

9

無人商店導覽員 ——幫你規畫如何逛商場

指導老師：曾士桓

組長：吳承翰

組員：任家駿、林柏修、黃冠瑜

Part 00

摘要

summary

摘要

前景：

無人商店的興起降低了人力需求，卻造成無法協助消費者解決商品尋找問題

主要功能：

相機校正以解決影像形變問題

姿勢估計取得座標位

以NAOqi API實現語音辨識

A Star演算法達成路徑規劃

聲納避障

Part 01

背景趨勢

Background

背景趨勢



您是否有過，在商店內找不到商品而苦惱、不知道商品位置在哪而像無頭蒼蠅亂竄或商場太大而迷路的經驗。

現今的商店已經有“無人化”的趨勢，而在無人商場中發生問題該找誰幫助？Pepper是一款可以滿足辨識商品、路徑規畫及帶路等功能的機器人。

背景趨勢



1. 無人商店中對於結帳或是商品辨識方面，已經有許多企業去著手研發。
2. 有許多無人商店的試營運，甚至是正式營運，例如：Amazon Go。
3. 無人商店通常標榜快速結帳，商品辨識。
4. 不清楚商品陳列於哪時，即使結帳在快速便利，也無法快速完成購物。

無人商店優缺點

優點	缺點
減少人力資源耗損	缺乏人情味
方便快捷結帳	假設遇到非預設問題，難以排除
方便系統式管理	若找不到所需物品無服務人員幫忙
方便客戶快速了解自身購買哪些商品與金額	

如何讓客戶適應無人商店，並且在遇到困難時，可以尋求幫助，也算是無人商店的挑戰之一。

Part 02

研究動機

Motivation



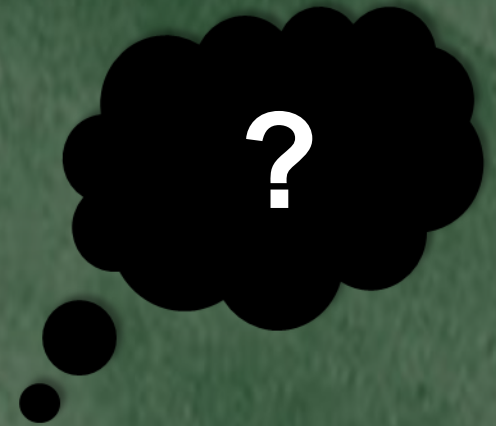
24h準確
快速服務

Service Any Time



減少
人力需求

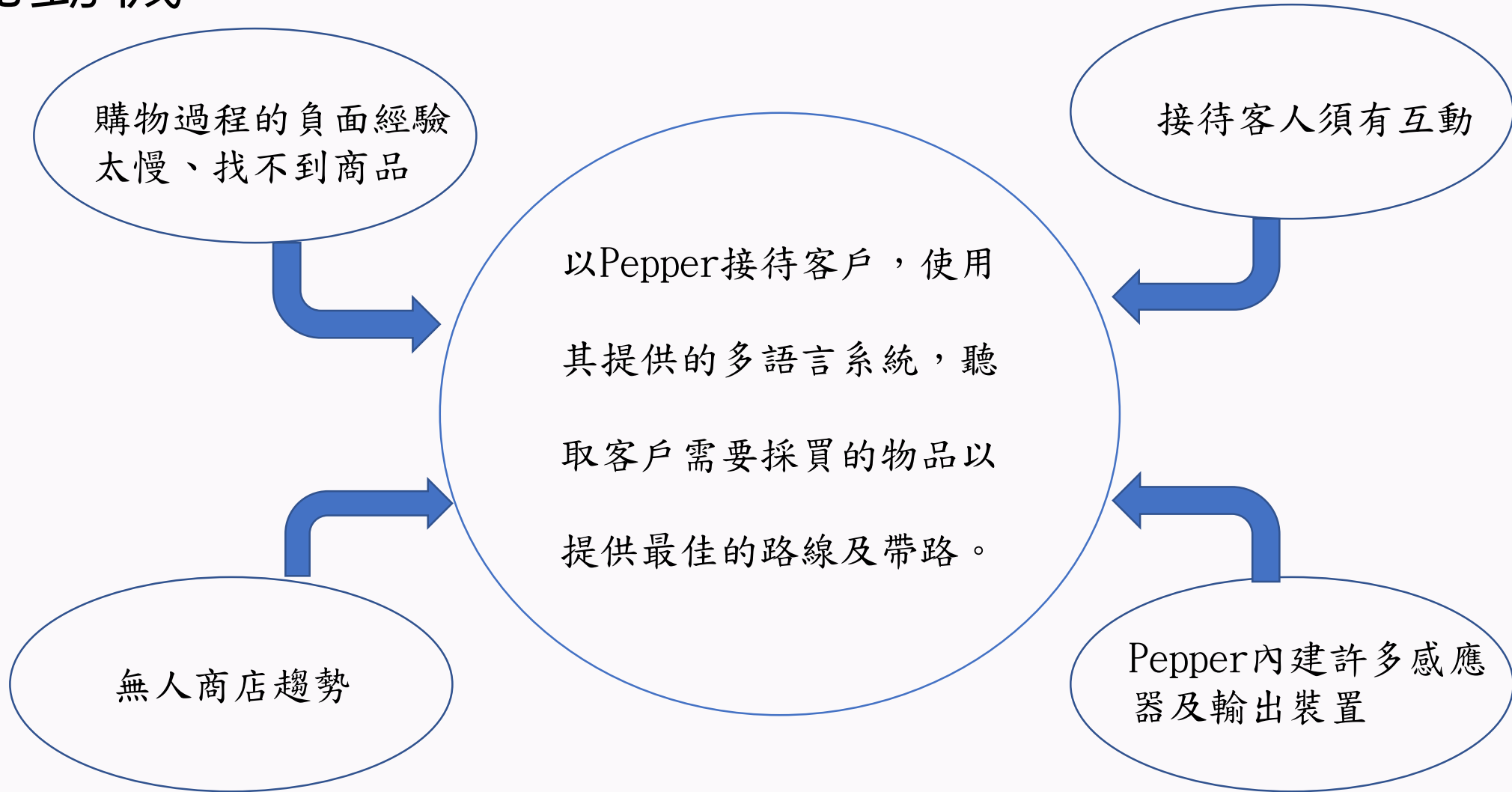
Reduce
Manpower



解決客人
帶路問題

Solve
Problem

研究動機



Part 03

研究目的

Propose

研究目的

1. 為了降低Pepper相機進行物體判斷時的誤差，須執行相機校正。
2. 利用Pepper的相機進行與目標物的偵測，取得目標的世界座標與機器座標，進而推導出座標轉換的數值。
3. 利用座標轉換取得Pepper自身的世界座標。
4. 利用路徑規劃找出採購所有商品的最佳路徑。

Part 04

文獻探討及回顧

Literature discussion and review

類似產品：「Siriusbot」機器人

優點：

- 中/英/日語音對話
- 導航功能
- 自動避障
- 以RFID電子標籤進行點貨

缺點：

- 只能設定單一目的地，無法進行多項商品的路徑規劃



XYZROBOT整合型服務機器人

優點：

- 引導指引
- 線上客服
- 自動返回充電
- 雷射及超音波感測避障

缺點：

- 溝通方式單一，僅能以螢幕點取服務內容



PEPPER機器人文獻

- 感應器分布

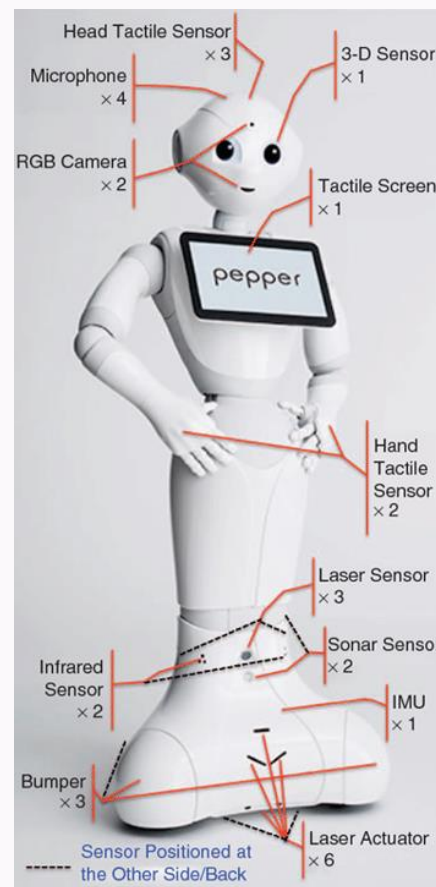
頭：麥克風×4、RGB攝影機×2、3D感應器×1、頭部感應器×3
胸：陀螺儀感應器×1、手：手部感應器×2、腳：聲納感應器×2、
雷射感應器×6、腳部感應器×3、陀螺儀感應器×1、紅外線感應器×2

- 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）



優缺點整合

經由上述三種機器人於市面上的優缺點整理

- XYZROBOT缺乏對話功能
- Siriusbot缺乏多點路徑規劃
- 而Pepper擁有對話功能與影像處理所需要的感應器，而搭配程式可以使其達到多點路徑規劃。

ARMA程式庫—

增強現實的圖像追蹤

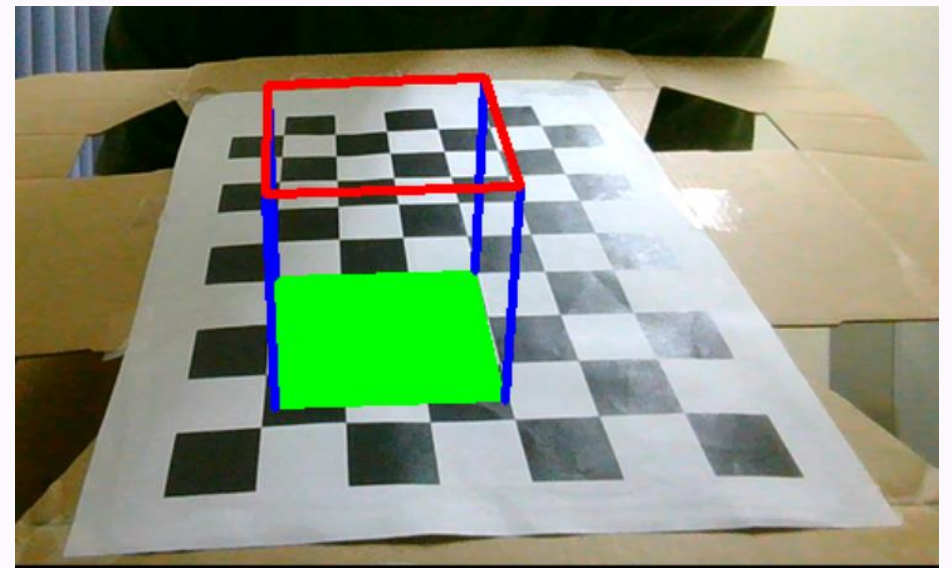
(PATTERN TRACKING FOR AUGMENTED REALITY)

增強現實系統(Augmented Reality systems)

- ✓ 在拍攝到的場景中顯示3D物件或標記
- ✓ 解決外部定向問題(相機座標與世界座標的轉換)
---因而又稱:外部校準(extrinsic calibration)或
姿勢估計(pose estimation)

基本配置:

- ✓ 內部相機矩陣:
將固有的相機內部矩陣傳入Arma程式庫, 以做影像處理。
- ✓ 加載所需要追蹤的圖片

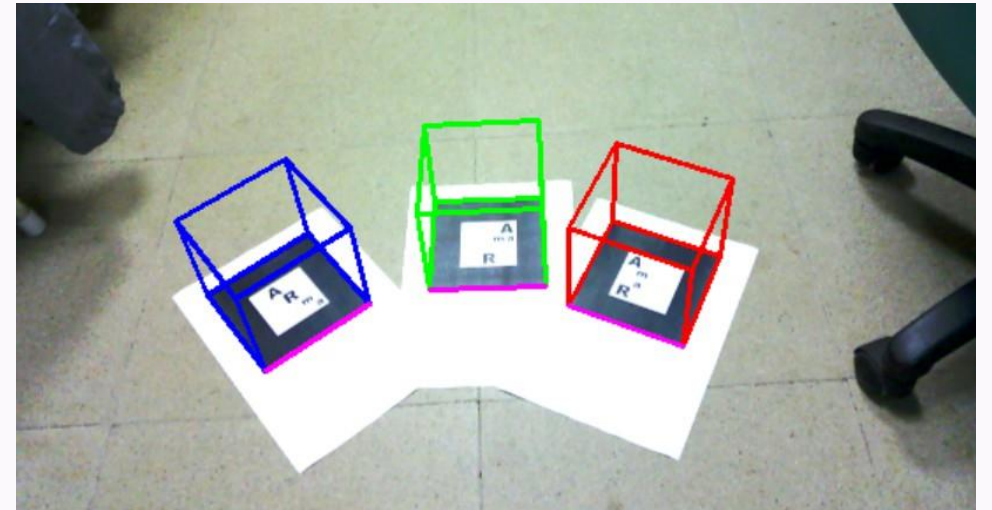
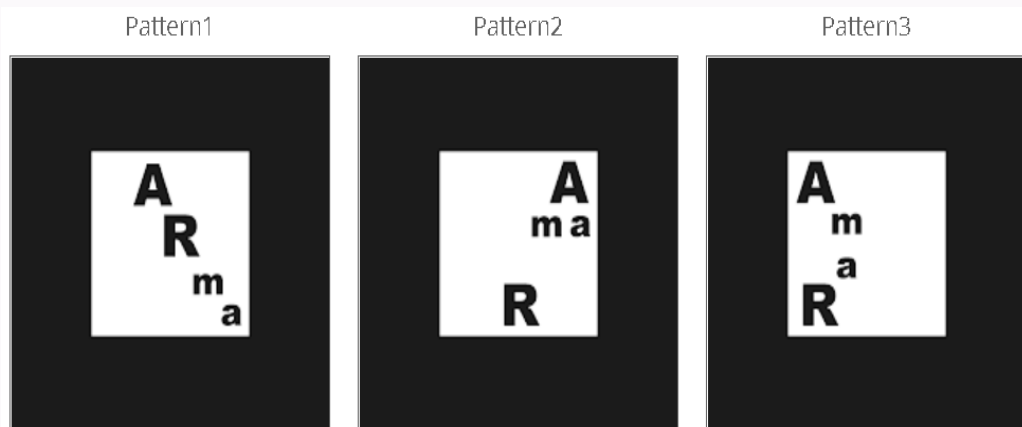


ARMA程式庫— 增強現實的圖像追蹤

(PATTERN TRACKING FOR AUGMENTED REALITY)

運作：

1. 在捕獲的影像中檢測圖案的位置 (corners)
2. 保留圖案ROI(region of interesting, 感興趣區域)，並將其和加載的圖案 進行比較。
3. 識別圖案後，測出”相機座標系”和”影像座標系”之間的轉換矩陣。
4. 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。

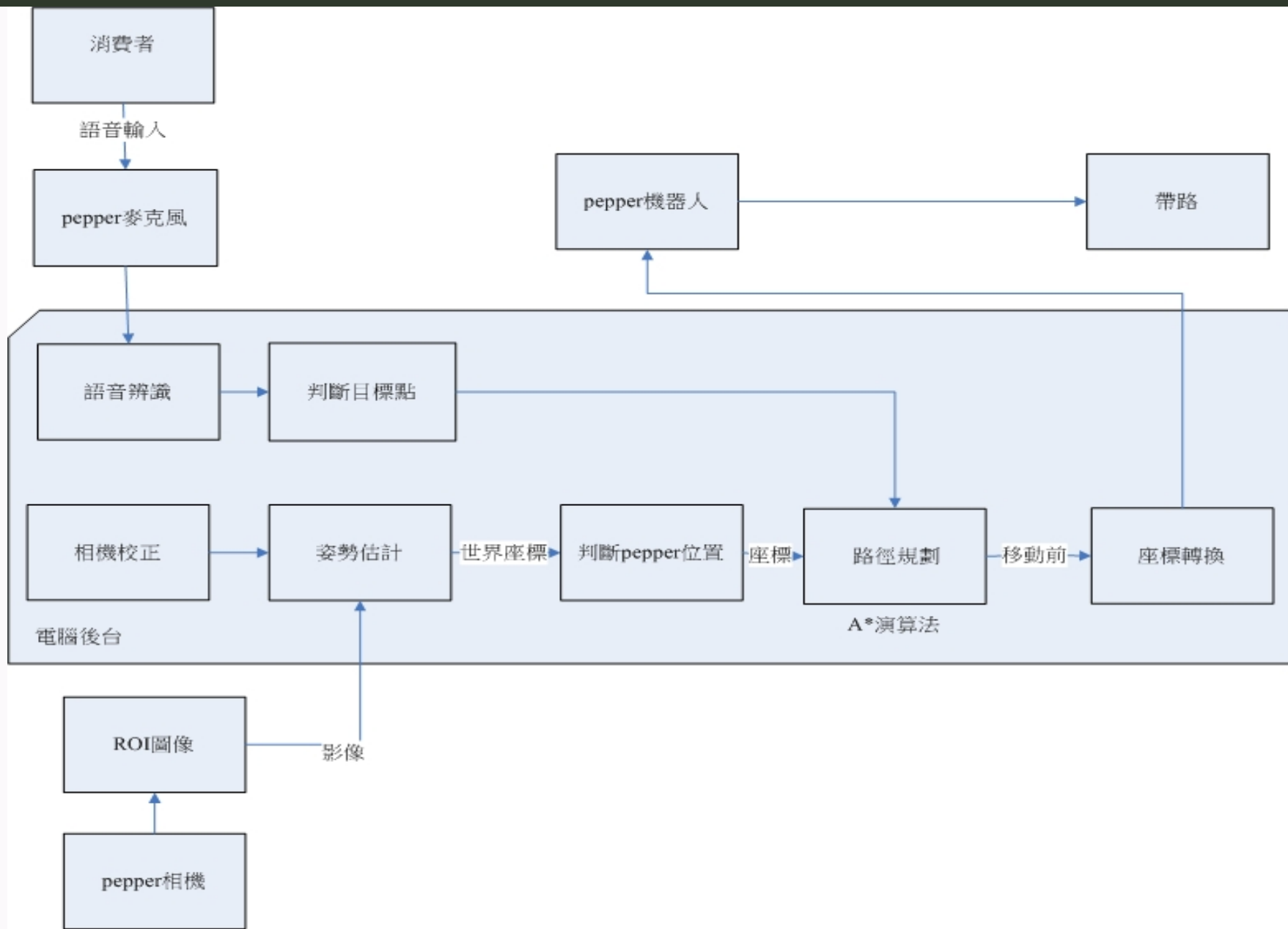


Part 05

研究方法與步驟

Research methods and procedures

研究步驟圖



研究方法—語音辨識

- 流程：當消費者與Pepper對話時，開啟語音辨識的事件，運用API判斷如何回覆與處理收到的資訊
- 使用工具：SOFTBANK提供的NAOqi中的ALDialog API
- 與其相似工具：ALSpeechRecognition
- 選擇原因：ALDialog可有效去控制機器人接收到語音後要做甚麼回覆，ALSpeechRecognition則因為會以事前放入Vocabulary的單字去辨別最接近的關鍵字，導致在辨識句子對話時容易產生錯誤，因此不適合用在此專題上。

```
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

研究方法—相機校正

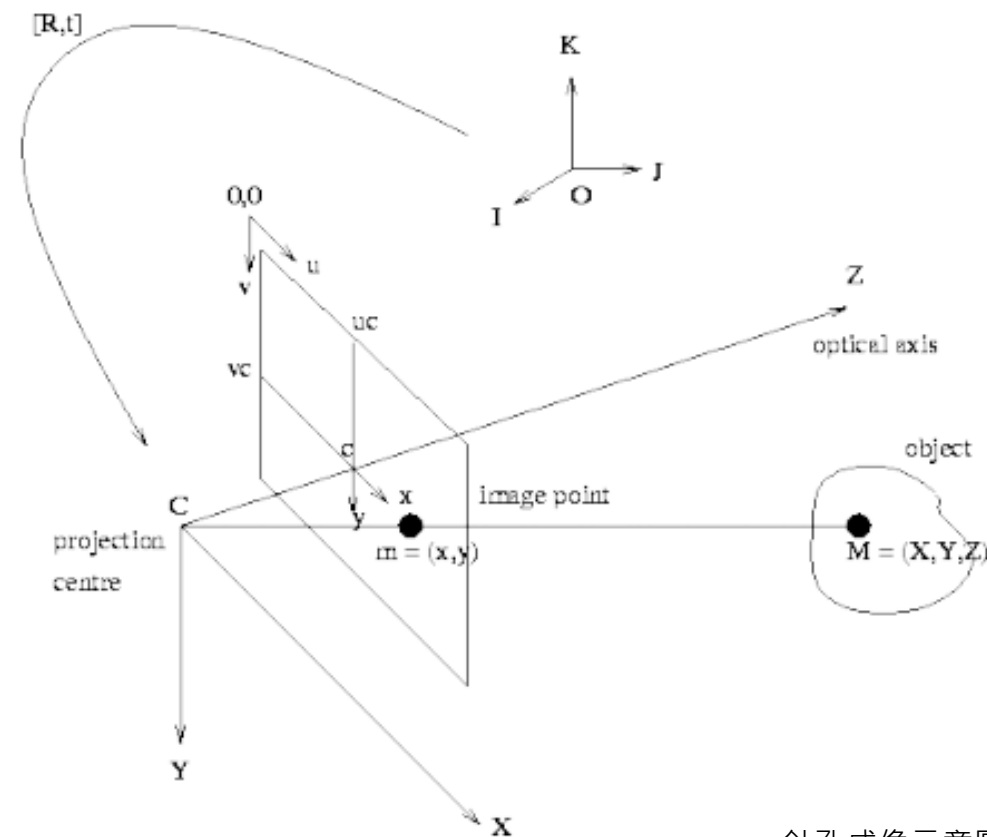
相機校正：世界座標系轉換成影像座標系的過程
功能：為了確定空間物體表面與其在圖像中某點的幾何位置與其在圖像中對應點之間的相互關係。

->建立相機成像的幾何模型，這些幾何模型參數就是相機參數。

內部參數和外部參數矩陣最基本的應用就是從 2D 影像去推得 3D 世界中的座標。

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

成像平面座標 內部參數矩陣 外部參數矩陣 世界座標



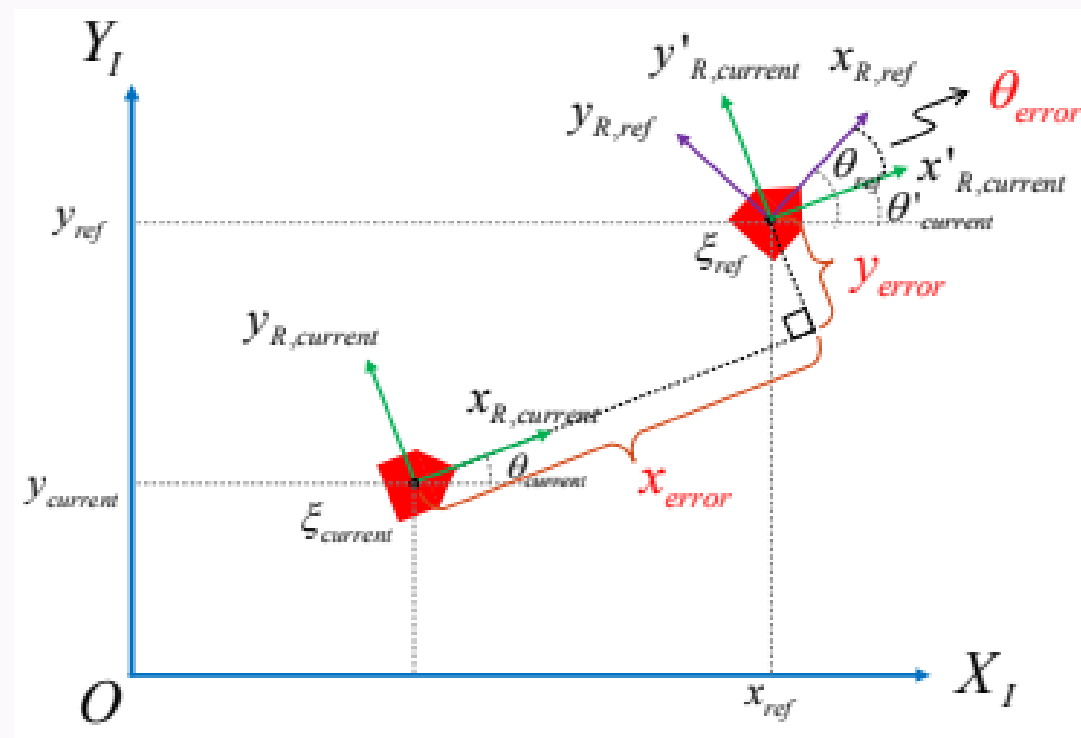
針孔成像示意圖

研究方法—座標轉換

- 世界座標：一個固定的絕對座標系，用於場地固定物品的座標。
- 機器人座標：機器人開機時以 $(0, 0)$ 開始，紀錄每次移動的距離，和機器人移動有關。

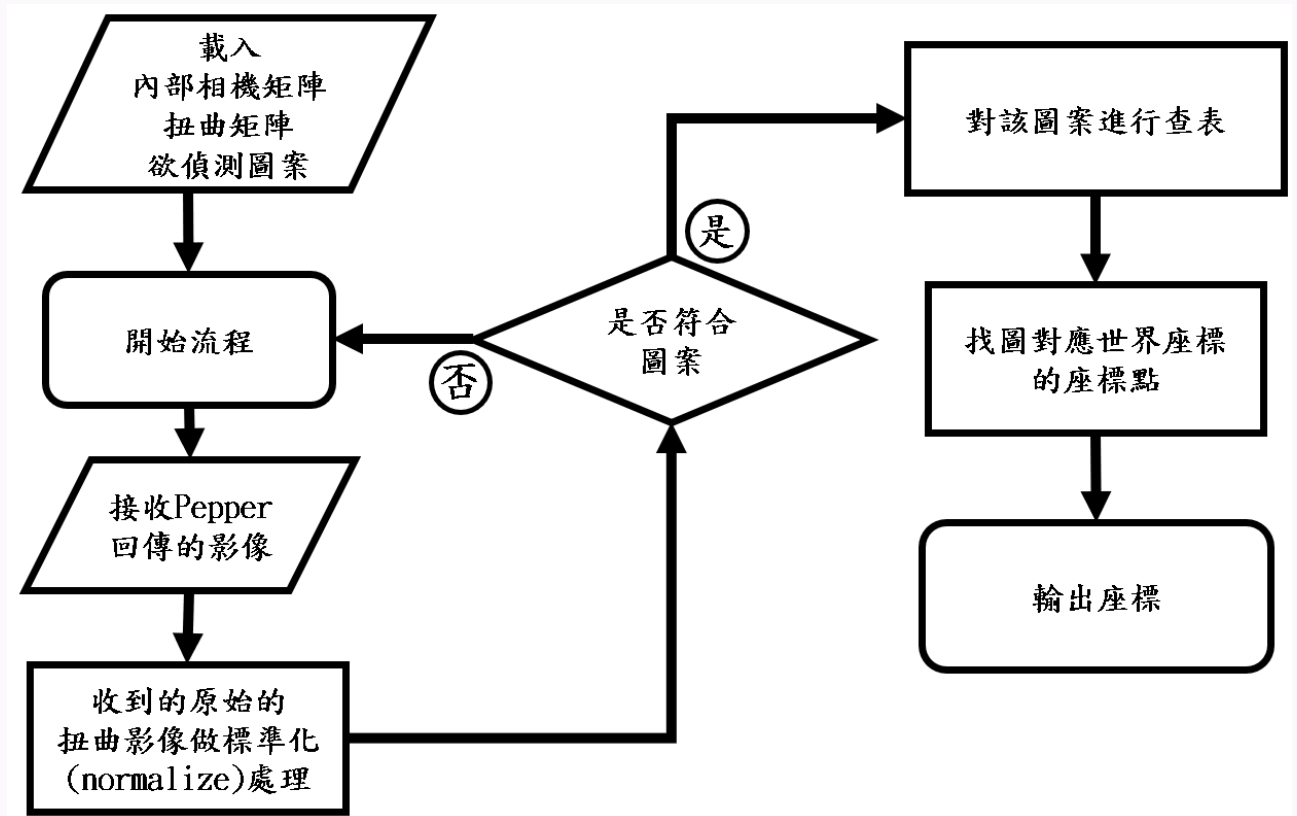
座標轉換

- 問題：每次機器人在移動時，世界座標與機器人座標的起始點不同。
- 目的：轉換世界座標及機器人座標這兩個不同的座標系，使其起始點相同。



研究方法—姿勢估計(pose estimation)

1. 電腦持續收到Pepper傳來的影像。
2. 將每次收到的原始扭曲影像做標準化(normalize)處理。
3. 找出是否有符合ROI(感興趣區域)，判斷它們為眾多圖中的哪些圖。
4. 從先前建立的世界座標字典去查表。
5. 輸出對應的世界座標，以供座標轉換做使用。



研究方法—路徑規劃

- 使用演算法：A Star (A*) 演算法
- 路徑規畫相關演算法：Dijkstra、廣度優先演算法
- 選擇原因：Dijkstra與廣度優先演算法使用在節點之間移動的代價為相等時結果相同，而A*演算法則因為使用評估函式(Heuristic Estimate)將許多較差路徑排除，進而算出一條較佳路徑。

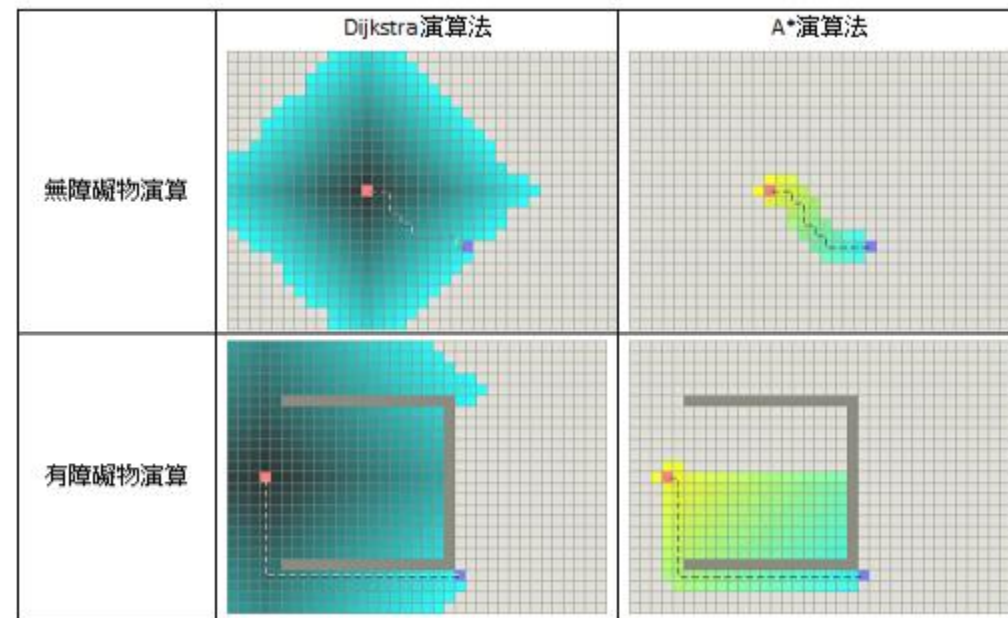
- 評估函式Heuristic Estimate：

$$f(n) = g(n) + h(n)$$

$g(n)$ ：從啟始點到目前節點的距離

$h(n)$ ：預測目前節點到結束點的距離(此為A* 演算法的主要評價公式)

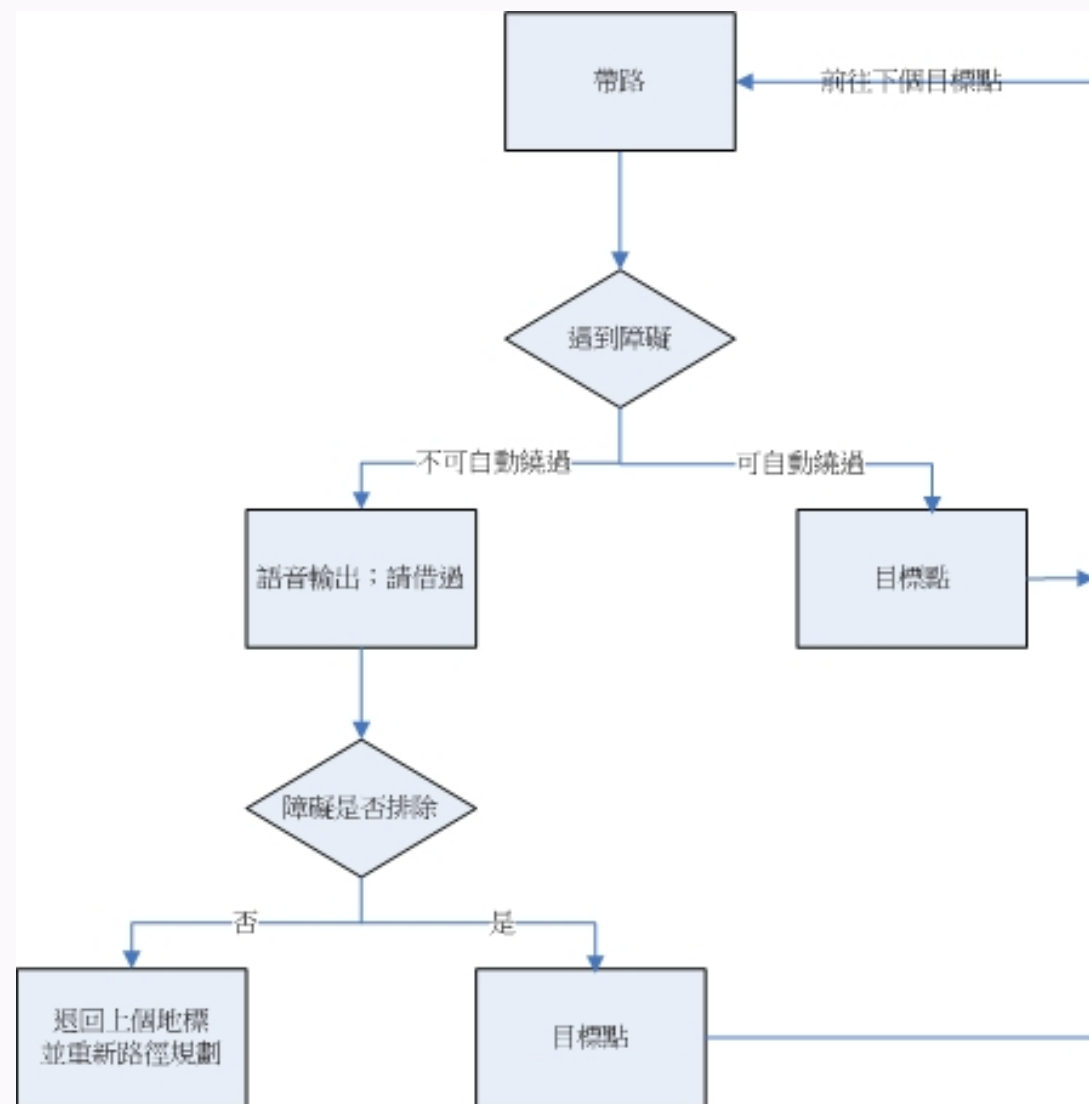
$f(n)$ ：目前節點的評價分數



- 以上為Dijkstra 與 A*演算法圖形化後運算過程中產生的比較圖

研究方法—避障功能

- 利用演算法所得的最佳路徑搭配由相機校正所得的自身座標點，來進行路徑尋訪。
- 若途中遇到障礙，先以語音輸出【請借過】
- 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。



Part 05

預期結果

Excepted Outcome

預期結果

- Pepper-

語音對話

路徑規劃

導航功能

避障功能

- 影像辨識-

相機校正

姿勢估計

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								

Part 06

展示

Demo

Demo

- 以Pepper聽取消費者所需商品並擷取目標商品關鍵字後儲存
- 以Arma程式庫實現姿勢估計偵測ROI區域以取得座標點

Demo-偵測ROI區域以取得座標點

偵測到三個圖像，區分為Patterns1、2、3
並持續回傳座標

```
Number of detected Patterns: 3
Detected pattern index: 13
X : 13.0
Y : 13.0
Detected pattern index: 12
X : 12.0
Y : 12.0
Detected pattern index: 11
X : 11.0
Y : 11.0
Number of detected Patterns: 3
Detected pattern index: 13
X : 13.0
Y : 13.0
Detected pattern index: 12
X : 12.0
Y : 12.0
Detected pattern index: 11
X : 11.0
Y : 11.0
Number of detected Patterns: 3
Detected pattern index: 13
X : 13.0
Y : 13.0
Detected pattern index: 12
X : 12.0
Y : 12.0
Detected pattern index: 11
X : 11.0
Y : 11.0
```


參考資料

「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/x04l6N>

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

<https://reurl.cc/GrLeKd>

XYZrobot整合型服務機器人：

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

Pepper機器人：

<https://reurl.cc/x04l6N>

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

<https://reurl.cc/GrLeKd>

參考資料

OpenCV座標轉換:

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

<https://opencv.org/>

相機校正:

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

影像追蹤:

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

Arma程式庫

<https://reurl.cc/R1Ev2e>

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

<https://opencv->

pythontutorial.readthedocs.io/en/latest/py_tutorials/py_c

[alib3d/py_pose/py_pose.html#goal](https://pythontutorial.readthedocs.io/en/latest/py_tutorials/py_c_alib3d/py_pose/py_pose.html#goal)

A*演算法

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