一 图像处理基础

1. 图像类型：
2. 矢量图像：采用数字矢量的方法记录图像的内容，以线条和色块为主。这种方法容易进行放大，缩小或者旋转等操作，精确度高，可绘制3D图像。缺陷在于不易制成色调丰富或色彩变化太多的图像，图形不逼真。适合于不同软件间交换文件。
3. 位图图像：位图的方式就是讲图像的每一个像素点转换为一个数据。如以8位来记录，便可以表现出256种颜色或色调，因此使用的位元素越多表现出来的色彩就越丰富，同时也可以在不同软件之间交换文件，但对硬盘内存要求较高。通常使用的颜色有16色，256色，增强16色和真彩色24位（适合用于内容复杂的图像和真实照片）。
4. 索引图像
5. 强度（灰度）图像
6. RGB（真彩图像）
7. 图像术语：
8. 像素（Pixel）

像素是构成数字图像的最小单位，图像由像素点以矩阵的方式排列而成的，分辨率也高，像素点就越小，图像越清晰。

1. DPI（Dot Per Inch）

DPI是指各类输出设备每英寸上所产生的像素点数，一般表示输出设备（如激光打印机分辨率在600~1200dpi）的分辨率，即设备分辨率。

1. PPI（Pixel Per Inch）

PPI是指每英寸的像素值，用于衡量图像输入设备（如扫描仪）的分辨率高低，反映图像中存储信息量的多少，决定了图像的根本质量，一幅粗糙的图像绝不会因为有一台高ppi的设备而变得细腻起来。

1. 屏幕分辨率

屏幕分辨率是指显示器上最大可显示的像素数集合，一般用水平与垂直方向的像素点数来表示。最大的分辨率为1024X768的显示器，其满屏最多可以产生786432个像素点。显示器像素点越多，分辨率越高，图像也就越大，越细致。

1. 位（bit）与颜色（color）

颜色使用数字“位”来实现，它们之间的关系为：颜色数=2n，其中n为所占位数，平常所说的高彩色，即16位显示模式，共有64K种颜色，或者说24位模式下，能处理16M种颜色的真彩色图像。

1. 图像处理MATLAB基本指令
2. A=imread(filename,fmt)

MATLAB用imread指令读取一幅图像，filename字符串指定了灰度图像或彩色图像，fmt指定了文件格式。如果图像不在指定目录或者MATLAB路径下，就要指定完整路径。Imread函数的返回数组A表示的是图像数据，如灰度图像，则A为一个mXn的二维数组，如为彩色图像，则A为一个mxn的三维数组 。数组的类型由图像的数据类型来决定。

[X,map]=imread(…)用于读取引索图像，X用于存储引索图像数据，即相关颜色映射表的序号值，map用于存储与该引索色图像相关的颜色映射表。

[…]=imread(filename)表示根据待读取图像数据是真实像素值或引索色图像的相应颜色映射表的序号而分别采用以上两种调用格式。

[…]=imread(URL,…)用于读取引自Internet URL的图像，URL（Uniform Resource Locator，统一资源定位符）要求必须包含协议类型（protocol type），如http://。

[…]=imread(…,Param1,Val1,Param2,Val2,…)针对某些特殊类型的图像读取语句采用的其他属性参数。

注：通常imread读取的大多数图像都是8位的。当把这些图像加载到内存中时，MATLAB就将其存储为uint8类型。此外MATLAB还支持16位的PNG图像和TIFF图像。当用户读取此类图像时，MATLAB就将其存储为uint16类型。

A=imread('D:\Matlab\DSP\girl.png')直接由路径来寻找图片位置。

1. imwrite(A,filename,fmt)

把图像数据A写入到由filename指定的输出文件中去，存储格式由fmt来指定。A不能为空，对灰度图像来说，A是一个mXn的数组，对彩色图形来说，A是一个mXnX3的三维数组。如果fmt指定的是TIFF格式，则imwrite函数可以接受mXnX4的三维数组。

imwrite(X,map,filename,fmt)用于写入引索色图像，其中X表示引索色图像数据数组，map表示其关联的颜色映射表，filename为fmt格式的输出文件，如果X为uint8或uint16类型的数组，imwrite将数组中的实际数据写入到文件filename中。如果X是double型数组，imwrite函数采用uint8（X-1）表示数组中的值并写入到相应的输出文件中。参数map必须是有效的MATLAB颜色映射表。

Imwrite(…,filename)在写入文件到图像时，从filename的扩展名推断图像的文件格式，该扩展名要求必须是MATLAB所支持的类型值。

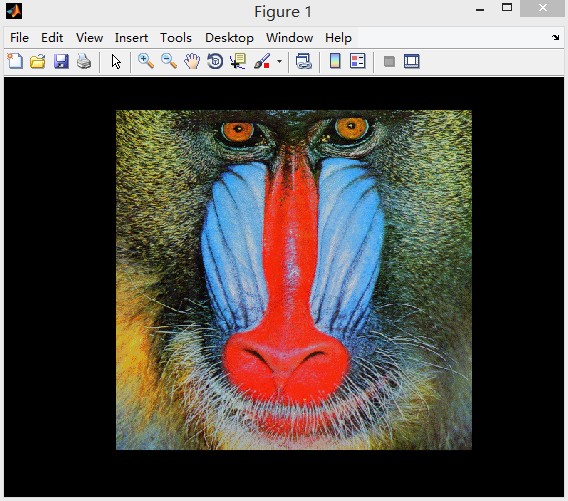
imwrite(…,Param1,Val1,Param2,Val2,…)用于指定HDF、JPEG、PBM、PGM、PNG、PPM和TIFF等类型输出文件的不同参数。如当写入一个HDF文件时，可以指定输出图像的质量（Quality）、压缩性（Compression）和写入模式（WriteMode），而写入JPEG文件时，则可指定像素位数（BitDepth）、注释（Comment）、模式（Mode）和质量（Quality）等。另外，当利用imwrite函数保存图像时，MATLAB默认的存储方式是将其简化为uint8的数据类型，MATLAB还支持保存1位的PNG图像和TIFF图像，当用户保存此类文件时，MATLAB就将其保存为uint16类型。

1. imshow用于实现图像的显示，用户用指令imshow函数显示一幅图像时，该函数将自动设置图像窗口、坐标轴和图像属性。

imshow(I)用于显示灰度级图像I。

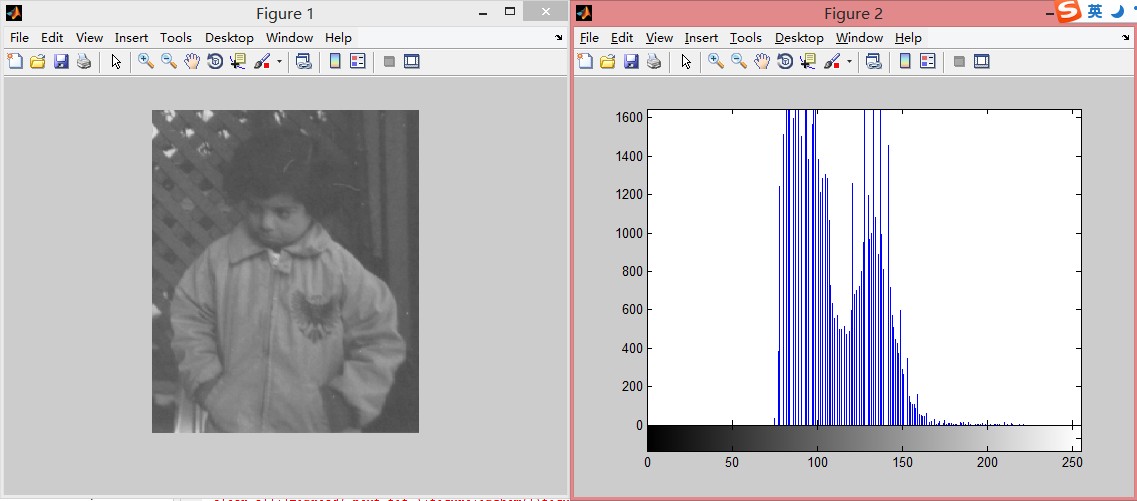
imshow(I,[low high])表示把图像I作为一幅灰度图像来显示，[low high]指定了图像I的数据范围。图像中所有灰度级不超过low的像素显示为黑色，灰度级不低于high的像素显示为白色，灰度级处于限定范围内的像素按照其原来的灰度级显示。imshow(RGB)用于显示真彩色（RGB）图像。imshow(BW)用于显示BW（二值）图像。imshow(X,map)用于显示引索色图像X，map是与其相关的颜色映射表。imshow(filename)用于显示存储文件格式filename。himage=imshow(…)，himage表示被显示图像的句柄。imshow(…param1,val1,param2,val2,…)用于显示指定其参数名及参数值的图像。

1. image(C)\image(x,y,C)\image(x,y,C,’PropertyName’,Property Value,…)\image(‘PropertyName’,Property Value,…)\handle=image(…)说明：image函数也可以用于彩色图像的显示。Eg：clear all;load mandrill figure(‘color’,’k’) image(X) colormap(map) axis off axis image

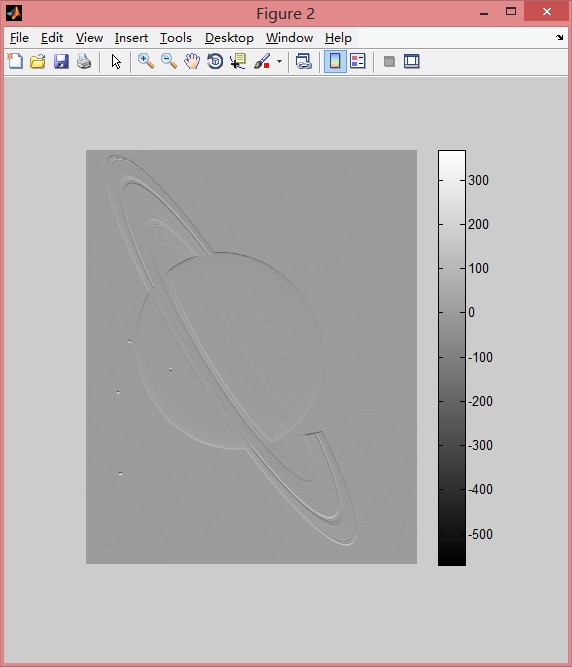


1. imfinfo提供了用于查询图像图像文件的信息函数。Info=imfinfo(filename,fmt)\info=imfinfo(filename)\info=imfinfo(URL,…)该函数所获得的信息依赖于文件类型的不同而不同，但至少包含以下类容：filename（文件名）、filemoddata文件最后一次修改时间、filesize文件大小，单位为字节（Byte）、format文件格式（MATLAB所支持的文件格式）、formatversion文件格式的版本号、width图像的宽度、height图像的高度。BitDepth每个像素的位数、colortype图像类型，即RGB图像、灰度图像还是引索图像。Eg：pathtool（设置好工作路径之后就可以引用了）info=imfinfo(‘girl.png’)
2. 直方图imhist函数的实现，调用如下：imhist(I)绘制灰度图像的直方图\imhist(I,n)绘制n个离散的灰度级图像I的直方图\imhist(X,map)显示引索图像的直方图\[counts,x]=imhist(…)返回图像直方图的计数counts及其计算直方图的位置x。

eg：clear all;I=imread(‘pout.tif’);figure;imshow(I)figure;imhist(I)结果如下图，右图中的的纵坐标为灰度值的递增，纵坐标为此种灰度条件下在图片中出现的频率。



1. 添加色轴可以采用colorbar函数来实现，添加色轴来表示不同位置处不同颜色所对应的数值。调用格式：colorbar\colorbar(‘vert’)\colorbar(‘horiz’)\colorbar(h)\h=colorbar(…)\colorbar(…,’peer’,axes\_handle)其中colorbar(‘vert’)和colorbar(‘horiz’)分别指定了色轴的显示方式为垂直与水平，默认值为垂直(‘vert’)。colorbar(h)将颜色调放置到指定的坐标轴h上，h为句柄。h=colorbar(…)返回颜色坐标轴的句柄。Eg：过滤一幅uint8的图像，产生一些超出[0,255]范围的数据，显示过滤后的图像并为图像添加色轴。clear all;RGB=imread(‘saturn.png’);I=rgb2gray(RGB);h=[1 2 1;0 0 0;-1 -2 -1];I2=filter(h,I);imshow(I2,’DisplayRange’,[]);colorbar



1. 帧图像表示，在MATLAB中，支持HDF和TIFF两种类型的多帧图像文件格式。文件一旦被读入MATLAB平台中，多帧图像的显示帧数即由矩阵的第四维数值决定。指令：imshow函数单独显示每一图像帧。montage函数同时显示所有的图像帧。immovie函数将图像帧转化为电影。EG：1、显示灰度数组I的第5帧：

clear all;

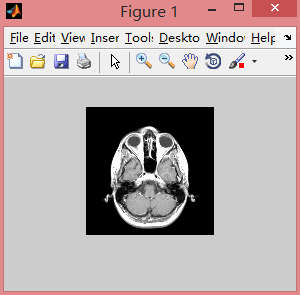
mri=uint8(zeros(128,128,1,27));

for frame=1:27

[mri(:,:,:,frame),map]=imread('mri.tif',frame);

end

imshow(mri(:,:,:,5),map);



2、多帧显示：要同时显示多帧图像，可以调用montage函数来实现，montage函数将对图形窗口进行划分，各帧显示在不同的显示区域中，如montage(filename):显示多帧指定MATLAB格式的图像。

montage(I):显示多帧灰度图像。

montage(X,map):显示多帧引索图像。

montage(…,param1,value1,param2,value2,…):根据指定的参数名及参数值显示多帧图像。

h=montage(…):返回多帧图像的句柄值。

Eg：clear all;

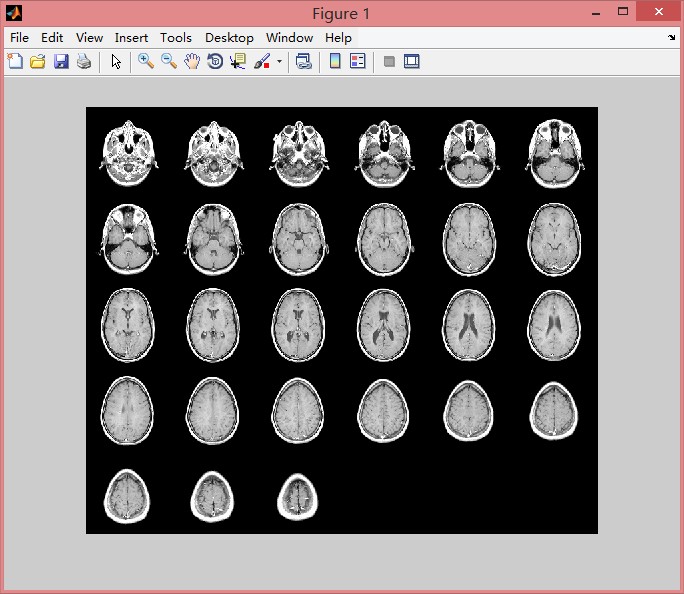
mri=uint8(zeros(128,128,1,27));

for frame=1:27

[mri(:,:,frame),map]=imread('mri.tif',frame);

end

montage(mri,map);



* 1. 显示动画：immovie函数来显示动画，调用 格式如下：

mov=immovie(X,map):以动画形式显示引索图像的多帧图像。

mov=immovie(RGB):以动画形式显示真彩色图像的多帧图像，对于创建好的电影，在MATLAB中还提供了movie函数进行播放，调用方法为：movie(M):在默认时间内显示引索图像的所有帧。movie(M,n):在指定的n秒内显示引索图像所有帧。movie(M,n,fps):fps为每秒指定播放几帧，默认为每秒播放12帧。movie(h,…):在指定坐标轴内播放动画。Eg：clear all;

mri=uint8(zeros(128,128,1,27));

for frame=1:27

[mri(:,:,:,frame),map]=imread('mri.tif',frame);

end

M=immovie(mri,map);

movie(M)

1. 多幅图像显示指令：subimage(X,map):用于显示引索色的图像。subimage(I):用于显示灰度图像。subimage(BW):用于显示二值图像。subimage(RGB):用于显示真彩色图像。subimage(x,y,…):表示将图像按指定坐标（x，y）显示，注意要设置横纵坐标范围。h= subimage(…):表示返回图像的句柄。Eg：

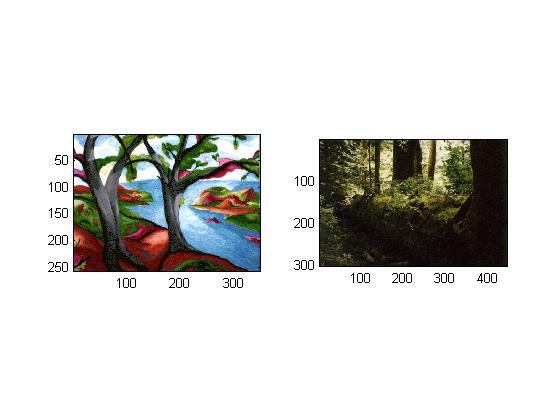
clear all;

load trees

[X2,map2]=imread('forest.tif');

subplot(1,2,1),subimage(X,map)

subplot(1,2,1),subimage(X2,map2)



1. 纹理映射：在MATLAB中纹理映射是一种将二维图像映射到三维图形表面的技术，通过转换颜色数据二维图像与三维图像的表面保持一致。MATLAB中的纹理映射是利用双线性变换算法来实现的。实现函数：warp(X,map):表示将引索色图像映射到矩阵平面区域上显示。warp(I,n):将灰度图像映射到矩形平面区域上显示。warp(BW):将二值图像映射到矩形平面区域上显示。warp(RGB):将真彩色图像映射到矩形平面区域上显示。warp(z,…):表示将图像映射到表面z上。warp(x,y,z,…):表示将图像映射到由（x，y，z）指定的表面上。h=warp(…):表示返回纹理映射后的图像句柄。Eg：clear all; I=imread('peppers.png'); [x,y,z]=sphere;

warp(x,y,z,I) set(gcf,'color','w') figure(2), imshow(I)

* 1. 图像处理高级应用基本函数：

（1）：regionprops用于测量图像区域属性（斑点分析）。调用如下:STATS=regioprops(L,properties):测量每个标签区域L的一系列属性。L可以是一个标签矩阵或者多维矩阵，当L是一个标签矩阵时，L中的正整数元素对应不同的区域。如果L中的元素值为1的话，则对应的区域为1，2对应2区域，依次类推。返回值STATS是一个结构数组，长度为max(L(:))。结构队列中的内容表示每个区域不同的测量法，如同属性指定的那样。STATS=regionprops(…,I,properties)：测量2-D或N-D灰度图像I中每一个标签区域的一系列属性。L是一个标签矩阵，标示中的区域和I的尺寸相同。首先将二值图像转化为标签矩阵：L=bwlabel(BW)或者L=double(BW)，其中BW为二值图像，L为转换后得到的标签矩阵。Eg：

clear all;

>> P=imread('ngc6543a.jpg');

>> I=rgb2gray(P);

>> L=bwlabel(I);

>> s=regionprops(L,'centroid');

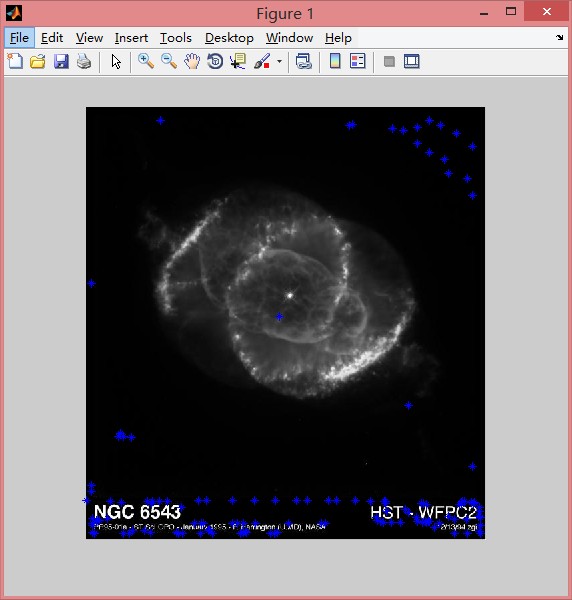
>> centroids=cat(1,s.Centroid);

>> imshow(I,[]);

hold on;

>> plot(centroids(:,1),centroids(:,2),'b\*')

>> hold off



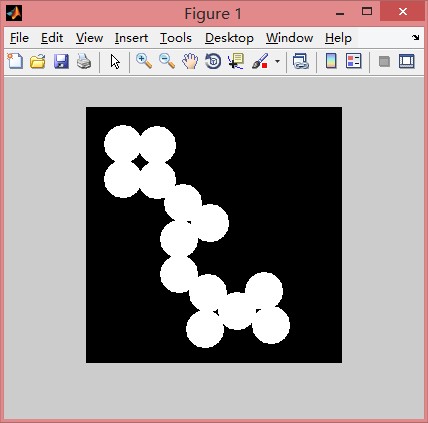
（2）：图像的类型：

1、引索图像：引索图像是一种把向世俗值直接作为RGB调色板的图像。在MATLAB中，索引图像包含一个数据矩阵X和一个颜色映射矩阵map。其中，数据矩阵可以是uint8、uint16或双精度型的，颜色映射矩阵map是一个mx3的数据阵列，其中每一个值都是[0,1]之间的双精度浮点型数据，map矩阵的每一行分别表示红色，绿色和蓝色的颜色值索引图像可以把像素直接映射为调色板数值，每一个像素的颜色通过map的下标来获得，如值1指向矩阵map中的第一行，值2指向第二行，以此类推。

2、灰度图像：灰度图像就是只有强度信息，而没有颜色信息的图像，灰度图像保存在一个矩阵中，矩阵的每一个元素代表一个像素点，矩阵可以是双精度型，其值域为[0,1]；也可以是uint8型，其值域为[0,256]。矩阵的每个元素值代表不同的亮度或灰度级，亮度值为0，表示黑色，亮度值为1（或uint8类型的256），表示白色。I=imread(‘moon.tif’);imagesc(I,[0,1]);colormap(gray)

3、RGB图像：RGB图像又称为真色彩图像，用R、G、B分别表示红、绿、蓝3种不同的颜色，通过三基色可以合成出任何颜色。RGB图像就是利用R、G、B这三个分量来表示图像中一个像素的颜色，所以对于一个尺寸为mXn的彩色图像来说，在MATLAB中存在一个mXnX3的多为数据数组，其中数组中的元素定义了图像中每一个像素的红，绿，蓝颜色值。MATLAB的RGB矩阵可以是双精度浮点型，8位或16位无符号整型。在一个双精度型的RGB矩阵中，每一个颜色分量都是一个[0,1]范围内的数值。颜色分量为（0,0,0）的像素显示为黑色；颜色分量为（1,1,1）的像素显示为白色。每一个像素的三个颜色分量都存储在矩阵的第三维中。如像素（8,5），它的红、绿、蓝颜色值分别保存在元素RGB（8,5,1）、RGB（8,5,2）、RGB（8,5,3）中。显示一幅RGB图像的代码为：RGB=imread('girl.tif'); image(RGB)

4、二值图像：表示二值图像的二维矩阵仅由0和1构成。二值图像可以看成是一个仅包含黑与白的特殊灰度图像，或是仅有两种颜色的引索图像。二值图像可以保存为双精度或uint8类型的数组，显然，使用uint8型更节省空间，在图像处理工具箱中，任何一个返回二值图像的函数都是以uint8型逻辑数组来返回值的。显示一幅二值图像的代码为：BW=imread('circles.png');imshow(BW)



（3）：图像类型的转换：许多图像处理工作都对图像有着特殊的要求，比如要对一幅引索图像进行滤波，首先将其转化为真彩色图像，而直接滤波是毫无意义的。在MATLAB中，个图像类型之间的转换关系如图：

引索图像

灰度图像

数据矩阵

真彩图像

二值图像

* 1. 抖动算法图像转换：在显示设备有限的情况下，要显示有多种色彩的图像就需要使用其他技术，其中常用的一种就是彩色抖动。彩色抖动的工作原理与黑白抖动的原理相同，他们的基本思想都是通过牺牲空间分辨率来换取灰度或色彩分辨率。假设使用4个像素来表示一个大像素，那么如果原来只有两种色彩，现在就有多种色彩。MATLAB提供了dither函数实现图像的抖动。调用格式如下：

X=dither(RGB,map):表示将真彩色图像RGB按指定的色图map抖动成索引图像X，但map不能超过65536种颜色。

X=dither(RGB,map,Qm,Qe):表示以定义Qm和Qe两个参数来将真彩色图像RGB转换成引索图像X，参数Qm定义了从转换色图map的每个颜色轴量化的位，参数Qe定义从颜色空误差计算的量化的为。如果Qe<Qm，则不能完成返回X的转换，默认值为Qm=5，Qe=8。

BW=dither(I):表示灰度图像I抖动成二值图像BW。

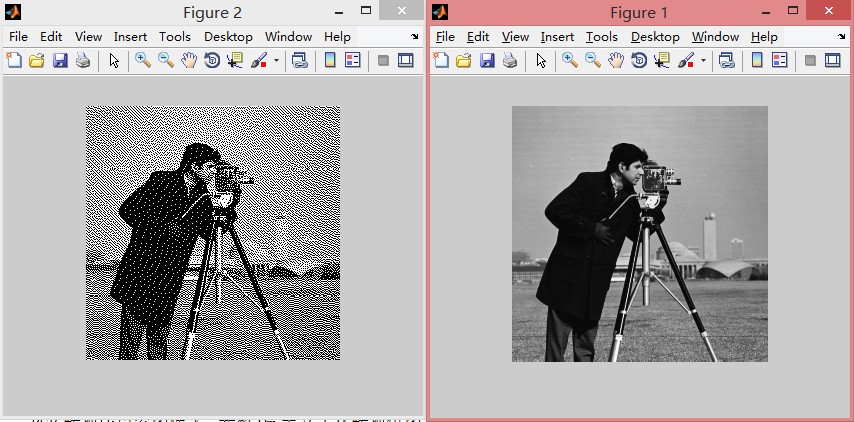
Eg：clear all;

I=imread('cameraman.tif');

BW=dither(I);

imshow(I);

figure,imshow(BW)



* 1. 灰度图像转化为二值图像：MATLAB还提供了gray2ind函数将灰度图像或二值图像转化为引索图像。调用格式：[X,map]=gray2ind(I,n):表示按指定的灰度级数n将灰度图像I转化成引索图像X，n的默认值是64。[X,map]=gray2ind(BW,n)表示按照指定的灰度级数n将二值图像BW转化成引索图像X，n的默认值为2.参数n的范围为1~65536。

Eg：

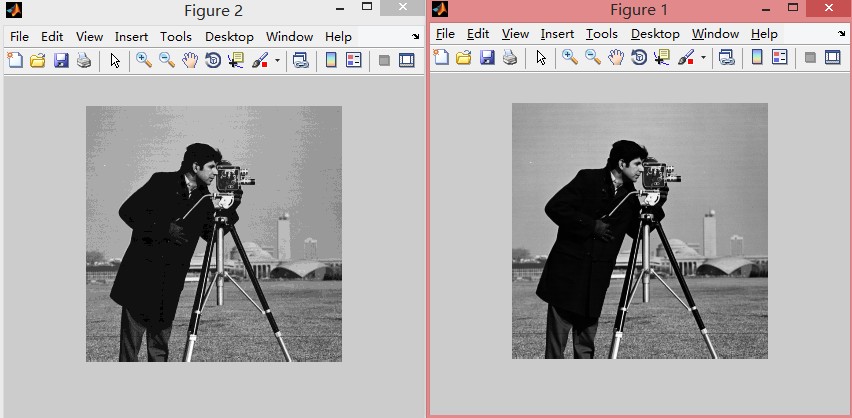
clear all;

I=imread('cameraman.tif');

[X,map]=gray2ind(I,16);

imshow(I,[]);

figure,imshow(X,map);



* 1. 阈值法图像转换为二值图像：im2bw函数通过设置亮度将阈值灰度、真彩、引索图像转化为二值图像。BW=im2bw(I,level):表示将灰度图像换成二值图像（white与black）。BW=im2bw(X,map,level):表示将引索图像X（色图map）转换成二值图像，BW=im2bw(RGB,level):表示将真彩色图像RGB转换成二值图像（white与black）。Lever为归一化的阈值，取值在[0,1]区间内。Level可以由函数graythresh(I)计算得到。Eg:

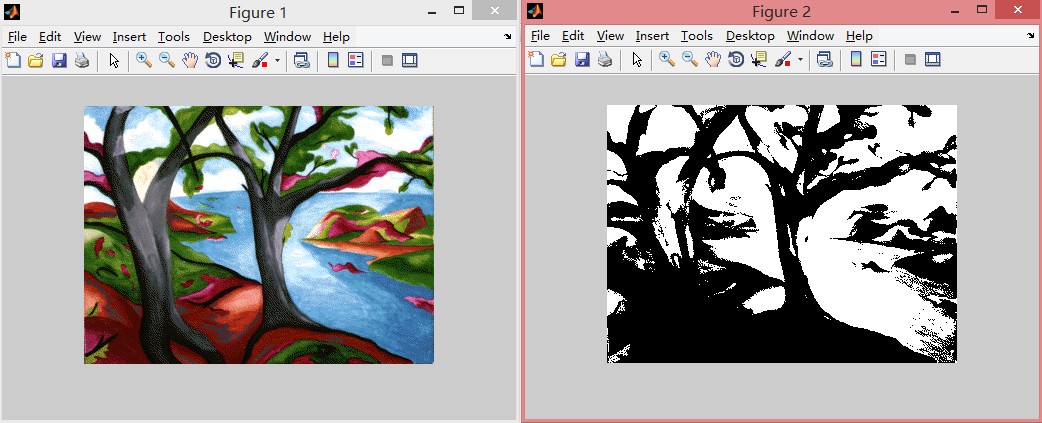
clear all;

load trees

BW=im2bw(X,map,0.4);

imshow(X,map)

figure,imshow(BW)



* 1. 引索图像转化为RGB图像，RGB=ind2rgb(X,map):将矩阵X及相应的颜色图map转换成真彩色图像RGB。实际实现时，就是产生一个三维数组，然后将引索图像中颜色图的颜色值赋给三维数组。输入的图像X可以是双精度类型或uint8类型，输出图像RGB为双精度类型。Eg：

clear all;

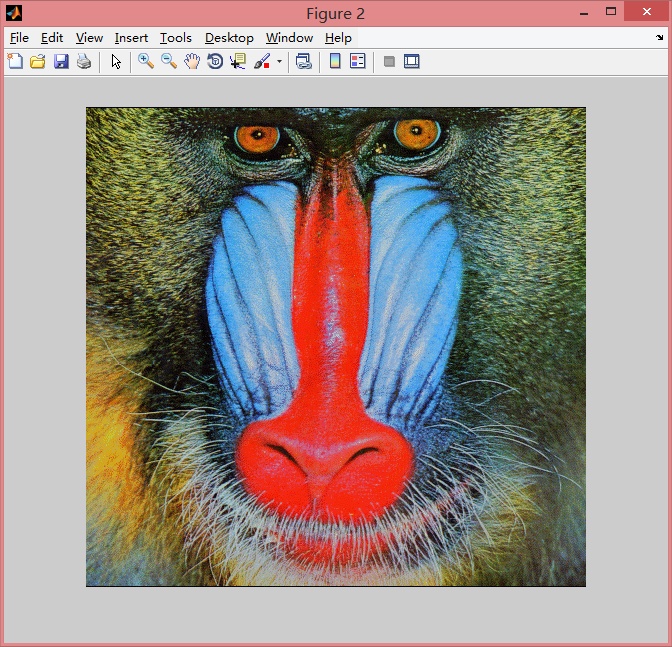
load mandrill

image(X)

colormap(map)

I=ind2rgb(X,map);

figure,imshow(I)



* 1. 数据矩阵转换为灰度图像：MATLAB中提供了mat2gray函数用于将一个数据矩阵转换成一幅灰度图像。其调用格式如下。I=mat2gray(A,[amin amax]):表示按照指定的取值区间[amin，amax]将数据矩阵A转换为灰度图像I，amin为灰度最暗值（0值），amax为灰度最亮值（1值）。I=mat2gray（A）：MATLAB自动把矩阵A中的最小值设定为amin，最大值设定为amax。输入图像A和输出图像I都是double类型。Eg：用Sobel算子对图像进行滤波，并将滤波后的数据矩阵转换为灰度图像。实现程序如下：

clear all;

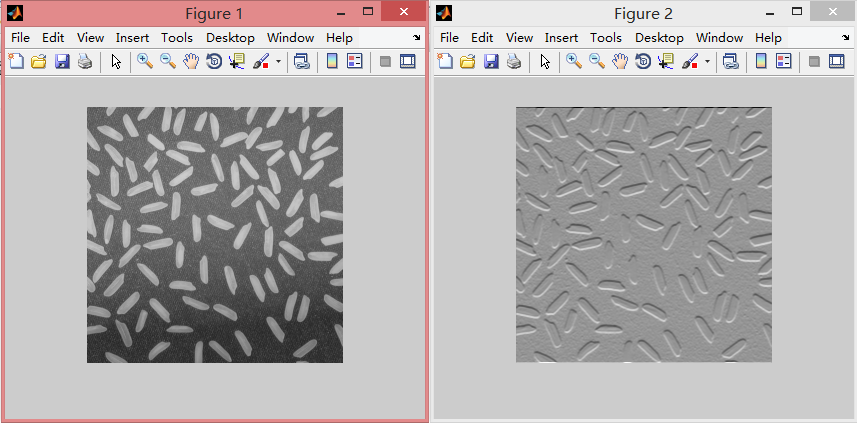
I=imread('rice.png');

J=filter2(fspecial('sobel'),I);

K=mat2gray(J);

imshow(I);

figure,imshow(K)



* 1. 真彩色图像转换为灰度图像:

在MATLAB中采用rgb2gray指令来将真彩色图像转换为灰度图像，指令方法如下：I=rgb2gray(RGB):表示将RGB图像转换为灰度图像I。

Newmap=rgb2gray（map）：表示将彩色图map转换为灰度图像。Eg1：

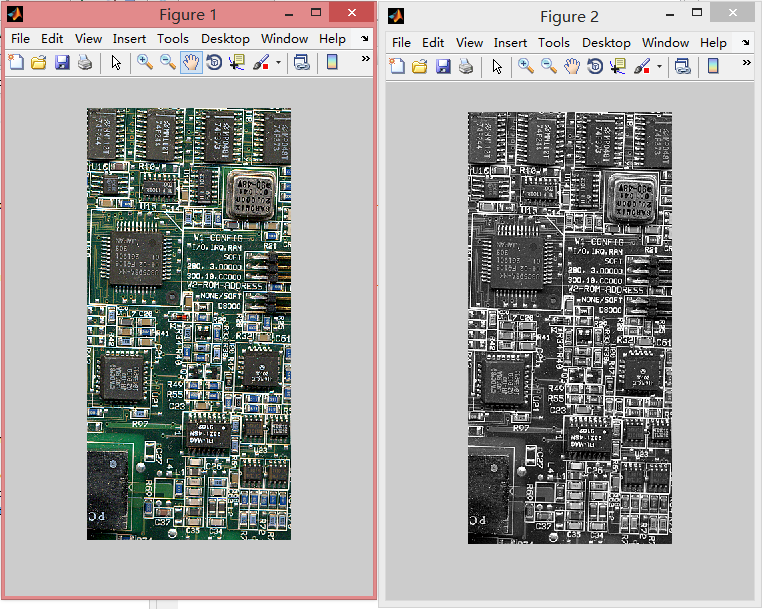
clear all;

I=imread('board.tif');

J=rgb2gray(I);

imshow(I);

figure,imshow(J)



Eg2:将彩色颜色图转换为灰度颜色图：imapprox指令

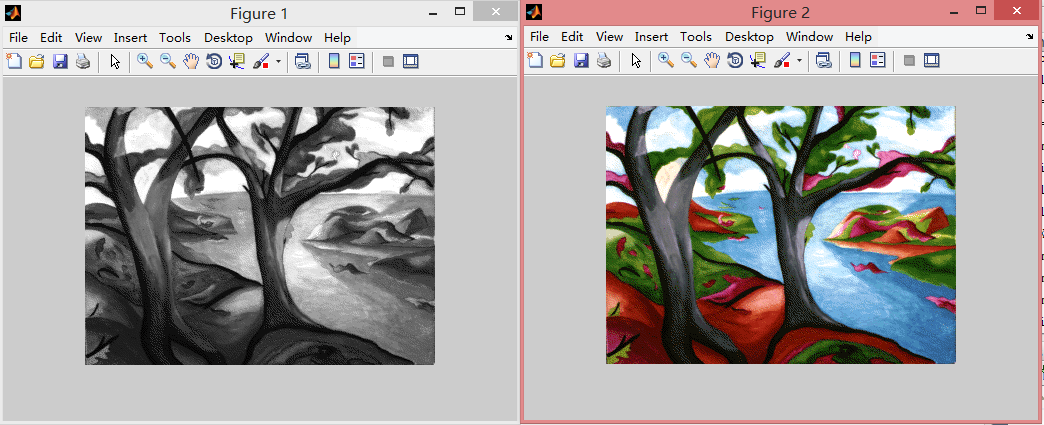
clear all;

[X,map]=imread('trees.tif');

gmap=rgb2gray(map);

imshow(X,gmap)

figure,imshow(X,map)



* 1. 图像褪色处理：eg：

clear all;

load mandrill;

figure('color','k')

image(X)

colormap(map)

disp('褪色前的图像大小：')

size(map)

axis off

axis image

figure('color','k')

[Y,newmap]=imapprox(X,map,16);

disp('褪色处理后的图像大小：')

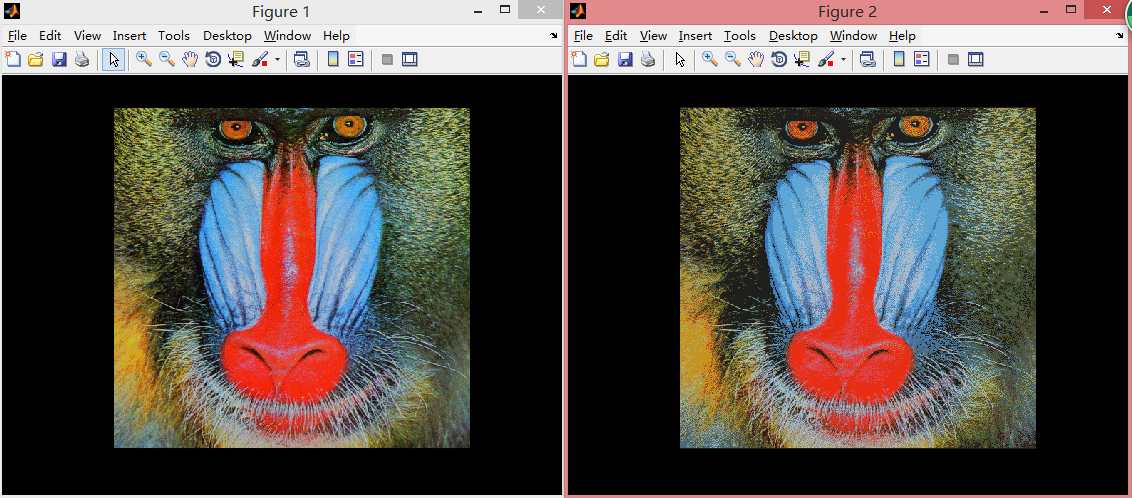
size(newmap)

image(Y)

colormap(newmap)

axis off

axis image



* 1. 图像的线性运算：clear all;

I=imread('rice.png');

J=I\*0.45+55;

rice2=uint8(J);

subplot(1,2,1);imshow(I);

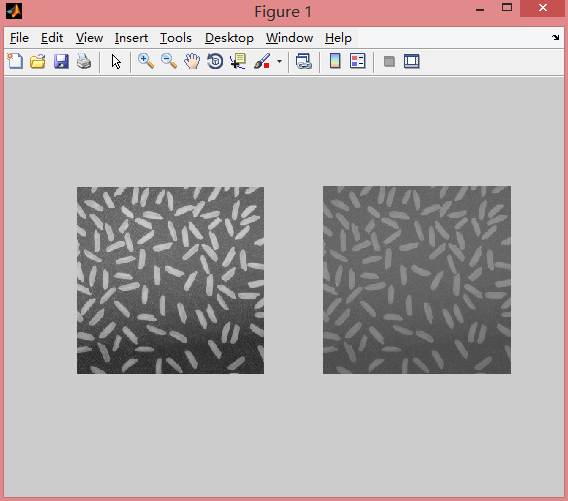
xlabel('a)原始图像')；

xlabel('a)原始图像')；

xlabel('a)原始图像');

subplot(1,2,2);imshow(rice2);

xlabel('b)点运算后的图像')



|  |  |  |
| --- | --- | --- |
| 函数名 | 说明 | 例子 |
| imabsdiff | 两幅图像的绝对差值 |  |
| imadd | 两幅图像的点点加法 | clear all;X=uint8([255 0 75;44 225 100]);Y=uint8([50 50 50;50 50 50]); Z=imadd(X,Y) |
| imcomplement | 一幅图像的反色图像 |  |
| imdivide | 两幅图像之间的除法 |  |
| immultiply | 两幅图像之间的乘法 | clear all;I=imread('moon.tif');  I16=uint16(I);J=immultipiy(I16,I16);  J=immultiply(I16,I16);  subplot(1,2,1);imshow(I);  subplot(1,2,2);imshow(J) |
| imsubtract | 两幅图像的代数减法 | clear all;I=imread('rice.png');  subplot(1,2,1);imshow(I);  xlabel('a')Iq=imsubtract(I,50);  subplot(1,2,2);imshow(Iq)xlabel('b') |
| imlincomb | 两幅图像的线性组合 |  |
| imnoise | 噪声添加函数 | J=imnoise(I,type)  J=imnoise(I,type,parameter）  J=imnoise(I,’gaussian’,m,v)  J=imnoise(I,’localvar’,V）  J=imnoise(I,’poisson’）  J=imnoise(I,’salt & pepper’,d） |

* 1. 图像增强：

1. 采用空间滤波增强：

图像平滑方法：1、噪声消除法2、邻域平均法3、中值滤波法4、梯度倒数加权5、选择式掩膜平滑法

图像锐化方法：1、梯度法2、Laplacian算子滤波法3、掩膜匹配法4、统计差法

1. 频域增强：步骤：通过进行图像的频域变换，得到图像的图像信息，在通过处理图像的频域信息，最后在使用逆变换变回空间域，从而实现频域增强。 对比度=亮度最大值/亮度最小值

图像处理课程讲解：

1.线性灰度变换：

（1）可以提升图像的对比度

（2）可以进行灰度倒置

（3）分段线性变换（提取出某一亮度条件下的区域图形）

2.非线性灰度变换：

（1）对数变换：g(x,y)=c=log(1+f(x,y)) 在后面的编程过程中要注意不要轻易的使用除法，因为可能分母为零，因此可以在分母上加上很小的数，从而避免分母为0.

（2）幂指数变换：幂指数＞1时：进行低灰度压缩高灰度拉升.幂指数可以取大于1、=1、小于1.