

《数字图像处理》

实验指导书

武汉理工大学

信息工程学院

2021 年 1 月

实验 1 图像采集及图像变换

实验 2 图像增强

实验 3 图像压缩

实验 4 车牌检测及识别

实验 1 图像采集及图像变换

1.1 实验目的

- (1) 掌握运用 MV-EM 系列千兆网工业相机和相关软件进行图像采集。
- (2) 掌握图像变换的方法，熟悉图像傅立叶变换，观察图像变换的效果。

1.2 实验内容

- (1) 利用 MV-EM 系列千兆网工业相机进行图像采集
- (2) 快速卷积实验，对图像进行傅立叶变换的 Matlab 编程实现，验证卷积定理。

1.3 实验方法

利用维视智造 MV-EM 系列千兆网工业相机图像采集使用方法，利用 MATLAB 工具和图像处理工具箱进行实验。

1.4 实验过程及参考程序

(一) 利用 MV-EM 系列千兆网工业相机进行图像采集

此部分的详细操作说明参见《MV-E EM 系列千兆以太网工业相机快速使用说明 V1.0》。其主要 4 步如下。

Step1 检查相机与主机的连接线是否连接好。

有 2 根线。1 根是相机通过电源适配器连接到电源；另 1 根是相机与主机相连的信号线，连接到主机的 LAN 口 1 上，属于 RJ45 接口。

Step2 驱动程序与软件安装。

运行光盘目录下的安装程序 MV_SDK_x64_v2.0.exe，按照提示进行安装，安装完毕后重启电脑。如果此程序已经安装，则无需再次安装。

Step3 网卡及 IP 设置。

- (1) 以管理员权限运行桌面上程序[NIC_conf]，检查网卡状态是否正常。

如出现红色字，表示不正常，需要点击下方对应按钮进行设置，选“批量设置网卡 IP”选项。可参考下面的图 1.1 的参数设置。

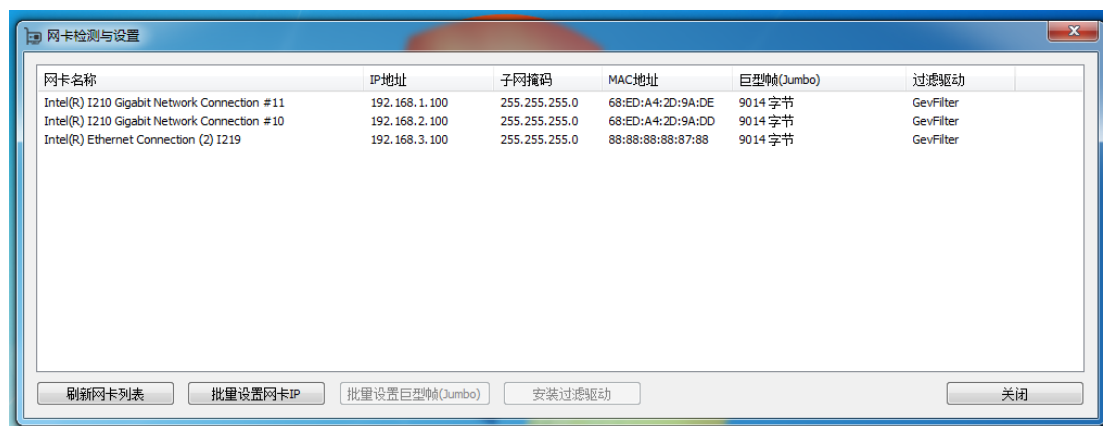


图 1.1 网卡配置信息

- (2) 设置网卡 IP，使网卡 IP 与相机 IP 处于同一网段。

相机默认的 IP 为 192.168.1.200，设置方法为：打开”网络和共享中心“—”本地连接“—”属性“，双击”Internet 协议版本 4（TCP/IPV4）“，比如，可以设置如下地址：

IP 地址（I）：192.168.3.100
子网掩码（U）：255.255.255.0
DNS 服务器地址（E）：空

（3）运行桌面程序[Camconf]，确认相机与网卡处于同一网段。
利用此工具修改相机 IP，使得相机与网卡处于同一网段上。比如可以将相机 IP 地址设置为 192.168.3.200，这样相机 IP 与网卡 IP 处于同一网段了。如下图 1.2 所示。



图 1.2 相机 IP 配置信息

Step4 利用相机的 Demo 软件进行图像采集。

运行桌面程序[MV_Demo]，可以进行图像或视频的采集和保存。此部分的详细操作说明参见《MV-VS1000（MATLAB）实验指导书 v3.0.1910》的”第三章 如何利用 MATLAB 采集图像“。

（二）图像傅立叶变换实验

1. 理论基础

在图像处理的广泛应用领域中，傅立叶变换起着非常重要的作用，具体表现在图像分析，图像增强和图像压缩方面。

设 $f(m,n)$ 是个离散空间中的二维函数，则该函数的二维傅立叶变换的定义如下：

$$F(\omega_1, \omega_2) = \int_{m=-\infty}^{+\infty} \int_{n=-\infty}^{+\infty} f(m,n) e^{-j\omega_1 n} e^{-j\omega_2 n} dm dn$$

其中 ω_1, ω_2 是频谱变量，单位是弧度/采样单元，周期都是 2π

二维傅立叶反变换的定义如下：

$$f(m,n) = \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} F(\omega_1, \omega_2) e^{j\omega_1 n} e^{j\omega_2 m} d\omega_1 d\omega_2$$

*利用计算机进行傅立叶变换的通常形式为离散傅立叶变换，采用这种形式的傅立叶变换，主要是由于：离散傅立叶变换的输入输出都是离散值，适合用计算机运算操作，并且可以提高运算速度。

离散傅立叶变换的定义如下：

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j2\pi(ux/M + vy/N)]$$

二维傅立叶反变换公式如下：

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp[j2\pi(ux/M + vy/N)]$$

其中 $x = 0, 1, \dots, M-1; y = 0, 1, \dots, N-1$

通常情况下，数字图像处理用像素方阵表示，即 $M=N$ ，此时二维傅立叶变换可简化为：

$$F(u,v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j2\pi(ux + vy)/N] \quad x, y = 0, 1, \dots, N-1$$

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \exp[j2\pi(ux + vy)/N] \quad u, v = 0, 1, \dots, N-1$$

2. MATLAB 中提供的变换函数

(1) fft2:用于计算二维快速傅立叶变换，语句格式：

$B = \text{fft2}(I, m, n)$

按指定的点数计算 m ，返回矩阵 B 的大小为 $m \times n$ ，不写默认为原图像大小。

(2) fftn:用于计算 n 维快速傅立叶变换

(3) fftshift:用于将变换后的图像频谱中心从矩阵的原点移到矩阵地中心，语法格式：

$B = \text{fftshift}(I)$

(4) ifft2:用于计算图像的二维傅立叶反变换，语法格式

$B = \text{ifft2}(i)$

(5) ifftn:用于计算 n 维傅立叶变换

3. 快速卷积实验：傅立叶变换一个重要特性是可以实现快速卷积。

设 A 为 $M \times N$ 矩阵， B 为 $P \times Q$ 的矩阵，快速卷积方法如下：

*对 A 和 B 补 0，使其大小都为 $(M+P-1) \times (N+Q-1)$

*利用 fft2 对矩阵 A 和 B 进行二维变换

*将两个 FFT 结果相乘，利用 ifft2 对得到的乘积进行傅立叶反变换

现在令 $A = \text{magic}(3)$ ， $B = \text{ones}(3)$ ，请按上述步骤自行完成实验。

参考如下：

启动 MATLAB，在命令窗口中输入

```
A=magic(3);  
B=ones(3);  
A(8,8)=0;  
B(8,8)=0;  
C=ifft2(fft2(A).*fft2(B));  
C=C(1:5,1:5);  
C=real(C)
```

(1) 观察运行结果，并记录。

(2) 调用 MATLAB 提供的卷积函数 conv2 进行验证，对比结果是否正确

```
C=conv2(A,B);
```

```
C=C(1:5,1:5)
```

运行并对比记录结果。

4. 图像的二维傅立叶变换实验

对一副灰度图片进行二维傅立叶变换，并且显示其幅度值

(1) 启动 MATLAB，设置当前工作路径

(2) 在当前路径下的文件夹中拷入个灰度图片，比如“ex2.jpg”文件。

(3) 输入如下命令，完成实验

```
figure(1);  
RGB=imread('ex2.jpg');  
imshow(RGB);  
figure(2);  
B=fftshift(fft2(RGB));  
imshow(log(abs(B)),[]),colormap(jet(64)),colorbar;
```

(4) 观察结果，并记录。

(5) 对程序的语句给出注释说明。

1.5 思考题

(1) 如何调整焦距、物距、光源，得到需要的清晰图像。

(2) 并独立完成以下实验：

创建一副图像，大小为 128*128，背景为黑色，中间开出一个 8×8 白色的窗口；运用 ff2 函数，对上面产生的图像做二维离散傅立叶变换。

1.6 实验报告

(1) 内容包括实验目的，实验内容，实验过程，实验结果，实验结果分析，心得体会。

(2) 要求有较详细的图文说明。

实验 2 图像增强

2.1 实验目的

掌握图像空间域增强和频率域增强方法。掌握如何用 matlab 工具分别实现图像空间域增强和频率域增强。

2.2 实验内容

掌握图像增强原理，用 matlab 工具设计实现图像空间域和频率域增强程序，观察图像增强的效果，并对结果进行分析。

2.3 实验方法

用相机采集图像，用 Matlab 编程实现图像空间域增强和频率域增强。

2.4 实验过程及参考程序

(一) 利用 EM 系列相机进行图像采集。

操作与实验 1 的“(一)利用 MV-EM 系列千兆网工业相机进行图像采集”相同。

(二) 空间域增强

1. 空间域灰度变换增强

增强对比度是增强原图像的各部分的反差。实际中往往是通过改变原图中某两个灰度值之间的动态范围来实现的（如图 2.1）。

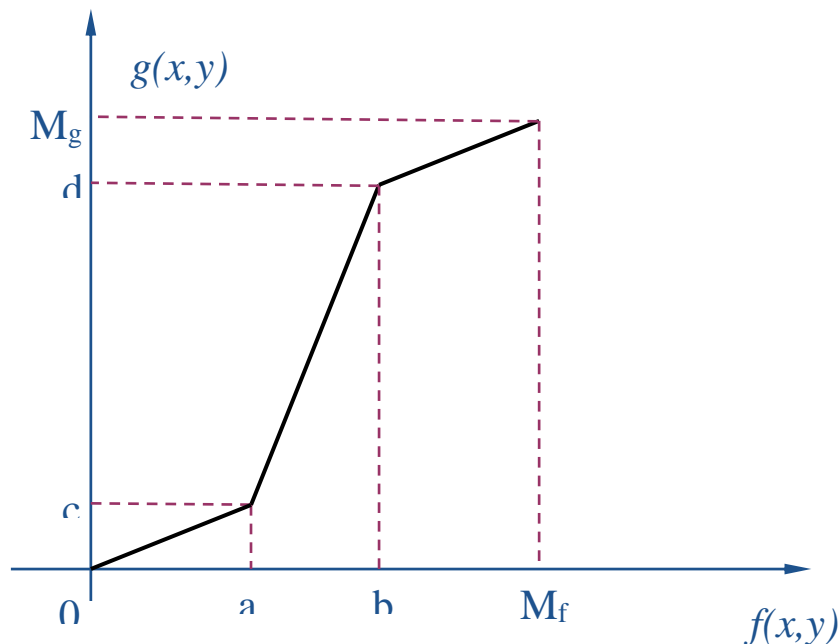


图 2.1 增强对比度

从图 2.1 中可以看出，通过变换可以减少原图的较高 ($f(x, y) > b$) 的和较低 ($f(x, y) < a$) 的灰度值的动态范围，而增大在二者之间 ($a < f(x, y) < b$) 的动态范围，从而使其在指定范围的对比度增加了。可以写成如下表达式，

其中 $M_f=M_g=255$ 。

$$g(x, y) = \begin{cases} \frac{M_g - d}{M_f - b} [f(x, y) - b] + d & b \leq f(x, y) \leq M_f \\ \frac{d - c}{b - a} [f(x, y) - a] + c & a \leq f(x, y) < b \\ \frac{c}{a} f(x, y) & 0 \leq f(x, y) < a \end{cases}$$

问题的关键是根据实际图像，确定需要增大和减少的图像的灰度级范围，即确定对应的参数 a ， b ， c ， d 。

利用 MATLAB 进行编程，参考的 MATLAB 代码：

```
X1=imread('image.tif');
```

```
figure, imshow(X1)
```

```
f0=0;g0=0;
```

```
f1=70;g1=30;
```

```
f2=180;g2=230;
```

```
f3=255;g3=255;
```

```
r1=(g1-g0)/(f1-f0);
```

```
b1=g0-r1*f0;
```

```
r2=(g2-g1)/(f2-f1);
```

```
b2=g1-r2*f1;
```

```
r3=(g3-g2)/(f3-f2);
```

```
b3=g2-r3*f2;
```

```
[m,n]=size(X1);
```

```
X2=double(X1);
```

```
for i=1:m
```

```
    for j=1:n
```

```
        f=X2(i, j);
```

```
        g(i, j)=0;
```

```
        if (f>=0)&(f<=f1)
```

```
            g(i, j)=r1*f+b1;
```



```

elseif (f>=f1)&(f<=f2)
    g(i,j)=r2*f+b2;
elseif (f>=f2)&(f<=f3)
    g(i,j)=r3*f+b3;
end
end
end
figure,imshow(mat2gray(g))

```

这是根据分段线性变换表达式，用 **Matlab** 环境编写的程序。实际问题中需要根据具体的图像特点，设计不同的线性变换。比如选择一幅灰度级比较狭窄的图像，设计一个灰度变换函数，确定参数 a , b , c , d 的值，说明确定参数 a , b , c , d 的值的依据；写出对应的 **Matlab** 程序对其进行灰度变换，比较变换前后的效果，并进行分析。

2. 空间域平滑滤波器

(1) 平滑滤波器属于低通滤波器，可滤除掉灰度值具有较大较快变化的部分。

(2) 模拟噪声生成函数 `imnoise`，它可以对图像添加一些典型噪声。

语法格式：`J=imnoise(I,type)` $type$ 为噪声类型，如高斯噪声 ‘gaussian’，椒盐噪声 ‘salt & pepper’ 等。

(3) 均值滤波（线性滤波）的实现。

(a) 在当前工作文件夹中拷入个图片，比如 “lena.jpg” 文件。

(b) 输入如下命令完成实验

```

I=imread('lena.jpg');
I=imnoise(I,'gaussian',0,0.02); %添加高斯噪声
imshow(I);
h=[1 1 1
   1 1 1
   1 1 1]; %产生均值滤波模板,
           %可以采用不同的均值滤波模板
h=h/9; %滤波模板归一化
J=conv2(I,h); %滤波
figure,imshow(J,[])

```

(c) 观察并记录结果，说明该程序的功能。

(4) 以上都是用 3×3 的模板对噪声进行滤波，现再自行设计一个 5×5 或 7×7 的模板进行处理，比较不同模板造成的差别。

（三）频率域图像增强

1. 基本原理

卷积理论是频域技术的基础。设函数 $f(x, y)$ 与线性移不变算子 $h(x, y)$ 的卷积结果是 $g(x, y)$ ，即 $g(x, y) = h(x, y) * f(x, y)$ 。

那么根据卷积定理在频域有：

$$G(u, v) = H(u, v)F(u, v)$$

其中 $G(u, v)$ 、 $H(u, v)$ 、 $F(u, v)$ 分别是 $g(x, y)$ 、 $h(x, y)$ 、 $f(x, y)$ 的傅立叶变换。

频域滤波增强的主要步骤是：

- （1）对图像的傅立叶变换；
- （2）将其与一个（根据需要设计的）转移函数相乘；
- （3）再将结果进行傅立叶反变换以得到增强的图像。

频域增强的两个关键步骤：

- （1）将图像从空域转换到频域所需的变换及将图像从频域空间转换回空域所需的变换；
- （2）在频域空间对图像进行滤波增强的处理操作。

2. 低通滤波

图像的能量大部分集中在幅度谱的低频和中频度，而图像的边缘和噪声对应于高频部分。因此能降低高频成分幅度的滤波器就能减弱噪声的影响。

Butterworth 低通滤波器是一种物理上可以实现的低通滤波器， n 阶、截断频率为 d_0 的 Butterworth 低通滤波器的转移函数为：

$$H(u, v) = \frac{1}{1 + [d(u, v)/d_0]^{2n}}$$

用 MATLAB 实现对图像加入“高斯”噪声后进行 Butterworth 低通滤波器的参考代码如下所示：

```
%读入图像
I1=imread('lena.bmp');
figure, imshow(I1)
%加入高斯噪声
```

```

I2=imnoise(I1,'gaussian',0,0.002);
figure,imshow(I2)
%对图像作傅里叶变换
f=double(I2);
g=fft2(f);
g=fftshift(g);
[N1,N2]=size(g);
%生成 Butterworth 低通滤波器，对图像进行滤波
n=2;
d0=50;
n1=fix(N1/2);
n2=fix(N2/2);
for i=1:N1
    for j=1:N2
        d=sqrt((i-n1)^2+(j-n2)^2);
        h=1/(1+(d/d0)^(2*n));
        result(i,j)=h*g(i,j);
    end
end
%作傅里叶逆变换，将图像变换回到空间域
result=ifftshift(result);
X2=ifft2(result);
X3=uint8(real(X2));
figure,imshow(X3)

```

选择一幅灰度图像，对其加入“高斯”噪声后，利用频域方法对其进行滤波。先对加噪声后的图像进行 Fourier 变换，设计一个滤波传递函数（说明选取滤波函数及其参数的理由），参考上述程序对其进行滤波，再作逆变换，比较处理前后的效果，并进行分析。

2.5 思考题

- (1) 对参考程序给出功能注释。
- (2) 对不同的图像，如何选择灰度变换函数实现图像的增强处理？即如何确定灰度变换表达式中参数 a ， b ， c ， d 的值。
- (3) 实现图像的空间域锐化滤波处理。
- (4) 实现图像的高斯低通滤波、Butterworth 高通滤波、高斯高通滤波处理。

2.6 实验报告

- (1) 内容包括实验目的，实验内容，实验过程，实验结果，实验结果分析，心得体会。
- (2) 要求有较详细的图文说明。

实验 3 图像压缩

3.1 实验目的

掌握图像压缩编码的方法，掌握用 Matlab 工具编程实现图像的压缩编码。

3.2 实验内容

掌握图像压缩编码原理，用 matlab 工具设计实现图像的 DCT 压缩编码程序，观察图像压缩的效果，并对结果进行分析。

3.3 实验方法

用相机采集图像，用 Matlab 编程实现图像 DCT 压缩编码。

3.4 实验过程及参考程序

(一) 利用 EM 系列相机进行图像采集。

操作与实验 1 的“(一)利用 MV-EM 系列千兆网工业相机进行图像采集”相同。

(二) 图像变换编码

1. 图像变换编码原理

图像经过正交变换后，把原来分散在原空间的图像数据在新的坐标空间中得到集中。对于大多数图像，大量变换系数很小，只需要删除接近于零的系数，并且对较小的系数进行粗量化，而保留包含图像主要信息的系数，以此进行压缩编码。在重建图像进行解码（逆变换）时，所损失的将是一些不重要的信息，几乎不会引起图像的失真。图像的变换编码就是利用这些来压缩图像的，这种方法可得到较高的压缩比。

图 3.1 给出了变换编码系统图，从图中可以看出，变换编码并不是一次对整幅图像进行变换和编码，而是将图像分成 $n \times n$ （常用的 n 为 8 或 16）个子图像后分别处理。这是因为：

(1) 小块图像的变换计算容易。

(2) 距离较远的像素之间的相关性比距离较近的像素之间的相关性小。

变换编码首先将一幅 $N \times N$ 大小的图像分割成 $(N/n)^2$ 个子图像。然后对子图像进行变换操作，解除子图像像素间的相关性，达到用少量的变换系数包含尽可能多的图像信息的目的。接下来的量化步骤是有选择地消除或粗量化带有很少信息的变换系数，因为它们对重建图像的质量影响很小。最后是编码，一般用变长码对量化后系数进行编码。解码是编码的逆操作，由于量化是不可逆的，所以在解码中没有对应的模块。其实压缩并不是在变换步骤中取得的，而是在量化变换系数和编码时取得的。

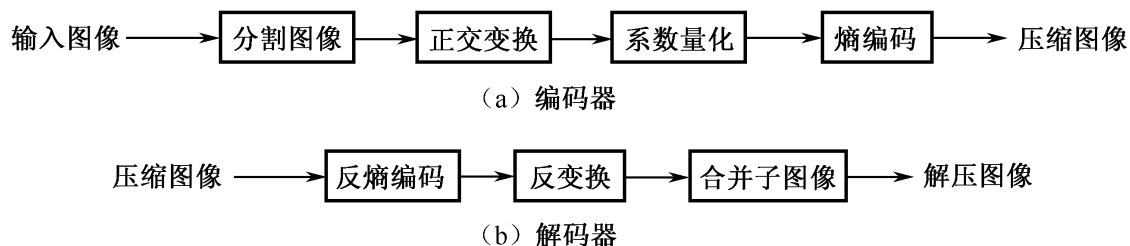


图 3.1 变换编码系统

2. 基于 DCT 的图像压缩编码。

图像的 DCT（离散余弦变换）具有把高度相关数据能量集中的能力，这一点和傅里叶变换相似，且 DCT 得到的变换系数是实数。因此，DCT 广泛用于图像压缩。下面以 1 个实例作为参考，进行说明。

对于 1 个 512*512 的灰度图像 lena 进行 DCT 压缩编码，设置压缩比为 cr。首先将图像分块成 $(512/8) * (512/8)$ 个 8*8 子图像，对每个子图像进行 DCT，这样每个子图像有 64 个变换系数。如果压缩比 cr=0.5，那么就要舍去 50% 小的变换系数，即保留 32 个系数，进行 2:1 的压缩。其参考 Matlab 程序如下：

```
%设置压缩比
cr=0.5;
%读取并显示原图像
I1=imread('lena.bmp');
I1=rgb2gray(I1);
I1=double(I1)/255;
%获取图像的大小，行数 l，列数 c
[l,c]=size(I1);
figure(1);
imshow(I1);
%对图像进行 DCT
t=dctmtx(8);
dctceo=blkproc(I1,[8 8],'P1*x*P2',t,t');
coevar=im2col(dctceo,[8 8],'distinct');
coe=coevar;
[Y,ind]=sort(coevar);
[m,n]=size(coevar);
%舍去不重要的系数
snum=64-64*cr;
for i=1:n
    coe(ind(1:snum),i)=0;
end
b2=col2im(coe,[8 8],[1 c],'distinct');
%对截取后的变换系数进行 DCT 逆变换
I2=blkproc(b2,[8 8],'P1*x*P2',t',t);
%显示经过压缩解压后的图像
figure(2);
imshow(I2);
%计算原图像与经过压缩解压后的图像之间的均方根误差
e=double(I1)-double(I2);
[m,n]=size(e);
erms=sqrt(sum(e(:).^2)/(m*n))
```

上述程序中 cr=0.5，压缩比为 2:1，如果 cr=0.125，则压缩比为 8:1。

我们可以根据实际问题对压缩比的要求来设置 cr 的值。

3.5 思考题

- (1) 当 $cr=0.125$ 时，实现对图像的 DCT 编码，并计算其均方根误差。
- (2) 如何在程序中计算原图像与经过压缩解压后的图像的压缩比，验证其是否等于 cr 。
- (3) 图像的 FFT 压缩编码如何编程实现。
- (4) 了解 JPEG 压缩编码原理及其实现的过程。

3.6 实验报告

- (1) 内容包括实验目的，实验内容，实验过程，实验结果，实验结果分析，心得体会。
- (2) 要求有较详细的图文说明。

实验 4 车牌检测及识别

4.1 实验目的

- (1) 综合利用数字图像处理技术，研究设计一个数字图像处理系统。
- (2) 培养综合利用数字图像处理技术解决实际复杂工程的能力。

4.2 实验内容

设计一个“车牌检测与识别”系统，包括车牌图像采集、彩色图像灰度化、图像滤波去噪和增强、边缘检测与数学形态学处理、车牌区域定位、车牌区域分割、字符分割、字符识别等功能处理模块。

4.3 实验方法

用相机采集图像，用 Matlab 编程实现车牌检测及识别。

4.4 实验过程及参考程序

(一) 车牌图像采集。

利用模仿制作的车牌图像，通过相机进行车牌图像的采集，操作与实验 1 的“（一）利用 MV-EM 系列千兆网工业相机进行图像采集”相同。

(二) 车牌区域定位与分割

1. 车牌图像预处理

预处理包括彩色图像的灰度化、图像滤波去噪和增强处理、边缘检测和数学形态学处理。将彩色图像转换为占用存储空间相对较小的灰度图像，可以使用 matlab 中的 `rgb2gray` 函数进行灰度图的转换。图像滤波去噪和增强处理使得到的图像有适当的亮度、较大的对比度和清晰可辨，用边缘检测算子提取图像的车牌轮廓，用数学形态学的开、闭运算去除无关小对象、填充孔洞，得到车牌区域突出的二值图像，便于车牌区域定位和分割。

2. 车牌区域的定位与分割

其主要目的是在经图像预处理后的原始灰度图像中确定车牌的具体位置，并将包含车牌字符的一块子图像从整个图像中分割出来，分割的准确与否直接关系到整个车牌字符识别系统的识别率。

采用水平与垂直双向投影法，可将车牌的四个边界值确定下来，得到车牌的矩形区域，再将该矩形区域从原图像中裁剪下来。即从原图像中分割出矩形车牌区域。

(三) 车牌识别

1. 车牌区域预处理

对车牌区域图像进行滤波去噪增强和数学形态学处理，以得到字符清晰独立的车牌区域二值图像。

2. 车牌字符分割

将车牌区域图像切割出 7 个独立的字符图像。

3. 车牌字符识别

对切割出来的 7 个字符图像进行分别识别。模板匹配是 1 个计算相对简单运算速度比较快的方法。在模板匹配中，首先建立标准模板库，模板库包含 30 个省直辖市自治区的简称汉字、26 个英文大写字母和 10 个数字字符。库中字符使

用统一的尺寸，同时将待识别的字符也保存为与模板大小一致。对分割出来的字符图像与模板图像进行特征比较，得到识别结果。

主要过程为，读取分割出来的字符，第 1 个字符与模板中的汉字模板进行匹配，第 2 个字符与模板中的字母模板进行匹配，后 5 个字符与模板中的字母与汉字模板进行匹配，识别完毕，输出模此板对应的值，输出的 7 个字符的值即为车牌号码。

（四）程序实现

根据上面的分析过程，用 Matlab 工具编程实现，并对结果进行分析。

4.5 思考题

- （1）分析本实验中用到数字图像处理的哪些技术？
- （2）在车牌检测与分割系统中，有哪些不同的方法？有哪些新技术？

4.6 实验报告

- （1）内容包括实验目的，实验内容，实验过程，实验结果，实验结果分析，心得体会。
- （2）要求有较详细的图文说明。