

智慧型互動機器人設計

The Design of Intelligent Interactive Service Robot

江叔盈

蔣藝荃

楊心蝶

Shu-Yin Chiang

Yi-Quan Jiang

Hsin-Tieh Yang

王家慶

李聿宸

Chia-Chin Wang

Yu-Chen Lee

銘傳大學電腦與通訊工程學系

Department of Information and Telecommunications Engineering,

Ming Chuan University

摘要

本研究以開發智慧型多功能互動機器人為目標，達成陪伴及娛樂等互動功能。由於機器人需要獲取所處環境信息，故本研究以 Kinect 深度影像為視覺系統平臺，通過筆記型電腦進行影像處理的運算，將分析影像處理結果應用於機器人行為的規劃，並搭配全像輪、高功率馬達、FPGA 開發板及以及控制板作為移動系統，設計組成下半身為輪型平臺，上半身為人型的智慧型互動機器人。本研究使用 Kinect 的深度影像資訊，繪製出 3D 即時影像地圖，整合機器人移動定位系統，透過影像、定位系統、陀螺儀參數等將建置出室內環境地圖，使機器人在移動過程能達成避障與建立即時環境地圖之功能。之後，機器人能透過 Kinect 對人進行骨架偵測追蹤與人臉表情識別，讓機器人能根據人的表情及身體姿勢，與人進行互動之行為，達到智慧型互動機器人之設計。

關鍵詞：智慧型互動機器人、Kinect、機器人定位、室內環境地圖、避障

Abstract

This study aims to develop the intelligent interactive multi-functional robot to provide companion and entertainment. Because the robot needs to obtain the environmental information, this study uses Kinect depth image as the visual system platform and calculates results through the computer image processing operations. Then, the analyzed the image processing results is applied to the robot behavior planning, and with the integration of omnidirectional wheels, high power motors, FPGA development board and control panel as a mobile system, a lower body for the wheel platform, the upper body of the intelligent interactive robot is designed. In this study, we use the depth image information of Kinect to draw the 3D real-time image map, integrate the robot mobile positioning system, and build the instant indoor environment map to achieve the obstacle avoidance. The robot through Kinect skeleton detection and human facial expression recognition to interact with people and accomplish the task of the intelligent interactive robot.

Keywords: Intelligent interactive robot, Kinect, robot positioning, indoor environment map, obstacle avoidance.

1. 研究動機與研究問題

現今台灣已經邁入高齡化社會，再加上少子化，有愈來愈多的老年人處於獨居的生活型態，平常有子女或親友的陪伴及照護，在健康、經濟以及精神層面有著依靠，但是隨著高齡人口不斷升高，對於工作人口所帶來的經濟需求及安養負擔也愈來愈大。為此，我們藉由此機器人來陪伴老年人，能夠熟悉家庭內的環境，適時避開障礙物，並期望能感測情緒達到安撫及愉悅他人的效果，使機器人能夠步入家庭裡，扮演著像家人般的角色。

為讓機器人更加智慧化，本研究的主要設計一台機器人，使用馬達、全向輪、FPGA 控制板等達成硬體設計，組合成下半身為輪型平臺，上半身為人型的智慧型互動機器人。再經由 Kinect 的深度影像就可讓機器人能夠在未知的環境中自動地規劃路徑、自動行走整個環境，自動繪製出室內地圖，而不需要遙控機器人或人為輸入室內地圖和設定。為了能讓機器人擁有避障的基本功能，機器人需要自行判斷在眼前影像內的物件中辨別何為障礙物，何為可通過的地面。經過分析所得的深度影像，我們發現僅需要從物件中找出地面部分，其餘的物件便是障礙物，因此我們可以通過過濾地面來分辨障礙物。在過濾完地面之後，我們需要將障礙物分類成一個個獨立的地標，由此我們實作 Union-Find 演算法來分類障礙物。當我們把每張深度影像轉換成俯視圖時發現，每張俯視圖中的障礙物無法顯示厚度，所以將讓機器人在室內移動，從不同方向觀察物體來得到物體完整厚度資訊，使用轉換而來的俯視圖堆疊拼接成完整室內地圖。因此透過 SLAM(Simultaneous Localization And Mapping)演算法，實現機器人移動中同步定位與地圖構建，在構建的環境地圖基礎上並搭配路徑搜尋演算法，讓機器人在所處環境快速規劃路徑並能避免機器人與環境中的障礙物發生碰撞。

在機器人與人類互動方面，我們將利用機器人 D-H 參數及逆向運動學求得特定姿態為目標位置時，各軸馬達之角度，在用正向運動學將馬達角度回推目前手臂末端點正確位置進行回授，再與影像及策略做整合使手臂能夠跟人類互動而作擺動。

2. 文獻回顧與探討

2.1. 機器人定位系統

在構建室內地圖時，機器人需要定位系統進行自我定位。機器人定位系統，可以透過場地視覺建立環境比對進行定位，如白線定位匹配法[1]，或是環境之物件匹配法[2-4]。例外，[5-6]使用蒙地卡羅演算法來做位置估測與追蹤機器人位置，或是粒子濾波器進行定位[7]。也可以透過機器人平台中的馬達編碼器和陀螺儀的回授[8]，得到近似於里程計的功能進行定位。另外，Chiang 等[9]提出光學回授定位系統，該定位系統利用滑鼠的光學回授取得機器人位移量進行定位。本研究採用馬達編碼器和陀螺儀的回授方法並做出調整：我們讓機器人均勻緩慢前進以減少速度所帶來的誤差，並且整合 SLAM 演算法[10,11]輔助定位並消除誤差。

2.2. 使用 FastSLAM 演算法的地圖建置

Murphy [14]等曾提出 FastSLAM 後驗可由機器人路徑事後機率(path posterior)和地標事後機率(landmark posterior)組成，機器人路徑事後機率由粒子濾波器(Particle Filter)估計，而地標後驗由擴展後的卡爾曼濾波器(Extended Kalman Filtering, EKF)估計，其中用一個 EKF 負責一個地標位置，估計路徑的粒子濾波器中的每個粒子運行著多個 EKF。Montemerlo [11]等曾提出粒子方程式(1)：

$$S_t^{[m]} = \left\langle s^{t,[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \right\rangle \quad (1)$$

在方程式(1)中，[m]表示粒子的索引(這裡為第 m 個索引)， $S_t^{[m]}$ 表示在時間 t 的第 m 個粒子， $s^{t,[m]}$ 表示第 m 個粒子在時間 t 的路徑估計， $\mu_{n,t}^{[m]}$ 和 $\Sigma_{n,t}^{[m]}$ 分別為在路徑估計 $s^{t,[m]}$ 下第 n 個地標的高斯平均值和共變異數(covariance)。粒子濾波器在過濾時，將每個粒子加權，然後用權重取樣推導出新的粒子集合。圖 1 和圖 2 為構建的室內地圖[11]，紅色的粒子分佈密度高的為高機率區。在圖 2 中，經過濾波，事後機率被收斂到近似單峰分佈。

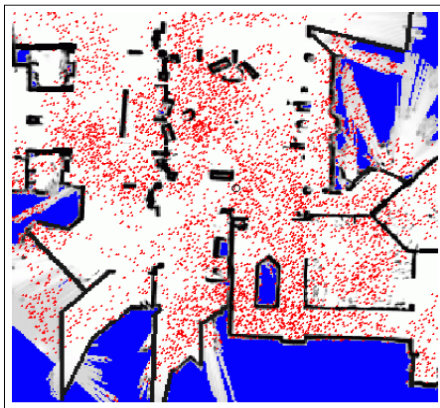


圖 1 資訊時粒子分佈



圖 2 粒子濾波器過濾後粒子分佈

3. 研究方法及步驟

本研究是以中型足球機器人移動平臺做為底層架構，其中包括了四顆直流馬達、全向輪、Maxon 馬達控制晶片、Kinect V2 以及 MyRio 嵌入式開發版，並以筆記型電腦作為運算中樞。Kinect V2 為機器人眼睛，結合自製的機械手臂(由 MX28 所組成)當作機器人雙手，以便於協助人類進行簡單作業。此外，移動平臺上方使用三支鋁擠型(40mm*40mm)做為支撐架，可將手臂與 Kinect V2 固定，本研究之硬體架構圖如圖 3、圖 4、圖 5 所示。



圖3 自製機器手臂



圖4 機器人硬體架構圖

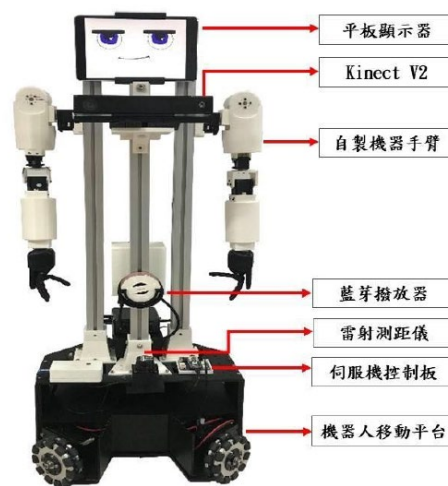


圖5 機器人外觀

3.1. 偵測地面

在已知 Kinect 高度下，查詢深度影像的每一個像素點在 Kinect SDK 的 Camera Space3D 座標系中對應的 Y 值， Y_{ij} (物件相對於 Kinect 所在平面的垂直距離值)，用這個 Y_{ij} 值與 Kinect 所在高度值 Y_k 做比較，若兩值相差在一個容忍範圍，則對應的像素點被認為是地面部分，如公式 (2)。圖 6 為 RGB 影像，圖 7 則為該影像透過 Camera Space3D 座標系獲得地面資訊(綠色部分)。

$$\text{If } |Y_{ij} - Y_k| < \text{threshold, } Y_{ij} \text{ is ground} \quad (2)$$



圖6 與右圖相對應的RGB影像

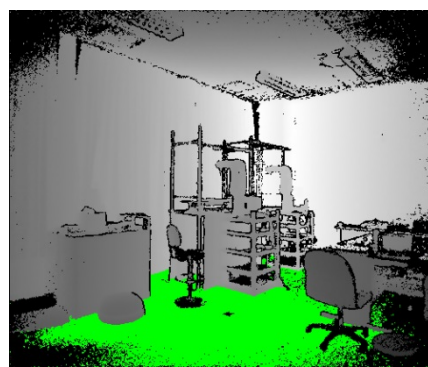


圖7 以Camera Space3D座標系為偵測依據的地面過濾後深度影像

3.2. 繪製俯視圖

本研究利用 Kinect for Windows SDK v2 將深度影像中的深度值轉換成攝影機空間坐標系 (Camera Space) 的標記法，如圖 8 所示，Camera Space 將 3D 可視區域中的點用 X, Y 和 Z 值表示，其中 3D 坐標系的原點為 Kinect 本身，Z 值為深度值，Y 軸垂直於 Kinect 所在平面，其數值可以推斷出障礙物各點的高度，X 軸垂直於 Y 軸和 Z 軸所在平面。將深度影像轉換到 Camera Space 3D 坐標系統，則可通過這個 3D 坐標系計算深度影像中物件的寬度、長度和角度等繪製俯視圖所需資訊。本研究在俯視圖中只繪製限制機器人運動的物件部分(即高度為 1.6 公尺以下的)

定義為障礙物，根據每點的 X 值和 Z 值分別決定該點在俯視圖的橫軸和縱軸位置。將圖 9 深度影像，透過俯視圖繪製結果，如圖 10 所示，綠色部分為過濾出的地面部分，除綠色以外的彩色色塊為不同的障礙物，黑色區域為未知地帶。

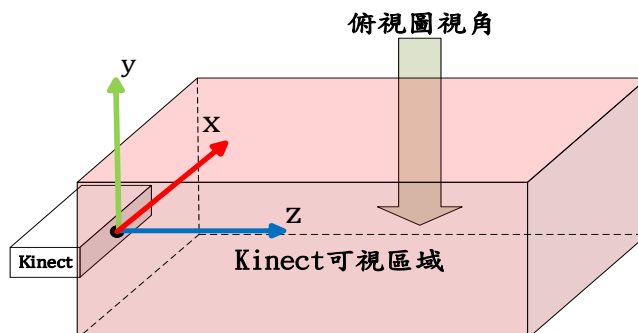


圖8 CameraSpace3D坐標系和俯視圖視角示意圖

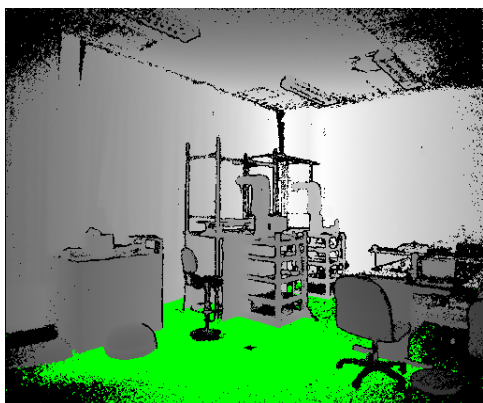


圖9 與圖10對應的深度影像

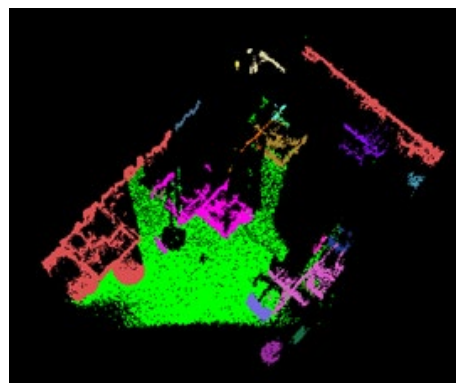


圖10 俯視圖結果

3.3. 構建室內地圖

室內地圖意指機器人所處完整室內環境的俯視圖，包含了所處環境下的地面及所有障礙物，由機器人遍歷該環境後產生。為構建室內地圖，除了需要每個時刻的障礙物資訊，還需要機器人的定位及 Kinect 的正面朝向。本研究利用現有的馬達編碼器定位及陀螺儀獲得機器人位置座標和 Kinect 的正面朝向，並結合障礙物資訊自行推導出能夠將每一時刻的深度影像轉換成室內地圖的方程式。

圖 11 為本研究分析出的室內地圖的示意圖，藍色矩形為機器人，A 點為機器人座標點，E 點為機器人觀測到的某一物體座標點，l1 和 l2 分別垂直於地圖 X 軸和機器人面朝，此刻從陀螺儀獲知機器人順時針旋轉了角度 θ (前一刻機器人朝向平行於 y 軸)， X_c 和 Z 分別為 Camera Space 的 X 值和 Z 值。假設已知 A 點座標、 θ 、 X_c 和 Z，為求出物體在地圖上的位置 E 點，需要計算物體 E 相對於機器人位置的 X 軸和 Y 軸偏移量(即 AD 兩點距離和 DE 兩點距離)。本研究運用以上的幾何計算，在如圖 11 所示之坐標系上用方程式(3)、(4)、(5)推導出方程式(6)和(7)。

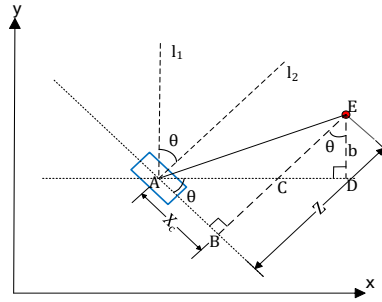


圖11 室內地圖坐標系，其中紅色E點為物件位置，藍色方框為機器人，A點為機器人位置

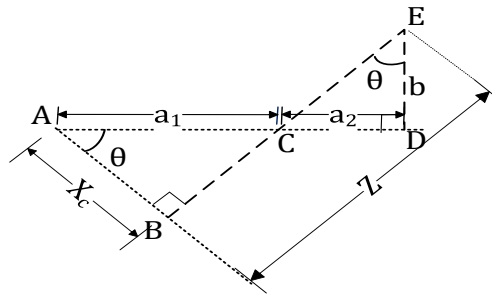


圖12 室內地圖坐標系圖11之局部放大圖

$$a_1 = \frac{X_c}{\cos \theta} \quad (3)$$

$$a_2 = (Z - a_1 \sin \theta) \cdot \sin \theta \quad (4)$$

$$b = \left(\frac{Z}{a_1} - \sin \theta \right) X_c \quad (5)$$

$$\Delta x_{map} = a_1 + a_2 = X_c \cos \theta + Z \sin \theta \quad (6)$$

$$\Delta y_{map} = b = Z \cos \theta - X_c \sin \theta \quad (7)$$

其中，推導出的 Δx_{map} 和 Δy_{map} 即物體相對於機器人位置的 x 軸和 y 軸偏移量。最後，機器人的座標加上物體偏移量，即可得到物體在環境地圖上的座標。

機器人位置	物體相對機器人位置偏移量	物體位置
(x_R, y_R)	$(\Delta x_{map}, \Delta y_{map})$	$(x_R + \Delta x_{map}, y_R + \Delta y_{map})$

3.4. 偵測與追尋人體骨架

在取得深度影像的基礎上，我們利用了 Kinect SDK 中 Body 這個介面類別，通過分析深度資訊抓取人體上的 25 個關節點(如圖 13)，抓取到的每個關節點會對應到 Camera space 坐標系中的一組 X、Y、Z 值。

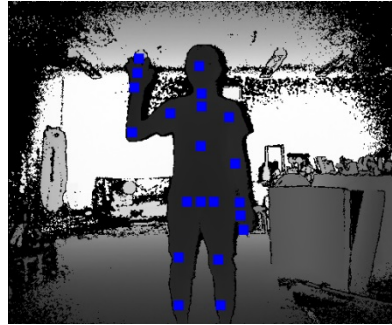


圖13人體骨架影像

機器人預設將脊椎中央這個關節點的位置視為人體位置（若脊椎中央這個關節恰巧偵測不到，機器人會尋找其他偵測到的偏人體中央的關節點）。在追尋人體骨架的狀態下，機器人會一直跟著人走，時刻將人體骨架保持在影像畫面中間，同時與人保持半米的安全距離。我們利用人體脊椎中央這個關節點的 Z 值取得人離機器人的距離，單位為公分。為了確定人與機器人方位的關係，我們定義機器人的正前方為 90 度，正後方為 270 度。人相對於機器的方位由方程式(8)表示。其中， P_Z 與 P_X 分別為一關節點的 Z 值和 X 值，若計算出來的角度 θ 範圍為 -90~0 度，我們會將角度正規化到 270~359。

$$\theta = 90 - \tan^{-1}\left(\frac{P_X}{P_Z}\right) \quad (8)$$

3.5. SLAM 演算法

SLAM 演算法的全名為同步定位與建構地圖(Simultaneous Localization and Mapping),它的目的是讓機器人在移動過程中通過傳感器來自我定位，同時建構所處環境的地圖。本研究利用現有的馬達編碼器定位及陀螺儀獲得機器人位置座標和機器人的面朝角度，並用 Kinect 深度攝像頭或雷射測距儀來獲取影像。我們定義 x_t 為機器人在時刻 t 的定位； z_t 為時刻 t 所得到的傳感器資訊； u_t 為 x_t 到 x_{t+1} 的移動向量；m 為建構中的地圖。根據以上參數，我們用圖 14 所示的狀態圖來展示 SLAM 演算法建構地圖的流程：機器人從定位點 x_0 出發，在每次移動一段時間後，定位系統運算得出的位移量 u_t ，並根據上一刻的定位 x_{t-1} 與 u_t 估計出此時機器人定位 x_t ，機器人在每個定位點 x_t 都會得到一份 Kinect 或雷射的深度資訊 z_t ，而 z_t 的深度資訊為地圖 m 中的部分環境。因此我們根據對應的定位點來拼接多份影像，即可得到比較完整的地圖 m。SLAM 演算法可由方程式(9)表示：根據機器人所有移動向量 u_t 和傳感器資訊 z_t ，估計機器人在每個時刻下的定位 x_t 和地圖 m 的機率分佈。

$$P(x_{0:T}, m | z_{1:T}, u_{1:T}) \quad (9)$$

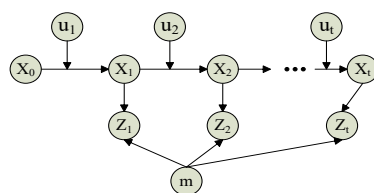


圖 14 SLAM演算法結構圖

本研究分別利用 Kinect 和雷射測距儀的俯視圖，結合 SLAM 演算法構建室內地圖。圖 15 為此實驗中現實的室內環境，圖 16 為利用 Kinect 俯視圖構建的室內地圖。Kinect 提供三維深度影像，雖然可以觀測物體的全貌，但是在精準度上沒有雷射高，因此使深度資訊 Z_i 含有較大的雜訊。在本研究中，我們將 Kinect 安裝在機器人胸口前，與地面有 90 公分的高度，由於 Kinect 視角(FOV)為 70 公分 x 60 公分，Kinect 下方會出現無法偵測的區域，這導致機器人即使走完整個環境，地圖中依然有阻礙機器人行進的未知區域。基於以上因素，我們將 Kinect 主要運用在物件分類與辨識和人體骨架辨識的應用上。

本研究利用雷射測距儀構建的室內地圖由圖 16 所示，圖中灰色的區域為未知區域，黑色部分為障礙物，白色部分為地面。雷射測距儀較廣的視角(240 度)使機器人在建圖時無需頻繁改變面朝方向就可建出較完整的地圖，同時雷射測距儀的雷射是平行於地面發射的，在沒有觸碰到障礙物的區段皆可視為地面，這不僅為實作 SLAM 演算法提供了便利性，也讓地圖更加完整美觀。相較 Kinect 構建的地圖，雷射的地圖更適合機器人路徑規劃。因此我們主要將雷射測距儀用於構建地圖和後文的雷射定位的研究上。

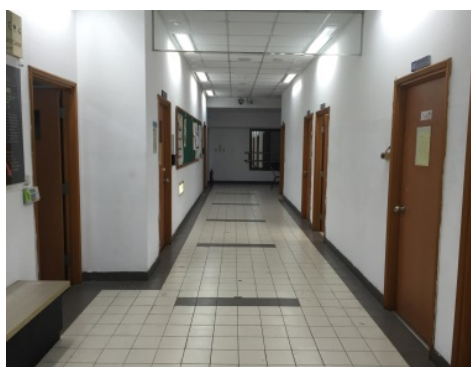


圖 15 雷射建圖時的彩色影像



圖 16 Kinect構建的室內地圖



圖 17 雷射構建的室內地圖

3.6. 手勢辨識

在本研究中我們定義揮手的動作為幾個要素組成，如圖 18 所示：圖中下方的圓點表示人體的肘關節，其他的圓點皆為手掌，揮手時手掌需要高於肘關節，且手臂需要左至右擺動，同時擺動幅度必須要超過揮手中間姿勢(如圖中藍色區域範圍)，揮手過程中必須左右擺動數次，並於額定時間內完成。因為揮手動作是一連續過程，須要觀察多幀影像畫面才能完成揮手的偵測，故在

我們設計的程式中，每一個新出現的骨架(即使用者)會被分配一套參數，此參數具備資料庫的功能，其中的成員包括：Counter、Timestamp、CurrentPosition、State。Counter 為手掌的左右擺動次數；Timestamp 為開始偵測揮手時的時間戳記；CurrentPosition 為目前手掌相對於軸關節的位置，有左、中或右三種；State 為目前揮手偵測的狀態，有未判斷、手勢判斷中及判斷成功這三種狀態。揮手偵測的演算法流程圖如圖 19 所示：其中，每當一個新的影像畫面產生，程式會先判斷手掌與肘關節是否被偵測到，於這兩個關節點偵測到的情況下檢查是否在手勢判斷中超過時間，透過檢查此時的系統時間戳記與此用戶的 Timestamp 變數的差值是否大於額定時間。若未大於設定時間，接下來檢查手掌關節點的 Y 值是否大於軸關節的 Y 值。圖中 Reset 函式表示將該用戶資料庫之變數還原到初始值，Counter 會歸零。若手掌高於軸關節，表示此時用戶可能正在揮手，故接著分析手掌相對於軸關節的位置，只有當相對位置與上一刻的位置改變時(如從中間姿勢變為位於手軸的左側)，才會被定義為揮動一次手臂，此用戶的 Counter 變數會增加一個單位，最後判斷 Counter 的值是否大於一給定閾值，若大於該閾值則揮手偵測成功，並觸發揮手完成之事件。

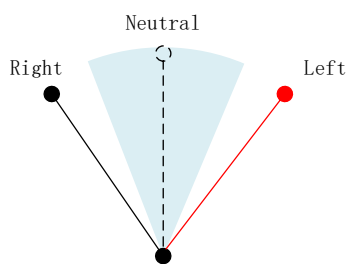


圖 18 揮手示意圖

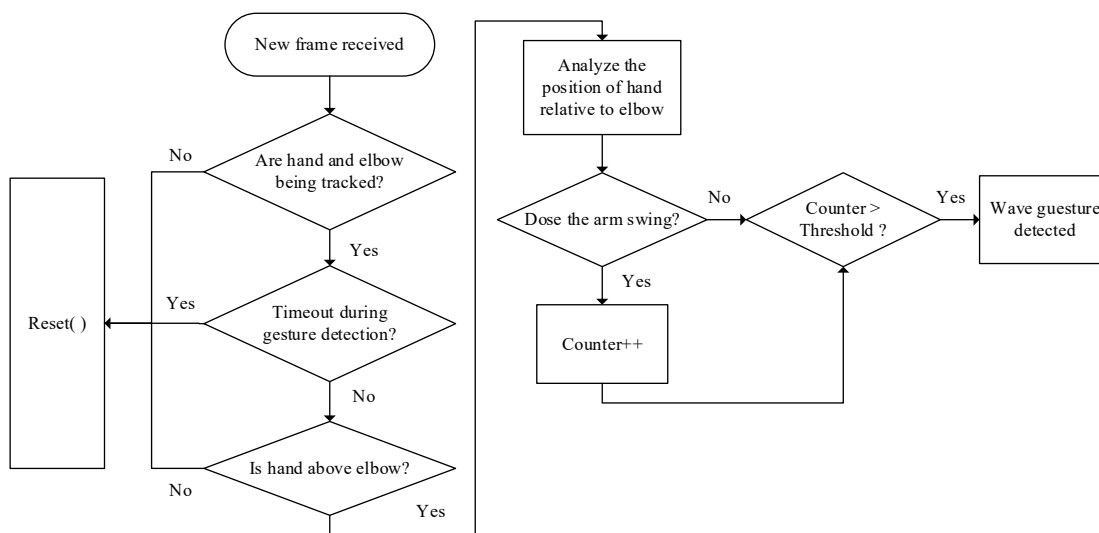


圖 19 偵測揮手姿勢的演算法流程圖

由於每個人的習慣不同，所以單純利用多層判斷式偵測特定手勢，而沒有基於大量手勢樣本歸納手勢特點的手勢偵測演算法只能偵測標準的手勢。為了讓手勢偵測不會因為每個人的動作習慣不同而降低偵測的正確率，我們藉助 Visual Gesture Builder(VGB)來建立完整的手勢辨識資料庫。在產生資料庫前，我們會請多個使用者來做特定的手勢，並把他們做手勢的過程錄製下來

存成樣本。然後我們把這些樣本輸入到 VGB，在樣本中標記該手勢出現的片段，VGB 會收集標記的片段進行機器學習，歸納總結出該手勢的特徵。VGB 根據分類器進行歸納，最後會列出前十個最重要的手勢特徵同時產生該手勢的辨識資料庫。程式會在手勢判斷前將資料庫讀入，在手勢判斷時利用資料庫分析當前骨架，然後得到一個手勢匹配值，程式會將這個匹配值與給定的閾值做比較，大於閾值即手勢判斷成功。圖 20 為 VGB 對測試影片用某手勢的資料庫偵測的結果，圖中的影片時間軸上藍色方塊為我們預先標記的手勢片段，綠色凸起線段為根據資料庫偵測到的手勢片段，從圖中可以看出偵測到的與被標記的片段幾乎一致，可見用資料庫可以準確地偵測這種手勢。

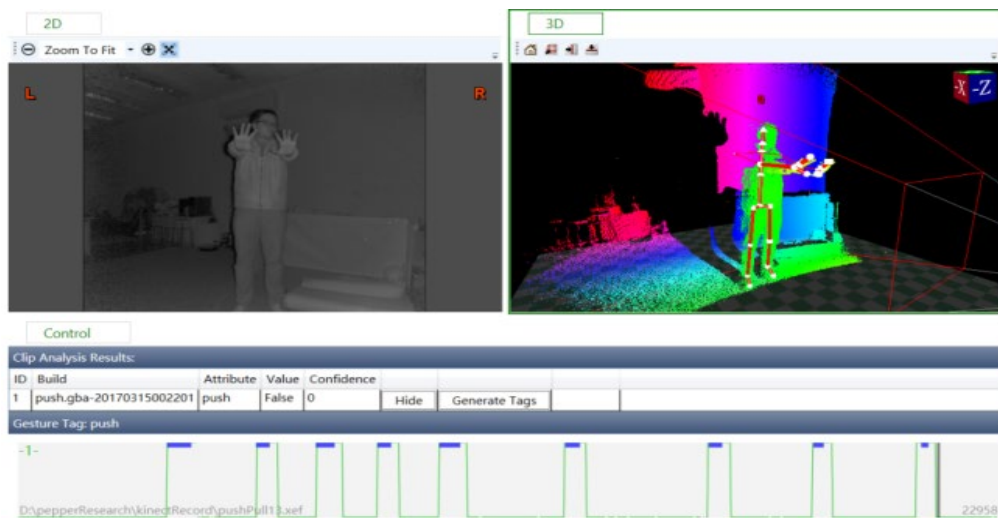


圖 20 VGB建立手勢辨識資料庫偵測特定手勢

3.7. 語音辨識

為了達成人們與機器人有更多的互動，本研究透過語音辨識使機器人聽見人類發出的語音內容而做出相對應的動作與回應。我們利用 Windows 桌面語音技術實作語音辨識(圖 21)，而電腦接收到語音指令後並傳回策略程式，進而做出相對應的動作。

首先，我們先自行作語意分類並建立資料庫(表 1)，程式執行時，先依序讀取資料庫內容至語音辨識器。而後，啟動語音辨識便開始傾聽，當辨識器收到聲音後，會在辨識系統內轉換成文字並與資料庫內的詞彙作匹配，假如與其中一類語意的詞彙相符合，電腦並會作出相對應的回應(表 1)，當回應結束後，即停止辨識以避免機器人再度聽到自身播出的語音內容而影響收音造成誤判，當回應執行結束後再次啟動語音辨識。為了加強語音辨識的準確度，我們也加入了關鍵詞與資料庫內字串作比對(表 2)，就可以提高辨識準確度。

Windows 語音辨識仰賴於 Microsoft Speech API，主要是讓使用者透過語音指令來操控電腦，並提供訓練功能，可以提高辨識的準確度，但比較有利於電腦的固定使用者，而我們則是利用此語音辨識來進行跟人類之間的互動，但因為每個人聲音不同的頻率及大小而會影響辨識的結果。

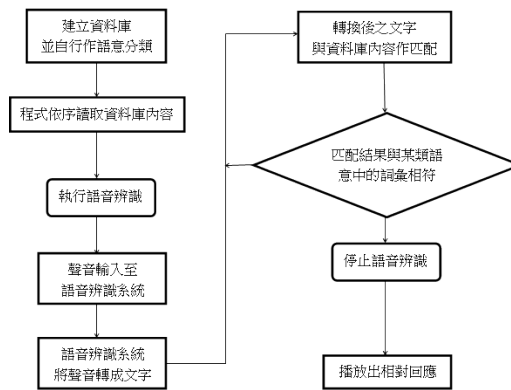


圖21 語音辨識流程圖

表1 資料庫內容與回應

語意分類	語句	回應
性別	a. 你是男生還是女生	我是男生喔，我是非人類的機器人寶寶。
	b. 你是人類嗎	
調侃	a. 你一點都不聰明	我才未滿一歲，之後會變得越來越好的。
	b. 你好笨	
身高	a. 你多高	我的身高有一米三!之後不知道會不會再長高。
	b. 你幾公分	
體重	a. 你多重	這是我的秘密，才不會跟你說!
	b. 你幾公斤	
年紀	a. 你幾歲	我還未滿一歲唷!之後會再成長的。
	b. 你年紀多大	
稱讚	a. 我覺得你很可愛	這樣我會害羞的!不過還是謝謝你的稱讚呀!
	b. 你好可愛	
吃	a. 你都吃什麼	我不需要吃東西喔!可以幫忙省錢。
	b. 你肚子會餓嗎	

表2 設定關鍵詞

語意類別	性別	調侃	身高	體重	年紀	稱讚	吃
關鍵詞	男生	聰明	多高	多重	年紀	可愛	吃什麼
	女生	笨	公分	公斤	多大		餓
	人類						

4. 結論

本研究自行完成一台機器人設計，利用 Kinect 深度攝影機和雷射測距儀來實現 SLAM 演算法。使用手勢和語音辨識演算法使機器人根據使用者的手勢和語音與使用者進行互動。其中，通過分析在 Kinect 深度影像上偵測人體關節點，我們實作出一套針對特定手勢的手勢辨識演算法。為

能偵測更多不同的演算法，我們藉助 Visual Gesture Builder 用機器學習的方法來建立多種手勢的資料庫。最後，本研究撰寫一套互動策略系統，整合以上的功能，控制多種功能的並行或循序地運作；並於伺服馬達控制板上編寫機器人手臂的多種動作，使機器人對使用者做出適當的行為反應，達成智慧互動型機器人之目標。

參考文獻

- [1] Shu-Yin Chiang, Xingzhi Guo and Hsien-Wen Hu, “Real Time Self-Localization of Omni-Vision Robot by Pattern Match System,” The second annual International Conference on Advanced Robotics and Intelligent Systems, pp. pp.46-50, June 2014.
- [2] Liu, Jianshen, Yin, Baoyong, Liao, Xinxing, “Robot self-localization with Optimized Error Minimizing for Soccer Contest. J. Comput. 6(7), 1485–1492, 2011.
- [3] Liu, B., Fan, J., Zhou, J., Li, K., Xie, Y., “A Self-localization Method through Pose Point Matching for Autonomous Soccer Robot based on Omni-vision,” The 9th International Conference on Electronic Measurement and Instruments, pp. 246–249, 2009.
- [4] Lauer, M., Lange, S., Riedmiller, M., “Calculating the Perfect Match: an Efficient and Accurate Approach for Robot Self-localization,” RoboCup 2005: Robot Soccer World Cup IX, vol.4020 of the series Lecture Notes in Computer Science, pp. 142–153.
- [5] Kim, H., Oh, T., Lee, D., Myung, H., “Image-based Localization Using Prior Map Database and Monte Carlo Localization,” The 11th International Conference on Ubiquitous Robots and Ambient Intelligence, pp. 308–310, 2014.
- [6] Heinemann, P., Haase, J., Zell, A., “A Combined Monte-Carlo localization and Tracking Algorithm for RoboCup,” Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1535–1540, 2006.
- [7] Hsu, C.-C., Wong, C.-C., Teng, H.-C., Li, N.-J., Ho, C.-Y., “Localization of Mobile Robots via an enhanced particle filter,” IEEE on Instrumentation and Measurement Technology Conference, pp. 323–327, 2010.
- [8] Shu-Yin Chiang, Chi-An Wei and Ching-Yi Chen, “Real-Time Self-Localization of a Mobile Robot by Vision and Motion System,” International Journal of Fuzzy System, DOI 10.1007/s40815-016-0220-y, 18(6):999–1007, Dec. 2016.
- [9] Shu-Yin Chiang, Che-Wei Chang and Kuan-Jheng Liao, “Self-Localization of Robot by Laser Sensor and Omni-Vision System,” International conference on Advanced Robotics and Intelligent Systems, August 31-September 2, 2016.
- [10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSlam: A Factored Solution to the Simultaneous Localization and Mapping Problem,” Proceeding 18th National Conference on Artificial Intelligence, pp. 593-598, 2002.

- [11] Michael Montemerlo, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association”, Carnegie Mellon University, Ph.D thesis, 2003.
- [12] Patrick Lester, “A* Pathfinding for Beginners,”
<http://www.policyalmanac.org/games/aStarTutorial.htm>
- [13] K. Murphy. Bayesian map learning in dynamic environments. In Advances in Neural Information Processing Systems (NIPS). MIT Press, 1999.
- [14] Zvi Galil, Giuseppe F. Italiano, “Data Structures and Algorithms for Disjoint Set Union Problems,” Journal ACM Computing Surveys Archive, Vol. 23 Issue 3, Sept. pp. 319-344, 1991.
- [15] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard, “3-D Mapping With an RGB-D Camera,” IEEE Transactions on Robotics, Vol. 30, No. 1, February 2014.