

## 第 4 章 目標物件的建立與篩選



### 4-1 目標物件定義

前一章中已經做到可以將目標變成一個一個的封閉曲線，本章中我們要進一步實作出「目標物件」。就是將可視的黑色區塊目標，變成有寬高或對比強度等屬性的程式物件，這在物件導向程式的架構來說就是要建立一個類別，但是在 VB 程式的架構下有較為簡單的語法，就是建立一個 **Structure** 結構來定義一種物件的靜態屬性。因為他們只是被其他程式操作的靜態物件，並不需要建立包含事件或方法的較複雜 **Class**(類別)。程式碼如下：

'目標物件結構

Public **Structure** TgInfo

Dim np As Integer '目標點數

Dim P As ArrayList '目標點的集合

Dim xmn As Short, xmx As Short, ymn As Short, ymx As Short '四面座標極值

Dim cx As Integer, cy As Integer '目標中心點座標

Dim width As Integer, height As Integer '寬與高

Dim pm As Integer '目標與背景的對比強度

Dim ID As Integer '目標依據對比度的排序

End **Structure**

當然一個目標物件可以建立非常多的屬性，後續有需要時還可以在此繼續新增。為繼續本章的實作說明，請先建立一個類似前章的程式專案，一樣匯入 **FastPixel** 模組，建立主功能表的下列幾個按鍵：**Open**, **Binary**, **Outline**, **Targets**, **Filter** 與 **Sort**。



在表單程式碼的最前方也請先建立好以下的全域變數：

```
Dim B() As Byte '灰階陣列
Dim Z() As Byte '全圖二值化陣列
Dim Q() As Byte '輪廓線陣列
Dim Gdim As Integer = 40 '計算區域亮度區塊的寬與高
Dim Th() As Integer '每一區塊的平均亮度，二值化門檻值
Dim C As ArrayList '目標物件集合
Dim Mb As Bitmap '顯示目標的影像
```

與前兩章稍有不同的是，我們用 **B** 陣列代表灰階陣列，如果你想用綠光亮度作為灰階，只要在取得 **RGB** 資訊後設定 **B=Gv** 即可。對於同一張影像的辨識案例，我們常常需要在各個流程中回溯使用到該圖的灰階(**B**)、輪廓(**Q**)與二值化(**Z**)陣列，所以將它們升格為全域變數是有必要的！目標物件集合 **C** 也是一樣的狀況，基本上它就是辨識程序後半段的主角。

**Open** 按鍵程式內容微調如下：

```
If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
    Dim bmp As New Bitmap(OpenFileDialog1.FileName)
    Bmp2RGB(bmp) '擷取影像資訊
    B = Gv '以綠光為灰階
    PictureBox1.Image = bmp '顯示
End If
```

## 4-2 二值化與輪廓化的功能整併

為了讓讀者逐步深入，充分理解每一個辨識流程，在每一章都能專注於新介紹的程序，我們的程式專案都是滾動式進化的！前面已經學會的程序會在下一章的程式專案中，逐步被封裝成一些獨立的副程式，不需要一一展示的過程影像功能也會被省略，這一章我們的 **Binary** 功能程式碼就整合成這樣了：

```
'二值化
Private Sub BinaryToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles BinaryToolStripMenuItem.Click
    Th = ThresholdBuild(B) '建立二值化使用之門檻值陣列
    ReDim Z(nx - 1, ny - 1) '建立二值化陣列
    For i As Integer = 1 To nx - 2
        Dim x As Integer = i \ Gdim 'x 座標換算
        For j As Integer = 1 To ny - 2
```

```

        Dim y As Integer = j \ Gdim 'y 座標換算
        If B(i, j) < Th(x, y) Then
            Z(i, j) = 1 '低於亮度門檻設為目標點
        End If
    Next
Next
Dim bmp As Bitmap = BWImg(Z) '建立二值化圖
PictureBox1.Image = bmp '顯示
End Sub
'門檻值陣列建立
Private Function ThresholdBuild(ByVal b(,) As Byte) As Integer(,)
    Dim kx As Integer = nx \ Gdim, ky As Integer = ny \ Gdim
    Dim T(kx, ky) As Integer
    '累計各區塊亮度值總和
    For i As Integer = 0 To nx - 1
        Dim x As Integer = i \ Gdim
        For j As Integer = 0 To ny - 1
            Dim y As Integer = j \ Gdim
            T(x, y) += b(i, j) '亮度值累加
        Next
    Next
    '區塊亮度平均值計算
    For i As Integer = 0 To kx - 1
        For j As Integer = 0 To ky - 1
            T(i, j) /= Gdim * Gdim
        Next
    Next
    Return T
End Function

```

門檻值的建立過程變成一個 **ThresholdBuild** 副程式，在 **Binary** 程序中直接被呼叫使用。以本章的新影像範例執行二值化的結果如下：



接下來是 Outline 的程式碼與執行結果：

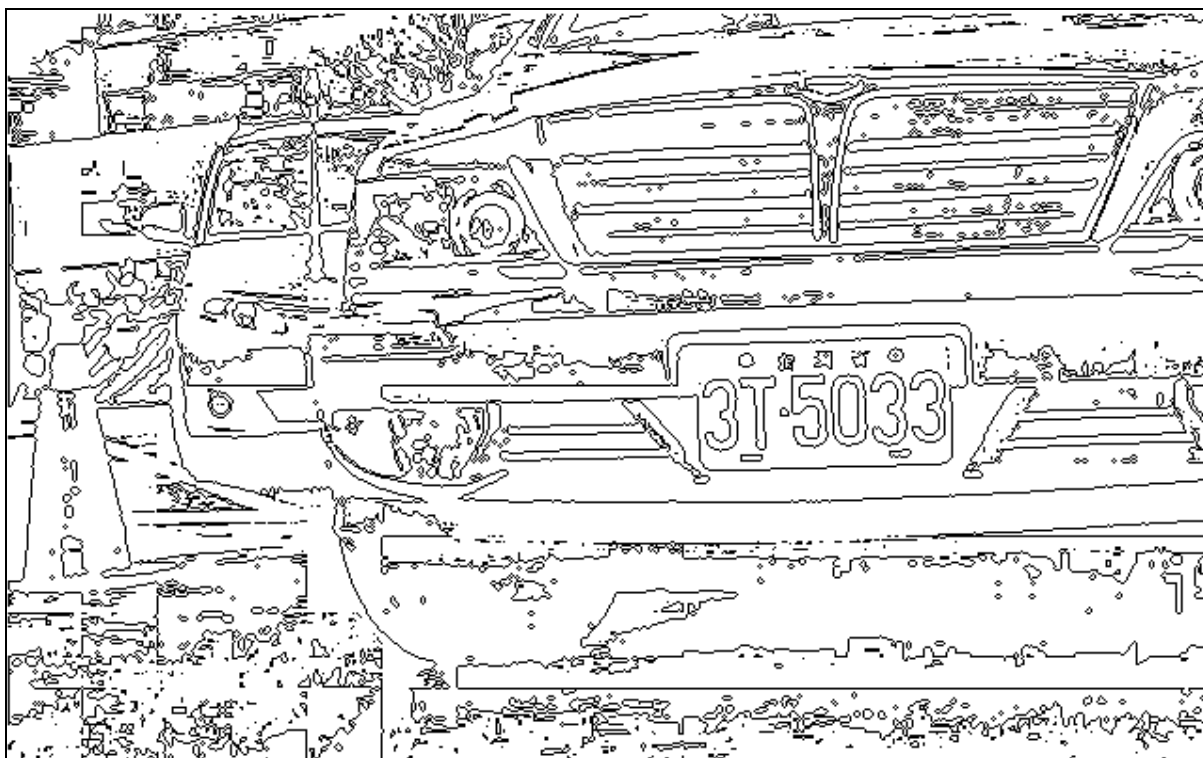
'輪廓線

```
Private Sub OutlineToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles OutlineToolStripMenuItem.Click
    If IsNothing(Z) Then Exit Sub '無二值化圖忽略
    Q = Outline(Z) '建立輪廓點陣列
    Dim bmp As Bitmap = BWImg(Q) '建立輪廓圖
    PictureBox1.Image = bmp '顯示輪廓圖
```

End Sub

'建立輪廓點陣列

```
Private Function Outline(ByVal b(,) As Byte) As Byte(,)
    Dim Q(nx - 1, ny - 1) As Byte '輪廓點陣列
    For i As Integer = 1 To nx - 2
        For j As Integer = 1 To ny - 2
            If b(i, j) = 0 Then Continue For '非輪廓點忽略
            If b(i, j - 1) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
            If b(i - 1, j) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
            If b(i + 1, j) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
            If b(i, j + 1) = 0 Then Q(i, j) = 1 '確認為輪廓點
        Next
    Next
    Return Q
End Function
```



### 4-3 建立目標物件

前一章我們已經介紹了使用氾濫式演算法找出一個封閉曲線的程式，在此就是要將程式改成可以巨細靡遺的將「所有」的獨立曲線都分離出來，並以這些曲線為依據建立出圖上所有的目標物件。就是將輪廓線矩陣饋入一個 **getTargets Function** 副程式，並回傳一個包含所有目標物件的集合物件。程式碼如下：

'以輪廓點建立目標陣列，排除負目標

Function getTargets(ByVal q() As Byte) As ArrayList

Dim A As New ArrayList

Dim b() As Byte = q.Clone '建立輪廓點陣列副本

For i As Integer = 1 To nx - 2

For j As Integer = 1 To ny - 2

If b(i, j) = 0 Then Continue For

Dim G As New TgInfo

G.xmn = i : G.xmx = i : G.ymn = j : G.ymx = j : G.np = 1 : G.P = New ArrayList

Dim nc As New ArrayList '每一輪搜尋的起點集合

nc.Add(New Point(i, j)) '輸入之搜尋起點

G.P.Add(New Point(i, j))

b(i, j) = 0 '清除此起點之輪廓點標記

Do

Dim nb As ArrayList = nc.Clone '複製此輪之搜尋起點集合

nc = New ArrayList '清除準備蒐集下一輪搜尋起點之集合

For m As Integer = 0 To nb.Count - 1

Dim p As Point = nb(m) '搜尋起點

'在此點周邊 3X3 區域內找輪廓點

```

For ii As Integer = p.X - 1 To p.X + 1
    For jj As Integer = p.Y - 1 To p.Y + 1
        If b(ii, jj) = 0 Then Continue For '非輪廓點忽略
        Dim k As New Point(ii, jj) '建立點物件
        nc.Add(k) '本輪搜尋新增的輪廓點
        G.P.Add(k)
        G.np += 1 '點數累計
        If ii < G.xmn Then G.xmn = ii
        If ii > G.xmx Then G.xmx = ii
        If jj < G.ymn Then G.ymn = jj
        If jj > G.ymx Then G.ymx = jj
        b(ii, jj) = 0 '清除輪廓點點標記
    Next
Next
Next
Next
Loop While nc.Count > 0 '此輪搜尋有新發現輪廓點時繼續搜尋
If Z(i - 1, j) = 1 Then Continue For '排除白色區塊的負目標，起點左邊是黑點
G.width = G.xmx - G.xmn + 1 '寬度計算
G.height = G.ymx - G.ymn + 1 '高度計算
A.Add(G) '加入有效目標集合
Next
Next
Return A '回傳目標物件集合
End Function

```

先解釋一下這個 **Function** 的內容，饋入的輪廓點陣列會先建立一個副本，以此副本在全圖做左上到右下的依序搜尋，碰到任何輪廓點就先建立一個空的 **Tginfo** 目標物件 **G**，以此點為起點，做一個氾濫式演算法的搜尋，蒐集所有這個封閉曲線內的輪廓點到 **G.P** 集合，同時也逐步建立此目標的上下左右四個座標極值 **xmn**, **xmx**, **ymn** 與 **ymx**，最終可以用此四個極值算出目標的寬(**width**)、高(**height**)與中心點座標等屬性。

在氾濫式演算進行中，已經被找到並記錄於目標內的輪廓點(副本)就會被清除，所以不會有重複搜尋的問題，雖然看起來輪廓點很多，計算時間還算合理，不會太慢。也因為輪廓點會被逐步擦掉，所以我們一開始才要建立副本，副本的輪廓點終究會被氾濫演算法清光，但原本的輪廓點 **Q** 陣列會被保留，以供後續的辨識程序使用。



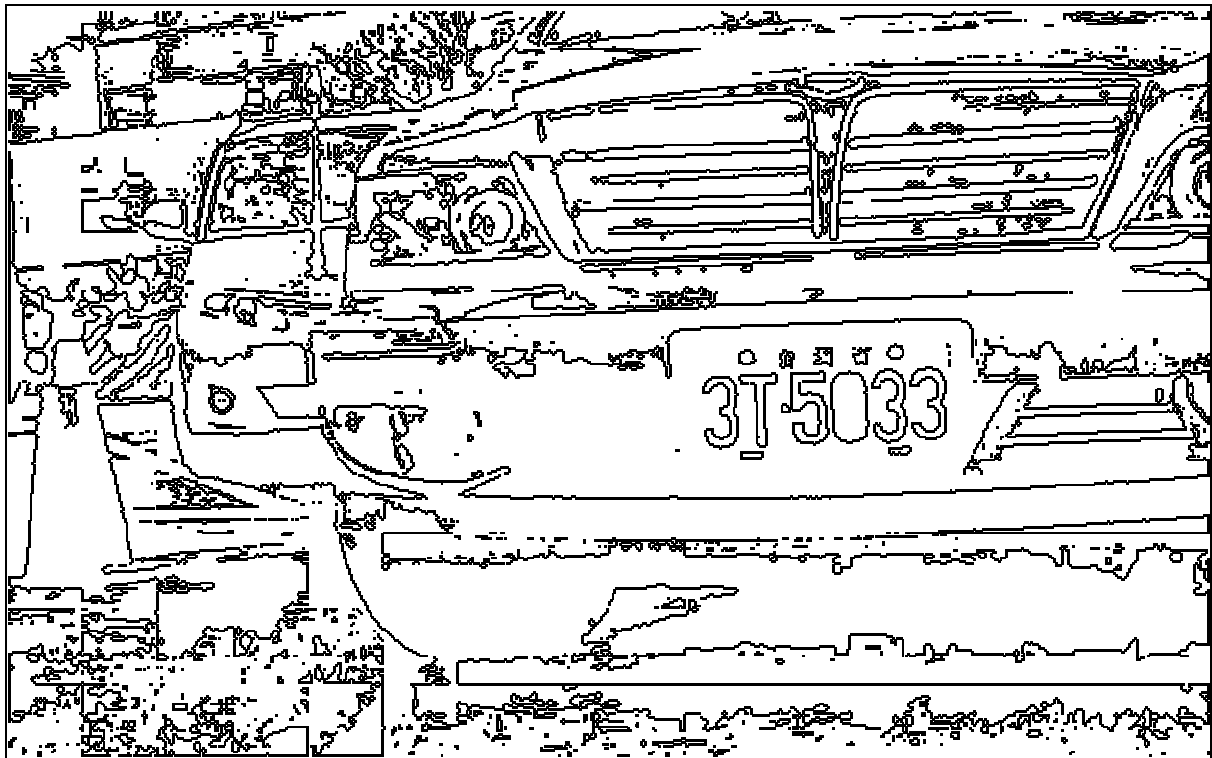
以輪廓點的封閉區線為建立目標的基礎會有一個問題，就是其實我們想要的目標是

實心的「黑色」區塊，但是「白色」的區塊一樣可以形成封閉曲線，看上圖你就明白了！如果要辨識那個 0 字，內圈白色目標是沒有用處的，應該要被剔除，但怎麼剔除呢？這裡可以用一個很巧妙的方法，就是：

每一個汨濫式演算法的起點應該是該目標最左邊的一點，此點本身是輪廓點，在二值化圖上當然是黑點，如果它的左邊是白點，表示此目標將會是個實心黑區，反之，就是白色目標了！所以只要一行程式 **If Z(i - 1, j) = 1 Then Continue For** 就可以排除白色目標了！接下來就是表單上 Targets 功能按鍵的程式碼了：

```
'建立目標物件
Private Sub TargetsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles TargetsToolStripMenuItem.Click
    C = getTargets(Q) '建立目標物件集合
    '繪製目標輪廓點
    Dim bmp As New Bitmap(nx, ny)
    For k As Integer = 0 To C.Count - 1
        Dim T As TgInfo = C(k)
        For m As Integer = 0 To T.PCount - 1
            Dim p As Point = T.P(m)
            bmp.SetPixel(p.X, p.Y, Color.Black)
        Next
    Next
    PictureBox1.Image = bmp '顯示目標輪廓
    Mb = bmp.Clone
End Sub
```

基本上就是用 `getTargets` 建立所有的目標集合 C，過程中已經排除了白色區塊的目標，接著就是畫出所有目標內的點集合。執行後的畫面會如下圖，可以看出將實心白色目標剔除後的輪廓線已經簡化很多了！



#### 4-4 依據寬高篩選目標物件

前一節我們已經將所有的黑色目標找出來了，但是我們的最終目標是車牌字元，超過預期的車牌字合理寬或高的目標可以依據目標屬性加以排除。這就是 **Filter** 按鍵的功能，程式碼如下：

'依據目標大小篩選目標

Dim minHeight As Integer = 10, maxHeight As Integer = 80 '有效目標高度範圍

Dim minWidth As Integer = 2, maxWidth As Integer = 80 '有效目標寬度範圍

Private Sub FilterToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) \_

Handles FilterToolStripMenuItem.Click

Dim D As New ArrayList

For k As Integer = 0 To C.Count - 1

Dim T As TgInfo = C(k)

If T.height < minHeight Then Continue For

If T.height > maxHeight Then Continue For

If T.width < minWidth Then Continue For

If T.width > maxWidth Then Continue For

D.Add(T) '有效目標

Next

C = D '回置目標集合

'繪製有效目標

Dim bmp As New Bitmap(nx, ny)

For k As Integer = 0 To C.Count - 1

Dim T As TgInfo = C(k)

For m As Integer = 0 To T.PCount - 1

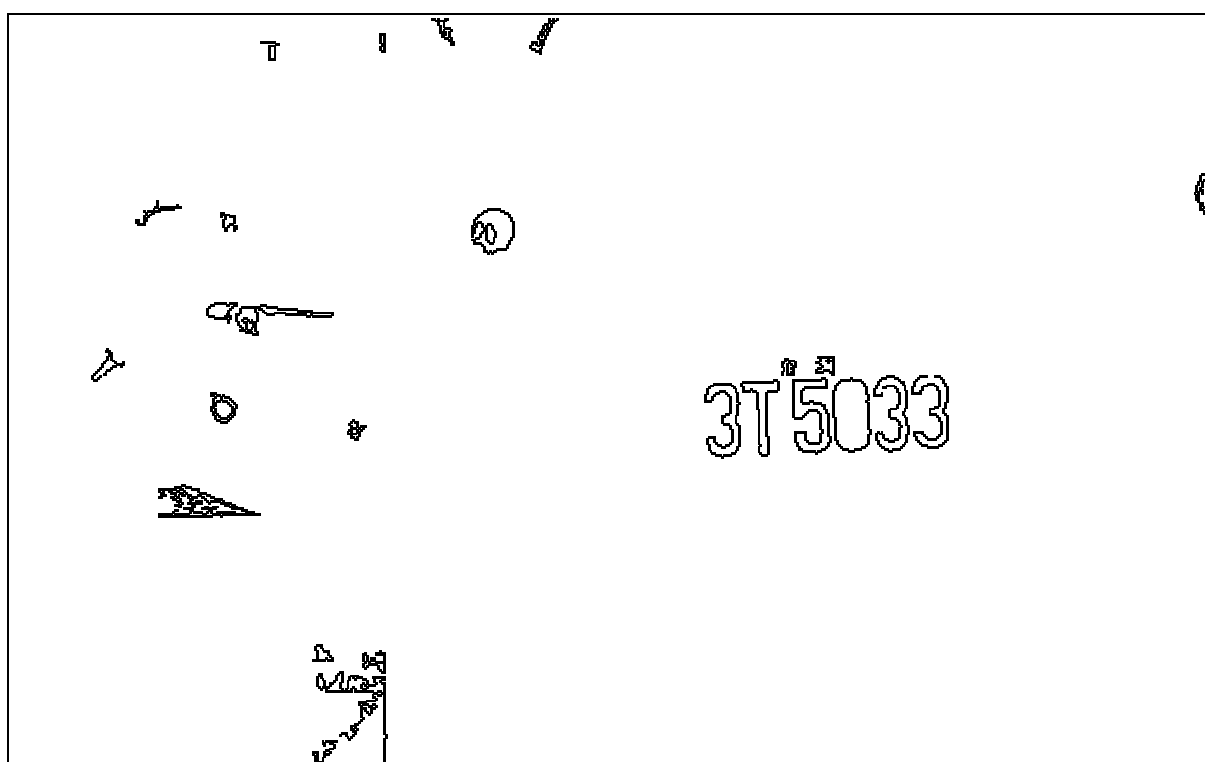


```

        Dim p As Point = T.P(m)
        bmp.SetPixel(p.X, p.Y, Color.Black)
    Next
Next
PictureBox1.Image = bmp '顯示目標輪廓
Mb = bmp.Clone
End Sub

```

在此我們是將寬度 2(畫素)以下，80 以上，高度 10 以下，80 以上的目標直接忽略，收集合格目標當作新的目標資料集合，為何寬度要設得特別小呢？是因為字元中會有「1」這種特別窄的字元。當然我們也可以用寬高比等其他更繁複的條件設計篩選條件，就看各為對問題的理解與需要自行斟酌了！執行 Filter 之後會變成這樣：



#### 4-5 依據目標與背景的亮度對比篩選

以多數的目標辨識來說，我們要找的常常是相對清楚，就是與背景亮度對比較高的目標，如果我們可以正確知道哪個目標較明顯？哪些目標對比度很低，只是背景雜訊！就可以大幅降低誤認的可能性，優先處理較清楚的目標也可以讓我們更快找到正確的車牌。下面是針對一個輪廓點計算它與周遭背景點亮度差距最大的對比度的 Function：

```

'輪廓點與背景的對比度
Private Function PointPm(ByVal p As Point) As Integer
    Dim x As Integer = p.X, y As Integer = p.Y
    Dim mx As Integer = B(x, y) '周邊最亮點，依據灰階陣列 B
    If mx < B(x - 1, y) Then mx = B(x - 1, y)
    If mx < B(x + 1, y) Then mx = B(x + 1, y)

```

```

If mx < B(x, y + 1) Then mx = B(x, y + 1)
If mx < B(x, y - 1) Then mx = B(x, y - 1)
Return mx - B(x, y) '最亮點與輪廓點的差值
End Function

```

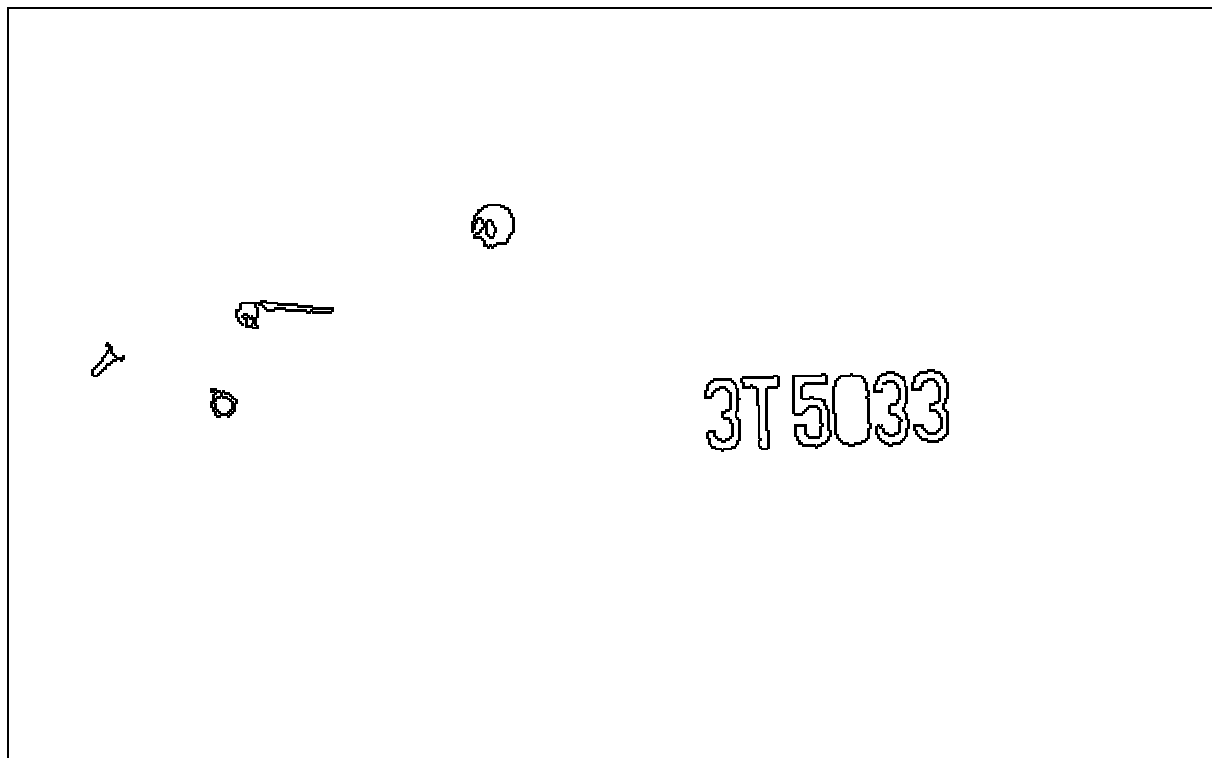
它的作用是送入一個應該是深色的輪廓點，查看它的上下左右四個鄰點，取最亮點與此輪廓點的灰階差異，就是此點的對比度了！我們只要將每個目標內蒐集的每個輪廓點，都做上述的計算，找出最大值作為整個目標的對比度(pm 屬性)即可！按鍵 **Sort** 的功能就是先建立好目前所有目標的對比度，然後用排序演算法找出前 10 名最強對比的目標，通常車牌就呼之欲出了！程式碼與執行結果如下：

```

'依據對比度排序前十大目標
Private Sub SortToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles SortToolStripMenuItem.Click
    '建立目標物件的對比度屬性
    For k As Integer = 0 To C.Count - 1
        Dim T As TgInfo = C(k)
        For m As Integer = 0 To T.PCount - 1
            Dim pm As Integer = PointPm(T.P(m))
            If pm > T.pm Then T.pm = pm '最高對比度的輪廓點
        Next
        C(k) = T '回置目標到原屬集合
    Next
    Next
    For i As Integer = 0 To 9
        For j As Integer = i + 1 To C.Count - 1
            Dim T As TgInfo = C(i), G As TgInfo = C(j)
            If T.pm < G.pm Then
                C(i) = G : C(j) = T '互換位置，高對比目標在前
            End If
        Next
    Next
    Next
    '繪製有效目標
    Dim bmp As New Bitmap(nx, ny)
    Dim D As New ArrayList
    For k As Integer = 0 To 9
        If k > C.Count - 1 Then Continue For
        Dim T As TgInfo = C(k)
        D.Add(T)
        For m As Integer = 0 To T.PCount - 1
            Dim p As Point = T.P(m)
            bmp.SetPixel(p.X, p.Y, Color.Black)
        Next
    Next
    Next
    C = D
    PictureBox1.Image = bmp '顯示目標輪廓
    Mb = bmp.Clone

```

End Sub



#### 4-6 檢視目標屬性

做到這裡，你已經知道圖上顯示的目標，每一個都有完整的屬性值了！但是你在程式介面上好像無法直接檢視？難道每次都要用除錯模式下的中斷點，到 Visual Studio 軟體去翻找眾多的變數集合嗎？這樣我們做實驗研究就很不方便了！所以我們設計了一個 PictureBox1 的 MouseDown 事件副程式如下：

'點選目標顯示位置與屬性

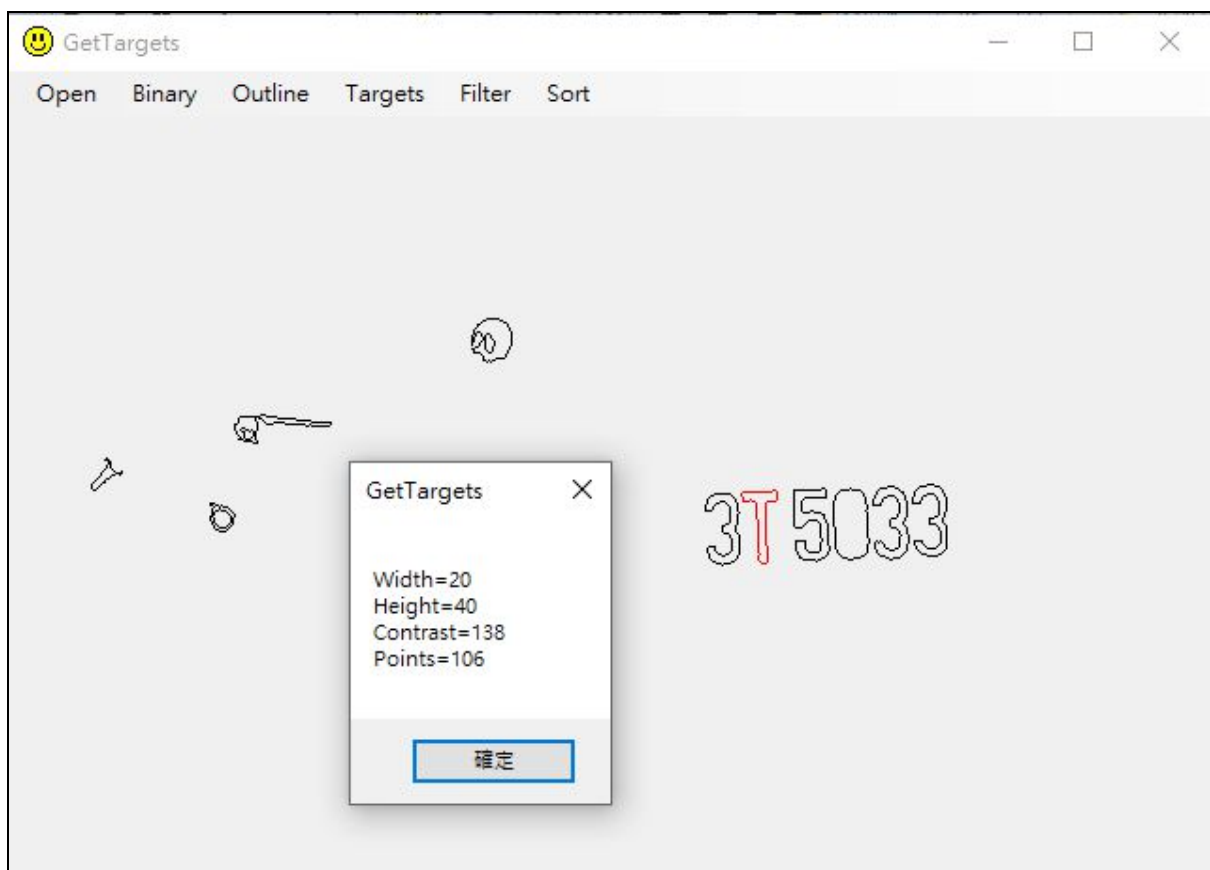
```
Private Sub PictureBox1_MouseDown(ByVal sender As Object, ByVal e As MouseEventArgs) _  
    Handles PictureBox1.MouseDown  
    If IsNothing(Mb) Then Exit Sub  
    If e.Button = MouseButtons.Left Then  
        Dim m As Integer = -1  
        For k As Integer = 0 To C.Count - 1  
            Dim T As TgInfo = C(k)  
            If e.X < T.xmn Then Continue For  
            If e.X > T.xmx Then Continue For  
            If e.Y < T.ymn Then Continue For  
            If e.Y > T.ymx Then Continue For  
            m = k '被點選目標  
        Exit For  
    Next  
    If m >= 0 Then '有被選目標時  
        Dim bmp As Bitmap = Mb.Clone
```

```

Dim T As TgInfo = C(m)
For k As Integer = 0 To T.P.Count - 1
    Dim p As Point = T.P(k)
    bmp.SetPixel(p.X, p.Y, Color.Red)
Next
PictureBox1.Image = bmp
'指定目標的資訊
Dim S As String = "Width=" + T.width.ToString
S += vbNewLine + "Height=" + T.height.ToString
S += vbNewLine + "Contrast=" + T.pm.ToString
S += vbNewLine + "Points=" + T.np.ToString
MsgBox(S)
End If
End If
End Sub

```

它的邏輯就是先看滑鼠點在哪一個目標的 X 與 Y 範圍之內？點到就算選取目標成功了！它會以目前所有目標的輪廓線為底圖，將選取目標的點畫成紅色，同時跳出一個訊息視窗顯示此目標的所有屬性。畫面大概如下：



#### 4-7 建立目標物件是必要的里程碑

請大家一定要記得本章的內容是學習 OCR 影像辨識的一個重要里程碑！很讓作者不解的一點是：多數傳統影像辨識書籍，一到這個階段就會忽然語焉不詳，好像作完正確切割影像目標，影像辨識的故事就結束了？但事實上整個辨識程序才進行到一半，如果不講清楚如何建立目標物件，就難以繼續將目標正規化到可以與字模進行比對，當然也無法到達辨識出目標文字意義的終點！

所以本書接下來的幾個章節幾乎就是被傳統影像辨識書籍所刻意忽略，**導致很少人會實作影像辨識程式的主因！**要完成任何辨識的工作，接下來以目標物件為基礎的處理過程是非常重要也必要的步驟。個人猜想，多數書籍到這邊就不再詳細舉例說明，應該是從這裡開始就會因為不同性質目的的辨識而有很大的程序設計差異，很難有一致性的完整通用程序。

這也是本書會以車牌辨識為主要範例的原因，唯有如此才能貫穿介紹整個影像辨識實作的完整過程！但也暗示著每一個不同目的的影像辨識**不會有完全通用的程序**，你必須充分掌握基本的技術動作因應目的調整，才可能設計出高辨識率的辨識核心，硬要套用所謂**通用**的演算法，即使可以勉強辨識成功，辨識率也不會太高的！

## 完整專案

Public Class Form1

Dim B(,) As Byte '灰階陣列

Dim Z(,) As Byte '全圖二值化陣列

Dim Q(,) As Byte '輪廓線陣列

Dim Gdim As Integer = 40 '計算區域亮度區塊的寬與高

Dim Th(,) As Integer '每一區塊的平均亮度・二值化門檻值

Dim C As ArrayList '目標物件集合

Dim Mb As Bitmap '顯示目標的影像

'目標物件結構

Public Structure TgInfo

Dim np As Integer '目標點數

Dim P As ArrayList '目標點的集合

Dim xmn As Short, xmx As Short, ymn As Short, ymx As Short '四面座標極值

Dim cx As Integer, cy As Integer '目標中心點座標

Dim width As Integer, height As Integer '寬與高

Dim pm As Integer '目標與背景的對比強度

Dim ID As Integer '目標依據對比度的排序

End Structure

'開啟檔案

Private Sub OpenToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) \_

Handles OpenToolStripMenuItem.Click

If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then

Dim bmp As New Bitmap(OpenFileDialog1.FileName)

Bmp2RGB(bmp) '擷取影像資訊

B = Gv '以綠光為灰階

PictureBox1.Image = bmp '顯示

End If

End Sub

'二值化

Private Sub BinaryToolStripMenuItem\_Click(ByVal sender As Object, ByVal e As EventArgs) \_

Handles BinaryToolStripMenuItem.Click

Th = ThresholdBuild(B) '建立二值化使用之門檻值陣列

ReDim Z(nx - 1, ny - 1) '建立二值化陣列

For i As Integer = 1 To nx - 2

Dim x As Integer = i \ Gdim 'x 座標換算

For j As Integer = 1 To ny - 2

Dim y As Integer = j \ Gdim 'y 座標換算

If B(i, j) < Th(x, y) Then

Z(i, j) = 1 '低於亮度門檻設為目標點

End If

Next

Next

Dim bmp As Bitmap = BWImg(Z) '建立二值化圖

PictureBox1.Image = bmp '顯示

End Sub

'門檻值陣列建立

Private Function ThresholdBuild(ByVal b(,) As Byte) As Integer(,)

Dim kx As Integer = nx \ Gdim, ky As Integer = ny \ Gdim

Dim T(kx, ky) As Integer

'累計各區塊亮度值總和

For i As Integer = 0 To nx - 1

Dim x As Integer = i \ Gdim

For j As Integer = 0 To ny - 1

Dim y As Integer = j \ Gdim

T(x, y) += b(i, j) '亮度值累加

```

        Next
    Next
    '區塊亮度平均值計算
    For i As Integer = 0 To kx - 1
        For j As Integer = 0 To ky - 1
            T(i, j) /= Gdim * Gdim
        Next
    Next
    Return T
End Function
'輪廓線
Private Sub OutlineToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles OutlineToolStripMenuItem.Click
    If IsNothing(Z) Then Exit Sub '無二值化圖忽略
    Q = Outline(Z) '建立輪廓點陣列
    Dim bmp As Bitmap = BWImg(Q) '建立輪廓圖
    PictureBox1.Image = bmp '顯示輪廓圖
End Sub
'建立輪廓點陣列
Private Function Outline(ByVal b(,) As Byte) As Byte(,)
    Dim Q(nx - 1, ny - 1) As Byte '輪廓點陣列
    For i As Integer = 1 To nx - 2
        For j As Integer = 1 + 1 To ny - 2
            If b(i, j) = 0 Then Continue For '非輪廓點忽略
            If b(i, j - 1) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
            If b(i - 1, j) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
            If b(i + 1, j) = 0 Then Q(i, j) = 1 : Continue For '確認為輪廓點
            If b(i, j + 1) = 0 Then Q(i, j) = 1 '確認為輪廓點
        Next
    Next
    Return Q
End Function
'建立目標物件
Private Sub TargetsToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles TargetsToolStripMenuItem.Click
    C = getTargets(Q) '建立目標物件集合
    '繪製目標輪廓點
    Dim bmp As New Bitmap(nx, ny)
    For k As Integer = 0 To C.Count - 1
        Dim T As TgInfo = C(k)
        For m As Integer = 0 To T.P.Count - 1
            Dim p As Point = T.P(m)
            bmp.SetPixel(p.X, p.Y, Color.Black)
        Next
    Next
    PictureBox1.Image = bmp '顯示目標輪廓
    Mb = bmp.Clone
End Sub
'以輪廓點建立目標陣列，排除負目標
Function getTargets(ByVal q(,) As Byte) As ArrayList
    Dim A As New ArrayList
    Dim b(,) As Byte = q.Clone '建立輪廓點陣列副本
    For i As Integer = 1 To nx - 2
        For j As Integer = 1 To ny - 2
            If b(i, j) = 0 Then Continue For
            Dim G As New TgInfo
            G.xmn = i : G.xmx = i : G.ymn = j : G.ymx = j : G.np = 1 : G.P = New ArrayList
            Dim nc As New ArrayList '每一輪搜尋的起點集合

```

```

nc.Add(New Point(i, j)) '輸入之搜尋起點
G.P.Add(New Point(i, j))
b(i, j) = 0 '清除此起點之輪廓點標記
Do
    Dim nb As ArrayList = nc.Clone '複製此輪之搜尋起點集合
    nc = New ArrayList '清除準備蒐集下一輪搜尋起點之集合
    For m As Integer = 0 To nb.Count - 1
        Dim p As Point = nb(m) '搜尋起點
        '在此點周邊 3X3 區域內找輪廓點
        For ii As Integer = p.X - 1 To p.X + 1
            For jj As Integer = p.Y - 1 To p.Y + 1
                If b(ii, jj) = 0 Then Continue For '非輪廓點忽略
                Dim k As New Point(ii, jj) '建立點物件
                nc.Add(k) '本輪搜尋新增的輪廓點
                G.P.Add(k)
                G.np += 1 '點數累計
                If ii < G.xmn Then G.xmn = ii
                If ii > G.xmx Then G.xmx = ii
                If jj < G.ymn Then G.ymn = jj
                If jj > G.ymx Then G.ymx = jj
                b(ii, jj) = 0 '清除輪廓點標記
            Next
        Next
    Next
    Loop While nc.Count > 0 '此輪搜尋有新發現輪廓點時繼續搜尋
    If Z(i - 1, j) = 1 Then Continue For '排除白色區塊的負目標 · 起點左邊是黑點
    G.width = G.xmx - G.xmn + 1 '寬度計算
    G.height = G.ymx - G.ymn + 1 '高度計算
    A.Add(G) '加入有效目標集合
Next
Next
Return A '回傳目標物件集合
End Function
'儲存目前影像
Private Sub SaveImageToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs)
-
    Handles SaveImageToolStripMenuItem.Click
    If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
        PictureBox1.Image.Save(SaveFileDialog1.FileName)
    End If
End Sub
'依據目標大小篩選目標
Dim minHeight As Integer = 10, maxHeight As Integer = 80 '有效目標高度範圍
Dim minWidth As Integer = 2, maxWidth As Integer = 80 '有效目標寬度範圍
Private Sub FilterToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles FilterToolStripMenuItem.Click
    Dim D As New ArrayList
    For k As Integer = 0 To C.Count - 1
        Dim T As TgInfo = C(k)
        If T.height < minHeight Then Continue For
        If T.height > maxHeight Then Continue For
        If T.width < minWidth Then Continue For
        If T.width > maxWidth Then Continue For
        D.Add(T) '有效目標
    Next
    C = D '回置目標集合
    '繪製有效目標
    Dim bmp As New Bitmap(nx, ny)

```



```

For k As Integer = 0 To C.Count - 1
    Dim T As TgInfo = C(k)
    For m As Integer = 0 To T.P.Count - 1
        Dim p As Point = T.P(m)
        bmp.SetPixel(p.X, p.Y, Color.Black)
    Next
Next
PictureBox1.Image = bmp '顯示目標輪廓
Mb = bmp.Clone
End Sub
'依據對比度排序前 X 大目標
Private Sub SortToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles ToolStripMenuItem.Click
    '建立目標物件的對比度屬性
    For k As Integer = 0 To C.Count - 1
        Dim T As TgInfo = C(k)
        For m As Integer = 0 To T.P.Count - 1
            Dim pm As Integer = PointPm(T.P(m))
            If pm > T.pm Then T.pm = pm '最高對比度的輪廓點
        Next
        C(k) = T '回置目標到原屬集合
    Next
    For i As Integer = 0 To 9
        For j As Integer = i + 1 To C.Count - 1
            Dim T As TgInfo = C(i), G As TgInfo = C(j)
            If T.pm < G.pm Then
                C(i) = G : C(j) = T '互換位置 · 高對比目標在前
            End If
        Next
    Next
    '繪製有效目標
    Dim bmp As New Bitmap(nx, ny)
    Dim D As New ArrayList
    For k As Integer = 0 To 9
        If k > C.Count - 1 Then Continue For
        Dim T As TgInfo = C(k)
        D.Add(T)
        For m As Integer = 0 To T.P.Count - 1
            Dim p As Point = T.P(m)
            bmp.SetPixel(p.X, p.Y, Color.Black)
        Next
    Next
    C = D
    PictureBox1.Image = bmp '顯示目標輪廓
    Mb = bmp.Clone
End Sub
'輪廓點與背景的對比度
Private Function PointPm(ByVal p As Point) As Integer
    Dim x As Integer = p.X, y As Integer = p.Y
    Dim mx As Integer = B(x, y) '周邊最亮點 · 依據灰階陣列 B
    If mx < B(x - 1, y) Then mx = B(x - 1, y)
    If mx < B(x + 1, y) Then mx = B(x + 1, y)
    If mx < B(x, y + 1) Then mx = B(x, y + 1)
    If mx < B(x, y - 1) Then mx = B(x, y - 1)
    Return mx - B(x, y) '最亮點與輪廓點的差值
End Function
'點選目標顯示位置與屬性
Private Sub PictureBox1_MouseDown(ByVal sender As Object, ByVal e As MouseEventArgs) _

```

```

Handles PictureBox1.MouseDown
If IsNothing(Mb) Then Exit Sub
If e.Button = MouseButtons.Left Then
    Dim m As Integer = -1
    For k As Integer = 0 To C.Count - 1
        Dim T As TgInfo = C(k)
        If e.X < T.xmn Then Continue For
        If e.X > T.xmx Then Continue For
        If e.Y < T.ymn Then Continue For
        If e.Y > T.ymx Then Continue For
        m = k '被點選目標
    Exit For
Next
If m >= 0 Then '有被選目標時
    Dim bmp As Bitmap = Mb.Clone
    Dim T As TgInfo = C(m)
    For k As Integer = 0 To T.P.Count - 1
        Dim p As Point = T.P(k)
        bmp.SetPixel(p.X, p.Y, Color.Red)
    Next
    PictureBox1.Image = bmp
    '指定目標的資訊
    Dim S As String = "Width=" + T.width.ToString
    S += vbNewLine + "Height=" + T.height.ToString
    S += vbNewLine + "Contrast=" + T.pm.ToString
    S += vbNewLine + "Points=" + T.np.ToString
    MsgBox(S)
End If
End If
End Sub
End Class

```