



第2章 MATLAB 图像处理工具箱



第2章 MATLAB图像处理工具箱

2.1 MATLAB简介

2.2 MATLAB常用的基本命令

2.3 图像处理工具箱简介

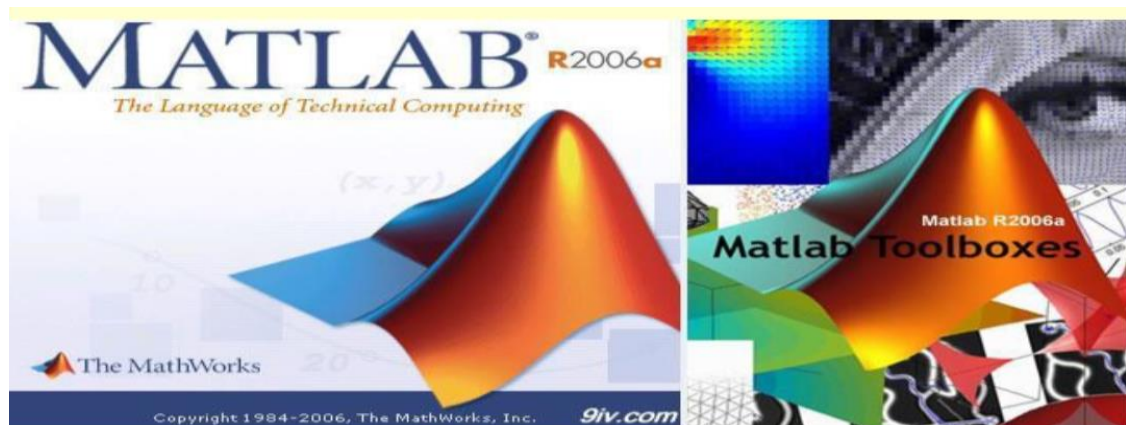
2.4 MATLAB中的图像类型及类型转换

2.5 图像文件的读写和查询

2.6 图像文件的显示



2.1 MATLAB简介



MATLAB是矩阵实验室（Matrix Laboratory）的简称，和Mathematical、Maple并称为三大数学软件。它在数学类科技应用软件中在数值计算方面首屈一指。MATLAB可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等，主要应用于工程计算、控制设计、信号处理与通讯、图像处理、信号检测、金融建模设计与分析等领域。



2.1 MATLAB简介

Matlab 工具箱

所谓Matlab工具箱就是一些M文件的集合，用户可以修改工具箱中的函数，更为重要的是用户可以通过编制M文件来任意地添加工具箱中原来没有的工具函数。此功能充分体现了matlab语言的开放性。

有三十多个工具箱大致可分为两类：**功能型工具箱**和**领域型工具箱**。功能型工具箱主要用来扩充MATLAB的符号计算功能、图形建模仿真功能、文字处理功能以及与硬件实时交互功能，能用于多种学科。而领域型工具箱是专业性很强的。如控制系统工具箱、信号处理工具箱(Signal Processing Toolbox)、财政金融工具箱(Financial Toolbox)等。



2.2 MATLAB常用的基本命令

1. 常用矩阵的生成

(1) 全0矩阵

$A = \text{zeros}(n)$: 生成 $n \times n$ 的全0矩阵。

$A = \text{zeros}(m, n)$: 生成 $m \times n$ 的全0矩阵。

$A = \text{zeros}(a1, a2, a3, \dots)$: 生成 $a1 \times a2 \times a3 \times \dots$ 的全0 矩阵。

$A = \text{zeros}(\text{size}(B))$: 生成与矩阵B大小相同的全0矩阵。



2.2 MATLAB常用的基本命令

1、常用矩阵的生成

(2) 全1矩阵

$A = \text{ones}(n)$: 生成 $n \times n$ 的全1矩阵。

$A = \text{ones}(m, n)$: 生成 $m \times n$ 的全1矩阵。

$A = \text{ones}([m, n])$: 生成 $m \times n$ 的全1矩阵。

$A = \text{ones}(a1, a2, a3, \dots)$: 生成 $a1 \times a2 \times a3 \times \dots$ 的全1矩阵。

$A = \text{onse}(\text{size}(B))$: 生成与矩阵B大小相同的全1矩阵。



2.2 MATLAB常用的基本命令

1、常用矩阵的生成

(3) 单位矩阵

$A = \text{eye}(n)$: 生成 $n \times n$ 的单位矩阵。

$A = \text{eye}(m, n)$: 生成 $m \times n$ 的单位矩阵。

$A = \text{eye}([m, n])$: 生成 $m \times n$ 的单位矩阵。

$A = \text{eye}(\text{size}(B))$: 生成与矩阵B大小相同的单位矩阵。



2.2 MATLAB常用的基本命令

1、常用矩阵的生成

(4) 均匀分布的随机矩阵

$A=\text{rand}(n)$ ：生成 $n \times n$ 的随机矩阵。

$A=\text{rand}(m, n)$ ：生成 $m \times n$ 的随机矩阵。

$A=\text{rand}([m, n])$ ：生成 $m \times n$ 的随机矩阵。

$A=\text{rand}(a1, a2, a3, \dots)$ ：生成 $a1 \times a2 \times a3 \times \dots$ 的随机矩阵。

$A=\text{rand}(\text{size}(B))$ ：生成与矩阵B大小相同的随机矩阵。

说明： rand 函数产生0、1之间均匀分布的随机数。



2.2 MATLAB常用的基本命令

1、常用矩阵的生成

(5) 正态分布的随机矩阵

$A = \text{randn}(n)$: 生成 $n \times n$ 的随机矩阵。

$A = \text{randn}(m, n)$: 生成 $m \times n$ 的随机矩阵。

$A = \text{randn}([m, n])$: 生成 $m \times n$ 的随机矩阵。

$A = \text{randn}(a1, a2, a3, \dots)$: 生成 $a1 \times a2 \times a3 \times \dots$ 的随机矩阵。

$A = \text{randn}(\text{size}(B))$: 生成与矩阵 B 大小相同的随机矩阵。

说明: rand 函数产生 -1 、 1 之间均匀分布的随机数。



2.2 MATLAB常用的基本命令

2、简单矩阵的生成

(1) 直接输入矩阵元素

- 矩阵每一行的元素必须用空格或逗号分开；
- 在矩阵中，采用分号或回车表明每一行的结束；
- 整个输入矩阵必须包含在方括号中。



2.2 MATLAB常用的基本命令

生成一个 3×3 的矩阵只要输入：

$$A=[1, 4, 7; 2, 2, 4; 3, 6, 2]$$

输出结果如下：

A=

1	4	7
2	2	4
3	6	2



2.2 MATLAB常用的基本命令

2、简单矩阵的生成

(2) 从外部数据文件调入矩阵元素

用MATLAB生成的矩阵存储成二进制文件或包含数值数据的文本文件可以生成矩阵。文本文件中，数据必须排成一个数据表，数据之间用空格分隔，文件的每行包含矩阵的一行，并且每一行的元素个数必须相等。



2.2 MATLAB常用的基本命令

例如： 如有一个名为dad. dat的文件，包含以下数据：

4	5	2	9
5	9	6	7
4	4	6	8
6	9	5	1

用Matlab将此文件的数据调入工作空间并生成变量dad。语句为：

```
load dad. dat      %将dad. dat中的内容调入工作空间  
dad                %显示变量
```



2.2 MATLAB常用的基本命令

输出结果：

`dad=`

4	5	2	9
5	9	6	7
4	4	6	8
6	9	5	1

说明：采用本方法可以创建和保存矩阵的大小没有限制，还可以将其他程序生成的矩阵直接调入Matlab中进行处理。



2.2 MATLAB常用的基本命令

2、简单矩阵的生成

(3) 利用用户文件创建的M文件矩阵

用户可以使用M文件生成自己的矩阵，M文件是一种包含Matlab代码的文本文件，这种文件的扩展名为“.m”，所包含的内容就是把在MATLAB的命令行上键入的矩阵生成命令存入一个文件。



2.2 MATLAB常用的基本命令

2、简单矩阵的生成

(4) 利用小矩阵生成大矩阵

例：用矩阵A生成大矩阵B。

$$A =$$

$$\begin{bmatrix} 4 & 5 & 2 & 9 \\ 5 & 9 & 6 & 7 \\ 4 & 4 & 6 & 8 \\ 6 & 9 & 5 & 1 \end{bmatrix}$$

$$B = [A \quad A+32; \quad A+48 \quad A+16]$$



2.2 MATLAB常用的基本命令

输出结果:

B =

4	5	2	9	36	37	34	41
5	9	6	7	37	41	38	39
4	4	6	8	36	36	38	40
6	9	5	1	38	41	37	33
52	53	50	57	20	21	18	25
53	57	54	55	21	25	22	23
52	52	54	56	20	20	22	24
54	57	53	49	22	25	21	17



2.2 MATLAB常用的基本命令

2、向量的生成

(1) 利用冒号“:”生成向量

冒号“:”适用下列格式生成均匀等分向量:

- $x=j:k$
- $x=j:i:k$

例如: 输入语句

```
x1=1:6  
x2=1:0.5:3  
x3=6:-1:1
```

输出结果如下:

x1=

1 2 3 4 5 6

x2=

1.0000 1.5000 2.0000 2.5000 3.0000

x3=

6 5 4 3 2 1



2.2 MATLAB常用的基本命令

2、向量的生成

(2) 利用linspace函数生成向量

`linspace`函数生成线性等分向量，它的功能类似冒号算子，但是它直接给出元素的个数。从而给出各个元素的值。其格式如下：

- `x= linspace(a, b)`
- `x= linspace(a, b, n)`

例如：输入语句 `x= linspace(1.2, 5, 4);`
输出如下结果：

`x=`

1.2000 2.4667 3.7333 5.0000



2.2 MATLAB常用的基本命令

2、向量的生成

(3) 利用logspace函数生成向量

Logspace函数生成对数等分向量，直接给出元素个数，从而给出各个元素的值。其格式如下：

- $x = \text{logspace}(a, b)$
- $x = \text{logspace}(a, b, n)$
- $x = \text{logspace}(a, \pi)$

例如：输入语句 $x = \text{logspace}(1, 2, 4)$;

输出如下结果：

$x =$

10.0000 21.5443 46.4159 100.0000



2.4 MATLAB中的图像类型及类型转换

2.4.1 图像和图像数据

2.4.2 图像处理工具箱所支持的图像类型

2.4.3 MATLAB图像类型转换



2.4 MATLAB中的图像类型及类型转换

1、图像和图像数据

Double 型：64位浮点型，

优点是使用中不需要类型的转换。

Unit 8 型：无符号整数，优点是节省空间。

Matlab工具箱的大部分操作和函数，都不支持Unit8型，计算时需要转换成Double型。



2.4 MATLAB中的图像类型及类型转换

1、图像和图像数据

Unit8 到double的转换

图像类型	MATLAB语句
索引色	$B = \text{double}(A) + 1$
索引色或真彩色	$B = \text{double}(A) / 255$
二值图像	$B = \text{double}(A)$



2.4 MATLAB中的图像类型及类型转换

1、图像和图像数据

double到unit8 的转换

图像类型	MATLAB语句
索引色	$B = \text{unit8}(\text{round}(A-1))$
索引色或真彩色	$B = \text{unit8}(\text{round}(A*255))$
二值图像	$B = \text{logical}(\text{unit8}(\text{round}(A)))$



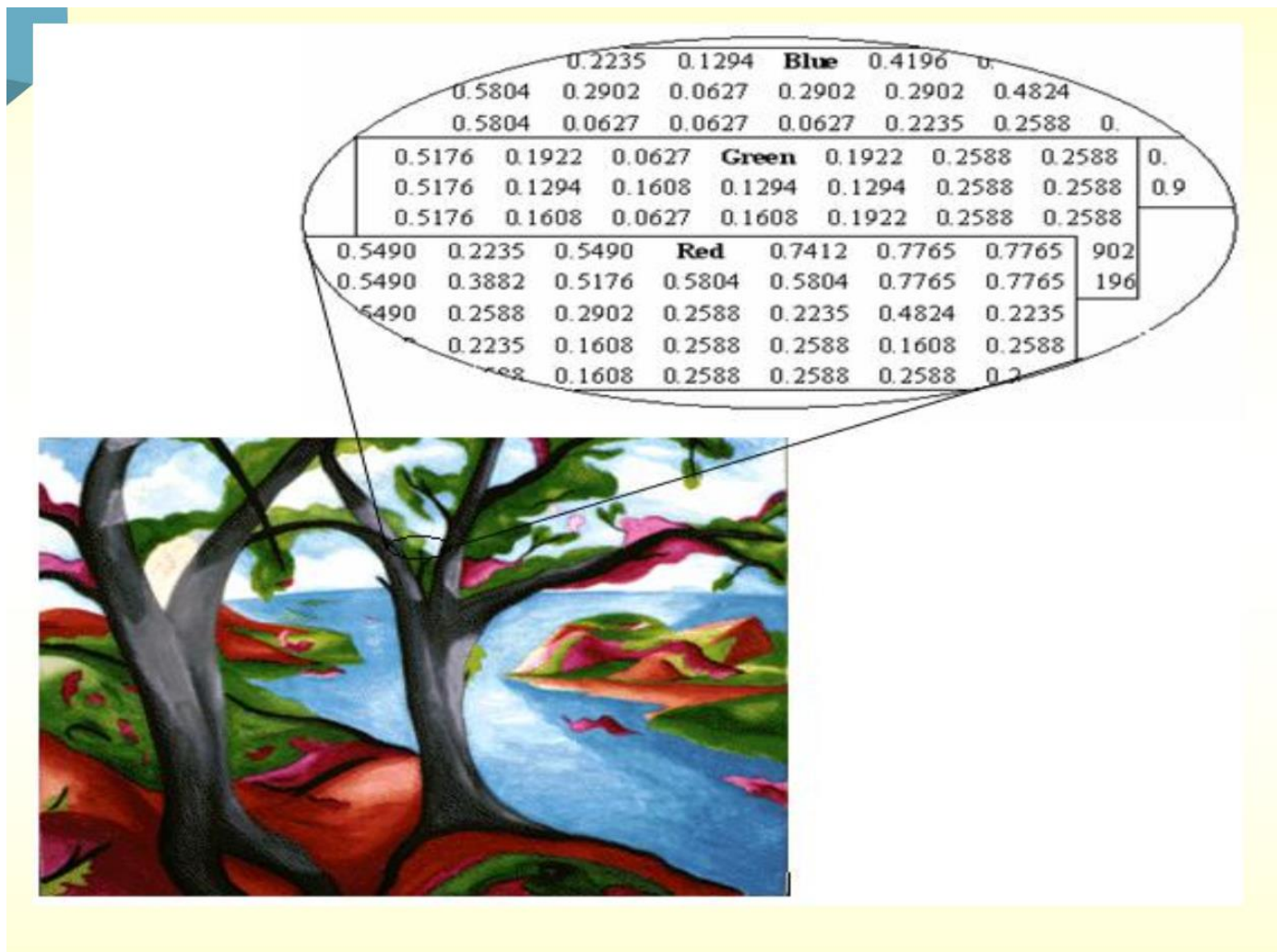
2.4 MATLAB中的图像类型及类型转换

2、工具箱所支持的图像类型

- 真彩色图像 R、G、B三个分量表示一个像素的颜色。对于一个尺寸为 $m \times n$ 的真彩色图像，其数据结构为 $m \times n \times 3$ 。
- 索引色图像 R包括两个结构：调色板、图像数据矩阵。
- 灰度图像 二维数据矩阵，double型，值域为 $[0, 1]$ ；unit8型，值域为 $[0, 255]$ 。
- 二值图像 二维数据矩阵，每个像素只取两个灰度值，0或1。
- 图像序列 四维数据矩阵，例如，5幅 400×300 真彩色序列，其大小为：
 $400 \quad 300 \quad 3 \quad 5$



真彩色图像的结构



真彩色图像的结构



索引色图像的结构



	12	21	40			
	14	17	21	21	53	53
5	8	5	8	10	30	15
1	15	18	31	31	18	16
	4	18	31	31	31	
	0		0		0	
	0.0627		0.0627		0.0314	
	0.2902		0.0314		0	
	0		0		1.0000	
	0.2902		0.0627		0.0627	
	0.3882		0.0314		0.0941	
	0.4510		0.0627		0	
	0.2588		0.1628		0.0627	

索引色图像的结构



灰度图像的结构

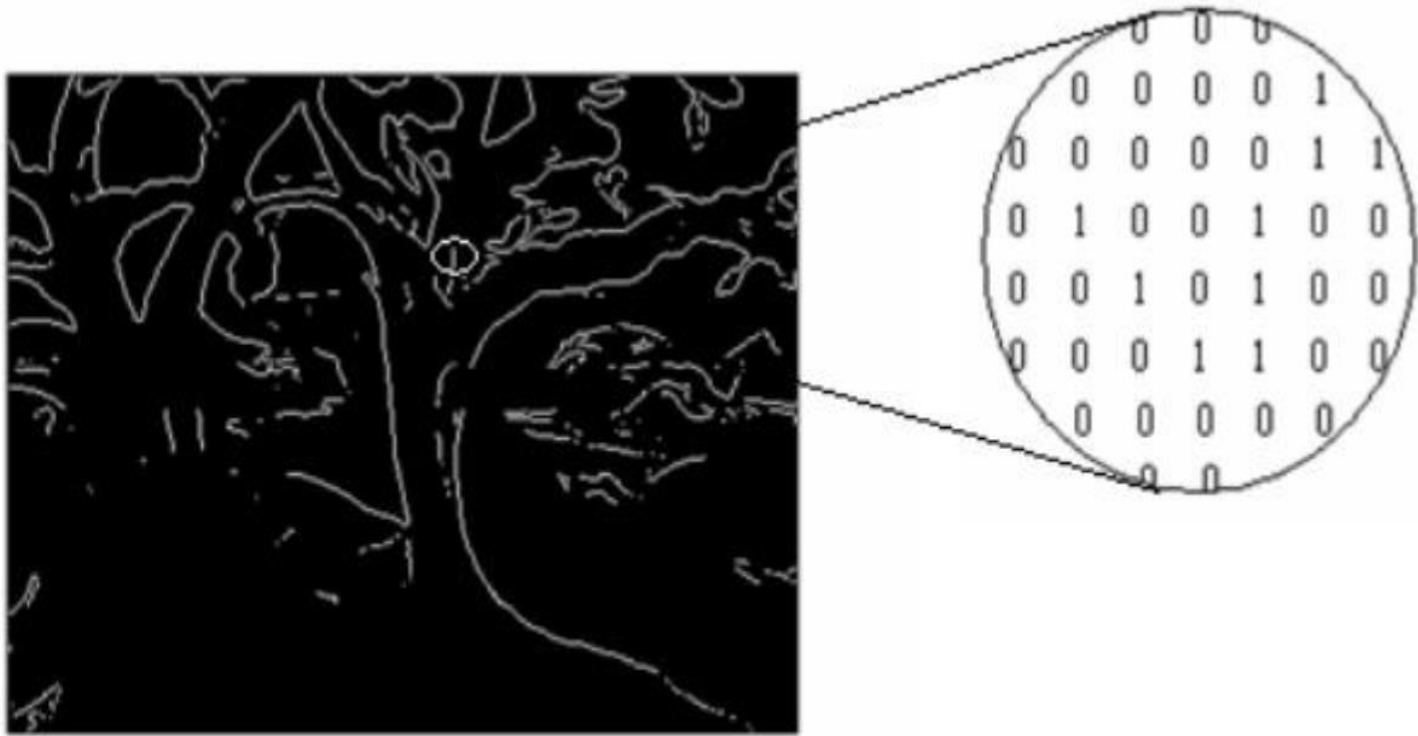
0.2051	0.2517	0.2826	0.3822	0.4391		
0.5342	0.2251	0.2563	0.2826	0.2826	0.4391	0.4391
0.5342	0.1789	0.1307	0.1789	0.2051	0.3256	0.2483
0.4308	0.2483	0.2624	0.3344	0.3344	0.2624	0.2549
0.3344	0.2624	0.3344	0.3344	0.3344	0.3344	



灰度图像的结构



二值图像的结构



二值图像的结构



2.4 MATLAB中的图像类型及类型转换

3、图像类型转换

图像抖动--dither函数

格式

`X=dither(I1, map)`

`bw=dither(I2)`

说明

`X=dither(I1, map)`将真彩色图像I1按指定的调色板map抖动成索引色图像X，`bw=dither(I2)`将灰度图像I2抖动成二值图像bw。

输入图像可以是double或uint8类型，输出图像若是二值图像或颜色种类不超过256的索引色图像，则是uint8类型，否则为double型。

例2-4-1



例:

```
clear all
```

```
load chess;           %使用索引图像chess.mat的颜色图
```

```
RGB=imread('autumn.tif'); %读入RGB图autumn.tif
```

```
subplot(2, 1, 1),
```

```
imshow(RGB)
```

```
II=dither(RGB, map); %通过抖动map中的颜色, 产生索引图像
```

```
subplot(2, 1, 2),
```

```
imshow(II)
```



仿真结果:



真彩色图像

索引图像





2.4 MATLAB中的图像类型及类型转换

3、图像类型转换

图像转换---gray2ind函数

格式 $[X, \text{map}] = \text{gray2ind}(I, n)$

说明 按指定的灰度级数 n 和调色板 map 将灰度图像 I 转换成索引色图像 X ， n 的缺省值为64。



例:

```
I=imread( 'autumn.tif' );
```

```
[I1, map1]=gray2ind(I, 128);
```

```
[I2, map2]=gray2ind(I, 16);
```

```
figure(1);
```

```
imshow(I);
```

```
figure(2);
```

```
imshow(I1, map1);
```

```
figure(3);
```

```
imshow(I2, map2);
```

原始图像



使用map1模版
生成的索引图像

使用map2模版
生成的索引图像





grayscale函数

功能：通过设定阈值将灰度图像转换成索引色图像

格式

$X = \text{grayscale}(I, n)$

$X = \text{grayscale}(I, v)$

说明

$X = \text{grayscale}(I, n)$ 将灰度图像 I 均匀量化为 n 个等级，然后转换为伪彩色图像 X 。

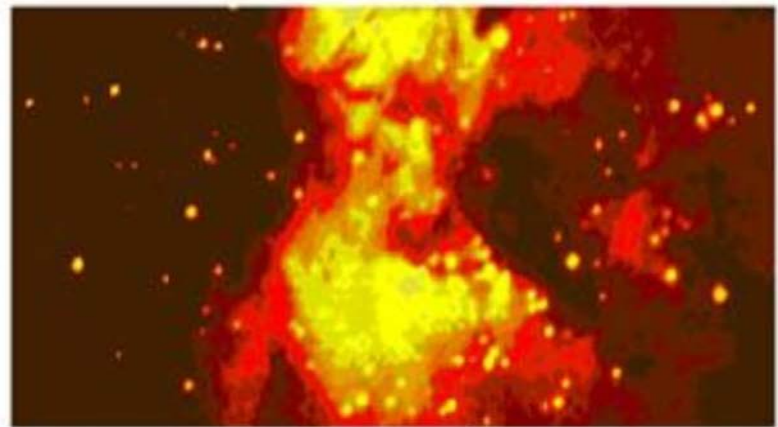
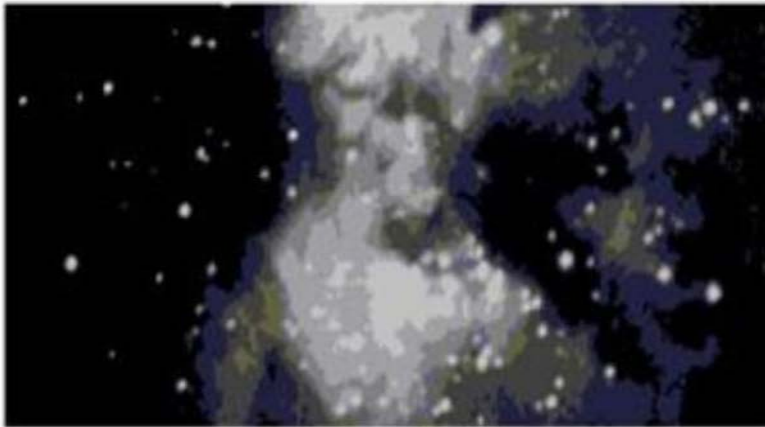
$X = \text{grayscale}(I, v)$ 按指定的阈值向量 v （起每一个元素都在0和1之间）对图像 I 的值域进行划分，而后转换成索引色图像 X 。

输入图像 I 可以是double或uint8类型。如果阈值数量小于则返回图像 X 的数据类型是uint8， X 的值域为 $[0, n]$ 或 $[0, \text{length}]$ 。否则，返回图像 X 为double类型，值域为 $[1, n+1]$ 或 $[1, \text{length}(v)+1]$ 。



例:

```
I=imread('ngc4024m.tif');  
X=grayslice(I,16);  
imshow(I)  
figure, imshow(X, hot(16))
```



灰度图像转换成索引色图像



im2bw函数

功能：阈值化方法将图像转化为二值图像。

$BW = \text{im2bw}(I, \text{level})$ 索引色图像转化为二值图像。

格式

· $BW = \text{im2bw}(X, \text{map}, \text{level})$ 归一化阈值，取值

$BW = \text{im2bw}(\text{RGB}, \text{level})$ RGB图像转化为二值图像。

$BW = \text{im2bw}(I, \text{level})$ 、 $BW = \text{im2bw}(X, \text{map}, \text{level})$

和 $BW = \text{im2bw}(\text{RGB}, \text{level})$ 分别将灰度图像、索引色

说明 图像和真彩色图像I二值化为图像BW。level是归一化阈值，取值在0,1之间。

输入图像可以是double或uint8类型，输出图像为uint8类型。



例:

```
load trees  
BW=im2bw(X, map, 0.4);  
imshow(X, map)  
figure, imshow(BW)
```



原始图像



二值图像



ind2gray函数

功能：索引图像转化为灰度图像。

格式

$I = \text{ind2gray}(X, \text{map})$

说明

$I = \text{ind2gray}(X, \text{map})$ 将具有调色板map的索引色图像转换成灰度图像I。



```
clear all
```

```
load chess;
```

```
I=ind2gray(X, map);
```

```
figure(1);
```

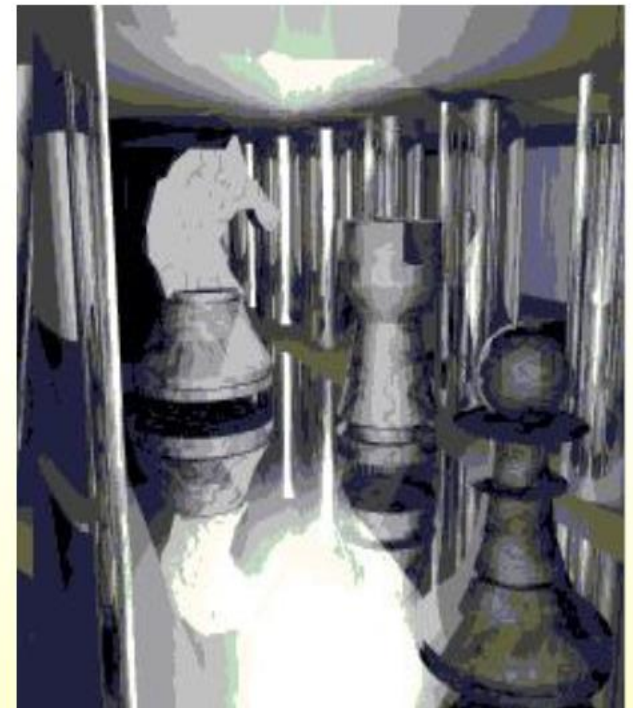
```
imshow(X, map);
```

```
figure(2);
```

```
imshow(I)
```



索引图像



灰度图像



ind2rgb函数

功能：索引图像转化为RGB图像。

格式

`RGB=ind2rgb(X, map)`

说明

将具有调色板map的索引色图像X转换成真彩色图像RGB，输入图像X可以是double或uint8类型，输出图像RGB为double类型。



mat2gray函数

将矩阵转化为灰度图像。

格式 `I= mat2gray(A, [amin amax])`
`I= mat2gray(A)`

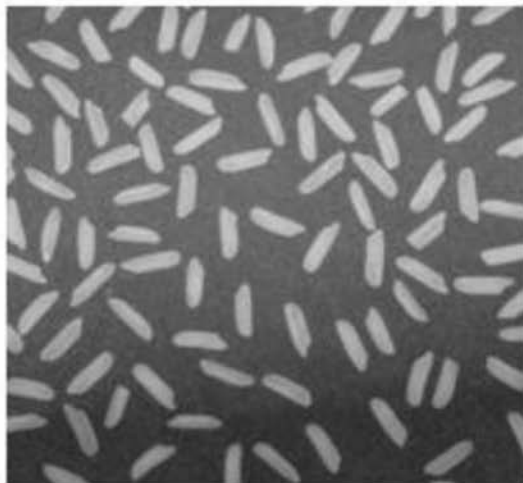
说明 `I= mat2gray(A, [amin amax])`按指定的取值区间[amin amax]将数据矩阵A转化为灰度图像I，amin对应灰度0（最暗），amax对应1（最亮）。如果不指定区间[amin amax]，则Matlab自动将A阵中的最小元设为amin，最大元设为amax。

例2-4-7



例:

```
I=imread('rice.tif');  
J=filter2(fspecial('sobel'),I);  
K=mat2gray(J);  
imshow(J)  
figure, imshow(K)
```



将数据图像转换成灰度图像



rgb2gray函数

功能：RGB图像转化为灰度图像。

`I= rgb2gray (RGB)`

格式

`newmap= rgb2gray (map)`

说明

`I= rgb2gray (RGB)` 将真彩色图像RGB转换成灰度图像I。
`newmap= rgb2gray (map)` 将彩色调色板map转换成灰度调色板。

如果输入的是真彩色图像，则可以是uint8或double类型，输出图像I与输入图像类型相同。如果输入的是调色板，则输入、输出都是double类型。

例2-4-4



```
clear all  
I=imread('autumn.tif');  
figure(1);  
imshow(I);  
I1=rgb2gray(I);  
figure(2);  
imshow(I1);
```



RGB图像



灰度图像



rgb2ind函数

功能：RGB图像转化为索引图像。

格式

$\text{RGB} = \text{rgb2ind}(\text{X}, \text{map})$

说明

使用调色板map将真彩色图像转换成索引色图像。



2.5 图像文件的读写和查询

imread函数

其语法如下：

```
A=imread(filename, fmt)
```

```
[X, map]=imread(filename, fmt)
```

```
[...]=imread(filename)
```

```
[...]=imread(filename, idx) （只对TIF格式的文件）
```

```
[...]=imread(filename, ref) （只对HDF格式的文件）
```

其中第一种为最常用的形式。

例如读取图像ngc6543a, jig的代码如下：

```
RGB=imread( 'ngc6543a. jpg' )
```



2.5 图像文件的读写和查询

imwrite函数

其语法如下：

```
imwrite(A, filename, fmt)
```

```
imwrite(X, map, filename, fmt)
```

```
imwrite(..., filename)
```

```
imwrite(..., parameter, value)
```

例如，我们可以通过下面的语句来实现保存图像：

```
imwrite(X, 'flowers.hdf' ...  
        'Compressio', 'none', ...  
        'WriteMode',  
        'append' ...)
```




2.5 图像文件的读写和查询

imfinfo函数

所获取的信息以来与文件类型的不同而不同。但是不管哪种类型的图像文件，至少包含下面的内容：

- 文件名。如果该文件不在当前路径下时，还包含该文件的完整路径。
 - 文件格式。
 - 文件格式的版本号
 - 文件修改时间。
 - 文件的字节大小。
 - 图像的宽度（像素）。
 - 图像的长度（像素）。
 - 每个像素的位数。
- 图像类型。即该图像是RGB(真彩)图像、强度（灰度）图像还是索引图像。



2.5 图像文件的读写和查询

例如:

我们在Matlab的命令行中输入以下语句,
查询文件lena.bmp的信息:

```
inf=imfinfo('lena.bmp')
```

回车执行后, 结果如下:

```
info =
```

```
    Filename: C:\fusion\.Lena.bmp'
```

```
    FileModDate: '10-Mar-2000 21:42:16'
```

```
    FileSize: 66616
```

```
    Format: 'bmp'
```

```
    FormatVersion: 'Version 3 (Microsoft Windows 3.x)'
```

```
    Width: 256
```

```
    Height: 256
```

```
    BitDepth: 8
```

```
    ColorType: indexed
```

```
    FormatSignature: 'BM'
```

```
    NumColormapEntries: 256
```



2.5 图像文件的读写和查询

Colormap: [256 3 double]

RedMask: []

GreenMask: []

BlueMask: []

ImageDataOffset: 1078

BitmapHeaderSize: 40

NumPlanes: 1

CompressionType: 'none'

BitmapSize: 0

HorzResolution: 2934

VertResolution: 2834

NumColorsUsed: 0

NumImportantColors: 0



2.6 图像文件的显示

2.6.1 图像的显示函数

2.6.2 不同类型的图像显示

2.6.3 直接从磁盘显示图像



2.6 图像文件的显示

image函数

语法格式为：

`image(C)`

`image(x, y, C)`

`image('PropertyName', Property Value, ...)`

`image('PropertyName', Property Value, ...)`

`handle=image(...)`

其中，`x`，`y`分别表示图像显示位置的左上角坐标，`C`表示所需显示的图像。函数`imagesc`与`image`函数类似，但是它自动标度输入数据。



2.6 图像文件的显示

例： 图像的左上角坐标为（10，10）。

```
load clown
```

```
image(10, 10, X)
```

```
colormap(map)
```

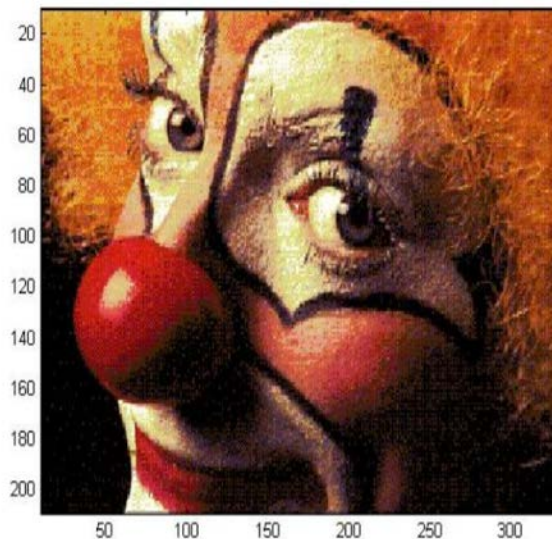


image 函数显示图像



2.6 图像文件的显示

imshow函数

语法格式为：

- `imshow(I, n)`
- `imshow(I, [low high])`
- `imshow(BW)`
- `imshow(X, map)`
- `imshow(RGB)`
- `imshow(..., display_option)`
- `imshow(x, y, A, ...)`
- `imshow filename`
- `h=imshow(...)`

第一和第二种调用格式用来显示灰度图像，其中n为灰度级数目，默认值为256。[low high]为图像数据的值域。



2.6 图像文件的显示

例:

```
I=imread('rice.png')
```

```
J=filter2([1 2;-1 -2], I)
```

%用模板 $\begin{bmatrix} 1 & 2 \\ -1 & -2 \end{bmatrix}$ 对图像滤波

```
imshow(I)
```

```
figure, imshow(J, [])
```

%由于滤波后图像灰度范围与滤波之前不同，所以用`[]`来作为参考



用imshow显示滤波前后的图像



2.6 图像文件的显示

2、不同类型图像文件的显示

- 1、索引图像及其显示
- 2、灰度图像及其显示
- 3、RGB图像及其显示
- 4、二进制图像及其显示