拓展资源 3.3　实 验 指 导

3.3.1　图像的点运算

1．实验内容

对一灰度图像，通过选择不同的灰度级变换函数 $s=T(r)$ 实现图像的灰度范围线性扩展和非线性扩展，以及图像的灰度倒置和二值化。

2．实验原理

1）线性扩展

灰度级变换函数 $s=T(r)$ 可为

$$s=\begin{cases} d & f(x,y)>b \\ \dfrac{d-c}{b-a}[r-a]+c & a\leqslant f(x,y)\leqslant b \\ c & f(x,y)<a \end{cases} \tag{3.18}$$

灰度级变换函数如图 3.6 所示。

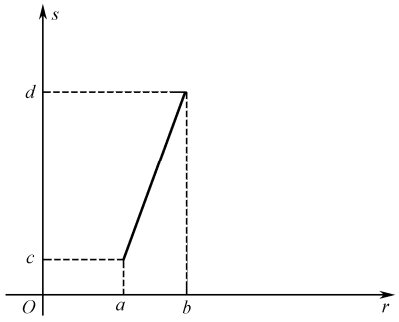


图 3.6　灰度级变换函数

2）非线性扩展

灰度级变换函数 $s=T(r)$ 可为

$$s=\mathrm{c}\log(1+r) \tag{3.19}$$

3）灰度倒置

灰度级变换函数 $s=T(r)$ 可为

$$s=255-r \tag{3.20}$$

4）二值化

确定一阈值 r_1，$0<r_1<255$。$r>r_1$ 的灰度值置白，$r<r_1$ 的灰度值置黑。

3．实验方法及程序

（1）选择一幅图像 lena.jpg，设置输入/输出变换的灰度级范围，a=0.3, b=0.6, c=0.1, d=0.9。

（2）设置非线性扩展函数的参数 c=2。

（3）采用灰度倒置变换函数 s=255–r 进行图像变换。

（4）设置二值化图像的阈值，分别为 level=0.4，level=0.7 参考程序如下。

```
I=imread('i_lena.jpg');
figure;
subplot(1,3,1);
imshow(I);
title('原图');
J=imadjust(I,[0.3;0.6],[0.1;0.9]);   %设置灰度变换的范围
subplot(1,3,2);
imshow(J);
title('线性扩展');
I1=double(I);                        %将图像转换为 double 类型
I2=I1/255;                           %归一化此图像
C=2;
K=C*log(1+I2);                       %求图像的对数变换
subplot(1,3,3);
imshow(K);
title('非线性扩展');
M=255-I;                             %将此图像取反
figure;
subplot(1,3,1);
imshow(M);
title('灰度倒置');
N1=im2bw(I,0.4);                     %将此图像二值化，阈值为 0.4
N2=im2bw(I,0.7);                     %将此图像二值化，阈值为 0.7
subplot(1,3,2);
imshow(N1);
title('二值化阈值 0.4');
subplot(1,3,3);
imshow(N2);
title('二值化阈值 0.7');
```

4．实验结果与分析

实验结果如图 3.7 所示。

（a）原图　（b）线性扩展　（c）非线性扩展

（d）灰度倒置　（e）二值化阈值为 0.4　（f）二值化阈值为 0.7

图 3.7　实验 3.3.1 的结果图

结果分析如下。

（1）线性扩展可以将一幅图像的某一灰度级范围线性扩展到另一灰度级范围；如果 $d-c>b-a$，则输出图像的对比度增大。

（2）对数非线性扩展将使图像的低灰度级扩展，高灰度级压缩，即图像加亮、减暗。

（3）灰度倒置变换将图像中白的变成黑的，黑的变成白的，其他的灰度级也发生相应的变化；

（4）二值化变换将图像变成黑白图像，阈值不同得到的结果将不同，低于阈值的灰度将变为 0，高于阈值的灰度变为 1。

5．思考题

线性扩展与对数非线性扩展各有什么特点？对数非线性变换能否使图像的低灰度级压缩，高灰度级扩展或者低灰度级扩展，高灰度级压缩？选择不同的阈值，观察阈值对图像二值化的影响。

3.3.2　图像的代数运算

1．实验内容

选择两幅图像，一幅是物体图像，一幅是背景图像，采用正确的图像代数运算方法，分别实现图像叠加、混合图像的分离和图像的局部显示效果。

2．实验原理

代数运算是指对两幅或两幅以上输入图像进行点对点的加、减、乘、除运算而得到目标图像的运算。图像处理代数运算的 4 种基本形式分别如下。

$$C(x,y)=A(x,y)+B(x,y) \tag{3.21}$$

$$C(x,y)=A(x,y)-B(x,y) \tag{3.22}$$

$$C(x,y)=A(x,y)\times B(x,y) \quad (3.23)$$

$$C(x,y)=A(x,y)\div B(x,y) \quad (3.24)$$

式中，$A(x,y)$ 和 $B(x,y)$ 为输入图像表达式；$C(x,y)$ 为输出图像表达式。

3．实验方法及程序

（1）选取两幅大小一样的灰度图像 i_lena.jpg 和 rice.png，将两幅图像进行加法运算。程序如下，结果如图 3.8（a）所示。

```
I=imread('i_lena.jpg');
I=rgb2gray(I);
J=imread('rice.png');
I=im2double(I);          %将图像转换为 double 型
J=im2double(J);
K=I+0.3*J;               %两幅图像相加
subplot(1,3,1);
imshow(I);
title('人物图');
subplot(1,3,2);
imshow(J);
title('背景图');
subplot(1,3,3);
imshow(K);
title('相加后的图');
imwrite(K,'i_lena1.jpg');
```

（2）选取一幅混合图像，如图 3.8（a）相加得到的图像 i_lena.jpg，将混合图像与背景图像做减法运算，程序如下，结果如图 3.8（b）所示。

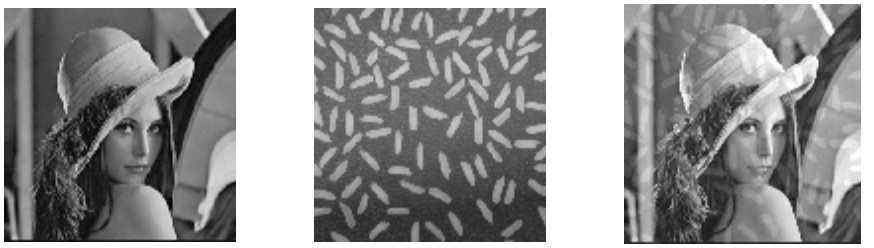
```
A=imread('i_lena1.jpg');
B=imread('rice.png');
C=A-0.3*B;               %混合图减去背景图
subplot(1,3,1);
imshow(A);
title('混合图');
subplot(1,3,2);
imshow(B);
title('背景图');
subplot(1,3,3);
imshow(C);
title('分离后的图');
```

（3）选取一幅尺寸为 256×256 像素的灰度图，如 i_lena.jpg。设置掩模模板，对于需要保留下来的区域，掩模图像的值置为 1，而在需要被抑制掉的区域，掩模图像的值置为 0。程序如下，结果如图 3.8（c）所示。

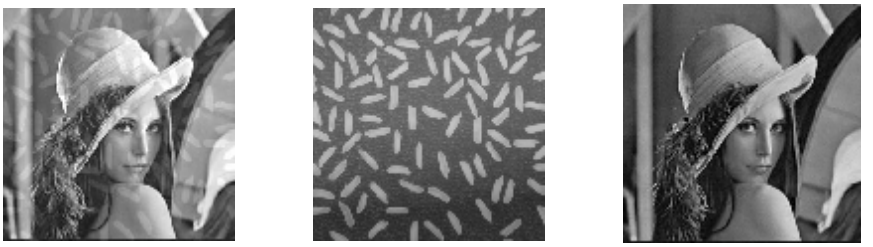
```
A=imread('i_lena.jpg');
A=rgb2gray(A);
A=im2double(A);
subplot(1,2,1);
imshow(A);
title('原图');
B=zeros(256,256);        %设置模板
B(40:200,40:200)=1;
K=A.*B;                  %两幅图像相乘
subplot(1,2,2);
imshow(K);
title('局部图');
```

4. 实验结果与分析

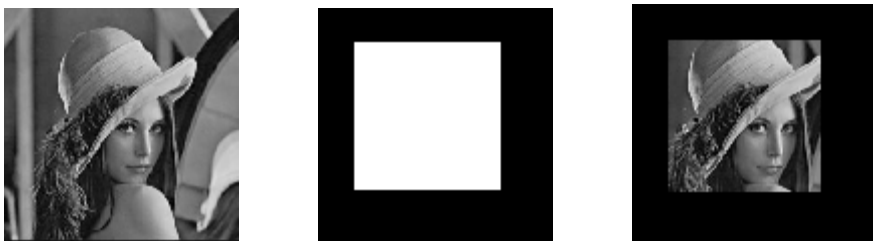
实验结果如图 3.8 所示。



（a）图像的叠加



（b）混合图像的分离



（c）图像的局部效果图

图 3.8　实验 3.3.2 的结果图

由所得结果可知，两幅图像进行相加运算可以实现图像的叠加，而通过减法运算可以实现混合图像的分离。乘法运算可以用来获取对图像感兴趣的部分，对于需要保留下来的区域，

掩模图像的值置为 1，而在需要被抑制掉的区域，掩模图像的值置为 0。

5．思考题

任意两幅图片相叠加得到一幅混合图像，将此混合图像减去其中的一幅原图像能得到另一幅原图像吗？通过以上实验小结各种图像代数运算的应用特点。

3.3.3 图像的缩放

1．实验内容

对一幅图像实现按比例缩小和不按比例任意缩小的效果，以及图像的成倍放大和不按比例放大效果。

2．实验原理

数字图像的比例缩放是指将给定的图像在 x 方向和 y 方向按相同的比例 a 缩放，从而获得一幅新的图像，又称为全比例缩放。如果 x 方向和 y 方向缩放的比例不同，则图像的比例缩放会改变原始图像像素间的相对位置，产生几何畸变。设原始图像中的点 $A_0(x_0,y_0)$ 比例缩放后，在新图中的对应点为 $A_1(x_1,y_1)$，则 $A_0(x_0,y_0)$ 和 $A_1(x_1,y_1)$ 之间的坐标关系可表示如下。

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} \tag{3.25}$$

即

$$\begin{cases} x_1 = ax_0 \\ y_1 = ay_0 \end{cases} \tag{3.26}$$

3．实验方法及程序

选取一幅大小为 256×256 像素的图像，如 i_lena.jpg.分别将图比例放大 1.5 倍，比例缩小 0.7 倍，非比例放大到 420×384 像素，非比例缩小到 150×180 像素。程序如下，结果如图 3.9 所示。

```
A=imread('i_lena.jpg');
B1=imresize(A,1.5);          %比例放大 1.5 倍,默认采用的是最近邻法进行线性插值
B2=imresize(A,[420 384]);    %非比例放大到 420：384
C1=imresize(A,0.7);          %比例缩小 0.7 倍
C2=imresize(A,[150 180]);    %非比例缩小到 150：180
figure;
imshow(B1);
title('比例放大图');
figure;
imshow(B2);
title('非比例放大图');
figure;
imshow(C1);
title('比例缩小图');
```

```
figure;
imshow(C2);
title('非比例缩小图');
```

4．实验结果与分析

实验结果如图 3.9 所示。

（a）比例放大图　（b）非比例放大图

（c）比例缩小图　（d）非比例缩小图

图 3.9　实验 3.3.3 的结果图

由所得结果可以看出，按非比例缩小或放大的图像相对原图像有畸变，而按比例缩放的图像没有。由于 x 方向和 y 方向缩放的比例不同，则图像的非比例缩放会改变原始图像像素间的相对位置，产生几何畸变。

5．思考题

由非比例缩放得到的图片能够恢复到原图片吗？为什么？

3.3.4　图像的旋转

1．实验内容

将一幅图像分别旋转 45° 和 90°，与原图像对比，观察它们的区别。

2．实验原理

设原始图像的任意点 $A_0(x_0, y_0)$ 经旋转角度 β 以后到新的位置 $A(x, y)$。图像的旋转变换可以用矩阵形式表示如下。

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$$

图像旋转之后也可以根据新点求解原始新点的坐标，其矩阵表示形式如下。

$$\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

3．实验方法及程序

读取一幅图片，如 i_lena.jpg，设置图像旋转的角度分别为 45° 和 90°，采用图形旋转函数 imrotate 对图像进行旋转。程序如下，结果如图 3.10 所示。

```
I=imread('i_lena.jpg');
J=imrotate(I,45);                %图像进行逆时针旋转，默认采用最近邻插值法进行插值处理
K=imrotate(I,90);                %默认旋转出界的部分不被截出
subplot(1,3,1);
imshow(I);
subplot(1,3,2);
imshow(J);
subplot(1,3,3);
imshow(K);
```

4．实验结果与分析

实验结果如图 3.10 所示。

（a）原图　　（b）旋转 45° 图　　（c）旋转 90° 图

图 3.10　实验 3.3.4 的结果图

由所得结果可以看出，图像进行逆时针旋转，旋转出界的部分没有被截出。旋转 45° 的结果图相比原图要模糊一点，而旋转 90° 的结果图没有发生变化。因为原来的图像坐标点位于整数点，而旋转 45° 的图像经上述变换公式变换后，其像素坐标点不在整数坐标位置上，所以会产生一定程度的失真。

5．思考题

图像的旋转会导致图像的失真吗？若有，有什么办法可以解决这个问题？