\hat{Z}_{R_A n_i, x} * (\hat{Z}_{R_A n_i, z} - R_H)}{\hat{Z}_{R_A n_i, D}^2 * \hat{Z}_{R_A n_i, r}} & -\frac{\hat{Z}_{R_A n_i, y} * (\hat{Z}_{R_A n_i, z} - R_H)}{\hat{Z}_{R_A n_i, D}^2 * \hat{Z}_{R_A n_i, r}} & \frac{\hat{Z}_{R_A n_i, r}}{\hat{Z}_{R_A n_i, D}^2} \end{bmatrix} \quad (4.18)$$

第三項為共同特徵差值向量 $h_{n_i m_i}$ 對區域二的特徵狀態向量 $x_{R_B M}$ 做偏導數，同樣的，其僅與相同特徵之偏導數有意義，令變數 k_m 為第 i 個共同特徵差值向量 $h_{n_i m_i}$ 對應於區域二中相同的特徵點序列， $k_m = C_M^i$

$$\frac{\partial h_{n_i m_i}}{\partial x_{R_B M k_m}} \Big|_{(\hat{x}_{A \dots C})} = \begin{bmatrix} \frac{\partial h_{n_i m_i, r}}{\partial \hat{Z}_{R_B M k_m, x}} & \frac{\partial h_{n_i m_i, r}}{\partial \hat{Z}_{R_B M k_m, y}} & \frac{\partial h_{n_i m_i, r}}{\partial \hat{Z}_{R_B M k_m, z}} \\ \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{Z}_{R_B M k_m, x}} & \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{Z}_{R_B M k_m, y}} & \frac{\partial h_{n_i m_i, \theta}}{\partial \hat{Z}_{R_B M k_m, z}} \\ \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{Z}_{R_B M k_m, x}} & \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{Z}_{R_B M k_m, y}} & \frac{\partial h_{n_i m_i, \phi}}{\partial \hat{Z}_{R_B M k_m, z}} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{Abbr1}{\hat{Z}_{R_A M_i, r}} & -\frac{Abbr2}{\hat{Z}_{R_A M_i, r}} & 0 \\ \frac{Abbr2}{\hat{Z}_{R_A M_i, r}^2} & -\frac{Abbr1}{\hat{Z}_{R_A M_i, r}^2} & 0 \\ \frac{(\hat{Z}_{R_A M_i, z} - R_H) * Abbr1}{\hat{Z}_{R_A M_i, D}^2 * \hat{Z}_{R_A M_i, r}} & \frac{(\hat{Z}_{R_A M_i, z} - R_H) * Abbr2}{\hat{Z}_{R_A M_i, D}^2 * \hat{Z}_{R_A M_i, r}} & -\frac{\hat{Z}_{R_A M_i, r}}{\hat{Z}_{R_A M_i, D}^2} \end{bmatrix} \quad (4.19)$$

其中,為了簡化運算式令 Abbr1 以及 Abbr2 如下:

$$Abbr1 = \hat{Z}_{R_A M_i, x} * \cos(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) + \hat{Z}_{R_A M_i, y} * \sin(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) \quad (4.20)$$

$$Abbr2 = -\hat{Z}_{R_A M_i, x} * \sin(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) + \hat{Z}_{R_A M_i, y} * \cos(\hat{X}_{R_A R_B, \theta} - \hat{X}_{R_A, \theta}) \quad (4.21)$$

由上述推導結果,可求得 Kalman Gain:

$$K = P_{A..C}^- H_H^T (H_H P_{A..C}^- H_H^T)^{-1} \quad (4.22)$$

利用 Kalman gain 以及共特徵點差值向量更新融合的系統狀態向量:

$$\hat{X}_{A..C}^+ = \hat{X}_{A..C}^- - K h_H (\hat{X}_{A..C}^-) \quad (4.23)$$

更新系統之共變異數矩陣:

$$P_{A..C}^+ = (I - K H_H) P_{A..C}^- \quad (4.24)$$

得到地圖融合後的機器人狀態資訊 $\hat{X}_{A..C}^+$ 以及共變異數矩陣 $P_{A..C}^+$, 其中上標+號代表此資訊為經過地圖融合演算法後之結果。

3. 座標系轉換:

完成地圖的資訊融合後,還必要做一個座標系轉換的步驟,因為在地圖融合階段時,系統狀態資訊是由兩個不同座標系下的資訊所構成,在這個步驟,將所有座標系統以第一個座標系為基準,將區域二特徵點之三維座標轉換至區域一的座標系中,完成機器人系統狀態向量從世界座標 A 點走至世界座標 C 點的地圖融合。

地圖融合演算法最終獲得的系統狀態向量 $X_{A..C}$ 是機器人從世界座標 A 點移動到世界座標 C 點的機器人狀態資訊 $X_{R_A R_C}$ 以及行進間所有得到的特徵點三維座標資訊所構成,更新步驟將兩區域地圖間的差異進行修正,經過更新步驟後的機器人狀態資訊以及特徵點三維座標,透過座標轉換的運算,將系統狀態資訊內的

元素全部轉換為以區域一的座標系為基準。

$$X_{A...C} = \begin{bmatrix} X_{R_A R_C} \\ X_{R_A N_1} \\ \vdots \\ X_{R_A N_k} \\ X_{R_A M_1} \\ \vdots \\ X_{R_A M_g} \end{bmatrix} = \begin{bmatrix} \hat{X}_{R_A R_B}^+ \oplus \hat{X}_{R_B R_C}^+ \\ \hat{X}_{R_A N_1}^+ \\ \vdots \\ \hat{X}_{R_A N_k}^+ \\ \hat{X}_{R_A R_B}^+ \oplus \hat{X}_{R_B M_1}^+ \\ \vdots \\ \hat{X}_{R_A R_B}^+ \oplus \hat{X}_{R_B M_g}^+ \end{bmatrix} \quad (4.25)$$

共變異矩陣透過轉換矩陣的運算得到最終的結果：

$$P_{A...C} = \frac{\partial X_{A...C}}{\partial \hat{X}_{A...C}^+} P_{A...C}^+ \left(\frac{\partial X_{A...C}}{\partial \hat{X}_{A...C}^+} \right)^T \quad (4.26)$$

轉換矩陣的計算為經過座標轉換後的系統狀態向量對轉換前的系統狀態向量做偏微分的結果，其運算如 4.27 式，其中偏導數有意義的部分於 4.28 到 4.31 分別計算其偏導數的結果：

$$\frac{\partial X_{A...C}}{\partial \hat{X}_{A...C}^+} = \begin{bmatrix} \frac{\partial X_{R_A R_C}}{\partial \hat{X}_{R_A R_B}^+} & 0 & \frac{\partial X_{R_A R_C}}{\partial \hat{X}_{R_B R_C}^+} & 0 \\ 0 & I & 0 & 0 \\ \frac{\partial X_{R_A M}}{\partial \hat{X}_{R_A R_B}^+} & 0 & 0 & \frac{\partial X_{R_A M}}{\partial \hat{X}_{R_B M}^+} \end{bmatrix} \quad (4.27)$$

其中區域一中特徵點資訊對其相同座標系之相同特徵之偏導數的結果為單位向量矩陣 I。

融合後的機器人狀態向量 $X_{R_A R_C}$ 對區域一的機器人狀態向量 $X_{R_A R_B}$ 作偏導數的結果如下：

$$\frac{\partial X_{R_A R_C}}{\partial \hat{X}_{R_A R_B}^+} = \begin{bmatrix} \frac{\partial X_{R_A R_C, x}}{\partial \hat{X}_{R_A R_B, x}^+} & \frac{\partial X_{R_A R_C, x}}{\partial \hat{X}_{R_A R_B, y}^+} & \frac{\partial X_{R_A R_C, x}}{\partial \hat{X}_{R_A R_B, \theta}^+} \\ \frac{\partial X_{R_A R_C, y}}{\partial \hat{X}_{R_A R_B, x}^+} & \frac{\partial X_{R_A R_C, y}}{\partial \hat{X}_{R_A R_B, y}^+} & \frac{\partial X_{R_A R_C, y}}{\partial \hat{X}_{R_A R_B, \theta}^+} \\ \frac{\partial X_{R_A R_C, \theta}}{\partial \hat{X}_{R_A R_B, x}^+} & \frac{\partial X_{R_A R_C, \theta}}{\partial \hat{X}_{R_A R_B, y}^+} & \frac{\partial X_{R_A R_C, \theta}}{\partial \hat{X}_{R_A R_B, \theta}^+} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -X_{R_B R_C, x} * \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) - X_{R_B R_C, y} * \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) \\ 0 & 1 & X_{R_B R_C, x} * \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) - X_{R_B R_C, y} * \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) \\ 0 & 0 & 1 \end{bmatrix} \quad (4.28)$$

融合後的機器人狀態資訊 $X_{R_A R_C}$ 對區域二的機器人狀態向量 $X_{R_B R_C}$ 做偏導數：

$$\begin{aligned} \frac{\partial X_{R_A R_C}}{\partial \hat{X}_{R_B R}^+} &= \begin{bmatrix} \frac{\partial X_{R_A R_C, x}}{\partial \hat{X}_{R_B R_C, x}^+} & \frac{\partial X_{R_A R_C, x}}{\partial \hat{X}_{R_B R_C, y}^+} & \frac{\partial X_{R_A R_C, x}}{\partial \hat{X}_{R_B R_C, \theta}^+} \\ \frac{\partial X_{R_A R_C, y}}{\partial \hat{X}_{R_B R_C, x}^+} & \frac{\partial X_{R_A R_C, y}}{\partial \hat{X}_{R_B R_C, y}^+} & \frac{\partial X_{R_A R_C, y}}{\partial \hat{X}_{R_B R_C, \theta}^+} \\ \frac{\partial X_{R_A R_C, \theta}}{\partial \hat{X}_{R_B R_C, x}^+} & \frac{\partial X_{R_A R_C, \theta}}{\partial \hat{X}_{R_B R_C, y}^+} & \frac{\partial X_{R_A R_C, \theta}}{\partial \hat{X}_{R_B R_C, \theta}^+} \end{bmatrix} \\ &= \begin{bmatrix} \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) & -\sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) & 0 \\ \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) & \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (4.29)$$

轉換至 A 座標系的特徵點 $X_{R_A M_i}$ 對區域一的機器人狀態向量 $X_{R_A R_B}$ 微分：

$$\begin{aligned} \frac{\partial X_{R_A M_i}}{\partial X_{R_A R}^+} &= \begin{bmatrix} \frac{\partial X_{R_A M_i, x}}{\partial X_{R_A R_B, x}^+} & \frac{\partial X_{R_A M_i, x}}{\partial X_{R_A R_B, y}^+} & \frac{\partial X_{R_A M_i, x}}{\partial X_{R_A R_B, \theta}^+} \\ \frac{\partial X_{R_A M_i, y}}{\partial X_{R_A R_B, x}^+} & \frac{\partial X_{R_A M_i, y}}{\partial X_{R_A R_B, y}^+} & \frac{\partial X_{R_A M_i, y}}{\partial X_{R_A R_B, \theta}^+} \\ \frac{\partial X_{R_A M_i, \theta}}{\partial X_{R_A R_B, x}^+} & \frac{\partial X_{R_A M_i, \theta}}{\partial X_{R_A R_B, y}^+} & \frac{\partial X_{R_A M_i, \theta}}{\partial X_{R_A R_B, \theta}^+} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -Z_{R_B M_i, x} * \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) - Z_{R_B M_i, y} * \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) \\ 0 & 1 & Z_{R_B M_i, x} * \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) - Z_{R_B M_i, y} * \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (4.30)$$

轉換至 A 座標系之特徵點 M_i 對區域二中相同特徵做偏導數

$$\begin{aligned} \frac{\partial X_{R_A M_i}}{\partial \hat{X}_{R_B M_i}^+} &= \begin{bmatrix} \frac{\partial X_{R_A M_i, x}}{\partial \hat{X}_{R_B M_i, x}^+} & \frac{\partial X_{R_A M_i, x}}{\partial \hat{X}_{R_B M_i, y}^+} & \frac{\partial X_{R_A M_i, x}}{\partial \hat{X}_{R_B M_i, z}^+} \\ \frac{\partial X_{R_A M_i, y}}{\partial \hat{X}_{R_B M_i, x}^+} & \frac{\partial X_{R_A M_i, y}}{\partial \hat{X}_{R_B M_i, y}^+} & \frac{\partial X_{R_A M_i, y}}{\partial \hat{X}_{R_B M_i, z}^+} \\ \frac{\partial X_{R_A M_i, \theta}}{\partial \hat{X}_{R_B M_i, x}^+} & \frac{\partial X_{R_A M_i, \theta}}{\partial \hat{X}_{R_B M_i, y}^+} & \frac{\partial X_{R_A M_i, \theta}}{\partial \hat{X}_{R_B M_i, z}^+} \end{bmatrix} \\ &= \begin{bmatrix} \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) & -\sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) & 0 \\ \sin(X_{R_A R_B, \theta} - X_{R_A, \theta}) & \cos(X_{R_A R_B, \theta} - X_{R_A, \theta}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (4.31)$$

至此，完成地圖融合演算法，地圖融合演算法之流程如表 4.1

表 4.1 地圖融合演算法之流程

Map Joining Algorithm ($\hat{X}_{A...B}$ $\hat{X}_{B...C}$ $\hat{P}_{A...B}$ $\hat{P}_{B...C}$)	
The map joining step :	
1.	$\hat{X}_{A...C}^- = \begin{bmatrix} \hat{X}_{A...B} \\ \hat{X}_{B...C} \end{bmatrix}$
2.	$\hat{P}_{A...C}^- = \begin{bmatrix} \hat{P}_{A...B} & 0 \\ 0 & \hat{P}_{B...C} \end{bmatrix}$
The update step :	
3.	$h_H(\hat{X}_{A...C}^-) = \begin{bmatrix} h_{n_1, m_1} \\ \vdots \\ h_{n_w, m_w} \end{bmatrix}$
4.	$h_H(\hat{X}_{A...C}) \cong h_{n_r}(\hat{X}_{A...C}^-) + H_h(X_{A...C}^- - \hat{X}_{A...C}^-)$
5.	$H_h = \begin{bmatrix} \frac{\partial h_{n_1, m_1}}{\partial \hat{X}_{R_A R_B}} & \cdots & \frac{\partial h_{n_1, m_1}}{\partial \hat{X}_{R_A N_k}} & \frac{\partial h_{n_1, m_1}}{\partial \hat{X}_{R_B R_C}} & \frac{\partial h_{n_1, m_1}}{\partial \hat{X}_{R_B M_1}} & \cdots & \frac{\partial h_{n_1, m_1}}{\partial \hat{X}_{R_B M_g}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_{n_w, m_w}}{\partial \hat{X}_{R_A R_B}} & \cdots & \frac{\partial h_{n_w, m_w}}{\partial \hat{X}_{R_A N_k}} & \frac{\partial h_{n_w, m_w}}{\partial \hat{X}_{R_B R_C}} & \frac{\partial h_{n_w, m_w}}{\partial \hat{X}_{R_B M_1}} & \cdots & \frac{\partial h_{n_w, m_w}}{\partial \hat{X}_{R_B M_g}} \end{bmatrix}$
6.	$K = P_{A...C}^- H_H^T (H_H P_{A...C}^- H_H^T)^{-1}$
7.	$\hat{X}_{A...C}^+ = \hat{X}_{A...C}^- - K h_H(\hat{X}_{A...C}^-)$
8.	$P_{A...C}^+ = (I - K H_H) P_{A...C}^-$
The transformation step :	
9.	$X_{A...C} = \begin{bmatrix} X_{R_A R_C} \\ X_{R_A N_1} \\ \vdots \\ X_{R_A N_k} \\ X_{R_A M_1} \\ \vdots \\ X_{R_A M_g} \end{bmatrix} = \begin{bmatrix} \hat{X}_{R_A R_B}^+ \oplus \hat{X}_{R_B R_C}^+ \\ \hat{X}_{R_A N}^+ \\ \vdots \\ \hat{X}_{R_A N}^+ \\ \hat{X}_{R_A R_B}^+ \oplus \hat{X}_{R_B M_1}^+ \\ \vdots \\ \hat{X}_{R_A R_B}^+ \oplus \hat{X}_{R_B M_g}^+ \end{bmatrix}$
10.	$P_{A...C} = \frac{\partial X_{A...C}}{\partial \hat{X}_{A...C}^+} P_{A...C}^+ \left(\frac{\partial X_{A...C}}{\partial \hat{X}_{A...C}^+} \right)^T$
11.	Return $X_{A...C}$ $P_{A...C}$

4.6 討論

在同時定位以及建立地圖的過程中，所建立出來的地圖資訊其準確度與機器人起點的距離成近似正比的關係，機器人行走得越遠所產生的資訊誤差會比一開始的資訊來的多。在本論文中，區域地圖建立的方式是以預先設定的大小作為區域的範圍，隨著機器人的運行而逐步建立區域地圖，其中越早建立的區域地圖其世界座標下的準確度越高，而因為區域地圖在建立的過程是獨立的，且單一區域建立時的誤差並不會太大，透過將前一個地圖資訊和當前建立完成的地圖資訊以地圖融合演算的方式修正的結果，能讓序列較後面建立的區域地圖往較準確的結果修正，但是修正後的結果實質上仍存在著誤差而無法保證區域融合後的最終資訊是絕對準確的。當區域重和的事件發生時，以圖 4.3 為例，Map6 為重合區域而 Map5 以及 Map1 是與 Map6 相鄰的區域地圖，以實驗的結果得知，經過地圖融合演算法後 Map6 的地圖資訊會往 Map1 拉近，能夠大幅降低最後的區域地圖在世界座標系下的誤差，但是為了降低地圖融合階段所消耗的運算時間，以相鄰地圖資訊做融合的作法，使得在地圖融合階段被修正的只有與重合區域相鄰的區域。

第五章 實驗結果

上述各章節介紹定位系統的設計以及地圖結合演算法，本章設計四個實驗以驗證其應用於實際機器人平台之成效。5.1 節介紹本論文所使用的實驗平台，包括機器人本體之外架尺寸，以及所使用之硬體配備和感測裝置。5.2 節介紹運動控制命令的計算方式，5.3 節以機器人移動直線短距的的定位結果，驗證機器人定位系統在建立地圖階段之精確程度，5.4 節以短距離且來回不斷運行的方式驗證定位系統在已建立過的環境資訊下的定位結果，並分兩小節，分別以未加路徑控制以及使用控制器，比較機器人在運行時的差異與定位導航系統執行的成果，5.5 節以實驗室做為實驗環境，繞行較長距離驗證地圖接合演算法下，整個環境分為數個子區域的結果與地圖融合後定位系統的定位精度，最後，5.6 節以室內長距離之實驗，驗證機器人在長距離定位與導航上的應用。

5.1 機器人硬體介紹

本論文使用二輪差動式機器人移動平台如圖 5.1，此移動平台採用雙獨立驅動輪之運動機構，以速度控制實現移動式機器人運行，移動平台後方裝有一被動式輔助輪，其平衡整體重心位置並使機器人能平順穩定的在室內環境中移動，機器人外部尺寸(長 X 寬 X 高)為 30cmX36cmX85cm，在高度方面，地面至感測器的攝影機中心位置則為 83cm，機器人輪徑為 36cm。所使用的電池規格為 12V 7.2Ah 的鉛酸電池，分別供應控制器以及馬達。其中主控電腦為威強公司出產之工業電腦主機板(NOVA-HM551-R10)，並搭載 Intel i7 2.66GHZ 之 4 核 CPU，主要處理系統運算以及機器人運動控制命令。馬達是使用 FAULHAER 公司所出品，型號為 3863H-012C，並具型號為 HEDS5 之軸編碼器，頭部配置 Microsoft 所出產之深度及影像感測裝置 Kinect，其彩色攝影機的資料串流可提供 640X480 32-bit 每秒 30frame 的彩色影像，深度感測器的資料串流則可提供 320X240 16-bit 每秒 30frame 的深度資訊。圖 5.2 為機器人的硬體架構圖，以 Kinect 作為感測裝置，工業電腦作為運算核心，從電腦發出運動控制給馬達控制器，經由馬達驅動

器驅動馬達運轉，完成機器人的移動控制。

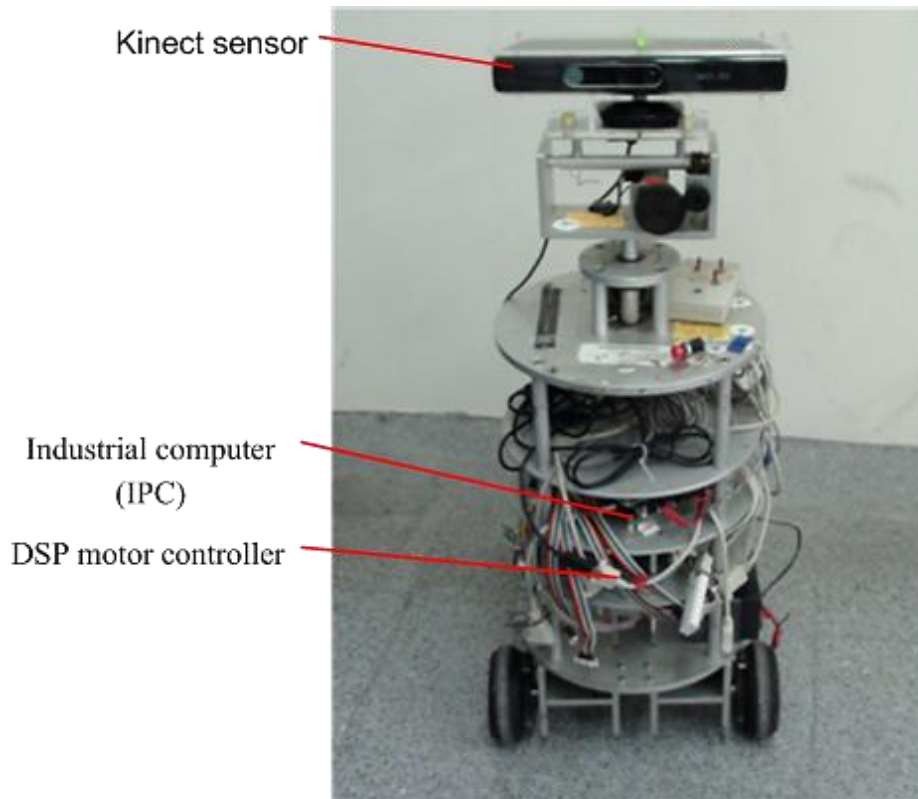


圖 5.1 機器人實體圖

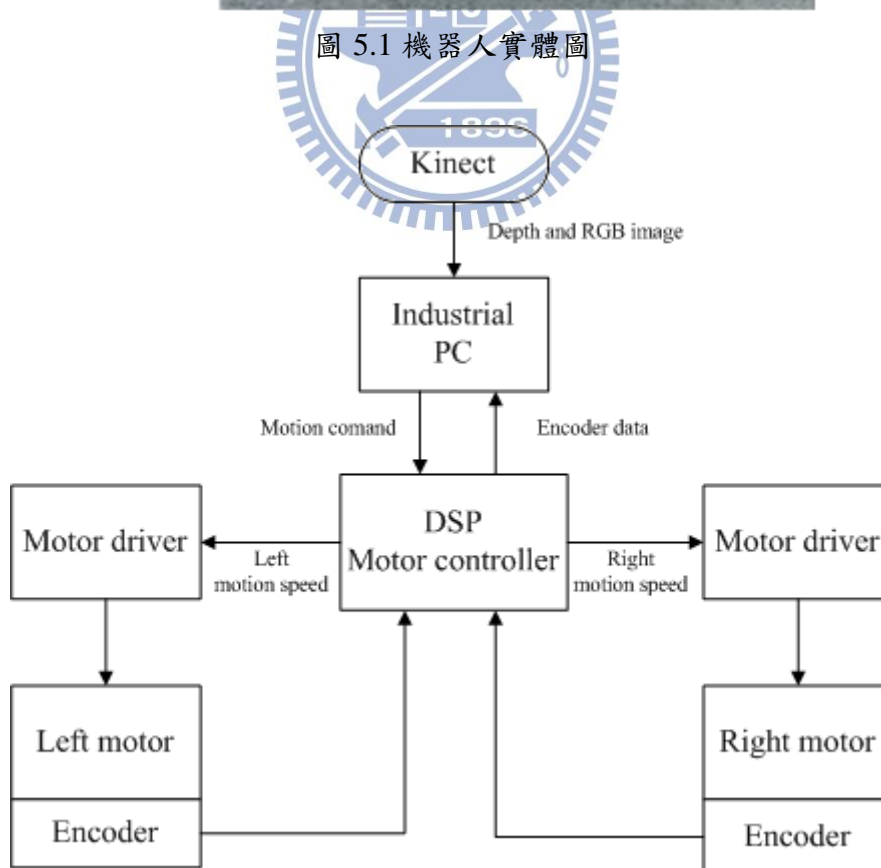


圖 5.2 機器人硬體架構圖

5.2 機器人運動控制

本論文在路徑控制器的設計架構如圖 5.3，機器人在環境中運行的準則是以事先設定在環境中的路徑點，使機器人以這些路徑點為目標前進，而機器人狀態資訊是透過定位系統修正後的狀態資訊，在運動控制上設計的方式上，直線速度是設定為固定大小的控制命令，而角速度的變化則是依機器人朝向與目標點間的夾角差異程度不同計算兩輪速差，兩者的關係是一比例的關係(圖中以 K_{wp} 表示)，而實際的演算規則如表 5.1 所示，該規則將機器人朝向與目標點間夾角的差異程度分為三個區間，為避免機器人旋轉速度太快，設定夾角差異大於 θ_{d_THMax} 對應的最大左右輪速差為 V_{TH_Max} ，其餘部分輸出的速差與夾角間差異成比例關係，其中 θ_{diff} 表示為機器人朝向與目標點間的夾角， θ_{d_Abs} 表示為 θ_{diff} 的絕對值， V_{diff} 表示計算而得的兩輪速差， V_{TH_1} 為一設定的速差量， θ_{TH_1} 為一設定的角度差閾值， P_{set} 為設定於環境中的路徑點。圖 5.4 顯示機器人與目標點之間的關係，其中 θ_1 表示目標點到機器人的連線與 X 座標軸之間的角度。

5.3 EKF-SLAM 定位系統實驗-三公尺直線運動

此實驗在於驗證以本論文所設計之定位系統同時定位與建立地圖下之定位精度，實驗環境為實驗室內一角，區域大小約為 4mX5m 的環境，機器人移動目標設定為直線移動三公尺。在三公尺的路徑里機器人 EKF 演算法達成同時定位

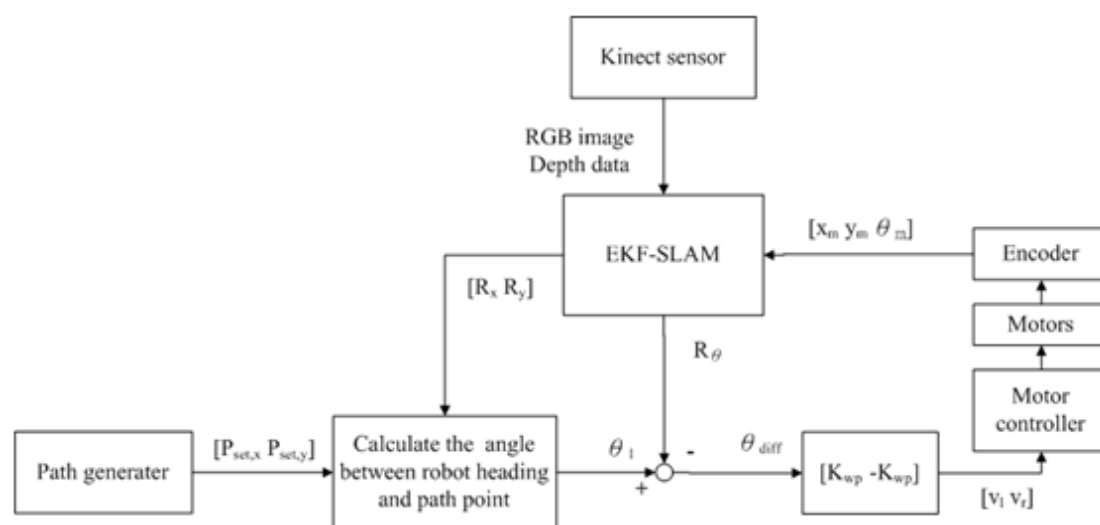


圖 5.3 路徑控制器架構圖

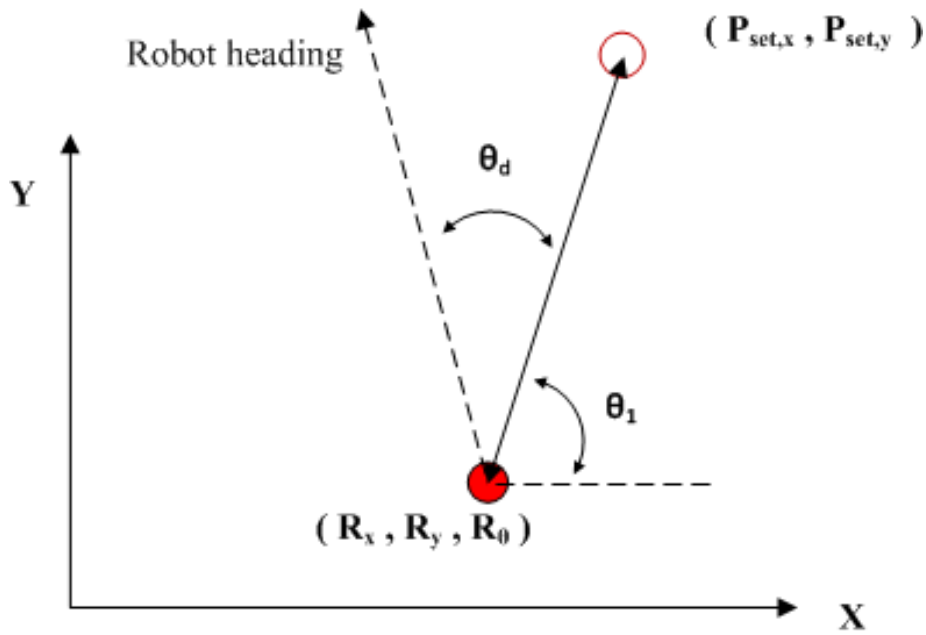


圖 5.4 路徑控制器之架構圖

表 5.1 運動控制計算流程

```

 $\theta_{diff} = \text{atan2}(P_{set,y} - R_{t,y}, P_{set,x} - R_{t,x}) - R_{t,\theta}$ 
 $\theta_{d\_Abs} = \text{abs}(\theta_{diff})$ 
if( $\theta_{d\_Abs} < \theta_{d\_TH1}$ )
     $V_{diff} = \theta_{d\_Abs} / (\theta_{d\_TH1} - 1) * V_{TH\_1}$ 
else if( $\theta_{d\_TH1} < \theta_{d\_Abs} \leq \theta_{d\_THMax}$ )
     $V_{diff} = (\theta_{d\_Abs} - \theta_{d\_TH1}) / (\theta_{d\_THMax} - \theta_{d\_TH1}) * V_{TH\_1}$ 
else
     $V_{diff} = V_{TH\_Max}$ 
 $V_t = V_{t\_set}$ 
if( $V_{diff} < 0$ )
    {  $V_l = -V_{diff}$  ;  $V_r = V_{diff}$  }
else
    {  $V_l = V_{diff}$  ;  $V_r = -V_{diff}$  }
return  $V_r, V_l$ 

```

與地圖建立之功能，圖 5.5 為此實驗下定位系統估測之機器人移動軌跡以及地圖環境資訊和實際環境配置圖結合的結果，藍色點為帶有距離的特徵點，其不做為定位系統所使用，而紅色點則做為定位系統所使用之環境資訊，藍色線段為機器人移動軌跡。

機器人的移動目標為設定之目標點(0,300)的位置，當機器人停止運行後記錄下其狀態資訊，運行次數共為 10 次，表 5.2 呈現實驗結果。其中 Ground truth 是量測實際環境距離資訊的結果，統計 10 次機器人的運行，實際與估測的機器人狀態資訊差，X 方向的平均誤差為 1.30 公分，Y 方向的平均誤差為 3.46 公分，而平均角度差則為 0.559 度，圖 5.6 則以直條圖顯示各次實驗的誤差。

5.4 EKF-SLAM 定位系統實驗-機器人直線來回實驗

本實驗使機器人行進在一約 3 公尺長的路徑範圍內，機器人在這三公尺的直線來回走動，目的在於驗證機器人能用已建立之地圖資訊持續修正機器人狀態資

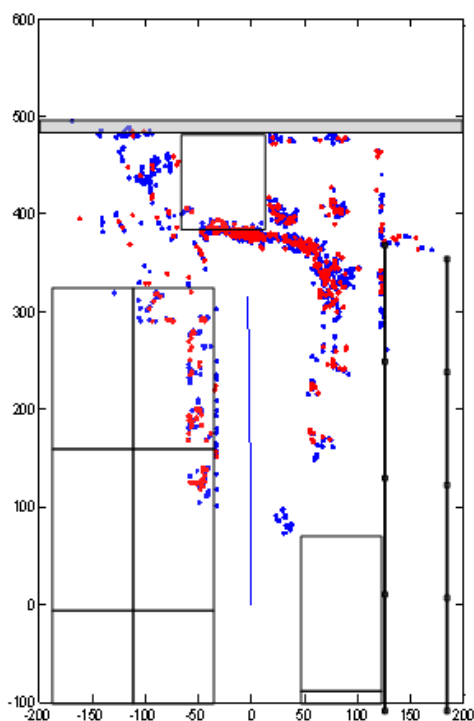
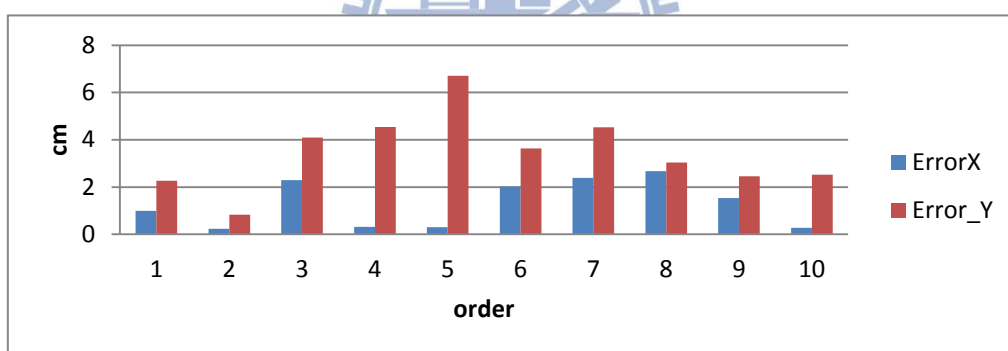


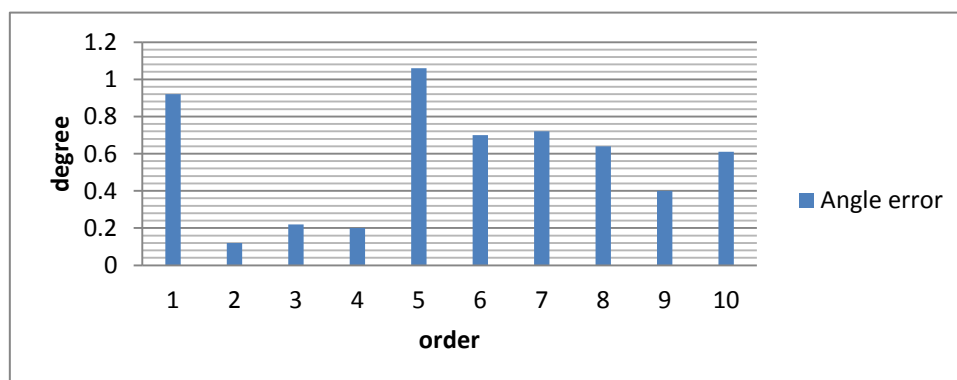
圖 5.5 直線移動實驗結果與實際環境配置對照圖

表 5.2 直線移動實驗之機器人位置估測結果

	Ground truth			EKF Estimation			Error		
Experiment 1	-4.71	314.32	92.5	-3.72	316.58	91.58	-1.00	-2.26	0.92
Experiment 2	-4.03	330.83	91.5	-3.80	330.00	91.38	-0.23	0.83	0.12
Experiment 3	-5.59	302.29	91.0	-3.30	306.38	91.22	-2.29	-4.09	-0.22
Experiment4	-2.73	295.23	91.0	-2.42	299.77	90.80	-0.31	-4.54	0.2
Experiment5	-2.67	295.04	92.0	-2.37	301.75	90.94	-0.31	-6.71	1.06
Experiment6	-4.33	302.13	91.5	-2.31	305.76	90.80	-2.02	-3.63	0.7
Experiment7	-4.40	301.94	90.8	-2.01	306.47	90.08	-2.39	-4.53	0.72
Experiment8	-5.03	303.93	91.5	-2.36	306.97	90.86	-2.67	-3.04	0.64
Experiment9	-3.15	304.24	90.5	-1.62	306.69	90.10	-1.53	-2.45	0.4
Experiment10	-3.46	307.84	92.1	-3.19	310.36	91.49	-0.27	-2.52	0.61



(a) 直線實驗之機器人二維座標誤差



(b)直線實驗之機器人角度差

圖 5.6 三公尺定點實驗之誤差直條圖

訊並更新特徵點訊息。此節並分為兩個部分，一者是將機器人之運動模式以速度控制器施予速度之命令而無角速度方向修正，也並無路徑控制器限制其行徑路線，與之比較的則為以修正機器人朝向角之運動控制器，控制機器人運動軌跡於兩目標點之直線上，藉以比較機器人在兩個模式下之差異，並驗證定位系統施行之成效。

5.4.1 無使用控制器之機器人直線來回實驗

無使用控制器之機器人來回實驗，機器人由起始位置出發，以前章節所述之演算法同時定位並建立地圖，當機器人到達所設定之目標點後，下達反向之速度命令使機器人朝向角與運動方向迴轉 180 度，由此開始，機器人以建立好之特徵點地圖資料庫做定位系統運算，當機器人回到起始點附近時，重新改變速度命令使其運動方向與朝向一致，機器人便在這兩個速度控制變化下不斷重複運行，本實驗以機器人來回一次為一趟，整個實驗來回 5 趟，移動距離約 30 公尺。

機器人由起點出發，系統同時定位與建立地圖將影像所取得的資訊化為特徵點地圖資料庫並修正機器人狀態，建立地圖的階段是從起點開始直到三公尺遠的折返點，當機器人完成最後一次地圖建立與修正機器人狀態後，運動命令使機器人以後退的方式移動，此時系統從地圖建立的模式轉換為使用已建立之地圖資料庫的模式，從此刻開始直到機器人停止運行系統使用已建立之地圖資料庫修正機器人的姿態以及資料庫特徵點的三維座標。

使機器人在以此前進與後退之運動模式的情形下，我們發現機器人會有明顯的角度變化，並隨著來回之次數提高，其角度差與起始朝向的差異也越嚴重，其原由來自於本論文所使用之機器人移動平台架構，本論文所使用之移動平台為雙獨立驅動輪，並於後方裝設一輔助輪，使機器人能自由且平滑的移動，但是，機器人以直線前進後運動命令反向 180 度不轉朝向直接後退，輔助輪轉向必須隨運動方向轉 180 度，在這轉向過程中影響了原本機器人的朝向角，在機器人的運動控制上，這會是個惱人的控制結果，但對於本論文而言，卻是驗證定位系統在機

器人狀態發生變異時是否能正確修正的極佳情境，而與 5.4.2 節利用運動制器修正機器人朝向角的實驗不同之處在於，本小節之實驗驗證當機器人狀態發生變化時定位系統之成效，5.4.2 節則證實以此定位系統的結果，我們能使機器人運行於設定之軌跡，從而達到導航之目的。

圖 5.7 為機器人在本實驗所擷取之實際運行狀況，每一個子圖上方為機器人實際於環境中的運行影像，在初始狀態下，機器人右輪對準綠色膠帶所標示的直線上(如圖 5.7 的(a))，接著(b)~(h)則分別取機器人接近三公尺目標點與起始點之結果，由子圖上方之外拍影像，可以觀察到機器人隨著來回趟數的增加，其偏離原始設定之直線路徑越來越多，而子圖下方影像則為定位系統所得到之機器人狀態向量和移動軌跡，以及環境特徵點地圖建立的結果，其中藍色箭頭為機器人狀態，其箭頭尖端代表當時機器人的二維座標點，箭頭方向為機器人朝向，黃色箭頭則是機器人之運動方向，箭頭長度則為速度與角速度總值之強度，藍色線條為估測之機器人移動軌跡，而環境中散佈的特徵點，其中白色點為帶有距離的特徵點，但不作為定位系統所使用，而紅色點則做為定位系統之輸入，由實驗結果所估測的機器人移動軌跡與實際機器人在環境中行進的路線做比較，可以發現估測的機器人軌跡隨著機器人來回的移動逐漸偏離原本設定的直線路徑，顯示系統能夠藉由已建立之地圖特徵點訊息有效的估測機器人姿態，使估測的機器人狀態與實際的機器人位置相符。

圖 5.8 為本實驗實驗結果與實際環境配置結合圖，其中藍色點為帶有距離資訊之特徵點，其不做為定位系統之輸入，紅色點則為定位系統所使用，藍色線段為整個實驗機器人的路徑軌跡，實際機器人的運動過程，機器人由前進轉為後退再由後退轉為前進，隨著運行的時間越久，機器人朝向角的偏移越大，機器人姿態也就與原本設定的直線軌跡相差越遠，而以本論文的系統所估測出來的機器人的路徑也是逐漸偏移的直線，顯示整體估測的趨勢是正確的，而位置估測的精確程度本論文以機器人停在原點附近的估測值和量測值作為評估的依據。

本次實驗定位系統估測結果與實際位置的比較圖如表 5.3，量測的機器人位

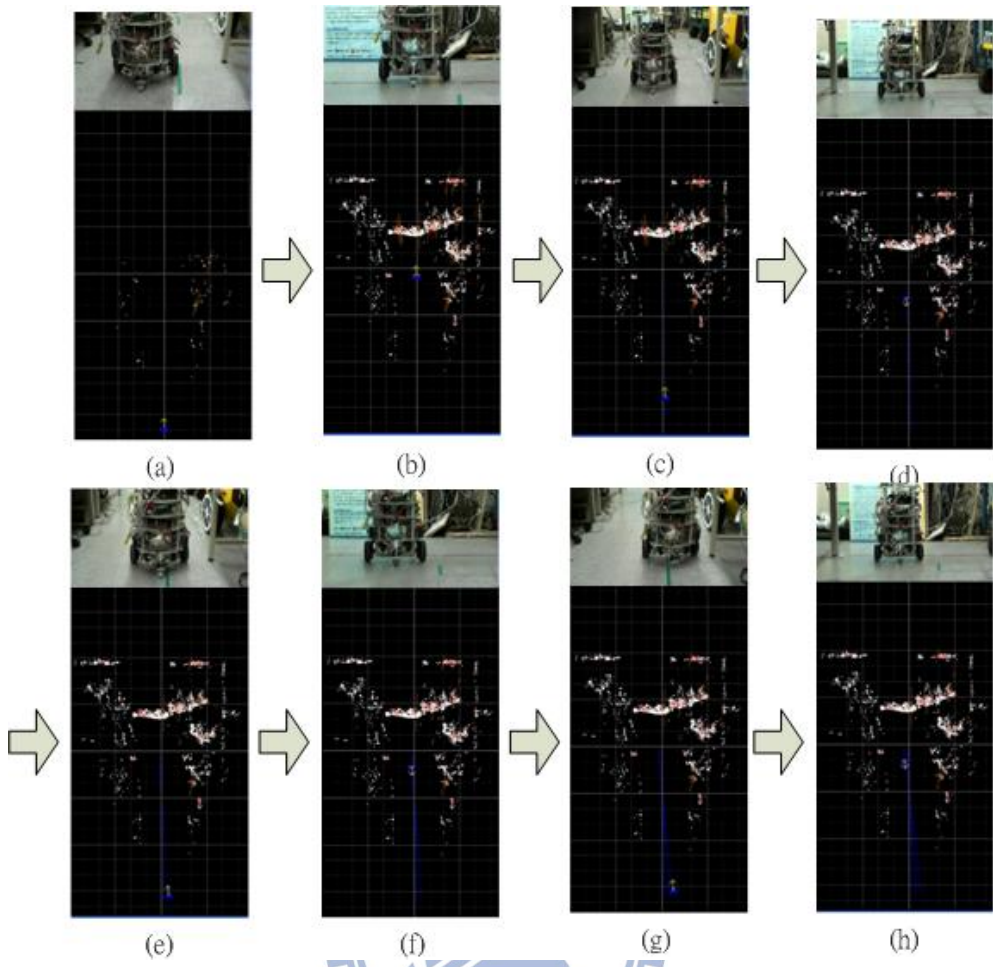


圖 5.7 機器人來回實驗之過程變化圖

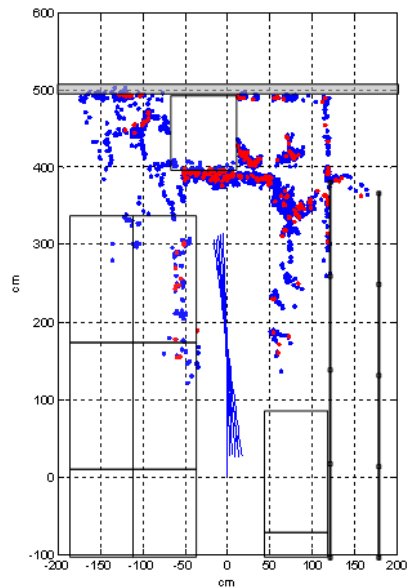


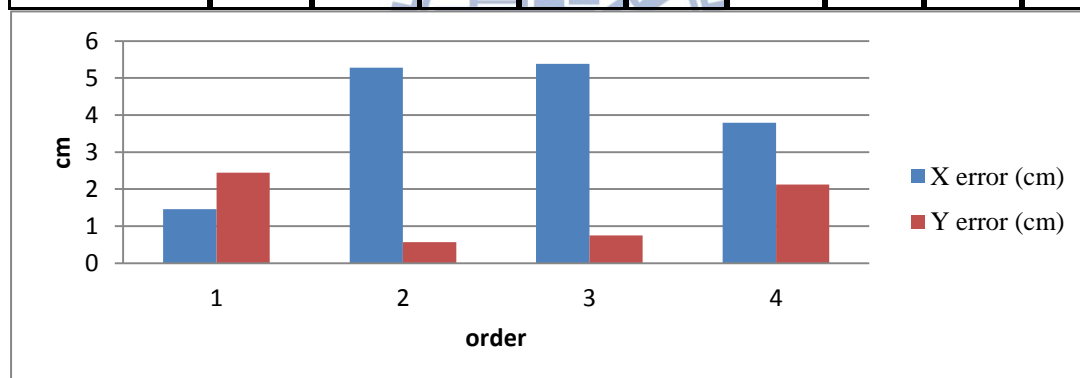
圖 5.8 無控制器直線來回實驗結果與實際環境配置對照圖

置資訊為靠近起始點附近的位置，Ground truth 的量測是將機器人初始狀態之座

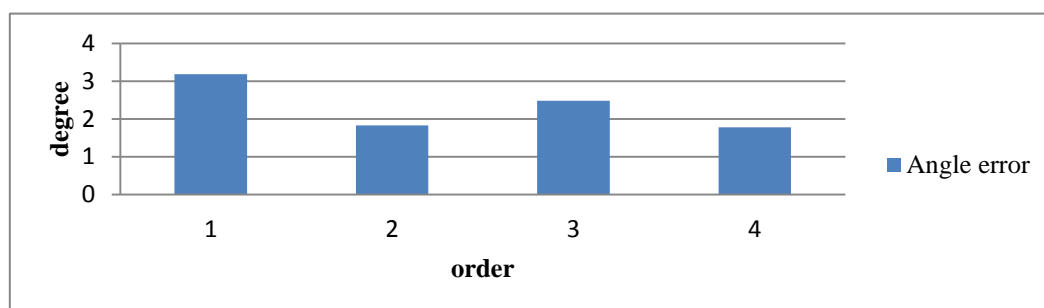
標系統對應於實際環境中，量測機器人停置後的二維平面座標以及朝向角，其與定位系統估測之機器人狀態資訊間的差異，以表 5.3 計算機器人位置與朝向角的平均誤差，由實驗結果統計得知，X 方向的平均誤差為 3.98 公分，Y 方向的平均誤差為 1.47 公分，平均角度差則為 2.32 度，圖 5.9 將二維平面座標的差值以及角度差以直條圖的方式表示。

表 5.3 無控制器直線來回實驗之機器人位置估測結果

	Ground truth			EKF Estimation			Error		
	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$
Experiment 1	19.67	28.91	96.32	18.21	31.35	99.5	-1.46	2.44	3.18
Experiment 2	22.31	27.86	98.5	17.03	27.29	96.67	-5.28	-0.57	-1.83
Experiment 3	18.21	29.21	97.8	12.83	28.46	95.32	-5.38	-0.75	-2.48
Experiment4	12.91	77.61	97.8	9.12	75.49	96.02	-3.79	-2.12	-1.78



(a) 二維座標誤差



(b) 角度差

圖 5.9 無控制器直線來回實驗之誤差直條圖

圖 5.10 記錄無路徑控制器之機器人人直線來回實驗的運動軌跡，以本論文所設計之定位系統所估測的機器人路徑，隨著來回趟數的增加機器人朝向角的變化也越大，使得 X 方向的位移也越嚴重，而定位系統能正確的修正機器人狀態向量，使估測結果趨近實際情況。

5.4.2 使用控制器的機器人直線來回運動

本節所進行之實驗與前一小節相同，皆以設定的兩目標點(0,300),(0,0)讓機器兩目標點間來回直線移動，不同的是本實驗增加一個以機器人朝向角與目標點間的角度差為輸入之運動控制器，藉以調整機器人朝向角，使其能在設定的兩目標點之間，盡可能貼近直線的運動軌跡，本實驗移動次數共為 4 趟，移動距離則約為 24 公尺。

圖 5.11 為以運動控制器修正機器人朝向角後的直線來回實驗結果，藍色點為帶有距離的特徵點，但其僅做為環境資訊而不用於定位系統，紅色點則為定位系統所使用之特徵點，藍色線段為機器人移動軌跡，將本實驗之結果與無使用路徑控制器之結果比較，透過控制器的使用，我們可以使機器人近乎以直線軌跡的

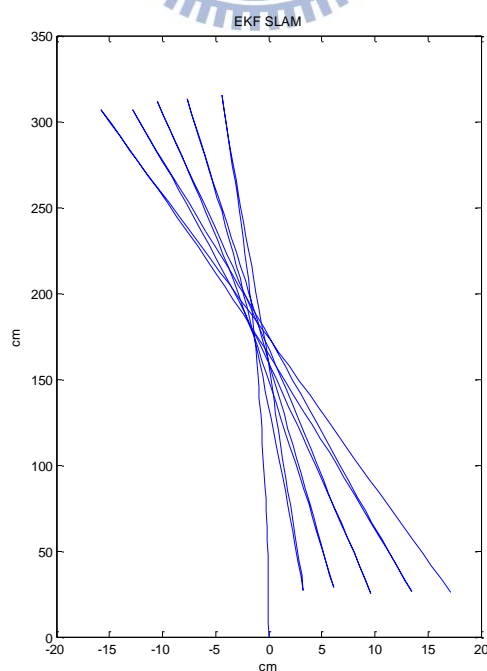


圖 5.10 無路徑控制器直線來回實驗之運動軌跡

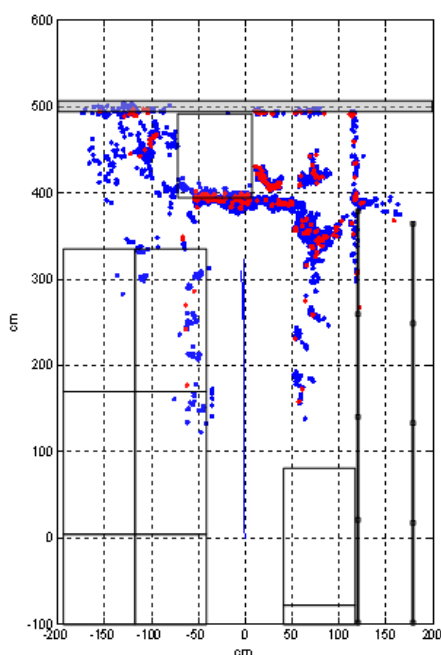


圖 5.11 具控制器直線來回實驗結果與環境配置對照圖

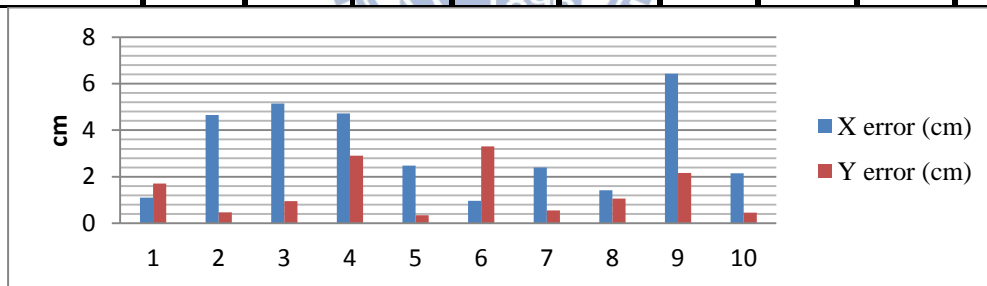
路線運動，因此，以本論文所設計之定位系統搭配路徑控制器的使用，我們可以使機器人運行於設定的路徑軌跡，達成機器人導航之目的。

本實驗 Ground truth 與估測之機器人狀態之結果如表 5.4，其中實際位置與估測之平均誤差依表 5.4 統計，其 X 方向平均誤差為 3.14 公分，Y 方向平均誤差為 1.39 公分，平均角度差則為 1.26 度，而最大 X 誤差為 6.42 公分，最大 Y 誤差為 3.3 公分，最大角度差為 1.26 度，圖 5.12 則將誤差結果以直條圖的方式繪製。

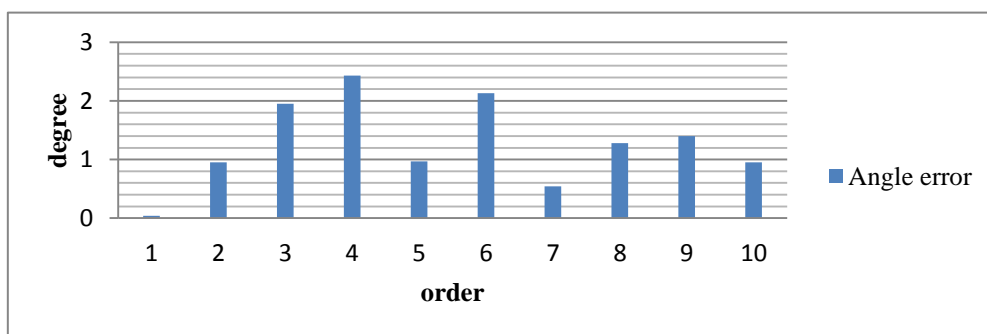
圖 5.13 為本實驗定位系統所估測之機器人路徑軌跡，本實驗以機器人朝向角與目標點之角度差為輸入的運動控制器，修正機器人移動時的朝向角，使得整個路徑不斷向 $X=0$ 的直線趨近，而圖 5.14 為使用運動控制器修正機器人朝向角，在整個實驗過程中的角度響應圖，機器人的朝向角在過程中皆在 90 度正負 2 度內，最大朝向角為 91.53 度，最小朝向角為 88.39，由實驗可證實本論文所使用之定位系統能使估測的機器人狀態趨近實際機器人運動軌跡，並能以運動控制器控制機器人移動軌跡達成機器人導航之功能。

表 5.4 具路徑控制器直線來回實驗之機器人狀態估測結果

	Ground truth			EKF Estimation			Error		
	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$
Experiment 1	-1.73	46.80	25.40	-0.63	45.10	90.56	1.10	-1.70	-0.04
Experiment 2	1.95	18.27	93.00	6.60	17.80	93.95	4.65	-0.47	0.95
Experiment 3	4.47	16.00	93.20	-0.67	16.95	91.25	-5.14	0.95	-1.95
Experiment4	0.59	25.87	88.50	-4.12	22.97	86.07	-4.71	-2.90	-2.43
Experiment5	1.98	16.62	92.50	-0.50	16.27	91.53	-2.48	-0.35	-0.97
Experiment6	2.17	26.38	88.80	3.14	23.08	90.93	0.97	-3.30	2.13
Experiment7	2.50	16.92	93.0	0.10	16.37	92.46	-2.40	-0.55	-0.54
Experiment8	1.93	17.79	92.00	0.51	18.84	90.72	-1.42	1.06	-1.28
Experiment9	4.93	24.52	88.50	-1.49	22.36	87.1	-6.42	-2.16	-1.40
Experiment10	3.30	20.02	90.5	1.15	19.56	89.55	-2.15	-0.46	-0.95



(a)二維座標誤差



(b)角度差

圖 5.12 具路徑控制器直線來回實驗之誤差直條圖

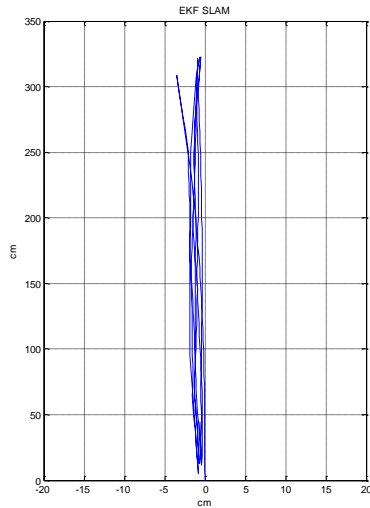


圖 5.13 具路徑控制器直線來回實驗之運動軌跡

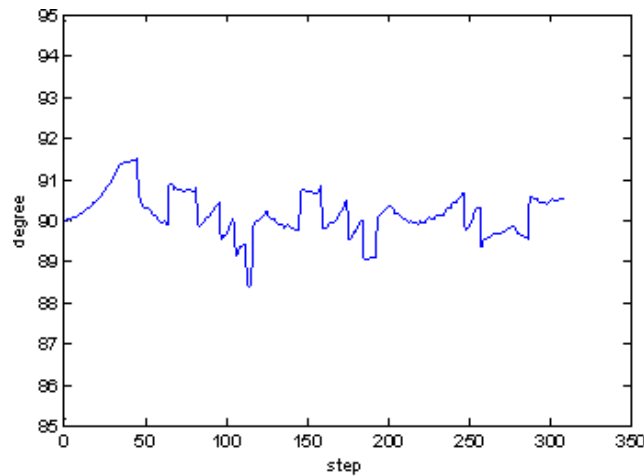


圖 5.14 機器人角度響應圖

5.5 EKF-SLAM 定位以及地圖接合 機器人方型軌跡運動

為了解決 EKF 定位系統在較大的環境運行時，隨著特徵點數量增加而隨之提高的運算成本，本論文藉由地圖接合的方式，將特徵點地圖以機器人路徑分界的方式分為數個區間，藉此驗證其降低定位系統所使用的特徵點數量，當區域地圖建立完成後以地圖融合演算法維持整體環境資訊，修正區域間的特徵點差異並修正機器人狀態資訊。

本實驗主要的目的，在於驗證 EKF 定位系統加入地圖接合演算法後在定位精度上的結果，並以無使用地圖接合演算法之 EKF SLAM 繞行此實驗環境一圈，比較兩者在即時運算能力上的差異。本實驗機器人所運行的實驗環境為 6mX10m

的實驗室，機器人由原點(0,0)出發，繞行 4.8mX1.8m 的方形路徑，機器人繞行第一圈時為建立地圖的階段，完成整個環境的地圖建構後，以建立好之特徵點地圖資訊作為環境資訊，持續修正機器人狀態並更新特徵點區域地圖，機器人重複繞行 4 圈，機器人於此實驗環境共繞行 5 圈。在實驗進行中，當機器人處於建構地圖的狀態時，為了降低因機器人移動對環境特徵取樣的影響，機器人最高的運動速度限制在 10 cm/sec；當機器人進入使用地圖資訊的階段後，其最高運動速度則設定為 20 cm/sec。

圖 5.15 為機器人運行於 4.8mX1.8m 的運行路徑繞行五圈所得到的結果，其結果包括估測之機器人移動軌跡以及所建立的特徵點地圖資訊，這些資訊與實際環境的配置圖結合，圖中藍色點為帶有距離資訊之特徵點，其不為定位系統所使用，而紅色點則為定位系統之輸入，藍色線段為機器人位置估測之軌跡線。

機器人運行的運動方式，主要是以 4 個設定於環境中的座標作為機器人移動的主要依據，這四個環境座標分別為(0,100)、(480,100)、(480,-80)和(0,-80)四個二維座標點，機器人的運動控制上是利用運動控制器以機器人朝向角與目標點的角度差做為輸入修正機器人的前進方向。當機器人繞行第一圈時為建立地圖的階段，依本論文所設計之同時定位與建立地圖之演算法以及地圖接合演算法建立環境資訊，以本實驗所行徑的路徑，依 4.1 節所敘述的區域地圖路徑邊界將整

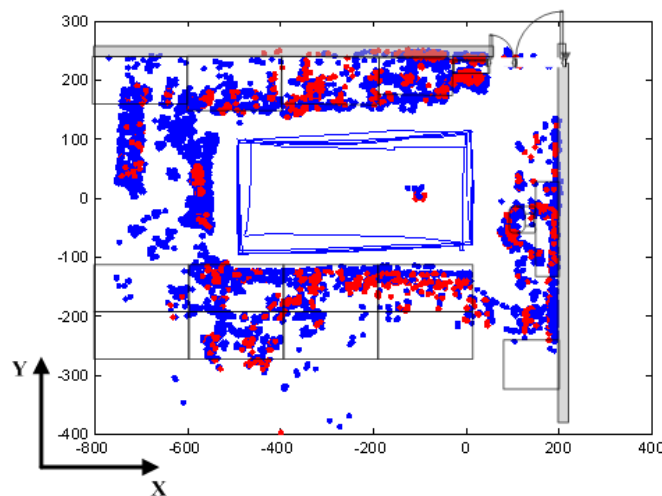


圖 5.15 實驗室方形路徑實驗結果與實際環境對照圖

個環境分為數個部分，此實驗會產生 6 個區域地圖(圖 5.16(a)~(f))，且第 6 個地圖依 4.3 節重合區域偵測的判定，判斷其為重合事件發生之重合區域。最後以地圖融合演算法完成整個實驗室環境的建構，當重合事件發生後，系統依機器人當前位置與區域地圖的路徑範圍的關係決定定位系統所使用的區域地圖，因此其後

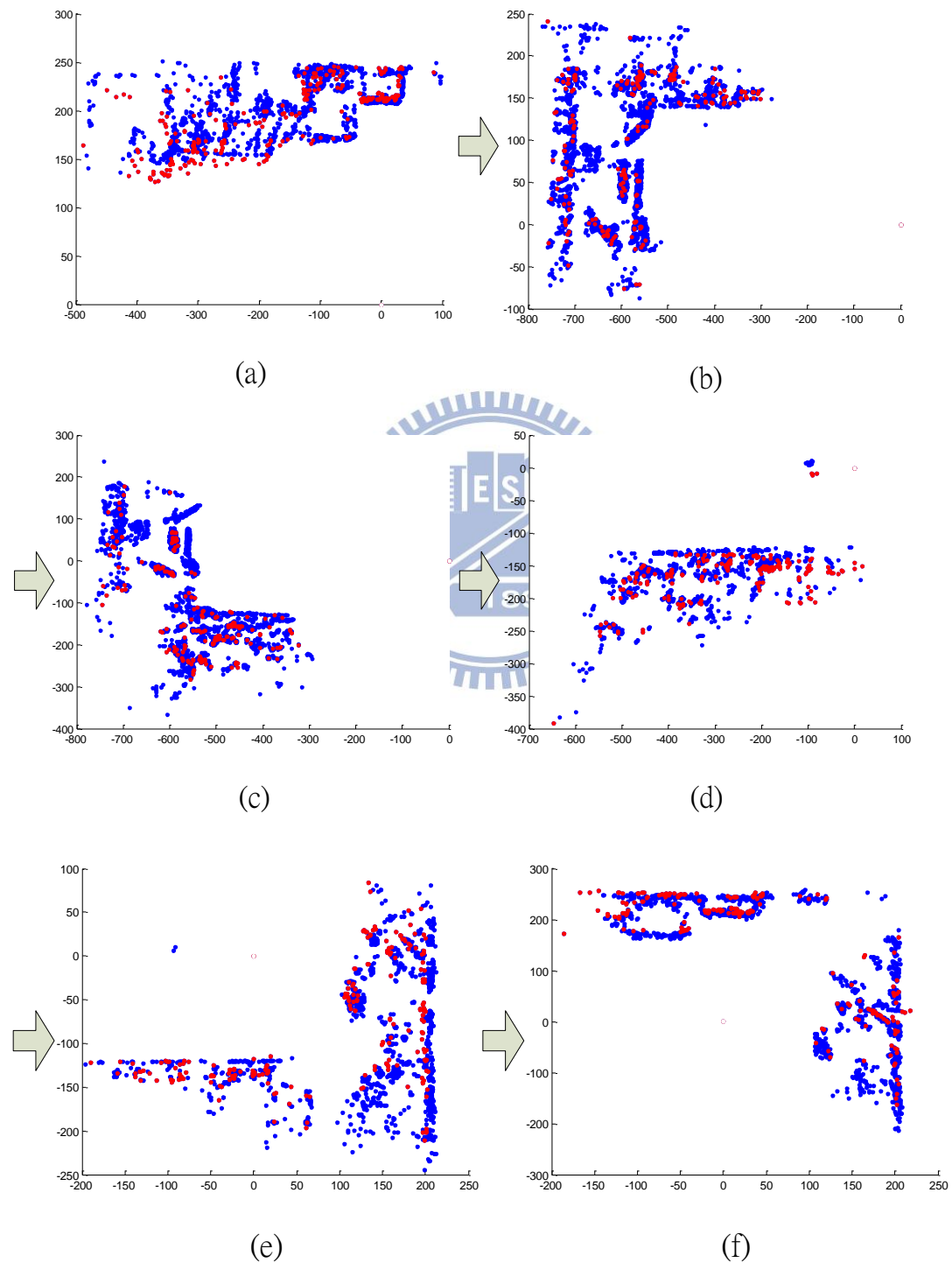


圖 5.16 實驗室方形路徑實驗之區域地圖

便以對應的區域地圖修正機器人狀態並修正特徵點三維座標資訊。依此方式做機器人的定位與導航，整個實驗繞行 5 圈後完成本實驗。

圖 5.17 記錄實驗室方形路徑實驗下每個定位演算步驟花費時間之結果，每一個定位演算步驟從感測器取得影像開始，經過 SURF 作特徵點擷取、特徵點三維座標的計算，直到整個定位系統修正機器人狀態與更新地圖資訊，圖 5.18、圖 5.19 以及圖 5.20，將整個運算時間之結果分為建構地圖時之運算部分，與重複路徑時定位系統所耗費之時間兩部分來檢視此實驗結果。

圖 5.18 顯示的是建立地圖階段每個定位演算步驟所花費之時間，一個區域地圖開始與結束範圍內之運算時間類似於一濼波狀的結果。在本論文中機器人實行定位系統之時機，是以每次機器人移動距離約 8 公分運算一次，而其後隨定位系統所花費時間改變而影響在兩間隔間機器人移動的距離，且在建立地圖的階段限制應用於定位系統的特徵點增加量，在本實驗中令定位系統演算每次最多擴增 7 個特徵點資訊，如此使定位演算在短間隔內持續修正，並減緩矩陣複雜度之增長，以區域 1 為例，區域內定位系統一共修正 46 次，系統狀態向量使用 322 個特徵資訊，因定位系統特徵點在每次步驟的增加數量為固定使得定位系統運算時間與步驟次序的關係圖呈現穩定的成長，而非起伏不定之波狀圖形。根據建構區域地圖時定位系統所耗費的運算時間，當器人重複運行該區域地圖時其定位演

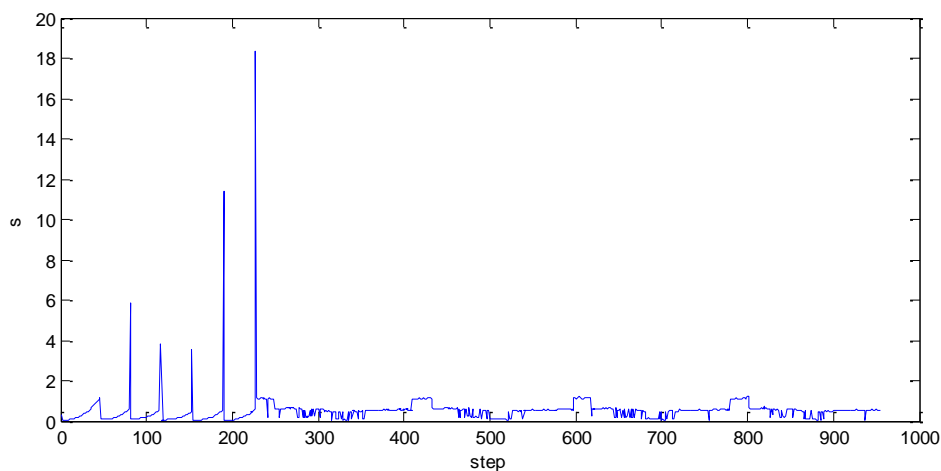


圖 5.17 方形路徑實驗之運算時間

算之時間約略將與建立地圖時之最大運算時間相等，這是因為機器人在區域內所使用的系統狀態維度是固定大小。由圖 5.18 中可以得知，耗費運算時間最多者為區域一，其運算時間為 1.19 秒。附帶一提的是，圖 5.18 在 step 為 1 時的運算時間明顯高於往後數個步驟，是因為 step1 是機器人初始化之階段，機器人需啟動感測裝置等而使運算時間高於接續的數個步驟。

圖 5.19 的部分與圖 5.18 的差異在於顯示地圖融合演算之運算時間，圖 5.18 中突出之脈衝所在即為地圖融合演算之位置，地圖融合演算使用的時機為區域結束與開始之交界，而區域地圖 1 (L_1) 因其為初始地圖亦為世界地圖資訊，故並無使用地圖融合演算，當區域地圖二 (L_2) 完成建構，地圖接合演算法將 L_1 以及 L_2 的資訊結合，這個步驟所花費的時間為 5.86s，其餘部分 L_2 以及 L_3 的資訊結合花費了 3.8s， L_3 以及 L_4 的資訊結合使用了 3.56s， L_4 以及 L_5 的資訊結合使用了 11.39s，最後區域 6 為重合區域，其地圖資訊與 L_5 與 L_1 的資訊結合，故其運算時間最久，其時間為 18.37 秒，地圖接合所使用的區域地圖和演算時間的關係如表 5.6。

圖 5.20 顯示機器人在方形路徑重複運行時之運算時間，依表 5.5 區域地圖建立之結果可知區域 1 最大運算時間為 1191ms、區域 2 為 621ms、區域 3 為 531ms，區域 4 為 455ms、區域 5 為 521ms 而區域 6 為 562ms，當機器人重複運行時，定

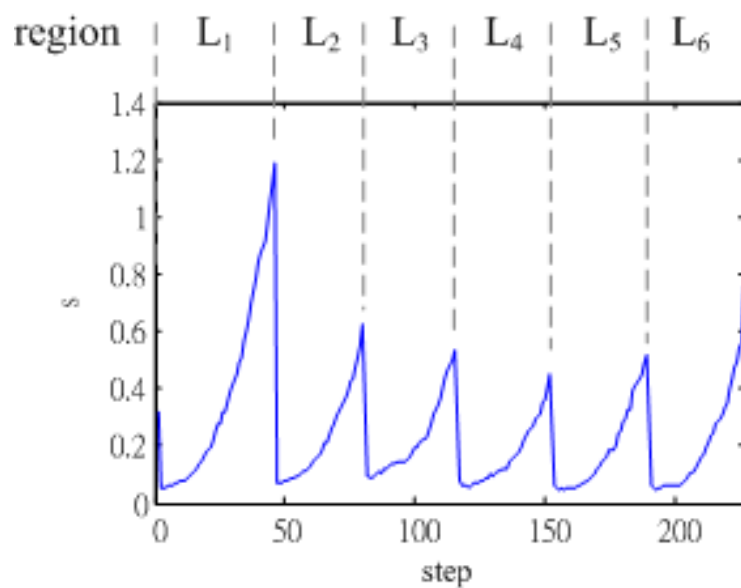


圖 5.18 方形路徑實驗之地圖建構運算時間(不包括地圖融合)

表 5.5 建構地圖階段區域最大運算時間

Map order	Spend time
L_1	1191ms
L_2	621ms
L_3	531ms
L_4	455ms
L_5	521ms
L_6	562ms
Average	646ms

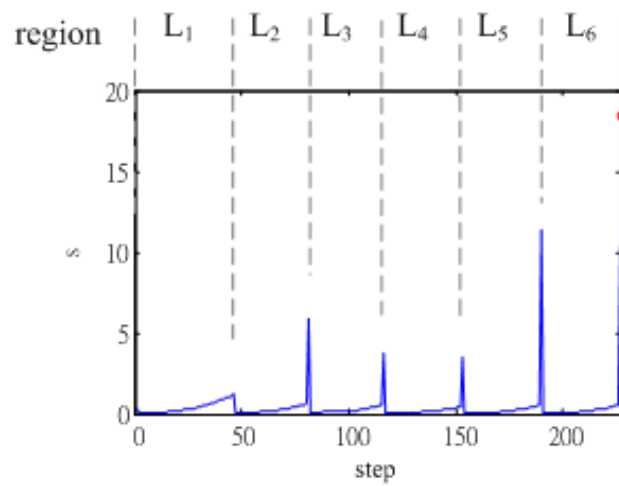


圖 5.19 方形路徑實驗地圖建構之運算時間(包括地圖融合)

表 5.6 建構地圖階段地圖接合運算時間

Join maps order	Spend time
L_1 and L_2	5.86s
L_2 and L_3	3.8s
L_3 and L_4	3.56s
L_4 and L_5	11.39s
L_1 、 L_5 and L_6	18.37s

位系統便依選擇之區域使用該區域之地圖資料庫，而運算時間則與建立該區域之最大運算時間相近，然圖 5.20 中有數個區間帶有劇烈起伏之情況，此劇烈之震盪變化代表其間有數個步驟發生無法比對到定位系統所需特徵點之情形。對應於機器人實際運行時之情況下，其通常發生在區域間重疊區域，依 4.4 節中之地圖選擇判斷區域轉換之發生，但區域轉換出現前區域無法比對到相同特徵，而下一區域比對成功之總數低於閾值，使區域沒有正確的轉換而導致之結果，且當機器人無正常使用定位演算法，機器人在最小移動距離 8cm 後便會再次啟動定位修正，因此有大幅且劇烈之變化。

表 5.7 為本實驗所得到之機器人位置估測結果，此數據是採機器人接近起點 (0,0) 時，當機器人停止運行後所記錄下來的資訊，其目的在於驗證機器人以本論文設計之地圖接合所達成之定位精度，同時機器人繞行相同環境 5 圈，除第一圈為建立環境資訊外，其餘圈數皆是以建立之地圖資訊修正機器人狀態資訊，代表機器人能以資料庫訊息重複性的進行定位演算。統計十次的實驗結果，得到機器人的平均誤差資訊，其中 X 方向的平均誤差為 2.96 公分，Y 方向的平均誤差為 3.56 公分，平均角度差為 1.17 度，而最大誤差的各分項分別為，X 方向最大誤差為 6.44 公分，Y 方向最大誤差為 8.39 公分，角度上的最大誤差則為 3.25 度，圖 5.21 以直條圖的方式呈現機器人實際與估測之狀態資訊的差異。

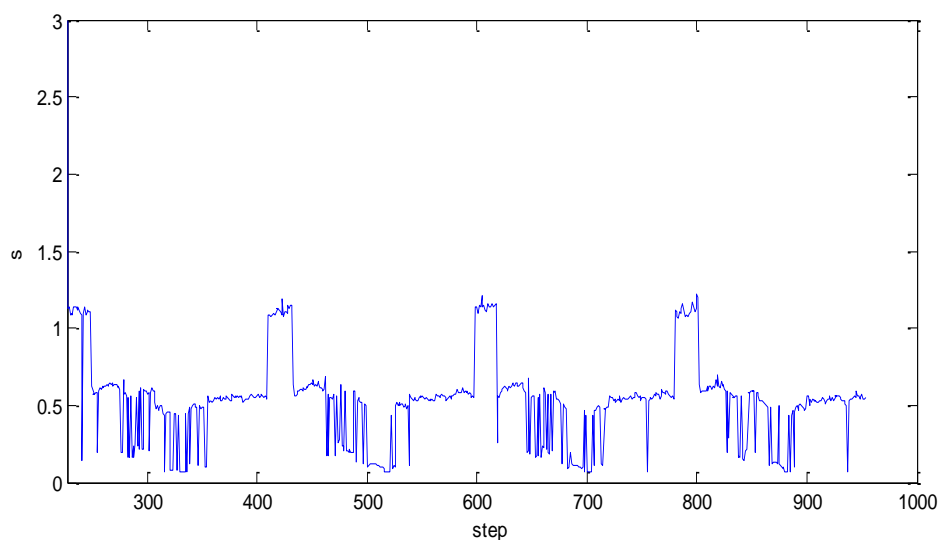
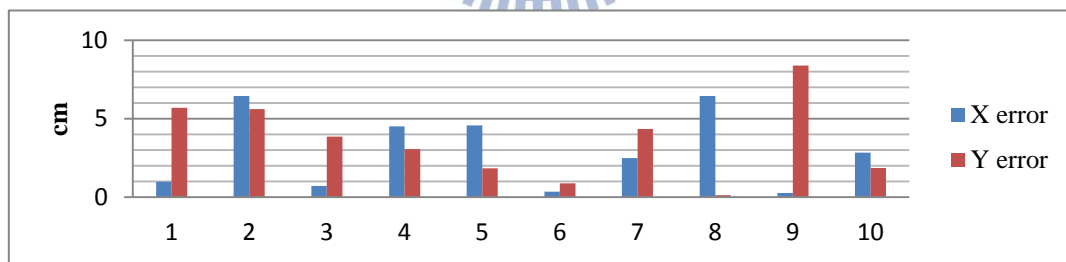


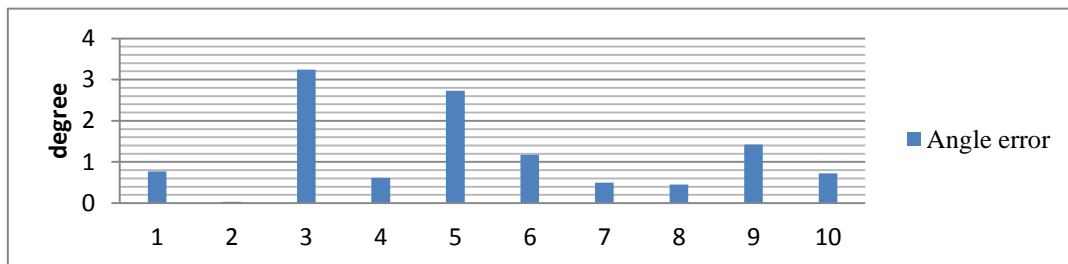
圖 5.20 方形路徑實驗之重複路徑運算時間

表 5.7 實驗室方形路徑實驗之機器人位置估測結果

	Ground truth			EKF Estimation			Error		
	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$
Experiment 1	0.07	5.96	90.60	1.04	0.27	89.83	-0.98	5.69	0.77
Experiment 2	12.49	34.32	91.70	6.05	28.72	91.74	6.44	5.60	-0.04
Experiment 3	13.35	12.35	96.00	12.63	8.50	92.76	0.72	3.85	3.25
Experiment4	8.20	24.14	93.40	3.70	27.19	92.78	4.50	-3.05	0.62
Experiment5	11.19	-2.22	-94.10	6.62	-4.05	91.37	4.57	1.83	2.73
Experiment6	4.42	5.03	91.50	4.76	5.91	90.32	-0.341	-0.883	1.176
Experiment7	11.42	7.13	91.50	8.93	2.78	91.00	2.49	4.35	0.50
Experiment8	17.66	13.43	92.30	11.22	13.56	91.85	6.44	-0.13	0.45
Experiment9	-5.21	5.424	93.3	-5.47	-2.96	91.88	0.26	8.39	1.43
Experiment10	18.43	13.00	97.00	15.61	14.853	96.274	2.83	-1.85	0.73



(a) 二維座標誤差



(b) 角度差

圖 5.21 實驗室方形路徑實驗之誤差直條圖

為了展示地圖接合演算所帶來的好處，另外實作以無地圖接合演算在整個實驗環境中單以 EKF 完成地圖建構，其機器人行進的路徑與特徵點地圖建立的結果如圖 5.22。藍色線段的部分是 EKF 估測機器人位置所得到的移動軌跡，紅色點為定位系統所使用的特徵點，藍色點則為帶有距離資訊的特徵點，但不做為定位系統所使用，與使用地圖接合演算的實驗結果圖 5.15 比較，機器人從原點(0,0)的地方出發，直到繞行一圈後完成地圖的建構。無地圖接合演算的機器人運行到路徑後段時，由於每次移動所需的間隔提高，所以取樣的頻率下降，環境中用於定位系統的特徵點數量也較圖 5.15 的使用數量來的少。另外，機器人的移動軌跡也因為取樣間隔的間隔較長，而難以使機器人運動軌跡符合所設定的路徑。

圖 5.23 則是機器人以無地圖融合演算在建構地圖的過程中每次步驟所花費的運算時間，此結果與圖 5.18 比較，在完成地圖建構所使用的步驟次數上，由於兩個實驗皆是在相同的實際環境下運行，因為兩者每次定位演算所花費的時間不同所以所使用的步驟總數是不相同的。以無地圖融合演算建構地圖，機器人每次運算時間隨逐步增加，當機器人繞行實驗路徑一圈，最後一次定位系統演算的時間為 7.32 秒，以兩者的比較可知藉由地圖接合之做法，可以有效提高機器人定位與建立地圖即時運算之性能。

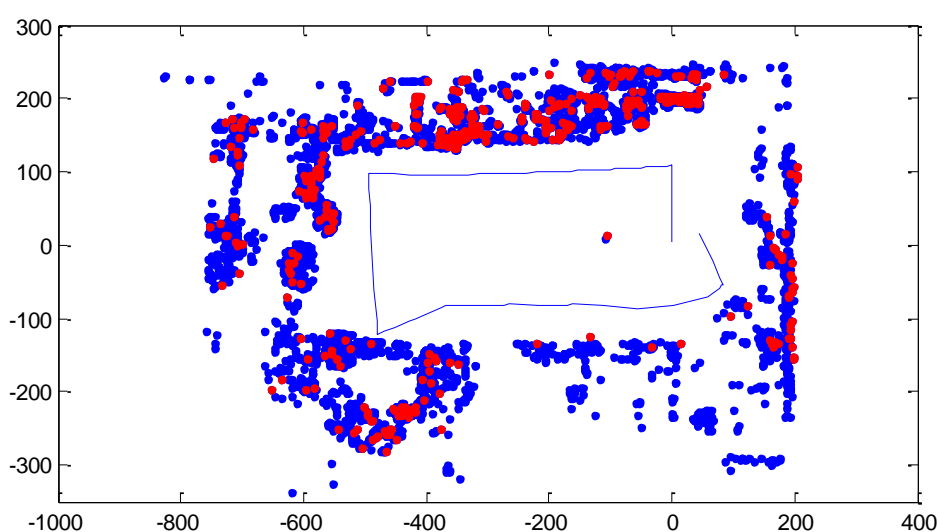


圖 5.22 方形路徑實驗無地圖接合演算之地圖建構結果

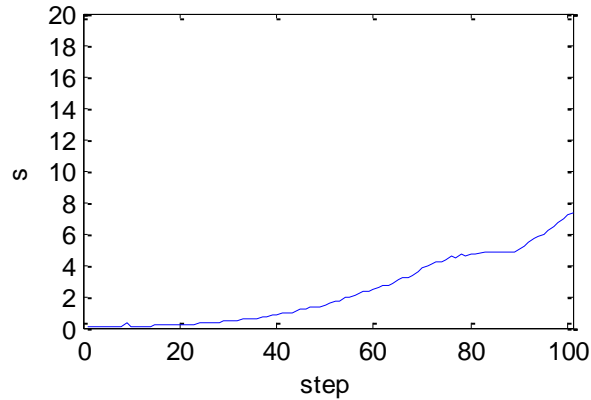


圖 5.23 方形路徑實驗無地圖接合演算之地圖建構運算時間

5.6 EKF-SLAM 機器人室內長距離移動實驗

本實驗中，機器人依本論文所設計之定位系統與地圖接合演算法，運行於一長距離的環境，驗證定位系統估測機器人於此環境之估測精度與地圖建立之結果，整體實驗環境大小約為 16m X 6.65m，依建築格局被劃分為兩個區間，如圖 5.24 所示。本實驗機器人之運行環境如圖 5.24 所示，整個環境由兩個區間所構成，機器人於此環境中之運動方式是以事先設定好的數個環境座標做為機器人的行動準則，大體上機器人運行的方式是先繞行區間 1 一圈，建立區間 1 的環境資訊，接著進入區間 2，完成地圖建立後重新繞行區間 1，接著再繞區間 2，最後繞行區間 1 後完成本實驗，整個機器人移動路徑之總長度約為 82.8 公尺。依本論文所設計之定位系統，機器人行進於各區間所執行的各項步驟詳述如下。

令機器人在環境中設定的行進中繼點位置如圖 5.24 所標 A~G，機器人從原點開始運行，沿途依序經過點 A -> B -> C -> D -> A 建立完區間 1 的地圖，依區域地圖算建立的範圍做判斷，區間 1 會得到 5 個區域地圖，其特徵點分佈(區域地圖建立之結果)如圖 5.25 中(a)~(e)，其中機器人從位置 D 到位置 A 時，由重合區域偵測的判定區域地圖(e)以及區域地圖(a)為重合區域，機器人地圖狀態屬於重合地圖的類型，此時地圖融合演算將(d)(e)(a)三個地圖資訊結合，而完成區間 1 的環境地圖，接著，機器人往區間 2 的方向前進，當機器人從 A -> B 時，機器人使用的地圖資訊屬於已建立之地圖資訊，機器人會使用先前建立好之特徵點

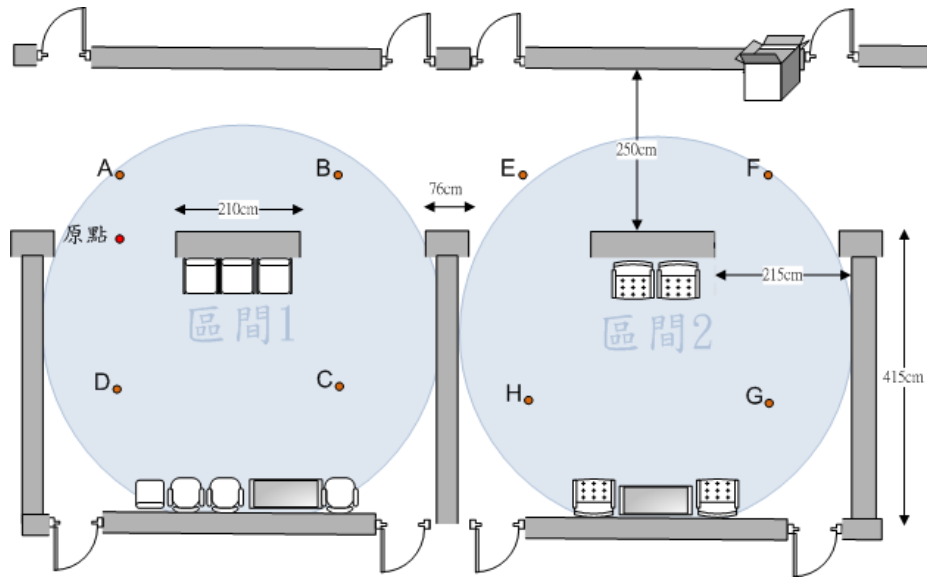
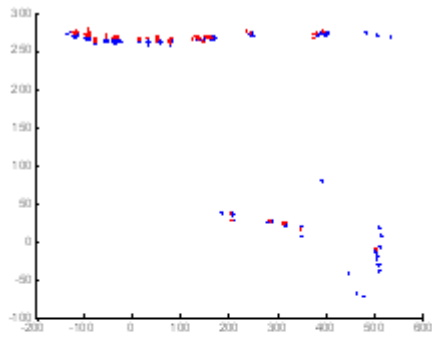
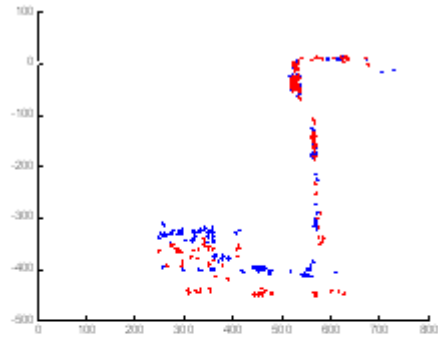


圖 5.24 長距離實驗環境配置圖

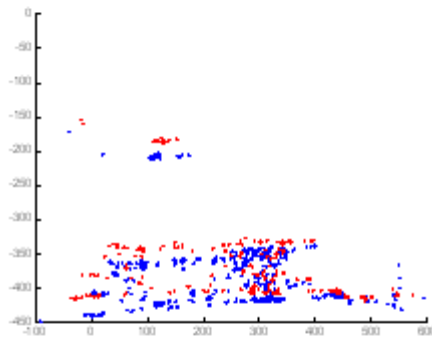
資料庫，做為定位系統之環境資訊，修正機器人狀態向量並且持續更新該段區域的特徵點狀態。接著，機器人從 B → E，機器人由已建立過之地圖範圍進入未到達過的環境，故依演算法之設計開啟新的區域地圖，機器人由位置 B → E 進入區間 2 的環境，依移動位置順序 E → F → G → H → E 繞行區間 2，依區域地圖路徑範圍的判斷結果，區間 2 由 5 個區域地圖所構成，其特徵點分佈(區域地圖建立之結果)如圖 5.26 中(a)~(e)所示。至此，機器人完成所有環境的圖建立。接著，機器人從 E → B 回到區間 1 的區域，然後依行走順序 B → C → D → A → B 重複區間 1，再從 B → E 走至區間 2，並依 E → F → G → H → E 再一次繞行區間 2，最後再繞行區間 1 兩圈，依序為 E → B → C → D → A → B → C → D → 原點，完成本次實驗。在實驗進行中，當機器人處於建構地圖的狀態時，其最高的運動速度限制在 10 cm/s；而當機器人進入使用地圖資訊的階段後，其最高的運動速度則放寬至 14 cm/sec。與其他實驗時採用較快之速度比較，本實驗之機器人運動速度放慢是因為實驗環境不同而做的調整。由於定位系統依賴由環境中偵測到之特徵點對機器人狀態進行修正，修正的方式是以每次影像中與資料庫比對成功的特徵點資訊作為輸入，為避免本實驗之環境在某些區段能夠獲得之特徵點數量較為稀少所可能造成之誤差，實用中採用較慢的速度來提高定位演算的次數，並確保能得到較為精確的特徵點資訊。



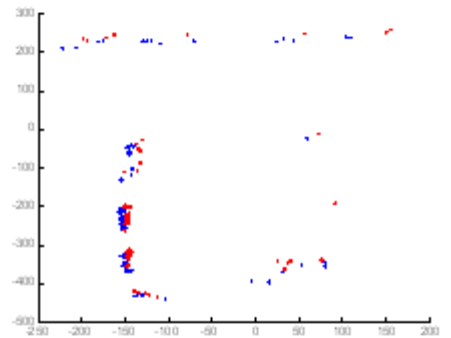
(a)



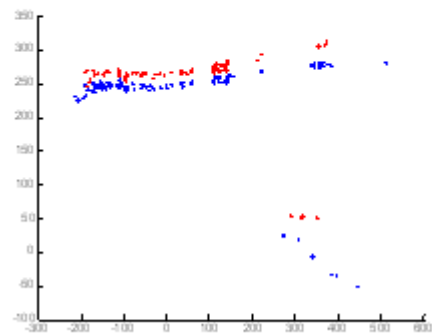
(b)



(c)



(d)



(e)

圖 5.25 室內長距離實驗區間 1 建立之區域地圖

圖 5.27 展示機器人繞行此環境之特徵點地圖與區域地圖之路徑範圍，其為機器人實際運行時各種資訊的結合，圖中左側青色的大圓點為原點之位置，是機器人開始運行的起點所在，黃色箭頭為機器人實際運行時在環境地圖的位置，箭頭尖端為機器人所處的二維座標點，箭頭方向代表當時機器人的朝向角，藍色箭頭則為機器人運動方向，箭頭長度代表速度與角速度總值的強度，格線做為地圖環境的距離參考線，由隔線所構成的方格對應實際環境大小為 90cm X 90cm，而

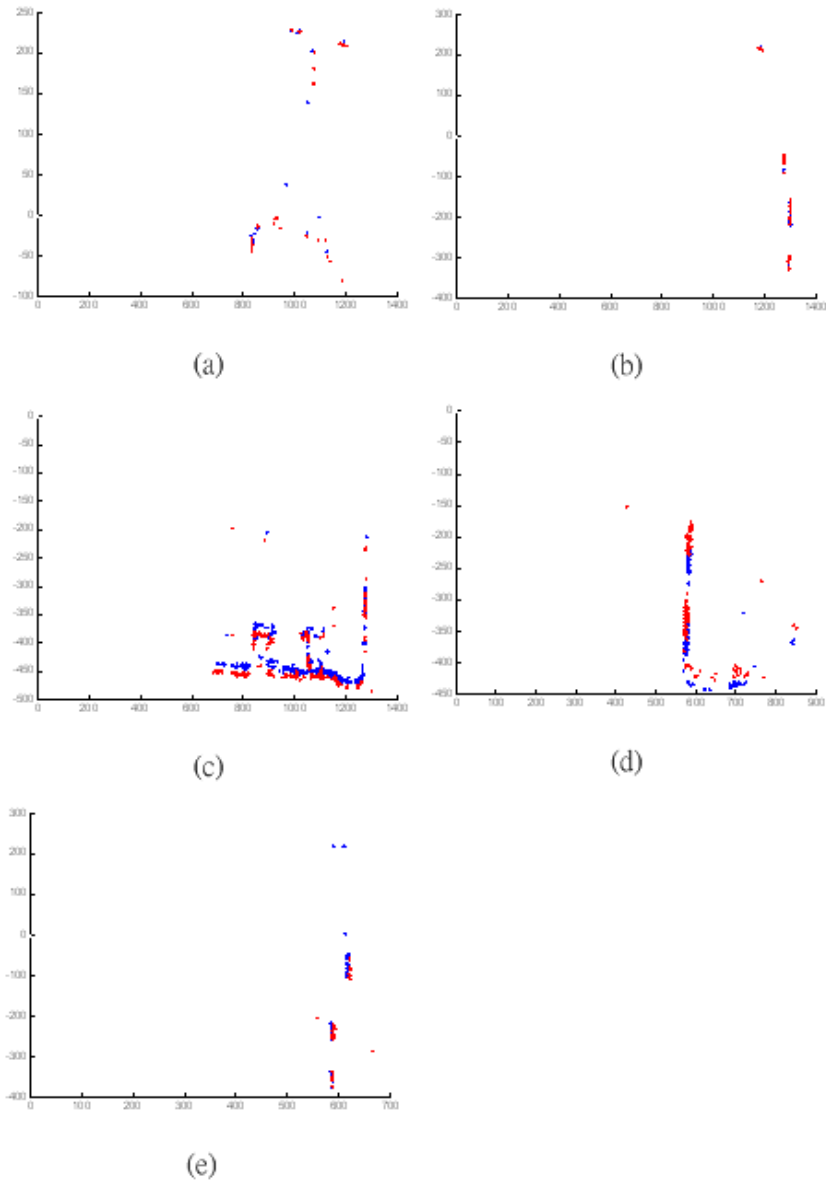


圖 5.26 室內長距離實驗區域 2 建立之區域地圖

黑色點為具有距離資訊的特徵點，但不做為定位系統的特徵點，青色點則做為定位系統的輸入資訊，用以修正機器人狀態向量，而每一個灰色大圓的範圍是機器人判斷該使用哪一個區域地圖之依據，當機器人位置落在圓形範圍內時，機器人選擇對應的區域地圖做定位系統的演算法，而當區域之間有所重疊時，則依 4.4 所述以特徵點符合數量的比較，來決定是否要切換區域地圖的使用。

圖 5.28 為機器人運行之環境特徵點地圖與實際環境配置之結合圖，圖中藍色點為帶有距離的特徵點，其不為定位系統所使用，而紅色點則為定位系統所使用的環境資訊，藍色線段為定位系統估測的機器人移動軌跡。

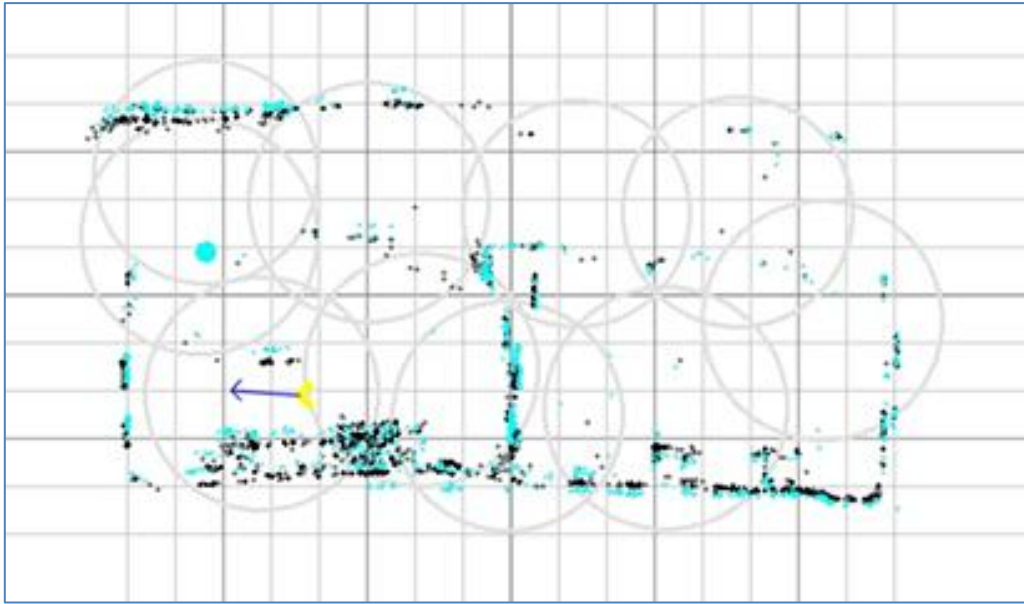


圖 5.27 長距離運行機器人建立之環境地圖

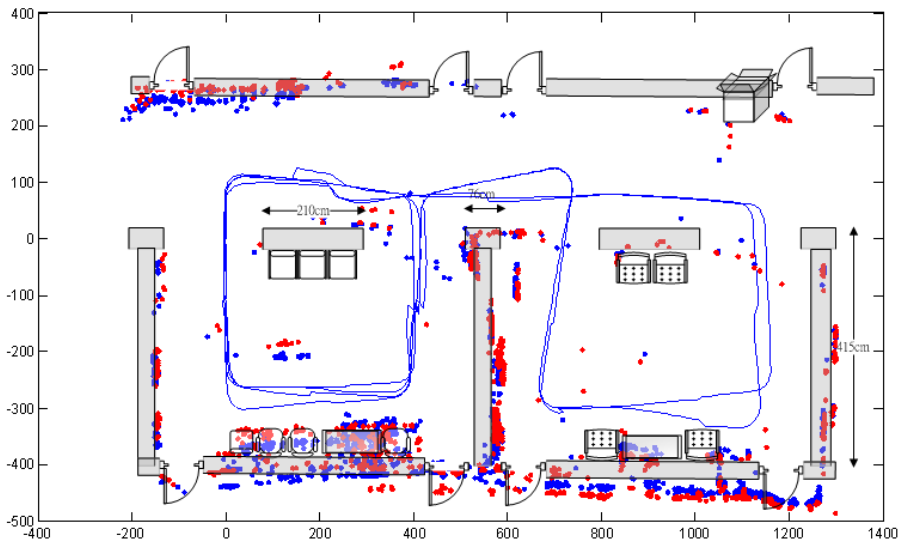
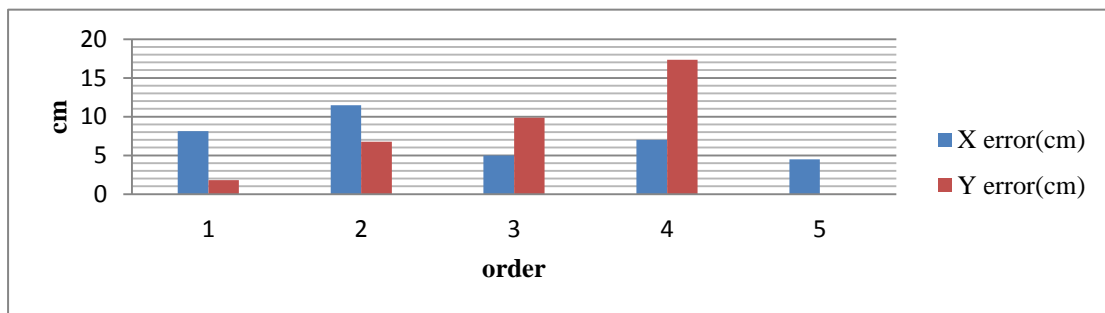


圖 5.28 長距離機器人運行實驗結果與實際環境配置之對照圖

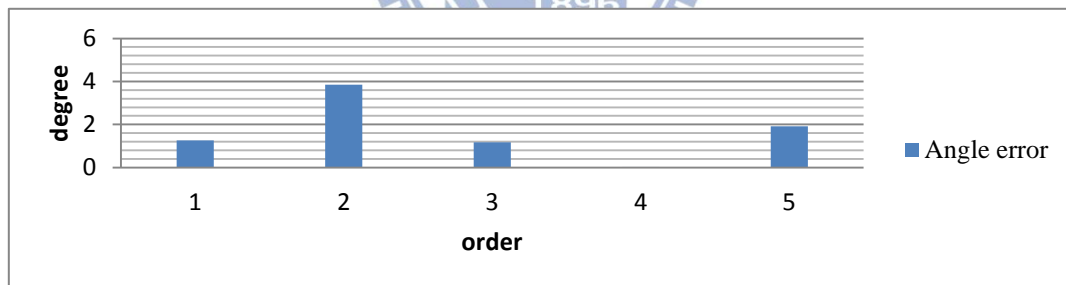
表 5.8 為機器人於本實驗在室內環境行進約 82.8 公尺後原點附近之位置，表中分別詳列機器人環境中實際位置與定位系統估測之狀態資訊，以及估測和實際情況之誤差數值，實驗結果顯示，機器人經過長距離運行，透過環境資訊以定位系統修正，在機器人回到原點附近估測結果與實際環境位置的差距，平均 X 方向誤差為 7.21 公分，平均 Y 方向誤差為 7.16 公分，平均角度差則為 1.64 度，圖 5.29 則以直條圖呈現機器人實際與估測狀態資訊之差異。

表 5.8 長距離實驗機器人實際與估測之狀態資訊

	Ground truth			EKF Estimation			Error		
	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$	x(cm)	y(cm)	$\theta(^{\circ})$
Experiment 1	-16.26	1.587	96.80	-24.39	-0.22	95.54	8.13	1.80	1.26
Experiment 2	-15.06	-10.73	97.30	-3.59	-3.98	93.46	-11.47	-6.75	3.85
Experiment 3	-1.25	-2.628	92.20	3.74	-12.48	91.03	-4.99	9.86	1.17
Experiment4	14.28	-13.01	92.80	7.29	4.33	92.83	7.00	-17.34	-0.03
Experiment5	-5.80	-5.20	90.00	-1.33	-5.17	91.91	-4.48	-0.03	-1.91



(a) 二維座標誤差



(b)

圖 5.29 室內長距離實驗座標誤差之直條圖

5.7 討論

從實驗的過程以及結果中可以看到使用影像作為環境資訊擷取的限制以及不足之處。在機器人移動路徑的選擇上，因為 Kinect 感測裝置取得的影像如同一般單眼攝影機，所取得的畫面僅有一個面向的資訊，因影像特徵點擷取並無法抵抗大幅的視角變化，使得機器人在重複行進於相同的環境時必須在相似的路徑上移動，比方說像直線長廊的環境，兩側的牆上取得的特徵點在去與回的行走方

式所得到的特徵有相當大的差異，若以地圖建構的方式分別建立去與回所得到的特徵點，會因為誤差的存在而讓建構出來的環境無法完全吻合，而若以地圖使用的方式拿對應區域地圖的特徵點訊息來使用，卻又會因為特徵點匹配的成功率低下而無法有效的修正機器人狀態。

在影像環境資訊擷取的限制上，在環境重複性高或是單調色彩的環境是使用影像特徵會有較多疑慮的地方，這些環境這會使特徵點匹配的結果錯誤或是沒有環境特徵可以使用，像是整面白色牆壁的房间或走廊，在以影像特徵點作為環境資訊建立上就會有所困難，特徵點數量稀少的結果，像是在 5.6 的實驗，在圖 5.30 中 a 所標記的實線圓圈的部分就因為所得到的環境特徵點數量稀少，而影響後續環境建立的準確度(這裡特徵點稀少的原因主要是因為機器人行進方向與 Kinect 距離感測的接收上難以得到牆上的特徵點深度資訊，因為本論文所使用的特徵點為帶有三維座標資訊的特徵點，無法得到距離資訊的影像特徵點會被捨棄，使得該段特徵點數量稀少)。而圖 5.30 中 b 所標記的虛線圓圈的部分，則是因為日光燈在的地面反射所擷取的特徵，這類的特徵點的不穩定性高，是應該要被去除的部分。

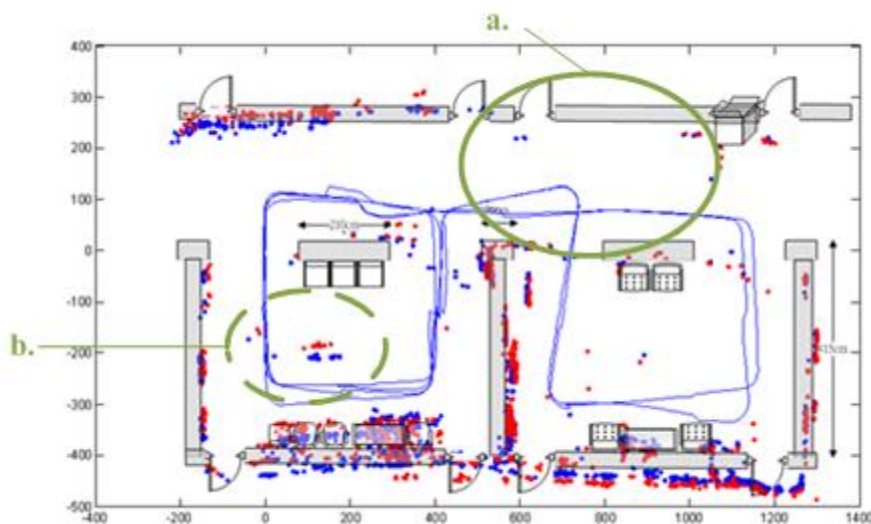


圖 5.30 標記影像特徵點對實驗結果影響之不穩定部分

第六章 結論與未來展望

6.1 結論

本論文以 Kinect 深度攝影機為感測裝置，結合視覺感測裝置與距離感測裝置，以 SURF 演算法對影像作特徵點擷取，對應深度資訊計算特徵點之三維座標，取得環境中具獨特性之特徵點並能快速取得對應之距離關係，這些特徵點訊息透過 EKF 定位系統演算法修正機器人狀態並更新地圖資料庫。在機器人持續運行之同時利用區域路徑範圍的判斷，將環境區隔為數個範圍較小的區域地圖降低定位系統的運算複雜度，達成機器人同時定位與地圖建立之即時運算。最後，透過地圖融合演算法，以地圖間共同特徵之差異整合地圖間資訊並修正區域融合後機器人狀態，達成具備即時運算且具完整地圖結果之機器人定位與地圖建立系統。

本論文透過實驗證實同時定位與建立地圖結合地圖接合之作法實現於雙獨立驅動輪式機器人之功效，以本論文所設計之方形路徑實驗，在已建立過之環境中運行時，耗費運算量最大之區域每次約需 1.2 秒完成所有定位系統之運算，在不同區域間運算的結果整體平均每次約需 646ms，當機器人應用於長距離之機器人定位與導航之實驗，機器人於一約 16m X 7m 之環境行走 83m 後，當機器人回至原點附近時的實際位置與估測狀態平均 X 方向誤差為 7.21 公分，平均 Y 方向誤差為 7.16 公分而平均角度差則為 1.64 度。

6.2 未來展望

影像以 SURF 取得之特徵點在抗尺度、光源變化影像旋轉以及運算速度等皆有極佳的表現，然對物體之視角有較大不同時，SURF 特徵點並無法有效的判斷並比對相同之環境資訊，若以增加感測器之視角以加強感測範圍，則又必須增加定位系統所需之運算成本，降低機器人即時運算之效能。這使得機器人在相同的環境以不偏離取得環境資料的觀測角度太多的周圍移動，大幅降低機器人 SLAM 應用於實際情境之可行性，若能以影像取得在視角變化下之相同環境資訊，則應能有效強化以視覺感測裝置在 SLAM 於實際環境下之應用。

本論文以深度攝影機 Kinect 結合視覺及距離感測裝置之作法是以視覺感測器為主取得影像特徵點，距離感測器為輔用以計算影像特徵點之三維座標，若以距離感測裝置取得環境結構與分佈特性做為第二種環境訊息結合於定位系統，則定位系統在特徵點稀少視覺系統難以發揮之場所將能有所依據，以兩者相輔相成可望讓機器人 SLAM 更廣泛應用於各種場所。



參考文獻

- [1] Hugh Durrant-Whyte and Tim Bailey, “TUTORIAL : Simultaneous Localization and Mapping : Part 1 ,” *IEEE Robotics & Automation Magazine*, Vol. 13, No. 2, pp 99-108, 2006
- [2] Tim Bailey and Hugh Durrant-Whyte, “TUTORIAL : Simultaneous Localization and Mapping : Part II,” *IEEE Robotics & Automation Magazine*, Vol. 13, No. 2, pp. 108 -117, 2006
- [3] 袁立德,“基於單眼視覺之機器人迴圈原點及同步定位與地圖建立,” 國立交通大學電機與控制系碩士論文, 2010.
- [4] Brian Williams, Paul Smith and Ian Reid,“ Automatic Relocalisation for a Single-Camera Simultaneous Localization and Mapping System,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2007, pp. 2784-2790.
- [5] 王兆戊,“全向式移動機器人同步定位與環境地圖建立,”國立交通大學電機與控制系碩士論文, 2008.
- [6] 黃富盛,“基於權向式影像之機器人同步定位與環境地圖建立,”國立交通大學電機與控制系碩士論文, 2008.
- [7] Olivier Koch, Matthew R. Walter, Albert S. Huang and Seth Teller, “ Ground Robot Navigation using Uncalibrated Cameras,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010, pp. 2423-2430.
- [8] Olivier Koch and Seth Teller, “ Body-Relative Navigation Guidance Using Uncalibrated Cameras,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009, pp. 1242 – 1249.
- [9] Cristof Schroeter and Horst-Michael Gross, “ A Sensor-Independent Approach to RBPF SLAM –Map Match SLAM applied to Visual Mapping,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* , Nice, France, 2008, pp. 2078-2083.
- [10] S. Thrun, D. Fox and W. Burgard, “A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots,” *Machine Learning and Autonomous Robots*, vol.31, pp. 29-53, 1998.
- [11] J. Tardós, J. Neira, P. Newman and J. Leonard, “ Robust Mapping and Localization in Indoor Environments using Sonar Data, ” *Int. J. Robotics Research*, vol.21, pp. 311-330, 2002.
- [12] N. Ganganath and H. Leung, “ Mobile Robot Localization Using Odometry And Kinect Sensor, ” in *Proceedings of IEEE International Conference on*

- Emerging Signal Processing Applications (ESPA)*, Las Vegas, NV, USA, 2012, pp. 91 – 94.
- [13] Gert Koostra, Sjoerd de Jong and Lambert R.B Schomaker, “Using Local Symmetry for Landmark Selection,” in *Proceedings of International Conference on Computer Vision Systems* , Liege, Belgium, 2009, pp. 94-103
- [14] Gert Koostra and Lambert R.B. Schomaker, “Using Symmetrical Regions of Interest to Improve Visual SLAM,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009, pp. 930-935.
- [15] J. Shi and C. Tomasi, “Good Features to Track,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 1994, pp. 593-600.
- [16] David G. Lowe, “ Distinctive Image Features from Scale-Invariant Keypoints, ” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [17] H. Bay, T. Tuytelaars and L. V. Gool, “SURF: Speeded Up Robust Features” in *Proc. of the 9th European Conf. on Computer Vision*, Graz Austria, 2006, pp. 404-417.
- [18] L.M. Paz, P. Jensfelt, J.D. Tardós and J. Neira, “EKF SLAM Updates in $O(n)$ with Divide and Conquer SLAM,” in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2007, pp. 1657-1663.
- [19] Carlos Estrada, J. Neira and Juan D. Tardos, “ Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments, ” in *IEEE Transactions on Robotics and Automation*, Vol.21 No.4, pp. 588-596, 2005.
- [20] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid and Juan Tardos, “An Image-to-map Loop Closing Method for Monocular SLAM,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 2053-2059.
- [21] Adrien Angeli, Stéphane Doncieux, Jean-Arcady Meyer and David Filliat, “Real-Time Visual Loop-Closure Detection,” in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008 ,pp. 1842-1847.
- [22] <http://www.play.com/Games/Xbox360/4-/10296372/Project-Natal/Product.html>
- [23] [http://www.primesense.com/\(Primesense\)](http://www.primesense.com/(Primesense))
- [24] Christopher Evans , “Note on the OpenSURF Library, ”, 2009.
- [25] [http://75.98.78.94/\(OpenNI\)](http://75.98.78.94/(OpenNI))
- [26] S. Thrun, D. Fox and W. Burgard, *Probabilistic Robotics*, Massachusetts Institute of Technology, 2005.
- [27] Jose’ Neira, Juan D. Tardos, and Jose A. Castellanos , “Linear Time Vehicle

- Relocation in SLAM, ” in *Proc. IEEE International Conference on Robotics and Automation(ICRA)*,Taipei, Taiwan, 2003, pp427-433.
- [28] Joan Sola, “ Consistency of the Monocular EKF-SLAM Algorithm for Three Different Landmark Parametrizations, ”in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010, pp. 3513-3518.
- [29] Chanki Kim and Wan Kyun Chung, “Delayed Resampling in a Rao-Blackwellized Particle Filter SLAM for Consistent Loop Closures,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Nice, France, 2008, pp. 2084-2090.
- [30] Mark Cummins and Paul Newman, “Probabilistic Appearance Based Navigation and Loop Closing,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, 2007,pp. 2042 - 2048.
- [31] Mark Cummins and Paul Newman,“ Accelerated Appearance-only SLAM, ”in *Proc. of IEEE International Conference on Robotics and Automation(ICRA)* , Pasadena, CA, USA, 2008, pp.1828-1833.
- [32] Jungho Kim, Ouk Choi and In So Kweon, “Efficient Feature Tracking for Scene Recognition using Angular and Scale Constraint,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Nice, France 2008, pp.4086-4091.

