

機械手臂之自動接球系統的實現

The Realization of Automatic Ball Catching System Using Robotic Arm

廖琬洲/吳揚聲/林冠廷/黃彬華/

朝陽科技大學資訊工程系

413 台中市霧峰區吉峰東路 168 號

Tel: (04) 23323000 ext. 7633

Fax: (04)23742375

E-mail: hcliao@cvut.edu.tw

摘要

為了讓機械手臂能自己自動操作，我們的研究開發了一個項目：自動接球系統。使用 PS3、還有 eye camera 結合影像（影像辨識技術），這個系統可以偵測球的位置。然後球的位置被傳送回機械手臂，然後機械手臂得以操作去接球。

這個系統不僅可以自己判斷，還能夠自動操作機械手臂完成動作。除此之外，這個系統穩定地改善機械手臂的靈活性。

在這項研究，最大的挑戰是球飛行的時間非常的短暫（小於 100 毫秒）。只有一次機會可以傳送接球指令給機械手臂。影響辨識系統成功地整合了接球系統。

研究結果顯示這個接球系統可以達到預期的目標。

關鍵詞：工業用機械手臂、PS3 Eye 攝影機、影像辨識系統。

Abstract

In order to make the robotic arm be able to operate by itself, a ball catching system is developed in this project. PS3 (Play Station) Eye camera and pattern recognition technique is used for detecting the position of the ball. Then, the position of the ball is predicted and the robotic arm is controlled to catch the ball.

In this project, the main challenge is the flying time of the ball is very short (less than 100ms). There is only one chance to send the control command to the robotic arm for catching the ball. The pattern recognition technique is integrated into the system successfully. The result shows that the system can achieve the expected goal of this project.

Keywords: Industrial robotic arm、PS3 Eye camera、Image recognition system

1 簡介

利用影像辨識系統整合工業用機械手臂和高速攝影機這兩個硬體設備，透過攝影機不斷的擷取即時影像，記錄移動中的球的路徑座標，再根據球的前進和落下的速度，進而計算球到機械手臂時的落點座標，讓機械手臂可以在球還沒到達落點座標時，提前移動來完成接球的動作。如下圖所示。



圖 1-1 系統示意圖

2 系統概述

利用 C#來撰寫影像辨識系統，在這個系統中使用了 HSV(Hue 色相, Saturation 飽和度, Value 明度)色彩空間來過濾影像，使系統能偵測到球在影像中的位置，便可記錄球的運動軌跡並計算球的落點座標，再將座標傳送給機械手臂，以達成接球的目的。如下流程圖所示。

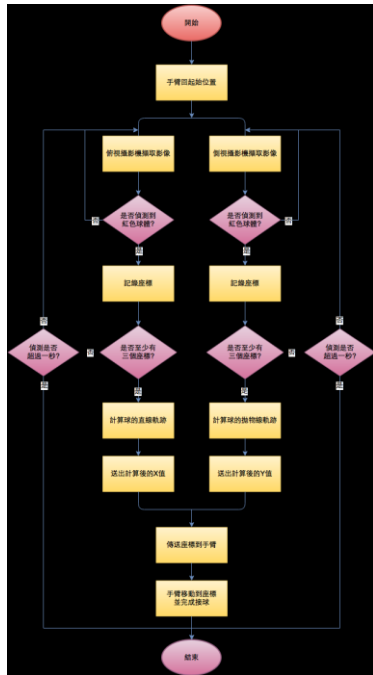


圖 2-1 系統流程圖

3 系統實作

3.1 取得影像

我們使用 `Emgu.CV.Capture` 這個類別函式來讀取攝影機影像，利用這個類別裡面的子函式 `QueryFrame` 取得攝影機影像。取得影像後，我們使用 `Emgu.CV.Image` 類別裡的子函式 `InRange`，如下圖所示。

```
public Image<Gray, byte> InRange(
    Image<TColor, TDepth> lower,
    Image<TColor, TDepth> higher)
```

圖 3-1 `InRange` 函式

給予使用者所設定的上、下界(圖 3-2)，得到了二值化過後的影像，白色部分為符合設定範圍的區塊(圖 3-3)。

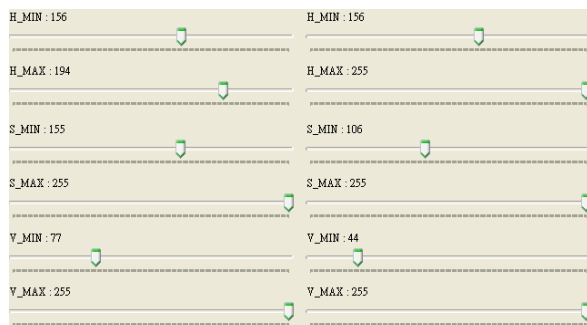


圖 3-2 利用 HSV 設定目標物的上、下界

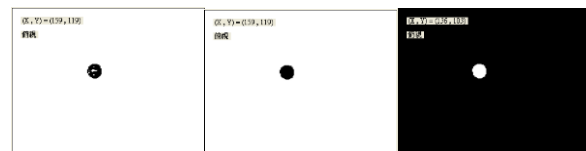


圖 3-3 過濾後二值化的圖

擷取紅球的位置

雖然這時候已經得到了二值化過後的影像，但還是有些許雜訊可能會影響我們擷取它的座標位置與面積大小，因此必須再經過以下過濾：

- (1) 利用 `AForge.Imaging.Filters.Invert` 類別將此二值化影像黑白相反，如圖(a)所示，才能將原本黑色的區域取出來。
- (2) 利用 `AForge.Imaging.Filters.ExtractBiggestBlob` 類別將最大的 Blob 減下來，取得最大的 Blob，再依據最大 Blob 來動太設定最小允許的 Blob 寬、高，再進行過濾。
- (3) 利用 `AForge.Imaging.Filters.BlobsFiltering` 類別將最大的白色留下，去掉白色區域的小黑色區塊，如圖(b)所示。
- (4) 最後再做一次黑白相反即可得到沒有雜訊的影像，如圖(c)所示。



(a) 黑白相反 (b) 留下最大的白色 (c) 再黑白相反一次

圖 3-4 (a)黑白相反(b)留下最大的白色

(c)再黑白相反一次

接著將過濾後的影像進行斑點檢測，我們使用 `AForge.Imaging.BlobCounter` 類別裡的子函式 `GetObjectsInformation`，利用此函式取得影像中白色區塊的座標與面積資訊，即可框出整顆球的位置，如下圖所示。



圖 3-5 擷取球的座標

3.2 座標計算與轉換

一、計算球的落點座標

得到座標後便可以開始計算球的落點座標，我們分別使用數學的拋物線公式以及直線公式來計算球的落點座標。

X 座標：

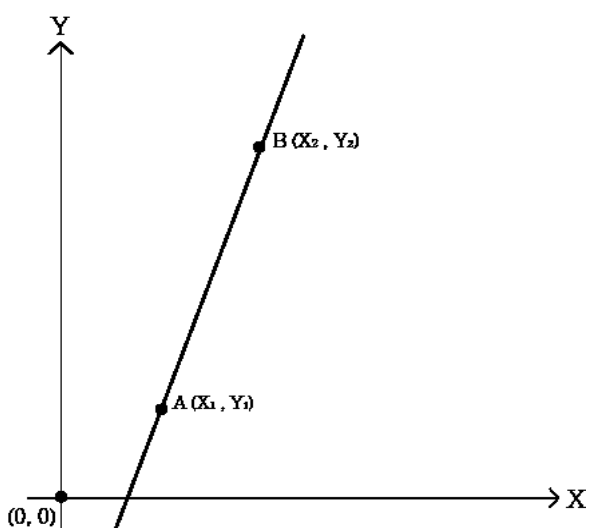


圖 3-6 俯視的直線落點座標計算示意圖

先假設一個直線方程式： $Y = aX + b$ ，再依序帶入已知的兩個座標，得到兩個方程式， $Y_1 = aX_1 + b$ 及 $Y_2 = aX_2 + b$ ，接著解聯立，得到 $a = (Y_1 - Y_2) / (X_1 - X_2)$ ，再把 a 的結果帶回原式即可得到 b ，就得到了一個直線方程式了，之後再帶入已知的 Y 座標求預測的 X 座標即可。

Y 座標：

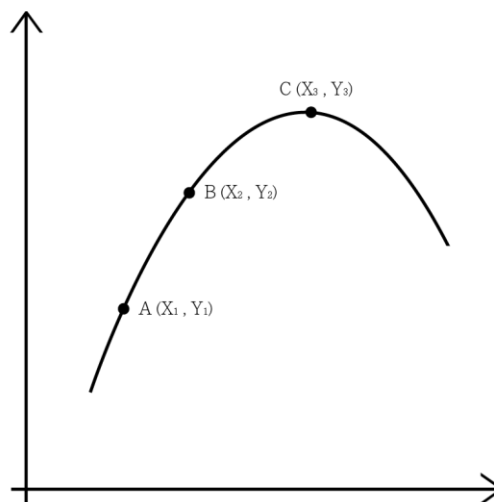


圖 3-7 側視的拋物線落點座標計算示意圖

假設一拋物線方程式： $y = ax^2 + bx + c$ ，利用平移將 A 座標當做原點 $(0, 0)$ ， B 座標 $(X_2 - X_1, Y_2 - Y_1) = (B_x, B_y)$ ， C 座標 $(X_3 - X_1, Y_3 - Y_1) = (C_x, C_y)$ 。將平移後的 A 座標帶入後得常數項 $c = 0$ ，再將 B 、 C 平移後的座標帶入，一樣得到兩個方程式後解聯立，得 $b = (B_x^2 C_y - C_x^2 B_y) / (B_x^2 C_x - B_x C_x^2)$ ，再帶回原式即可得到一個拋物線方程式，將要預測的 X 座標帶入算出 Y 座標後記得在平移回去才是所求的 Y 座標。

二、座標轉換

影像中的座標和機械手臂系統的座標是不同的，我們必須將影像中的座標轉換成機械手臂本身的座標，在這裡我們分別測量出側視攝影機的座標高度、俯視攝影機的座標寬度和機械手臂的座標高度和寬度，來計算出轉換座標的係數如(圖 3-8)和(圖 3-9)。

在機械手臂的座標中 Y 代表座標的水平軸， Z 代表座標的垂直軸。

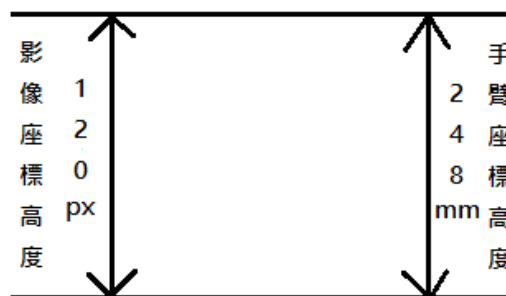


圖 3-8 側視攝影機和手臂的座標高度

手臂座標高度/影像座標高度=手臂高度座標的轉換係數由式子 $248 \div 120 = 2.0595$ 可以得知，係數 $= 2.0595$ 。

機械手臂的 Z 座標 $= y * 2.0595 + 725.86$ 。

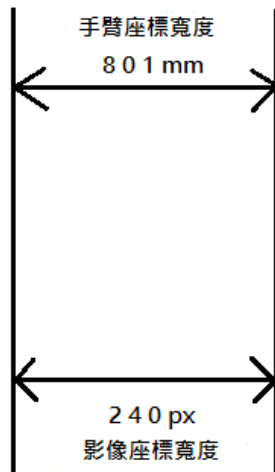


圖 3-9 俯視攝影機和手臂的座標寬度

手臂垂直座標高度/影像座標高度=手臂高度座標的轉換係數由式子 $801 \div 240 = 3.340583$ 可以得知，係數=3.340583。

機械手臂的 Y 座標 = $x * 3.340583 - 350$ 。

座標範圍：

機械手臂在動作上會有限制，運算完的座標傳給機械手臂時必須考慮到回傳的值是否超出手臂的工作範圍，如(圖 3-10)所示以中心點為初始點並測量出八個方位的工作座標，好讓程式在一開始就可以判斷是否在工作範圍。

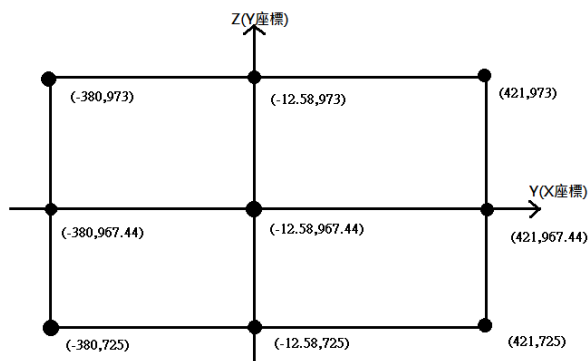


圖 3-10 手臂座標範圍

3.3 系統整合並執行

利用攝影機結合影像辨識擷取拋球時的拋體路徑，分別有俯視與側視兩種視角，如(圖 3-12)，並算出預測之落點影像座標，經由實際測量距離換算成機械手臂座標系統，以達到自動接球的目標。

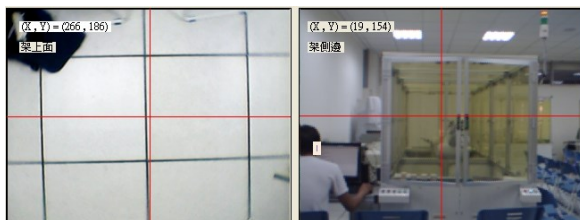
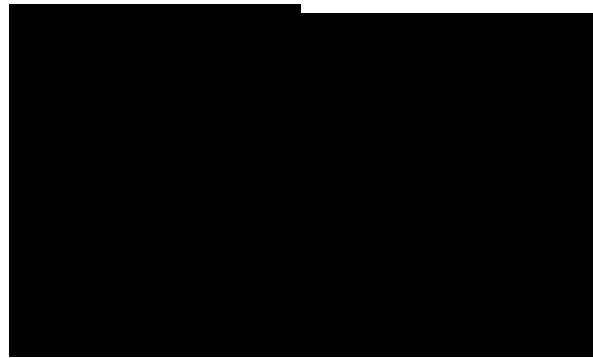


圖 3-11 俯視與側視



(a)

(b)

圖 3-12(a)現場長寬與距離(b)換算成像素

一、設定球條件

為了偵測球在影像中的位置，我們利用了前面提到的 HSV 色彩空間的過濾，個別設定 H、S、V 的上界與下界，將欲偵測物體過濾出來得到二值化後的圖。

到二值化影像之後，再用 Blob 區塊檢測方法得到白色區塊的座標與面積大小，便可以框出球的位置。如下圖所示：

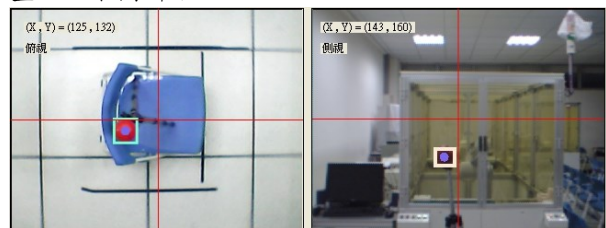


圖 3-13 利用 HSV 過濾出球的範圍

二、記錄球的座標

當兩台攝影機能夠框出球的位置，我們就開始執行程式，程式開始後，一有顏色在範圍內的物體開始截取點座標，直到物體離開影像範圍為止。如下圖所示。



圖 3-14 擷取經過軌跡座標

黑色點表示在該點時，程式開始計算落點位置並轉換成手臂座標系統，最後手臂做動到該位置來完成接球。灰色點表示預測的路徑座標。黃色點表示計算的落點座標。

三、轉換落點座標

取得落點座標後，側視攝影機的 Y 座標轉換成機械手臂的 Z 座標並傳送，側視攝影機的 X 座標轉換成機械手臂的 Y 座標並傳送，其轉換程式碼如下

圖所示。

- | |
|---------------------------------------|
| 1. Double sendY = x * 3.340583 - 350 |
| 2. Double sendZ = y * 2.0595 + 725.86 |

圖 3-15 影像座標轉換手臂系統座標並傳送

四、手臂完成接球

點選開始鍵後開始偵測球是否進入攝影範圍中，當球拋出並進入攝影範圍，程式開始記錄球經過的座標位置，直到俯視與側視個別記錄至少三個座標時開始計算落點座標並轉換後傳送給機械手臂移動到該位置，以完成接球。

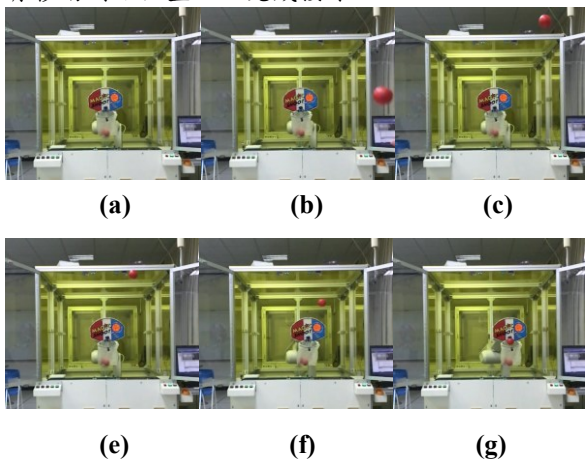


圖 3-16(a)初始狀態(b)紅色球拋出(c)球到最高點(d)球落下到手臂(e)手臂移至落點座標(f)完成接球動作

4 結論

我們成功的讓整個系統可以由攝影機自行判斷球並計算出落點座標然後讓機械手臂去完成接球的動作，使用影像識別系統整合機械手臂和攝影機兩種硬體設備，將手臂的應用更上一層樓，搭配高規格的攝影機，即使球在快速移動中，影像辨識系統也能夠更準確的抓取和記錄更多球的路徑座標。

將系統設定為預測落點座標的方式，大幅改善了機械手臂當初跟隨不到快速移動的球的問題，但是因為學校的場地受到限制，攝影機拍攝的範圍也有限，所以拋球的路徑必須在攝影機的範圍內，機械手臂才能比較精準的完成接球。

參考文獻

- [1] InRange 函式介紹
<http://www.emgu.com/wiki/files/2.3.0/document/html/19274bac-30c0-ea78-09b5-f8b429c01501.htm>
- [2] Blob 類別介紹

http://www.aforgenet.com/framework/features/blobs_processing.html

- [3] BlobCounter 類別介紹

<http://www.aforgenet.com/framework/docs/html/d7d5c028-7a23-e27d-ffd0-5df57cbd31a6.htm>

- [4] EmguCV 官方維基百科

http://www.emgu.com/wiki/index.php/Main_Page

- [5] AForge.Net 維基百科全書

<https://en.wikipedia.org/wiki/AForge.NET>