

東南技術學院

學生專題研究報告

題目：以視覺系統作物件判別之探討
學校：東南技術學院
班級：機械二技四年A班
指導老師：黃正光老師
學生：林世杰 3902022
 繆君鴻 3902026
 張順欽 3902037
 盧浩光 3902041

摘 要

本專題以影像辨認為主，以影像處理輔。是利用 CCD 連接 PLC 擷取影像，之後透過 Visual Basic 程式進行像素修改、光源等等條件的修改以利其辨別其顏色、大小、形狀。

影像處理應用在工業界已有多年的的情形了，隨著工業界的需求影像處理已不是一門簡單的學問了。本專題利用較簡單的影像處理來使各位進一步的了解到對影像處理這一門學問內容，並使各位了解目前工業界影像處理的程度。

目 錄

前 言

第 一 章	視覺系統的介紹	2
1 - 1	視覺系統的架構	2
1 - 2	顏色的判別	4
1 - 3	形狀辨別的方法	6
第 二 章	實驗架構	8
2 - 1	P L C 實驗架構	8
2 - 2	P L C 程式	1 2
2 - 3	V B 程式說明	1 5
第 三 章	實驗結果探討	2 5
結 論	3 0
引 用	3 1
使用設備圖	3 2

前 言

在近幾年科技製造技術的發達，光學業大幅的成長，以影像輔助製程的應用也越來越多，因此我們想藉由專題製作對這方面有一些初步的探討，使我們對影像處理的技術能更為加深認識，所以我們就選擇使用 Visual Basic 配合 PLC 和 CCD 作自動影像擷取的判別，而機械視覺顧名思義就是人的視覺，機械使用電腦控制感應器處理判別物件，而人眼所辨識影像的能力遠在電腦之上，但是在於物件判別的速度上，電腦計算的本領卻遠超過人類之上。

本專題主要是製作自動擷取影像之製程並判別物件，而我們選用 Visual Basic 的程式對影像做分析處理，在判別物件方面，是判別顏色和形狀，顏色是以 RGB 三色去分析，形狀方面以判別圓形和非圓形，而主要使用 Visual Basic 的程式去執行 PLC 及 CCD 的作動，而 PLC 接收程式後控制直流馬達之動作與靜止，配合 CCD 擷取圖案後傳回 Visual Basic 的程式中進行判別。

研究動機主要的方向是研究應用於工業界的各項檢測，而在使用此種自動化判別上，主要是在一些複雜之圖形方面需要電腦高度的運算，才能達到節省大量的金錢、時間、人力更能提高精確度。

第一章 視覺系統的介紹

1-1 視覺系統的架構

1-1-1 數位化的影像主要具備了下列優點：

(1) 高品質的畫面：一般的類比電子電路要得到較好的輸出結果，必須要做精細的調整和較正，且零件容易受周圍環境的影響而產生誤差、老化等現象。而數位影像訊號較不會受影響，數位電路只要設計無誤則有較佳的可靠度，也具有較高的畫面品質。

(2) 易於變化處理：數位影像處理可從事修飾、辨識以及分析圖像，影像編碼、影像轉換這些技術被廣泛的應用在工業、軍事、醫學、氣象及商業方面。

1-1-2 數位影像的特性

(1) 點素：數位影像處理大都利用數值矩陣的方式來處理圖形資料，矩陣中的元素稱之為點素。

(2) 灰階：在數位矩陣中之元素，其明亮度我們用一量化值來表示稱之為灰階。一般而言，灰階為 2 的冪次方，如 2^8 即表示有 256 灰階。

(3) 濃度分佈圖(gray level histogram)

濃度分佈圖，為描述影像有效的工具之一，在影像處理分析技術中，常常需要了解影像的特性在哪裡。在統計圖中橫座標為灰階亮度值，縱座標為影像中為各個灰階值所含的像素數量。我們可以由圖中判斷出影像亮度分佈的訊息。圖 1-1 為一灰階亮度統計圖，由圖中可以看

出亮度的分布成明顯的兩群。

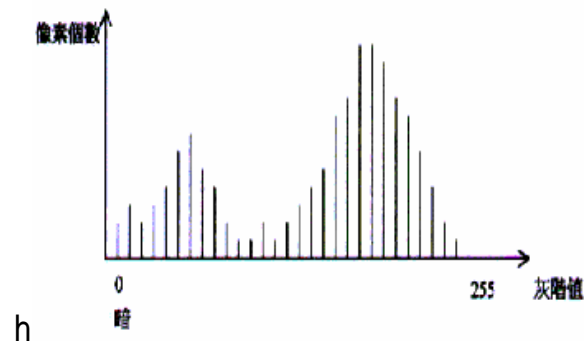


圖 1-1 灰階亮度統計圖

1-1-3 古典的影像處理技術

一般古典的數位影像處理技術包括點素運算(point operation)、空間運算(spatial operation)、轉換運算(transform operation)、假色運算(pseudo color operation)等皆是，至於二值化影像(binanzed images)、自動臨界值擷取(automatic thresholding)、邊緣強化(edge enhancement)等。

(1) 點素運算(point operation)：影像的點素運算處理是對影像的亮度分佈做修正或組合，不過點素運算作業並未將相鄰像素之間的關係考慮進去。

(2) 空間運算：我們可以對相鄰各點做進一步的處理，來提供更多數位影像處理憑據。

(3) 轉換運算：如果想要處理影像之空間頻域特性，如邊緣偵測、高通濾波、低通波等，常用的處理技巧就是利用轉換運算。如 Fourier、CZT、Hilbert 轉換等等，將影像轉至空間頻域後再實施空間濾波。

(4) 假色運算：使用假色運算對照表(look up table)，會得到一逼真又佔太大的記憶體の影像。 【1】

1-2 顏色的辨別

1-2-1 色彩簡介

我們所看到的色彩(color)基本上是眼睛察覺到光所產生的反應。因此色彩是光的一種特性，故可以用物理方法來表示，同時色彩一是人類視覺的一種感受，由於不同人的視覺感受不同，所以對於相同物理量，不同的人會有不同的色彩感受。

1-2-2 彩色系統

任何顏色皆可以紅色、綠色和藍色組合得到，此三個成份構成紅、綠、藍 (RGB) 彩色空間顯示器。然而對於影像處理，它並不是最適合的彩色系統。彩色印刷利用顏料的吸光性質顯示色彩，此為減色系統。對於影像壓縮而言由於眼睛對亮度及色彩的變化不同，因此採用一個將亮度和色彩分開來的彩色系統，將可得到更好的數據效果。

RGB 彩色系統 (Red , Green , Blue)

RGB 彩色系統為一直角座標系統，其中的三個基量是選擇可見光中的三原色光，為使色光定義能夠統一，列出三原色的波長為紅

(700nm) , 綠(546.1nm) , 藍(435.8nm)。

在各種色光混加中，均有一對色光可以混加成白色光，稱為補色光，對紅、綠、藍而言的補色光如下圖 1-2

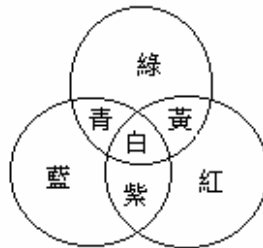


圖 1-2 色光混合圖

CMY 彩色系列 (Cyan , Mengenta , Yellow)

彩色印刷乃利用顏料的吸光特性顯示色調。通常採用青色，紫色和黃色為三原色，它們分別吸收各自的補色光。以這三種顏料依不同的比例如打在白紙上，在白光的照射下，反射出不同比列的紅、綠、藍光，而呈現各種色彩。

由以上分析，我們很容易導得 RGB 和 CMY 系統之間的轉換關係如下

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1-R \\ 1-G \\ 1-B \end{bmatrix}$$

而我們專題的部分是使用 RGB 彩色系統配合 CMY 彩色系統辦示顏色，方法是將所要辦示的顏色取其像素值。例如取一顏色其像素值為 (255 , 100 , 100) = R(255) , G(100) , B(100) 其結果就為紅色。簡單的說就

是辨別其像素值何者為高。

【2】

2-3 形狀辨別的方法

於圖案辨認的基本過程中，可分為劃分圖案的分類過程及圖案的辨認過程。首先，以圖 1-3 所示之圖案分類過程模式所顯現來解釋。

於圖 1-3 所示為圖案分類過程，有關各處理的影像例則說明於後。

1. 輸入：

為取得圖案之 2 維元信號用。此時，其由所用光學系，對影像作攝像，並以 A / D 轉換來作量子化。但因量子化之取樣幅並非極小，於此過程中，會造成輸入資料的劣化及損失，而使後處理造成問題。

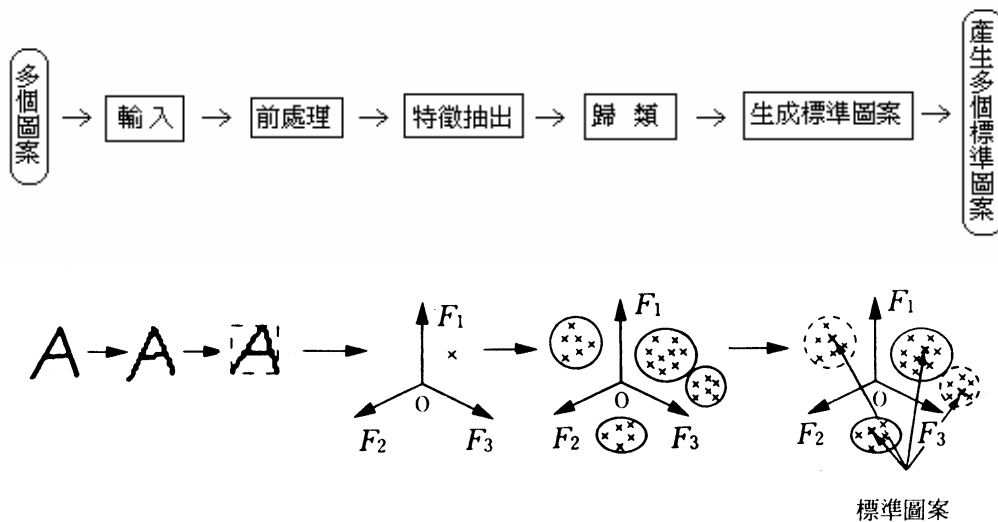


圖 1-3 圖案分類過程模式

2. 前處理：

對輸入固定大小圖案來作擴大或縮小，及作旋轉角度一致等幾何學補正，或對應做除去之雜訊之影像處理。這些處理可使後處理之執行更

容易。

3. 特徵空間的設定：

為使輸入圖案能適切表現，其選擇多個特徵，而設定特徵空間。

4. 特徵抽出：

對輸入圖案，其特徵空間個軸之特徵大小，而使輸入圖案能在特徵空間上得一對應點。

5. 歸類：

對多個輸入圖案之映像所構成特徵空間中之多個點，以多個族群來分割。

6. 標準圖案的生成：

特徵空間中決定代表個群的點。以點為其各群平均特徵，而為標準圖案。

於圖案的分類過程終了後，再做未知圖案的判別。其右圖案的輸入至特徵抽出，為與圖案分類過程相似。而於其後的圖案匹配處理，則對輸入圖案求其特徵，並與先前所述得標準特徵圖案做比較。

如此，以標準圖案來表其特徵空間中最靠近之群下，即可決定輸入圖案之所屬。圖 1-4 為其所示例。如圖所示，為對輸入圖案來與標準圖案之間作比較處理，並於特徵空間，求個圖案與各映像點間距離的處理，然亦有於特徵空間，就標準圖案的代表群存在部分領域內，來檢查是否含輸入圖案，而做圖案的辨別者。

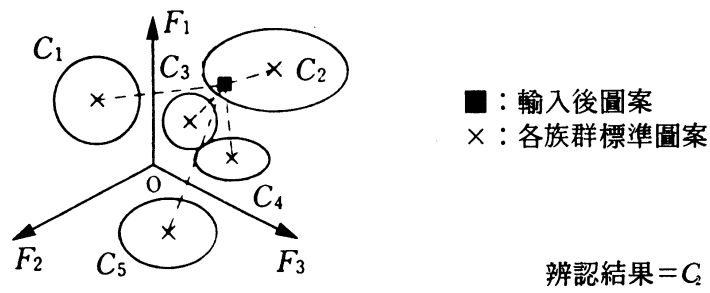


圖 1-4

在輸入圖案與即存個群之標準圖案歧見距離若太大，則輸入圖案無法考量為屬哪個群時，其必要以輸入圖案來購成新族群，而再做歸類處理。

【3】

第二章 實驗架構

2-1 PLC 實驗架構

PLC 控制器 (Programmable Logic Controller) , 乃是一種固態電子裝置, 主要利用輸入 / 輸出裝置的回授信號及儲存程式, 控制機械或程序的操作。在工廠自動化 (FA) 系統中, PLC 因為具備價格便宜、系統穩定及環境適應性佳的特點, 故一直為自動化業界所採用。

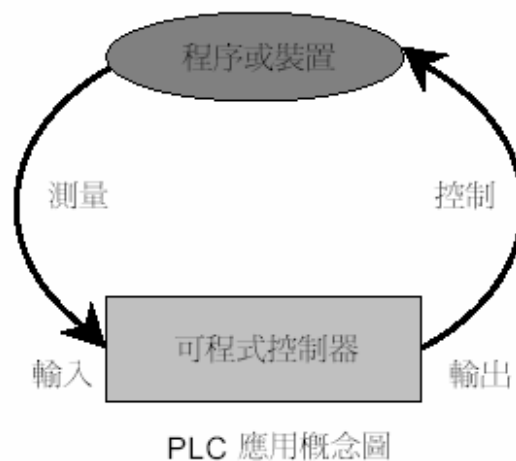
近幾年來, 各 PLC 製造廠家無不致力於新機種的研發, 所以在 CPU 處理速度、擴充模組及通訊的功能上, 相較於早期 PLC 控制器, 已有非常的進展。早期的自動控制都是以繼電器並配合計數器、計時器及感測器等控制元件以完成順序控制的目的。

由於工業的變革, 在生產製造方面, 傳統的方法已無法滿足, 因此有業者根據其需要, 訂出以下的規格, 希望有這樣的控制器, 能夠改善傳統的控制方式, 藉以提高效率, 增加產量。

1. 編程簡單方便, 可以在現場修改程序。
2. 硬體維護與檢修簡單。
3. 高度可靠性, 能適應惡劣的工作環境。
4. 體積小。
5. 資料之讀取和連線容易。
6. 價格具競爭性。
7. 擴展時, 原系統只做很少的改變

基於以上之規格及廣大市場之需求，促使 PLC 迅速的發展。

PLC 能儲存如順序指令、計時器指令、計數器指令、演算指令、資料控制指令以及通訊指令，並藉以控制工業的機械裝置及流程。如下圖所示為 PLC 在應用方面的概念圖。



PLC 是以微處理機為基礎，綜合了計算機與自動化技術而開發的工業控制裝置，這種控制器按照 IEC 國際標準的定義是可程式邏輯控制器又稱為 PC 或 PLC。這種裝置，內部儲存預先編寫相應的程序，待輸入輸出連接完成之後，PLC 運行這個程式，便可自動完成預定的操作，屬於一種程式記憶型的電子控制裝置，亦可視為一般的繼電器或計時器、計數器等集合體。

PLC 之操作模式乃藉監測按鍵 感應器及開關等等輸入之信號，假如這些輸入的信號被偵測到改變，控制系統就會透過使用者所已經載入的程式作動作，產生相對應的輸出信號。

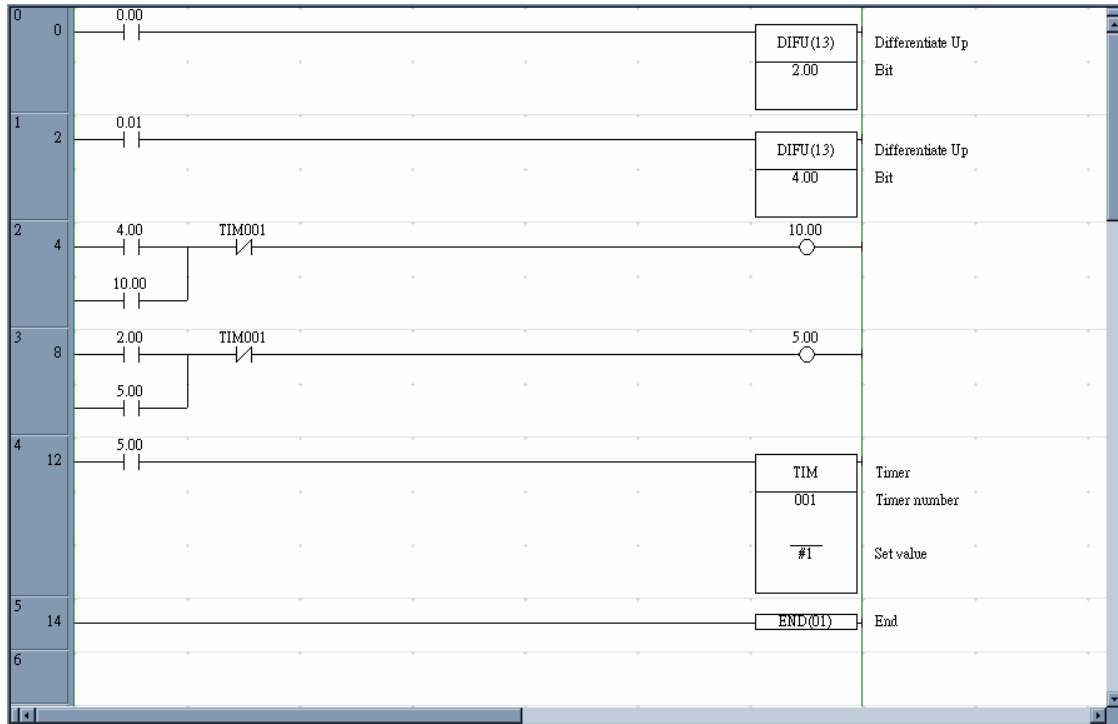
一般 PLC 以下的功能規格,使用者可根據該規格,評估該 PLC 是否符合所需之應用。

1. 輸入點數：該 PLC 能處理的信號點數。
2. 輸出點數：該 PLC 能控制的輸出點數。
3. 計時器數量：該 PLC 內部模擬的計時器數量。
4. 計數器數量：該 PLC 內部模擬的計數器數量。
5. PLC 支援指令：該 PLC 所能處理及涵蓋的指令範圍。
6. 執行速度：PLC 執行每個指令的執行度，這關係到整個系統的掃描時間(Scan-time)。
7. 使用者的程式空間：即 PLC 程式所佔有的最大空間。
8. 程式輸入的方式：由專有的書寫器輸入或是由其它方式載入。
9. 系統的擴充性：該 PLC 可以再作哪一方面的擴充等等。

【4】

3-2 PLC 程式

階梯圖：

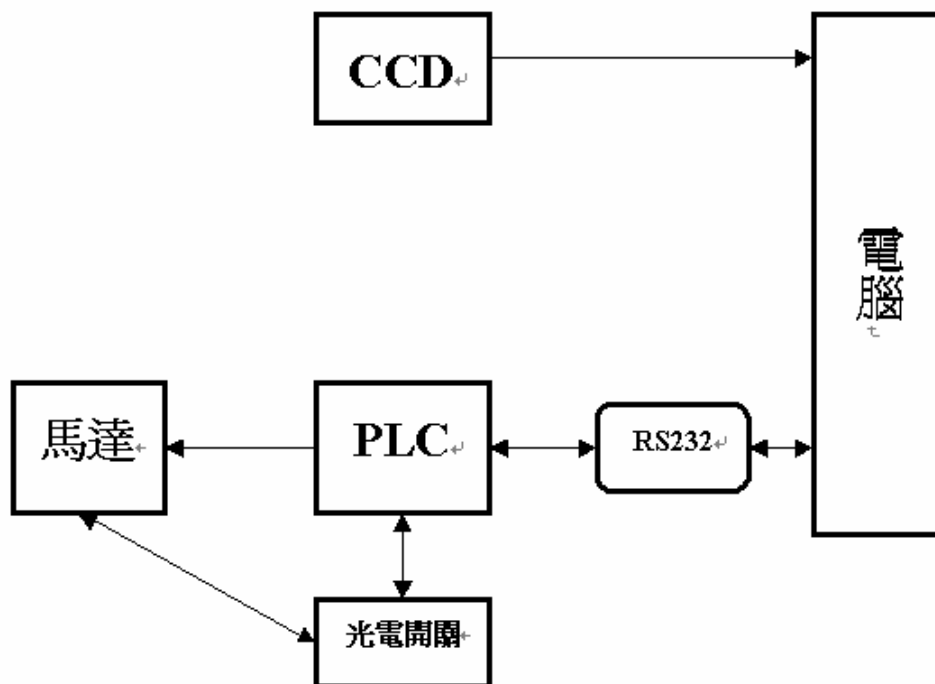


在此階梯圖中用到了上微分和下微分是因為在觸發光遮斷器後會發生下次想在驅動馬達時光遮斷器那邊會一直是 OFF 導致無法驅動馬達所以我們使用上微分和下微分的方式觸發開關。

程式碼：

Rung	Step	Instruction	Operand	Value	Comment
0	0	LD	0.00		
	1	DIFU(13)	2.00		
1	2	LD	0.01		
	3	DIFU(13)	4.00		
2	4	LD	4.00		
	5	OR	10.00		
	6	ANDNOT	TIM001		
	7	OUT	10.00		
3	8	LD	2.00		
	9	OR	5.00		
	10	ANDNOT	TIM001		
	11	OUT	5.00		
4	12	LD	5.00		
	13	TIM	001		
			#1		
5	14	END(01)			

機台架構圖



3-3 VB 程式說明

宣告此一程式之變數

Dim comd1, comd2 As String

'宣告命令 1, 命令 2 的字串

Dim rd, timercounter As Integer

'宣告計數器數值為一整數

Dim isb, iso, isr, isg, isgl As Integer

'判斷函數

Dim pointn, area As Long

宣告點與面積

Dim p()

宣告量測範圍

Dim ppx(1 To 50000), ppy(1 To 50000) As Integer

宣告量測範圍座標值

Dim ppx1, ppy1, avx, avy As Double

宣告重疊部分

Dim boundx(5000), boundy(50000)

宣告圈選範圍部分

Dim rota1, plcc As Boolean

'宣告電腦與 plc 之變數

設定訊號延遲時間

Private Sub delay(delay_n%)

延遲時間設定

For i% = 1 To delay_n%

設定延遲時間起始值 i=1

For j% = 1 To 20

延遲時間由 1-20 秒

pp = 10 * Sin(10 * (j% / 20))

Next j%

Next i%

End Sub

編寫此程式所需的運算式

Sub fscal(T1, T2 As String)

'設定程式內部運算式

l% = Len(T1)

設定 T1 字串長度

a = 0

For i = 1 To l%

Tp = Mid(T1, i, 1)

'Tp 為從 T1 字串的第 i 個字元取一個字元

a = Asc(Tp) Xor a

將字元轉換成數碼

Next i

T2 = Hex(a)

If Len(T2) = 1 Then

T2 = "0" + T2

End If

End Sub

將按鈕 1 設定為按下後,啟動電腦與 plc 的連線

Private Sub Command1_Click()

設定按鈕 1 的功用

plcc = True

當按鈕 1 按下時啟動 plc

If Not (Timer1.Enabled) Then

當按鈕 1 按下時計時器 1 啟動

Timer1.Enabled = True

```

End If
If Not (MScmm1.PortOpen) Then          當按鈕 1 按下時通訊埠開啟
    MScmm1.PortOpen = True
End If
Shape2.Visible = False                訊號燈關閉

```

End Sub

設定按鈕 2 按下後,CCD 開始擷取托盤上的影像,此時會跳出一個小視窗(此視窗為 CCD 驅動程式內建值),被影像擷取下來後顯示在表單的圖框中,圖框中的影像將被儲存在預設的檔案中(此檔案的圖像每變更一次,原有的圖像將被新的取代,因此可以重複判別新的影像),程式將判別檔案中圖像的外型與顏色,並顯示於表單中.

```

Private Sub Command2_Click()          設定按鈕 2 的功用
isb = 0
isg = 0
isy = 0
iso = 0
isr = 0
pointn = 0
timercounter = 0                    計數器設為 0
If Timer1.Enabled Then
    Timer1.Enabled = False          計時器 1 關閉
End If
If Not (Timer2.Enabled) Then
    Timer2.Enabled = True          計時器 2 開啟
End If
Picture1.AutoSize = True            圖框設為自動調整大小
Picture1.AutoRedraw = True          圖框內容有變更時畫面更新
Picture1.ScaleMode = 3              圖框的尺寸單位
py = Picture1.ScaleHeight           設定圖框高度為 px
px = Picture1.ScaleWidth            設定圖框寬度為 py
ReDim p(px, py)                    取代宣告值
temp = 256
temp = temp * 256
cutb = 10
For i = 0 To py - 1
    For j = 0 To px - 1
        If (i < cutb) Or (j < cutb) Or ((py - i) < cutb) Or ((px - j) < cutb) Then
            Picture1.PSet (j, i), RGB(255, 255, 255)

```

```

    p(j, i) = 0
    GoTo jp
End If
Vcolor = Picture1.Point(j, i)
Vblue = Int(Vcolor / temp)
Vgreen = Int((Vcolor Mod temp) / 256)
Vred = (Vcolor Mod temp) Mod 256
'Vtest1 = Vred / (Vgreen + 1)
'Vtest2 = Vblue / (Vred + 1)
'Vtest3 = Vgreen / (Vred + 1)
Vtest1 = Vred / 1
Vtest2 = Vblue / 1
Vtest3 = Vgreen / 1
Vy = 0.29 * Vred + 0.58 * Vgreen + 0.13 * Vblue
toodark = (Vred < 150) And (Vgreen < 100) And (Vblue < 100)
toobright = (Vred > 180) And (Vgreen > 150) And (Vblue > 150)
If toodark Or toobright Then
    'Picture1.PSet (j, i), RGB(255, 255, 255)
    p(j, i) = 0
    GoTo jp
End If
pointn = pointn + 1
p(j, i) = 1
ppx(pointn) = j
ppy(pointn) = i
nb = 70
' Text2.Text = Vtest1
' Text3.Text = Vtest2
' Text4.Text = Vtest3
If (Vtest2 > 100) Or (Vtest3 > 100) Then
    Select Case Vtest2 > 70
    Case True
        isb = isb + 1
        If isb > nb Then
            objcolor = "blue"
            ' isb = 0
            GoTo jp
        End If
    End Select
End If

```

```

Case False
  isg1 = isg1 + 1
If isg1 > nb Then
  objcolor = "green"
' isg = 0
  GoTo jp
End If
End Select
Else
  Select Case Vtest1
Case Is > 150
  isr = isr + 1
  If isr > nb Then
    objcolor = "red"
    GoTo jp
  End If
Case Is < 1.5
  isy = isy + 1
  If isy > nb Then
    objcolor = "yellow"
    GoTo jp
  End If
Case Is > 85
  iso = iso + 1
  If iso > nb Then
    objcolor = "orange"
    GoTo jp
  End If
Case Else
  objcolor = "unkown"
End Select

End If

jp:
  DoEvents
Next j
Next i
area1 = 0

```

```

ppx1 = 0
ppy1 = 0
For k = 1 To pointn
    ppx1 = ppx(k) + ppx1
    ppy1 = ppy1 + ppy(k)
Next k
avx = ppx1 / pointn
avy = ppy1 / pointn
boundn = 0
For j = 1 To py - 1
    For i = 1 To px - 1

        Select Case p(i, j)
            Case Is > p(i - 1, j), Is > p(i, j - 1)
                boundn = boundn + 1
                boundx(boundn) = i
                boundy(boundn) = j
                Picture1.PSet (i, j), RGB(255, 0, 0)
            Case Is > p(i + 1, j), Is > p(i, j + 1)
                boundn = boundn + 1
                boundx(boundn) = i
                boundy(boundn) = j
                Picture1.PSet (i, j), RGB(255, 0, 0)
            Case Else
        End Select
    Next i
Next j
ra = 0
rmax = 0
rmin = 0
If boundn > 1 Then
    rc = Sqr((avx - boundx(1)) ^ 2 + (avy - boundy(1)) ^ 2)
    rmax = rc
    rmin = rc
For i = 1 To boundn
    rc1 = Sqr((avx - boundx(i)) ^ 2 + (avy - boundy(i)) ^ 2)
    If rc1 > rmax Then
        rmax = rc1

```

Else	
If rc1 < rmin Then	
rmin = rc1	
End If	
End If	
Next i	
End If	
If rmin / rmax > 0.75 Then	
objshape = "circle"	此物件為圓形
Else	
objshape = "square"	'此物件為四方形
End If	
Shape2.FillStyle = 0	形狀判別視窗填充模式為充滿
Shape2.Visible = True	顯示形狀判別視窗
Select Case objcolor	物件顏色的種類
Case "blue"	藍色的狀況
Text1.Text = "blue"	當顏色框顯示為藍色時
Shape2.FillColor = RGB(0, 0, 255)	形狀判別視窗填充的顏色為藍色
Case "green"	綠色的狀況
Text1.Text = "green"	當顏色框顯示為綠色時
Shape2.FillColor = RGB(0, 255, 0)	形狀判別視窗填充的顏色為綠色
Case "red"	紅色的狀況
Text1.Text = "red"	顏色框顯示為紅色時
Shape2.FillColor = RGB(255, 0, 0)	判別視窗填充的顏色為紅色
Case "orange"	橘色的狀況
Text1.Text = "orange"	當形狀顯示為橘色時
Shape2.FillColor = RGB(255, 120, 0)	顏色判別視窗填充的顏色為橘色
Case "yellow"	黃色的狀況
Text1.Text = "yellow"	當顏色框顯示為黃色時
Shape2.FillColor = RGB(250, 250, 90)	形狀判別視窗填充的顏色為黃色
Case Else	其他情況
Text1.Text = "unknown"	'顏色框顯示無法識別
End Select	
Select Case objshape	
Case "circle"	當顯示為圓形時,形狀判別視窗框變為
圓形	
Shape2.Shape = 3	


```

Case Else
Shape2.Shape = 1
為正方形
End Select
Timer1.Enabled = True
End Sub

```

當顯示為其他種類,形狀判別視窗變

此按鈕為結束整個執行的程式

```

Private Sub Command3_Click()
End
End Sub

```

按下按鈕 3 結束程式

按下此按鈕電腦與 plc 傳輸訊號,此時 plc 啟動馬達,托盤開始旋轉,等光感應開關有接收訊號時,plc 再傳訊號回電腦,準備下一個動作

```

Private Sub Command4_Click()
Shape2.Visible = False
rotal = True
If Not (MSComm1.PortOpen) Then
MSComm1.PortOpen = True
End If
If Not (Timer1.Enabled) Then
Timer1.Enabled = True
End If
End Sub

```

設定按鈕 4 的功能

開啟通訊埠

開啟計時器

此部份為設定表單被開啟時的基本設定,表單被開啟時,計時器與通訊埠為關閉狀態,並設定通訊埠通訊模式,以及燈號的設定和影像擷取後存取之路徑與檔案格式

```

Private Sub Form_Load()
Timer1.Enabled = False
Timer1.Interval = 100
Timer2.Enabled = False
Timer2.Interval = 1000
If MSComm1.PortOpen Then
MSComm1.PortOpen = False
End If
Timer1.Interval = 10
MSComm1.CommPort = 2
MSComm1.OutBufferSize = 10000
MSComm1.InBufferSize = 10000
MSComm1.Settings = "9600,e,7,2"

```

表單基本設定

```

Shape1(0).FillStyle = 0
Shape1(0).FillColor = RGB(90, 0, 0)
Shape1(1).FillStyle = 0
Shape1(1).FillColor = RGB(90, 0, 0)
Shape2.Visible = False
rd = 90
ImgScan1.ScanTo = FileOnly
ImgScan1.FileType = BMP_Bitmap
ImgScan1.Image = "c:\專題製作\2002\vision\testvb\ccd1\temp.bmp"
ImgScan1.PageOption = OverwriteAllPages
plcc = False
rotal = False
isb = 0
isg = 0
isy = 0
iso = 0
isr = 0

```

End Sub

此部份為設定影像被擷取之後,圖框讀取影像檔之路徑

```

Private Sub ImgScan1_ScanDone() 影像擷取
Picture1.Picture = LoadPicture("c:\專題製作\2002\vision\testvb\ccd1\temp.bmp") 圖框顯示此路
徑中的圖檔
End Sub

```

```

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

```

'用滑鼠圈選圖框中想要的部分

```

Vcolor = Picture1.Point(X, Y)
temp = 256
temp = temp * 256
Vblue = Int(Vcolor / temp)
Vgreen = Int((Vcolor Mod temp) / 256)
Vred = (Vcolor Mod temp) Mod 256

```

End Sub

此部份為設定電腦如何與 plc 做溝通,讓電腦傳送 plc 接受的訊號格式與接收 plc 傳回的訊號

```

Private Sub Timer1_Timer() 設定計時器 1

```

```

If rd = 200 Then
    Shape1(0).FillColor = RGB(rd, 0, 0)
    rd = 90
Else
    Shape1(0).FillColor = RGB(rd, 0, 0)
    rd = 200
End If
If (plcc = True) And (rota1 = False) Then
    comd1 = "@00RR00000001"
End If
If (plcc = True) And (rota1 = True) Then
    comd1 = "@00WR00000002"
    rota1 = False
End If

Call fscal(comd1, comd2)
outstring = comd1 + comd2 + "*"
MSComm1.Output = outstring + Chr(13)
Call delay(2000)
instring = MSComm1.Input
tempstring = instring
If Len(tempstring) > 2 Then
    le = Len(tempstring)
    tempstring = Left$(tempstring, le - 2)
    tempstring = Right$(tempstring, 1)
End If
DoEvents
If tempstring = "1" Then
    ImgScan1.StartScan
    Timer1.Enabled = False
End If
End Sub
此部份為控制當電腦與 plc 做傳輸訊號於燈號的變化
Private Sub Timer2_Timer()
    設定計時器 2
DoEvents
If rd = 200 Then
    Shape1(1).FillColor = RGB(rd, 0, 0)
    rd = 90

```

```
Else
Shape1(1).FillColor = RGB(rd, 0, 0)
rd = 200
End If
timercounter = timercounter + 1
If timercounter = 5 Then
If (MSComm1.PortOpen) And (Not (Timer1.Enabled)) Then
Timer1.Enabled = True
End If
Timer2.Enabled = False
End If
End Sub
```

第三章 實驗結果探討

本專題使用 Visual Basic 的程式對影像做分析處理，在程式進行方面，一開始先和 PLC 作連接，連接後執行影像擷取，轉盤會轉動到預定位置並執行 CCD 拍照，在進行影像分析時會得到顏色及形狀得判別，如圖 3-2，在此程式中較重要的元件就是“影像掃描”，此元件最主要的功用為在 Visual Basic 的程式中能和 CCD 相互配合。

在對圖形的顏色判別方面，可判別出紅、藍、綠三種顏色，而我們是以 RGB 三色系去分別出三種不同的色彩，例如：紅色(255, 0, 0)綠色(0, 255, 0)藍色(0, 0, 255)，圖 3-1(ABC)為實驗結果的圖形，可同時測出顏色和形狀。

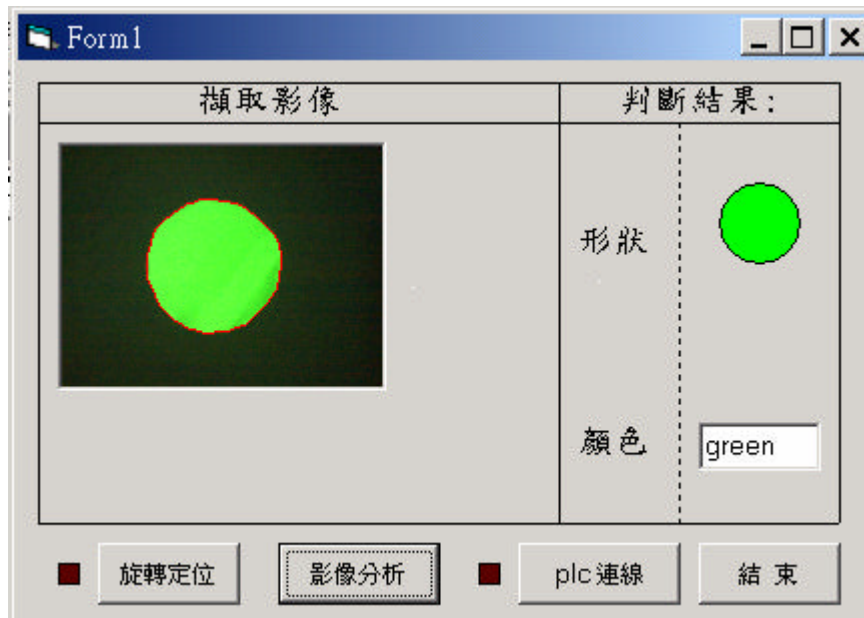


圖 3-1(A)

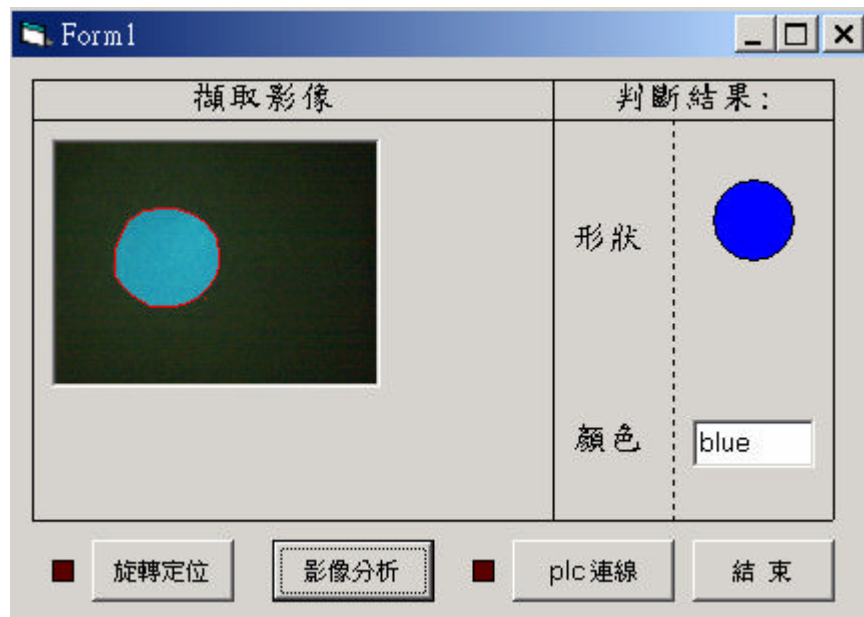


圖 3-1(B)

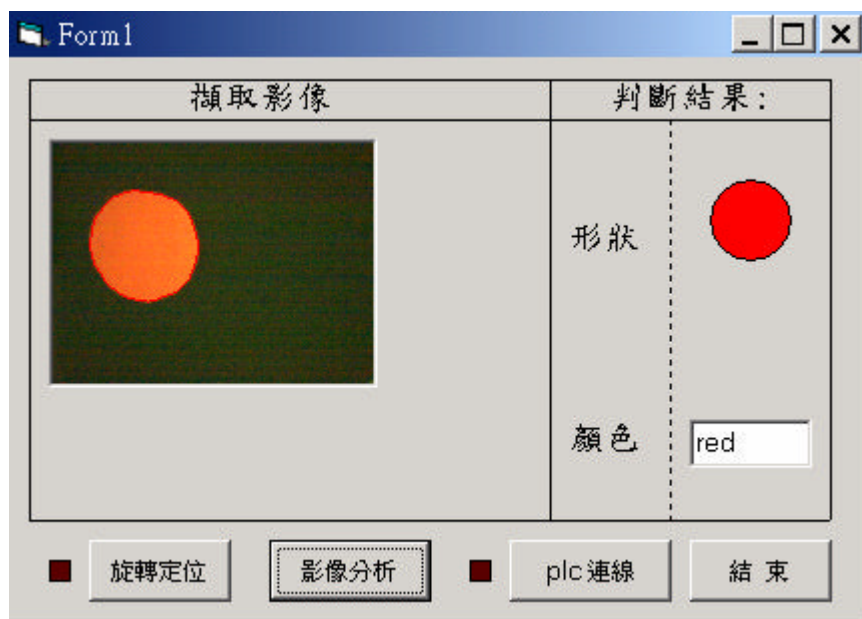


圖 3-1(C)

在對形狀方面我們針對圓形和正方形進行分析，其分析的圖形如圖 3-2(ABC)，我們根據邊緣距離中心點的距離來判別正方形和圓形，如果是正方形因它的最大半徑和最小半徑的差會等於 $\sqrt{2} - 1$ ，所以比值是 $1 : 1 : \sqrt{2}$ ，圓形方面的半徑都相等，即可分辨兩圖形。

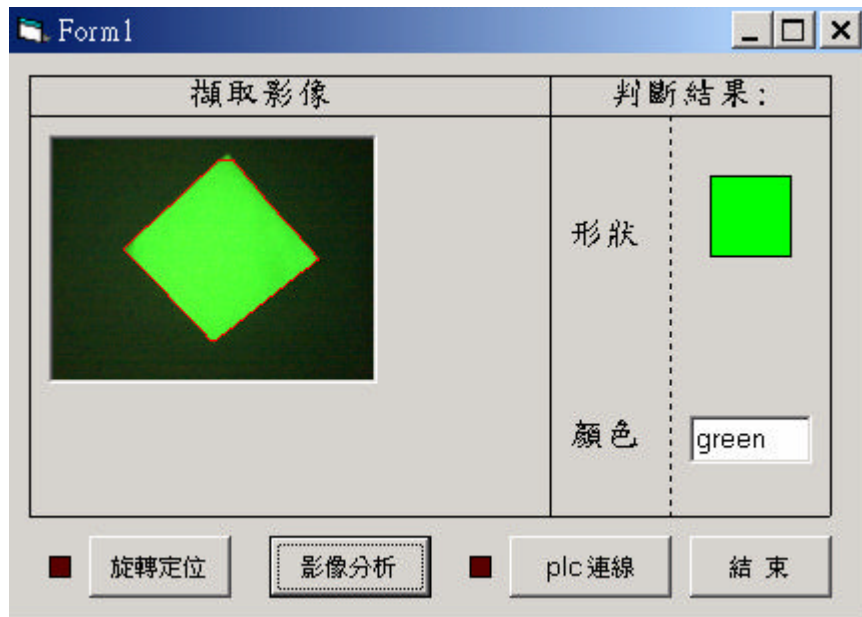


圖 3-2(A)

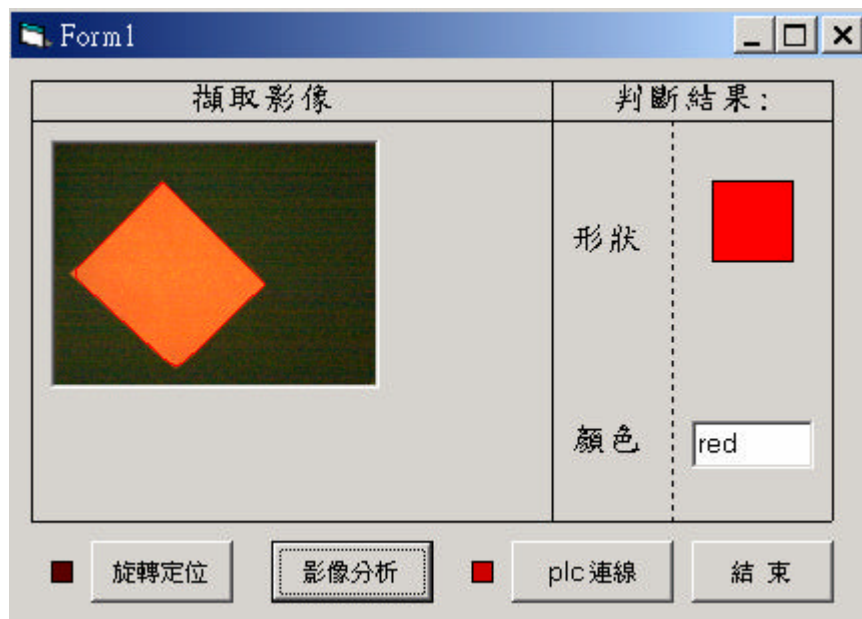


圖 3-2(B)

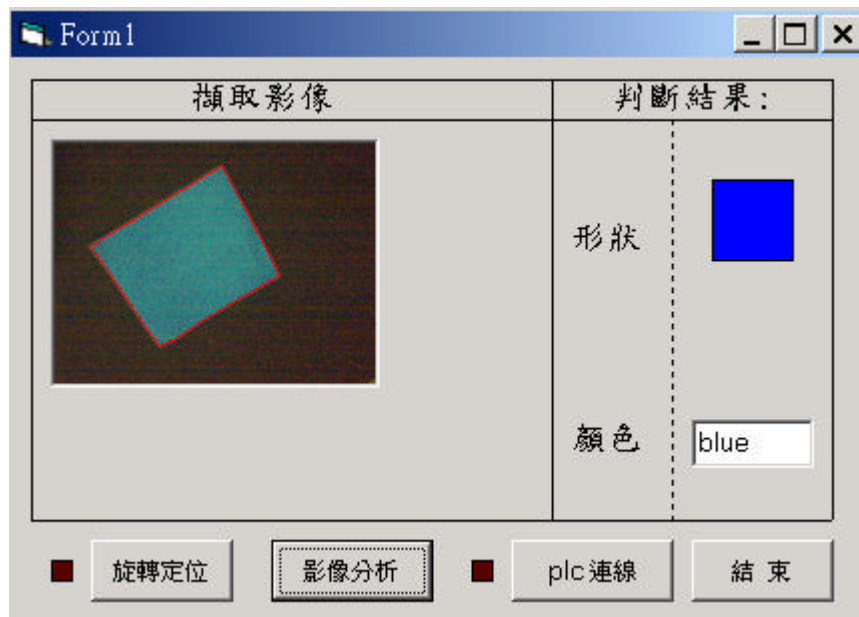


圖 3-2(C)

在光源方面，對 CCD 影像擷取圖形有相當大的影響，如圖 3-3(AB)，打光設備為黃色之照光設備和未調整過亮度在一般日光燈下所照取之圖形，圖 3-3(A)因燈光太強烈，使得在尋找邊界時亮度太強而變成無法辨識正確之圖形，而且被測物之色澤也因強光而偏離本色，無法得到實物知本體，圖 3-3(B)因在日光燈下照取，也因光源太強使得全部曝光，所以在我們探討調校後，我們以降低亮度來取得圖形，而在工業界方面，使用的打光設備都是特殊的 CCD 照光設施，以做成圓狀套於 CCD 鏡頭上，而其價格也較昂貴。



圖 3-3(A)

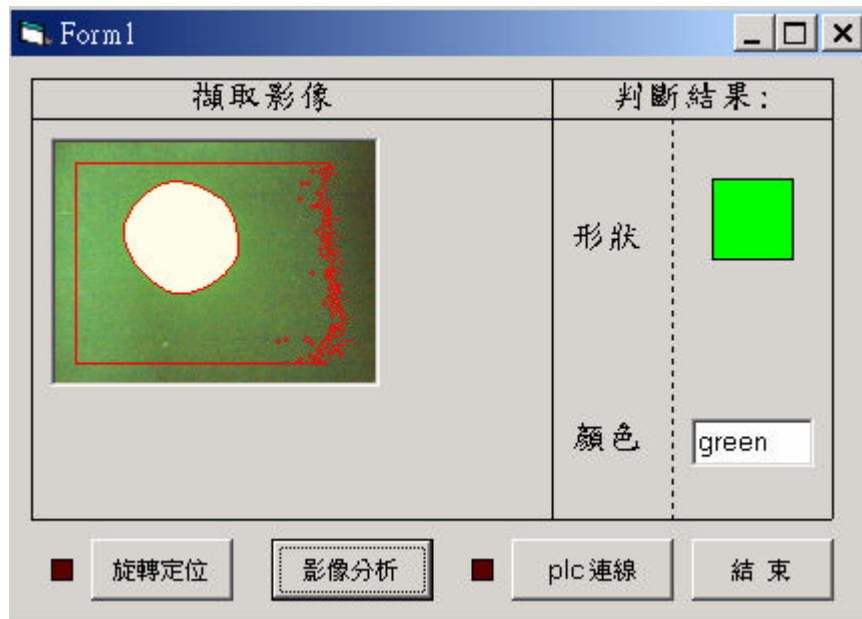


圖 3-3(B)

結 論

本專題實驗的成果豐碩，成功的利用了一般 CCD 連接 PLC 再使用 VB 程式分析出了紙張的顏色及形狀。其成果是利用一般的 CCD 擷取影像，再收尋各種資料後決定使用 PLC 控制 CCD 和各作動部份使其運轉。

專題剛開始遇到的問題就是，彩色影像擷取的問題，我們使用 CCD 成功的解決影像的問題。其次是光源的問題，因為光源影響了色素的改變，天色較亮其辨別出來的顏色就較白，天色較暗，辨別出來的顏色較深。待各問題解決了之後，便連接所有結構組成實體，以一次作動的原則來進行。

引 用

【1】引用自林宸生、邱創乾、陳德請之 ” 數位信號處理” , 1996。

取自張弘 ” 幾何光學 ” 東華書局 , 1987 年 10 月。

【2】引用自連國珍 ” 數位影像處理 ” 第十章 儒林書局 1992 年 9 月。

【3】引用自楊武智 ” 影像處理與辨認 ” 第六章 1994 年 3 月。

【4】可程式控制原理與應用 編著：廖文輝 校閱：周志宏 教授

<http://www.deltaww.com/tw/default.htm>

使用設備圖



附圖一. 整體架構圖



附圖二. PLC 構造圖



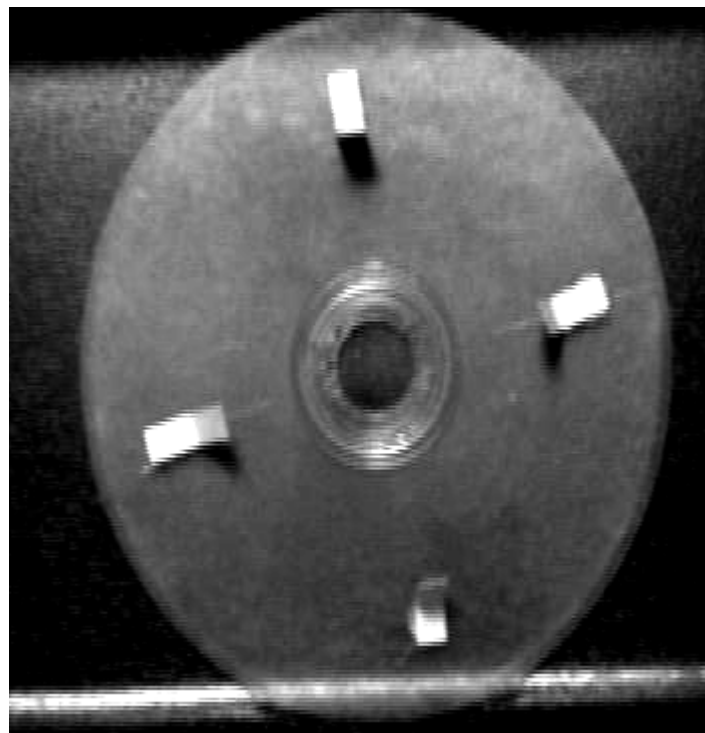
附圖三. CCD 構造圖



附圖四. 光感測系統



附圖五. 照光設備



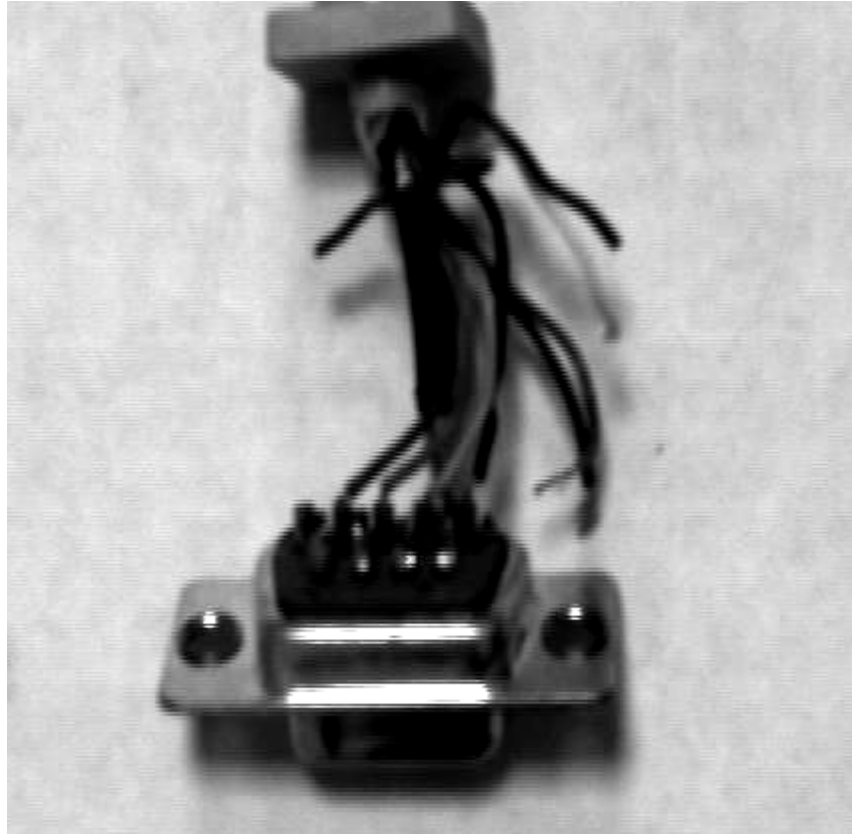
附圖六. 圓盤轉動器



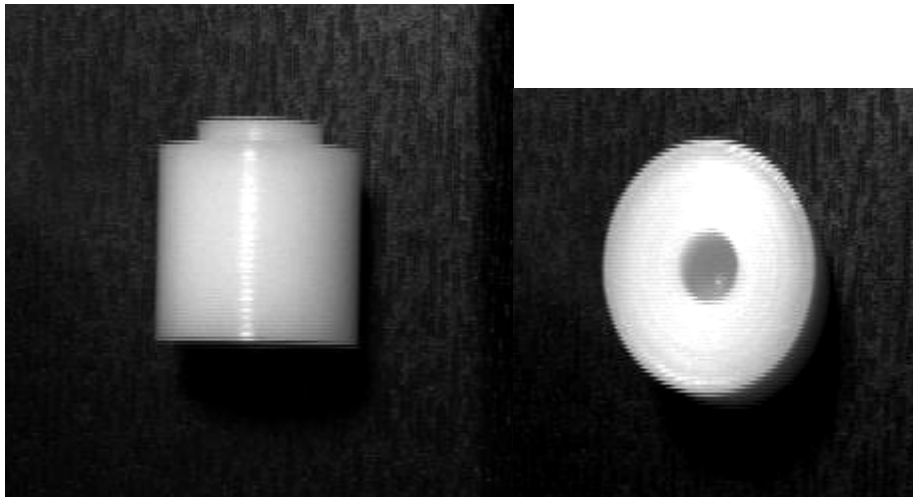
附圖七. RS232



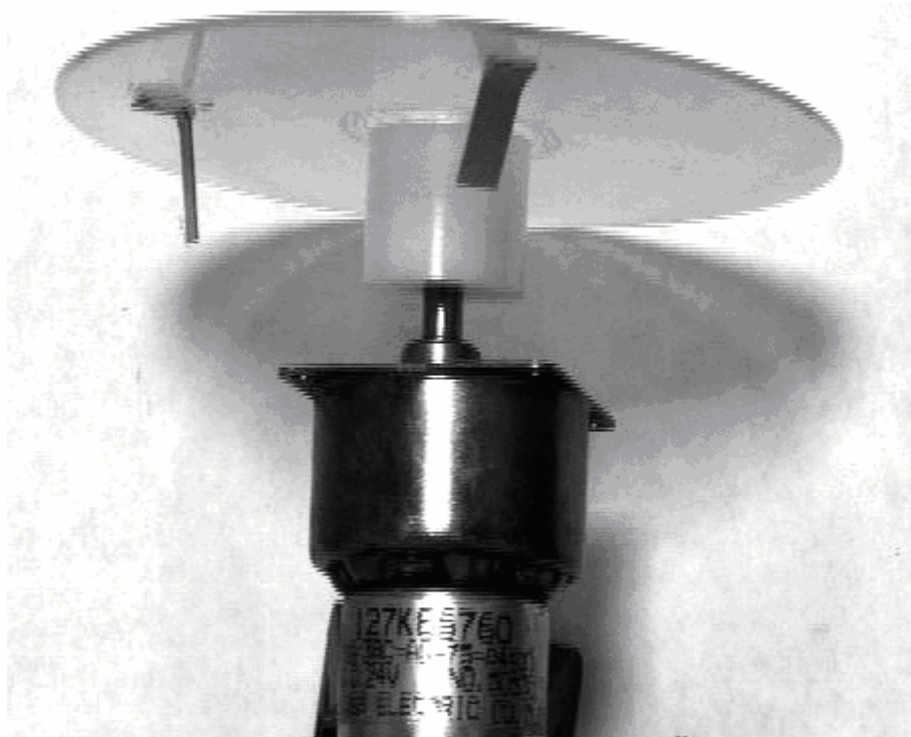
附圖八. 9PIN 連接阜



附圖九. 9PIN 連接埠的接頭連接線圖



附圖十. 馬達與轉盤中間之連接軸



附圖十一. 馬達驅動轉盤構造圖



附圖十二. 直流馬達



附圖十三. 可變電壓器