

厦门大学

# 研究生课程设计报告

车牌识别的研究与实现

姓 名：周青青  
学 号：23220101153252  
专 业 名 称：模式识别与智能系统  
课 程 名 称：数字图像处理  
指导教师姓名：周绮凤 副教授

2011 年 5 月 27 日

车牌识别的研究与实现

摘要： 车牌识别技术 ( License Plate Recognition, LPR ) 是计算机视觉与模式识别技术在智能交通领域应用的重要课题之一。本文 采用一种基于垂直 Sobel 算子检测边缘和投影法的车牌定位方法 ，根据车牌区域竖直纹理突出的特点， 利用 Sobel 垂直算子提取边缘，然后投影得到车牌的水平投影图，利用水平投影图分割出车牌，再运用膨胀运算进行车牌垂直定位。利用旋转投影法寻找车牌倾斜角度，然后用双线性插值进行车牌图像倾斜矫正。对经过精定位的车牌利用垂直投影法 ，找出各个字符的中心点，根据字符中心点位置进行切割。同时将切割出来的图片与数据库中的每张模板图片进行减法运算，计算该图片与模板的均方误差，其均方误差最小的模板即为识别出的字符。 最后，本文设计了一个友好的 GUI 演示界面，实现车牌图像处理方便 直观，易于用户操作。

【设计方案】

为实现车牌的自动识别功能，本文的主要工作分为，图像灰度化、车牌粗定位、车牌倾斜校正、去除车牌边框、字符分割和字符识别几部分。首先把采集到的 24 位彩色图灰度化，转换成灰度图，便于下一步处理，车牌粗定位在整张图片中把车牌区域提取出来（该车牌区域包括了车牌的边框），由于摄像头拍摄角度与拍摄距离的问题，一般车牌图像会存在倾斜，所以必须进行倾斜校正。通过倾斜校正好的车牌图片再进行上下左右边框的切除后，获得精定位后的车牌图片，这时再把车牌中的七个字符分割出来，一个一个用模板匹配的方法来进行识别，最后显示识别结果。 整个系统的具体过程如图 1 所示。

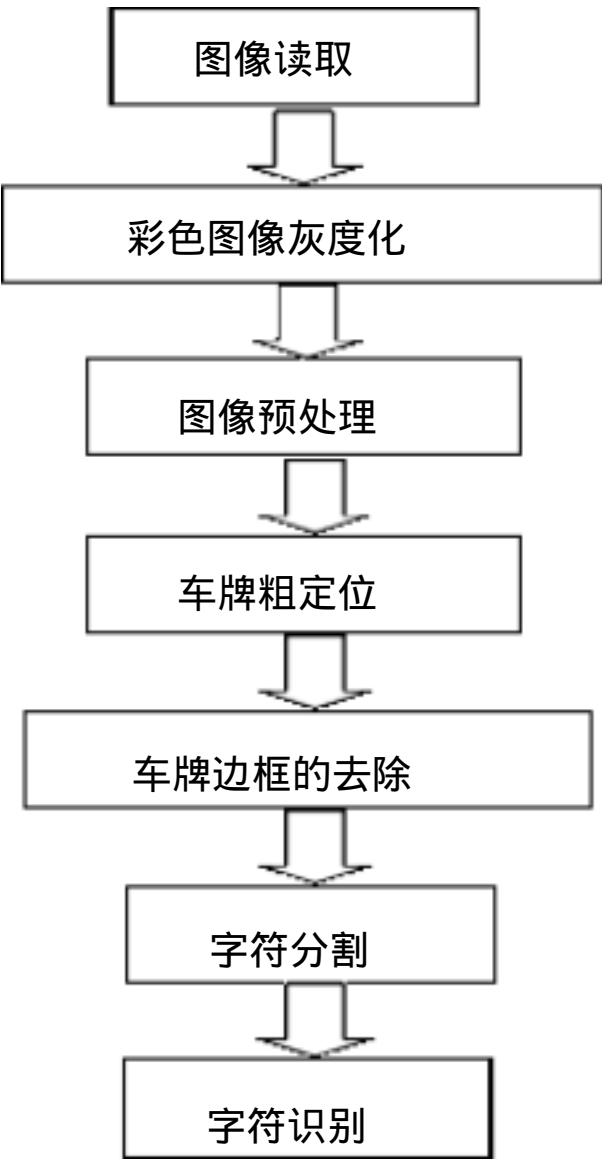


图 1 车牌识别流程

【设计步骤】

1、车牌初定位

车牌粗定位首先用边缘检测突出车牌区域，把车牌图像进行水平、垂直投影，再运用形态学膨胀等操作，把车牌区域从整个图像中提取出来。

1 ) 颜色转换。 本文采用基于灰度图像的车牌定位方法，先将原始 24 位的彩色图利用加权平均值法转化为灰度图，，采用的图像灰度值的转换公式为：

$$I = 0.299R + 0.587G + 0.114B \tag{1}$$

得到原始彩色图及灰度图如图 2 所示：



图 2 彩色图与灰度图对比

2 ) 灰度校正。在采集图片的过程中，由于受采集设备工作环境参数设置的影响，采集到的车牌图像总体对比度不尽相同，这使得一些车牌图像整体偏亮或偏暗，因此，本文采用灰度规格化将车牌图像的灰度变化调整到一个统一的范围，使不同图像具有相同的灰度均值和方差，使后续处理具有相同的基准。其规格化公式为：令 I 表示原始车牌图像，其大小为 M N，I (x, y) 表示图像中像素点 (x, y) 的灰度值。令 G 表示灰度规格化后的车牌图像，G (x, y) 为规格化后图像中像素点 (x, y) 的灰度，则：

$$G(x,y) = \begin{cases} \frac{Mean_0}{Mean} \sqrt{\frac{Var_0}{Dev}} \frac{(I(i,j) - Mean)^2}{Dev} & \text{if } I(i,j) > Mean \\ \frac{Mean_0}{Mean} \sqrt{\frac{Var_0}{Dev}} \frac{(I(i,j) - Mean)^2}{Dev} & \text{otherwise} \end{cases} \tag{2}$$

其中：

$$\begin{aligned} Mean &= \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i,j) \\ Var &= \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(x,y) - Mean)^2 \end{aligned} \tag{3}$$

Mean 和 Dev 分别是原始图像的灰度均值和方差， $Mean_0$  和  $Dev_0$  分别是期望的灰度均值和方差。在此  $Mean_0$  取 100， $Dev_0$  取 3000。校正后的灰度图如图 3 所示：



图 3 校正后灰度图

3) 车牌水平粗定位。首先利用垂直 Sobel 边缘检测方法对矫正后的灰度图进行检测，其检测结果如图 4 所示：



图 4 垂直 Sobel 边缘 检测

由上图所示，经过 Sobel 垂直算子检测之后只剩下需要定位的牌照区域和一些汽车的边框，其他的景物已经被消除。牌照区域的纹理有明显特征，车牌区域字符密集，有较多的竖条纹，所以可以运用该方法来突出图片中的车牌区域。

经过边缘检测后车牌图片成了二值化图像，利用水平投影法将车牌像素灰度值按列方向垂直累加，其计算公式为：

$$Y(i) = \sum_{j=1}^M f(i, j) \tag{4}$$

把  $Y(i)$  归一化，使得它的值的范围在 0 到 1 之内。

$$y(i) = \frac{Y(i) - \min(Y(i))}{\max(Y(i)) - \min(Y(i))}, i = 1, 2, \dots, N \tag{5}$$

得到如图 5 所示的投影图：

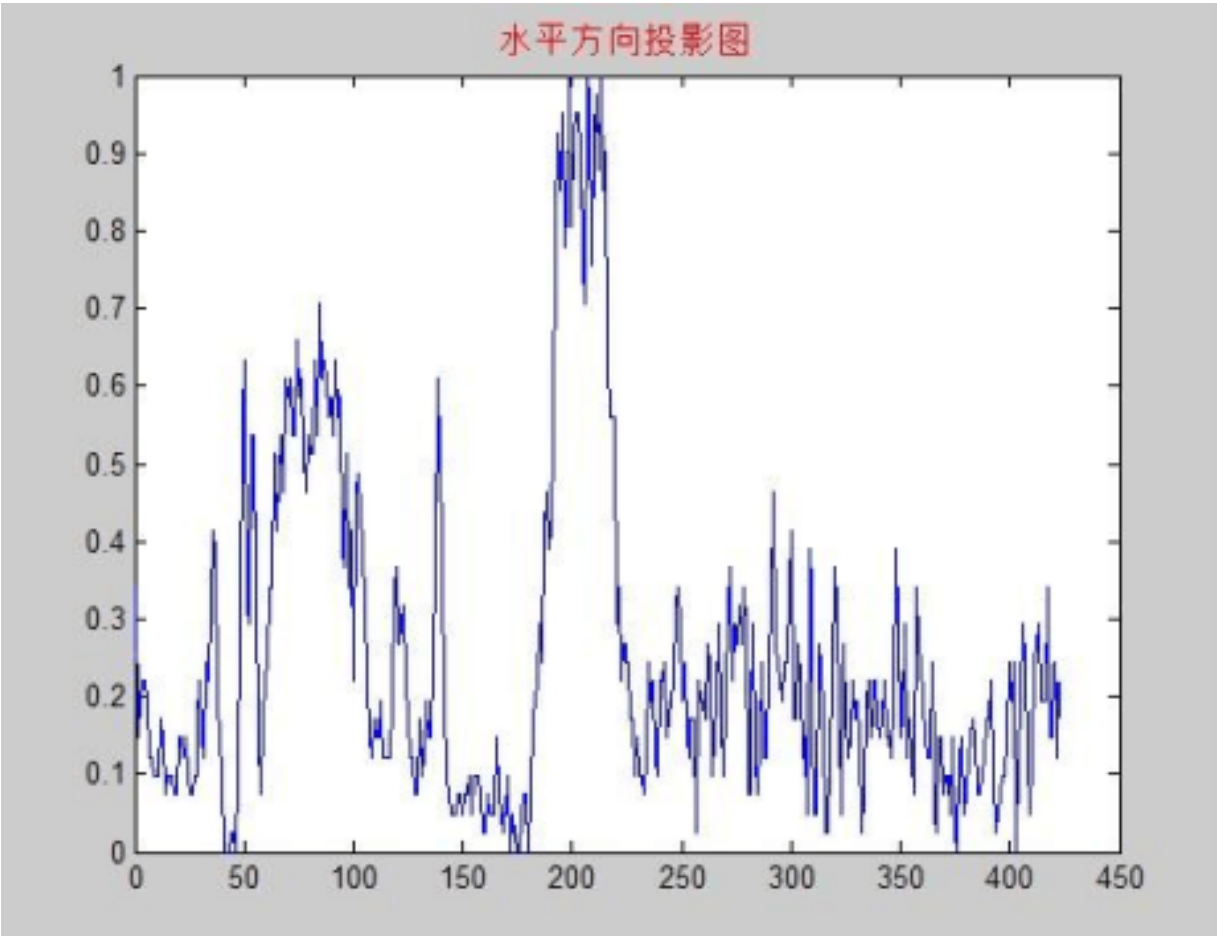


图 5 水平方向投影图

根据车牌悬挂位置的特点，从投影图中的下面开始向上扫描，找出最靠下的大于均值一行，然后继续向上扫描，直至小于均值的一行，并记下相应的行。如果上下两行之间的宽度在一定范围内，则认为找到车牌，否则就是伪车牌，去除之后继续向上找。车牌具有自身的特点，我们限定高度为（ 15 ， 55 ）之间的为车牌，否则为伪车牌。在确定上下边界后，将车牌提取出，如图 6 所示：

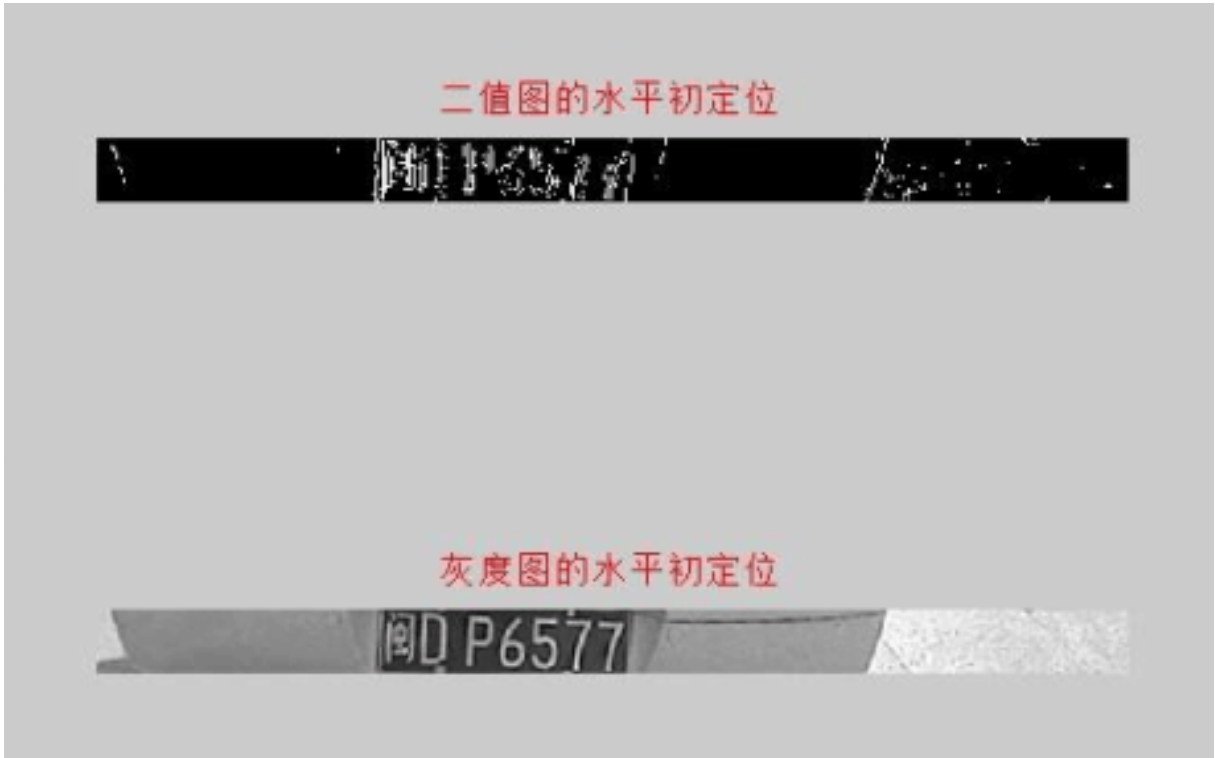


图 6 图像的水平初定位

4 ) 车牌垂直粗定位。对车牌图像进行垂直投影时，先对水平粗定位后提取的车牌进行形态学的膨胀，选取的结构元为  $2 \times 8$ ，使得车牌字符部分连成一片，在垂直投影图上



没有间断。再对膨胀后的图进行垂直投影，其结果如图 7 所示：

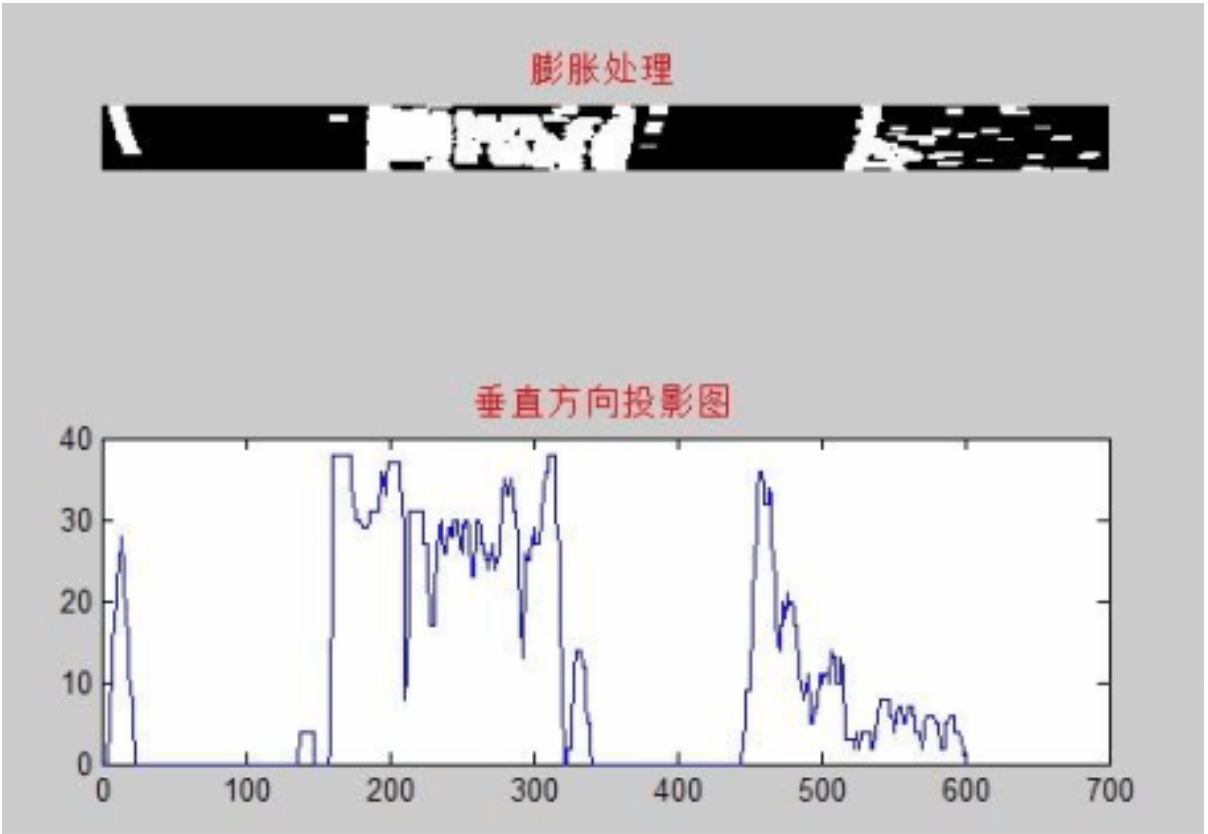


图 7 膨胀处理及其对应的投影图

同样按照水平投影的切割法，从投影图的右端开始扫描，根据经验值，认为宽度在（ 100， 250 ）之间的为车牌，否则为伪车牌，经切割后得到如图 8 所示的车牌初定位：

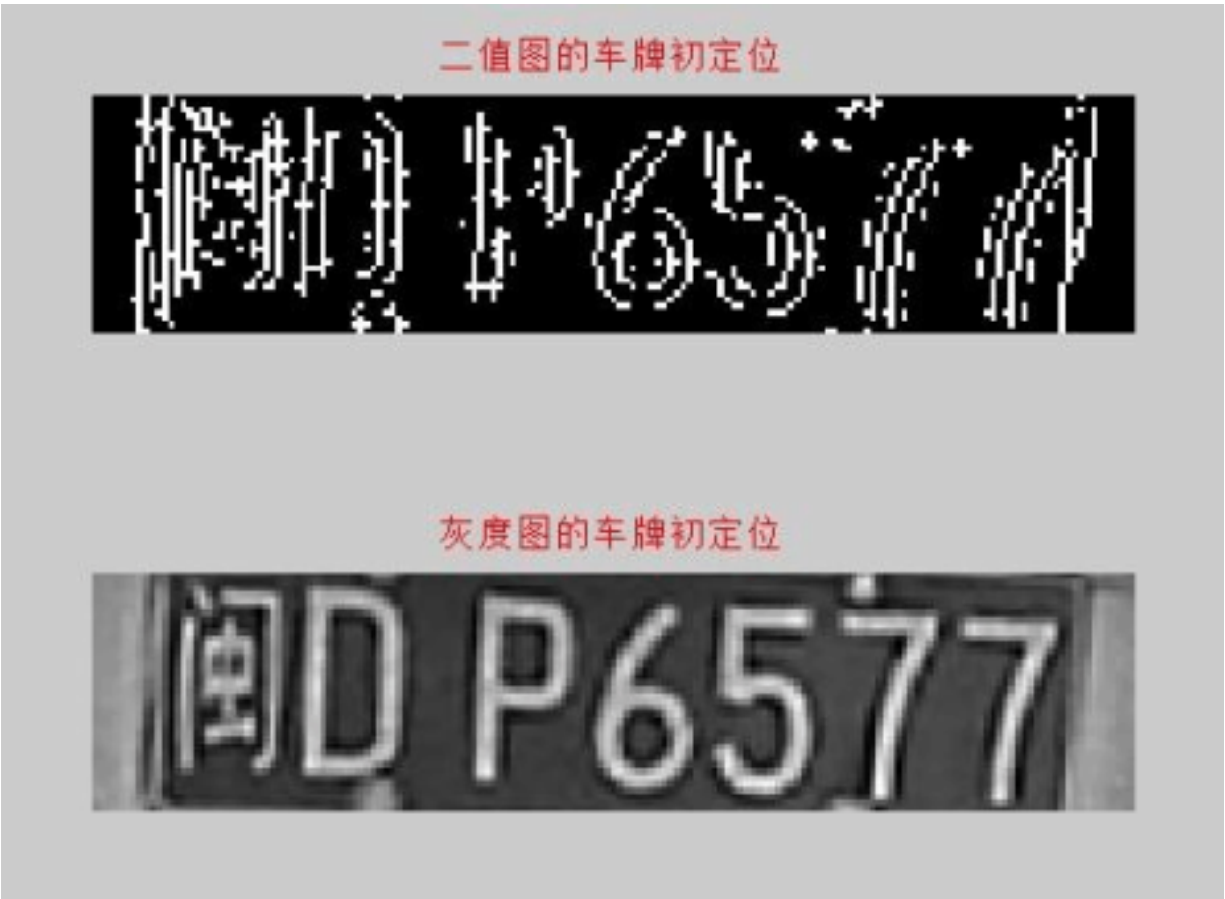


图 8 车牌的初定位

2、车牌精定位

1 ) 车牌图像的 倾斜矫正。 本文采用基于投影法的车牌校正技术 来对车牌图像 进行倾斜矫正，其算法如下：首先使用 Sobel 水平算子做水平边缘检测，突出水平边缘；然后再对它做腐蚀运算，为后续使用旋转、投影结合法求倾斜角做准备。 采用先旋转后判断的原则，把边缘图像在一定角度的小范围内用最近邻法（ nearest ）做旋转，然后再做水平投影，每旋转一次都求得一个单行最大值 。经倾斜校正后的图如图 9— 10 所示：

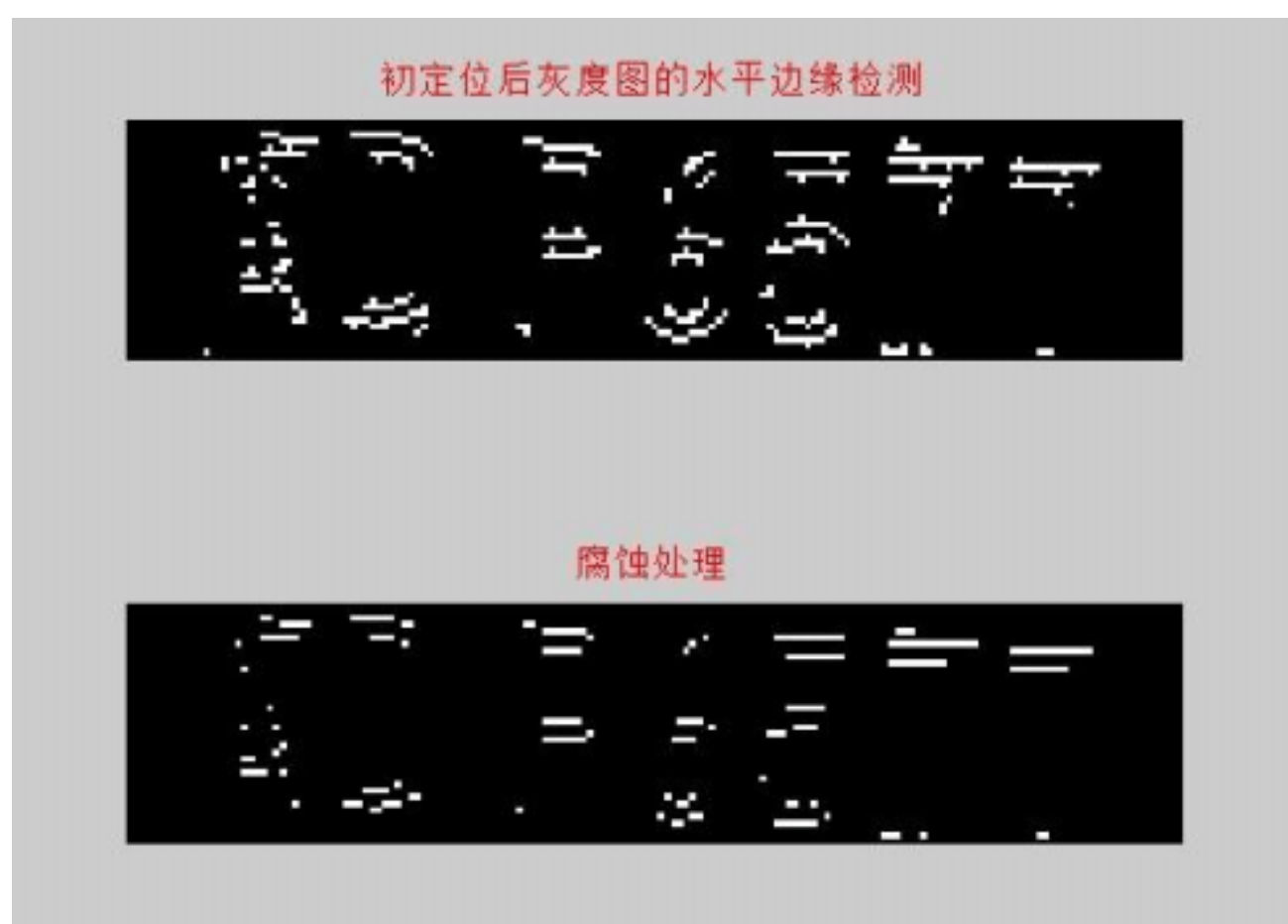


图 9 Sobel 水平边缘检测及腐蚀处理



图 10 倾斜校正图

2) 车牌边框的切除。车牌粗定位的目的是找到一个包含车牌的区域，区域面积要比车牌大，保证车牌信息不丢失。往往提取出来的车牌都带有边框，由于牌照边框的存在，使得字符的垂直投影的值全部大于零，也使得车牌中的字符粘连在一起，从而不能确定分割点，对后续字符分割和字符识别造成了很大困难。因此必须首先把边框去掉。

a) 车牌上下边框的切除。首先用垂直 Sobel 算子检测垂直边缘，突出车牌的垂直边缘，然后用膨胀运算使得车牌字符区域突出，其中取结构元为  $2 \times 2$ 。字符分布在车牌的中心区域，与上下边框都存在着一定的空隙，通过对牌照图象进行水平方向的投影，在水平投影图上可以找到一定的规律。由此来确定字符的上下边界。最后，利用经验值就可以切除上下边框，其具体过程如图 11—12 所示：

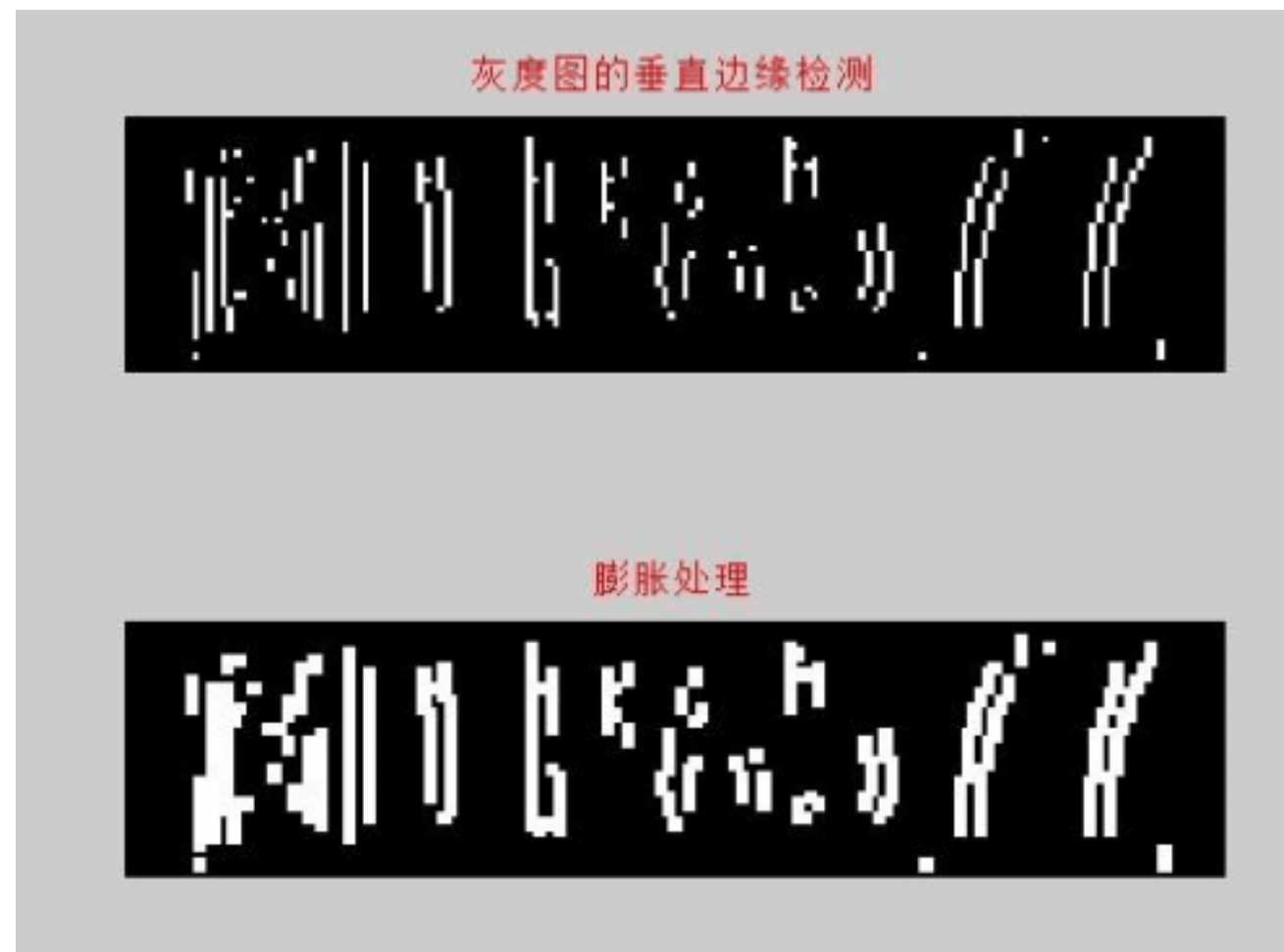


图 11 边缘检测及膨胀处理

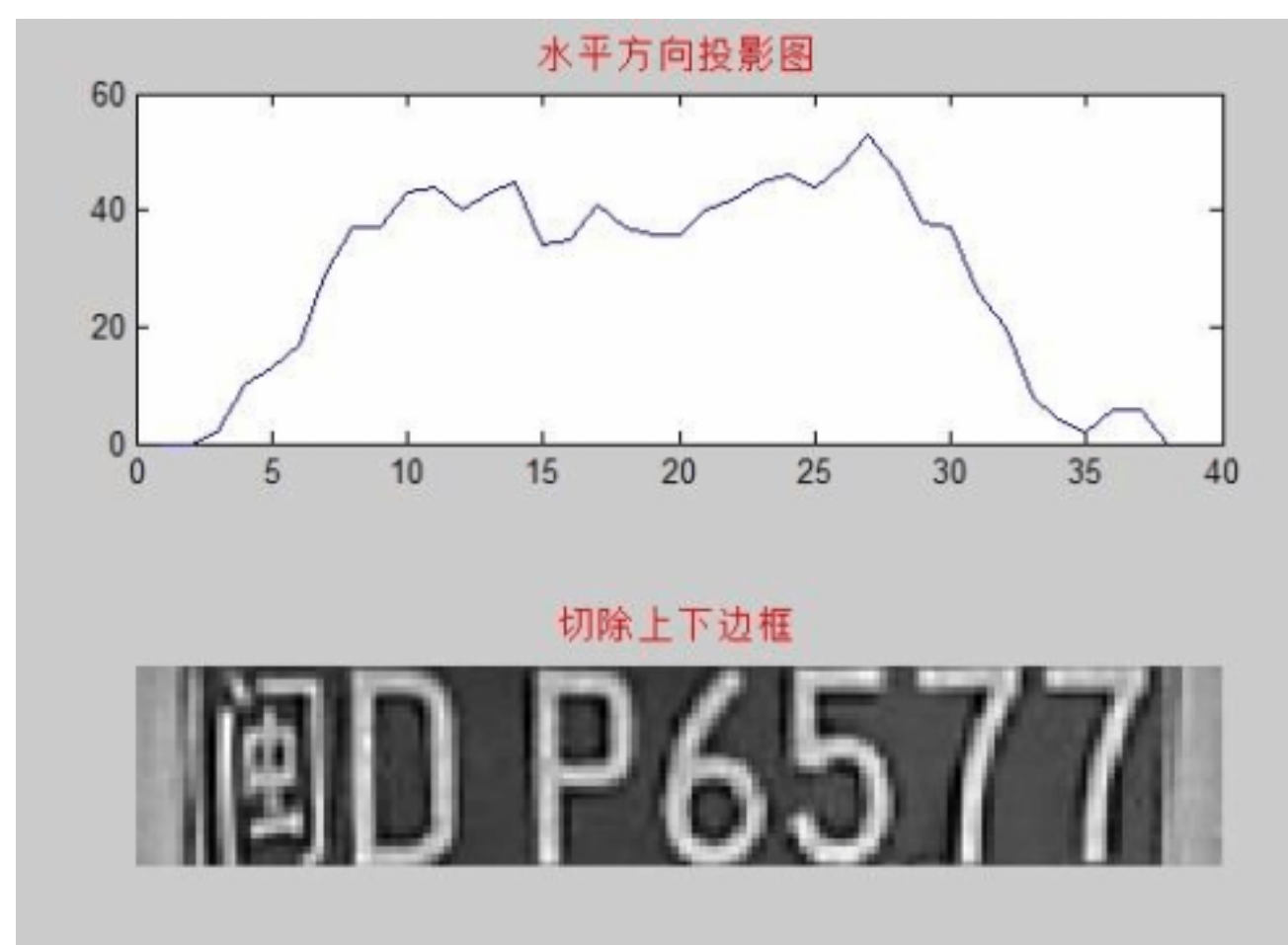


图 12 对应水平投影及切除上下边框

b) 车牌左右边框的切除。按照上述方法进行边缘检测并垂直投影，切除左右两边的黑色部分，其结果如图 13 所示：



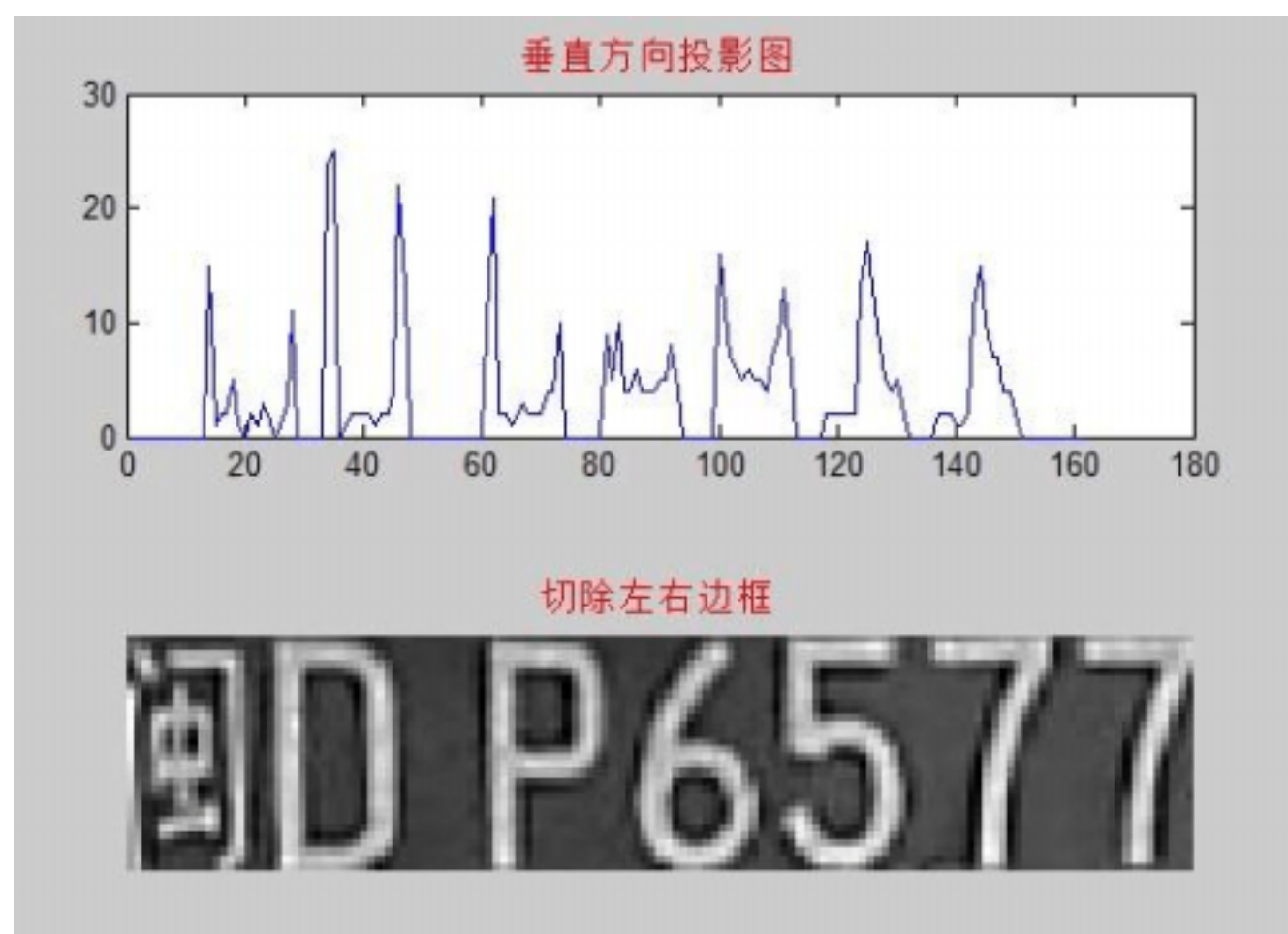


图 13 垂直投影及切除左右边框的效果图

c) 二值化处理。本文采用最传统的二值化方法——均值二值化，它是以整个图像的灰度值的平均值为阈值  $T$  来对图象进行二值化，其计算公式为：

$$T = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N f(i, j) \quad (6)$$

采用均值二值化简单，速度快，能达到满意的二值化效果：处理结果如图 11 所示：

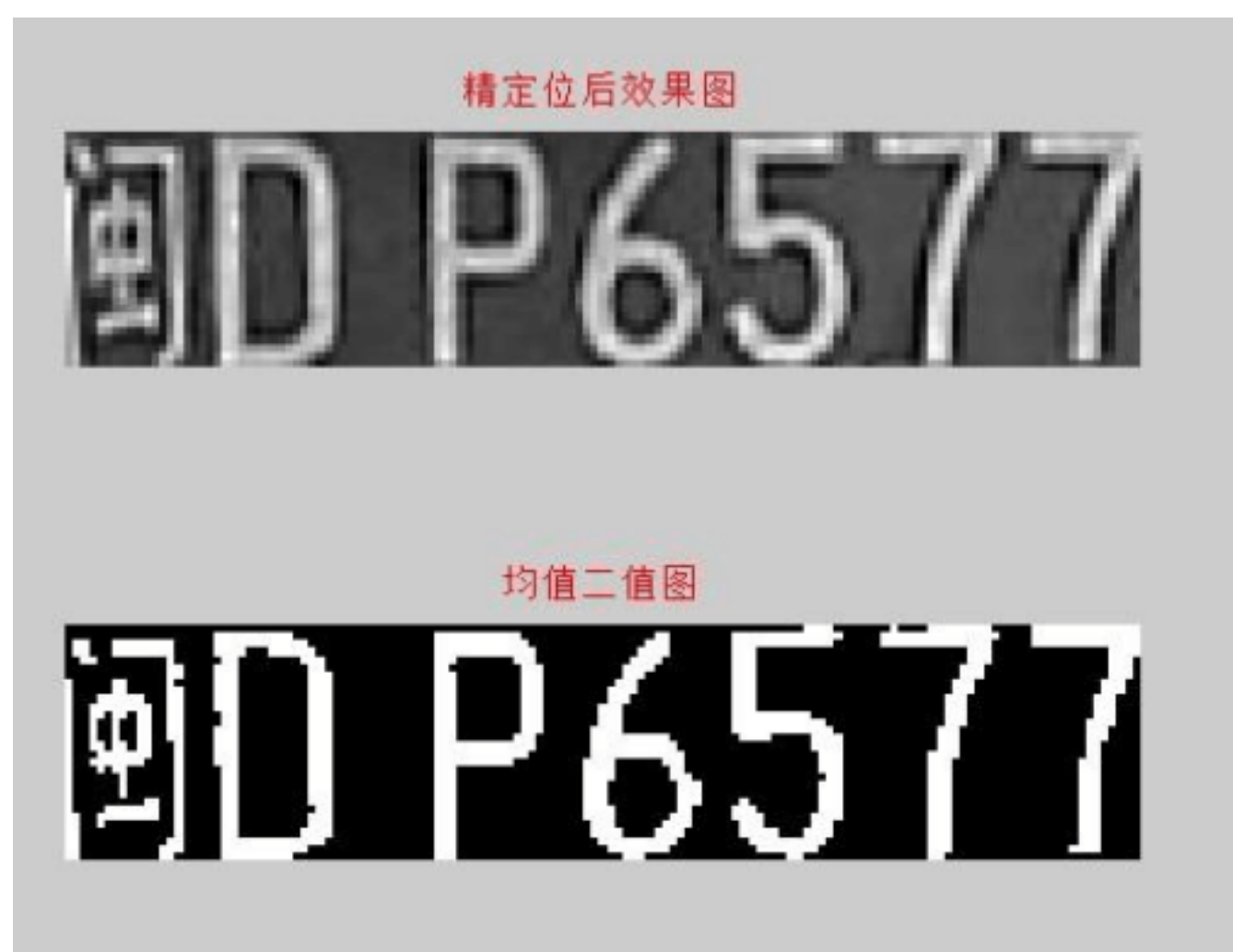


图 14 均值二值图

3、字符分割及识别

1) 字符分割。对以上均值二值图进行垂直方向的投影，投影结果如下图 15 所示。根据投影图，找出各个字符的起始位置和终止位置，计算每个字符的中心位置并记下，同时计算各个字符的宽度，找出最大宽度。字符切割的方法为根据各个字符的中心位置，向左右两边扩展最大字符的 1/2，进行切割。切割结果如图 16 所示。

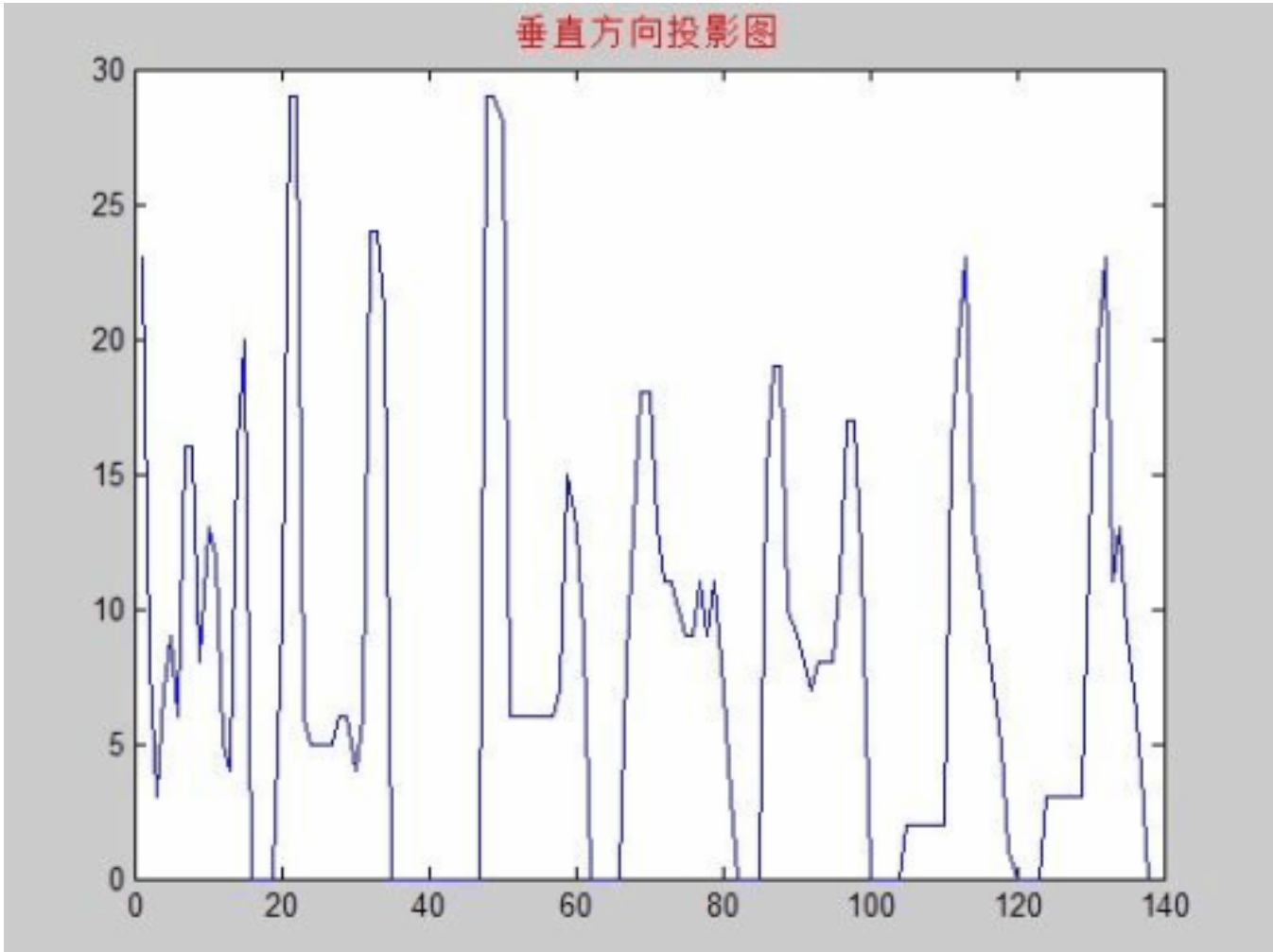


图 15 二值图的垂直投影图

2) 字符识别。 本文采用模板匹配方法对字符进行识别。即 把待识别的图像与 对应的模板进行比较，计算图像和模板灰度值之间的距离，用最小距离法判定所属类。本文选用的标准模板库中包含 “ 0-9 ， A-Z ， 闽桂海云鲁川京津沪 ”，字符均大小为 22 × 14 像素。为使分割得到的自如能顺利与模板进行匹配，先将分割得到的字符利用 MATLAB自带的函数 imresize 将图像大小转化为标准的 22 × 14 像素，便于进行匹配工作。将转化以后的字符与模版中的每个字符进行对应像素的差值并平方， 计算该字符与模版中每个字符的误差平方和，其误差平方和最小的字符即作为该字符的识别结果。其计算公式如下：

$$V = \sum_{i=1}^{14} \sum_{j=1}^{22} (S(i, j) - T(i, j))^2 \tag{7}$$

式中 V 为误差平方和， S(i, j) 为带识别字符第 i 行、第 j 列对应的灰度值， T(i, j) 为模板字符第 i 行、第 j 列对应的灰度值。识别结果如图 16 所示：



图 16 标准化过程及识别结果

#### 4、车牌识别系统演示界面设计

本系统的车牌自动识别是基于 matlab 环境下的，matlab 功能强大，提供了图形用户界面环境（GUI），通过窗口、选单、按钮、文字说明等对象构成一个美观的界面提供用户利用鼠标或键盘方便地实现操作。本文利用 GUI 界面设计了一个友好的界面，来实现对车牌图像处理方便直观，易于用户操作。界面设计如图 17 所示：

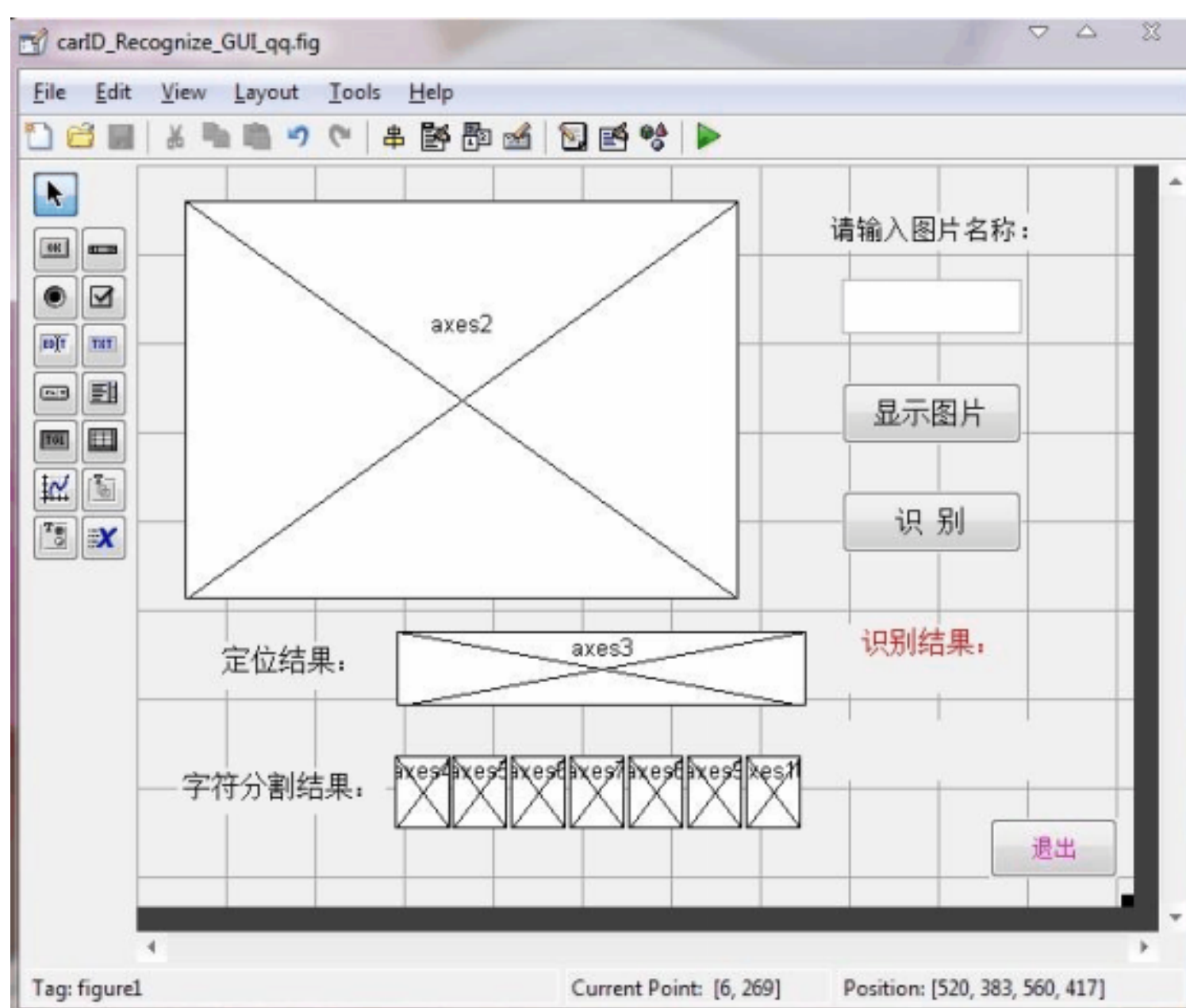


图 17 界面演示设计图



运行界面在可编辑文本框里输入图片名称“car”；点击“显示图片”这个按钮，该图片就被显示在界面中，再点击“识别”按钮，对图像进行车牌定位，字符分割，字符识别等操作，界面设计演示图如图 18 所示：



图 18 演示设计图

## 【参考文献】

- [1] 刘文峰,吴学毅,刘长富.基于 RGB 色度空间的车牌定位及矫正.武汉大学学报(信息科学版).2006,31(9):785-787
- [2] 安化萍,丁劲生,高满屯.基于 Sobel 垂直算子检测的车牌定位算法.计算机工程与设计.2005,8(26):2030-2032
- [3] 胡英,陈辉,王绪本.基于分形与投影法的汽车牌照定位.河北工业大学学报.2004,5(33):56-59
- [4] 左奇,史忠科.一种基于数学形态学的实时车牌图像分割方法.中国图象图形学报(A版).2003,8(3):281-285
- [5] 冈萨雷斯.数字图像处理(第二版).第一版.北京:电子工业出版社,2003.3:343-345
- [6] 张永宜.汽车牌照定位分割及车牌字符识别研究:[硕士学位论文].西北大学.2006.6
- [7] 王鉴.基于数字图像处理的车辆牌照识别技术的研究:[硕士学位论文].四川大学.2005,5

## 【附录】

MATLAB代码实现：

```
%%%1-0    清空环境变量
clc;
clear;

%%%1-1    读图并灰度化
I=imread('car.jpg');%    读取图像文件
figure,subplot(1,2,1);
imshow(I,[]),title('    原始彩色图像    ','color','r');
[h,w,d]=size(I);
I=double(I);
R=I(:,:,1);
G=I(:,:,2);
B=I(:,:,3);
J=0.299*R+0.587*G+0.114*B;%    采用加权平均值法进行灰度化
subplot(1,2,2);
imshow(J,[]),title('    灰度图（加权平均值法）    ','color','r');

%%%1-2    灰度校正
Mean0=100;
Dev0=3000;
Msum=0;%    灰度均值
Dsum=0;%    灰度方差
%求加权平均值法的灰度图像的灰度均值
for i=1:h
    for j=1:w
        Msum=Msum+J(i,j);
    end
end
Mean=Msum/(h*w);
%求原始图像灰度方差
for i=1:h
    for j=1:w
        Dsum=Dsum+(J(i,j)-Mean)^2;
    end
end
Dev=Dsum/(h*w);
%灰度规格化
for i=1:h
    for j=1:w
        if J(i,j)>Mean
            G(i,j)=Mean0+sqrt(Dev0*(J(i,j)-Mean)^2/Dev);
        else
            G(i,j)=Mean0-sqrt(Dev0*(J(i,j)-Mean)^2/Dev);
        end
    end
end
end
figure,imshow(G,[]),title('    校正后灰度图    ','color','r');%    显示灰度校正后图片

%%%1-3    车牌粗定位
```



```

f1=edge(G,'sobel','vertical'); % 垂直边缘检测
figure,imshow(f1),title(' 垂直 Sobel 边缘检测 ','color','r');

i1=0; % 找出最靠下面的行
i2=0; %i2 为车牌下边界
i3=0; %i3 为车牌上边界
i4=0;
F1=f1;% 边缘检测后的图
%Y_index=0;%% 未用到
length1=0;
Y1=zeros(1,h);
Y=zeros(1,h);
for i=1:h
    Y(i)=sum(f1(i,:));
end
y1=(Y-min(Y))./(max(Y)-min(Y));% 归一化
y1_mean=mean(y1);
figure,plot(y1)
hold on
plot(y1_mean,'r-'), title(' 水平方向投影图 ','color','r');
Y1=y1-y1_mean;
%找出大于均值的行
for i=1:h
    if Y1(i)>0;
        Y1(i)=1;
        i1=i; %最靠下的行
    else
        Y1(i)=0;
    end
end
%从投影图中的下面开始向上扫描
while i1>0
    i2=i1;% 下边界
    while Y1(i1)==1&& i1>=2
        length1=length1+1;
        i1=i1-1;
    end
    if length1>15&&length1<55 %判断是否为伪车牌, length 为经验值
        i3=i1;% 上边界
        i1=0;
    elseif length1>=55
        i4=i1+1;
        i1=0;
    end
    length1=0;
    i1=i1-1;
end
% 判断车牌位置
if i3>0
    y_index1=i3-1;% 上边界

```

```

        y_index2=i2+1;%    下边界
        F2=F1(y_index1:y_index2,:);%    二值图的水平初定位
        G1=G(y_index1:y_index2,:);%    灰度图的水平初定位
    end
    if i4>0
        y_index1=i4-1;
        y_index2=i2+1;
        F2=F1(y_index1:y_index2,:);
        G1=G(y_index1:y_index2,:);
    end
    figure,subplot(2,1,1);
    imshow(F2,[]),title('    二值图的水平初定位    ','color','r');%    车牌的水平初定位
    subplot(2,1,2)
    imshow(G1,[]),title('    灰度图的水平初定位    ','color','r');
        f3=F2;
        s=ones(2,8);            % 结构元需要特别选取
        f4=imdilate(f3,s);        % 膨胀，防止车牌字符断裂
    [h1,w1]=size(f4);
    figure,subplot(2,1,1)
    imshow(f4),title('    膨胀处理    ','color','r');

    Y2=zeros(1,w1);
    for j=1:w1
        for i=1:h1
            Y2(j)=Y2(j)+f4(i,j);
        end
    end
    subplot(2,1,2)
    plot(Y2),title('    垂直方向投影图    ','color','r');

length2=0;
left=0;
right=0;
i=w1;
while i>0
    right=i;%    右边界
    while(Y2(i)>0&& i>0)
        length2=length2+1;
        i=i-1;
    end
    if length2>100&&length2<250 %    经验取值
        left=i;%    左边界
        i=0;
    end
    length2=0;
    i=i-1;
end
    x_index1=left-1;
    x_index2=right+1;
    F3=F1(y_index1:y_index2,x_index1:x_index2);

```

```

F4=G(y_index1:y_index2,x_index1:x_index2);
figure,subplot(2,1,1),imshow(F3,[]),title('        二值图的车牌初定位','color','r');%    边缘检测后
的初定位
subplot(2,1,2),imshow(F4,[]),title('        灰度图的车牌初定位        ','color','r');%    灰度图的初定位
%%%1-4    车牌的精定位
f5=edge(F4,'sobel','vertical');        % 灰度图垂直边缘检测
figure,subplot(2,1,1)
imshow(f5,[]),title('    灰 度图的垂直边缘检测        ','color','r'); %    灰度图的初定位后再进行边缘
检测
s=ones(2,2);        % 结构元需要特别选取
f5=imdilate(f5,s);
subplot(2,1,2)
imshow(f5,[]),title('    膨 胀处理 ','color','r');
[h2,w2]=size(f5);
Y3=zeros(h2,1);
for i=1:h2
    for j=1:w2
        Y3(i)=Y3(i)+f5(i,j);
    end
end
figure,subplot(2,1,1)
plot(Y3),title('    水平方向投影图        ','color','r');
Ymax=max(Y3);
Yth=0.17*Ymax;
Y3=Y3-Yth;
n=1;
for i=1:h2
    if Y3(i)>0
        Y3over(n)=i;
        n=n+1;
    end
end
y2_index1=min(Y3over)-1;%    上界
y2_index2=max(Y3over);%    下界
F5=F4(y2_index1:y2_index2,:);
subplot(2,1,2)
imshow(F5,[]),title('    切除上下边框        ','color','r');        % 切除上下边框
L=F5;
[h3,w3]=size(F5);
for i=1:h3
    for j=1:w3
        if F5(i,j)>125
            F5(i,j)=1;
        else
            F5(i,j)=0;
        end
    end
end
end
%imshow(F5,[]),title('    对应二值图        ','color','r');%    二值化以后的图
Y4=zeros(1,w3);

```

```

for j=1:w3
    for i=1:h3
        Y4(j)=Y4(j)+F5(i,j);
    end
end
figure,subplot(2,1,1)
plot(Y4),title('    垂直方向投影图    ','color','r');

ilow=1;
itop=w3;
while ilow<w3
    if Y4(ilow)>0
        break;
    end
    ilow=ilow+1;
end
while itop>1
    if Y4(itop)>0
        break;
    end
    itop=itop-1;
end
x2_index1=ilow;
x2_index2=itop+1;
F6=L(:,x2_index1:x2_index2);%    切除左右边框
L1=F5(:,x2_index1:x2_index2);
%imshow(L1,[]),title('    切除左右边框    ','color','r');
subplot(2,1,2)
imshow(F6,[]),title('    切除左右边框    ','color','r');

%%% 字符分割
figure,subplot(2,1,1)
imshow(F6,[]),title('    切除左右边框    ','color','r');
L2=F6;
[h4,w4]=size(F6);
total=0;
for i=1:h4
    for j=1:w4
        total=total+F6(i,j);
    end
end
vag=total/(h4*w4);
for i=1:h4
    for j=1:w4
        if F6(i,j)>92
            F6(i,j)=1;
        else
            F6(i,j)=0;
        end
    end
end
end

```

```

end
subplot(2,1,2)
imshow(F6,[]),title('    二值化图    ','color','r');
[h5,w5]=size(F6);
Y5=zeros(1,w5);
for j=1:w5
    for i=1:h5
        Y5(j)=Y5(j)+F6(i,j);
    end
end

figure
plot(Y5),title('    垂直方向投影图    ','color','r');
for i=1:w5
    if Y5(i)>0
        Y5(i)=1;
    else
        Y5(i)=0;
    end
end

add=[];
for i=1:w5
    if i==1&&Y5(i)==1
        aa=i;
        add=[add,aa];
    elseif i==w5&&Y5(i)==1
        aa=i;
        add=[add,aa];
    elseif Y5(i)==0&&Y5(i)~=Y5(i-1)
        aa=i-1;
        add=[add,aa];
    elseif Y5(i)==1&&Y5(i)~=Y5(i-1)
        aa=i;
        add=[add,aa];
    end
end

bb=length(add)/2;
cc=[];
dd=[];
for i=1:bb
    wide=add(2*i)-add(2*i-1)+1;
    center=ceil((add(2*i)+add(2*i-1))/2);
    cc=[cc,wide];
    dd=[dd,center];
end
ee=length(dd);
ff=floor(max(cc)/2);
liccode=char(['0':'9' 'A':'Z' '    闽桂海云鲁川京津沪    ']); % 建立自动识别字符代码表
SubBw2=zeros(22,14);%    用于存放两图相减结果
figure

```



```

for i=1:ee
    if i==ee
        left=dd(i)-ff;
        F7=F6(:,left:end);
        subplot(2,ee,i)
        imshow(F7,[]);
    end
    left=dd(i)-ff;
    right=dd(i)+ff;
    F7=F6(:,left:right);
    subplot(2,ee,i),
    imshow(F7,[]);
    if i==ceil(ee/2)
        title(' 字符分割结果 ', 'color','r')
    end
    F8=imresize(F7,[22 14]);% 变换为 22 行 *14 列标准子图
    subplot(2,ee,i+ee),
    imshow(F8,[]);
    if i==ceil(ee/2)
        title(' 字符标准化 (22 行 *14 列 )', 'color','r')
    end
%     fname=strcat('image',int2str(i),'.jpg');
%     imwrite(F8,fname,'jpg') %%%% 用于输出切割图像
Error=[];
for j=1:length(liccode)
    fname=strcat('sam\Sam',liccode(j),'.jpg');
    F9=imread(fname);
    for r=1:22
        for c=1:14
            SubBw2(r,c)=(F8(r,c)-F9(r,c))^2; % 利用误差平方和方法进行识别
        end
    end
    Dmax=0;
    for r=1:22
        for c=1:14
            if SubBw2(r,c)~=0
                Dmax=Dmax+1;% 统计差值不为 0 的个数
            end
        end
    end
    Error(j)=Dmax;
end
MinError=min(Error);% 取误差的最小值
findc=find(Error==MinError);% 查找最小误差的图像
RegCode(i*2-1)=liccode(findc(1));
RegCode(i*2)=' ';% 输出最小误差图像
end
title([' 识别结果 ', RegCode], 'Color','r');

```