



MATLAB与数字信号处理

数字图像处理



5 基于MATLAB的图像处理

- 5.1 图像文件的读写与显示
- 5.2 图像运算
- 5.3 图像变换
- 5.4 图像增强
- 5.5 图像分割
- 5.6 图像分析

5.1 图像文件的读写及显示

支持4种图像类型:

二值、灰度、索引、RGB彩色。

1) 图形图像文件的读取

A=imread(filename)

A=imread(filename,fmp)

[X,map]=imread(filename,fmp)

2) 图形图像文件的写入

imwrite(A,filename)

3) 查询文件信息

info=imfinfo(filename)

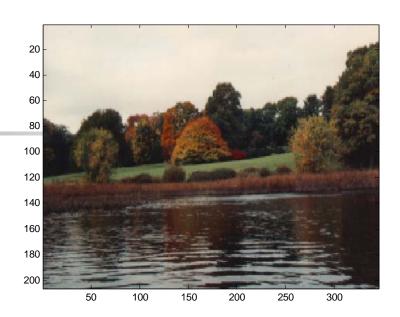


4)图像显示

image(I)
imagesc(I)
colormap(MAP)

imshow(I)
imshow(X,map)
imshow(I,[low,high])

colorbar 添加颜色条

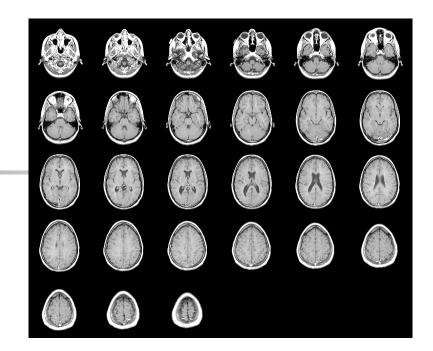


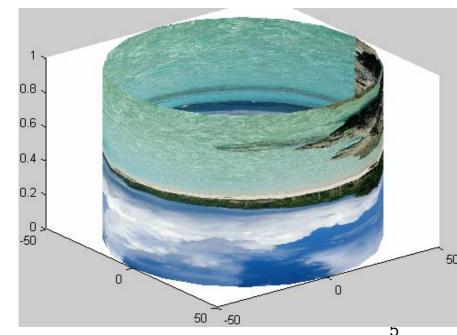
X=imread('autumn.tif');
image(X); imshow(X);





- 多帧图像显示
 load mri
 montage(D,map);
 mov=immovie(D,map);
 movie(mov);
- 纹理映射
 [x,y,z] = cylinder(50);I = imread('天堂.JPG');warp(x,y,z,I);





6) 图像类型转换

BW=im2bw(I,level)

BW=im2bw(X,map,level)

BW=im2bw(RGB,level)

I=mat2gray(A,[max,min])

[x,map]=gray2ind(I,n) 反之 ind2gray、ind2rgb

rgb2gray、rgb2ind、dither(抖动)

7) 颜色空间转换

RGB、HSV、YUV(ycbcr)、NTSC、HIS

函数名是对应的小写 rgb2hsv

5.2 图像运算

代数运算、几何运算、邻域操作和块操作。
1)代数运算:

指对两幅输入图象点对点的代数运算。

绝对差值函数 z=imabsdiff(x,y)

```
I=imread( 'cameraman.tif' );
J=uint8(filter2(fspecial('gaussian'),I));
K=imabsdiff(I,J);
subplot(1,2,1);imshow(I);
subplot(1,2,2);imshow(K,[]);
```



显示结果:





图象的叠加函数 z=imadd(x,y)

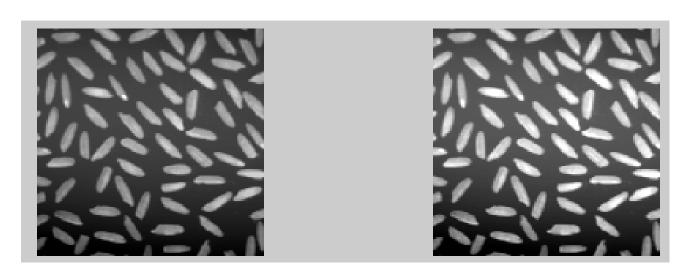
```
I=imread('cameraman.tif' );
J=imread('rice.tif');
K=imadd(I,J);
imshow(K,[]);
```





增加亮度

```
I=imread('rice.tif');
J=imadd(I,50);
subplot(2,2,1);imshow(I,[]);
subplot(2,2,2);imshow(J,[]);
```



4

图象的求补函数 z=imcomplement(x)

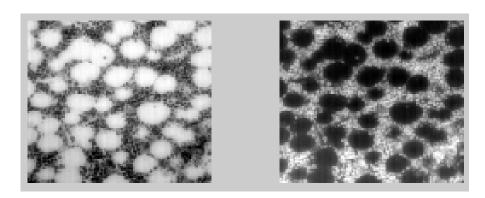
```
I = imread('text.tif');
J = imcomplement(I);
subplot(1,2,1);imshow(I,[]);
subplot(1,2,2);imshow(J,[])
```





灰度图象补运算

```
I = imread('bonemarr.tif');
J = imcomplement(I);
subplot(1,2,1);imshow(I,[]);
subplot(1,2,2);imshow(J,[])
```





其它计算:

- imdivide 探测变化、比例图像
- immultiply 掩模处理
- imsubtract 变化检测
- imlincomb



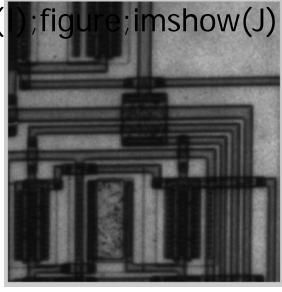
2) 几何操作

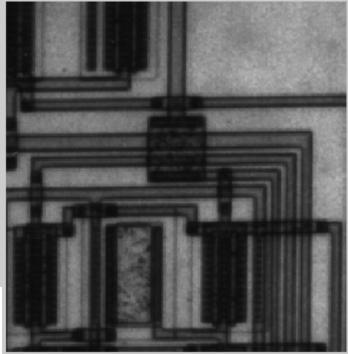
改变图象大小: y=imresize(x,m,method)

I = imread('circuit.tif');

J = imresize(I, 1.25);

imshow(







图象旋转: y=imrotate(x,angle,method)

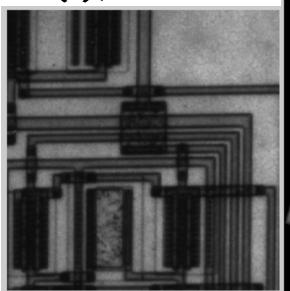
I = imread('circuit.tif');

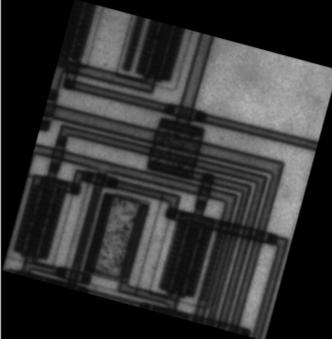
J = imrotate(I,-15,'bilinear');

imshow(I);

figure; imshow(J);

图像剪裁 Imcrop(I,rect);







3) 邻域和块操作

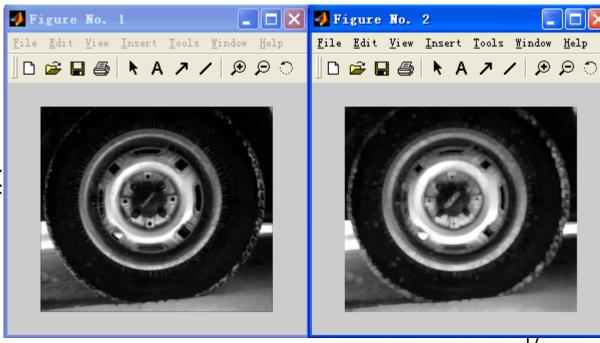
实现一个滑动邻域操作步骤:

- (1) 选择一个单独的像素。
- (2) 确定该像素的滑动邻域。
- (3) 对邻域中的像素值应用一个函数求值。
- (4)将结果作为对应像素的值。
- (5) 对输入图象的每一个像素都重复上面四个步骤。

滑动邻域操作函数:

y=nlfilter(x,[m n],fun, p1,p2,...)

inline('CE') inline('CE',arg1,arç



块操作函数:

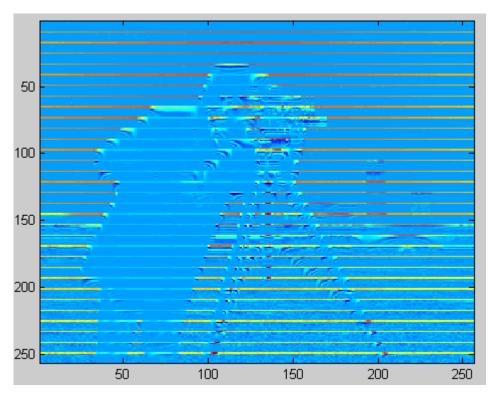
y=blkproc(x,[m n],fun, p1,p2,...)

I = imread('cameraman.tif');

f=@dct;

J=blkproc(I,[8 8],f);

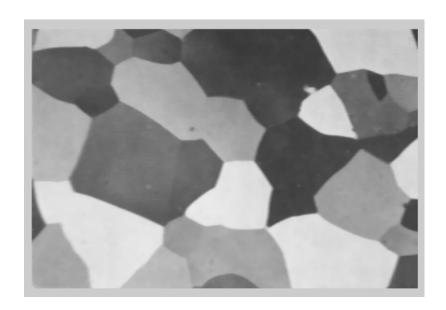
imagesc(J);

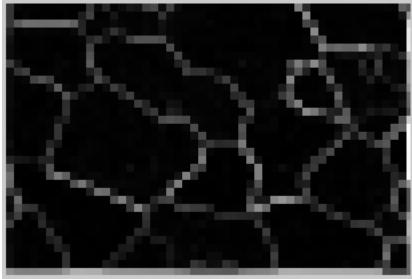




```
I = imread('alumgrns.tif')
fun = inline('std2(x)*ones(size(x))');
I2 = blkproc(I,[8 8],fun);
imshow(I)
figure, imshow(I2,[])
```







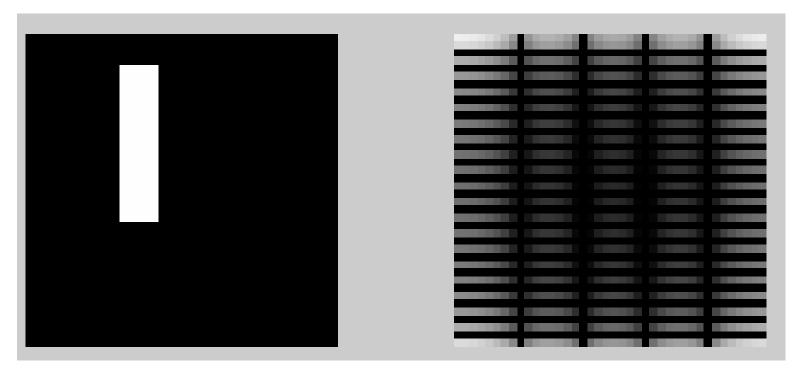
4

5.3 图象变换

1) FFT

 例:绘制一个二值图象矩阵,将其傅立叶变换可视化 f=zeros(40,40); f(5:24,13:17)=1; subplot(1,2,1);imshow(f); F=fft2(f); F2=log(abs(F)); subplot(1,2,2); imshow(F2,[-1 5]);

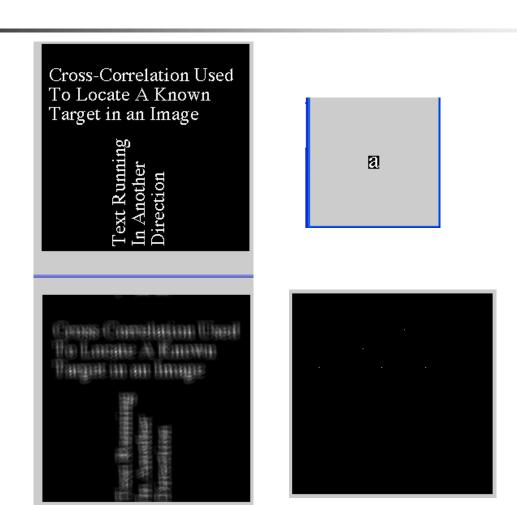






• 图像定位的应用

```
w=imread('text.tif');
a=w(59:71,81:91);
imshow(w);figure;imshow(a);
C=ifft2(fft2(w).*fft2(rot90(a,2),256,256));
figure;imshow(C,[]);
thresh=45;
figure;imshow(C>thresh);
```

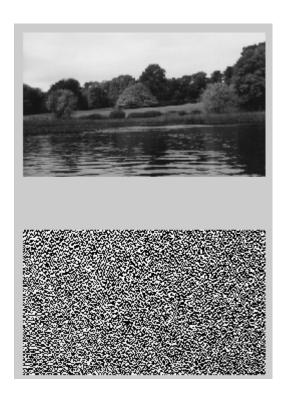




两种方法: dct2 dctmtx

• 例:对图象进行DCT变换和反变换

RGB=imread('autumn.tif'); I=rgb2gray(RGB); subplot(1,2,1);imshow(I); J=dct2(I); subplot(1,2,2);imshow(J);





将上例中DCT变换后的矩阵中小于100的设为0,

重构图象。

J(abs(J)<100)=0; K=idct2(J); figure; imshow(K,[0 255]);

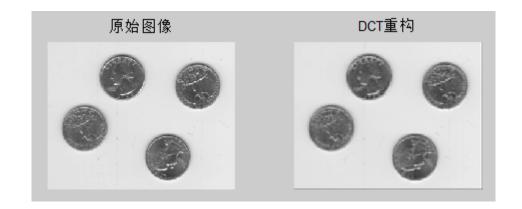


```
I=imread('eight.tif');
subplot(121),
  imshow(I);
  title('原始图像');
  I=im2double(I);
  T=dctmtx(8);
  B=blkproc(I,[8 8], 'P1*x*P2',T,T');
  Mask=[ 1 1 1 1 0 0 1 1
        11100001
        11000000
        1000000
        0000000
        0000000
        11000011
```

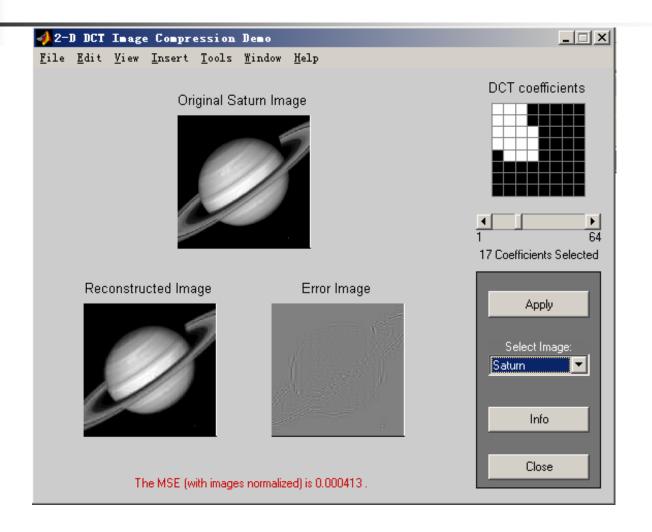
11000011];



- B2=blkproc(B,[8 8],'P1.*x',Mask); %此处为点乘
- I2=blkproc(B2,[8 8], 'P1*x*P2',T',T);
- subplot(122),
- imshow(I2); %重建后的图像
- title('DCT重构');



Dctdemo 重构图象和原始图象比较





Radon 变换

[R xp]=radon(I,theta)

• 例: 计算正方形图象在0和45方向上的3

I = zeros(100, 100);

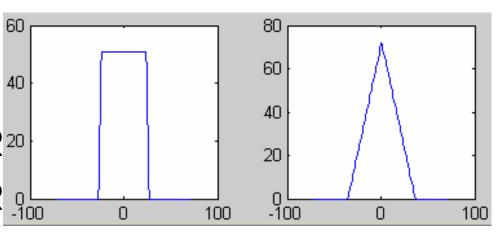
I(25:75,25:75)=1;

imshow(I);

[R,xp]=radon(I,[0 45]);

subplot(1,2,1); plot (xp,R^{20})

subplot(1,2,2); plot(xp,R $\frac{1}{2}$



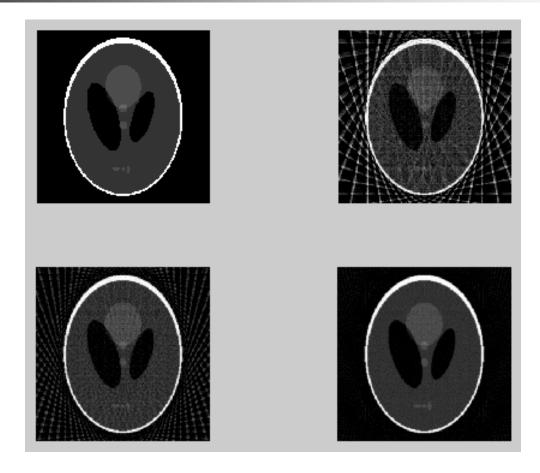
例:构造一个简单图象的投影并重建图象

```
P=phantom(256);
subplot(2,2,1);
imshow(P);
theta1=0:10:170;
theta2=0:5:175;
theta3=0:2:178;
[R1,xp]=radon(P,theta1);
[R2,xp]=radon(P,theta2);
[R3,xp]=radon(P,theta3);
```



```
I1=iradon(R1, theta1);
I2=iradon(R2, theta2);
I3=iradon(R3, theta3);
subplot(2,2,2);imshow(I1);
subplot(2,2,3);imshow(I2);
subplot(2,2,4);imshow(I3);
```







4) 小波变换

[CA,CD] = dwt(X,'wname')
[CA,CH,CV,CD] = dwt2(X,'wname')

例:

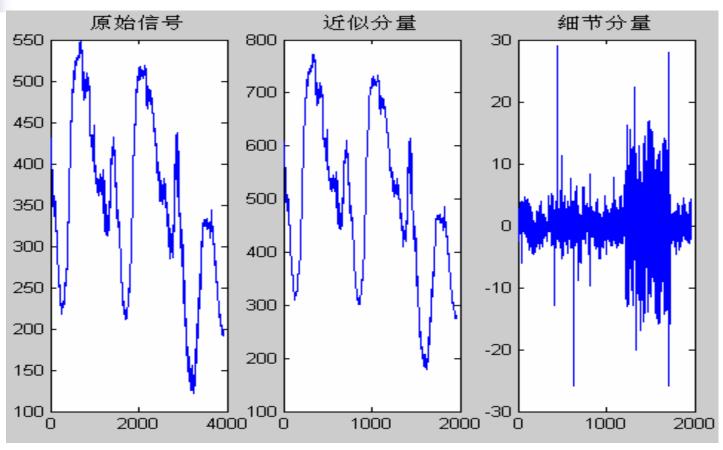
对电源信号进行小波分解和重构



load leleccum

```
s=leleccum(1:3920);
ls=length(s);
subplot(1,3,1);plot(s);title('原始信号');
[ca,cd]=dwt(s,'db1');
subplot(1,3,2);plot(ca);title('近似分量');
subplot(1,3,3);plot(cd);title('细节分量');
```





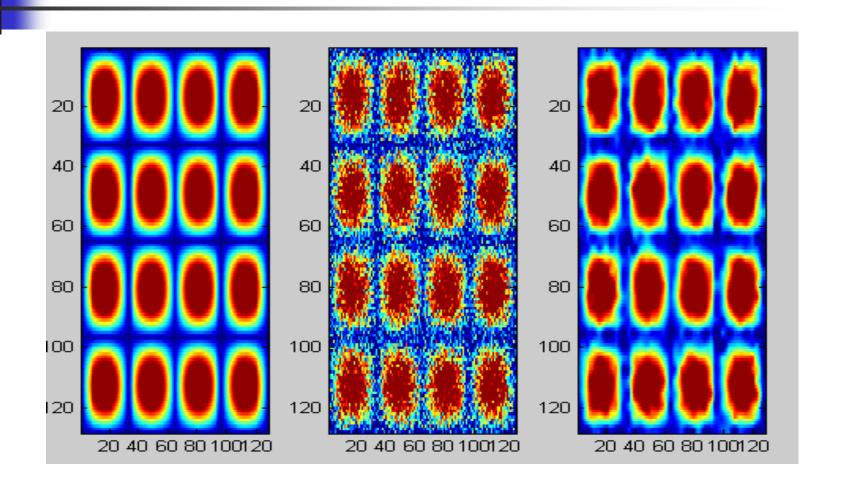
小波变换在图象处理中应用

> 例: 用小波变换对图象进行一层分解

X=imread('barbaran.tif');
[ca,ch,cv,cd]=dwt2(X,'bior3.7');
subplot(2,2,1);imshow(ca);
subplot(2,2,2);imshow(ch);
subplot(2,2,3);imshow(cv);
subplot(2,2,4);imshow(cd);



图象去噪/压缩: ddencmp(), wdencmp() load sinsin; subplot(1,3,1);image(X); x=X+15*randn(size(X));subplot(1,3,2);image(x); [thr,sorh,keepapp]=ddencmp('den','wv',x); xd=wdencmp('gbl',x,'sym4',2,thr,sorh,keepapp); subplot(1,3,3);image(xd);





5.4 图像增强

• 图象处理:

图象预处理—特征提取—识别分析 预处理:增强和复原

• 图象增强:

去除噪声、边缘增强、提高对比度、增加亮度等

• 按作用域划分:

空域增强法和频域增强法

1) 空域增强法

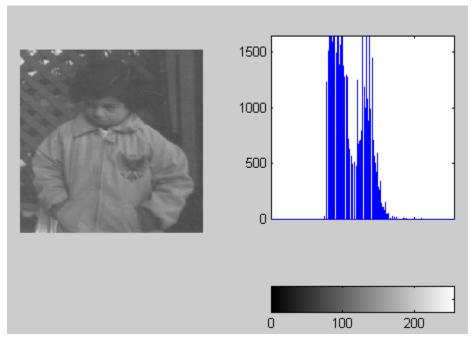
- 空域变换增强法(基于像素点)
- ❖ 直方图:

```
Matlab提供显示直方图的函数:
imhist(I, n)
imhist(X, map)
[counts, x]=imhist(...)
```



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

```
subplot(1,2,1);
imshow(I);
axis square;
subplot(1,2,2);
imhist(I);
axis square;
```





* 直方图处理

直方图均衡化: 给定图象的直方图分布改造成均匀直方图分布,增加对比度。

Matlab提供了直方图均衡化的函数

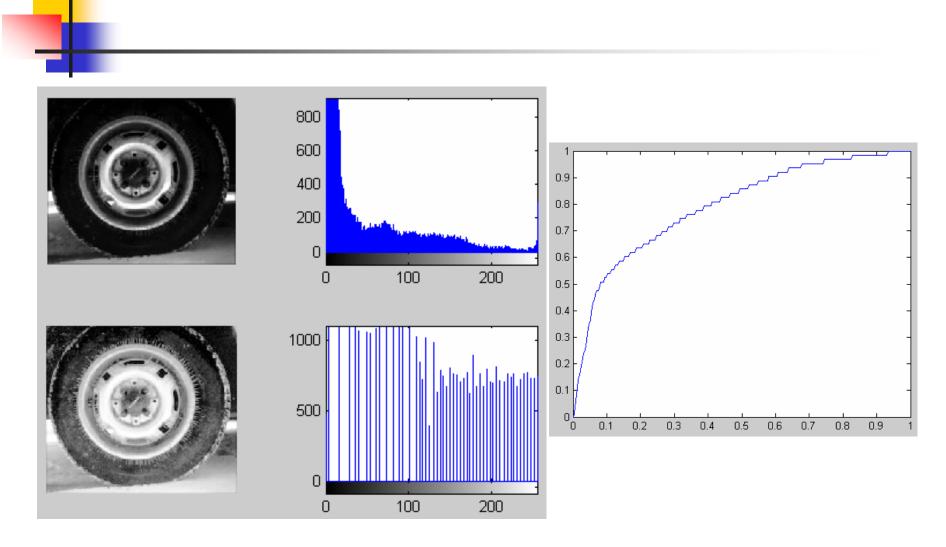
J=histeq(I, n)

J=histeq(I, hgram)

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



```
I=imread('tire.tif');
subplot(2,2,1);imshow(I);
subplot(2,2,2);imhist(I);
[J,T]=histeq(I);
subplot(2,2,3);imshow(J);
subplot(2,2,4);imhist(J);
figure; plot((0:1/255:1),T);
```





* 采用对数变换来改善图象质量

I = imread('pout.tif');

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

I = double(I);

J=log(I+1);

subplot(1,2,2);

imshow(mat2gray)







* 对比度扩展增强图象:

把灰度范围分布到整个灰度区间,使得亮的更亮,暗的更暗。

Matlab提供了实现图象灰度级变换的函数

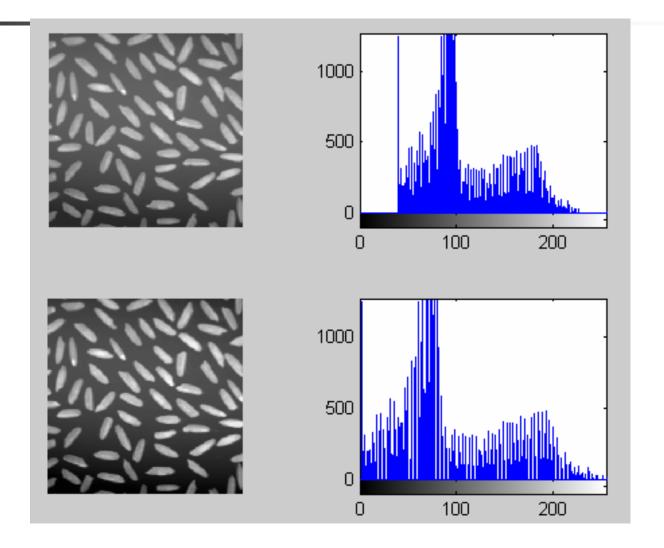
J=imadjust(I, [low_jn high_in], [low_out high_out], gamma)

I, J为输入和输出图象矩阵, low_in和high_in用来指定输入图象需要影射的灰度范围, low_out和high_out指定输出图象的灰度范围, gamma表示影射方式, 默认为线性影射。

4

```
例:
l=imread('rice.tif');
subplot(2,2,1);imshow(I);
subplot(2,2,2);imhist(I);
J=imadjust(I,[0.15 0.9],[0 1]);
subplot(2,2,3);imshow(J);
subplot(2,2,4);imhist(J);
```

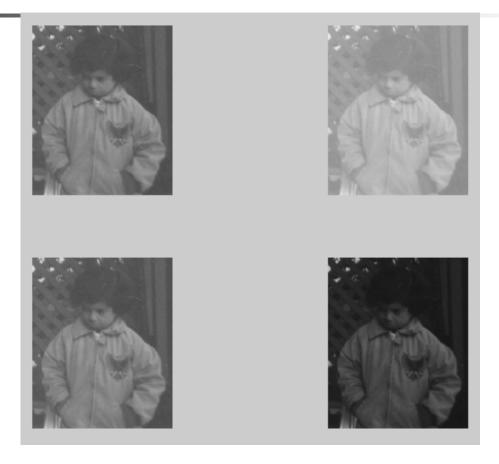






❖ Gamar参数校正 I=imread('pout.tif'); subplot(2,2,1);imshow(I); J1 = imadjust(I,[],[],0.5);J2=imadjust(I,[],[],1); J3=imadjust(I,[],[],2);subplot(2,2,2);imshow(J1); subplot(2,2,3);imshow(J2); subplot(2,2,4);imshow(J3);







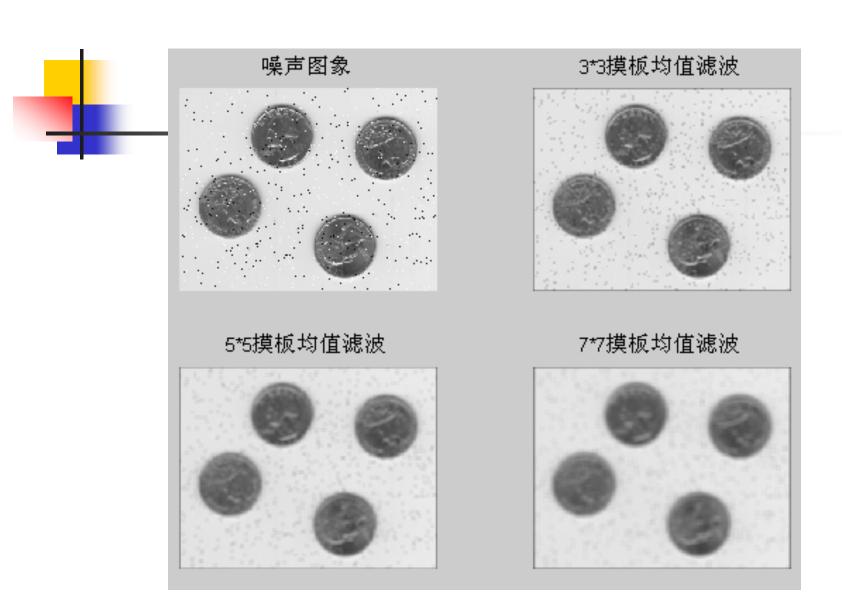
• 空域滤波增强(基于模板)

h=fspecial('type') h=fspecial('type', paramters) type:滤波器的种类 parameters:与滤波器有关的具体参数 filter2函数实现指定滤波器摸板对图象的操作 *平滑滤波器:消除图象噪声,图象变模糊。

4

例:线性平滑滤波:

```
I=imread('eight.tif');
J=imnoise(I,'salt & pepper',0.02);
subplot(2,2,1);imshow(J);title('噪声图象');
K1=filter2(fspecial('average',3),J)/255;
K2=filter2(fspecial('average',5),J)/255;
K3=filter2(fspecial('average',7),J)/255;
subplot(2,2,2);imshow(K1);title('3*3摸板均值滤波');
subplot(2,2,3);imshow(K2);title('5*5摸板均值滤波');
subplot(2,2,4);imshow(K3);title('7*7摸板均值滤波');
```





```
❖ 自适应滤波: j=wiener(I,[m,n])
```

◆ 中值滤波器:函数 B=medfilt2(A,[m n])

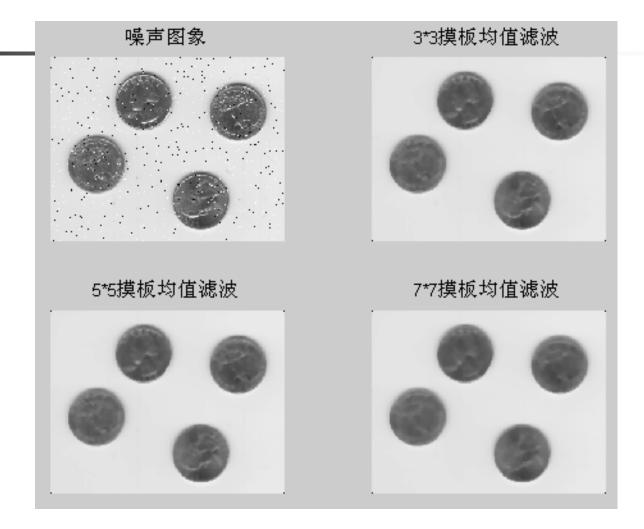
上例改为

K1 = medfilt2(J,[3 3]);

K2=medfilt2(J,[5 5]);

K3=medfilt2(J,[7 7]);







* 锐化滤波器:

```
I=imread('saturn.tif');
h=fspecial('laplacian');
                         可以sobel、canny等
                          原始图象
12=filter2(h,I);
subplot(1,2,1);
imshow(I);
title('原始图象');
subplot(1,2,2);imshow(12);
title('滤波后图象');
```

滤波后图象



2) 频域增强:

- 低通滤波
- 高通滤波
- 带通和带阻滤波
- 同态滤波



5.4 图像分割

图象分割:

将数字图象划分为不交叠的、连通的象素集的过程,其目的是把物体从背景中分离出来。

边缘检测

用边缘算子提取边缘点集;

在边缘点集合中去除某些边缘点,填充一些边缘点,将得到的边缘点集连接为线。

Matlab提供了edge 函数实现边缘检测



- BW=edge(I,method)
- BW=edge(I,method,thresh)
- BW=edge(I,method, thresh,direction)

method:使用算子的类型

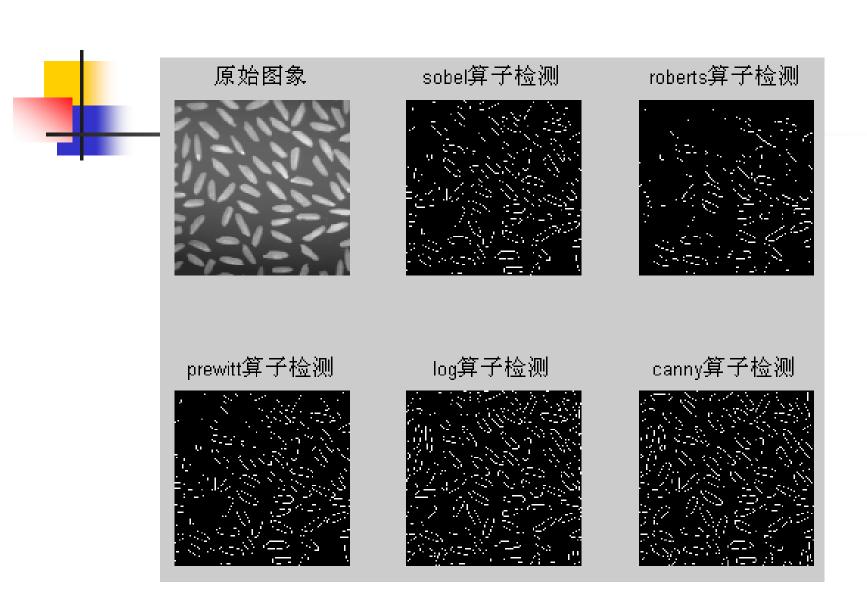
thresh:指定的域值

direction:对一些算子指定方向

BW:返回的二进制图象,1表示边缘。

4

```
I=imread('rice.tif');
BW1=edge(I,'sobel');
BW2=edge(I,'roberts');
BW3=edge(I,'prewitt');
BW4=edge(I,'log');
BW5=edge(I,'canny');
subplot(2,3,1);imshow(I);title('原始图象');
subplot(2,3,2);imshow(BW1);title('sobel算子检测');
subplot(2,3,3);imshow(BW2);title('roberts算子检测');
subplot(2,3,4);imshow(BW3);title('prewitt算子检测');
subplot(2,3,5);imshow(BW4);title('log算子检测');
subplot(2,3,6);imshow(BW5);title('canny算子检测');
```

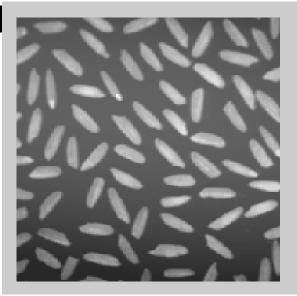


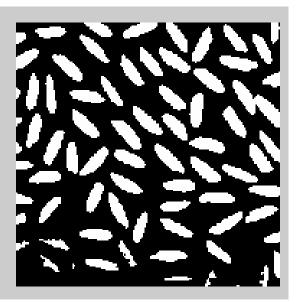


• 域值分割:

I=imread('rice.tif');
subplot(1,2,1);imshow(I);
BW=im2bw(I,graythresh(I));

subpl







• 四叉树分解:

将一块图象分成四块等大小的方块,判断每块是否满足一致性标准,如果满足,则不再分解;如果不满足,再进行细分使之成四块,进行一致性判断,重复迭代下去,直到满足一致性准则。

s=qtdecomp(I,threshold,[mindim maxdim])



```
I=imread('rice.tif');
subplot(1,2,1);imshow(I);
S=qtdecomp(I,0.2);
N=full(S);
subplot(1,2,2);ims
```

5.6 图像分析

• 形态学处理

```
基本形态学运算: 膨胀、腐蚀、开、闭 imdilat() imerode() imopen() imclose() strel(shape,para) bw = imread('text.png'); se=[0 1 0;1 1 1;0 1 0]; %se = strel('line',11,90); bw2 = imdilate(bw,se); imshow(bw), title('Original') figure, imshow(bw2), title('Dilated')
```



The term watershed refers to a ridge that ...

The term watershed refers to a ridge that ...

The term watershed refers to a ridge that ...

The term watershed refers to a ridge that ...

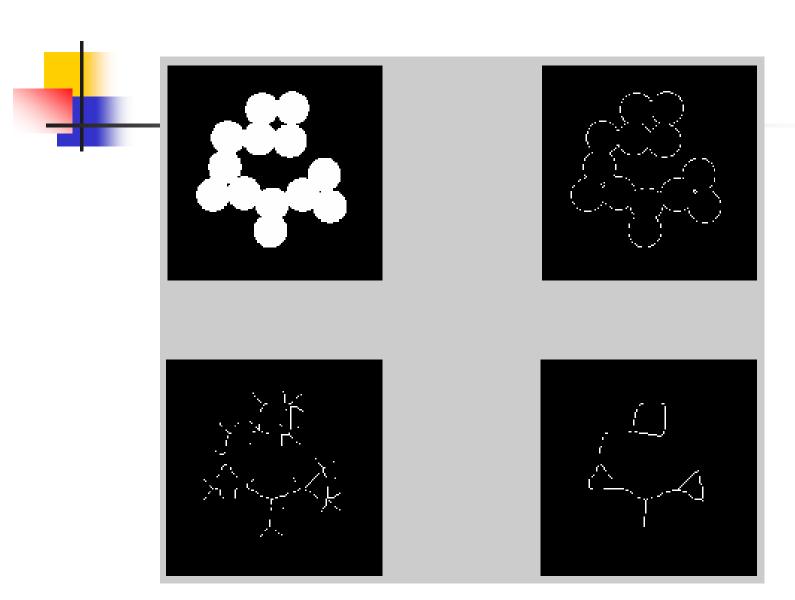
The term watershed refers to a ridge that ...

The term watershed refers to a ridge that ...

The term watershed refers to a ridge that ...

5.6 图像分析

组合形态学运算 BW2=bwmorph(BW1,opera) 例: BW1=imread('circles.tif'); subplot(2,2,1);imshow(BW1); BW2=bwmorph(BW1,'remove'); BW3=bwmorph(BW1,'skel',inf); BW4=bwmorph(BW1,'thin',inf); subplot(2,2,2);imshow(BW2); subplot(2,2,3);imshow(BW3); subplot(2,2,4);imshow(BW4);





• 特征提取和分析

灰度特征、纹理特征和几何形状特征等 形状分析、纹理分析、曲线和曲面分析等

图象面积: bwarea

边界标记: bwperim



```
BW1=im2bw(BW);
BW2=bwperim(BW1,8);
subplot(1,2,1);
imshow(BW1);
subplot(1,2,2);
imshow(BW2);
```

BW=imread('blood1.tif');

例:标识钢的显微图象中的不同颗粒

load imdemos steel;

bw_70=steel>70;

bw_210=steel>210;

imshow(steel);

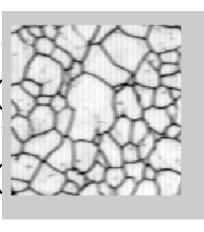
figure;

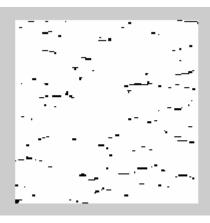
subplot(

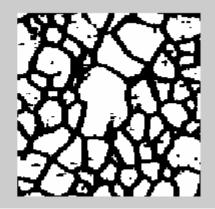
imshow(

subplot(

imshow(









```
[r,c] = find(bw_70 = 0);
bw_clean=bwselect(~bw_210,c,r,8);
imshow(bw_clean);figure;
bw_skel=bwmorph(bw clean,'skel',6);
subpl
imsho
imshow(bw_pruned);
```



```
grain_boundaries = ~ bw_pruned;
subplot(1,2,1);imshow(grain_boundaries);
[labeled] = bwlabel(grain_boundaries);
colored=label2rgb(labeled);
subplot(1,2
```