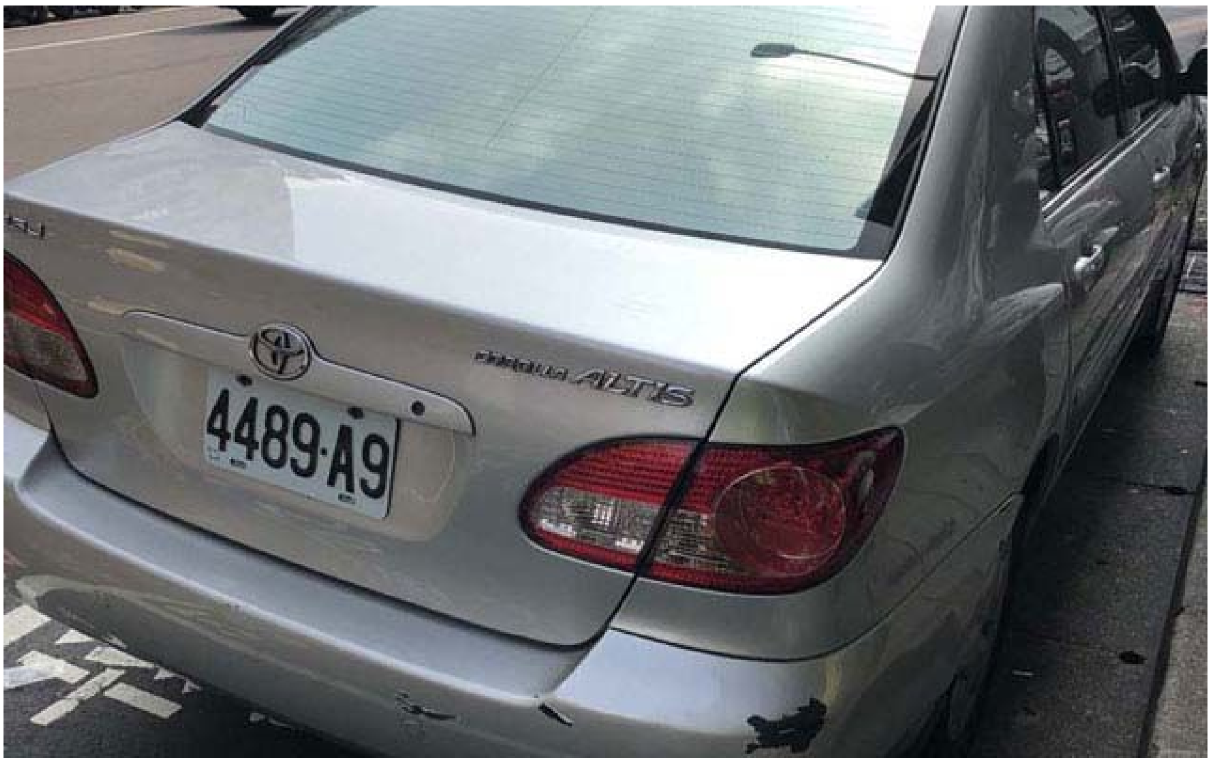
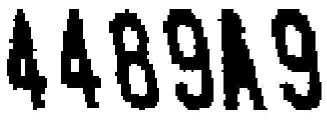
第 10 章 立體空間斜視影像的處理

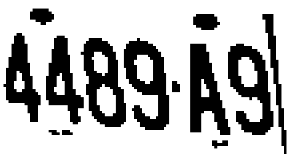


10-1 光是旋轉與縮放還是不夠的

前面九章似乎已經把車牌辨識會遇到的大部分問題都解決了，傳統車牌辨識系統最難作到的高角度水平傾斜我們都可以處理了！但這樣還是不夠的，上圖是第五章用的影像，當時我們只是介紹把車牌字元以群組方式找出來，但是如果你用第七章的程式將他們轉成車牌的虛擬影像會變成這樣，字源還是歪斜的！

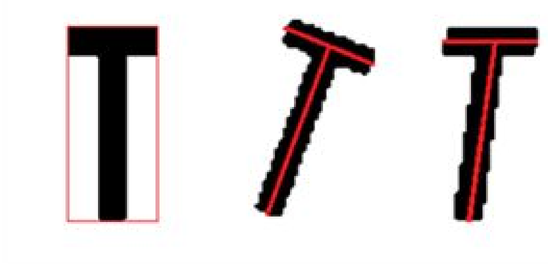


如果你懷疑是我的程式計算有誤，使用 PhotoShop 軟體作出來的結果也是完全一樣的！我們可以先將車牌字元盡量轉成水平，再作二值化處理將車牌字元切割出來，那些字元真的都還是明顯向左邊傾倒的！



可以預期拿這樣的影像去比對字模是一定不會成功的！為何如此？因為車牌影像是在立體(三維)世界拍攝的，我們前面的字元目標影像處理包括了：平移、旋轉與縮放，

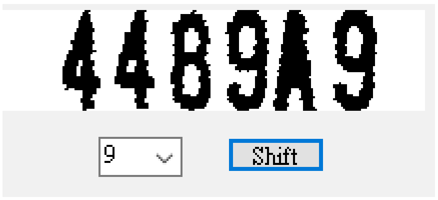
其中只有不限長寬比的縮放方式可以處理部分側向斜視的變形，但是如果側視與俯視同時發生，原來矩形的字元就不再保證是個矩形了！請看下面的示意圖：



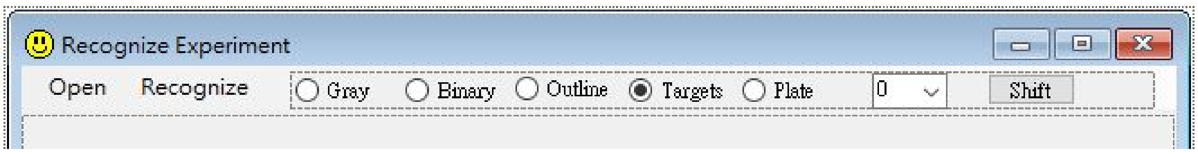
上圖左邊是標準字模，中間是單純的平面旋轉，右邊就是在三維空間斜視可能看到的變形 T 了！字元的橫線與直線不再保證垂直了！不管你怎麼旋轉都無法與字模對齊。這種情況蠻多的，多數近距離用手機拍的照片都有這種問題，近年出現的路邊停車柱的車牌辨識也有這個問題，當你的車牌辨識碰到這種看起來很正面清晰的車牌都無法辨識時，就很難對客戶解釋了！

如果我們知道變形的字元就是 T 字，其實不難處理，讓影像上下平移錯位，上面拉左邊下面拉右邊，拉到中軸線變成垂直的就好了！但是英數字的字元那麼多，我們也不會在沒比對字模之前就知道它是甚麼字？如果知道就根本不必處理影像了嘛！所以我們必須從前面的虛擬車牌影像著手！

我們必須使用全圖的特徵統計，找到一個最佳的上下平移錯位值，將整個虛擬車牌的字元一起轉正！正確的結果應該是像下圖一樣，怎麼作呢？本章會慢慢說給你聽！



10-2 上下平移錯位的實驗



首先請複製上一章的程式專案，新增一個 Plate 單選按鈕，加上一個有 1-13 選項的 ComboBox1，以及一個 Button1。先到公用變數區加入這一行宣告：

```
Dim V(,) As Byte '虛擬車牌二值化陣列
```

Plate 按鈕是用來繪製虛擬車牌的程式：

```
'車牌
```

```
Private Sub RadioButton5_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RadioButton5.CheckedChanged
    If RadioButton5.Checked Then
        If IsNothing(V) Then Exit Sub
        Dim bmp As Bitmap = BWImg(V)
        PictureBox1.Image = bmp
    End If
End Sub
```

我們先將 Recognize 主程式簡化為只做一次辨識的簡單狀態：

```
Private Sub RecognizeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles RecognizeToolStripMenuItem.Click
    Z = DoBinary(B) '二值化
    Q = Outline(Z) '建立輪廓點陣列
    CA = getTargets(Q) '建立所有目標物件集合
    ReDim skp(CA.Count - 1)
    Dim R As LPInfo = getLP(CA)
    Me.Text = R.A + "," + R.Sc.ToString + "," + R.cx.ToString + "," + R.cy.ToString
End Sub
```

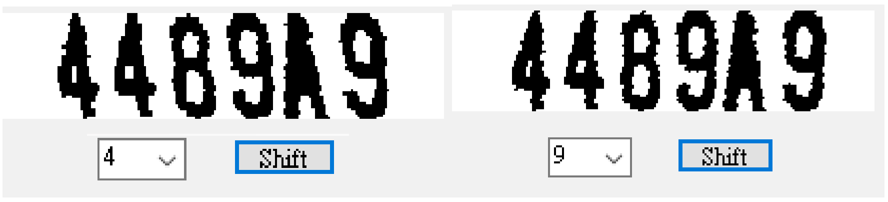
再到 getLP 副程式末段，加入以下粗體字部分的程式(其他程式碼是既有程式)：

```
    '計算最大字元間距與位置
    Dim dmx As Integer = 0, mi As Integer = 0
    For i As Integer = 0 To n - 2
        Dim d As Integer = (C(i + 1).cx - C(i).cx) ^ 2 + (C(i + 1).cy - C(i).cy) ^ 2
        If d > dmx Then dmx = d : mi = i
    Next
    '車牌全圖矩陣，字元間隔 2 畫素
    Dim wd As Integer = Pw + (Pw + 2) * n + Pw
    ReDim V(wd, Ph - 1)
    For k As Integer = 0 To n - 1
        Dim xs As Integer = Pw + (Pw + 2) * k 'X 偏移量
        For i As Integer = 0 To Pw - 1
            For j As Integer = 0 To Ph - 1
                V(xs + i, j) = MC(k)(i, j)
            Next
        Next
    Next
    For i As Integer = 0 To MC.Count - 1
        Dim D As ChInfo = BestC(MC(i))
        R.A += D.Ch '車牌字串累加
        R.Sc += D.ft '字源符合度累加
        If i = mi Then R.A += "-"c
    Next
```

新增的程式是用已經作過平面旋轉的目標群繪製一張虛擬的車牌，與第七章的相關程式功能一樣，接下來就是用 Button1(Shift 按鍵)執行上下平移錯位的程式了，上下端錯位的平移量由 ComboBox1 的內容決定：

```
'虛擬車牌上下端平移錯位實驗
Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) _
    Handles Button1.Click
    Dim n As Integer = C.Count '目標個數
    Dim wd As Integer = Pw + (Pw + 2) * n + Pw '虛擬車牌寬度
    Dim S(wd - 1, Ph - 1) As Byte '虛擬車牌陣列
    Dim f As Double = Val(ComboBox1.Text) / (Ph - 1) '每一畫素高度的 X 錯位量
    For j As Integer = 0 To Ph - 1
        Dim dx As Integer = f * (Ph - 1 - j) 'X 移動量
        For i As Integer = 0 To wd - 1
            Dim x As Integer = i + dx
            If x > wd - 1 Or x < 0 Then Continue For
            S(x, j) = V(i, j)
        Next
    Next
    '計算錯位後的虛擬車牌中有幾個垂直空白線？越多表示字元越正直
    Dim n0 As Integer = 0
    For i As Integer = 0 To wd - 1
        Dim m As Integer = 0
        For j As Integer = 0 To Ph - 1
            m += S(i, j)
        Next
        If m = 0 Then n0 += 1
    Next
    Dim bmp As Bitmap = BWImg(S)
    PictureBox1.Image = bmp
    MsgBox(n0)
End Sub
```

執行程式讀入本章首頁的圖檔，按下 Recognize 按鍵，再按 Plate 出現虛擬車牌之後即可開始作實驗，改變 ComboBox1 的內容，看看不同的平移量扭正車牌的效果，每次執行後跳出的訊息框會顯示虛擬車牌中有幾個 X 值的畫素是全白的，白色直線數量越多表示整個字元組的字越端正，**字元間的空隙越明顯**。



從 1 到 13 點錯位的白色直線數大致是這樣的：72，76，81，81，81，85，84，85，**87**，83，82，82，80。最高值是在 9，最佳偏移量就是 9 了！相對的，如果車牌本來就

拍得很正面，沒有太多這種變形呢？你可以拿前面幾章的車牌影像作實驗，最佳的偏移量都會很接近 0！

10-3 將平移錯位變形處理納入辨識程序

上一節的程式當然只是「教學用」的！如果你希望自己的車牌辨識可以自動處理好這種變形，當然就必須將以上運算變成辨識流程的一部份，看起來程式碼還蠻多的，就先寫成一個副程式吧！

```
'左右傾倒校正
Private Sub IncCorrect(ByVal V(,) As Byte)
    Dim n As Integer = C.Count '目標個數
    Dim wd As Integer = Pw + (Pw + 2) * n + Pw '虛擬車牌二值化陣列寬度
    Dim mx As Integer = 0, mk As Integer = 0 '空白垂直線最大數，最佳偏移量
    Dim S0(wd - 1, Ph - 1) As Byte '最佳校正之車牌二值化陣列
    For sx As Integer = -15 To 15 '嘗試偏移量
        Dim S(wd, Ph - 1) As Byte '虛擬車牌陣列
        Dim f As Double = sx / (Ph - 1) '每一垂直畫素偏移比例
        For j As Integer = 0 To Ph - 1
            Dim dx As Integer = f * (Ph - 1 - j)
            For i As Integer = 15 To wd - 16
                Dim x As Integer = i + dx
                S(x, j) = V(i, j)
            Next
        Next
        Dim n0 As Integer = 0
        For i As Integer = 0 To wd - 1
            Dim m As Integer = 0
            For j As Integer = 0 To Ph - 1
                m += S(i, j)
            Next
            If m = 0 Then n0 += 1
        Next
        If n0 > mx Then
            mx = n0 : mk = sx : S0 = S
        End If
    Next
    If mk <> 0 Then '需要調整傾倒字元，將修正後之車牌二值化影像重作目標擷取
        ReDim Z(nx - 1, ny - 1)
        For i As Integer = 0 To wd
            For j As Integer = 0 To Ph - 1
                Z(i + 10, j + 10) = S0(i, j)
            Next
        Next
        Q = Outline(Z) '建立輪廓點陣列
        CA = getTargets(Q) '建立所有目標物件集合
```

```
            ReDim skp(CA.Count - 1) '重設已處理標記陣列
            C = AlignTgs(CA) '群組化車牌目標
            n = C.Count '目標個數
            Dim T(n - 1) As TgInfo, M(n - 1) As Array, w(n - 1) As Integer, h(n - 1) As Integer
            For k As Integer = 0 To n - 1
                Dim G As TgInfo = C(k)
                M(k) = Tg2Bin(G) '建立單一目標的二值化矩陣
                T(k) = G '儲存旋轉後的目標物件
                w(k) = G.width '寬度陣列
                h(k) = G.height '高度陣列
            Next
            Array.Sort(w) '寬度排序小到大
            Array.Sort(h) '高度排序小到大
            Dim mw As Integer = w(n - 2) '取第二寬的目標為標準，避開意外沾連的極端目標
            Dim mh As Integer = h(n - 2) '取第二高的目標為標準，避開意外沾連的極端目標
            For k As Integer = 0 To n - 1
                MC(k) = NmBin(T(k), M(k), mw, mh) '個別字元正規化矩陣
            Next
        End If
    End Sub
```

這個副程式的前半部其實就是上一節的實驗，偏移量設定從-10 到+10 畫素，嘗試找到最佳的變形扭曲平移量 mk，如果 mk 就是 0，表示沒有變形，就不需要作任何處理。如果 mk<>0，當然就是將原始虛擬車牌的陣列用最佳變形處理後的 S0 陣列取代了！此時有點尷尬，我們的個別字元目標其實都被改變了！

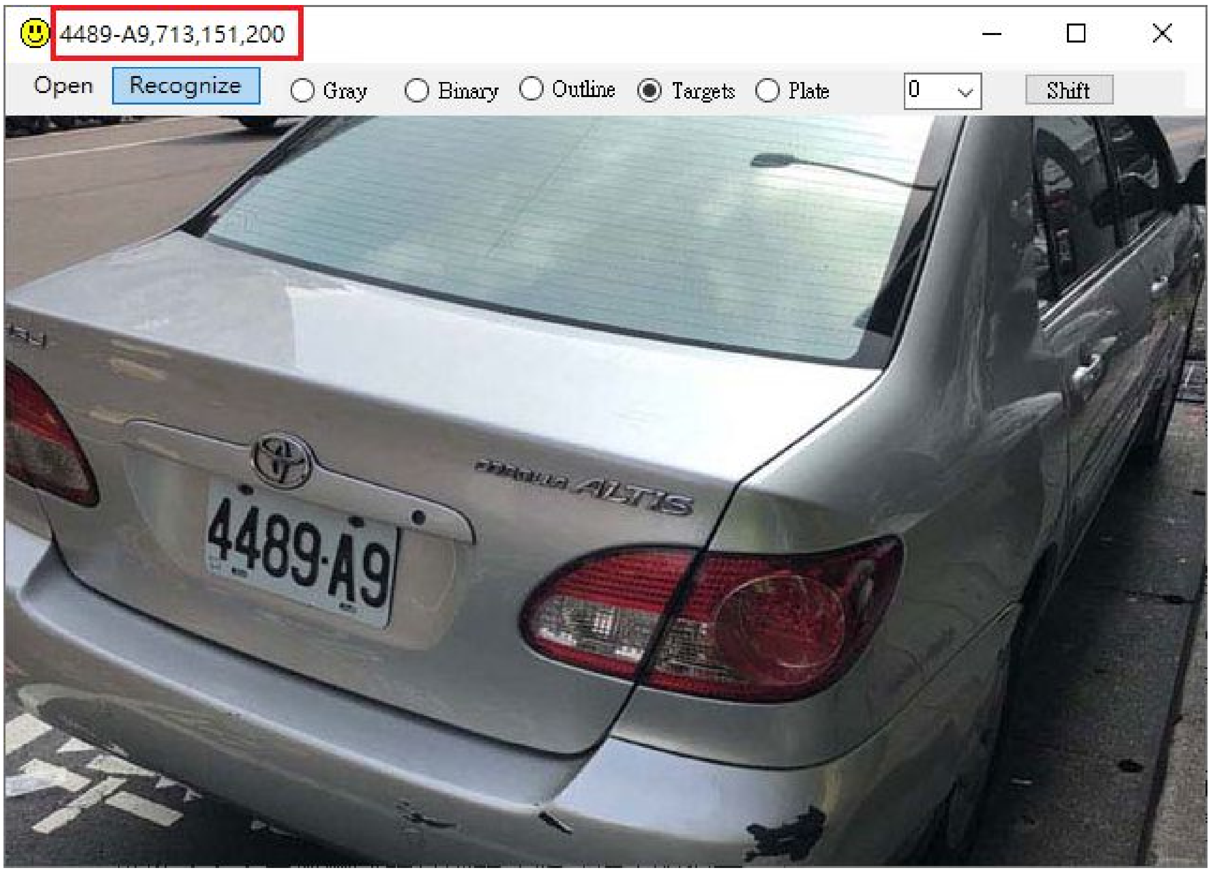
所以要接續作字模比對的話，目標必須重新設定。比較簡單的程序是將這個虛擬車牌當作是原圖的二值化陣列，再作一次完整的目標擷取程序，好像全圖只有這幾個已經轉正的二值化目標一樣！看起來會有很多辨識程序都重作了，但是因為實際的資訊量少了，也簡化了，所以並不會耗時太久的！

此副程式其實就是檢驗是否有這種變形？如果有！就重作一些處理程序，產出新的經過變形轉正處理的 **MC** 陣列，所以只要插入原來的 getLP 主辨識程序中適當的位置就可以了！下面黑體字就是副程式應該插入的位置：

```
        '車牌虛擬矩陣，字元間隔 2 畫素
        Dim wd As Integer = Pw + (Pw + 2) * n + Pw
        ReDim V(wd, Ph - 1)
        For k As Integer = 0 To n - 1
            Dim xs As Integer = Pw + (Pw + 2) * k 'X 偏移量
            For i As Integer = 0 To Pw - 1
                For j As Integer = 0 To Ph - 1
                    V(xs + i, j) = MC(k)(i, j)
                Next
            Next
```

```
Next
IncCorrect(V) '字元傾倒偵測校正
For i As Integer = 0 To MC.Count - 1
    Dim D As ChInfo = BestC(MC(i))
    R.A += D.Ch '車牌字串累加
    R.Sc += D.ft '字源符合度累加
    If i = mi Then R.A += "-"c
Next
```

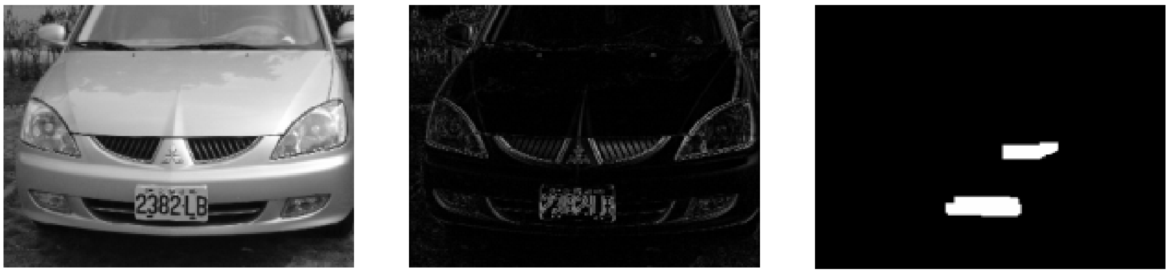
此時再嘗試辨識本章使用的影像範例，結果就會變成這樣，可以正確辨識了！


4489-A9,713,151,200

Targets [✓]

10-4 這一章教我們的事情

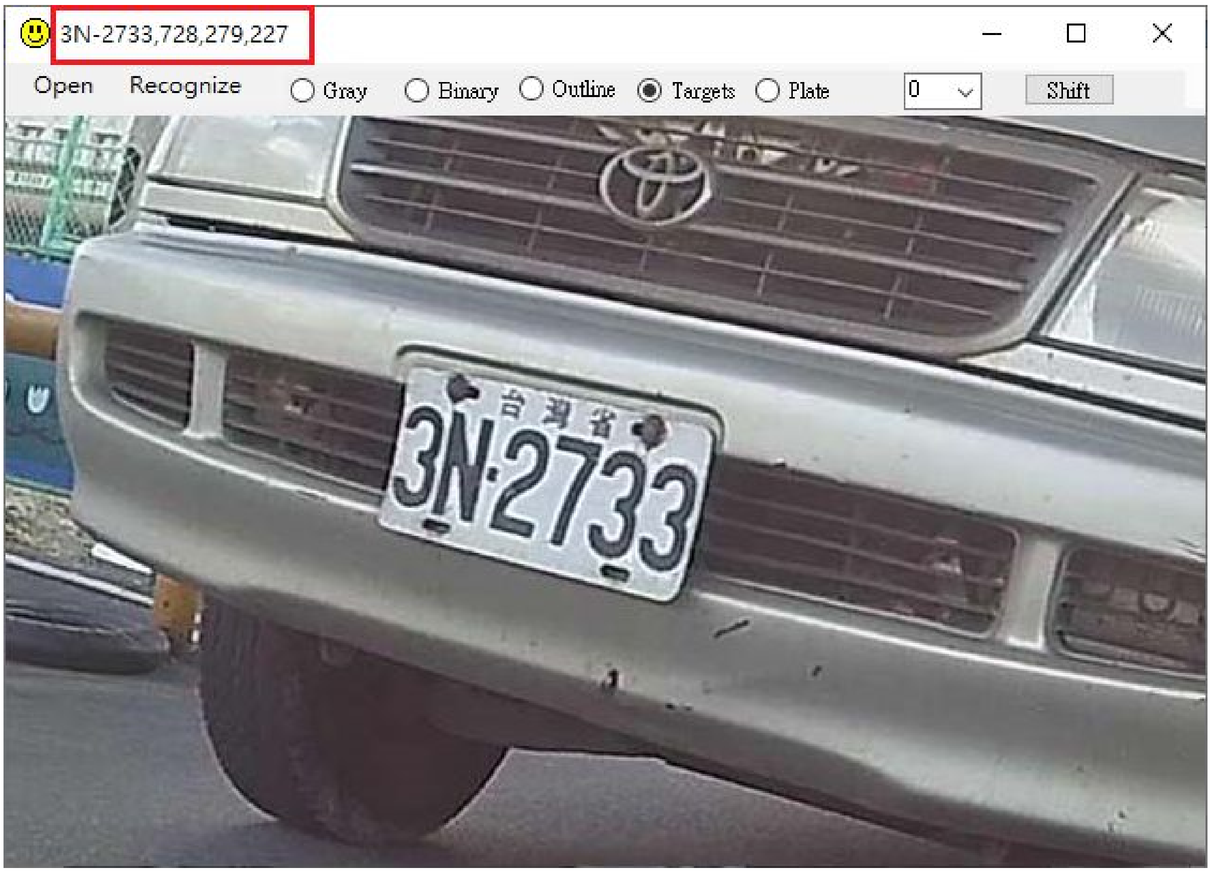
這一章的處理內容其實意義重大！傳統上多數已公開的車牌辨識演算法，都是以車牌目標是一個二維空間的 OCR 目標為假設，在全圖中它應該是一個矩形，像下面的這張車牌搜尋示意圖：



摘自：http://portal.ptivs.ptc.edu.tw/sys/lib/read_attach.php?id=1039

在這種假設之下，車牌當然必須拍得很正面不能有太大的傾斜！通常傾斜 10 度就會讓這種演算法失敗找不到車牌的位置，所以雖然本書算是入門等級的影像辨識書，嚴格講也不是本公司車牌辨識產品演算法的完整介紹，但是已經具有超越多數現有市售車牌辨識產品演算法的學術價值了！

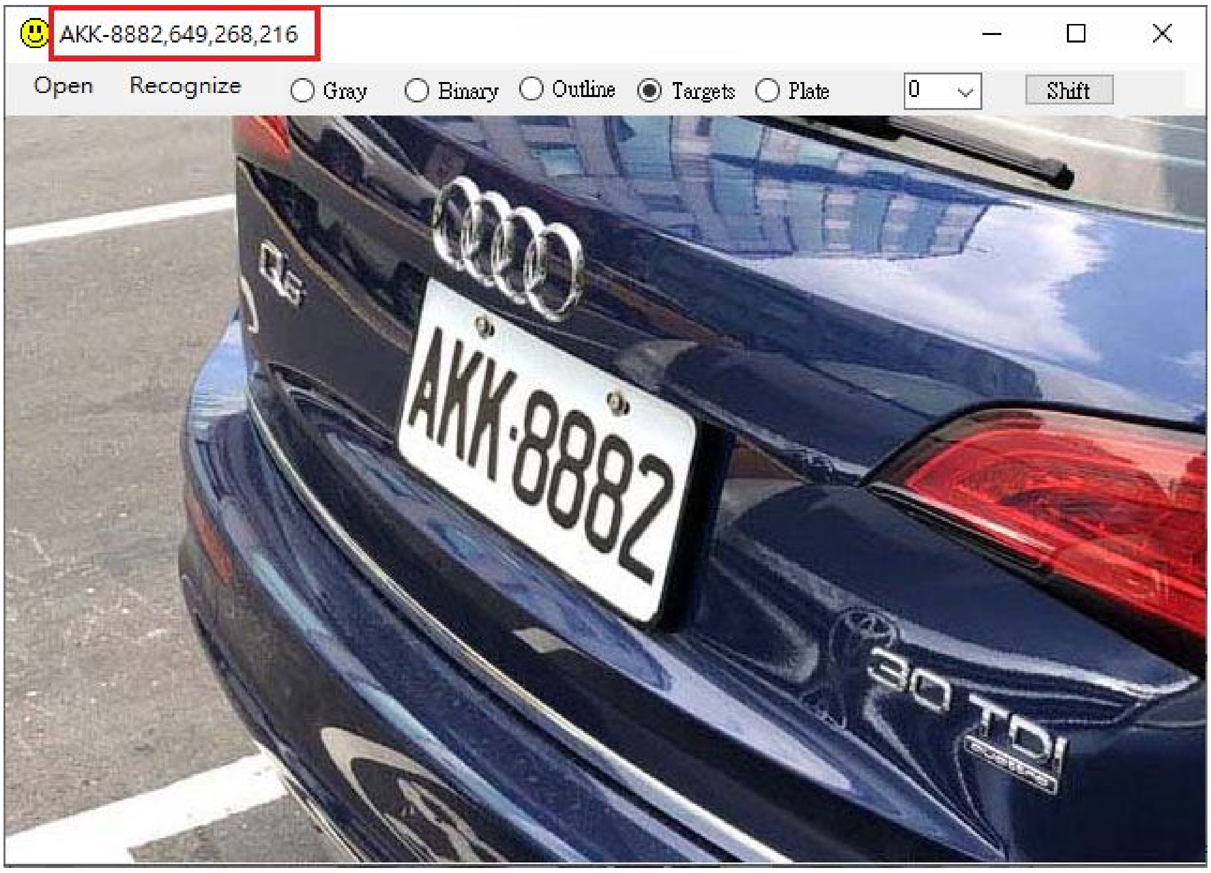
但是真正的立體辨識並非只是克服車牌的旋轉角度而已，譬如下圖影像就不需要本章介紹的演算程序，只用前面一章的程式也可以辨識成功！因為字元本身基本上還是符合矩形假設的，利用車牌字元的排列找到轉角度轉正目標即可。



3N-2733,728,279,227

Open Recognize ○ Gray ○ Binary ○ Outline ● Targets ○ Plate 0 Shift

但是如果像下圖的情況，因為斜視的關係，連字元目標本身也不再能假設為一個矩形，而是個平行四邊形時，本章的演算法就變成必要的辨識程序了！


AKK-8882,649,268,216

至此，我們的車牌辨識教學告一段落，但是要作出真正很通用高辨識率的軟體，你還需要非常多的細節處理能力。但是本書至此已經介紹了很重要的，在立體空間產生變形時的處理方式！傳統的 OCR 應該會教到如何將「平面」上經過掃描產生的影像作辨識處理，但是現在很多需要辨識的影像是來自手機或攝影機，因此會有斜視或目標本身距離取向點不一致的變形，這應該是影像辨識的一個重要新課題。

當然如果你是期待用機器學習技術進行辨識的人，可能就會失望了！因為這種空間變形的可能性是連續性，也幾乎沒有任何形狀限制的！一個矩形目標可以變成任意形狀的四邊形，這該如何經過機率統計的方式去「學習」呢？但無論如何，在立體空間造成的視覺變形必然是新一代的影像處理技術必須正視的問題。

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 minHeight As Integer = 20, 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
    Dim Tgmax As Integer = 20 '進入決選範圍的最明顯目標上限
    Dim C As ArrayList, CA As ArrayList '目標物件集合
    Dim Pw As Integer = 25, Ph As Integer = 50 '字模的寬與高
    Dim P(1, 35, Pw - 1, Ph - 1) As Byte '六七碼車牌所有英數字二值化陣列
    Dim P69(1, Pw - 1, Ph - 1) As Byte '變形的 6 與 9
    Dim MC() As Array '正規化完成後的字元二值化陣列
    Dim skp() As Boolean '已檢視目標註記
    Dim Inc As Double '車牌傾斜角度(>0 為順時針傾斜)
    Dim V(,) As Byte '虛擬車牌二值化陣列
    '字元對照表
    '0-9→0-9
    '10→A，11→B，12→C，13→D，14→E，15→F，16→G，17→H，18→I，19→J
    '20→K，21→L，22→M，23→N，24→O，25→P，26→Q，27→R，28→S，29→T
    '30→U，31→V，32→W，33→X，34→Y，35→Z
    Dim Ch() As Char = {"0"c, "1"c, "2"c, "3"c, "4"c, "5"c, "6"c, "7"c, "8"c, "9"c,
                        "A"c, "B"c, "C"c, "D"c, "E"c, "F"c, "G"c, "H"c, "I"c, "J"c,
                        "K"c, "L"c, "M"c, "N"c, "O"c, "P"c, "Q"c, "R"c, "S"c, "T"c,
                        "U"c, "V"c, "W"c, "X"c, "Y"c, "Z"c}
    '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
    '字元結構
    Public Structure ChInfo
        Dim Ch As Char '最符合字元
        Dim ft As Integer '符合度評分
        Dim kind As Integer '六或七碼字元，0→六碼，1→七碼
    End Structure
    '車牌資料結構
    Public Structure LPInfo
        Dim A As String '車牌號碼
        Dim Sc As Integer '符合度
        Dim N As Integer '車牌字元目標個數
        Dim kind As Integer '六或七碼字型
        Dim xmn As Integer, ymn As Integer, xmx As Integer, ymx As Integer '車牌四邊極值
        Dim width As Integer, height As Integer '車牌寬與高
        Dim cx As Integer, cy As Integer '車牌中心點座標
    End Structure
    '啟動程式載入字模
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        FontLoad()
    End Sub
    Private Sub FontLoad()
```

```
        Dim q() As Byte = My.Resources.font '使用字模資源檔
        Dim n As Integer = 0
        '匯入六與七碼字模
        For m As Integer = 0 To 1
            For k As Integer = 0 To 35
                For j As Integer = 0 To Ph - 1
                    For i As Integer = 0 To Pw - 1
                        P(m, k, i, j) = q(n) : n += 1
                    Next
                Next
            Next
        Next
        '匯入六碼變形 69 字模
        For k As Integer = 0 To 1
            For j As Integer = 0 To Ph - 1
                For i As Integer = 0 To Pw - 1
                    P69(k, i, j) = q(n) : n += 1
                Next
            Next
        Next
    End Sub
    '開啟檔案
    Private Sub OpenToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.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.Clone '以綠光為灰階
            PictureBox1.Image = bmp '顯示
        End If
    End Sub
    '辨識整個車牌
    Private Sub RecognizeToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RecognizeToolStripMenuItem.Click
        Z = DoBinary(B) '二值化
        Q = Outline(Z) '建立輪廓點陣列
        CA = getTargets(Q) '建立所有目標物件集合
        ReDim skp(CA.Count - 1)
        Dim R As LPInfo = getLP(CA)
        Me.Text = R.A + "," + R.Sc.ToString + "," + R.cx.ToString + "," + R.cy.ToString
    End Sub
    '負片
    Private Function Negative(ByVal b(,) As Byte) As Byte(,)
        For i As Integer = 0 To nx - 1
            For j As Integer = 0 To ny - 1
                b(i, j) = 255 - b(i, j)
            Next
        Next
        Return b
    End Function
    '依據可能目標組合辨識車牌
    Private Function getLP(ByVal A As ArrayList) As LPInfo
        Dim R As New LPInfo '建立車牌資訊物件
        C = AlignTgs(A) '找到最多七個的字元目標組合
        Dim n As Integer = C.Count '目標個數
        If n < 4 Then Return R '目標數目不足以構成車牌
        R.N = n '車牌目標個數
        For i As Integer = 0 To n - 1
    skp(C(i).ID) = True '標示目標已處理
Next
'末字中心點與首字中心點的偏移量，斜率計算參數
Dim dx As Integer = C(n - 1).cx - C(0).cx
Dim dy As Integer = C(n - 1).cy - C(0).cy
Inc = Math.Atan2(dy, dx) '字元排列傾角
'旋轉所有目標
Dim T(n - 1) As TgInfo, M(n - 1) As Array, w(n - 1) As Integer, h(n - 1) As Integer
R.xmn = nx : R.xmx = 0 : R.ymn = ny : R.ymx = 0 '車牌四面極值
For k As Integer = 0 To n - 1
    Dim G As TgInfo = C(k)
    M(k) = Tg2Bin(G) '建立單一目標的二值化矩陣
    M(k) = RotateTg(M(k), G, Inc) '旋轉目標，G 為 ByRef
    T(k) = G '儲存旋轉後的目標物件
    w(k) = G.width '寬度陣列
    h(k) = G.height '高度陣列
    If G.xmn < R.xmn Then R.xmn = G.xmn
    If G.xmx > R.xmx Then R.xmx = G.xmx
    If G.ymn < R.ymn Then R.ymn = G.ymn
    If G.ymx > R.ymx Then R.ymx = G.ymx
Next
R.width = R.xmx - R.xmn + 1 : R.height = R.ymx - R.ymn + 1 '車牌寬高
R.cx = (R.xmn + R.xmx) / 2 : R.cy = (R.ymn + R.ymx) / 2 '車牌中心點
Array.Sort(w) '寬度排序小到大
Array.Sort(h) '高度排序小到大
Dim mw As Integer = w(n - 2) '取第二寬的目標為標準，避開意外沾連的極端目標
Dim mh As Integer = h(n - 2) '取第二高的目標為標準，避開意外沾連的極端目標
'目標正規化→寬高符合字模
ReDim MC(n - 1) '正規化後之字元二值化陣列
For k As Integer = 0 To n - 1
    MC(k) = NmBin(T(k), M(k), mw, mh) '個別字元正規化矩陣
Next
'計算最大字元間距與位置
Dim dmx As Integer = 0, mi As Integer = 0
For i As Integer = 0 To n - 2
    Dim d As Integer = (C(i + 1).cx - C(i).cx) ^ 2 + (C(i + 1).cy - C(i).cy) ^ 2
    If d > dmx Then dmx = d : mi = i
Next
'車牌虛擬矩陣，字元間隔 2 畫素
Dim wd As Integer = Pw + (Pw + 2) * n + Pw
ReDim V(wd, Ph - 1)
For k As Integer = 0 To n - 1
    Dim xs As Integer = Pw + (Pw + 2) * k 'X 偏移量
    For i As Integer = 0 To Pw - 1
        For j As Integer = 0 To Ph - 1
            V(xs + i, j) = MC(k)(i, j)
        Next
    Next
Next
IncCorrect(V) '字元傾倒偵測校正
For i As Integer = 0 To MC.Count - 1
    Dim D As ChInfo = BestC(MC(i))
    R.A += D.Ch '車牌字串累加
    R.Sc += D.ft '字源符合度累加
    If i = mi Then R.A += "-"c
Next
R.Sc /= MC.Count
R = ChkLP(R) '檢查是否為合格車牌
```

```
        R = ChkED(R) '修正英數字
        Return R '回傳車牌資料
End Function
'檢驗車牌是否正確的程式
Private Function ChkLP(ByVal R As LPInfo) As LPInfo
        If R.Sc < 600 Then Return New LPInfo '符合度低於及格分數
        If R.A.Length < 5 Then Return New LPInfo '包含分隔線在內字數小於 5
        Dim m As Integer = R.A.IndexOf("-") '格線位置
        If m = 1 Then Return New LPInfo '沒有 1-x  的字數區段格式
        Return R '合格車牌
End Function
'找車牌字元目標群組
Private Function AlignTgs(ByVal C As ArrayList) As ArrayList
        Dim R As New ArrayList, pmx As Integer = 0 '最佳目標組合與最佳度比度
        For i As Integer = 0 To C.Count - 1
                If skp(i) Then Continue For '已處理目標
                Dim T As TgInfo = C(i) '核心目標
                Dim D As New ArrayList, Dm As Integer = 0 '此輪搜尋的目標集合
                D.Add(T) : Dm = T.pm '加入搜尋起點目標
                Dim x1 As Integer = T.cx - T.height * 2.5, x2 As Integer = T.cx + T.height * 2.5 '搜尋 X 範圍
                Dim y1 As Integer = T.cy - T.height * 1.5, y2 As Integer = T.cy + T.height * 1.5 '搜尋 Y 範圍
                For j As Integer = 0 To C.Count - 1
                        If i = j Then Continue For '與起點重複略過
                        If skp(j) Then Continue For
                        Dim G As TgInfo = C(j)
                        If G.cx < x1 Then Continue For
                        If G.cx > x2 Then Continue For
                        If G.cy < y1 Then Continue For
                        If G.cy > y2 Then Continue For
                        If G.width > T.height Then Continue For '目標寬度太大略過
                        If G.height > T.height * 1.5 Then Continue For '目標高度太大略過
                        D.Add(G) : Dm += G.pm '合格目標加入集合
                        If D.Count >= 7 Then Exit For '目標蒐集個數已滿跳離迴圈
                Next
                If Dm > pmx Then '對比度高於之前的目標集合
                        pmx = Dm : R = D
                End If
        Next
        '目標群位置左右排序
        If R.Count > 1 Then
                Dim n As Integer = R.Count
                For i As Integer = 0 To n - 2
                        For j As Integer = i + 1 To n - 1
                                Dim Ti As TgInfo = R(i), Tj As TgInfo = R(j)
                                If Ti.cx > Tj.cx Then
                                        R(i) = Tj : R(j) = Ti
                                End If
                        Next
                Next
        End If
        Return R
End Function
'二值化
Private Function DoBinary(ByVal b(,) As Byte) As Byte(,)
        Dim Gdim As Integer = 40 '計算區域亮度區塊的寬與高
        Dim Th(,) As Integer = ThresholdBuild(b, Gdim) '建立二值化使用之門檻值陣列
        Dim Z(nx - 1, ny - 1) As Byte '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
            Next
        Next
        Return Z
    End Function
    '門檻值陣列建立
    Private Function ThresholdBuild(ByVal b(,) As Byte, ByVal Gdim As Integer) 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
    '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
    '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 '目標集合物件
        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.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) : nc.Add(k) '本輪搜尋起點
                                G.P.Add(k) '目標物件點集合
                                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
                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 : G.height = G.ymx - G.ymn + 1 '寬高度計算
                If G.height < minHeight Or G.height > maxHeight Then Continue For '太矮或太高
                If G.width < minwidth Or G.width > maxWidth Then Continue For '太窄或太寬
                G.cx = (G.xmn + G.xmx) / 2 : G.cy = (G.ymn + G.ymx) / 2 '中心點
                '計算目標的對比度
                For m As Integer = 0 To G.P.Count - 1
                    Dim pm As Integer = PointPm(G.P(m))
                    If pm > G.pm Then G.pm = pm '最高對比度的輪廓點
                Next
                A.Add(G) '加入有效目標集合
            Next
        Next
        '以對比度排序
        For i As Integer = 0 To A.Count - 2
            For j As Integer = i + 1 To A.Count - 1
                Dim T As TgInfo = A(i), G As TgInfo = A(j)
                If T.pm < G.pm Then A(i) = G : A(j) = T '互換位置，高對比目標在前
            Next
        Next
        '取得 Tgmax 個最明顯的目標輸出
        Dim C As New ArrayList
        For i As Integer = 0 To Tgmax - 1
            If i > A.Count - 1 Then Exit For '超過總目標數
            Dim T As TgInfo = A(i) : T.ID = i '建立以對比度排序的序號
            C.Add(T)
        Next
        Return C '回傳目標物件集合
    End 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 = 0 '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
    '建立單一目標的二值化矩陣
    Private Function Tg2Bin(ByVal T As TgInfo) As Byte(,)
        Dim b(nx - 1, ny - 1) As Byte '二值化陣列
        For k As Integer = 0 To T.P.Count - 1
            Dim p As Point = T.P(k) : b(p.X, p.Y) = 1 '起點
            '向右連通成實心影像
            Dim i As Integer = p.X + 1
            Do While Z(i, p.Y) = 1 And i < T.xmx
                b(i, p.Y) = 1 : i += 1
            Loop
            '向左連通成實心影像
            i = p.X - 1
            Do While Z(i, p.Y) = 1 And i > T.xmn
                b(i, p.Y) = 1 : i -= 1
            Loop
        Next
        Return b
    End Function
    '將單一目標轉正
    Private Function RotateTg(ByVal b(,) As Byte, ByRef T As TgInfo, ByVal A As Double) As Byte(,)
        If A = 0 Then Return b '無傾斜不須旋轉
        If A > 0 Then A = -A '順或逆時針傾斜時需要旋轉方向相反，經過推導 A 應該永遠為負值
        Dim R(1, 1) As Double '旋轉矩陣
        R(0, 0) = Math.Cos(A) : R(0, 1) = Math.Sin(A) : R(1, 0) = -R(0, 1) : R(1, 1) = R(0, 0)
        Dim x0 As Integer = T.xmn, y0 As Integer = T.ymx '左下角座標
        '旋轉後之目標範圍
        Dim xmn As Integer = nx, xmx As Integer = 0, ymn As Integer = ny, ymx As Integer = 0
        For i As Integer = T.xmn To T.xmx
            For j As Integer = T.ymn To T.ymx
                If b(i, j) = 0 Then Continue For '空點無須旋轉
                Dim x As Integer = i - x0, y As Integer = y0 - j '轉換螢幕座標為直角座標
                Dim xx As Integer = x * R(0, 0) + y * R(0, 1) + x0 '旋轉後 X 座標
                If xx < 1 Or xx > nx - 2 Then Continue For
                Dim yy As Integer = y0 - (x * R(1, 0) + y * R(1, 1)) '旋轉後 Y 座標
                If yy < 1 Or yy > ny - 2 Then Continue For
                b(i, j) = 0 : b(xx, yy) = 1 '清除舊點繪製新點
                '旋轉後目標的範圍偵測
                If xx < xmn Then xmn = xx
                If xx > xmx Then xmx = xx
                If yy < ymn Then ymn = yy
                If yy > ymx Then ymx = yy
            Next
        Next
        '重設目標屬性
        T.xmn = xmn : T.xmx = xmx : T.ymn = ymn : T.ymx = ymx
        T.width = T.xmx - T.xmn + 1 : T.height = T.ymx - T.ymn + 1
        T.cx = (T.xmx + T.xmn) / 2 : T.cy = (T.ymx + T.ymn) / 2
        '補足因為旋轉運算實產生的數位化誤差造成的資料空點
        For i As Integer = T.xmn To T.xmx
            For j As Integer = T.ymn To T.ymx
                If b(i, j) = 1 Then Continue For
                If b(i + 1, j) + b(i - 1, j) + b(i, j + 1) + b(i, j - 1) >= 3 Then
                    b(i, j) = 1
                End If
            Next
        Next
        Return b
    End Function
    '建立正規化目標二值化陣列
    Private Function NmBin(ByVal T As TgInfo, ByVal M(,) As Byte,
                           ByVal mw As Integer, ByVal mh As Integer) As Byte(,)
        Dim fx As Double = mw / Pw, fy As Double = mh / Ph
        Dim V(Pw - 1, Ph - 1) As Byte
        For i As Integer = 0 To Pw - 1
            Dim sx As Integer = 0 '過窄字元的平移量，預設不平移
            If T.width / mw < 0.75 Then '過窄字元，可能為 1 或 I
                sx = (mw - T.width) / 2 '平移寬度差之一半
            End If
            Dim x As Integer = T.xmn + i * fx - sx
            If x < 0 Or x > nx - 1 Then Continue For
            For j As Integer = 0 To Ph - 1
                Dim y As Integer = T.ymn + j * fy
                V(i, j) = M(x, y)
            Next
        Next
        Return V
    End Function
    '最佳字元
    Private Function BestC(ByVal A(,) As Byte) As ChInfo
        Dim C As New ChInfo '辨識字元資料結構
        For m As Integer = 0 To 1 '六或七碼字型
            For k As Integer = 0 To 35 '0-9，A-Z
                For x As Integer = -1 To 1 '左右偏移
                    For y As Integer = -1 To 1 '上下偏移
                        Dim n0 As Integer = 0, nf As Integer = 0 '字模黑點數，符合點數
                        For i As Integer = 0 To Pw - 1
                            Dim ix As Integer = i + x
                            If ix < 0 Or ix > Pw - 1 Then Continue For
                            For j As Integer = 0 To Ph - 1
                                Dim jy As Integer = j + y
                                If jy < 0 Or jy > Ph - 1 Then Continue For
                                If P(m, k, i, j) = 0 Then
                                    If A(ix, jy) = 1 Then nf -= 1 '目標與字模不符合點數-1
                                Else
                                    n0 += 1 '字模黑點數累計
                                    If A(ix, jy) = 1 Then nf += 1 '目標與字模符合點數+1
                                End If
                            Next
                        Next
                        Dim v As Integer = nf * 1000 / n0 '符合點數百分比
                        If v > C.ft Then '符合度最高字元
                            C.ft = v : C.Ch = Ch(k)    '符合度，字元
                        End If
                    Next
                Next
            Next
        Next
        For k As Integer = 0 To 1 '變形的 6 或 9
            For x As Integer = -1 To 1 '左右偏移
                For y As Integer = -1 To 1 '上下偏移
                    Dim n0 As Integer = 0, nf As Integer = 0
                    For i As Integer = 0 To Pw - 1
                        Dim ix As Integer = i + x
                        If ix < 0 Or ix > Pw - 1 Then Continue For
                        For j As Integer = 0 To Ph - 1
                            Dim jy As Integer = j + y
                            If jy < 0 Or jy > Ph - 1 Then Continue For
                            If P69(k, i, j) = 0 Then
                                If A(ix, jy) = 1 Then nf -= 1 '目標與字模符合點數
                            Else
                                n0 += 1 '字模黑點數累計
                                If A(ix, jy) = 1 Then nf += 1 '目標與字模符合點數
                            End If
                        Next
                    Next
                    Dim v As Integer = nf * 1000 / n0 '符合點數百分比
                    If v > C.ft Then '符合度最高字元
                        C.ft = v '符合度
                        If k = 0 Then C.Ch = "6"c Else C.Ch = "9"c
                    End If
                Next
            Next
        Next
        Return C
    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
    '灰階
    Private Sub RadioButton1_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RadioButton1.CheckedChanged
        If RadioButton1.Checked Then
            If IsNothing(B) Then Exit Sub
            PictureBox1.Image = GrayImg(B)
        End If
    End Sub
    '二值化圖
    Private Sub RadioButton2_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RadioButton2.CheckedChanged
        If RadioButton2.Checked Then
            If IsNothing(Z) Then Exit Sub
            PictureBox1.Image = BWImg(Z)
        End If
    End Sub
    '輪廓線
    Private Sub RadioButton3_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RadioButton3.CheckedChanged
        If RadioButton3.Checked Then
            If IsNothing(Q) Then Exit Sub
            PictureBox1.Image = BWImg(Q)
        End If
    End Sub
    '所有合格目標
    Private Sub RadioButton4_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RadioButton4.CheckedChanged
        If RadioButton4.Checked Then
            If IsNothing(CA) Then Exit Sub
            Dim bmp As New Bitmap(nx, ny)
            For k As Integer = 0 To CA.Count - 1
                Dim T As TgInfo = CA(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
        End If
    End Sub
    '嘗試依據英數字規範修改車牌答案
    Private Function ChkED(ByVal R As LPInfo) As LPInfo
        If IsNothing(R.A) Then Return R '無字串
        Dim C() As Char = R.A.ToCharArray '字串轉成字元陣列
        Dim n1 As Integer = R.A.IndexOf("-") '第一區段長度
        Dim n2 As Integer = C.Length - n1 - 1 '第二區段長度
        Dim d1 As Integer = 0, d2 As Integer = 0 '數字區的起終點
        If n1 = n2 Then Return R '無法判定純數字區段(2-2 或 3-3)
        If n1 > n2 Then '第一區段較長
            d1 = 0 : d2 = n1 - 1
        Else '第二區段較長
            d1 = n1 + 1 : d2 = C.Length - 1
        End If
        '嘗試將純數字區段的英文字改成數字
        For i As Integer = d1 To d2
            C(i) = E2D(C(i))
        Next
        '如果是七碼車牌，強制將前三碼中的數字改成英文
        If n1 = 3 And n2 = 4 Then
            For i As Integer = 0 To 2
                C(i) = D2E(C(i))
            Next
        End If
        R.A = "" '重組字串
        For i As Integer = 0 To C.Length - 1
            R.A += C(i)
        Next
        Return R '回傳字串
    End Function
    '嘗試將英文字母變成相似的數字
    Private Function E2D(ByVal C As Char) As Char
        If C = "B"c Then C = "8"c
        If C = "D"c Then C = "0"c
        If C = "O"c Then C = "0"c
        Return C
    End Function
    '嘗試將英文字母變成相似的數字
    Private Function D2E(ByVal C As Char) As Char
        If C = "8"c Then C = "B"c
        If C = "0"c Then C = "D"c
        Return C
    End Function
    '車牌
    Private Sub RadioButton5_CheckedChanged(ByVal sender As Object, ByVal e As EventArgs) _
        Handles RadioButton5.CheckedChanged
        If RadioButton5.Checked Then
            If IsNothing(V) Then Exit Sub
            PictureBox1.Image = BWImg(V)
        End If
    End Sub
    '上下端平移錯位的實驗
    Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) _
```

```
        Handles Button1.Click
        Dim n As Integer = C.Count '目標個數
        Dim wd As Integer = Pw + (Pw + 2) * n + Pw '虛擬車牌寬度
        Dim S(wd - 1, Ph - 1) As Byte '虛擬車牌陣列
        Dim f As Double = Val(ComboBox1.Text) / (Ph - 1) '每一畫素高度的 X 錯位量
        For j As Integer = 0 To Ph - 1
            Dim dx As Integer = f * (Ph - 1 - j) 'X 移動量
            For i As Integer = 0 To wd - 1
                Dim x As Integer = i + dx
                If x > wd - 1 Or x < 0 Then Continue For
                S(x, j) = V(i, j)
            Next
        Next
        '計算錯位後的虛擬車牌中有幾個垂直空白線？越多表示字元越正直
        Dim n0 As Integer = 0
        For i As Integer = 0 To wd - 1
            Dim m As Integer = 0
            For j As Integer = 0 To Ph - 1
                m += S(i, j)
            Next
            If m = 0 Then n0 += 1
        Next
        Dim bmp As Bitmap = BWImg(S)
        PictureBox1.Image = bmp
        MsgBox(n0)
    End Sub
    '左右傾倒校正
    Private Sub IncCorrect(ByVal V(,) As Byte)
        Dim n As Integer = C.Count '目標個數
        Dim wd As Integer = Pw + (Pw + 2) * n + Pw '虛擬車牌二值化陣列寬度
        Dim mx As Integer = 0, mk As Integer = 0 '空白垂直線最大數，最佳偏移量
        Dim S0(wd - 1, Ph - 1) As Byte '最佳校正之車牌二值化陣列
        For sx As Integer = -15 To 15 '嘗試偏移量
            Dim S(wd, Ph - 1) As Byte '虛擬車牌陣列
            Dim f As Double = sx / (Ph - 1) '每一垂直畫素偏移比例
            For j As Integer = 0 To Ph - 1
                Dim dx As Integer = f * (Ph - 1 - j)
                For i As Integer = 15 To wd - 16
                    Dim x As Integer = i + dx
                    S(x, j) = V(i, j)
                Next
            Next
            Dim n0 As Integer = 0
            For i As Integer = 0 To wd - 1
                Dim m As Integer = 0
                For j As Integer = 0 To Ph - 1
                    m += S(i, j)
                Next
                If m = 0 Then n0 += 1
            Next
            If n0 > mx Then
                mx = n0 : mk = sx : S0 = S
            End If
        Next
        If mk <> 0 Then '需要調整傾倒字元，將修正後之車牌二值化影像重作目標擷取
            ReDim Z(nx - 1, ny - 1)
            For i As Integer = 0 To wd
                For j As Integer = 0 To Ph - 1
```

```
                    Z(i + 10, j + 10) = S0(i, j)
                Next
            Next
            Q = Outline(Z) '建立輪廓點陣列
            CA = getTargets(Q) '建立所有目標物件集合
            ReDim skp(CA.Count - 1) '重設已處理標記陣列
            C = AlignTgs(CA) '群組化車牌目標
            n = C.Count '目標個數
            Dim T(n - 1) As TgInfo, M(n - 1) As Array, w(n - 1) As Integer, h(n - 1) As Integer
            For k As Integer = 0 To n - 1
                Dim G As TgInfo = C(k)
                M(k) = Tg2Bin(G) '建立單一目標的二值化矩陣
                T(k) = G '儲存旋轉後的目標物件
                w(k) = G.width '寬度陣列
                h(k) = G.height '高度陣列
            Next
            Array.Sort(w) '寬度排序小到大
            Array.Sort(h) '高度排序小到大
            Dim mw As Integer = w(n - 2) '取第二寬的目標為標準，避開意外沾連的極端目標
            Dim mh As Integer = h(n - 2) '取第二高的目標為標準，避開意外沾連的極端目標
            For k As Integer = 0 To n - 1
                MC(k) = NmBin(T(k), M(k), mw, mh) '個別字元正規化矩陣
            Next
        End If
    End Sub
End Class
```