中華大學資訊工程學系

專題期末報告

A Paper Presented to the Project Implementation Course
Department of Computer Science and Information Engineering
Chung Hua University

人機互動遊戲
之機器人設計

張健毅、 孫景承

B09702088、 B09702157

指導教授：周智勳 博士

Advisor: Chih-Hsun Chou, Ph.D.

中華民國 101 年 6 月

June 2012

摘要：

在現在這個時代中，有許多高科技的電影、動畫、漫畫，人們對機器人有一種崇景，希望機器人能做到幫助人類、強化人類，甚至是就像是一個人類。這份專題中，我們希望可以做到機器人與人的互動，由最簡單的猜拳來和使用者達到交流的一種效果，當機器人獲勝時會有很開心的勝利音樂，當機器人輸掉了就會發出失敗的音樂。

【目錄】

Chapter 1 前言

樂高積木設計精美且富變化性，色彩多變。和其他積木玩具不同的是擁有廣大的成年玩家，大多為自兒童時期就受到樂高積木的吸引，樂高積木在全球 125 個國家擁有市場，推估有 3 億孩童曾經是他們的顧客。

樂高積木表面看來平平無奇，但只要稍花心思便可以創出不同東西。樂高迷樂高公司都不斷實踐這句說話的意思，對樂高貫注大量創意，令這件在過去一直被視為小孩子專利的玩具演變成藝術品。

樂高公司和麻省理工學院最早在 1988 年開始合作研發「智慧型可程式化積木」，1998 年正式在紐倫堡、倫敦和紐約玩具展中推出 Mindstorms 和 Robotics Ivention System 產品，並分別在 1998、1999、2000 年推出 Robotics Ivention System 1.0、1.5、2.0 版本，現今 Robotics Ivention System 通常被稱為 RCX。 樂高公司於 2006 正式推出 Mindstorms NXT。

我們會以一段音樂作為機器人與人之間預備出拳的一個基準，當音樂結束人對著鏡頭出拳，機器人也同步出拳，此時攝影機會連拍人出的拳，去判斷人出了什麼拳，然後再和剛剛機器人出的拳去做比對判斷輸贏，多猜個幾次機器人會將各種拳作一個統計，預測人下一次可能出的拳，以達到一個學習及進化的效果。

Chapter 2 設備需求

2.1 設備總覽

硬體	軟體
1. 電　　腦　一台 2. 有線攝影機　一台 3. 藍芽傳輸器　一個 4. Lego NXT　一台	1. Microsoft .Net framework 3.5 以上版本 2. Microsoft Visual Studio 2008 3. EmguCV 2.2.1.1150 以上版本 4. Bricx Command Center

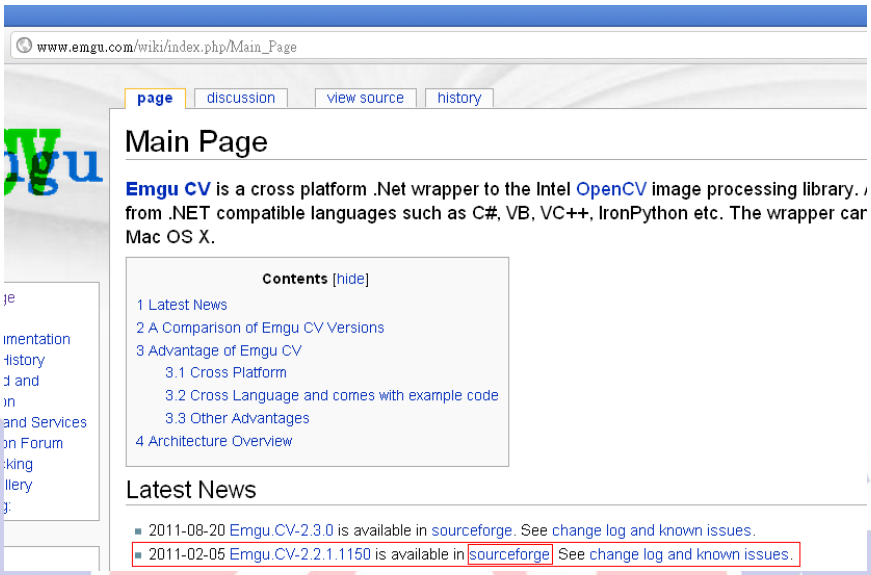
2.2 硬體設備介紹：

名稱	圖片	用途
Lego Mindstorms Nxt Brick		1. 所使用機器人的運算核心，內含一個 32bits 的 Arm7 處理器。 2. 具有 4 個輸入端與 3 個輸出。 3. 64 x 100 像素的可程式化液晶顯示器。 4. 內建藍芽。
Pro 9000 網路攝影機		1. 高畫素攝像。 2. 自動對焦。
D600 藍芽接收器		1. 傳輸距離最高可達 100 公尺 2. EDR 增強資料傳輸速率

Chapter 3 開發環境

3.1 安裝 EmguCV

3.1.1 先到 EmguCV 官網下載 EmguCV，如下圖所示：

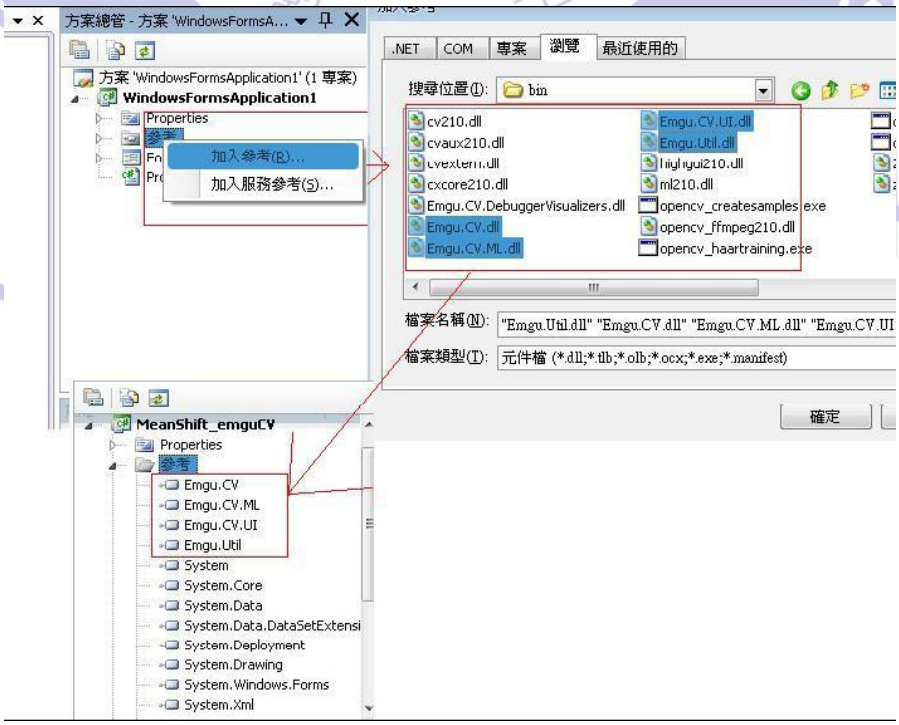


3.1.2 安裝 EmguCV 到電腦中，安裝過程都點選下一步。

3.1.3 將 EmguCV/bin 加入至環境變數的 Path。

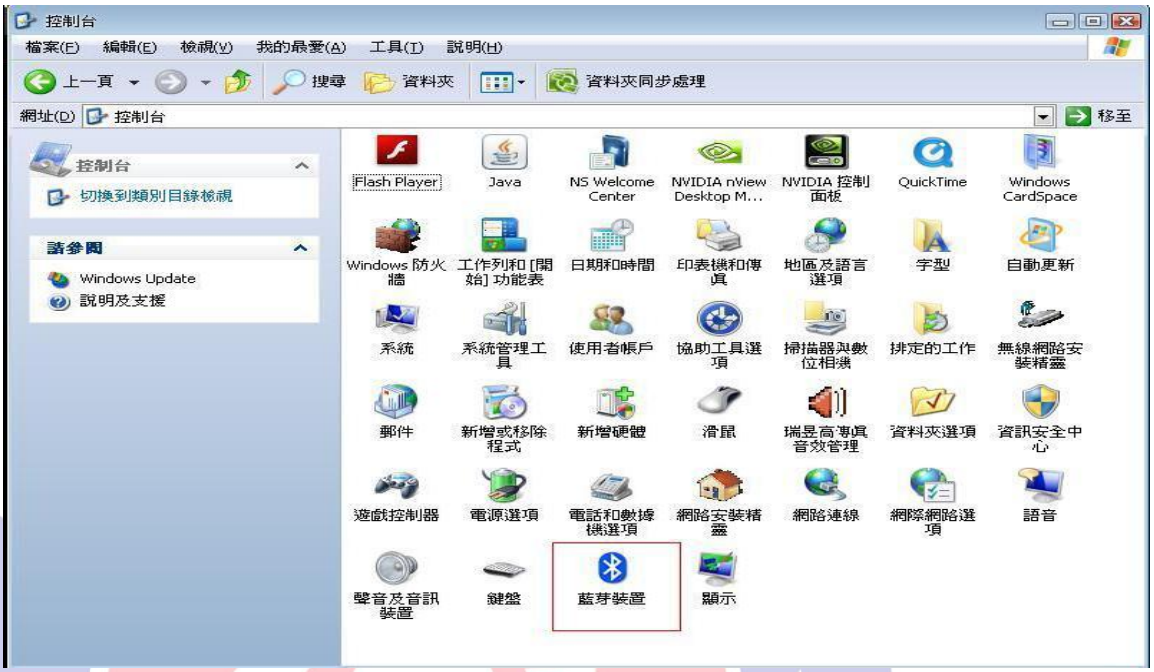
3.1.4 新增專案，並且加入相關參考。

[新增專案] -> [Visual C#] -> [Windows Form 應用程式]，之後要加入的參考如下圖所示：



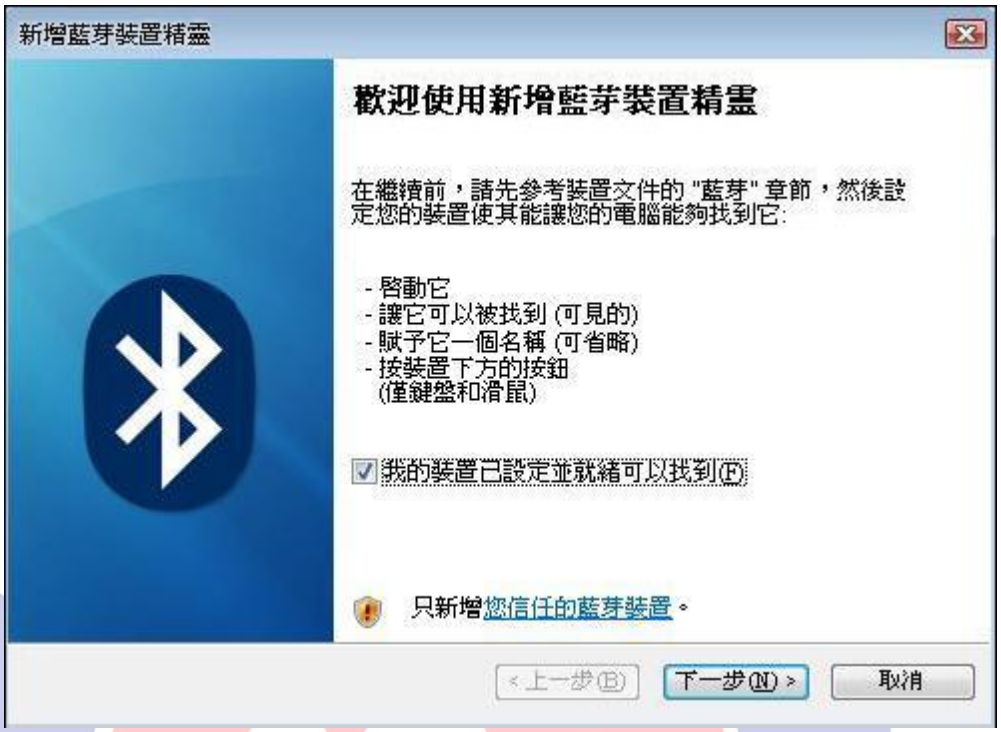
3.2 電腦端的藍芽設定

3.2.1 從［開始］->［控制台］->［藍芽裝置］



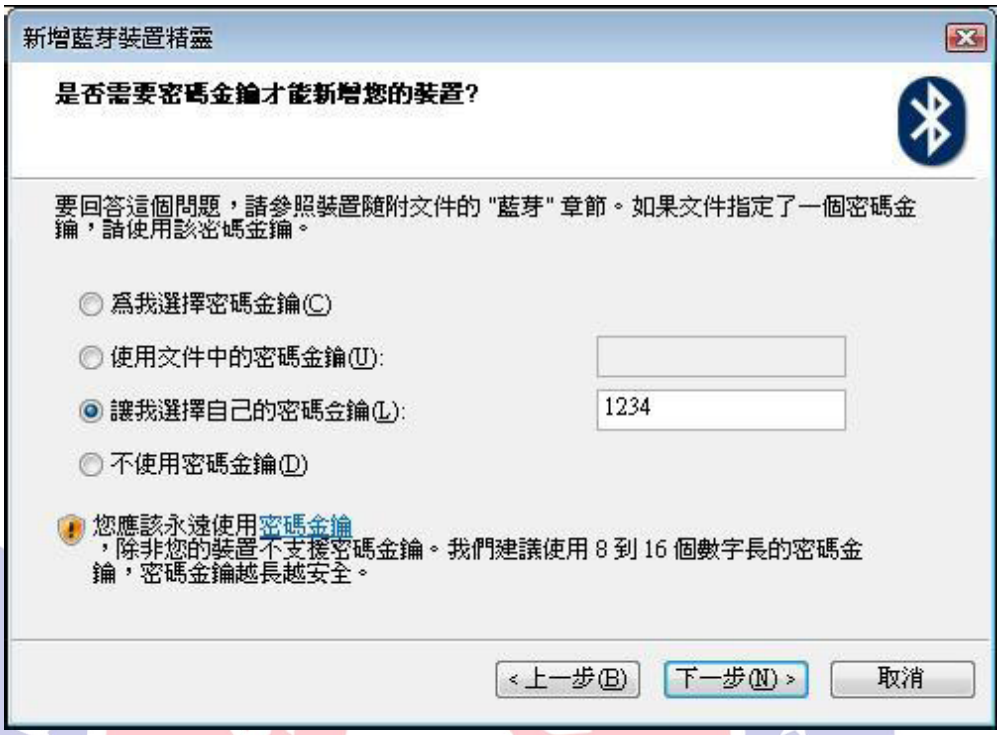
3.2.2 選［新增］，並確定NXT的藍芽已經開啟。

3.2.3 點選［我的裝置以設定並就緒可以找到］->［下一步］



3.2.4 找到NXT裝置後 ->［下一步］

3.2.5　到這個畫面，選［讓我選擇自己的密碼金鑰］，並在右方空格內輸入[1234]，這是NXT預設的密碼金鑰，之後按[下一步]。



3.2.6　到這個畫面NXT會發出嗶聲，看NXT視窗密碼是否為［ 1234 ］，是的話就按下橘色確認鈕。

3.2.7　之後會看到一個正在完成新增藍芽裝置精靈，要把［連出COM連接埠：COM7]，記下，之後電腦與NXT連接的設定中會用到。每台電腦與NXT的連接不一定都是COM7

3.3 EmguCV 連接攝影機語法簡介

3.3.1 連接前預先 using 下列程式碼

```
// EmguCV要引用的類別
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.Util;
using Emgu.CV.VideoSurveillance;
```

部分原始碼介紹(此為部分原始碼，欲見完整程式碼請參考原始檔案)：

public Capture capture; //捕捉攝影機畫面主要 class

capture.SetCaptureProperty(CAP_PROP.CV_CAP_PROP_FRAME_WIDTH, videoSize.Width) //用於設定 wecbam 畫面的寬。

第一個參數是屬性的類型，類型的內容在 Emgu.CV.CvEnum 中

第二個參數是設定屬性的值，型態為

double capture.QueryFrame();

取得攝影機目前畫面，回傳的型態為

Image<Bgr,byte>capture.FlipHorizontal

取得或設定擷取畫面可能被水忉翻轉

capture.FlipType

取得或設定翻轉的類型

Image<Bgr, byte> nextFrame; // Image<Bgr, byte> 為 emguCV 內建用於儲存圖片的 datatype,可從此 type 中直接做影像處理的動作。其中 Bgr 是顏色模型存放在 Emgu.CV.Structure 中 另外還有 Hsv、，Hls…等其他顏色模型可供使用；後面的型態 byte 也可以使用其他 double、Int32 等其他型態儲存資料。這些資料的不同將會影響到資料存放的陣列。以 Image<Bgr, byte> 為例，資料存放於.Data[,,] 型態為 byte 的三維陣列中,其中第一個 index 是指畫面 Y 的位置，第二個 index 是指畫面 X 的位置,第三個 index 是指哪一種顏色，分別為0 藍色、1 綠色以及 2 紅色。

3.4 C# 藍芽語法簡介

以下為本程式主要使用的藍芽語法，欲知更多用法、解釋及範例原始碼，可到 MSDN查詢。

SerialPort NXTconn = new SerialPort();// 藍芽變數宣告,需加入 System.IO.Ports 參考

NXTconn.IsOpen //偵測藍芽是否連線中

NXTconn.Close(); //中斷藍芽連線

NXTconn.PortName //設定藍芽的連接埠(COM ?)

NXTconn.Open(); //對目前設定的埠號進行連線

NXTconn.ReadTimeout = 1500; //設定連線逾時時間

NXTconn.Write //寫資料進去 NXT 裡面

以下為藍芽連線程式的部分原始碼，完整程式碼請參考原始檔案：

```
public string StartBlueToothConnect(string COM)
{
    //e.buttonConnect.Enabled = false;
    if (BluetoothConnection.IsOpen)
    {
        BluetoothConnection.Close();
        return "Connect";
    }
    else
    {
       // this.buttonConnect.Text = "Disconnect";
        BluetoothConnection.PortName = COM.Trim();
        BluetoothConnection.Open();
        BluetoothConnection.ReadTimeout = 1500;
        return "Disconnect";
    }
  //  this.buttonConnect.Enabled = true;
}
```

3.5 LEGO NXC 藍芽語法簡介

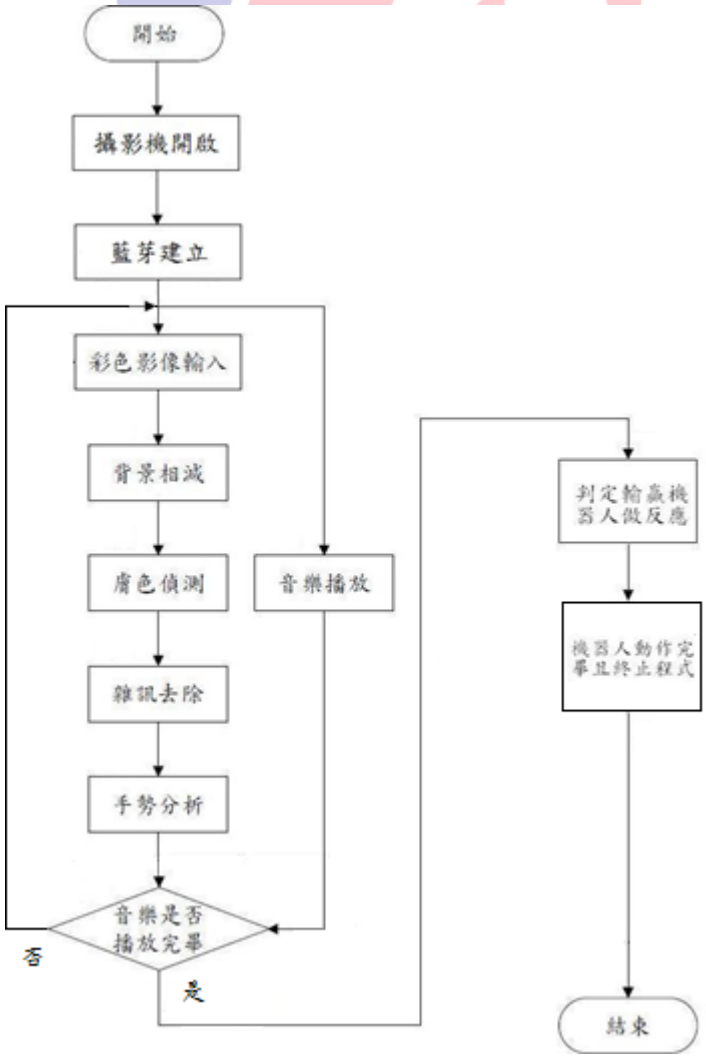
以下為本程式主要使用到的藍芽語法，欲知更多函式庫用法，可到 NXC Programmer's Guide，裡面也有完整的解釋及原始碼可看。[參考文獻]

- BluetoothStatus(connection) //NXC藍芽連線語法,若 connection帶入0 為接收端, 帶入1 為傳送端。
- ReceiveMessage(mailbox,flush,msg);// NXC藍芽接收訊息語法, mailbox:信箱編號, flushl:從信箱取出訊息後是否刪除, msg:接收到的訊息

Chapter 4 Lego 機器人控制系統

4.1 專題架構：

此專題最主要的任務為手勢辨識。我們將攝影機所擷取到的影像，經過背景相減、膚色偵測及雜訊去除等處理，然後將此處理過的影像進行手勢分析、辨識，再搭配一些機器人的動作及反應來達到臨場效果，以期能夠更讓玩家更有真正在與機器人互動的感覺。系統流程簡單說明如下：遊戲開始時，播放開場音樂。開場音樂播放時，電腦螢幕上同時隨機顯示電腦的剪刀、石頭、布，而在音樂停止時那一當下，同時辨識人的拳數及電腦隨機產生的拳數，並判定其輸贏，判定後，便播放勝負的音樂，其流程如圖：

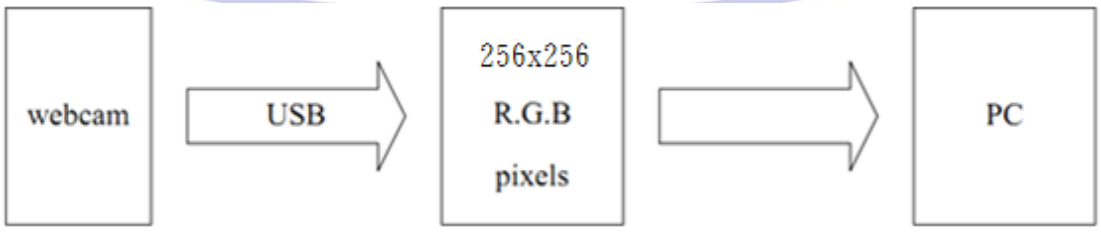


4.2 手勢辨識：

在手勢辨識中，要將手部從影像中準確的擷取出來，且要考慮系統在各個環境下的複雜背景中皆能使用，便是一個難題。因此，如何將手部從背景中分離出來，則佔有相當重要的份量，且足以決定最後分析結果的成功與否。首先，我們利用背景相減，以找出影像中手部的可能區域，經由背景相減的處理後，我們可以將手部從背景中分離出來，並且得知手部大略的區域。最後，如何在此區域中準確的找出完整的手部，也是我們所要考慮的問題。因此，我們從影像中取出的正確手部區域後，再根據手的一些特性來作為分析資料的準則，作為判定拳數的依據。在本章中將詳細說明如何利用一些影像處理之技術，以準確的找出影像中完整的手部區域，以利於後續的資料分析，再經由分析之結果，來決定手的拳數，來達到同步猜拳並搭配音效，增加其趣味性。

4.2.1 影像擷取流程

下圖為影像擷取流程圖，說明影像經由何種方式輸入及傳送影像至電腦中做運算分析。其中以網路攝影機(webcam)為系統的輸入裝置，擷取系統所需的影像資訊，透過 USB 介面傳輸至個人電腦中，其影像格式：RGB 色彩空間的 256 × 256 影像。



4.3 背景相減

為了將系統能使用在各個環境下的複雜背景，我們利用最直接且最有效的方法：背景相減。此方法可以將變動的區域圈選出來，且可以避免背景中含有與膚色相近的物品，遭到誤判為手部的可能性。

$$\mathrm{I_c(x,y)} = \begin{cases} 0, & if \quad I_{now}^{R}(x,y) - I_{back}^{R}(x,y) < Th_R \quad and \\ & \quad\; I_{now}^{G}(x,y) - I_{back}^{G}(x,y) < Th_G \quad and \\ & \quad\; I_{now}^{B}(x,y) - I_{back}^{B}(x,y) < Th_B \\ 1, & otherwise \end{cases}$$

因此，我們以攝影機開啟時，所抓取的第一張影像作為背景，而後將擷取到的影像與背景做相減的動作(RGB 色彩模型)，以求目前影像中與背景不同的區域。如上公式，其中Th_R、Th_G、Th_B分別為 R、G、B 的門檻值。

4.4 膚色偵測

為了找出手部的真實區域，我們便針對由背景相減後的影像變化區域做膚色偵測，採用 YCbCr 色彩模型。這是因為 RGB 色彩影像非常容易受到光線變化的影響，且環境光線變化對膚色的判定影響甚大。因此，我們將色系由 RGB 轉換到 YCbCr 色彩空間，以減少顏色對亮度的依賴，其中 Y(luminance)為亮度元素，Cb (blueness)、Cr (redness)分別為兩個彩度元素，由於對亮度的分離性高，方便與彩度分開操作，很適合於偵測膚色上使用。RGB 色彩空間轉 YCbCr 色彩空間的轉換關係如下公式：

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

4.5 雜訊去除

4.5.1 侵蝕、膨脹法則

在膚色偵測後，仍會有類似膚色的小雜訊。再者，由於我們是先經過背景相減的步驟，再對影像做膚色偵測後的二值化影像，此時的影像較為破碎，為了避免影響後續演算法的處理。因此，我們採用形態學(Morphology)中的閉合(Closing)運算。閉合運算包含了侵蝕(Erosion)與膨脹(Dilation)。先做膨脹運算來修補主體中的細小缺口，使得影像輪廓平滑，再做侵蝕運算來消除一些細小的雜訊。經過這樣的程序後，便可以將影像中的雜訊點移除。在此的侵蝕與膨脹，我們皆使用 3 × 3 大小的遮罩(Mask)，依序由左而右，由上而下的掃描整張影像，如下圖所示。

C_1	C_2	C_3
C_4	C_5	C_6
C_7	C_8	C_9

膨脹：首先判斷遮罩在影像中所在位置像素 C_2、C_4、C_5、C_6、C_8，在二值影像中是否有任一值為 1(5 點中只需其中 1 點為 1)。有，則 C 為 1，反之則為 0，亦即下公式，最後再將 C 代入 C_5。

$C = C_2 \cup C_4 \cup C_5 \cup C_6 \cup C_8$　　∪：代表 OR 運算　　(3

侵蝕：首先判斷遮罩在影像中所在位置像素 C_2、C_4、C_5、C_6、C_8，在二值影像中是否所有值皆為 1(5 點中必須皆為 1)。有，則 C 為 1，反之則為 0，亦即下公式，最後再將 C 代入 C_5。

$C = C_2 \cap C_4 \cap C_5 \cap C_6 \cap C_8$　　∩：代表 AND 運算

為尚未經過閉合運算的二值影像；為經過閉合運算後的二值影像。我們所獲得的膚色範圍二值影像，經閉合運算處理後，將可消除較小之雜訊點，且使得主體影像輪廓較為平滑。

4.6 抓取特徵

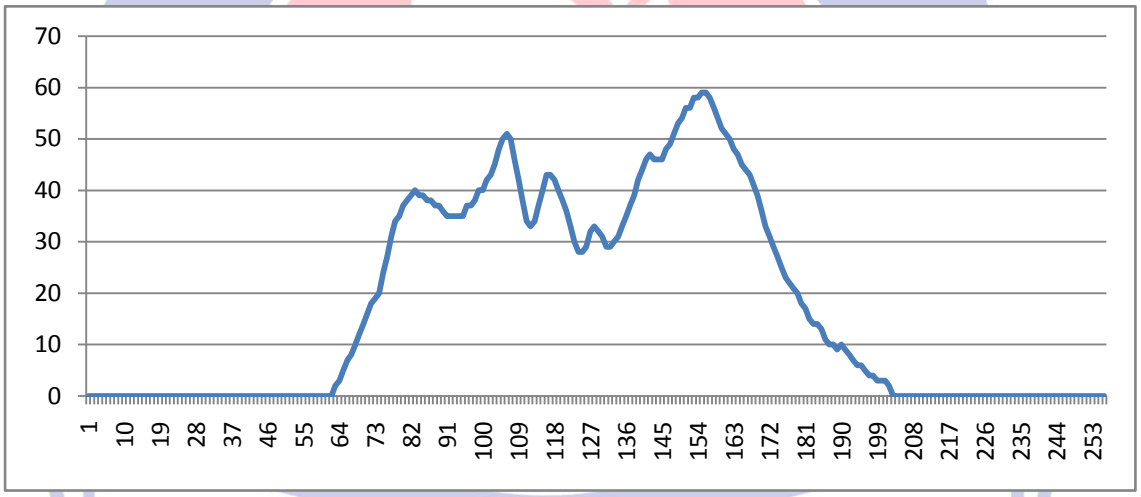
Histogram

在影像處理或電腦視覺中，Histogram(直方圖)常用來統計分佈。在此專題中，我們設定了數個 Histogram 容器，經過 Normalization(正規化的動作)，找出各個手式的分布區域。

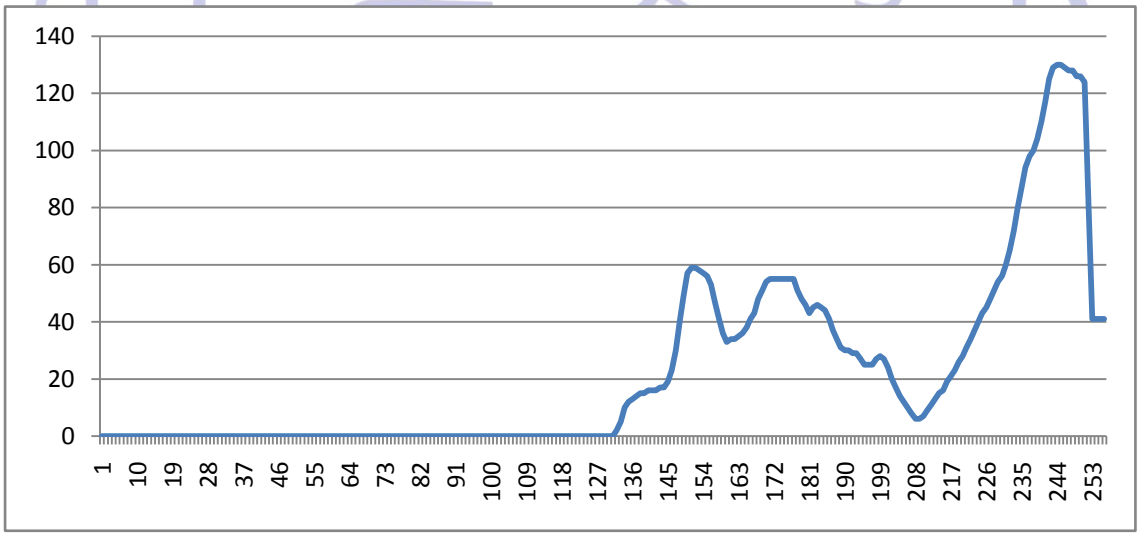
剪刀

正規化前

橫軸分布圖

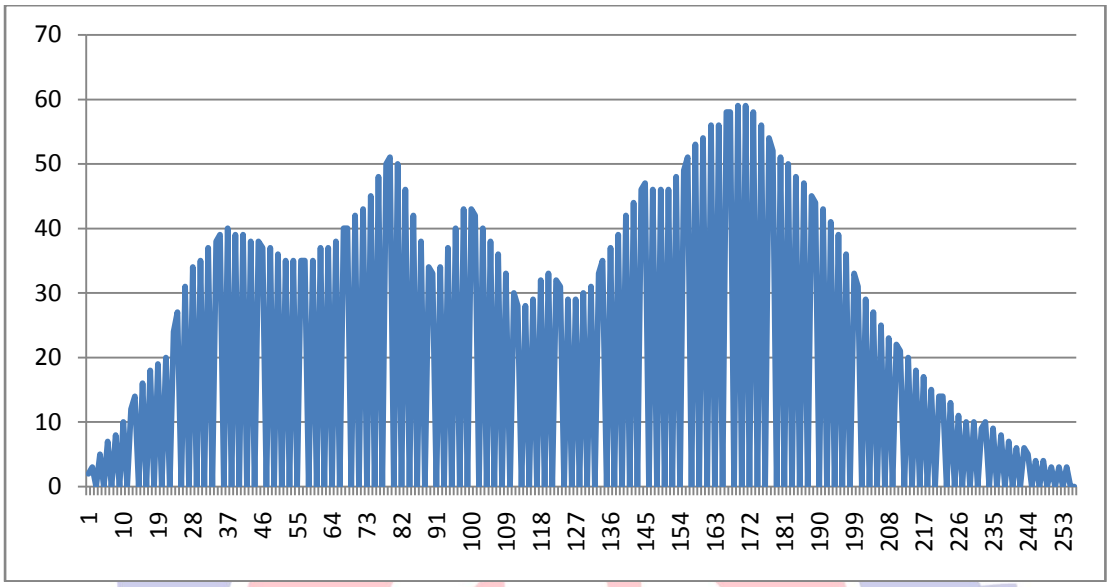


縱軸分布圖

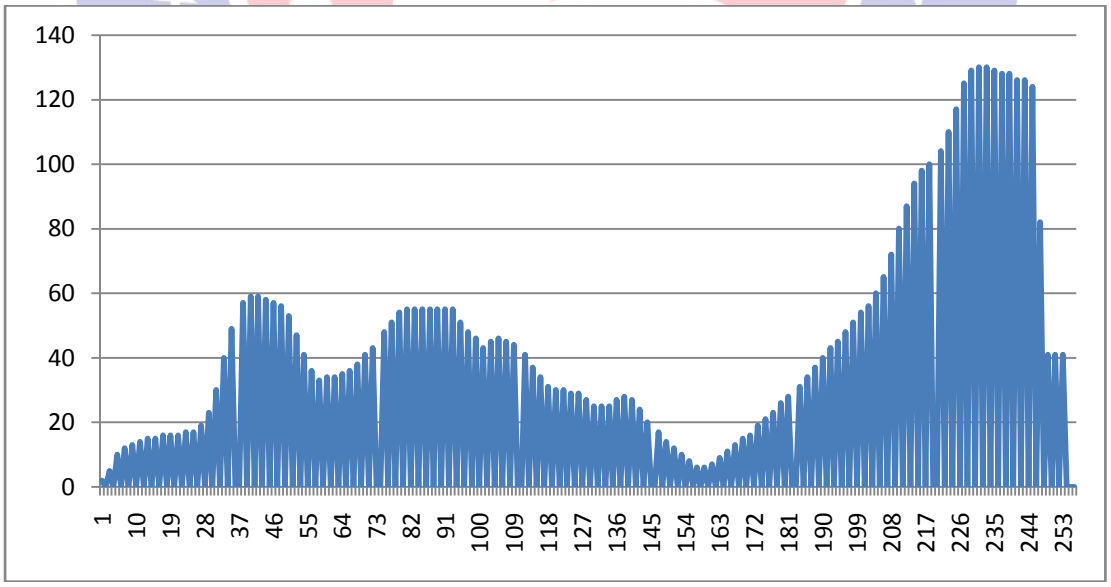


正規化後

橫軸分布圖



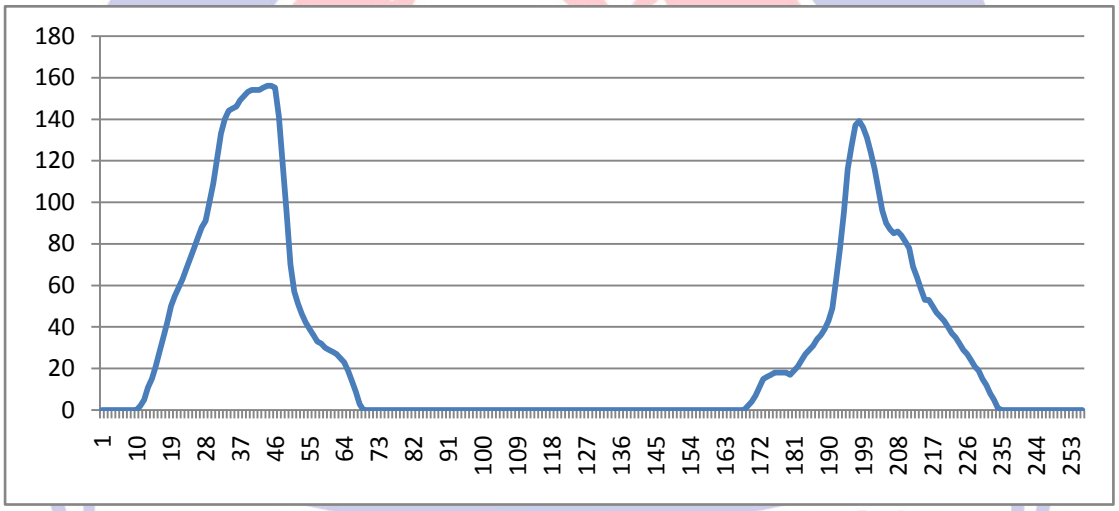
縱軸分布圖



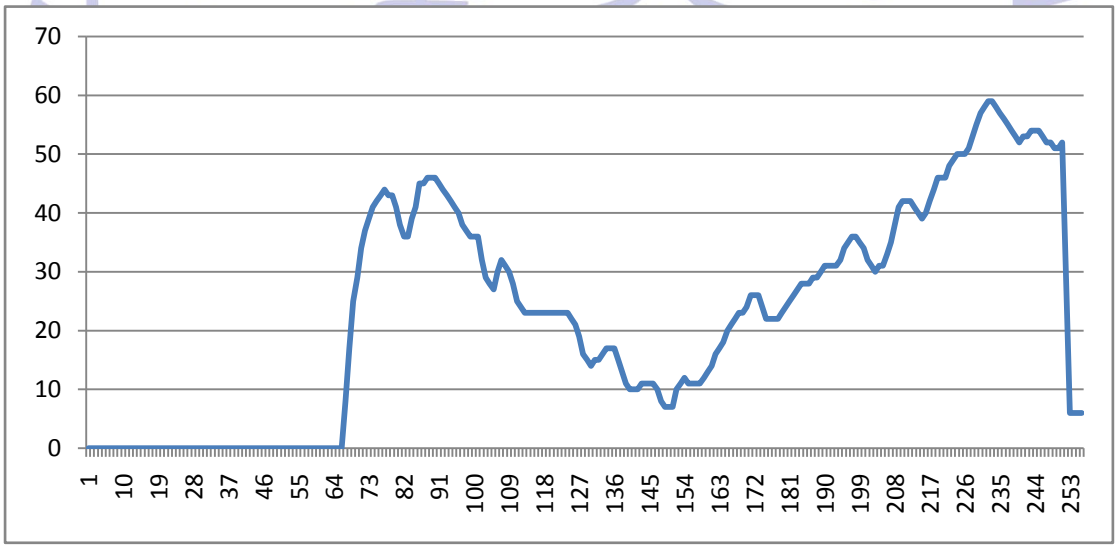
石頭

正規化前

橫軸分布圖

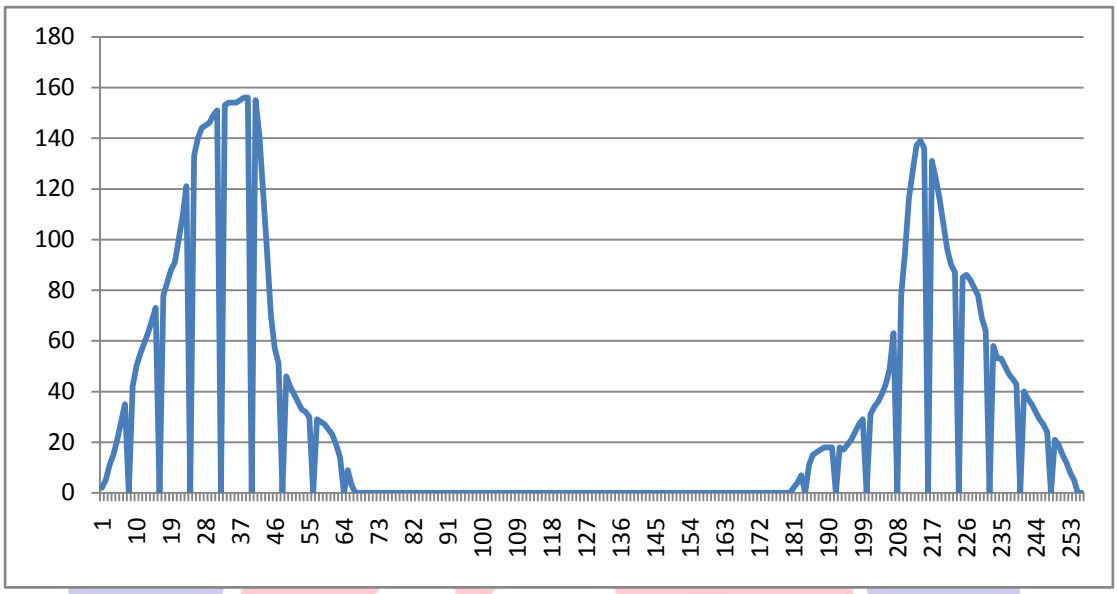


縱軸分布圖

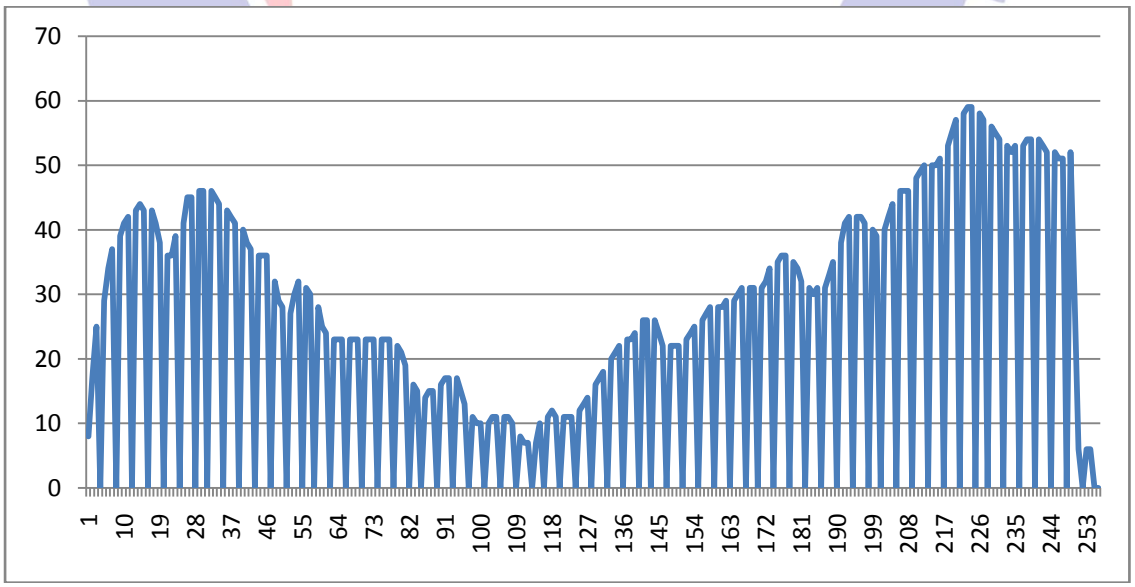


正規化後

橫軸分布圖



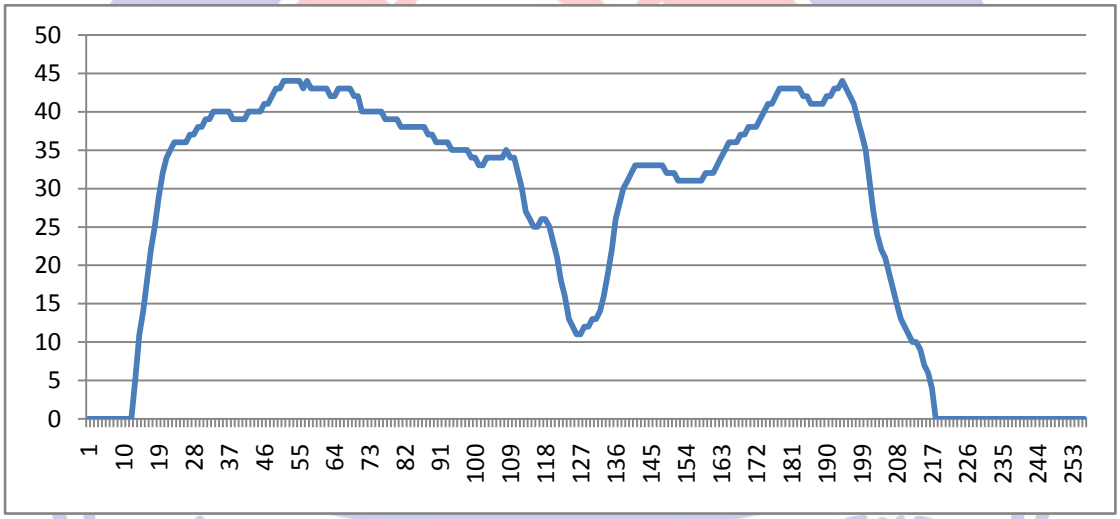
縱軸分布圖



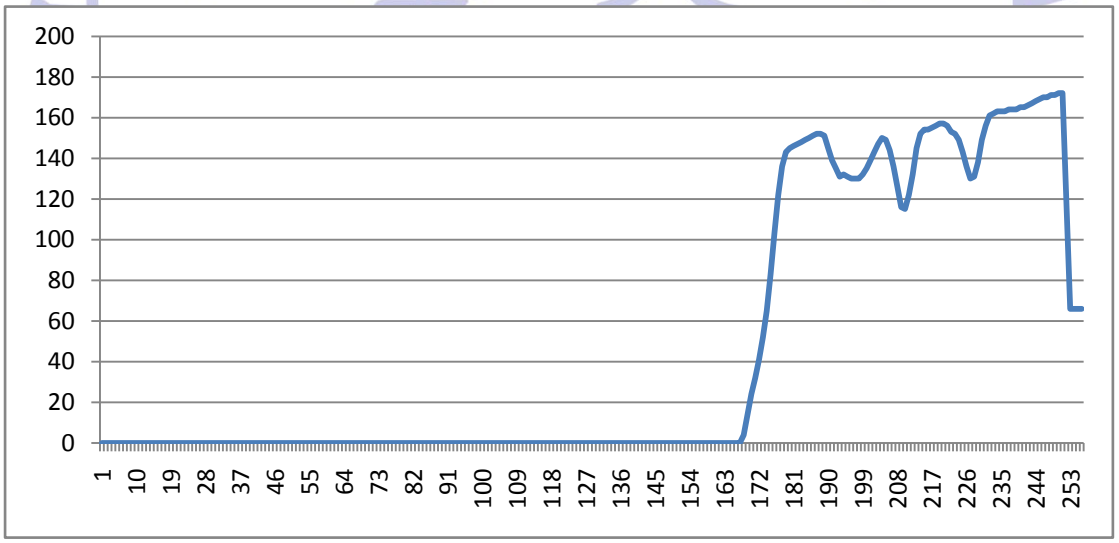
布

正規化前

橫軸分布圖

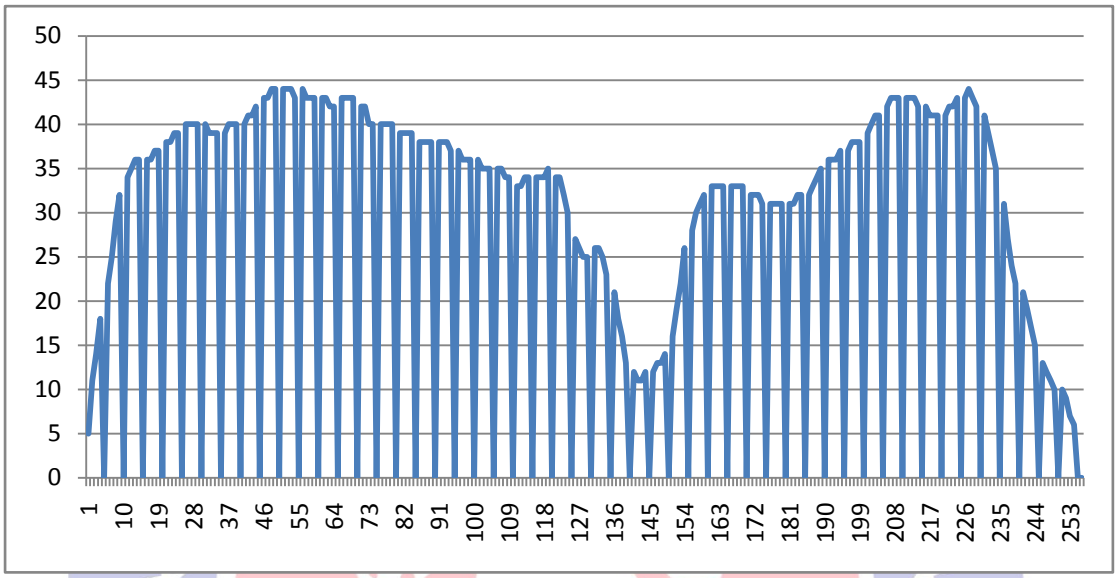


縱軸分布圖

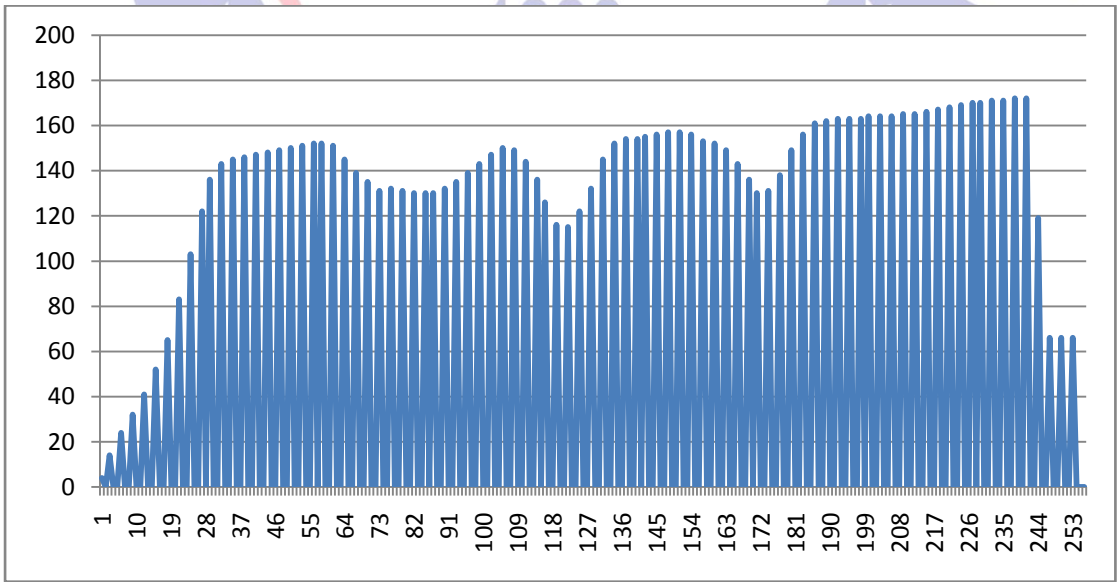


正規化後

橫軸分布圖



縱軸分布圖



Chapter 5 辨識率

	拳種	成功	失敗	剪刀	石頭	布	準確率(%)
測試者 A	剪刀(50 次)	34	16		2	14	68
	石頭(50 次)	49	1	1		0	98
	布 (50 次)	38	12	12	0		76
測試者 B	剪刀(50 次)	39	11		1	10	78
	石頭(50 次)	28	2	0		2	93.33333
	布 (50 次)	38	12	10	2		76
測試者 C	剪刀(30 次)	21	9		0	9	70
	石頭(30 次)	29	1	1		0	96.66667
	布 (30 次)	23	7	6	1		76.66667
測試者 D	剪刀(30 次)	23	7		0	7	76.66667
	石頭(30 次)	28	2	0		2	93.33333
	布 (30 次)	25	5	5	0		83.33333
總數	剪刀(160 次)	117	43		3	40	73.125
	石頭(160 次)	134	6	2		4	95.71429
	布 (160 次)	124	36	33	3		77.5

Chapter 6 評估與展望

我們的辨識率目前大概是七成左右，當然最希望能提升到 80% ~ 90%，而希望日後能再增加的功能是，統計玩家的出拳習慣而來預測玩家下一個出什麼拳，還有由機器人方來來啟動，讓整體看起來更像是與機器人互動。

Chapter 7 結語

這次的專題，當最後看到機器人動起來能與人互動時，既開心又覺得很有趣，這次的專題大部分都是在處理影像的部分，辨識的部分算是花了集大多部分的時間，而沒能從機器人方啟動算是美中不足的地方。

參考文獻:

[1] 機器人新視界 NXC 與 NXT

[2] [C#] OpenCV 初體驗, .NET 菜鳥自救會

http://www.dotblogs.com.tw/chou/archive/2009/06/13/8812.aspx

[3] Mindsqualls C# Code Examples

http://www.mindsqualls.net/Examples_1_2_1.aspx

[4] 台灣博碩士論文之是加值系統

http://ndltd.ncl.edu.tw/

[5] 微軟MSDN

http://msdn.microsoft.com/zh-tw/

[6] NXC Programmer's Guide

http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/main.html

附錄

LEGO 語法簡介

- RotateMotorEx() 精密控制
  RotateMotorEx('ports', 'speed', 'degrees', 'turnpct', 'sync', 'stop')
  ports : 分別是 OUT_A, OUT_B, OUT_C，要同時輸出二個可使用 OUT_AB, OUT_BC, OUT_AC，全部輸出用 OUT_ABC 動作函數。
  degrees :單位是角度 0~360。
  turnpct: 單位為 -100~100(%) 轉彎百分比的值代表同步的比率，例如 50 表示一輪轉一圈時，另一輪轉半圈，為負值時代表逆轉，以此類推。
  sync: true/false : 是否要同步。
  stop: true/false : 是否在輸出後剎車。

- TextOut(X,LCD_LINE2,String);
  X : 代表那一行的X坐標軸。
  LCD_LINEn : n 為參數，顯示是第 n 行。
  String : 要顯示螢幕的字串。

- PlayTone(frequency, duration)
  frequency：頻率的單位是 Hz(請參閱頻譜表)。
  duration：單位的值是 1/1000 s。

- bricxCC 的工作(task)和副程式(subroutines)，brick 可以使用 tasks 和 subroutines
  寫出多線程的程式(多工處理)。
  ```
  tank name()
  {
  }
  tank name2()
  {
  ```

}

- Acquire() 和 Release()
  Acquire(mutex name) ==> 等於是 Lock()
  Release(mutex name) ==> 等於是 UNLock()

- LEGO BricxCC Sensor 讀值和使用

LEGO 的 Brick 有四個輸入，分別叫做 IN_1, IN_2, IN_3, IN_4，可接幾種 Sensor，使
用前先用 SetSensor 來指定種類，而取得值則使用 SENSOR_1, SENSOR_2, SENSOR_3,
SENSOR_4（或是 S1, S2, S3, S4），接下來分別看各種 Sensor 怎麼使用。

1. 碰撞感測 Touch Sensor

設定 IN_1 是 touch sensor
SetSensor(IN_1,SENSOR_TOUCH);
或是
SetSensorTouch(IN_1);
讀值 SENSOR_1 例如：
if (SENSOR_1 == 1) { ... } // 有壓按(或碰觸)

2. 光線感測 Light Sensor -- 多用來沿線尋跡用

設定 IN_3 是光感
SetSensorLight(IN_3);

讀值 int val= Sensor(IN_3);
val 的值介於 0~100，暗的值低，亮的值高

預設是開燈檢測，如果要查週圍的光線值，可以設定關燈模式：
SetSensorType(IN_3,IN_TYPE_LIGHT_INACTIVE);
SetSensorMode(IN_3,IN_MODE_PCTFULLSCALE);
ResetSensor(IN_3);

這裡的感測值介於 0~100 之間，如果要更精密的話，可以改變感

測 MODE

```
SetSensorType(IN_1, SENSOR_TYPE_LIGHT);
SetSensorMode(IN_1, SENSOR_MODE_RAW);33
if ((SENSOR_1 < 100) || (SENSOR_1 > 750)){ ...}
```

在 RAW\模式中，SENSOR 的值會介於 1~1023 之間，而 LIGHT_SENSOR 的 RAW 會落在(亮)300~800(暗)間，值越大反而是代表越暗。

3. 聲音感測 Sound Sensor

設定 IN_2 聲感

```
SetSensorSound(IN_2);
```

讀值 SENSOR_2

```
int val= SENSOR_2;
```

val 的值介於 0~100

4. 超音波感測 Ultrasonic Sensor

設定 IN_4 超感

```
SetSensorLowspeed(IN_4);
```

讀值

```
int val= SensorUS(IN_4);
```

這個值是公分