实验一 数字图像处理初步与基本运算

一、 实验目的

1. 掌握 MATLAB 语言图像数据与信息的读、写方法。
2. 熟悉及掌握在 MATLAB 中能够处理哪些格式图像。
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. 熟悉图像代数变换与几何变换的原理、作用、特点；掌握图像代数运算、几何运算的方法。2.理解和掌握图像的平移、垂直镜像变换、水平镜像变换、缩放和旋转的原理和应用。掌握图像几何变换的基本原理，熟练掌握数字图像的缩放、旋转、平移、镜像和转置的基本原理及其 MATLAB 编程实现方法。
6. 掌握绘制灰度直方图的方法，理解灰度直方图的灰度变换及均衡化的方法。

二、 实验原理

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。将坐标值数字化成为取样；将振幅数字化成为量化。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。

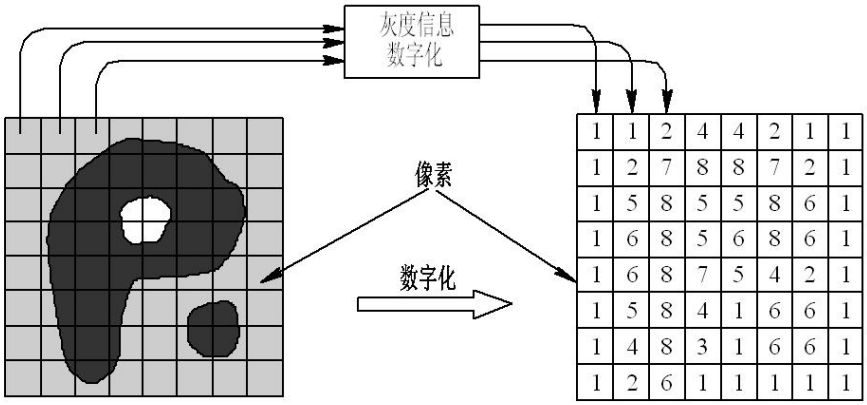

图 1 图像的采样和量化

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
```

```
% 这种格式只适用于 jpg 格式，压缩存储图像，q 是 0-100 之间的整数
imfinfo filename    imwrite(I,'filename.jpg','quality',q);

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('filename');               % 将图像转为二值图像
figure, imshow(gg)                  % 显示二值图像
```

3、图像的代数运算

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1-3 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。

（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。

（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)；

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)；

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。

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。

4、图像的几何运算

1 初始坐标为（x，y）的点经过平移（x0，y0），坐标变为（x'，y'），两点之间的关系为：

$$\begin{cases} x' = x + x_0 \\ y' = y + y_0 \end{cases}$$

以矩阵形式表示为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2 **图像的镜像变换**是以图象垂直中轴线或水平中轴线交换图像的变换，分为垂直镜像变换和水平镜像变换，两者的矩阵形式分别为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3 **图像缩小和放大变换**矩阵相同：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

当$S_x \leqslant 1$，$S_y \leqslant 1$ 时，图像缩小；$S_x \geqslant 1$，$S_y \geqslant 1$ 时，图像放大。

4 **图像旋转**定义为以图像中某一点为原点以逆时针或顺时针方向旋转一定角度。其变换矩阵为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

该变换矩阵是绕坐标轴原点进行的，如果是绕一个指定点（a，b）旋转，则现要将坐标系平移到该点，进行旋转，然后再平移回到新的坐标原点。

三、 实验内容与要求

（1）图像基本操作部分

熟悉 MATLAB 语言中对图像数据读取，显示等基本函数。以 MATLAB 为例，需要熟悉下列命令：熟悉 imread()函数、imwrite()函数、size()函数、subplot()函数、figure()函数。

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

2．利用 whos 命令提取该读入图像 strawberry.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 strawberry.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 strawberry.bmp。

7．用 imread()读入图像：Lenna.tif 和 cameraman.tif；

8．用 imfinfo()获取图像 Lenna.tif 和 cameraman.tif 的大小；

9．用 figure,imshow()分别将 Lenna.tif 和 cameraman.tif 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。

（2）图像代数运算部分

11．将 Fig5.bubbles 和 Fig6. pollen 两幅图像相加，并将图像显示出来；

12．将 pollen 背景亮度图像减去，并在同一张图上显示两幅图像；（估计背景亮度的代码：background=imopen(image,strel('disk',15));

13．将 pollen 原图和减去背景的图像相除，显示相除后的图像；

（3）图像几何运算部分

14．读入一幅图像 room.tif（下同），对其进行水平平移、垂直平移 50 像素(图像扩大)；

15．读入一幅图像，对其进行等比例缩放；

16．实现对图像进行垂直镜像变换的代码；

17．实现对图像进行水平镜像变换的代码；

18．读入一幅图像，对其进行水平转置、垂直转置；

19．实现对图像进行旋转操作的代码。

四、 实验图像

Fig.1strawberry.tif

Fig.2 room.tif

Fig.3 Lenna.tif

Fig.4 cameraman.tif

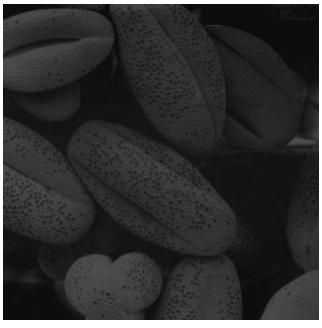

Fig5.bubble

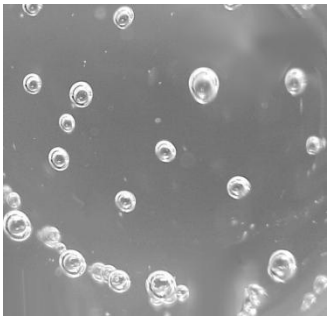

Fig6. pollen

五、 思考题

请在实验报告最后进行简答：

(1) 简述 MatLab 软件的特点。MatLab 软件可以支持哪些图像文件格式？

(2) 说明函数 imread 的用途格式以及各种格式所得到图像的性质。

(3) 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？

(4) 分析说明图像平移过程中如果不做的放大会出现什么效果；

(5) 分析说明能图像镜像与平移的异同；

(6) 分析说明对一图像先进行缩小，再进行放大，能与原图像相同吗？

六、 实验报告要求

请同学们完成上述实验：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/处理后的图像以及相应的解释。