**第2次实验 基于MATLAB的图像基本操作**

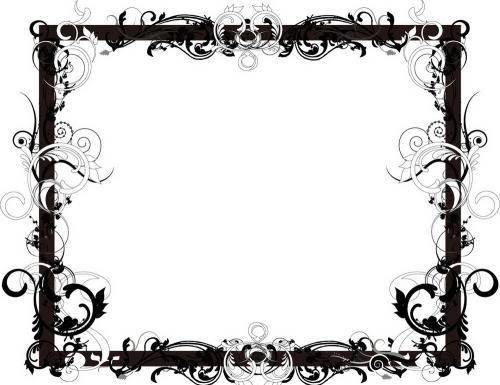
**一 、实验目的：**

* + 1. 掌握MATLAB图像处理工具箱的基本使用方法
    2. 理解图像的算术运算在数字图像处理中的初步应用和作用。
    3. **学会使用帮助系统解决问题**

1. **实验内容和要求：**

**1．**实现图像Baboon.bmp（MATLAB自带）的读入（可使用**imread**）和显示（可使用**imshow**）操作，代码加上足够的注释，需要建立一个M文件实现。

2．已知两张图(test01.jpg和test04.jpg)，如图1(a), (b)所示。



(a) (b)

图 1

1. 将图1(a), (b)两图进行加、减、乘、除算术运算，显示并观察输出结果。思考：运算结果矩阵元素值范围？如何求得？结果值类型？在算术除运算时尤其要注意什么？【答案写进实验报告】

**Tip：**矩阵算术乘除符号为**.\*, ./**；求矩阵极大值和极小值可调用**max和min函数。**

Tip：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. **将图1(b)转换成灰度图**，然后分别乘以1.5和0.5，显示并观察结果。**思考：**这样的操作分别对原图起到什么作用？
2. 将图1(a), (b)两图按以下线性公式进行组合，观察其输出结果，并简要说明其含义。

C=0.4\*A+0.6\*B

1. 上述各种操作能否实现下图所示叠加效果(二次曝光)？为什么？那你有何办法实现？请写出完整的方案和代码，并在**同一个Figure显示**图1(a), (b)和叠加后的结果，同时为每张图添上合适的标题。



图2 叠加效果参考图

3.利用帮助系统了解im2double,imsubtract，imadd函数的作用和语法，并利用这些函数处理已知图像pout.tif（MATLAB自带）并显示处理前后效果在同一个Figure。**观察**：imsubtract，imadd函数调用结果与直接用-和+算术运算的结果有何不同？

4.拓展[选作]：利用加法消除图像叠加的**均值为零的白噪声**。

原图：source.tif

噪声图：Noisy.tif[即在原图上叠加均值为零的白噪声的结果]

5．**所有实验内容完成后参考“实验报告模板.doc”完成实验报告。**

**Tip：**M 文件命名不可以是纯数字，好奇的话可以试试，看看会有什么结果^\_^。

1. **相关实验原理**

**3.1 MATLAB图像的读写与显示**

函数imread可以从任何MATLAB支持的图形图像文件格式中以任意位深度读取一幅图像，其基本调用格式如下：

[X,MAP]=imread（filename，‘format’）

其中filename为需要读入的图像文件名，format为图像格式，如果不指定format参数，系统将根据文件名自动判断图像类型。输出参数X为表示存储图像数据的矩阵名，当图像为索引图像时，MAP（非索引图像时缺省）为该图像的调色板。举例如下：

1. 读matlab自带的一幅图像：I=imread(‘rice.tif’);
2. 读自己的图像：addpath(‘F:\新建文件夹\photo’); imread(‘lena.jpg’);

或者：imread(‘F:\新建文件夹\photo \lena.jpg’);

imread还可以分幀读取一个多幀图像文件。如，以下语句将读取mri.tif文件的第5幀图像：

imread（‘mri.tif’，5）；

函数imwrite可以将一幅图像写成一个MATLAB支持的格式图形文件,其基本调用格式如下：

imwrite（X, MAP, filename，‘format’）；

各个参数的含义与读取图像类似，不再赘述。

在MATLAB中，显示一幅图像的基本方法是调用image函数，这个函数将创建一个图形对象句柄，其调用格式如下：

image（X,Y,C）

其中X,Y表示图像显示左上角坐标，C表示需要显示的图像。另一个图像显示函数imagesc与image函数类似，但它可以自动标度输入数据。函数imshow是MATLAB图像处理工具箱提供的一个高级图像显示函数，具体语法请读者查帮助系统。使用举例如下：

1. 显示matlab自带一幅图像: imshow(‘rice.tif’)
2. 显示一幅自己的图像: addpath(‘F:\新建文件夹\photo’);

imshow(‘lena.jpg’)

或者：imshow(‘F:\新建文件夹\photo \lena.jpg’);

1. 显示图像信息：imfinfo(‘filename’)
2. 创建图形窗口：figure
3. 在一个窗口中显示多幅图像：subplot(2,1,1)，imshow(‘filename1’)

Subplot(2,1,2), imshow(‘filename2’)

上例中filename可以是图像文件名也可以是图像矩阵数据名。

**3.2 MATLAB支持的图像文件格式**

* + BMP格式：有1、4、8、24位非压缩图像和8位行程编码图像。
  + JPEG格式：联合图像专家组的图像压缩格式。
  + TIFF格式：可处理1、4、8、24位图像数据和1位CCITT压缩图像**。**

**3.3 MATLAB支持的图像文件类型**

* + - 索引图像

包括一个数据矩阵（I）和一个颜色影像表矩阵（Map），像素颜色由I作为索引值向Map进行索引。其中Map是一个m\*3的矩阵，每个元素的值都是0～1之间的双精度浮点型数据，每行分别指定红绿蓝。

* + - 灰度图像

灰度图像是一个数据矩阵I，每个元素代表一个像素，I的数据表示在一定范围内的灰度值。I可以是双精度浮点型，其值域为[0.0,1.0];也可以是unit8类型，其值域为[0，256]。

* + - 二值图像

这类图像只包含一个由0，1构成的矩阵，可以保存为双精度或unit8类型的数组。二值图像可以看成一个仅由黑白两色组成的特殊的灰度图。

* + - RGB图像

RGB图像在matlab中存储为一个n\*m\*3的三维数据数组。n,m分别为图像的行列数，数组可以是双精度浮点型或unit8类型。例如有一幅RGB图像的像素（4，7），而它的红绿蓝颜色值分别保存在元素RGB(4,7,1)、RGB(4,7,2)、RGB(4,7,3)。

**3.4不同图像文件类型的转换**

在处理图像过程中，用户在进行某些操作时，往往需要将一幅图像转换成另一种图像类型。比如当用户希望对一幅存储为索引图像的彩色图像进行滤波时就应该首先将图像格式转换为RGB格式，此时再对RGB图像使用滤波时，MATLAB将恰当的滤掉图像中部分灰度值。如果用户企图对一幅索引图像进行滤波，那么MATLAB只能简单的对索引图像矩阵的下标进行滤波，得到一个毫无意义的结果。MATLAB图像处理工具箱中所有的图像类型转换函数如表1所示。

表1 MATLAB图像类型转换函数

|  |  |
| --- | --- |
| 函数 | 功能 |
| dither | 使用抖动方法，根据灰度图西那个创建二进制图像或者根据RGB图像创建索引图像 |
| gray2ind | 根据一幅灰度图像创建索引图像 |
| grayslice | 使用阈值截取方法，根据一幅灰度图像创建索引图像 |
| im2bw | 使用阈值截取方法，根据一幅灰度图像、索引图像或RGB图像创建二进制图像 |
| ind2gray | 根据一幅索引图像创建一幅灰度图像 |
| ind2rgb | 根据一幅索引图像创建一幅RGB图像 |
| mat2gray | 通过数据缩放，再根据矩阵数据创建一幅灰度图像 |
| rgb2gray | 根据一幅RGB图像创建一幅灰度图像 |
| rgb2ind | 根据一幅RGB图像创建一幅索引图像 |

MATLAB图像处理工具箱还提供了图像存储类型间的转换函数，这些函数包括im2double、im2uint8和im2uint16，这些函数可以自动进行原始数据的重新标度和偏移。这三个函数的调用格式都很简单，输入参数为图像矩阵，输出为转换后的图像。具体使用就不再赘述，请读者利用帮助系统自行实现。

**3.4.1 函数使用方法示例**

**1）.索引图像转换为灰度图像**

语句：I=ind2gray(X,map)

以下代码的功能是根据图1（a）所示索引图像来创建一幅灰度图像（显示结果如图1（b）所示）：

load woman

I=ind2gray(X,map);

subplot(1,2,1);

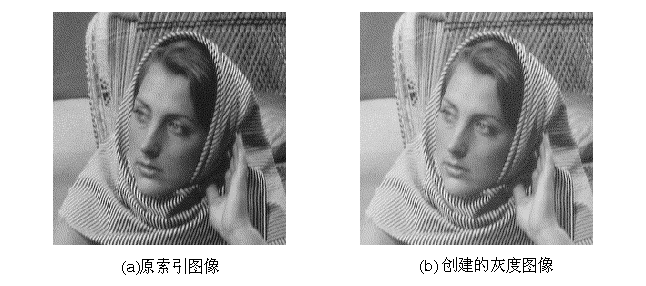
imshow(X,map);

title(‘索引图’)；

subplot(1,2,2);

imshow(I);

title(‘灰度图’)；

****

**图3 ind2gray根据索引图像创建灰度图像效果对比**

**2）.RGB图像转换为灰度图像**

语句：I=rgb2gray(RGB)

以下代码的功能是根据图2（a）所示RGB图像来创建一幅灰度图像（显示结果如图2（b）所示）：

RGB=imread(‘flowers.tif’);

I=rgb2gray(RGB)；

subplot(1,2,1);

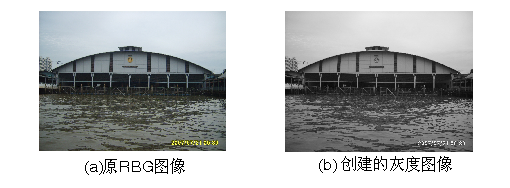
imshow(RGB);

title(‘RBG图像’)；

subplot（1，2，2）；

imshow（I）；

title(‘灰度图像’)；

****

**图4 rgb2gray根据RGB图像创建灰度图像效果对比**

**3）. MATLAB处理图像示例及注释**

clear; ％清除工作空间变量

close all; ％关闭所有显示的图形窗口

I=imread(‘pout.tif’); ％读入一幅图象

imshow(I) ％ 显示一幅图象

whos ％查看内存中的图像存储信息

imwrite(I2,‘pout2.png’); ％保存图像

imfinfo(‘pout2.png’) ％显示图像的信息

**3.6特殊的图像显示技术**

除了前面介绍的imshow函数外，MATLAB图像处理工具箱还提供了一些实现特殊显示功能的函数，例如可以利用colorbar函数将颜色条添加到坐标轴对象中，添加的颜色条用来指示不同颜色对应的数据值。MATLAB还提供了在多幀图像阵列中同时显示所有的图像的函数montage，将多幀图像阵列转换为电影动画的函数immovie。MATLAB的图像处理工具箱还提供了一个纹理映射函数warp专门将二维图像映射到三维图形表面，将映射图像显示在一个特定的三维空间中。

**3.7图像的算术运算**

图像的算术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)，则图像的算术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)

图像的算术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 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。图像的算术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。