实验一 图像处理基础及灰度、直方图变换

课程名称：图像处理基础

实验学时：4 学时

综合性、设计性实验： ☑是 ☑否

面向专业和班级： 19 级软件工程（1 班、2 班、3 班、4 班、中澳班、卓越班）

学生人数：102

任务编制人：课程负责人

实验内容

一、 MATLAB 数字图像处理初步

二、 图像的代数运算

三、 图像增强-灰度变换

四、 图像增强-直方图变换

一、 MATLAB 数字图像处理初步

（一）实验目的与要求

1．熟悉及掌握 MATLAB 中能够处理的图像格式

2．熟练掌握在 MATLAB 中如何读取图像

3．掌握如何利用 MATLAB 来获取图像的大小、颜色、高度、宽度等相关信息

4．掌握如何在 MATLAB 中按照指定要求存储一幅图像的方法

5．图像间如何转化

（二）实验原理及知识点

1、数字图像的表示和类别

一幅图像可以被定义为一个二维函数 f(x,y),其中 x 和 y 是空间(平面)坐标，f 在任何坐标处(x,y)处的振幅称为图像在该点的亮度。灰度是用来表示黑白图像亮度的一个术语，而彩色图像是由单个二维图像组合形成的。例如，在 RGB 彩色系统中，一幅彩色图像是由三幅独立的分量图像(红、绿、蓝)组成的。因此，许多为黑白图像处理开发的技术适用于彩色图像处理，方法是分别处理三副独立的分量图像即可。

图像关于 x 和 y 坐标以及振幅连续。要将这样的一幅图像转化为数字形式，就要求数字化坐标和振幅。将坐标值数字化称为取样；将振幅数字化称为量化。采样和量化的过程如图 1 所示。因此，当 f 的 x、y 分量和振幅都是有限且离散的量时，称该图像为数字图像。

作为 MATLAB 基本数据类型的数值数组本身十分适于表达图像，矩阵的元素和图像的像素之间有着十分自然的对应关系。

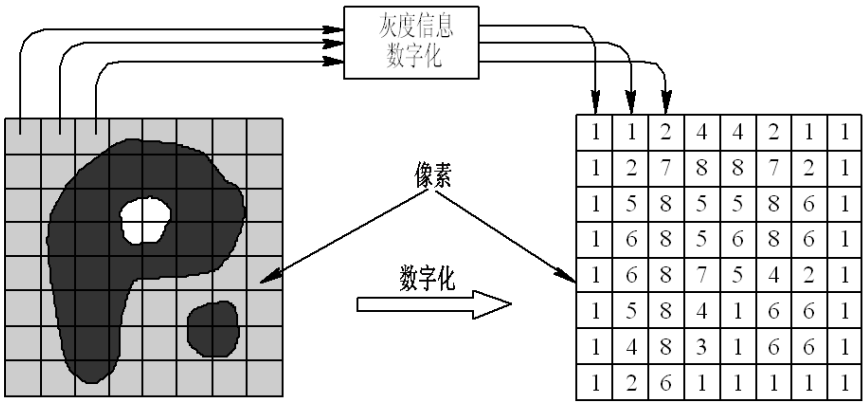


图 1 图像的采样和量化

根据图像数据矩阵解释方法的不同，MATLAB 把其处理为 4 类：

- 亮度图像(Intensity images)
- 二值图像(Binary images)
- 索引图像(Indexed images)

- RGB 图像(RGB images)

(1) 亮度图像

一幅亮度图像是一个数据矩阵，其归一化的取值表示亮度。若亮度图像的像素都是 uint8 类或 uint16 类，则它们的整数值范围分别是[0，255]和[0，65536]。若图像是 double 类，则像素取值就是浮点数。规定双精度型归一化亮度图像的取值范围是[0，1]

(2) 二值图像

一幅二值图像是一个取值只有 0 和 1 的逻辑数组。而一幅取值只包含 0 和 1 的 uint8 类数组，在 MATLAB 中并不认为是二值图像。使用 logical 函数可以把数值数组转化为二值数组或逻辑数组。创建一个逻辑图像，其语法为：

B=logical(A)

其中，B 是由 0 和 1 构成的数值数组。

要测试一个数组是否为逻辑数组，可以使用函数:

islogical(c)

若 C 是逻辑数组，则该函数返回 1；否则，返回 0。

(3) 索引图像

索引颜色通常也称为映射颜色，在这种模式下，颜色都是预先定义的，并且可供选用的一组颜色也很有限，索引颜色的图像最多只能显示 256 种颜色。

一幅索引颜色图像在图像文件里定义，当打开该文件时，构成该图像具体颜色的索引值就被读入程序里，然后根据索引值找到最终的颜色。

(4) RGB 图像

一幅 RGB 图像就是彩色像素的一个 M×N×3 数组，其中每一个彩色相似点都是在特定空间位置的彩色图像相对应的红、绿、蓝三个分量。按照惯例，形成一幅 RGB 彩色图像的三个图像常称为红、绿或蓝分量图像。

令 fR，fG 和 fB 分别代表三种 RGB 分量图像。一幅 RGB 图像就利用 cat(级联)操作将这些分量图像组合成彩色图像：

rgb_image=cat(3,fR,fG,fB)

在操作中，图像按顺序放置。

2、数据类和图像类型间的转化

表 1 中列出了 MATLAB 和 IPT 为表示像素所支持的各种数据类。表中的前 8 项称为数值数据类，第 9 项称为字符类，最后一项称为逻辑数据类。

工具箱中提供了执行必要缩放的函数(见表 2)。以在图像类和类型间进行转化。

表 1-1　MATLAB 和 IPT 支持数据类型

名称	描述
double	双精度浮点数，范围为-10^{308}~10^{308}
uint8	无符号 8 比特整数，范围为[0 255]
uint16	无符号 16 比特整数，范围为[0 65536]

uint32	无符号 32 比特整数，范围为[0 4294967295]
int8	有符号 8 比特整数，范围为[-128 127]
int16	有符号 16 比特整数，范围为[-32768 32767]
int32	有符号 32 比特整数，范围为[-2147483648 2147483647]
single	单精度浮点数，范围为-10^{308}~10^{308}
char	字符
logical	值为 0 或 1

表 1-2 格式转换函数

名称	将输入转化为	有效的输入图像数据类
im2uint8	uint8	logical,uint8,uint16 和 doulbe
im2uint16	uint16	logical,uint8,uint16 和 doulbe
mat2gray	double,范围为[0 1]	double
im2double	double	logical,uint8,uint16 和 doulbe
im2bw	logical	uint8,uint16 和 double

下面给出读取、压缩、显示一幅图像的程序(%后面的语句属于标记语句，编程时可不用输入)

```
I=imread('原图像名.tif');   % 读入原图像,tif 格式
whos I                      % 显示图像 I 的基本信息
imshow(I)                   % 显示图像
imfinfo filename
imwrite(I,'filename.jpg','quality',q); % 这种格式知识用于 jpg 格式，压缩存储图像，q 是 0-100 之间的整数
imwrite(I,'filename.bmp');          % 以位图（BMP）的格式存储图像
% 显示多幅图像，其中 n 为图形窗口的号数
figure(n), imshow('filename');
gg=im2bw(I);                % 将图像转为二值图像
figure, imshow(gg)                  % 显示二值图像
```

（三）实验内容及步骤

1．利用 imread( )函数读取一幅图像，假设其名为 flower.tif，存入一个数组中；

2．利用 whos 命令提取该读入图像 flower.tif 的基本信息；

3．利用 imshow()函数来显示这幅图像；

4．利用 imfinfo 函数来获取图像文件的压缩，颜色等等其他的详细信息；

5．利用 imwrite()函数来压缩这幅图象，将其保存为一幅压缩了像素的 jpg 文件,设为 flower.jpg；语法：imwrite(原图像，新图像，'quality',q), q 取 0-100。

6.同样利用 imwrite()函数将最初读入的 tif 图象另存为一幅 bmp 图像，设为 flower.bmp。

7．用 imread()读入图像：Lenna.jpg 和 camema.jpg；

8．用 imfinfo()获取图像 Lenna.jpg 和 camema.jpg 的大小；

9. 用 figure,imshow()分别将 Lenna.jpg 和 camema.jpg 显示出来，观察两幅图像的质量。

10. 用 im2bw 将一幅灰度图像转化为二值图像，并且用 imshow 显示出来观察图像的特征。

11. 将每一步的函数执行语句拷贝下来，写入实验报告，并且将得到第 3、9、10 步得到的图像效果拷贝下来。

（四）考核要点

1、熟悉在 MATLAB 中如何读入图像、如何获取图像文件的相关信息、如何显示图像及保存图像等，熟悉相关的处理函数。

2、明确不同的图像文件格式，由于其具体的图像存储方式不同，所以文件的大小不同，因此当对同一幅图像来说，有相同的文件大小时，质量不同。

（五）实验仪器与软件

(1) PC 计算机

(2) MatLab 软件/语言包括图像处理工具箱(Image Processing Toolbox)

(3) 实验所需要的图片

（六）实验报告要求

描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果和源代码，并进行必要的讨论，必须包括原始图像及其计算/处理后的图像。

（七）思考题

(1) 简述 MatLab 软件的特点。

(2) MatLab 软件可以支持哪些图像文件格式？

(3) 说明函数 imread 的用途格式以及各种格式所得到图像的性质。

(4) 为什么用 I = imread('lena.bmp') 命令得到的图像 I 不可以进行算术运算？

（八）实验图像

Fig.1 flower.tif

Fig.2 elephant.jpg

Fig.3 Lenna.jpg

Fig.4 camema.jpg

附录：MATLAB 简介

1、Matlab软件Image Processing Toolbox

MatLab的原文是Matrix Laboratory，它包括若干个工具箱如Communication control wavelet toolbox、Image processing toolbox等等，其中图像处理工具箱的函数可以完成Geometric operation、neighborhood and block operations、linear filtering、transform image analysis、enhancement binary、image operation等操作。

MATLAB6.5 函数 imread 和 imwrite 所支持的一些常用的图像/图形格式

格式名称	描述	可识别扩展符
TIFF	加标示的图像文件格式	.tiff , .tif
JPEG	联合图像专家组	.jpg , .jpeg
GIF	图像交换格式	.gif
BMP	Windows 位图	.bmp
PNG	可移植网络图形	.png
XWD	X Window 转储	.xwd

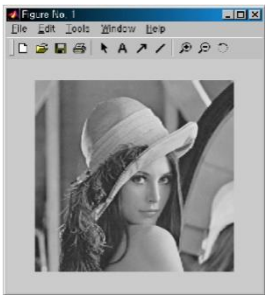
2 Matlab 软件Image processing commands 的使用

A。 Read image file from disk 使用MatLab命令读入图像
Matlab 可以支持BMPF、JPEG、BMP等图像文件格式
MatLab Command ： **imread**
键入
> I = imread('lena.bmp')； % 将图像文件Lena.bmp的数据读入矩阵(array)I中

B。 Image Display 使用MatLab 命令显示图像

MatLab Command ： **figure** 生成图像窗口
键入
> figure(1)； 生成一个图像窗口1
MatLab Command ： **imshow**
键入

- imshow(I,[])；



上图为显示结果。

C。 MatLab 命令的连续执行

D。 本实验中需要了解/使用的函数

本实验中需要学生了解并学会使用下列命令或函数

clear 清除所有变量。执行本命令将会清除内存中的全部变量。

title(' ') 在当前的图像窗口中加上标题，例如 title('原始图像'); % 在当前显示的图像上加上标题

imread（） 读入一个图像文件，在使用图像进行运算时，图像的像素必须使用双精度浮点制式，而 I = imread (‘lena.bmp’) 只能得到 8 比特的数字图像，可以显示但不能满足运算的要求，因此在读入文件时，采用这样的格式[Im, map] = imread ('girl.bmp')；这个命令将文件 girl.bmp 中的图像作为索引图像读入矩阵 Im 之中，索引图像包括一个图像数据矩阵 Im 和一个调色板矩阵 map ，然后使用 MatLab 提供了数制转换的函数，将其转换成双精度浮点制式的灰度函数，I = ind2gray(Im, map)； 这时的图像 I ，便成为双精度浮点制式的灰度图像。这样，就可以适用于各种运算的要求。

Imcrop（） 图像剪裁。 使用格式为 i = imcrop (I, [Xmin, Ymin, dX, dY])； 这里 i 为剪裁后的图像,I 为被剪裁的原始图像。Xmin 为检测的水平起始点,即 X 方向的最小值,Ymin 为检测的垂直起始点,即 Y 方向的最小值(请注意图像坐标与普通平面直角坐标的区别)。dX 为水平方向的剪裁宽度，即剪裁下来的图像在水平方向所具有的像素数量；dY 为垂直方向的剪裁高度，即剪裁下来的图像在垂直方向所具有的像素数量。例如 i = imcrop (I, [180, 200, 200, 200])的操作会将输入的原始图像 I 从坐标(180, 200)处起，剪切下 200 × 200 的一块子图像 i。

MatLab 中图像的运算：在 MatLab 中，图像就是一个矩阵，用一个符号代替运算时当作一个变量即可，因此运算的书写十分简洁，故 MatLab 有草稿纸式的算法语言之称。下面是一些运算实例：

I_a = I + 0。5；%为原始图像 I 加上一个常数 0。5，并将结果赋予变量 I_a；以此类推可以书写出减法 I_s = I - 0。5； 乘法(数乘)I_m = I * 2； 除法 I_d = I

/ 3；

E。 其它可能用到命令有

imresize（） 调节图像大小

imcontour（） 画出图像的轮廓

imadjust(f,[low_in high_in],[low_out high_out]) 对灰度图进行亮度变化

imhist（） 计算图像的直方图

histeq（） 直方图均衡

rgb2gray（） 将 RGB 图像转换成灰度图像

g=ice('porperty name','property value');

3 参考资料

[1] 冈萨雷斯，数字图像处理(MATLAB版)，电子工业出版社.

[2] 章毓晋，图像工程（上，中)，清华大学出版社，2008.

二、图像的代数运算

（一）实验目的

1．了解图像的算术运算在数字图像处理中的初步应用。

2．体会图像算术运算处理的过程和处理前后图像的变化。

（二）实验原理

图像的代数运算是图像的标准算术操作的实现方法，是两幅输入图像之间进行的点对点的加、减、乘、除运算后得到输出图像的过程。如果输入图像为A(x,y)和B(x,y)，输出图像为C(x,y)，则图像的代数运算有如下四种形式：

C(x,y) = A(x,y) + B(x,y)

C(x,y) = A(x,y) - B(x,y)

C(x,y) = A(x,y) * B(x,y)

C(x,y) = A(x,y) / B(x,y)

图像的代数运算在图像处理中有着广泛的应用，它除了可以实现自身所需的算术操作，还能为许多复杂的图像处理提供准备。例如，图像减法就可以用来检测同一场景或物体生产的两幅或多幅图像的误差。

使用MATLAB的基本算术符(+、-、*、/ 等)可以执行图像的算术操作，但是在此之前必须将图像转换为适合进行基本操作的双精度类型。为了更方便地对图像进行操作，MATLAB图像处理工具箱包含了一个能够实现所有非稀疏数值数据的算术操作的函数集合。下表列举了所有图像处理工具箱中的图像代数运算函数。

表 2-1 图像处理工具箱中的代数运算函数

函数名	功能描述
Imabsdiff	两幅图像的绝对差值
Imadd	两幅图像的加法
Imcomplement	补足一幅图像
Imdivide	两幅图像的除法
Imlincomb	计算两幅图像的线性组合
Immultiply	两幅图像的乘法
imsubtract	两幅图像的减法

使用图像处理工具箱中的图像代数运算函数无需再进行数据类型间的转换，这些函数能够接受uint8和uint16数据，并返回相同格式的图像结果。虽然在函数执行过程中元素是以双精度进行计算的，但是MATLAB工作平台并不会将图像转换为双精度类型。

代数运算的结果很容易超出数据类型允许的范围。例如，uint8数据能够存储的最大数值是255，各种代数运算尤其是乘法运算的结果很容易超过这个数值，有时代数操作（主要是除法运算）也会产生不能用整数描述的分数结果。图像的代数运算函数使用以下截取规则使运算结果符合数据范围的要求：超出数据范围的整型数据将被截取为数据范围的极值，分数结果将被四舍五入。例如，如果数据类型是uint8，那么大于255的结果（包括无穷大inf）将被设置为255。

注意：无论进行哪一种代数运算都要保证两幅输入图像的大小相等，且类型相同。

（三）实验步骤

1．图像的加法运算

图像相加一般用于对同一场景的多幅图像求平均效果，以便有效地降低具有叠加性质的随机噪声。直接采集的图像品质一般都较好，不需要进行加法运算处理，但是对于那些经过长距离模拟通讯方式传送的图像（如卫星图像），这种处理是必不可少的。

在MATLAB中，如果要进行两幅图像的加法，或者给一幅图像加上一个常数，可以调用imadd函数来实现。imadd函数将某一幅输入图像的每一个像素值与另一幅图像相应的像素值相加，返回相应的像素值之和作为输出图像。imadd函数的调用格式如下：

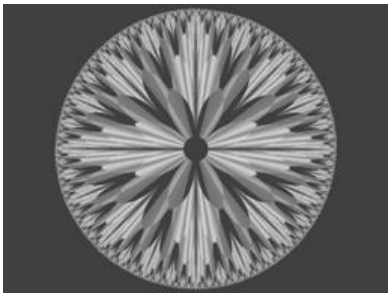
Z = imadd（X，Y）

其中，X和Y表示需要相加的两幅图像，返回值Z表示得到的加法操作结果。

图像加法在图像处理中应用非常广泛。例如，以下代码使用加法操作将图2.1中的(a)、(b)两幅图像叠加在一起：

```
I = imread('2_1(a).tif');
J = imread('2_1(b).tif');
K = imadd(I,J);
imshow(K);
```

叠加结果如图2.2所示。

2_1(a).tif　　2_1(b).tif

图2.1 待叠加的两幅图像

图2.2 叠加后的图像效果

给图像的每一个像素加上一个常数可以使图像的亮度增加。例如，以下代码将增加图3(a)所示的RGB图像的亮度，加亮后的结果如图3(b)所示。

```
RGB = imread('flower2.tif');
RGB2 = imadd(RGB,50);
subplot(1,2,1); imshow(RGB);
subplot(1,2,2); imshow(RGB2);
```

flower2.tif


加50　　减50

图2.3 亮度增加与变暗

两幅图像的像素值相加时产生的结果很可能超过图像数据类型所支持的最大值，尤其对于uint8类型的图像，溢出情况最为常见。当数据值发生溢出时，imadd函数将数据截取为

数据类型所支持的最大值，这种截取效果称之为饱和。为了避免出现饱和现象，在进行加法计算前最好将图像转换为一种数据范围较宽的数据类型。例如，在加法操作前将uint8图像转换为uint16类型。

2．图像的减法运算

图像减法也称为差分方法，是一种常用于检测图像变化及运动物体的图像处理方法。图像减法可以作为许多图像处理工作的准备步骤。例如，可以使用图像减法来检测一系列相同场景图像的差异。图像减法与阈值化处理的综合使用往往是建立机器视觉系统最有效的方法之一。在利用图像减法处理图像时往往需要考虑背景的更新机制，尽量补偿由于天气、光照等因素对图像显示效果造成的影响。

在MATLAB中，使用imsubtract函数可以将一幅图像从另一幅图像中减去，或者从一幅图像中减去一个常数。imsubtract函数将一幅输入图像的像素值从另一幅输入图像相应的像素值中减去，再将这个结果作为输出图像相应的像素值。imsubtract函数的调用格式如下：

Z = imsubtract(X,Y)；

其中，Z是X-Y操作的结果。以下代码首先根据原始图像（如图2.4(a)所示）生成其背景亮度图像，然后再从原始图像中将背景亮度图像减去，从而生成图2.4(b)所示的图像：

```
camema = imread('camema.jpg')；
background = imopen(camema, strel('disk',15))；
%imopen() 形态学中的开运算，开运算数学上是先腐蚀后膨胀的结果，开运算的结果为完全删除了不能包含结构元素的对象区域，平滑了对象的轮廓，断开了狭窄的连接，去掉了细小的突出部分。
output = imsubtract(camema, background)；
subplot(1,2,1)；imshow(camema)；
subplot(1,2,2)；imshow(rice)；
```

camema.jpg　　output

图2.4 原始图像、减去背景图像

如果希望从图像数据I的每一个像素减去一个常数，可以将上述调用格式中的Y替换为一个指定的常数值，例如：

Z = imsubtract(I,50)；


原图　　减50

减法操作有时会导致某些像素值变为一个负数，对于uint8或uint16类型的数据，如果发生这种情况，那么imsubtract函数自动将这些负数截取为0。为了避免差值产生负值，同时避免像素值运算结果之间产生差异，可以调用函数imabsdiff。imabsdiff将计算两幅图像相应像素差值的绝对值，因而返回结果不会产生负数。该函数的调用格式与imsubtract函数类似。

3. 图像的乘法运算

两幅图像进行乘法运算可以实现掩模操作，即屏蔽掉图像的某些部分。一幅图像乘以一个常数通常被称为缩放，这是一种常见的图像处理操作。如果使用的缩放因子大于1，那么将增强图像的亮度，如果因子小于1则会使图像变暗。缩放通常将产生比简单添加像素偏移量自然得多的明暗效果，这是因为这种操作能够更好地维持图像的相关对比度。此外，由于时域的卷积或相关运算与频域的乘积运算对应，因此乘法运算有时也被作为一种技巧来实现卷积或相关处理。

在MATLAB中，使用immultiply函数实现两幅图像的乘法。immultiply函数将两幅图像相应的像素值进行元素对元素的乘法操作（MATLAB点乘），并将乘法的运算结果作为输出图形相应的像素值。immulitply函数的调用格式如下：

Z = immulitply(X,Y)

其中，Z=X*Y。例如，以下代码将使用给定的缩放因子对图2.5(a)所示的图像进行缩放，从而得到如图2.5(b)所示的较为明亮的图像：

```
I = imread('room.tif');
J = immultiply(I,1.2);
subplot(1,2,1); imshow(I);
subplot(1,2,2); imshow(J);
```

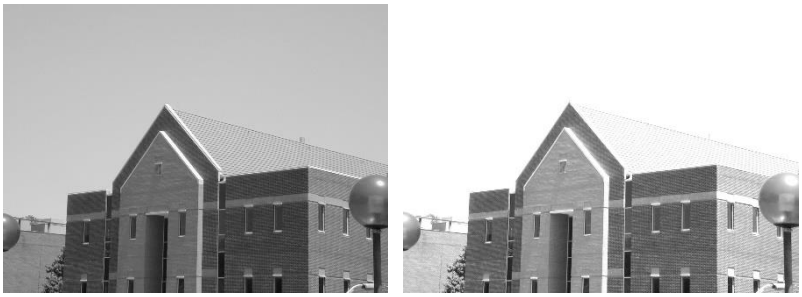


图2.5 原图和乘以因子1.2 的图像

uint8图像的乘法操作一般都会发生溢出现象。Immultiply函数将溢出的数据截取为数据类型的最大值。为了避免产生溢出现象，可以在执行乘法操作之前将uint8图像转换为一种数据范围较大的图像类型，例如uint16。

4．图像的除法运算

除法运算可用于校正成像设备的非线性影响，这在特殊形态的图像（如断层扫描等医学图像）处理中常常用到。图像除法也可以用来检测两幅图像间的区别，但是除法操作给出的是相应像素值的变化比率，而不是每个像素的绝对差异，因而图像除法也称为比率变换。

在MATLAB中使用imdivide函数进行两幅图像的除法。imdivide函数对两幅输入图像的所有相应像素执行元素对元素的除法操作（点除），并将得到的结果作为输出图像的相应像素值。imdivide函数的调用格式如下：

Z = imdivide(X,Y)

其中，Z=X/Y。例如，以下代码将图4所示的两幅图像进行除法运算，请将这个结果和减法操作的结果相比较，对比它们之间的不同之处：

```
Image1 = imread('camema.jpg');
Ip = imdivide(Image1, output)；%output是2中减法操作的输出值
Ipim=mat2gray(Ip);
imshow(Ipim, []);
```

除法操作的结果如图2.6所示。

图2.6 原图和减背景后的图像相除的图像效果

5．图像的四则代数运算

可以综合使用多种图像代数运算函数来完成一系列的操作。例如，使用以下语句计算两幅图像的平均值：

```
I = imread('rice.tif');
I2 = imread('cameraman.tif');
K = imdivide(imadd(I,I2),2);
```

建议最好不要用这种方式进行图像操作，这是因为，对于uint8或uint16数据，每一个算术函数在将其输出结果传递给下一项操作之前都要进行数据截取，这个截取过程将会大大减少输出图像的信息量。执行图像四则运算操作较好的一个办法就是使用函数imlincomb。函数imlincomb按照双精度执行所有代数运算操作，而且仅对最好的输出结果进行截取，该函数的调用格式如下：

Z = imlincomb(A,X,B,Y,C);

其中，Z=A*X+B*Y+C。MATLAB会自动根据输入参数的个数判断需要进行的运算。例如，以下语句将计算Z=A*X+C：

Z = imlincomb(A,X,C)

而以下语句将计算Z=A*X+B*Y：

Z = imlincomb(A,X,B,Y,)

（四）实验报告要求

1 描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果并进行必要的讨论。

2 必须包括原始图像及其计算处理后的图像以及相应的解释。

（五）思考题

由图像算术运算的运算结果，思考图像减法运算在什么场合上发挥优势？

三、图像增强—灰度变换

（一）实验目的：

1、了解图像增强的目的及意义，加深对图像增强的感性认识，巩固所学理论知识。

2、学会对图像直方图的分析。

3、掌握直接灰度变换的图像增强方法。

（二）实验原理及知识点

术语‘空间域’指的是图像平面本身，在空间与内处理图像的方法是直接对图像的像素进行处理。空间域处理方法分为两种：灰度级变换、空间滤波。空间域技术直接对像素进行操作其表达式为

$$g(x,y)=T[f(x,y)]$$

其中f(x,y)为输入图像，g(x,y)为输出图像，T是对图像f进行处理的操作符，定义在点(x,y)的指定领域内。

定义点(x,y)的空间邻近区域的主要方法是，使用中心位于(x,y)的正方形或长方形区域，。此区域的中心从原点(如左上角)开始逐像素点移动，在移动的同时，该区域会包含不同的领域。T应用于每个位置(x,y)，以便在该位置得到输出图像g。在计算(x,y)处的g值时，只使用该领域的像素。

灰度变换T的最简单形式是使用领域大小为1×1，此时,(x,y)处的g值仅由f在该点处的亮度决定，T也变为一个亮度或灰度级变化函数。当处理单设(灰度)图像时，这两个术语可以互换。由于亮度变换函数仅取决于亮度的值，而与(x,y)无关，所以亮度函数通常可写做如下所示的简单形式：

$$s=T(r)$$

其中，r表示图像f中相应点(x,y)的亮度，s表示图像g中相应点(x,y)的亮度。

（三）实验内容：

1、图像数据读出

2、计算并分析图像直方图

3、利用直接灰度变换法对图像进行灰度变换

下面给出灰度变化的MATLAB程序

```
f=imread('medicine_pic.jpg'); %
imhist(f,256);          %显示其直方图,要求图像为灰度图像,如果是 RGB 如何处理？rgb2gray
g1=imadjust(f,[0 1],[1 0]);      %灰度转换，实现明暗转换(负片图像)
figure,imshow(g1)

%将 0.5 到 0.75 的灰度级扩展到范围[0 1]
g2=imadjust(f,[0.5 0.75],[0 1]);
figure,imshow(g2)
g=imread('point.jpg');
h=log(1+double(g));              %对输入图像对数映射变换
imshow(h)                        %观察结果，有什么问题？
```

```
h=mat2gray(h);                    %将矩阵 h 转换为灰度图片
h=im2uint8(h);                    %将灰度图转换为 8 位图
figure,imshow(h)
```

（四）实验仪器

PC 一台 ， MATLAB 软件

（五）实验图片

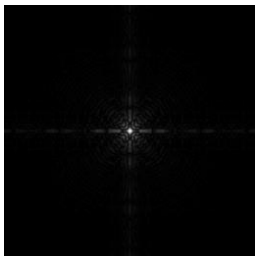


Fig.1 point.jpg

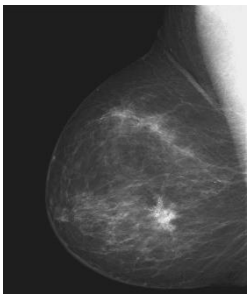


Fig.2 medicine_pic.jpg

四、图像增强—直方图变换

（一）实验目的

1．掌握灰度直方图的概念及其计算方法；
2．熟练掌握直力图均衡化和直方图规定化的计算过程；
3．熟练掌握空域滤波中常用的平滑和锐化滤波器；
4．掌握色彩直方图的概念和计算方法
5．利用MATLAB程序进行图像增强。

（二）实验原理

图像增强是指按特定的需要突出一幅图像中的某些信息，同时，消弱或去除某些不需要的信息的处理方法。其主要目的是处理后的图像对某些特定的应用比原来的图像更加有效。图像增强技术主要有直方图修改处理、图像平滑化处理、图像尖锐化处理和彩色处理技术等。本实验以直方图均衡化增强图像对比度的方法为主要内容，其他方法同学们可以在课后自行联系。

直方图是多种空间城处理技术的基础。直方图操作能有效地用于图像增强。除了提供有用的图像统计资料外，直方图固有的信息在其他图像处理应用中也是非常有用的，如图像压缩与分割。直方图在软件中易于计算，也适用于商用硬件设备，因此，它们成为了实时图像处理的一个流行工具。

直方图是图像的最基本的统计特征，它反映的是图像的灰度值的分布情况。直方图均衡化的目的是使图像在整个灰度值动态变化范围内的分布均匀化，改善图像的亮度分布状态，增强图像的视觉效果。灰度直方图是图像预处理中涉及最广泛的基本概念之一。

图像的直方图事实上就是图像的亮度分布的概率密度函数，是一幅图像的所有象素集合的最基本的统计规律。直方图反映了图像的明暗分布规律，可以通过图像变换进行直方图调整，获得较好的视觉效果。

直方图均衡化是通过灰度变换将一幅图像转换为另一幅具有均衡直方图，即在每个灰度级上都具有相同的象素点数的过程。

下面给出直方图均衡化增强图像对比度的MATLAB程序：

```
I=imread('pollen.jpg');        % 读入原图像
J=histeq(I);                   %对原图像进行直方图均衡化处理,要求图像为灰度图象,RGB 如何处理？
imshow(I);                     %显示原图像
title('原图像');               %给原图像加标题名
%对原图像进行屏幕控制；显示直方图均衡化后的图像
figure;imshow(J);
%给直方图均衡化后的图像加标题名
title('直方图均衡化后的图像') ;
%对直方图均衡化后图像进行屏幕控制；作一幅子图，并排两幅图的第 1 幅
figure; subplot(1,2,1) ;
imhist(I,64);                  %将原图像直方图显示为 64 级灰度
title('原图像直方图') ;        %给原图像直方图加标题名
```

```
subplot(1,2,2);                 %作第 2 幅子图
imhist(J,64) ;                  %将均衡化后图像的直方图显示为 64 级灰度
title('均衡变换后的直方图') ;   %给均衡化后图像直方图加标题名
```

处理后的图像直方图分布更均匀了，图像在每个灰度级上都有像素点。从处理前后的图像可以看出，许多在原始图像中看不清楚的细节在直方图均衡化处理后所得到的图像中都变得十分清晰。

（三）实验步骤

1打开计算机，启动MATLAB程序；程序组中“work”文件夹中应有待处理的图像文件；

2调入“实验四”中获取的数字图像，并进行计算机均衡化处理；

3显示原图像的直方图和经过均衡化处理过的图像直方图。

4记录和整理实验报告

（四）实验仪器

1．计算机；　　2．MATLAB程序；

3．移动式存储器（软盘、U盘等）；　　4．记录用的笔、纸。

（五）实验报告内容

1．叙述实验过程；

2．提交实验的原始图像和结果图像。

（六）思考题

1．直方图是什么概念？它反映了图像的什么信息？

2．直方图均衡化是什么意思？它的主要用途是什么？

（七）实验图片

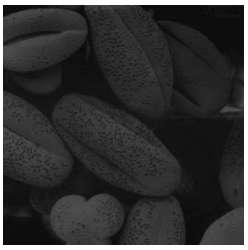


Fig.1 pollen.jpg