

# 汽车号牌识别系统设计

\*\*

中文摘要： 随着二十一世纪到来，经济快速发展和人们生活水平显著提高，汽车逐渐成为家庭的主要交通工具。汽车的产量快速增多，车辆流动也变得越来越频繁，因此给交通带来了严重问题，如交通堵塞、交通事故等，智能交通系统 (Intelligent Transportation System) 的产生就是为了从根本上解决交通问题。在智能交通系统中车牌识别技术占有重要位置，车牌识别技术的推广普及必将对加强道路管理、城市交通事故、违章停车、处理车辆被盗案件、保障社会稳定等方面产生重大而深远的影响。

该设计主要研究基于 MATLAB 软件的汽车号牌识别系统设计，系统主要包括图像采集、图像预处理、车牌定位、字符分割、字符识别五大核心部分。系统的图像预处理模块是将图像经过图像灰度化、图像增强、边缘提取、二值化等操作，转换成便于车牌定位的二值化图像；利用车牌的边缘、形状等特征，再结合 Roberts 算子边缘检测、数字图像、形态学等技术对车牌进行定位；字符的分割采用的方法是将二值化后的车牌部分进行寻找连续有文字的块，若长度大于设定的阈值则切割，从而完成字符的分割；字符识别运用模板匹配算法完成。以上每个功能模块用 MATLAB 软件实现，最后识别出车牌，在研究设计的同时对其中出现的问题进行具体分析、处理，并寻求更优的方法。

关键词： MATLAB, 车牌识别系统，字符识别，图像处理

## 一、总体设计

汽车号牌识别系统技术是从一幅车辆图像中准确定位出车牌区域，然后经过字符切割和字符识别来实现车辆牌照的自动识别。主要流程图如下：

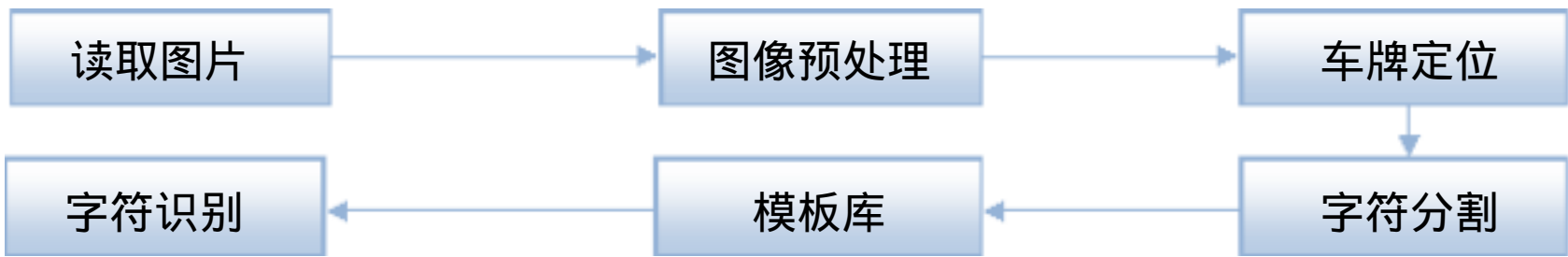


图 1.1

## 二、总体功能模块

基于 MATLAB 车牌识别系统主要包括图像采集、图像预处理、车牌定位、字符分割、字符识别五个关键环节<sup>[11]</sup>，其基本工作如下：

(1) 图像采集：使用摄像头、照相机拍摄采集图像。

- (2) 图像预处理：把图像转换成便于定位的二值化图像，需要经过图像灰度化、图像增强、边缘提取、二值化操作。
- (3) 车牌定位：利用车牌的边缘、形状等特征，再结合 Roberts 算子边缘检测、数字图像、形态学等技术对车牌进行定位。
- (4) 字符分割：以二值化后的车牌部分进行寻找连续有文字的块，若长度大于设定的阈值则切割，从而完成字符的分割。
- (5) 字符识别：运用模板匹配算法将分割后的字符二值化，并将其尺寸大小缩放为模板库中字符的大小，然后与所有的模板进行匹配，准确地识别车牌。输出识别结果，并进行数据存储。

由于纹理特征车牌灰度图像的边缘、图像水平方向上的方差、水平方向上的梯度等比较稳定而且易于提取，所以本系统车牌定位算法采用纹理特征作为车牌的主要特征。在汽车号牌字符识别中，由于汉字的复杂性所以本设计的模板库字符包含 5 个汉字，26 个大写英文字母及 10 个阿拉伯数字。首字符为汉字，第 2-6 个字符为英文字母或数字。本系统采用的是边缘检测的方法实现车牌定位的，寻找连续有文字的块的方法实现字符分割，模板匹配法来确定最终的识别结果。

三、具体设计

下图 3.1 流程图简要的概述了基本步骤：

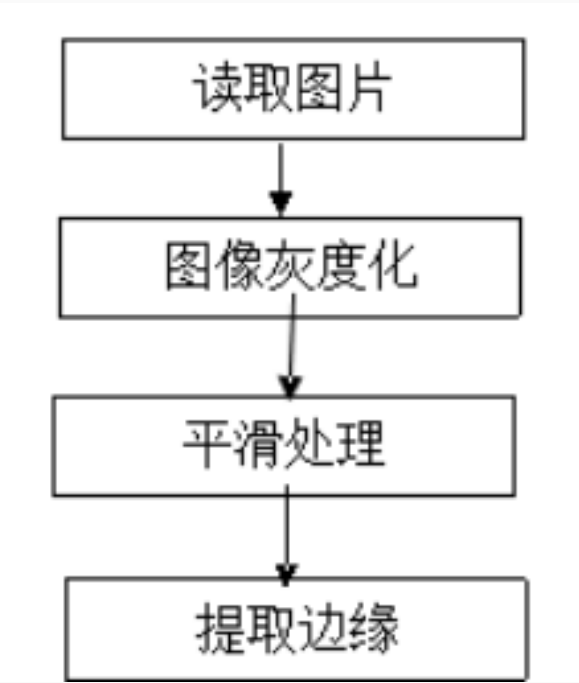


图 3.1



图 3.2

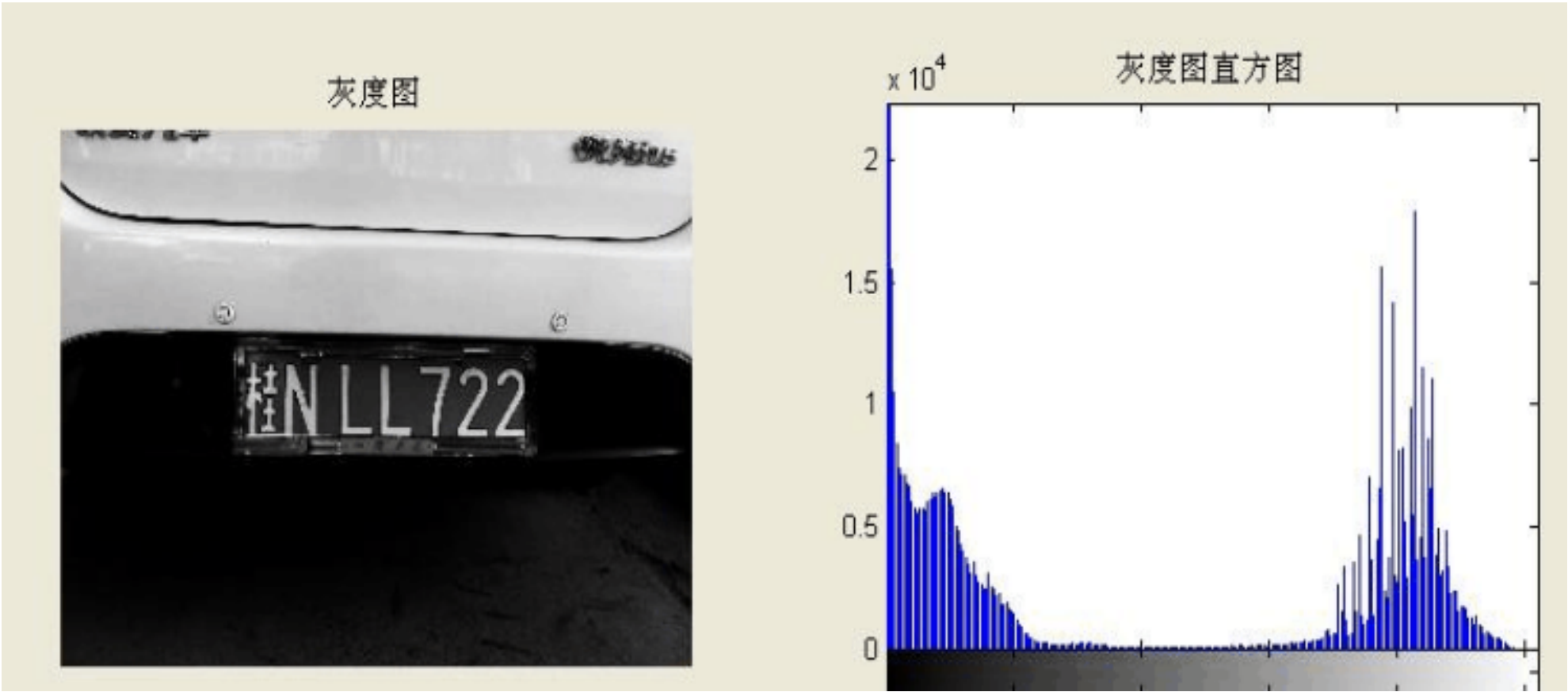


图 3.3



图 3.4



图 3.5

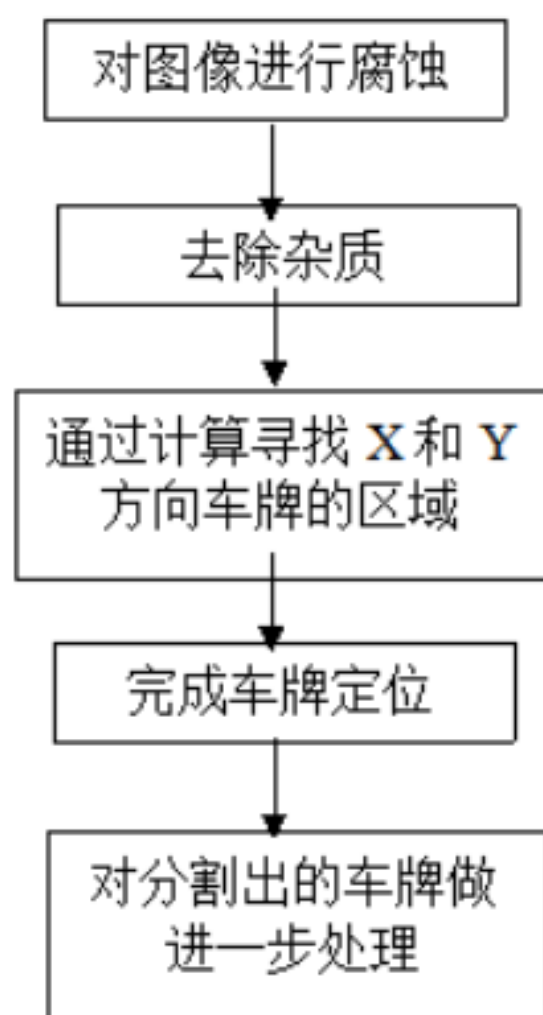


图 3.6



图 3.7



图 3.8

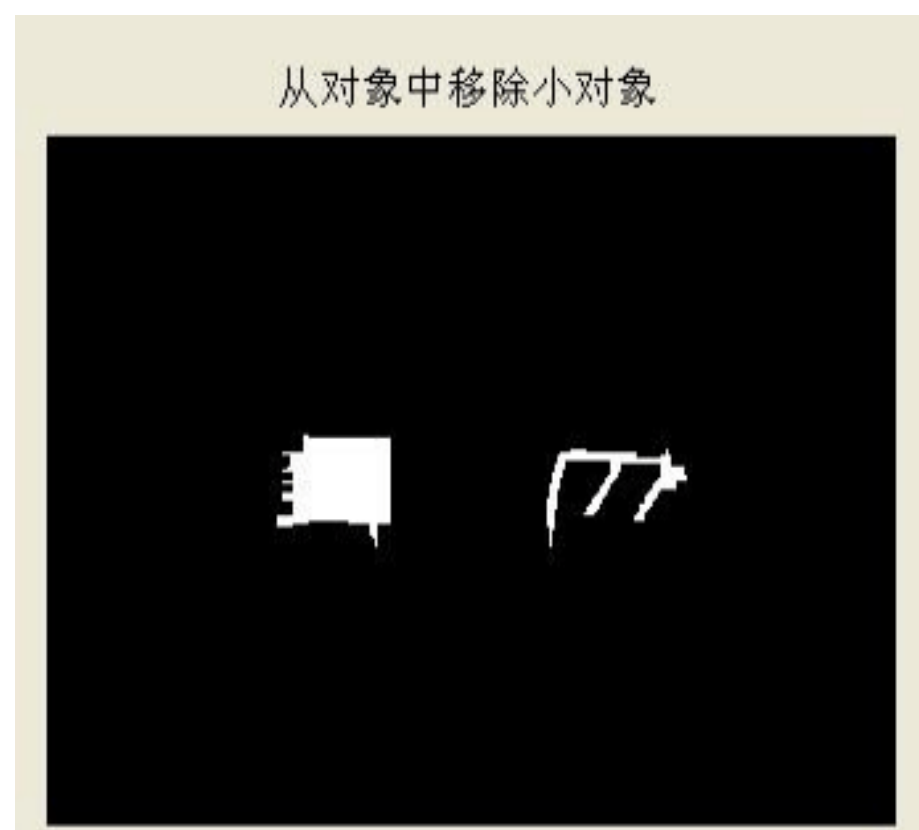


图 3.9



图 3.10

由于本系统设计只限用于车牌底色为蓝色的车牌，对于其他颜色如黄色、白色等为底色的车牌则不成功。所以本系统具有一定的局限性。下面是分割后的车牌图：



图 3.11

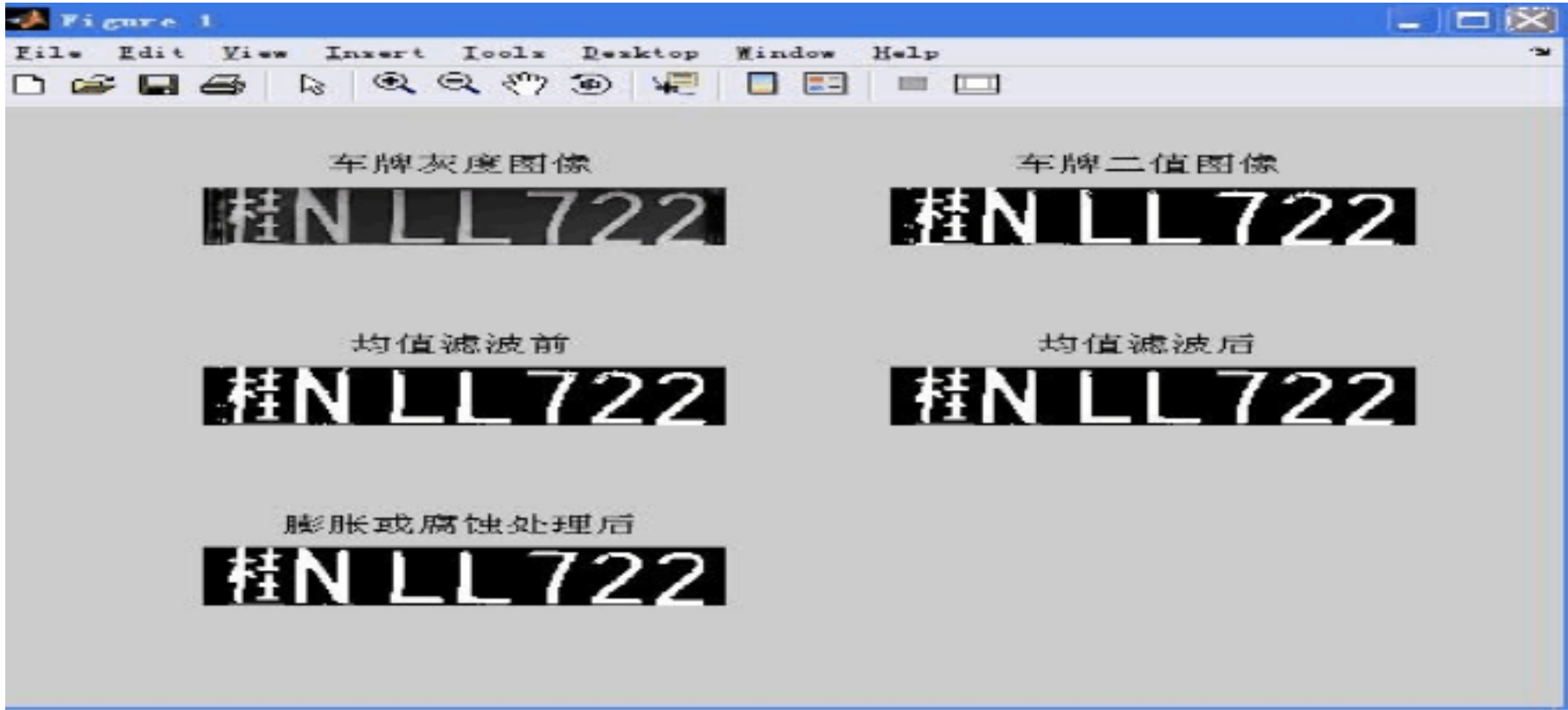


图 3.12

下图是字符的分割与归一化的流程图：

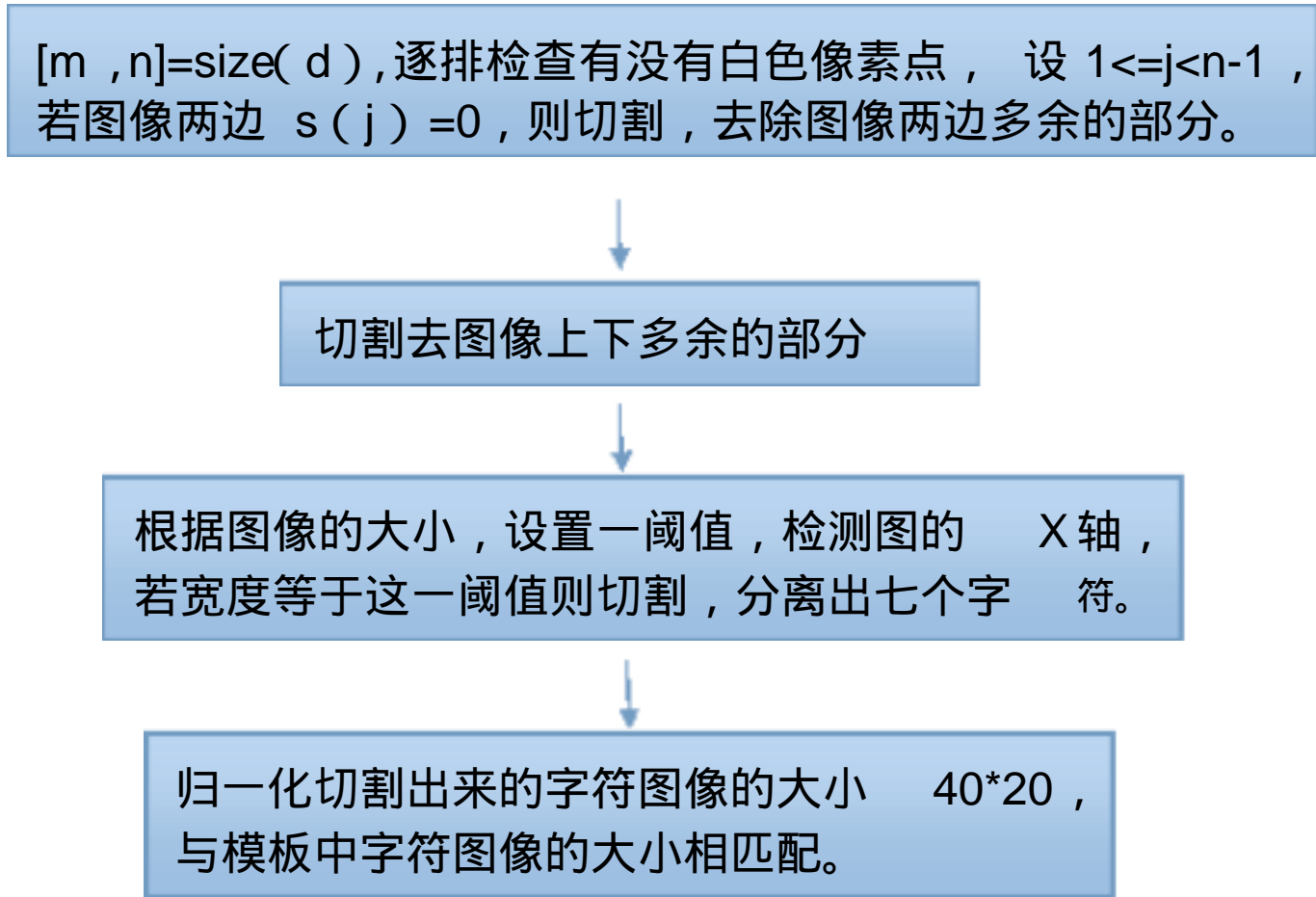




图 3.13

本系统设计采用的方法是寻找连续有文字的块，若长度大于设定的阈值，则认为该块有两个字符组成，需要分割。



图 3.14

分割出来的字符可能存在一定的大小差距，所以要进行进一步的处理，以便接下来的字符识别能准确的识别出字符。本系统设计对分割出来的字符进行了归一化处理，归一化切割出来的字符图像的大小为 40\*20，与模板中字符图像的大小相匹配。

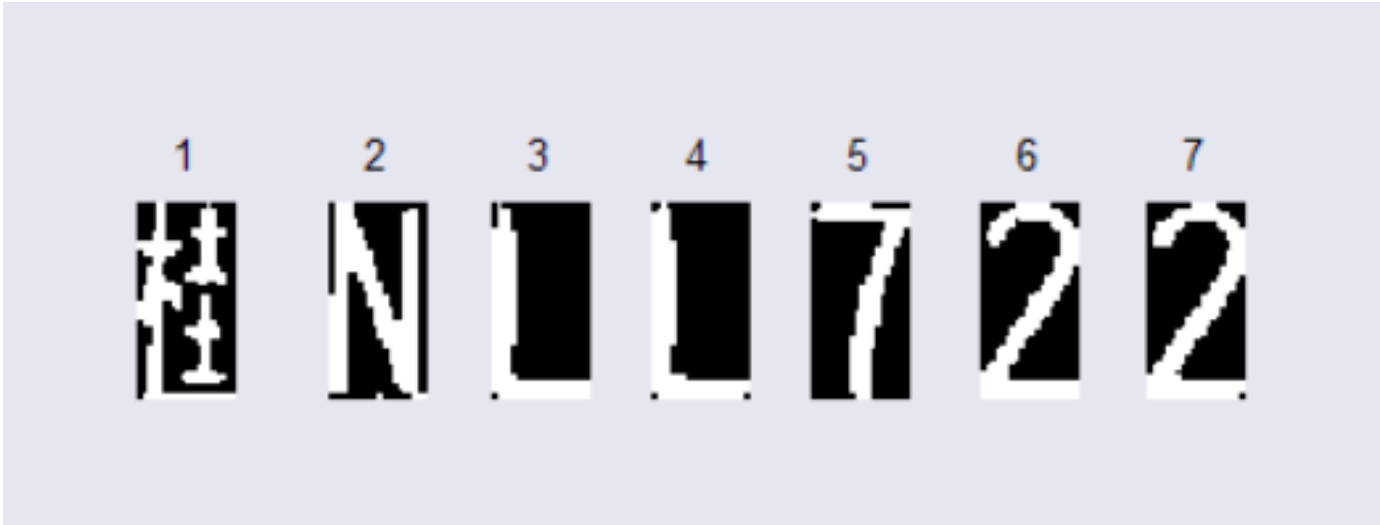


图 3.15

本系统的模板匹配的字符识别基本过程是：首先对待识别字符进行二值化并将其尺寸大小缩放为字符数据库中模板的大小，然后与所有的模板进行匹配，最后选最佳匹配作为结果。模板匹配的主要特点是实现简单，当字符较规整时对字符图像的缺损、污迹干扰适应力强且识别率相当高。流程图如下：

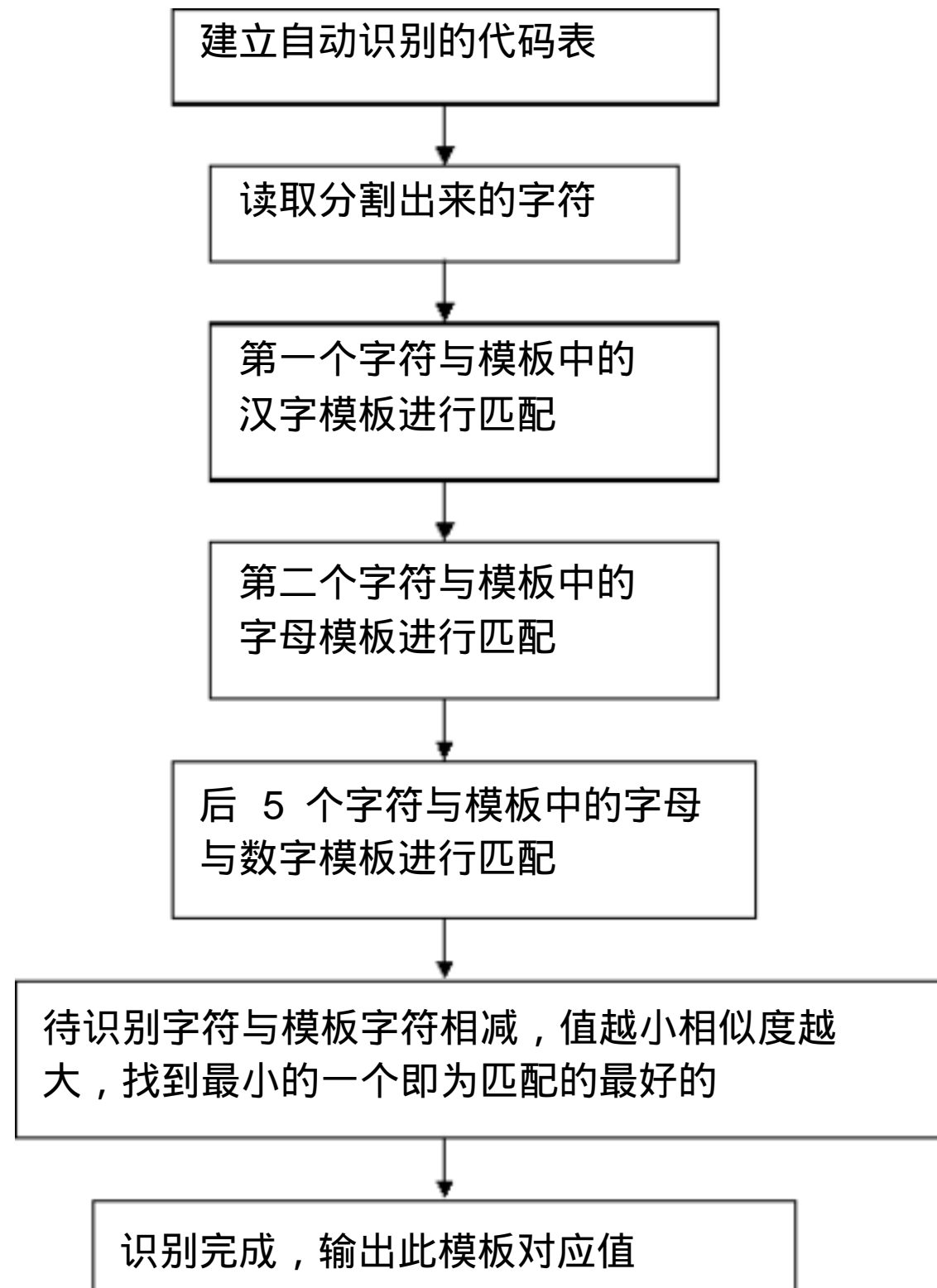


图 3.16

本系统设计采用相减的方法找到出分割字符与模板库中的字符相似度最大的字符，然后输出。汽车号牌的字符一般有七个，大部分车牌第一位是汉字，通常代表车辆所属省份，接着的是字母与数字。车牌字符汉字共约 50 多个，大写英文字母 26 个，数字 10 个。为了提高系统的识别率与实验的方便，本次系统设计只建立了 5 个汉字 26 个字母与 10 个数字的字符模板库。

首先取模板库的字符，接着依次取待识别字符与模板进行匹配，将其与模板库字符相减，得到的 0 越多那么匹配成功率越高。把每一幅相减后的图的 0 值个数保存，即为识别出来的结果并将结果保存在 Data.xls 文件中。字符识别图如下：





图 3.17

四、优化后实现



图 3.18



图 3.19

## 五、 分析总结

汽车号牌识别系统是一个复杂的系统，考虑到时间和本人能力等因素，在这里我只做了一些初步的研究，很不够完善的地方，还需进一步的研究改进。

- (1) 汽车号牌识别系统是针对车牌为蓝底白字，7 个字符水平排列的汽车车牌进行研究。有些光照条件不理想的图片，需要先进行图象增强处理，让图象灰度动态范围扩展和对比度增强，再进行车牌定位和分割，这样可以提高分割的成功率。色彩通道的车牌区域分割算法充分利用了车牌图象的色彩信息，简化了算法的实现，加快了图象的处理速度，具有较高的正确率，而且整个程序用 MATLAB 语言编程实现，运算速度快。但是也存在一些识别效果不是很理想的图片，这些图片需要做一些前提工作后才能识别出相应的字符。
- (2) 车牌定位和分割中利用的车牌区域的宽度信息以及字符尺寸信息，是根据采集到的车辆图像通过人工或者经验测算出来的，实际中需要自动检测；
- (3) 由于基于寻找连续有文字的块的字符分割方法容易受噪声和环境光线变化的影响，所以在车牌字符分割的预处理中，需要对分割出的字符车牌进行均值滤波，膨胀或腐蚀的处理。经过这些处理可以把字符与字符之间的杂色点去除，只有白色的字符和黑色的背景存在，有利于的字符分割进行<sup>[19]</sup>。
- (4) 字符识别方法中运用模板匹配的方法，方法简洁但识别率较低。模板匹配法，是利用要识别的字符图片与字符库中的图片进行两幅图片相减的方法，找到相减后值最小的图片，其相似程度最大的。模板库的字符制作很重要，必须要用精确的模板，否则就不能正确的识别。

改进方法：

- (1) 在现有的基础上利用水平投影的方法检测非横向排列的 7 个字符车牌，根据车牌在水平方向上的投影的峰值特征判断牌照所属类型。
- (2) 在已定位车牌基础上检测牌照字符笔画的宽度，通过投影检测字符的尺寸信息；也可以通过摄像机架设的高度、角度与车道的关系，将这些参数作为系统参数进。
- (3) 改进字符分割的算法，在车牌定位以后对图像进行去噪处理，或者可以通过照明设备照射车辆，由于车辆牌照区域的反光特性，可以大大改善图像采集的质量，同时突出牌照区域，间接达到减少噪声的效果。
- (4) 可以考虑单独设备汉字识别器以及数字字母合一的识别来提高识别器的识别率！

## 附录：程序清单

### 1、车牌分割程序：

```
function
[PY2,PY1,PX2,PX1]=chepai_fenge(I5)
[y,x,z]=size(I5);
myl=double(I5);
tic
Y_threshlow=5; %这个数值很重要。决定了
提取的彩图的质量
X_firrectify=5;
%===== Y方向
=====
Blue_y=zeros(y,1);
for i=1:y
    for j=1:x
        if(myl(i,j,1)==1)
            Blue_y(i,1)=
Blue_y(i,1)+1; % 蓝色像素点统计
        end
    end
end
[temp MaxY]=max(Blue_y); % Y方
向车牌区域确定 temp( 最多点数 ): 所有行
中, 最多的累积像素点 MaxY ( 最多点所在
行 ): 该行中蓝点最多
PY1=MaxY; %有最多蓝点的行付给
PY1
while
((Blue_y(PY1,1)>=Y_threshlow)&&(PY1>1
)) %找到图片上边界
    PY1=PY1-1;
end
PY2=MaxY; %PY1: 存储车牌上边界
值
while
((Blue_y(PY2,1)>=Y_threshlow)&&(PY2<y
)) %阈值为 5
    PY2=PY2+1;
end
PY1, PY2
figure(1),imshow(Blue_y),title('Y
方向确定 ');
pause(2);
```

```
%=====X方向
=====
X_threshhigh=(PY2-PY1)/11; %这
个数值很重要。决定了提取的彩图的质量,
适当提高可抗干扰, 但是小图会照成剪裁太
多
Blue_x=zeros(1,x); % 进一
步确定 X方向的车牌区域
for j=1:x
    for i=PY1:PY2
        if(myl(i,j,1)==1)
            Blue_x(1,j)=
Blue_x(1,j)+1;
        end
    end
end
[temp MaxX]=max(Blue_x);
PX1=MaxX-6*(PY2-PY1);
if PX1<=1
    PX1=1;
end
while
((Blue_x(1,PX1)<=X_threshhigh)&&(PX1<
x)) %阈值
    PX1=PX1+1;
end %确定出 X方
向车牌起点
PX2=MaxX+6*(PY2-PY1);
if PX2>=x
    PX2=x;
end
while
((Blue_x(1,PX2)<=X_threshhigh)&&(PX2>
PX1)) %阈值
    PX2=PX2-1;
end %确定出 X方向车
牌终点
PX1, PX2
figure(2),imshow(Blue_x),title('X
方向
确定 ');
```

## 2、车牌校正程序：

```
function
[PY2,PY1,PX2,PX1,threshold]=chepai_xi
uzheng(PY2,PY1,PX2,PX1)
S=(PY2-PY1)*(PX2-PX1)
if S<=25000
    threshold=50;
    Y_secrectify=3;
    X_secrectify=3;
elseif S>25000&&S<=45000
    threshold=100;
    Y_secrectify=-3;
    X_secrectify=3;
elseif S>45000&&S<=80000
    threshold=200;
    Y_secrectify=-3;
    X_secrectify=3;
elseif S>80000&&S<=150000
    threshold=300;
    Y_secrectify=-10;
    X_secrectify=-10;
elseif S>150000&&S<=400000
    threshold=600;
    Y_secrectify=-10;
    X_secrectify=-10;
else
    threshold=1800;
    Y_secrectify=-10;
    X_secrectify=-10;
end
PY1=PY1-Y_secrectify;
PY2=PY2+Y_secrectify;
PX1=PX1;
PX2=PX2;%对车牌区域的进一步修正
```

## 3、字符分割程序：

```
function e=zifufenge(d)
[m,n]=size(d);
top=1;bottom=m;left=1;right=n;
while sum(d(top,:))==0 && top<=m
    top=top+1;
```

```
end
while sum(d(bottom,:))==0 && bottom>=1
    bottom=bottom-1;
end
while sum(d(:,left))==0 && left<=n
    left=left+1;
end
while sum(d(:,right))==0 && right>=1
    right=right-1;
end
dd=right-left;
hh=bottom-top;
e=imcrop(d,[left top dd hh]);
```

## 4、获取字符程序：

```
function [word,result]=getword(d) %获取
字符
word=[];flag=0;y1=8;y2=0.5;
while flag==0
    [m,n]=size(d);
    wide=0;
    while sum(d(:,wide+1))~=0 &&
wide<=n-2
        wide=wide+1;
    end
    temp=zifufenge(imcrop(d,[1 1
wide m]));
    [m1,n1]=size(temp);
    if wide<y1 && n1/m1>y2
        d(:,[1:wide])=0;
        if sum(sum(d))~=0
            d=zifufenge(d); % 切
割出最小范围
        else word=[];flag=1;
        end
    else
```

```
word=zifufenge(imcrop(d,[1 1 wide m]));
    d(:,[1:wide])=0;
    if sum(sum(d))~=0;
d=zifufenge(d);flag=1;
    else d=[];
```

```

        end
    end
end
result=d;

```

##### 5、主函数程序代码：

```

function varargout = main(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name' ,
    'gui_Singleton' ,
    gui_Singleton, ...
    'gui_OpeningFcn' ,
    @main_OpeningFcn, ...
    'gui_OutputFcn' ,
    @main_OutputFcn, ...
    'gui_LayoutFcn' ,
    [] , ...
    'gui_Callback' ,
    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
pause(1);
function main_OpeningFcn(hObject, eventdata, handles, varargin)
set(handles.process, 'enable','on' )
handles.output = hObject;
guidata(hObject, handles);
function varargout = main_OutputFcn(hObject, eventdata, handles)
tic
varargout{1} = handles.output;

```

```

function pushbutton1_Callback(hObject, eventdata, handles)
[filename
pathname]=uigetfile({'*.jpg'; '*.bmp' },
'File Selector' );
I = imread([pathname '\ ' filename]);
handles.I = I;
guidata(hObject, handles);
axes(handles.axes1);
imshow(I);title(' 原始图片 ' )
set(handles.process, 'enable' , 'on' )
function process_Callback(hObject, eventdata, handles)
I = handles.I;
I1=rgb2gray(I);
guidata(hObject, handles);
axes(handles.axes2);
imshow(I1);title(' 灰度图 ' );
axes(handles.axes3);
imhist(I1);title(' 灰度图直方图 ' );
pause(2);
I2=edge(I1,'robert',0.15,'both');
guidata(hObject, handles);
axes(handles.axes2);
imshow(I2);title('robert 算子边缘检测 ' );
pause(2);
se=[1;1;1];
I3=imerode(I2,se);
guidata(hObject, handles);
axes(handles.axes3);
imshow(I3);title(' 腐蚀后图像 ' );
pause(2);
se=strel('rectangle',[10,25]);
I4=imclose(I3,se);
guidata(hObject, handles);
axes(handles.axes2);
imshow(I4);title(' 平滑图像的轮廓 ' );
pause(2);
I5=bwareaopen(I4,2000);
guidata(hObject, handles);
axes(handles.axes2);
imshow(I5);title(' 从对象中移除小对象 ' );

```

```

pause(2);
[PY2,PY1,PX2,PX1]=chepai_fenge(I5); %
调用分割车牌
global threshold;
[PY2,PY1,PX2,PX1,threshold]=chepai_xi
uzheng(PY2,PY1,PX2,PX1); %调用车牌校正
IY=I(PY1:PY2,:,:);
Plate=I5(PY1:PY2,PX1:PX2);
global dw;
dw=Plate;
PX1=PX1-1;%对车牌区域的校正
PX2=PX2+1;
dw=I(PY1:PY2-8,PX1:PX2,:);
axes(handles.axes2);
imshow(dw),title( ' 车牌区域的校正 ');
pause(2);
t=tic;
guidata(hObject, handles);
axes(handles.axes2);
imshow(IY),title( ' 水平方向合理区域 ');
axes(handles.axes3);
imshow(dw),title( ' 定位剪切后的彩色车
牌图像 ');
pause(2);
imwrite(dw, 'New number plate.jpg' );
[filename,filepath]=uigetfile( 'New
number plate.jpg' , ' 输入一个定位裁剪后
的车牌图像 ');
jpg=strcat(filepath,filename);
a=imread( 'New number plate.jpg' );
b=rgb2gray(a);
figure(3),subplot(3,2,1),imshow(b),ti
tle( ' 车牌灰度图像 ');
g_max=double(max(max(b)));
g_min=double(min(min(b)));
T=round(g_max-(g_max-g_min)/2); %T为
二值化的阈值
[m,n]=size(b);
d=(double(b)>=T); % d: 二值图像
figure(3),subplot(3,2,2),imshow(d),ti
tle( ' 车牌二值图像 ');
figure(3),subplot(3,2,3),imshow(d),ti
tle( ' 均值滤波前 ');
pause(1);

```

```

h=fspecial( 'average', 3); % 滤波
d=im2bw(round(filter2(h,d)));
figure(3),subplot(3,2,4),imshow(d),ti
tle( ' 均值滤波后 ');
se=eye(2); % eye(n) returns the n-by-n
identity matrix 单位矩阵
[m,n]=size(d);
if bwarea(d)/m/n>=0.365
    d=imerode(d,se);
elseif bwarea(d)/m/n<=0.235
    d=imdilate(d,se);
end
figure(3),subplot(3,2,5),imshow(d),ti
tle( ' 膨胀或腐蚀处理后 ');
pause(2);
%寻找连续有文字的块， 若长度大于某阈值，
则认为该块有两个字符组成，需要分割
d=zifufenge(d);
[m,n]=size(d);
guidata(hObject, handles);
axes(handles.axes3);imshow(d);
k1=1;k2=1;s=sum(d);j=1;
while j~=n
    while s(j)==0
        j=j+1;
    end
    k1=j;
    while s(j)~=0 && j<=n-1
        j=j+1;
    end
    k2=j-1;
    if k2-k1>=round(n/6.5)
        [val,num]=min(sum(d(:,[k1+5:k2-5])));
        d(:,k1+num+5)=0 % 分割
    end
end
d=zifufenge(d); % 再分割
% 切割出 7 个字符
y1=10;y2=0.25;flag=0;word1=[];
while flag==0
    [m,n]=size(d);
    left=1;wide=0;
    while sum(d(:,wide+1))~=0

```

```

        wide=wide+1;
    end
    if wide<y1      % 认为是左侧干扰
        d(:,[1:wide])=0;
        d=zifufenge(d);
    else
        temp=zifufenge(imcrop(d,[1 1
wide m]));
        [m,n]=size(temp);
        all=sum(sum(temp));

two_thirds=sum(sum(temp([round(m/3):2
*round(m/3)],:)));
        if two_thirds/all>y2
            flag=1;word1=temp;
        end
        d(:,[1:wide])=0;
        d=zifufenge(d);
    end
end
[word2,d]=getword(d);      % 分割出第二个
字符
pause(1);
[word3,d]=getword(d);
pause(1);
[word4,d]=getword(d);
pause(1);
[word5,d]=getword(d);
pause(1);
[word6,d]=getword(d);
pause(1);
[word7,d]=getword(d);      % 分割出第七个
字符
pause(1);
guidata(hObject, handles);
axes(handles.axes4);imshow(word1),tit
le( '1' );
guidata(hObject, handles);
axes(handles.axes4);imshow(word2),tit
le( '2' );
guidata(hObject, handles);
axes(handles.axes4);imshow(word3),tit
le( '3' );
guidata(hObject, handles);

```

```

axes(handles.axes4);imshow(word4),tit
le( '4' );
guidata(hObject, handles);
axes(handles.axes4);imshow(word5),tit
le( '5' );
guidata(hObject, handles);
axes(handles.axes4);imshow(word6),tit
le( '6' );
guidata(hObject, handles);
axes(handles.axes4);imshow(word7),tit
le( '7' );
[m,n]=size(word1);
word1=imresize(word1,[40 20]);      % 商用
系统程序中归一化大小为 40*20, 此处演示
word2=imresize(word2,[40 20]);
word3=imresize(word3,[40 20]);
word4=imresize(word4,[40 20]);
word5=imresize(word5,[40 20]);
word6=imresize(word6,[40 20]);
word7=imresize(word7,[40 20]);
guidata(hObject, handles);
axes(handles.axes4);imshow(word1),tit
le( '1' );
guidata(hObject, handles);
axes(handles.axes5);imshow(word2),tit
le( '2' );
guidata(hObject, handles);
axes(handles.axes6);imshow(word3),tit
le( '3' );
guidata(hObject, handles);
axes(handles.axes7);imshow(word4),tit
le( '4' );
guidata(hObject, handles);
axes(handles.axes8);imshow(word5),tit
le( '5' );
guidata(hObject, handles);
axes(handles.axes9);imshow(word6),tit
le( '6' );
guidata(hObject, handles);
axes(handles.axes10);imshow(word7),ti
tle( '7' );
imwrite(word1, '1.jpg' );
imwrite(word2, '2.jpg' );
imwrite(word3, '3.jpg' );

```





```
cla(handles.axes10);
```