

無人商店導覽員

(1) 摘要

近年來科技的進步帶動無人商店的興起，無人商店旨在減少人力需求，卻有無法為客人處理找尋商品問題的疑慮。

本計畫著重於開發商場中具有導航服務的智慧機器人，我們對機器人的攝影機進行了相機校正以解決影像形變的問題並進行姿勢估計來取得感興趣區域，對話方面使用 NAOQI API 來實現語音辨識讓機器人能與人互動並得知消費者愈採買的物品，由於機器人座標與世界座標有著參考點不同的問題，需進行座標轉換以供機器人正常移動，路徑規劃則使用了 A Star 演算法，使整個購物過程更快速流暢，帶路過程中也同時利用聲納進行自動避障的功能，判斷障礙物並做出路徑修正。以此來開發成一台可供消費者最佳購物路徑並協助導航的賣場服務機器人。

(2) 研究動機與研究目的

近年來由於無人商店逐漸發展，已經有部分企業開始研發試驗無人商店的運行，並經由媒體進入人們的眼中，不再是憑空設想，例如國外的 Amazon GO 與便利商店 7-11 的 X-STORE，而這些無人商店以快速結帳、商品辨識做為商店特色，並降低了對人力的需求。無人商店雖然降低了人力的需求，卻也引發了一些人對無人商店問題處理的能力產生了質疑，因沒有服務員在場提供服務，導致在購物過程中遇到找不到商品陳列於何處時，沒有服務員能夠提供協助，進而引發購物時的負面體驗。此時如果在無人商店內有個負責帶路的機器人，能有效解決消費者找不到商品的問題，搭配語音辨識，聽取消費者的採買清單，並以路徑規劃使採購路線更順暢，使消費者購物體驗上升。

本計畫預計提供的服務有：

1. 提供消費者順暢的購物路線
2. 帶領消費者前往商品所在位置
3. 商品查詢
4. 與消費者做簡單互動

當消費者需要服務時，對著服務機器人講述欲採購商品，述說完後，機器人依照商品內容推得最佳路線並將帶領消費者前往採購。

(3) 文獻回顧與探討

目前市面上已經出現不少種有關於賣場服務功能的機器人，像是由日本企業合作研發的「Siriusbot」機器人及在特力屋服務的「XYZrobot」機器人(圖 3-2)。

「Siriusbot」機器人的服務功能包括方便溝通的中/英/日語音對話、帶領客

人前往目的地的導航功能、避開路途中障礙的自動避障及點貨等功能。儘管功能強大，但此機器人的導航功能只能設定一目的地，若消費者須購買多樣商品需要帶位時則略顯雞肋，無法給予消費者快速且順暢的購買路線。

「XYZrobot 整合型服務機器人」機器人的服務功能包括路徑規劃、呼叫線上客服、掃描商品資訊及自動避障等功能。但此機器人的操作方式為觸控式，需操控機器人上的平板來選擇所需功能，可能有衛生問題或是雙手提滿東西沒手操作的問題導致降低使用機器人的意願。

基於上面的問題，本計畫旨在開發一款站在消費者角度，結合帶路功能及路徑規劃，使消費者能夠以最短路徑、最快速的找到並採購完所需物品，使採購時間大幅縮短。



圖 3-1 「Siriusbot」機器人



圖 3-2 「XYZrobot 整合型服務機器人」

Pepper 機器人文獻：

感應器分布：

頭：麥克風×4、RGB 攝影機×2、3D 感應器×1

頭部感應器×3

胸：陀螺儀感應器×1、手：手部感應器×2

腳：聲納感應器×2、雷射感應器×6、腳部感應器×3

陀螺儀感應器×1、紅外線感應器×2

馬達：頭 X2、腰圍 X2、膝蓋 X1、肩膀 2X2(L/R)

肘 2X2(L/R)、腕 1X2(L/R)、手 1X2(L/R)

萬向滾輪 X3

目前使用例子：

國外：家樂福（Claye-Souilly 克萊蘇伊分店）

國內：亞太電信（三創數位生活園區、三軍總醫院）

圖 3-

3pepper 配置分布圖

Arma 程式庫-增強現實的模式跟踪：

這個 Arma 程式庫的作者為 Georgios Evangelidis, Phd，它主要是透過 openCV 來完成增強現實的模式追蹤(Pattern Tracking for Augmented Reality)的實作，而增強現實系統(Augmented Reality systems)這個技術是藉由通過在場景中適當顯示 3D

物件或標記來增強影像。

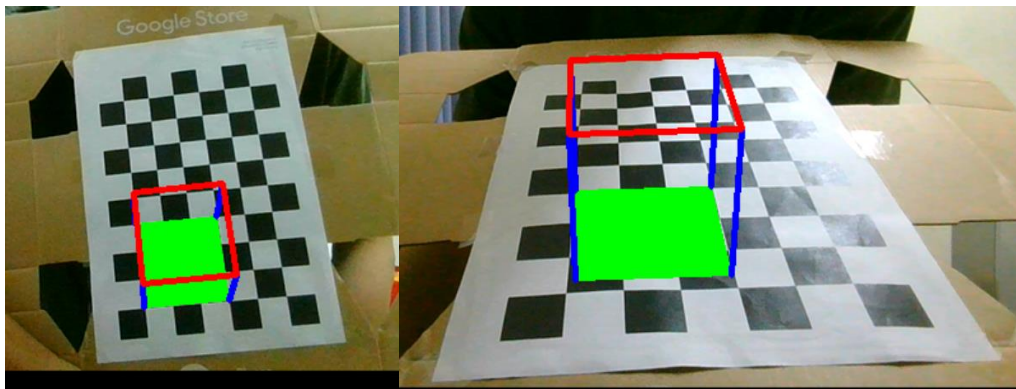


圖 3-4 增

強現實系統

這種系統的主要需求是解決外部影像座標的問題，又稱姿勢估計 (pose estimation)，也就是估計從對象坐標係到攝影像機坐標系的轉換。這些問題的解決依賴於 3D / 2D 對應。對於完整的擴充，還必須知道固有參數，因為任何 3D 虛擬對像都必須從相機坐標系進一步投影到圖像坐標系（圖像平面）。然而，固有參數是固定的，並且對於攝影像機/場景運動是不變的，所以可以被提取。

平面上印刷的二進制（黑白）圖案更易於跟踪，並已被廣泛用作增強現實應用程序中的標記。而這種系統的步驟如下：

基本配置：

- 內部相機矩陣
- 加載所需要追蹤的圖片

運作：

1. 在捕獲的影像中檢測圖案的位置 (corners) (如圖 3-5)
2. 將收到的原始扭曲影像做標準化(normalize)處理
3. 保留圖案的 ROI(region of interesting, 感興趣區域)，並將其 和加載的圖案進行比較。
4. 識別圖案後，測出"相機坐標系"和"影像坐標系"之間的轉換矩陣。該量測依賴於 2D / 3D 對應關係。
5. 使用上面的轉換矩陣進行進一步的影像增強(如圖 3-6)



圖 3-5 欲檢測的圖案

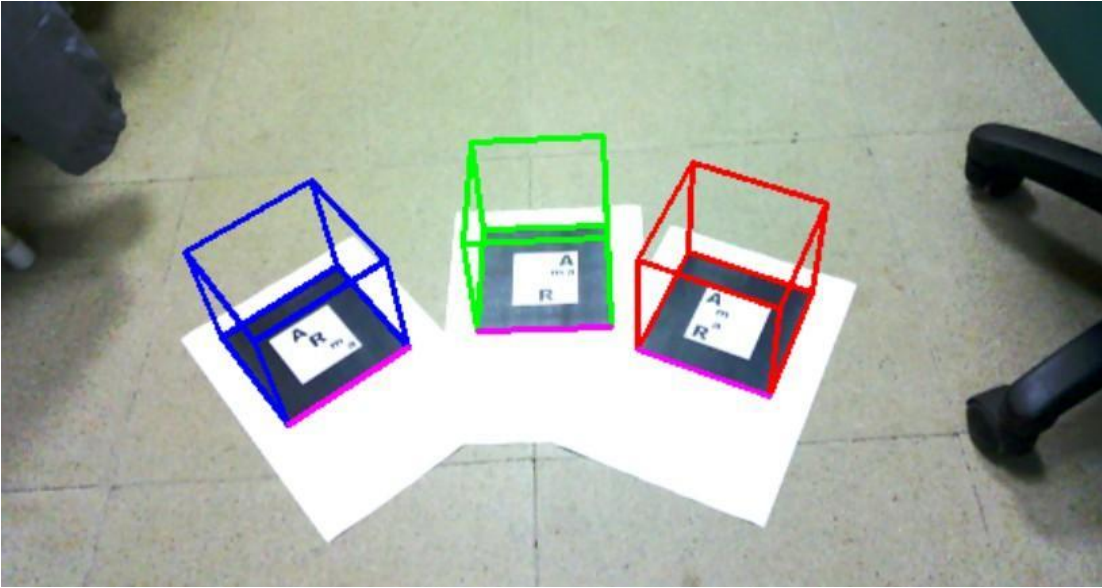


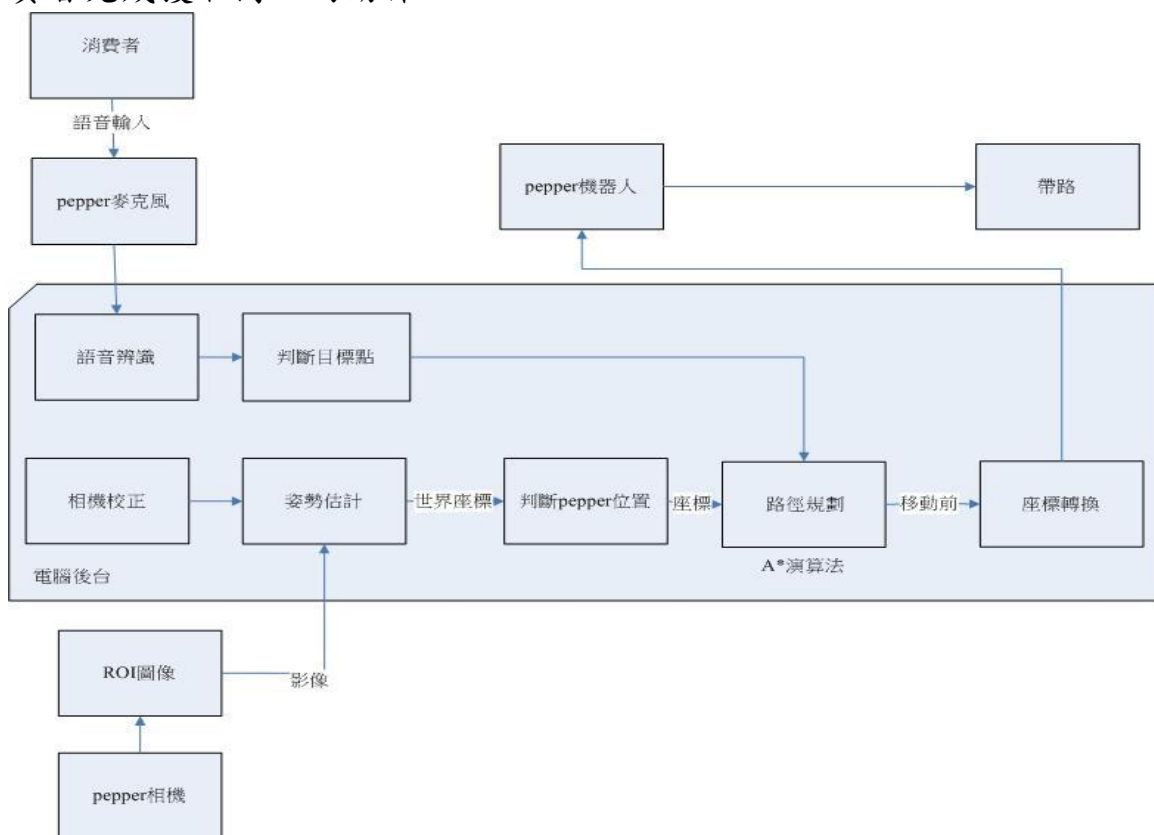
圖 3-6 影

像增強後的圖案

(4) 研究方法及步驟

1. 研究步驟

當消費者需要快速知道所需採買商品置於何處時，可直接至賣場放置的機器人做商品協尋帶路的服務，本計畫將設計語音對話像機器人傳達所需要採買的商品內容，當客戶站在機器人面前時呼喚【Pepper】，機器人將開始聽取客戶所需採買的商品內容，而獲取商品內容後，判斷機器人前方是否有可以偵測到的座標點，若沒有則先尋找座標點，偵測到座標點後判斷當前位置，並開始規畫所有獲取商品的所在地最佳路線，並依照路線將消費者依序帶往所標示的地區使消費者完成獲取商品的動作。



2. 語音辨識（判斷所需商品）

語音辨識使用 SOFTBANK 提供的 NAOqi 中的 ALDialog API，而同樣用在語音辨識中的 ALSpeechRecognition API，ALSpeechRecognition 在辨識句子時會去 Vocabulary 中尋找類似的單字，將其當成輸入結果，因此造成了辨別句子時容易導致辨識錯誤，而 ALDialog 則可以辨識整個句子提高辨識的成功率，並且回答預設的句子，增加與消費者的互動性。

```
topic: ~greetings
language: enu

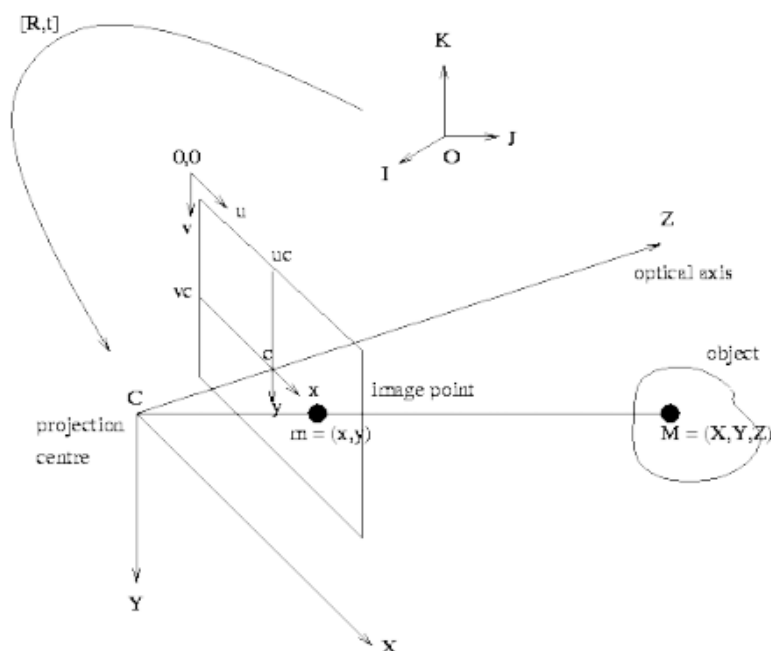
u: (Hello Nao how are you today) Hello human, I am fine thank you and you?
u: ({ "Good morning" } {Nao} did you sleep * well) No damn! You forgot to switch me off!
proposal: human, are you going well ?
  u1: (yes) I'm so happy!
  u1: (no) I'm so sad
```

以上是 SOFTBANK 提供的 ALDialog 範例

3. 相機校正

I. Camera Calibration

相機校正是將從外部感測到的世界座標系轉換成相對於照片上影像坐標系的過程。在圖像進行觀測時，為了確定在世界座標系空間中物體表面上某點的位置與其在圖像中對應點之間的相互關係，所以必須建構出相機成像的幾何模型，而幾何模型中的各種參數就是相機參數。相機參數可以分成外部參數、內部參數，外部參數是對從外部偵測到的世界座標進行轉換，內部參數是對外部參數轉換過的座標進行轉換，使之成為影像座標。



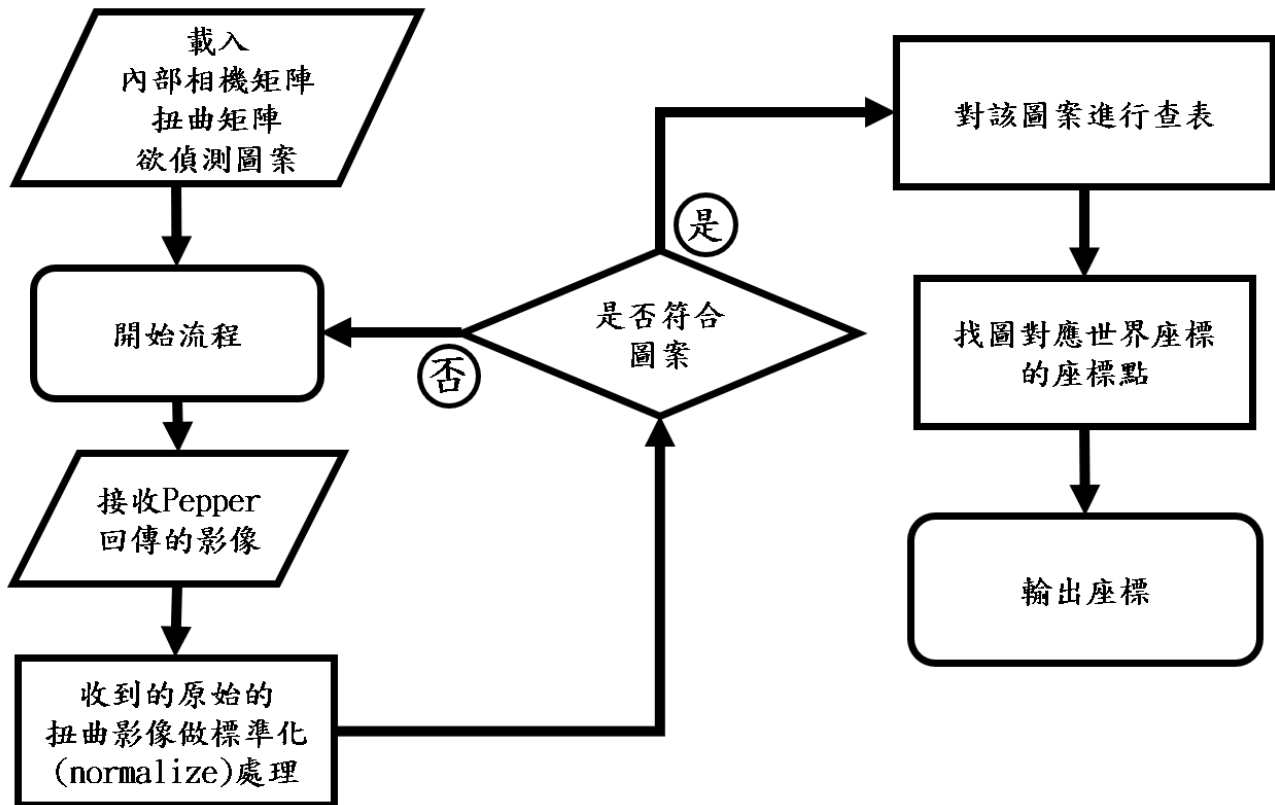
也就是世界座標(3D)經過相機外部矩陣的作用轉換成相機座標(3D)，而相機座標(3D)在經過內部參數矩陣的投影作用轉換成影像座標(2D)。根據針孔成像原理，假設Q點的世界座標為(X, Y, Z)，影像座標的座標為(U, V)，可以得到以下相機矩陣關係：

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & u_o \\ 0 & \alpha_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \begin{bmatrix} R & | & t \end{bmatrix} Q.$$

成像平面座標
內部參數矩陣
外部參數矩陣
世界座標

II. Pose Estimation(姿勢估計)

前置作業為將相機校準中得出相機內部矩陣和相機扭曲係數與要偵測的圖片載入。

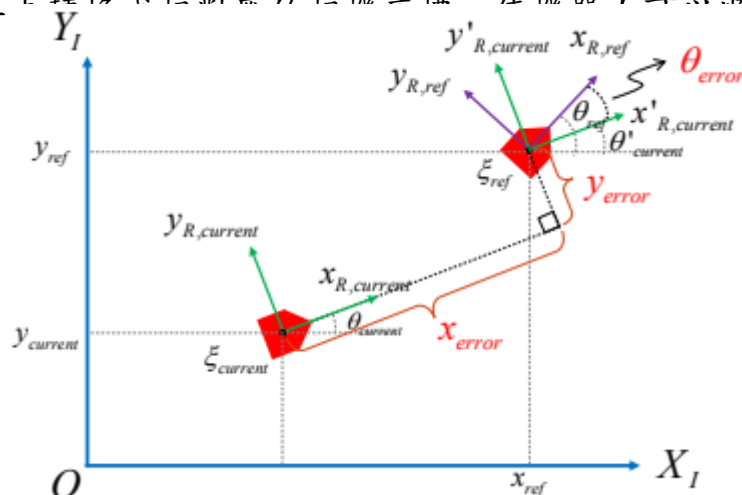


由 Pepper 機器人每次回傳到電腦的原始扭曲影像做標準化(normalize)處理，然後將處理過的影像去判斷是否有哪些區域符合先前載入的圖檔，若有符合的區域ROI(region of interesting，感興趣區域)，就進行更進一步的運算，判斷它為眾多圖中的哪張圖，並從座標字典去查表，定義它在世界地圖所對應的世界座標點，並輸出世界座標，以供之後的座標轉換做使用。

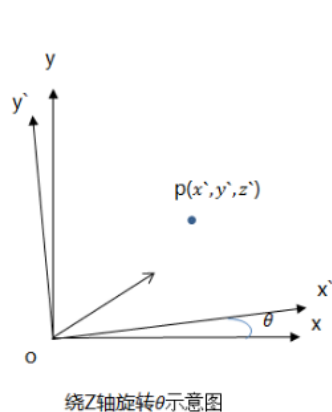
4. 座標轉換

世界座標與機器人座標本身並不相同，機器人座標是以自己本身為坐標系的起點(0, 0)為出發，而世界座標則是一個固定的絕對坐標系，因此機器人

座標與世界座標的起始點不同，而機器人要移動時，所輸入的移動距離是以機器人座標計算，而對於場地中所有物品固定的只有世界座標，因此會先判斷目標物的世界座標，再



而世界座標與相機座標可以用繞著不同的座標軸旋轉，得到相對應的旋轉矩陣。



$$\begin{cases} x = x' \cos \theta - y' \sin \theta \\ y = x' \sin \theta + y' \cos \theta \\ z = z' \end{cases}$$



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_1 \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

同理，绕x轴、y轴旋转 φ 和 ω ，可得到：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_2 \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \omega & 0 & -\sin \omega \\ 0 & 1 & 0 \\ \sin \omega & 0 & \cos \omega \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_3 \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

于是可以得到旋转矩阵 $R = R_1 R_2 R_3$

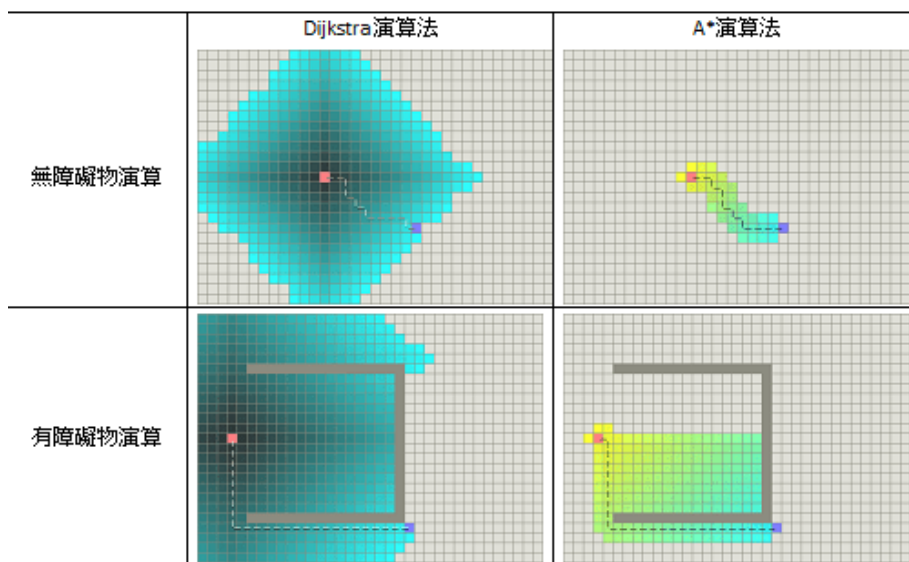
獲得選轉所需的角後取得旋轉矩陣 R，就可以獲得下列公式。

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T$$

5. 路徑規劃

A Star (A*) 演算法

A*是一種普遍用於路徑查詢的演算法，而相對於同樣運用在路徑規畫的 Dijkstra 演算法，Dijkstra 需要計算每個節點距離起點的總移動代價，同時還需要一個優先佇列結構，而當每個節點之間的移動代價為相等時，Dijkstra 將和廣度優先演算法一樣，因此採用 A*演算法



以上為 Dijkstra 與 A*演算法圖形化後運算過程中產生的比較圖

A*採用了 (Heuristic Estimate) 公式，而此公式避免了一些不必要的路徑排除，所以能用較高的演算效率計算出一條最佳結果。

HeuristicEstimate 的公式如下：

$$F(n) = G(n) + H(n)$$

n：目前節點

G(n)：從起始點到目前節點實際移動的距離

H(n)：目前節點到終點的估算值

F(n)：目前節點的評價分數總和

6. 避障功能

運用上述取得之路徑，搭配前面所用到的座標點判斷，來進行路徑尋訪。若途中遇到障礙，先以語音輸出【請借過】後判斷障礙物是否移除，若依然無法排除障礙，退回至前一個座標點並將無法到達的座標點移除，再次進行路徑規劃，直至到達所有所需前往之目標點。

(5) 預期結果

本計畫預期產出結果為：

1. pepper 機器人：

語音對話-利用 NAOqi API 中的 ALDialog 來實現人機對話

路徑規劃-利用 a star 演算法供消費者良好購物路線

導航功能-引領消費者前往購物

避障功能-以聲納判斷障礙物並進行迴避

2. 影像辨識：

相機校正-為了改善影像形變問題

姿勢估計-由看到影像取得向量值，以辨別位置點。

本計畫執行於 109 年 7 月至 110 年 2 月之進度規劃甘特圖如下：

任務名稱\月	7 月	8 月	9 月	10 月	11 月	12 月	1 月	2 月
Pepper 機器人								
語音對話								
語音輸入辨識								
座標轉換								
路徑規劃								
帶路功能								
避障功能								
影像辨識								
相機校正								
姿勢估計								
系統整合								
系統整合								

(6) 參考文獻

1. 「Siriusbot」 機器人:

<https://xtech.nikkei.com/atcl/nxt/mag/nc/18/062600056/062600002/>

<https://www.2lic.com/article/799509.html>

<https://reurl.cc/x0416N>

<https://zh.wikipedia.org/wiki/Pepper>

<https://reurl.cc/GrLeKd>

2. XYZrobot 整合型服務機器人:

<https://www.ithome.com.tw/newstream/103417>

<https://www.techbang.com/posts/41001-xyzrobot-integrated-service-robot-into-the-hola-taichung-mall-with-indoor-positioning-and-navigation-real-time-customer-service>

3. Pepper 機器人:

<https://reurl.cc/x0416N>

<https://zh.wikipedia.org/wiki/Pepper>

<https://reurl.cc/GrLeKd>

4. OpenCV 座標轉換:

<https://www.itread01.com/content/1548998303.html>

<https://opencv.org/>

5. 相機校正:

<https://silverwind1982.pixnet.net/blog/post/153218861>

6. 影像追蹤:

<https://sites.google.com/site/georgeevangelidis/arma/>

7. Arma 程式庫

<https://reurl.cc/R1Ev2e>

<https://sites.google.com/site/georgeevangelidis/arma/>

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_pose/py_pose.html#goal

8. A*演算法

<https://tedsieblog.wordpress.com/2016/07/08/a-star-algorithm-introduction/>

(7) 需要指導教授指導內容