

以自走車進行物件抓取之實現

The Implementation of Object Grabbing on a Robot Car

專題成員：鍾孟岳

指導教授：楊中平



簡報大綱

- 專題動機
- 設計解決方法與系統架構
- 成果展示
- 結論與未來展望

專題動機

➤ 緣起

- 身體不方便移動及需要居家照顧的人，不易24小時都有人服侍在側，若有方便取物的居家機器人協助，生活便利性可大大增加。

➤ 期望

- 具輕巧靈活移動性，不需貴重感測器及昂貴的設備
- 能自動完成任務，不需遙控指揮
- 擴充性強，善用AI學習功能及電腦視覺
- 以資訊系所學完成系統整合(python)

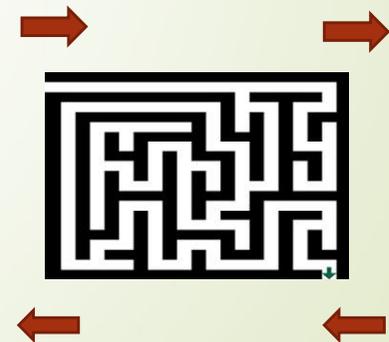
設計解決方法與系統架構

➤ 情境動作設計

- 假想 居家機器人 需要從1個地點到另1個地點取回想要的物件，除了可以避開障礙物外，並找到想要取回的物件位置及角度，順利抓取物件後，沿無障礙路徑回到原點。

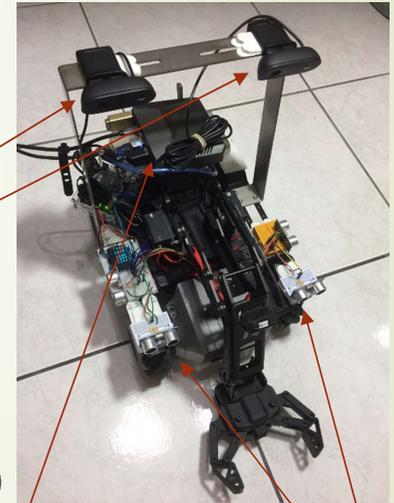
➤ 需要解決的問題(可移動及能抓取物件之載具除外)

- 如何避開障礙物
- 如何找到想要取回的物件
- 如何知道物件抓取位置及角度
- 如何順利回到原點



設計解決方法與系統架構

- 可移動及能抓取物件之載具部分 (RoboMaster EP car)
 - ✓ 具備移動、定位及夾取物件功能
 - ✓ 具備WiFi連線功能
 - ✓ 具備單一固定攝影機(未使用)
 - ✓ 具備1台紅外線感測器(未使用)
- 增加設備部分
 - ✓ 2台Logitech C270 HD攝影機(電腦視覺定位)
 - ✓ 1台Raspberry Pi4 (RP4為**計算核心**、WiFi連線及控制自走車)
 - ✓ 1台Arduino+4個超音波感測器(與RP4用USB Series連線及測距)



設計解決方法與系統架構

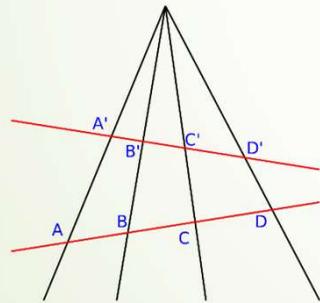
➤ 避開障礙物的方法

- ✓ 以迷宮路徑模擬為行進中障礙物
- ✓ 1台arduino+4個超音波感測器(前2左右各1)負責計算自走車與障礙物間距並傳輸結果至RP4
- ✓ 由RP4利用DFS演算法，控制自走車走出迷宮
- ✓ 移動中隨時進行位置校正，避免路面不平而偏移
- ✓ 計算及儲存可走出迷宮路徑(無障礙路徑)，作為夾取物件後的返回路徑

設計解決方法與系統架構

➤ 影像校正的方法(攝影機若調整角度需重新校正)

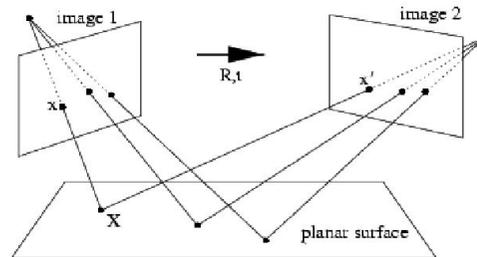
- ✓ The cross-ratio is the basic projective invariant that is preserved by the projective transformations of a projective line.
- ✓ 利用cross-ratio不變性，計算左右2張影像與實際平面影像的矩陣參數H(後續以實際偵測點對應2個攝影點解出實際偵測點的空間座標)



cross-ratio

$$\frac{\overline{AB} * \overline{CD}}{\overline{AC} * \overline{BD}} = \frac{\overline{A'B'} * \overline{C'D'}}{\overline{A'C'} * \overline{B'D'}}$$

The projective transformation between two images induced by a world plane



Definition: Projective transformation

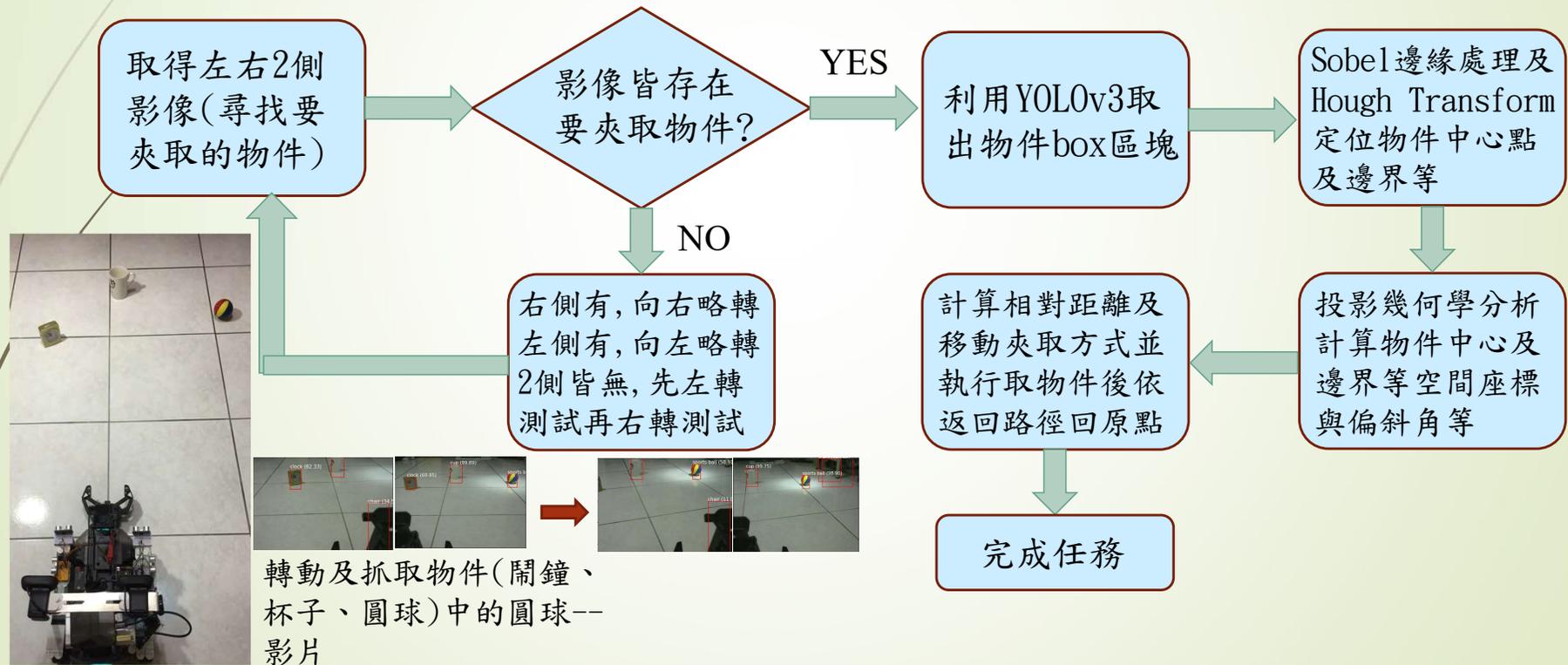
$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{or} \quad x' = \mathbf{H}x \quad 8\text{DOF}$$



從自走車上取得的左右側影像
(白色格線處為校正板)

設計解決方法與系統架構

➤ 抓取物件的方法



設計解決方法與系統架構

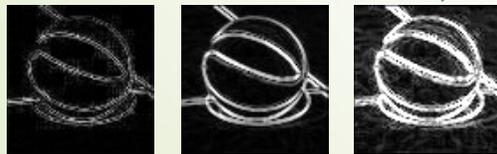
➤ Sobel邊緣處理及Hough Transform定位物件中心點及邊界 (以圓球為例)



右側有物件左側無
向右偏轉12°



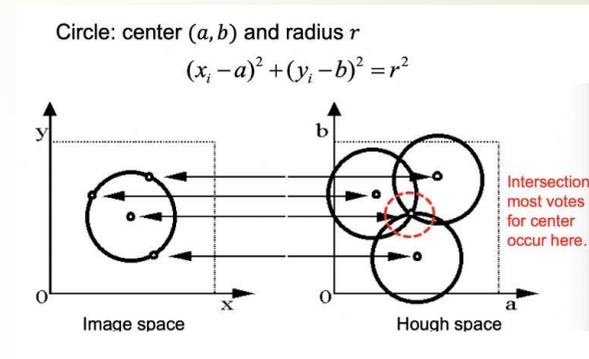
兩側皆有物件才進行
BOX位置物件影像分析



Laplacian Sobel Scharrx

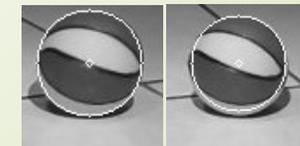
執行
YOLOV3

邊緣運算以
Sobel結果
較優



Hough Transform可偵測圖片中的直線、圓形及橢圓形，甚至能夠識別任何形狀。缺點是運算量大，運算區域不能太大。

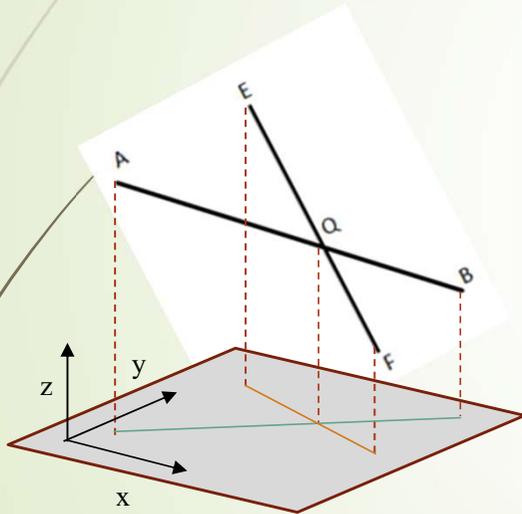
Hough Transform
偵測圓形及圓心



設計解決方法與系統架構

➤ 分析物件中心及邊界等空間座標與偏斜角(以圓球為例)

假設空間中2個投影線AB及EF交於Q點(假設為球之中心點)，
E、A點分別為左右2個camera的座標點。



利用矩陣參數H及Hough Transform偵測到的中心點
計算投影到參考面的B、F點

將A、B、E、F及Q點座標垂直投影到地面

各點的投影點於地面x, y軸的線段將形成等比關係

利用物件的2個角點形成的直線，亦可計算空間中物件邊緣直線形成的空間偏斜角

利用Q點x, y座標值及投影線關係再推算z值

利用等比關係計算出Q點的x, y座標值



成果展示

DEMO

結論與未來展望

- 目前2張640*480的影像計算物件位置大約需10幾秒(一般電腦)，若RP4計算則約需一般電腦6倍左右的時間。
- 計算物件位置的誤差在幾個mm之內，角度誤差在1度內，若光線不足會因邊界計算偏差而增加誤差。
- 改善計算時間的方式，可傳送影像由外部快速電腦協助運算完成後，再送計算結果給RP4執行抓取任務。
- 未來若核心運算速度可行，可連續計算得知物件相對位置及角度，並且同時執行抓取任務。
- 未來可結合遠端連結方式，下達取物指令後，由機器人自動取物到定點。



感謝聆聽
敬請指教