

Matlab 数字图像处理函数

1、数字图像的变换

① fft2: fft2 函数用于数字图像的二维傅立叶变换，如：`i=imread('104_8.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('104_8.tif');
```

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

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

2、模拟噪声生成函数和预定义滤波器

① imnoise: 用于对数字图像生成模拟噪声，如：

```
i=imread('104_8.tif');
```

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

② fspecial: 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
```

3、数字图像的增强

①直方图: imhist 函数用于数字图像的直方图显示，如：

```
i=imread('104_8.tif');
```

```
imhist(i);
```

②直方图均化: `histeq` 函数用于数字数字图像的直方图均化, 如:

```
i=imread('104_8.tif');
```

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

③对比度调整: `imadjust` 函数用于数字数字图像的对比度调整, 如:

```
i=imread('104_8.tif');
```

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

④对数变换: `log` 函数用于数字数字图像的对数变换, 如:

```
i=imread('104_8.tif');
```

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

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

⑤基于卷积的数字数字图像滤波函数: `filter2` 函数用于数字数字图像滤波, 如:

```
i=imread('104_8.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('104_8.tif');
```

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

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

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

⑦中值滤波: `medfilt2` 函数用于数字数字图像的中值滤波, 如:

```
i=imread('104_8.tif');
```

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

⑧锐化

(1) 利用 Sobel 算子锐化数字数字图像, 如:

```
i=imread('104_8.tif');
```

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

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

(2) 利用拉氏算子锐化数字数字图像, 如:

```
i=imread('104_8.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;
```

4、数字图像边缘检测

①sobel 算子 如:

```
i=imread('104_8.tif');
```

```
j =edge(i,'sobel',thresh)
```

②**prewitt** 算子 如:

```
i=imread('104_8.tif');
```

```
j =edge(i,'prewitt',thresh)
```

③roberts 算子 如:

```
i=imread('104_8.tif');
```

```
j =edge(i,'roberts',thresh)
```

④log 算子 如:

```
i=imread('104_8.tif');
```

```
j =edge(i,'log',thresh)
```

⑤canny 算子 如:

```
i=imread('104_8.tif');
```

```
j =edge(i,'canny',thresh)
```

⑥Zero-Cross 算子 如:

```
i=imread('104_8.tif');
```

```
j =edge(i,'zerocross',thresh)
```

5、形态学数字图像处理

①膨胀: 是在二值化数字图像中“加长”或“变粗”的操作, 函数 `imdilate` 执行膨胀运算, 如:

```
a=imread('104_7.tif');%输入二值数字数字图像
```

```
b=[0 1 0;1 1 1;0 1 0];
```

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

②腐蚀: 函数 `imerode` 执行腐蚀, 如:

```
a=imread('104_7.tif');%输入二值数字数字图像
```

```
b=strel('disk',1);
```

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

③开运算：先腐蚀后膨胀称为开运算，用 `imopen` 来实现，如：

```
a=imread('104_8.tif');  
  
b=strel('square',2);  
  
c=imopen(a,b);
```

④闭运算：先膨胀后腐蚀称为闭运算，用 `imclose` 来实现，如：

```
a=imread('104_8.tif');  
  
b=strel('square',2);  
  
c=imclose(a,b);
```

数字图像增强

1. 直方图均衡化的 Matlab 实现

1.1 imhist 函数

功能：计算和显示数字数字图像的色彩直方图

格式：imhist(I,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) 就计算和显示索引色数字数字图像 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：显示数字数字图像的等灰度值图

格式：imcontour(I,n),imcontour(I,v)

说明：n 为灰度级的个数，v 是有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],[bottomtop],gamma)

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

说明: `J=imadjust(I,[low high],[bottomtop],gamma)` 其中, `gamma` 为校正量 r , `[lowhigh]` 为原数字数字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,[lowhigh],[bottom top],gamma)` 调整索引色数字数字图像的调色板 `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”, 即将原是图像 `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,...)` 返回从能将数字数字图像 `I` 的灰度直方图变换成数字数字图像 `J` 的直方图的变换 `T`; `newmap=histeq(X,map)` 和 `[new,T]=histeq(X,...)` 是针对索引色数字数字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)` 返回对数字数字图像 `I` 添加典型噪声后的有噪数字数字图像 `J`, 参数 `type` 和 `parameter` 用于确定噪声的类型和相应的参数。

3. 数字图像滤波的 Matlab 实现

3.1 conv2 函数

功能: 计算二维卷积

格式: `C=conv2(A,B)`

$C = \text{conv2}(\text{Hcol}, \text{Hrow}, A)$

$C = \text{conv2}(\dots, 'shape')$

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

$C = \text{conv2}(\text{Hcol}, \text{Hrow}, A)$ 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； $C = \text{conv2}(\dots, 'shape')$ 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 $\text{size}(A) > \text{size}(B)$ 时， $\text{size}(C) = [Ma - Mb + 1, Na - Nb + 1]$ 。

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： $Y = \text{filter2}(B, X)$

$Y = \text{filter2}(B, X, 'shape')$

说明：对于 $Y = \text{filter2}(B, X)$ ， 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 $Y = \text{filter2}(B, X, 'shape')$ ， 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， $\text{size}(Y) > \text{size}(X)$ ；

》 $same$ 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 $\text{size}(Y) < \text{size}(X)$ 。

3.4 fspecial 函数

功能：产生预定义滤波器

格式： $H = \text{fspecial}(\text{type})$

$H = \text{fspecial}('gaussian', n, \sigma)$ 高斯低通滤波器

$H = \text{fspecial}('sobel')$ Sobel 水平边缘增强滤波器

<code>H=fspecial('prewitt')</code>	Prewitt 水平边缘增强滤波器
<code>H=fspecial('laplacian',alpha)</code>	近似二维拉普拉斯运算滤波器
<code>H=fspecial('log',n,sigma)</code>	高斯拉普拉斯 (LoG) 运算滤波器
<code>H=fspecial('average',n)</code>	均值滤波器
<code>H=fspecial('unsharp',alpha)</code>	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：将原始数字数字图像 `A` 按指定的滤波器 `h` 进行滤波增强处理，增强后的数字数字图像 `B` 与 `A` 的尺寸和类型相同

数字图像的变换

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` 表示输入数字数字图像；`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：数字数字图像的二维傅立叶频谱

% 读入原始数字数字图像


```
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×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×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×n 的 DCT 变换矩阵，输出矩阵 D 为 double 类型。

3. 数字数字图像小波变换的 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 $0\sim NB$ ，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幻； 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。

数字图像处理工具箱

1. 数字图像和数字图像数据

缺省情况下，MATLAB 将数字数字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)，即数字数字图像矩

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

数字数字图像类型 MATLAB 语句

索引色 B=double(A)+1
索引色或真彩色 B=double(A)/255
二值数字数字图像 B=double(A)

从 double 到 uint8 的转换

数字数字图像类型 MATLAB 语句

索引色 B=uint8(round(A-1))
索引色或真彩色 B=uint8(round(A*255))
二值数字数字图像 B=logical(uint8(round(A)))

2. 数字数字图像处理工具箱所支持的数字数字图像类型

2.1 真彩色数字数字图像

R、G、B 三个分量表示一个像素的颜色。如果要读取数字数字图像中(100,50)处的像素值，

可查看三元数据(100,50,1:3)。

真彩色数字数字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 索引色数字数字图像

包含两个结构，一个是调色板，另一个是数字数字图像数据矩阵。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。

索引色数字数字图像数据也有 double 和 uint8 两种类型。

当数字数字图像数据为 double 类型时，值 1 代表调色板中的第 1 行，值 2 代表第 2 行……

如果数字数字图像数据为 uint8 类型，0 代表调色板的第一行，，值 1 代表第 2 行……

2.3 灰度数字数字图像

存储灰度数字数字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 二值数字数字图像

二值数字数字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 工具箱中以二值数字数字图像作为返回结果的函数都使用 `uint8` 类型。

2.5 数字数字图像序列

MATLAB 工具箱支持将多帧数字数字图像连接成数字数字图像序列。

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

分散的数字数字图像也可以合并成数字数字图像序列，前提是各数字数字图像尺寸必须相同，若是索引色数字数字图像，调色板也必须相同。

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

3. MATLAB 数字数字图像类型转换

数字数字图像类型转换函数

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

4. 数字数字图像文件的读写和查询

4.1 图形数字数字图像文件的读取

利用函数 `imread()` 可完成图形数字数字图像文件的读取，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，读取的大多数数字图像均为 8bit，当这些数字图像加载到内存中时，Matlab 就将其存放在类 uint8 中。此外，Matlab 还支持 16bit 的 PNG 和 TIF 数字图像，当读取这类文件时，Matlab 就将其存储在 uint16 中。

注意：对于索引数字图像，即使数字图像阵列的本身为类 uint8 或类 uint16，imread 函数仍将颜色映射表读取并存储到一个双精度的浮点类型的阵列中。

4.2 图形数字图像文件的写入

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 函数保存数字图像时，Matlab 缺省的方式是将其简化为 uint8 的数据格式。

4.3 图形数字图像文件信息的查询 imfinfo()函数

5. 数字图像文件的显示

5.1 索引数字图像及其显示

方法一：

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

方法二：

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

5.2 灰度数字图像及其显示

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

(1) `imshow` 函数显示灰度数字图像

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

由于 Matlab 自动对灰度数字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` 函数显示灰度数字图像

下面的代码是具有两个输入参数的 `imagesc` 函数显示一副灰度数字图像

```
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数字图像。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 数字图像及其显示

(1) `image(RGB)`

不管 RGB 数字图像的类型是 `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 二进制数字图像及其显示

(1) `imshow(BW)`

在 Matlab 7.0 中, 二进制数字图像是一个逻辑类, 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数字图像。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 直接从磁盘显示数字图像

可使用一下命令直接进行数字图像文件的显示:

```
imshow filename
```

其中, `filename` 为要显示的数字图像文件的文件名。

如果数字图像是多帧的, 那么 `imshow` 将仅显示第一帧。但需注意, 在使用这种方式时, 数字图像

数据没有保存在 Matlab7.0 工作平台。如果希望将数字图像装入工作台中, 需使用 `getimage` 函数, 从当前的句柄图形数字图像对象中获取数字图像数据,

命令形式为: `rgb = getimage;`

`bwlabel`

功能:

标注二进制数字图像中已连接的部分。

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

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

`isbw`

功能:

判断是否为二进制数字图像。

语法:

`flag = isbw(A)`

相关命令:

`isind, isgray, isrgb`

74. `isgray`

功能:

判断是否为灰度数字图像。

语法:

`flag = isgray(A)`

相关命令:

`isbw, isind, isrgb`

11. `bwselect`

功能:

在二进制数字图像中选择对象。

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:

转换数字图像为二进制数字图像。

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)
```