

一 图像的读写

1 imread

imread 函数用于读入各种图像文件，如：`a=imread('e:\w01.tif')`

注：计算机 E 盘上要有 w01 相应的.tif 文件。

2 imwrite

imwrite 函数用于写入图像文件，如：`imwrite(a,'e:\w02.tif','tif')`

3 imfinfo

imfinfo 函数用于读取图像文件的有关信息，如：`imfinfo('e:\w01.tif')`

二 图像的显示

1 image

image 函数是 MATLAB 提供的最原始的图像显示函数，如：

```
a=[1,2,3,4;4,5,6,7;8,9,10,11,12];
```

```
image(a);
```

2 imshow

imshow 函数用于图像文件的显示，如：

```
i=imread('e:\w01.tif');
```

```
imshow(i);
```

3 colorbar

colorbar 函数用显示图像的颜色条，如：

```
i=imread('e:\w01.tif');
```

```
imshow(i);
```

```
colorbar;
```

4 figure

figure 函数用于设定图像显示窗口，如：`figure(1);` /`figure(2);`

三 图像的变换

1 fft2

fft2 函数用于数字图像的二维傅立叶变换，如：

```
i=imread('e:\w01.tif');
```

```
j=fft2(i);
```

2 ifft2

ifft2 函数用于数字图像的二维傅立叶反变换，如：

```
i=imread('e:\w01.tif');
```

```
j=fft2(i);
```

```
k=ifft2(j);
```

3 利用 fft2 计算二维卷积

利用 fft2 函数可以计算二维卷积,如:

```
a=[8,1,6;3,5,7;4,9,2];
```

```
b=[1,1,1;1,1,1;1,1,1];
```

```
a(8,8)=0;
```

```
b(8,8)=0;
```

```
c=ifft2(fft2(a).*fft2(b));
```

```
c=c(1:5,1:5);
```

利用 conv2(二维卷积函数)校验, 如:

```
a=[8,1,6;3,5,7;4,9,2];
```

```
b=[1,1,1;1,1,1;1,1,1];
```

```
c=conv2(a,b);
```

四 模拟噪声生成函数和预定义滤波器

1 imnoise

imnoise 函数用于对图像生成模拟噪声, 如:

```
i=imread('e:\w01.tif');
```

```
j=imnoise(i,'gaussian',0,0.02);%模拟高斯噪声
```

2 fspecial

fspecial 函数用于产生预定义滤波器, 如:

```
h=fspecial('sobel');%sobel 水平边缘增强滤波器
```

```
h=fspecial('gaussian');%高斯低通滤波器
```

```
h=fspecial('laplacian');%拉普拉斯滤波器
```

```
h=fspecial('log');%高斯拉普拉斯 (LoG) 滤波器
```

```
h=fspecial('average');%均值滤波器
```

五 图像的增强

1 直方图

imhist 函数用于数字图像的直方图显示, 如:

```
i=imread('e:\w01.tif');
```

```
imhist(i);
```

2 直方图均化

histeq 函数用于数字图像的直方图均化, 如:

```
i=imread('e:\w01.tif');
```

```
j=histeq(i);
```

3 对比度调整

imadjust 函数用于数字图像的对比度调整, 如:

```
i=imread('e:\w01.tif');
```

```
j=imadjust(i,[0.3,0.7],[]);
```

4 对数变换

log 函数用于数字图像的对数变换, 如:

```
i=imread('e:\w01.tif');
```

```
j=double(i);
```

```
k=log(j);
```

5 基于卷积的图像滤波函数

filter2 函数用于图像滤波, 如:

```
i=imread('e:\w01.tif');
```

```
h=[1,2,1;0,0,0;-1,-2,-1];
```

```
j=filter2(h,i);
```

6 线性滤波

利用二维卷积 conv2 滤波, 如:

```
i=imread('e:\w01.tif');
```

```
h=[1,1,1;1,1,1;1,1,1];
```

```
h=h/9;
```

```
j=conv2(i,h);
```

7 中值滤波

medfilt2 函数用于图像的中值滤波，如：

```
i=imread('e:\w01.tif');
```

```
j=medfilt2(i);
```

8 锐化

(1) 利用 Sobel 算子锐化图像，如：

```
i=imread('e:\w01.tif');
```

```
h=[1,2,1;0,0,0;-1,-2,-1];%Sobel 算子
```

```
j=filter2(h,i);
```

(2) 利用拉氏算子锐化图像，如：

```
i=imread('e:\w01.tif');
```

```
j=double(i);
```

```
h=[0,1,0;1,-4,0;0,1,0];%拉氏算子
```

```
k=conv2(j,h,'same');
```

```
m=j-k;
```

六 举例

二维傅立叶变换和二维傅立叶反变换：

```
i=imread('e:\w01.tif');
```

```
figure(1);
```

```
imshow(i);
```

```
colorbar;
```

```
j=fft2(i);
```

```
k=fftshift(j);
```

```
figure(2);
```

```
l=log(abs(k));
```

```
imshow(l,[]);
```

```
colorbar
```

```
n=ifft2(j)/255;
```

```
figure(3);
```

```
imshow(n);
```

```
colorbar;
```

Matlab 中图像函数大全

图像增强

1. 直方图均衡化的 Matlab 实现

1.1 imhist 函数

功能：计算和显示图像的色彩直方图

格式：imhist(l,n)

imhist(X,map)

说明: `imhist(I,n)` 其中, n 为指定的灰度级数目, 缺省值为 256; `imhist(X,map)` 计算和显示索引色图像 X 的直方图, map 为调色板。用

`stem(x,counts)` 同样可以显示直方图。

1.2 `imcontour` 函数

功能: 显示图像的等灰度值图

格式: `imcontour(I,n)`, `imcontour(I,v)`

说明: n 为灰度级的个数, v 是用户指定所选的等灰度级向量。

1.3 `imadjust` 函数

功能: 通过直方图变换调整对比度

格式: `J=imadjust(I,[low high],[bottom top],gamma)`

`newmap=imadjust(map,[low high],[bottom top],gamma)`

说明: `J=imadjust(I,[low high],[bottom top],gamma)` 其中, γ 为校正量 r , $[low\ high]$ 为原图像中要变换的灰度范围, $[bottom\ top]$

指定了变换后的灰度范围; `newmap=imadjust(map,[low high],[bottom top],gamma)` 调整索引色图像的调色板 map 。此时若 $[low\ high]$ 和

$[bottom\ top]$ 都为 2×3 的矩阵, 则分别调整 R 、 G 、 B 3 个分量。

1.4 `histeq` 函数

功能: 直方图均衡化

格式: `J=histeq(I,hgram)`

`J=histeq(I,n)`

`[J,T]=histeq(I,...)`

`newmap=histeq(X,map,hgram)`

`newmap=histeq(X,map)`

`[new,T]=histeq(X,...)`

说明: `J=histeq(I,hgram)` 实现了所谓“直方图规定化”, 即将原图像 I 的直方图变换成用户指定的向量 $hgram$ 。 $hgram$ 中的每一个元素

都在 $[0,1]$ 中; `J=histeq(I,n)` 指定均衡化后的灰度级数 n , 缺省值为 64; `[J,T]=histeq(I,...)` 返回从能将图像 I 的灰度直方图变换成

图像 J 的直方图的变换 T ; `newmap=histeq(X,map)` 和 `[new,T]=histeq(X,...)` 是针对索引色图像调色板的直方图均衡。

2. 噪声及其噪声的 Matlab 实现

`imnoise` 函数

格式: `J=imnoise(I,type)`

`J=imnoise(I,type,parameter)`

说明: `J=imnoise(I,type)` 返回对图像 I 添加典型噪声后的有噪图像 J , 参数 $type$ 和

parameter 用于确定噪声的类型和相应的参数。

3. 图像滤波的 Matlab 实现

3.1 conv2 函数

功能：计算二维卷积

格式：C=conv2(A,B)

C=conv2(Hcol,Hrow,A)

C=conv2(...,'shape')

说明：对于 C=conv2(A,B)，conv2 的算矩阵 A 和 B 的卷积，若 [Ma,Na] = size(A), [Mb,Nb]=size(B)，则 size(C)=[Ma+Mb-1,Na+Nb-1];

C=conv2(Hcol,Hrow,A) 中，矩阵 A 分别与 Hcol 向量在列方向和 Hrow 向量在行方向上进行卷积；C=conv2(...,'shape') 用来指定 conv2

返回二维卷积结果部分，参数 shape 可取值如下：

》full 为缺省值，返回二维卷积的全部结果；

》same 返回二维卷积结果中与 A 大小相同的中间部分；

valid 返回在卷积过程中，未使用边缘补 0 部分进行计算的卷积结果部分，当 size(A)>size(B) 时，size(C)=[Ma-Mb+1,Na-Nb+1]

。

3.2 conv 函数

功能：计算多维卷积

格式：与 conv2 函数相同

3.3 filter2 函数

功能：计算二维线型数字滤波，它与函数 fspecial 连用

格式：Y=filter2(B,X)

Y=filter2(B,X,'shape')

说明：对于 Y=filter2(B,X)，filter2 使用矩阵 B 中的二维 FIR 滤波器对数据 X 进行滤波，结果 Y 是通过二维互相关计算出来的，其大

小与 X 一样；对于 Y=filter2(B,X,'shape')，filter2 返回的 Y 是通过二维互相关计算出来的，其大小由参数 shape 确定，其取值如下

：

》full 返回二维相关的全部结果，size(Y)>size(X)；

》same 返回二维互相关结果的中间部分，Y 与 X 大小相同；

》valid 返回在二维互相关过程中，未使用边缘补 0 部分进行计算的结果部分，有 size(Y)<size(X) 。

3.4 fspecial 函数

功能：产生预定义滤波器

格式：H=fspecial(type)

H=fspecial('gaussian',n,sigma)

高斯低通滤波器

H=fspecial('sobel')

Sobel 水平边缘增强滤波器

H=fspecial('prewitt')

Prewitt 水平边缘增强滤波器

H=fspecial('laplacian',alpha)

近似二维拉普拉斯运算滤波器

H=fspecial('log',n,sigma)

高斯拉普拉斯 (LoG) 运算滤波器

H=fspecial('average',n)

均值滤波器

H=fspecial('unsharp',alpha)

模糊对比增强滤波器

说明：对于形式 H=fspecial(type)，fspecial 函数产生一个由 type 指定的二维滤波器 H，返回的 H 常与其它滤波器搭配使用。

4. 彩色增强的 Matlab 实现

4.1 imfilter 函数

功能：真彩色增强

格式：B=imfilter(A,h)

说明：将原始图像 A 按指定的滤波器 h 进行滤波增强处理，增强后的图像 B 与 A 的尺寸和类型相同

图像的变换

1. 离散傅立叶变换的 Matlab 实现

Matlab 函数 fft、fft2 和 fftn 分别可以实现一维、二维和 N 维 DFT 算法；而函数 ifft、ifft2 和 ifftn 则用来计算反 DFT。

这些函数的调用格式如下：

A = fft(X,N,DIM)

其中，X 表示输入图像；N 表示采样间隔点，如果 X 小于该数值，那么 Matlab 将会对 X 进行零填充，否则将进行截取，使之长度为

N；DIM 表示要进行离散傅立叶变换。

A = fft2(X,MROWS,NCOLS)

其中，MROWS 和 NCOLS 指定对 X 进行零填充后的 X 大小。

A = fftn(X,SIZE)

其中，SIZE 是一个向量，它们每一个元素都将指定 X 相应维进行零填充后的长度。

函数 ifft、ifft2 和 ifftn 的调用格式于对应的离散傅立叶变换函数一致。

例子：图像的二维傅立叶频谱

% 读入原始图像

I = imread('lena.bmp');

imshow(I)

```
% 求离散傅立叶频谱
J=fftshift(fft2(I));
figure;
imshow(log(abs(J)),[8,10])
```

2. 离散余弦变换的 Matlab 实现

2.1. dct2 函数

功能：二维 DCT 变换

格式：B=dct2(A)

B=dct2(A,m,n)

B=dct2(A,[m,n])

说明：B = dct2(A) 计算 A 的 DCT 变换 B，A 与 B 的大小相同；B = dct2(A,m,n) 和 B=dct2(A,[m,n]) 通过对 A 补 0 或剪裁，使 B 的大

小为 $m \times n$ 。

2.2. idct2 函数

功能：DCT 反变换

格式：B=idct2(A)

B=idct2(A,m,n)

B=idct2(A,[m,n])

说明：B = idct2(A) 计算 A 的 DCT 反变换 B，A 与 B 的大小相同；B = idct2(A,m,n) 和 B=idct2(A,[m,n]) 通过对 A 补 0 或剪裁，使 B

的大小为 $m \times n$ 。

2.3. dctmtx 函数

功能：计算 DCT 变换矩阵

格式：D = dctmtx(n)

说明：D = dctmtx(n) 返回一个 $n \times n$ 的 DCT 变换矩阵，输出矩阵 D 为 double 类型。

3. 图像小波变换的 Matlab 实现

3.1 一维小波变换的 Matlab 实现

(1) dwt 函数

功能：一维离散小波变换

格式：[cA,cD]=dwt(X,'wname')

[cA,cD]=dwt(X,Lo_D,Hi_D)

说明：[cA,cD]=dwt(X,'wname') 使用指定的小波基函数 'wname' 对信号 X 进行分解，cA、cD

分别为近似分量和细节分量；[cA,cD]=dwt(X,Lo_D,Hi_D) 使用指定的滤波器组 Lo_D、Hi_D 对信号进行分解。

(2) idwt 函数

功能：一维离散小波反变换

格式：X=idwt(cA,cD,'wname')

 X=idwt(cA,cD,Lo_R,Hi_R)

 X=idwt(cA,cD,'wname',L)

 X=idwt(cA,cD,Lo_R,Hi_R,L)

说明：X=idwt(cA,cD,'wname') 由近似分量 cA 和细节分量 cD 经小波反变换重构原始信号 X。

 'wname' 为所选的小波函数

 X=idwt(cA,cD,Lo_R,Hi_R) 用指定的重构滤波器 Lo_R 和 Hi_R 经小波反变换重构原始信号 X。

 X=idwt(cA,cD,'wname',L) 和 X=idwt(cA,cD,Lo_R,Hi_R,L) 指定返回信号 X 中心附近的 L 个点。

3.2 二维小波变换的 Matlab 实现

二维小波变换的函数

函数名	函数功能
dwt2	二维离散小波变换
wavedec2	二维信号的多层小波分解
idwt2	二维离散小波反变换
waverec2	二维信号的多层小波重构
wrcoef2	由多层小波分解重构某一层的分解信号
upcoef2	由多层小波分解重构近似分量或细节分量
detcoef2	提取二维信号小波分解的细节分量
appcoef2	提取二维信号小波分解的近似分量
upwlev2	二维小波分解的单层重构
dwtper2	二维周期小波变换
idwtper2	二维周期小波反变换

(1) wcodemat 函数

功能：对数据矩阵进行伪彩色编码

格式：Y=wcodemat(X,NB,OPT,ABSOL)

 Y=wcodemat(X,NB,OPT)

 Y=wcodemat(X,NB)

 Y=wcodemat(X)

说明：Y=wcodemat(X,NB,OPT,ABSOL) 返回数据矩阵 X 的编码矩阵 Y；NB 伪编码的最大值，即编码范围为 0~NB，缺省值 NB=16；

 OPT 指定了编码的方式（缺省值为 'mat'），即：

OPT = 'row' , 按行编码
OPT = 'col' , 按列编码
OPT = 'mat' , 按整个矩阵编码

ABSOL 是函数的控制参数 (缺省值为 '1'), 即:

ABSOL = 0 时, 返回编码矩阵
ABSOL = 1 时, 返回数据矩阵的绝对值 ABS(X)

(2) dwt2 函数

功能: 二维离散小波变换

格式: [cA,cH,cV,cD]=dwt2(X,'wname')

[cA,cH,cV,cD]=dwt2(X,Lo_D,Hi_D)

说明: [cA,cH,cV,cD]=dwt2(X,'wname')使用指定的小波基函数 'wname' 对二维信号 X 进行二维离散小波变换; cA, cH,cV,cD 分别为近似分

量、水平细节分量、垂直细节分量和对角细节分量; [cA,cH,cV,cD]=dwt2(X,Lo_D,Hi_D) 使用指定的分解低通和高通滤波器 Lo_D 和 Hi_D 分

解信号 X。

(3) wavedec2 函数

功能: 二维信号的多层小波分解

格式: [C,S]=wavedec2(X,N,'wname')

[C,S]=wavedec2(X,N,Lo_D,Hi_D)

说明: [C,S]=wavedec2(X,N,'wname') 使用小波基函数 'wname' 对二维信号 X 进行 N 层分解; [C,S]=wavedec2(X,N,Lo_D,Hi_D) 使用指定

的分解低通和高通滤波器 Lo_D 和 Hi_D 分解信号 X。

(4) idwt2 函数

功能: 二维离散小波反变换

格式: X=idwt2(cA,cH,cV,cD,'wname')

X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R)

X=idwt2(cA,cH,cV,cD,'wname',S)

X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S)

说明: X=idwt2(cA,cH,cV,cD,'wname') 由信号小波分解的近似信号 cA 和细节信号 cH、cH、cV、cD 经小波反变换重构原信号 X

; X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R) 使用指定的重构低通和高通滤波器 Lo_R 和 Hi_R 重构原信号 X; X=idwt2(cA,cH,cV,cD,'wname',S)

和 X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S) 返回中心附近的 S 个数据点。

(5) waverec2 函数

说明: 二维信号的多层小波重构

格式：X=waverec2(C,S,'wname')
X=waverec2(C,S,Lo_R,Hi_R)

说明：X=waverec2(C,S,'wname') 由多层二维小波分解的结果 C、S 重构原始信号 X，'wname'

为使用的小波基函数；X=waverec2(C,S,Lo_R,Hi_R) 使用重构低通和高通滤波器 Lo_R 和 Hi_R 重构原信号。

图像处理工具箱

1. 图像和图像数据

缺省情况下，MATLAB 将图像中的数据存储为双精度类型(double)，64 位浮点数，所需存储量很大；MATLAB 还支持另一种类型无符号整型(uint8)，即图像矩阵中每个数据占用 1 个字节。

在使用 MATLAB 工具箱时，一定要注意函数所要求的参数类型。另外，uint8 与 double 两种类型数据的值域不同，编程需注意值域转换。

从 uint8 到 double 的转换

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

从 double 到 uint8 的转换

图像类型	MATLAB 语句
索引色	B=uint8(round(A-1))
索引色或真彩色	B=uint8(round(A*255))
二值图像	B=logical(uint8(round(A)))

2. 图像处理工具箱所支持的图像类型

2.1 真彩色图像

R、G、B 三个分量表示一个像素的颜色。如果要读取图像中(100,50)处的像素值，可查看三元数据(100,50,1:3)。

真彩色图像可用双精度存储，亮度值范围是[0,1]；比较符合习惯的存储方法是用无符号整型存储，亮度值范围[0,255]

2.2 索引色图像

包含两个结构，一个是调色板，另一个是图像数据矩阵。调色板是一个有 3 列和若干行的色彩映象矩阵，矩阵每行代表一种颜色，3 列分别代表红、绿、蓝色强度的双精度数。

注意：MATLAB 中调色板色彩强度[0,1], 0 代表最暗, 1 代表最亮。
常用颜色的 RGB 值

颜色	R	G	B	颜色	R	G	B
黑	0	0	1	洋红	1	0	1
白	1	1	1	青蓝	0	1	1
红	1	0	0	天蓝	0.67	0	1
绿	0	1	0	橘黄	1	0.5	0
蓝	0	0	1	深红	0.5	0	0
黄	1	1	0	灰	0.5	0.5	0.5

产生标准调色板的函数

函数名	调色板
Hsv	色彩饱和度, 以红色开始, 并以红色结束
Hot	黑色 - 红色 - 黄色 - 白色
Cool	青蓝和洋红的色度
Pink	粉红的色度
Gray	线型灰度
Bone	带蓝色的灰度
Jet	Hsv 的一种变形, 以蓝色开始, 以蓝色结束
Copper	线型铜色度
Prim	三棱镜, 交替为红、橘黄、黄、绿和天蓝
Flag	交替为红、白、蓝和黑

缺省情况下, 调用上述函数灰产生一个 64×3 的调色板, 用户也可指定调色板大小。

索引色图像数据也有 double 和 uint8 两种类型。
当图像数据为 double 类型时, 值 1 代表调色板中的第 1 行, 值 2 代表第 2 行……
如果图像数据为 uint8 类型, 0 代表调色板的第一行,, 值 1 代表第 2 行……

2.3 灰度图像

存储灰度图像只需要一个数据矩阵。
数据类型可以是 double, [0, 1]; 也可以是 uint8, [0,255]

2.4 二值图像

二值图像只需一个数据矩阵, 每个像素只有两个灰度值, 可以采用 uint8 或 double 类型存储。
MATLAB 工具箱中以二值图像作为返回结果的函数都使用 uint8 类型。

2.5 图像序列

MATLAB 工具箱支持将多帧图像连接成图像序列。

图像序列是一个 4 维数组，图像帧的序号在图像的长、宽、颜色深度之后构成第 4 维。

分散的图像也可以合并成图像序列，前提是各图像尺寸必须相同，若是索引色图像，调色板也必须相同。

可参考 cat() 函数 `A = cat(4,A1,A2,A3,A4,A5)`

3. MATLAB 图像类型转换

图像类型转换函数

函数名	函数功能
dither	图像抖动，将灰度图变成二值图，或将真彩色图像抖动成索引色图像
gray2ind	将灰度图像转换成索引图像
grayslice	通过设定阈值将灰度图像转换成索引色图像
im2bw	通过设定亮度阈值将真彩色、索引色、灰度图转换成二值图
ind2gray	将索引色图像转换成灰度图像
ind2rgb	将索引色图像转换成真彩色图像
mat2gray	将一个数据矩阵转换成一副灰度图
rgb2gray	将一副真彩色图像转换成灰度图像
rgb2ind	将真彩色图像转换成索引色图像

4. 图像文件的读写和查询

4.1 图形图像文件的读取

利用函数 `imread()` 可完成图形图像文件的读取，语法：

```
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）
```

通常，读取的大多数图像均为 8bit，当这些图像加载到内存中时，Matlab 就将其存放在类 `uint8` 中。此为 Matlab 还支持 16bit 的 PNG 和 TIF 图像，当读取这类文件时，Matlab 就将其存贮在 `uint16` 中。

注意：对于索引图像，即使图像阵列的本身为类 `uint8` 或类 `uint16`，`imread` 函数仍将颜色映象表读取并存贮到一个双精度的浮点类型的阵列中。

4.2 图形图像文件的写入

使用 `imwrite` 函数，语法如下：

```
imwrite(A,filename,fmt)
imwrite(X,map,filename,fmt)
imwrite(...,filename)
imwrite(...,parameter,value)
```

当利用 `imwrite` 函数保存图像时，Matlab 缺省的方式是将其简化为 `uint8` 的数据格式。

4.3 图形图像文件信息的查询 `imfinfo()`函数

5. 图像文件的显示

5.1 索引图像及其显示

方法一：

```
image(X)
colormap(map)
```

方法二：

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

5.2 灰度图像及其显示

Matlab 7.0 中，要显示一副灰度图像，可以调用函数 `imshow` 或 `imagesc`（即 `imagescale`，图像缩放函数）

(1) `imshow` 函数显示灰度图像

使用 `imshow(I)` 或 使用明确指定的灰度级数目：`imshow(I,32)`

由于 Matlab 自动对灰度图像进行标度以适合调色板的范围，因而可以使用自定义大小的调色板。其调用格式如下：

```
imshow(I,[low,high])
```

其中，`low` 和 `high` 分别为数据数组的最小值和最大值。

(2) `imagesc` 函数显示灰度图像

下面的代码是具有两个输入参数的 `imagesc` 函数显示一副灰度图像

```
imagesc(1,[0,1]);
colormap(gray);
```

`imagesc` 函数中的第二个参数确定灰度范围。灰度范围中的第一个值（通常是 0），对应于颜色映象表中的第一个值（颜色），第二个值（通常是 1）则对应与颜色映象表中的最后一个值（颜色）。灰度范围中间的值则线性对应与颜色映象表中剩余的值（颜色）。

在调用 `imagesc` 函数时，若只使用一个参数，可以用任意灰度范围显示图像。在该调用方式下，数据矩阵中的最小值对应于颜色映象表中的第一个颜色值，数据矩阵中的最大值对应于颜色映象表中的最后一个颜色值。

5.3 RGB 图像及其显示

(1) image(RGB)

不管 RGB 图像的类型是 double 浮点型，还是 uint8 或 uint16 无符号整数型，Matlab 都能通过 image 函数将其正确显示出来。

```
RGB8 = uint8(round(RGB64×255)); % 将 double 浮点型转换为 uint8 无符号整型
RGB64 = double(RGB8)/255;      % 将 uint8 无符号整型转换为 double 浮点型
RGB16 = uint16(round(RGB64×65535)); % 将 double 浮点型转换为 uint16 无符号整型
RGB64 = double(RGB16)/65535;   % 将 uint16 无符号整型转换为 double 浮点型
```

(2) imshow(RGB) 参数是一个 $m \times n \times 3$ 的数组

5.4 二进制图像及其显示

(1) imshow(BW)

在 Matlab 7.0 中，二进制图像是一个逻辑类，仅包括 0 和 1 两个数值。像素 0 显示为黑色，像素 1 显示为白色。

显示时，也可通过 NOT(~)命令，对二进制图像进行取反，使数值 0 显示为白色；1 显示为黑色。

例如： imshow(~BW)

(2) 此外，还可以使用一个调色板显示一副二进制图像。如果图形是 uint8 数据类型，则数值 0 显示为调色板的第一个颜色，数值 1 显示为第二个颜色。

例如： imshow(BW,[1 0 0;0 0 1])

5.5 直接从磁盘显示图像

可使用一下命令直接进行图像文件的显示：

```
imshow filename
```

其中，filename 为要显示的图像文件的文件名。

如果图像是多帧的，那么 imshow 将仅显示第一帧。但需注意，在使用这种方式时，图像数据没有保存在 Matlab 7.0 工作平台。如果希望将图像装入工作台中，需使用 getimage 函数，从当前的句柄图形图像对象中获取图像数据，

命令形式为： rgb = getimage;

bwlabel

功能：

标注二进制图像中已连接的部分。

```
L = bwlabel(BW,n)
```

`[L,num] = bwlabel(BW,n)`

`isbw`

功能：

判断是否为二进制图像。

语法：

`flag = isbw(A)`

相关命令：

`isind, isgray, isrgb`

74 . `isgray`

功能：

判断是否为灰度图像。

语法：

`flag = isgray(A)`

相关命令：

`isbw, isind, isrgb`

11 . `bwselect`

功能：

在二进制图像中选择对象。

语法：

`BW2 = bwselect(BW1,c,r,n)`

`BW2 = bwselect(BW1,n)`

`[BW2,idx] = bwselect(...)`

举例

`BW1 = imread('text.tif');`

`c = [16 90 144];`

`r = [85 197 247];`

`BW2 = bwselect(BW1,c,r,4);`

`imshow(BW1)`

`figure, imshow(BW2)`

47 . `im2bw`

功能：

转换图像为二进制图像。

语法：

`BW = im2bw(I,level)`

`BW = im2bw(X,map,level)`

`BW = im2bw(RGB,level)`

举例

`load trees`

`BW = im2bw(X,map,0.4);`

`imshow(X,map)`

Matlab 中如何读出写入图像文件以及对图像的简单处理

MATLAB 图像处理工具箱支持四种基本图像类型:索引图像、灰度图像、二进制图像和 RGB 图像。MATLAB 直接从图像文件中读取的图像为 RGB 图像。它存储在三维数组中。这个三维数组有三个面,依次对应于红(Red)、绿(Green)、蓝(Blue)三种颜色,而面中的数据则分别是这三种颜色的强度值,面中的元素对应于图像中的像素点。设所得矩阵为 X 三维矩阵 (256,256,3), X(:,:,1)代表红颜色的 2 维矩阵 X(:,:,2)代表绿颜色的 2 维矩阵, X(:,:,3)代表兰颜色的 2 维矩阵。[X, map]=imread('34.bmp');r=double(X(:,:,1)); %r 是 256 x 256 的红色信息矩阵 g=double(X(:,:,2)); %g 是 256 x 256 的绿色信息矩阵 b=double(X(:,:,3)); %b 是 256 x 256 的兰色信息矩阵

索引图像数据包括图像矩阵 X 与颜色图数组 map,其中颜色图 map 是按图像中颜色值进行排序后的数组。对于每个像素,图像矩阵 X 包含一个值,这个值就是颜色图数组 map 中的索引。颜色图 map 为 m×3 双精度矩阵,各行分别指定红、绿、蓝(R、G、B)单色值,map=[RGB],R、G、B 为值域为[0,1]的实数值,m 为索引图像包含的像素个数。

对于相同的数据,采用 uint8 格式比双精度格式节省内存空间,从而更经济。在 MATLAB 中如果索引图像的颜色图小于 256 行,则它的图像矩阵以 uint8 格式存储,否则以双精度格式存储。

一: imread:从图像文件夹中读取图像。

A = imread(FILENAME,FMT) 读取图像到 A, 如果文件是包含一灰度图像, A 是一二维矩阵, 如果文件是包含一真彩色图像 (RGB), A 是一三维矩阵 (M-by-N-by-3)。FILENAME : 图像文件名; FMT: 图像文件格式;

文件必须在当前目录下, 或在 Matlab 的一路径上。如果 imread 不能够找到一名称为 FILENAME 的文件,那么它将找一名称为 FILENAME.FMT 的文件

[X,MAP] = imread(FILENAME,FMT) 把图像 FILENAME 读入与它相关的图像色彩信息写入 MAP, 图像色彩信息值在范围[0,1]中自动地重新调整。

[...] = imread(FILENAME)这种方式是试图得到文件的格式从文件所包含的信息。

[...] = imread(URL,...)从一 Internet URL 上读图像 URL 必须包含协议(即: "http://")。

1.2 数据类型:

TIFF 的特殊语法:

[...] = imread(...,IDX) 从很多图像 TIFF 文件中 读一个图像; IDX 是一个整数值, 它显示了所读图像在文件中的顺序, 例如: 如果 IDX 是 3, imread 将读文件中的第三个图像。 如果省略了这个变量, imread 将读文件中的第一个图像。

IMREAD 支持的图像文件格式: JPEG TIFF GIF BMP PNG HDF PCX XWD ICO CUR RAS PBM PGM PPM

相关信息也可在 Matlab 中查看: imfinfo, imwrite, imformats, fread,

二: imwrite 输出图像

imwrite(A,FILENAME,FMT) 把图像 A 写入图像文件 FILENAME.

imwrite(X,MAP,FILENAME,FMT) 把 X 和它的相关色彩信息 MAP 写入 FILENAME.

imwrite(...,FILENAME) 把图像 写入图像文件 FILENAME,并推测可能的格式用来做 filename 的扩展名。扩展名必须是 FMT 中一合法名。

imwrite(...,PARAM1,VAL1,PARAM2,VAL2,...) 不同的参数控制输出文件的各种不同特征。参数要是当前所支持的 HDF, JPEG, TIFF, PNG, PBM, PGM, 和 PPM 文件

三: image 显示图像.image(C) 把矩阵 C 转成一图像. C 可以是一 M×N 或 M×N×3 维的矩

阵,且可以是包含 double, uint8,或 uint16 数据.image 是用来显示附标图像,即显示的图像上有 x,y 坐标轴的显示,可以看到图像的像素大小。但可以加上 axis off 命令即可把坐标去掉。

imshow 只是显示图像。用 colormap 来定义图像显示用的颜色查找表,比如用 colormap(pink),可以把黑白图像显示成带粉红色的图像。

图像像素矩阵的数据类型:(1)显示真彩色图像像素三维矩阵 X,如果是 uint8 类型,要求矩阵的数据范围为 0-255,(2)如果是 double 型,则其数据范围为 0-1,要不就会出错或者出现空白页。

类型转换:(1)如果你原来的数值是 uint8,在运算中转换为 double 后,实际要显示的数值没有改变的话,只要用 uint8(X)就可转换为 uint8 型,如果不想转换频繁,也可在显示时用 X/255 来转换为符合 0-1double 类型范围要求的数值显示。(2)如果显示索引图像(二维矩阵),如果索引图像像素数值是 double 型,则它的取值范围为 1-length(colormap),数值起点为 1,则矩阵中数值为 1 的对应 colormap 中第一行数据,如果索引图像像素数值是 uint8,则取值范围为 0-255,数值起点为 0,则矩阵中数值为 0 的对应 colormap 中第一行数据,所以索引图像这两个数据类型之间的转换,要考虑到+1 或-1。直接用 uint8 或 double 转换则会查找移位,产生失真情况。uint16 数据类型与 uint8 类似,取值范围为 0-65536。

四:其它常用图像操作:

图像显示于屏幕有 imshow(), image()函数;

图像进行裁剪 imcrop();

图像的插值缩放 imresize()函数实现;

旋转用 imrotate()实现。

五:具体的操作

下面通过运用图像处理工具箱中的有关函数对下图(nice.bmp)进行一些变换。见后面的 transfer.m 内容!

变换前图片: (nice.bmp)

变换后所得图片: newpic.bmp

例,在电脑 F:\picture 下有一彩色图像文件 nice.bmp,则可由下述语句读取:

下面是对图像 nice.bmp 以 y 轴为对称轴所做的一个对称变换。

```
% Transfer1.m
```

```
clear all
```

```
figure
```

```
[x,map]=imread('F:\picture\nice.bmp');% 所得 x 为一 375x420x3 的矩阵
```

```
[w1,w2,w3]=size(x); % 375 X 420
```

```
w22=floor(w2/2);
```

```
image(x); %显示出图像
```

```
title('HELLO! @This is the first pose of me')%则显示出图像 nice.bmp
```

```
axis off; % 去掉图像中的坐标
```

```
colormap(map); % colormap(),图像查找表函数。函数结构为 colormap(map),设置当前的图像查找表到 map。
```

```
imwrite(x,map,'nice.bmp')
```

```

for i=1:w1
for j=1:w22          % 图像关于 y 轴对折
t=x(i,j);
x(i,j)=x(i,w2-j+1);
x(i,w2-j+1)=t;
end
end
figure
image(x);
axis off
title('HELLO!!@@ Can you find any difference of my two picture! ') colormap(map);
imwrite(x,map,'newpic.bmp') %把 x 写到 newpic2.bmpz 中去
% Transfer1.m 文件中包含了最基本也是最常用的对读像处理的命令。
在对图像处理的整个过程中，实质上是对[x,map]=imread('figure')函数中所得 x 矩阵的各种
变换！

```