图像变换代码

Demo1

I=imread('tire.tif');

fun=@(block\_struct)std2(block\_struct.data) \* ones(size(block\_struct.data));

I2=blockproc(I,[8 8],fun);

imshow(I);

figure,imshow(I2,[]);

Demo2

fun = @(block\_struct) imresize(block\_struct.data,0.15);

I = imread('pears.png');

I=im2double(I);

I2 = blockproc(I,[100 100],fun);

figure;imshow(I);

figure;imshow(I2);

fun1=@(block\_struct)mean(mean(block\_struct.data));

I3=blockproc(I,[2 2],fun1);

figure,imshow(I3);

fun2=@(block\_struct)mean(block\_struct.data(:));

I4(:,:,1)=blockproc(I(:,:,1),[2 2],fun2);

I4(:,:,2)=blockproc(I(:,:,2),[2 2],fun2);

I4(:,:,3)=blockproc(I(:,:,3),[2 2],fun2);

figure;imshow(I4);

Demo3

B = reshape(uint8(1:25),[5 5])'

C = im2col(B,[1 5])

A = col2im(C,[1 5],[5 5],'distinct')

Demo4

A = reshape(linspace(0,1,16),[4 4])'

B = im2col(A,[2 2])

M = mean(B)

newA = col2im(M,[1 1],[3 3])

demo5

I = imread('eight.tif');

imshow(I);

BW=roipoly;

[x, y, BW, xi, yi] = roipoly %解释x，y等个参数的意义。

Demo6

load clown

BW = roicolor(X,10,20);

imshow(X,map)

figure,imshow(BW)

demo7

I=imread('eight.tif');

%先手动方式获取ROI的坐标并赋值给c和r。

BW=roipoly(I,c,r);

h=fspecial('unsharp'); %指定滤波算子为unsharp

J=roifilt2(h,I,BW);

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

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

demo8

I=imread('rice.jpg');

c=[52 72 300 270 221 194];

r=[71 21 75 121 121 75]; %指定滤波区域为 c和 r确定的多边形

J=roifill(I,c,r);

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

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

demo9

clear;

%生成一个黑白图形，图形中间有一个400\*10的白色区域

height=500;width=300;

a=200;b=5;

A((1:height),(1:width))=1;

A(((height-2\*a)/2:(height-2\*a)/2+2\*a),((width-2\*b)/2:(width-2\*b)/2+2\*b))=0;

imshow(A);

%傅里叶变换

B=fftshift(fft2(A));

%显示变化结果

figure('color','white'), imshow(log(abs(B)),[]);colormap(jet(64)),colorbar;

figure('color','white'), mesh(log(abs(B)));

demo10

%土星的傅里叶变换

load imdemos saturn2;

imshow(saturn2);

B=fftshift(fft2(saturn2));

figure,imshow(log(abs(B)+1),[]);

colormap(jet(64));

colorbar;

demo11

A=1:10

B=fftshift(A)

A=magic(11)

B=fftshift(A)

Demo12

%傅里叶反变换还原图像

ifft2(fft(A))

demo 12.01 自定义高斯低通滤波器

%only for gray image

clear;

f=imread('rice.png');

if isa(f,'double')==0

f=im2double(f);

end

subplot(221),imshow(f),title('origin image');

[M,N]=size(f);

x=linspace(-M,M,2\*M);

y=linspace(-N,N,2\*N);

[X,Y]=meshgrid(y,x);

D=sqrt(M.\*M+N.\*N);

sigma=D/10;

Z=exp(-(X.\*X+Y.\*Y)/2/sigma/sigma);

subplot(223),mesh(Z),title('gaussian filter');

F=fft2(f,2\*M,2\*N);

H=fftshift(Z);

subplot(224),mesh(H);title('shift filter center');

FH=F.\*H;

g=abs(ifft2(FH));

subplot(222),imshow(g(1:M,1:N)),title('filtered image');

demo12.02自定义理想低通滤波器

%only for gray image

clear;

f=imread('rice.png');

if isa(f,'double')==0

f=im2double(f);

end

subplot(221),imshow(f);title('origin image');

[M,N]=size(f);

x=linspace(-M,M,2\*M);

y=linspace(-N,N,2\*N);

[X,Y]=meshgrid(y,x);

D=sqrt(M.\*M+N.\*N);

sigma=D/5;

Z=sqrt(X.\*X+Y.\*Y)<=sigma;

subplot(223),mesh(Z),title('ILPF filter');

F=fft2(f,2\*M,2\*N);

H=fftshift(Z);

subplot(224),mesh(H);title('shift filter center');

FH=F.\*H;

g=abs(ifft2(FH));

subplot(222),imshow(g(1:M,1:N)),title('filtered image');

demo 12.03自定义巴特沃斯低通滤波器

%only for gray image

clear;

f=imread('rice.png');

if isa(f,'double')==0

f=im2double(f);

end

subplot(221),imshow(f),title('origin image');

[M,N]=size(f);

x=linspace(-M,M,2\*M);

y=linspace(-N,N,2\*N);

[X,Y]=meshgrid(y,x);

D=sqrt(M.\*M+N.\*N);

sigma=D/5;

n=1;

Z=1./(1+(sqrt(X.\*X+Y.\*Y)./sigma).^(2\*n));

subplot(223),mesh(Z),title('btw filter');

F=fft2(f,2\*M,2\*N);

H=fftshift(Z);

subplot(224),mesh(H),title('shift filter center');

FH=F.\*H;

g=abs(ifft2(FH));

subplot(222),imshow(g(1:M,1:N)),title('filtered image');

demo12.5高斯滤波平滑和锐化图像

clc;

clear;

I=imread('lena.jpg');

%I=rgb2gray(I);

%I=imnoise(I, 'gaussian', 0, 0.01); %%给图像添加高斯噪声；

subplot(231),imshow(I);

title('原图像');

s=fftshift(fft2(I)); %%低频部分移动到中心

subplot(234),imshow(log(abs(s)+1),[]);

title('图像傅里叶变换取对数所得频谱');

[a,b]=size(s);

a0=round(a/2);

b0=round(b/2);

%%%%%%%低通滤波

d=min(a0,b0)/10; %滤波的范围，以频谱图上的欧氏距离为依据

filter=zeros(a,b);

for i=1:a %设置低通滤波

for j=1:b

distance=sqrt((i-a0)^2+(j-b0)^2);

if distance<=d

h=1;

else

h=0;

end

filter(i,j)=h;

end

end

Lp=filter.\*s;

subplot(235),imshow(log(abs(Lp)+1),[]);

title('低通滤波频谱');

LPJ=uint8(real(ifft2(ifftshift(Lp))));

subplot(232),imshow(LPJ);

title('低通滤波所得图像');

Hp=(~filter).\*s;

subplot(236),imshow(log(abs(Hp)+1),[]);

title('高通滤波频谱');

HPJ=uint8(real(ifft2(ifftshift(Hp))));

subplot(233),imshow(HPJ);

title('高通滤波所得图像');

new=Hp.\*1.5+Lp.\*1;

NewIm=uint8(real(ifft2(ifftshift(new))));

figure,imshow(NewIm);

title('增强高频信号所得图像');

demo12.51高斯滤波平滑和锐化图像

clear;

I=imread('lena.jpg');

subplot(3,3,1),imshow(I),title('原始图像');

S=fftshift(fft2(I));

subplot(3,3,4),imshow(log(abs(S)),[]),colormap(jet(64)),title('FFT变换结果');

[a,b]=size(S);

a0=round(a/2);

b0=round(b/2);

d=round(min(a0,b0)/10);

Lp=zeros(a,b);

Lp(a/2-d:a/2+d,b/2-d:b/2+d)=1;

% imshow(Lp);

Lps=Lp;

Lp=Lp.\*S;

subplot(3,3,5),imshow(log(abs(Lp)),[]),colormap(jet(64)),title('低通滤波器');

LPJ=uint8(real(ifft2(ifftshift(Lp))));

subplot(3,3,2),imshow(LPJ),title('低通滤波结果');

Hp=1-Lps;

Hp=Hp.\*S;

subplot(3,3,6),imshow(log(abs(Hp)),[]),colormap(jet(64)),title('高通滤波器');

HPJ=uint8(abs(ifft2(ifftshift(Hp))));

subplot(3,3,3),imshow(HPJ),title('高通滤波结果');

new=Hp\*1.5+Lp\*1;

newIm=uint8(abs(ifft2(ifftshift(new))));

subplot(3,3,7),imshow(newIm),title('锐化结果');

demo12.5101三个高通滤波器

% high pass filter

%gaussian

M=256; N=256;

x=linspace(-M,M,2\*M);

y=linspace(-N,N,2\*N);

[X,Y]=meshgrid(y,x);

D=sqrt(M.\*M+N.\*N);

sigma=D/8;

G=exp(-(X.\*X+Y.\*Y)/2/sigma/sigma);

subplot(231),mesh(G),title('gaussian high pass filter');

subplot(234),imshow(G);

%ihpf

I=sqrt(X.\*X+Y.\*Y)<=sigma;

subplot(232),mesh(I),title('ILPF highpass filter');

subplot(235),imshow(I);

%btw

n=1;

B=1./(1+(sqrt(X.\*X+Y.\*Y)./sigma).^(2\*n));

subplot(233),mesh(B),title('btw high pass filter');

subplot(236),imshow(B);

demo13

clear;

A=magic(3);

B=ones(3);

A(8,8)=0; %为了进行快速卷积，将矩阵 A补零

B(8,8)=0; %为了进行快速卷积，将矩阵 B补零

C=ifft2(fft2(A).\*fft2(B))

C=C(1:5,1:5)

C=real(C)

%卷积验证

D=conv2(A,B);

D=D(1:5,1:5)

Demo14

%学生自己做，给学生图片

clear;

a=imread('a.png'); %读入字母 a

a=im2bw(a);

a=mat2gray(a);

[ma,na]=size(a); %求出 a的大小

I=imread('text.png'); %读入图像'text.tif

I=mat2gray(I);

[mi,ni]=size(I); %求出图像的大小

afft=fft2(a);

Ifft=fft2(I);

N=na+ni-1;

M=ma+mi-1; %求出卷积之后图像的大小

afft(M,N)=0; %对字符 a的 fft变换补零，以利于快速卷积

Ifft(M,N)=0; %对text图像的 fft变换补零，以利于快速卷积

filtered=ifft2(afft.\*Ifft); %利用 fft和卷积的关系快速计算矩阵卷积

imshow(abs(filtered),[]);

filtered=filtered(1:mi,1:ni); %只保留原图像大小

filtered=filtered/max(max(filtered,[],1)); %对卷积之后的图像归一化

result=filtered>0.90; %设置门限求出匹配点

figure,imshow(result);

demo15

RGB=imread('autumn.tif');%将彩色图像转化为灰度图像，以便进行DCT变换

I=rgb2gray(RGB);

imshow(I)

J=dct2(I);

figure,imshow(log(abs(J)),[]);

colormap(jet((64)));

colorbar;

demo16

%接上例

J(abs(J)<10)=0;%将DCT变换值小于10的元素设为0。

K=idct2(J)/255;%对逆DCT变换值归一化。

figure,imshow(K);

demo17

I=imread('cameraman.tif');

I=im2double(I);

T=dctmtx(8); %产生二维DCT变换矩阵

fun = @(block\_struct)T\*block\_struct.data\*T';

B=blockproc(I,[8 8],fun);

mask=[1 1 1 1 0 0 0 0

1 1 1 0 0 0 0 0

1 1 0 0 0 0 0 0

1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0];%二值掩模，用来压缩 DCT的系数

fun2=@(block\_struct)mask.\*block\_struct.data;

B2=blockproc(B,[8 8],fun2);%只保留DCT变换的10个系数

fun3=@(block\_struct)T'\*block\_struct.data\*T;

I2=blockproc(B2,[8 8],fun3);%逆DCT变换，用来重构图像

imshow(I);

figure,imshow(I2);

demo18

clear;

I=zeros(100,100); %产生一个正方形的黑框

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

imshow(I);

[R,xp]=radon(I,[0 45]); %计算黑框的radon变换

figure,plot(xp,R(:,1)),title('R\_{0^o}(x\prime)'); %显示黑框在0º的 radon变换

figure,plot(xp,R(:,2)),title('R\_{45^o}(x\prime)'); %显示黑框在45º的radon变换

demo19

%直线检测

clear;

I=zeros(200,200);

A=eye(100,100);

I(101:200,1:100)=A;

figure,imshow(I);title('origin image');

theta=0:179;

[R,xp]=radon(I,theta);

figure,imagesc(theta,xp,R);title('R\_{\theta} X');

xlabel('{\theta}(degree)');

ylabel('x\prime');

set(gca,'XTick',0:5:180);%设置坐标轴属性

set(gca,'YTick',-100:10:100);%设置坐标轴属性

colormap(hot);

colorbar;

demo20

%大脑重建

>> P=phantom(256);

>> imshow(P);

>> theta1=0:10:170;[R1,xp1]=radon(P,theta1);

>> theta2=0:5:175;[R2,xp2]=radon(P,theta2);

>> theta3=0:2:178;[R3,xp3]=radon(P,theta3);

>> figure,imagesc(theta3,xp3,R3);title('R\_{theta}X');

>> colormap(hot);

>> IR1=iradon(R1,theta1);

>> figure,imshow(IR1);

>> IR2=iradon(R2,theta2);

>> figure,imshow(IR2);

>> IR3=iradon(R3,theta3);

>> figure,imshow(IR3);

Demo21

查看分接结果。

load noisdopp;

plot(noisdopp);title('原始信号');

[A D]=dwt(noisdopp,'sym4');

figure,plot(A);title('原始信号近似信号');

figure,plot(D);title('原始信号细节信号')

[A1 D1]=dwt(A,'sym4'); %对近似分量再进行分解。

figure,plot(A1);title('近似信号近似信号');

figure,plot(D1);title('近似信号细节信号');

Demo22

% Set wavelet name.

wname = 'db5';

% Compute the four filters associated with wavelet name given

% by the input string wname.

[Lo\_D,Hi\_D,Lo\_R,Hi\_R] = wfilters(wname);

subplot(221); stem(Lo\_D);

title('Decomposition low-pass filter');

subplot(222); stem(Hi\_D);

title('Decomposition high-pass filter');

subplot(223); stem(Lo\_R);

title('Reconstruction low-pass filter');

subplot(224); stem(Hi\_R);

title('Reconstruction high-pass filter');

xlabel('The four filters for db5')

demo23

%运行一下代码，查看各个向量的长度，验证分解后的小波长度公式

>> clear

>> load noisdopp;

>> [Lo\_D,Hi\_D]=wfilters('bior3.5','d');

>> [A D]=dwt(noisdopp,Lo\_D,Hi\_D);

Demo24

%小波重建，小波反变换

clear

load noisdopp;

[A,D]=dwt(noisdopp,'sym4');

X=idwt(A,D,'sym4');

plot(noisdopp);title('origin signal');

figure,plot(X);title('reconstructed signal');

deltaX=X-noisdopp;

figure,plot(deltaX); title('error');

Demo25

>> load leleccum;

>> s=leleccum(1:3920);

>> ls=length(s);

>> [C L]=wavedec(s,3,'db5');

>> a0=waverec(C,L,'db5');

>> err=norm(s-a0)

err =

1.6717e-09

Demo26

>> load leleccum;

>> sig=leleccum(1:3920);

>> [C L]=wavedec(sig,5,'sym4');

>> lev=3;

>> a3=appcoef(C,L,'sym4',lev);

>> subplot(2,1,1),plot(sig),title('raw signal');

>> subplot(2,1,2),plot(a3),title('level-3 approximation coefficients');

>> length(a3) %???

Demo27

>> load leleccum;

>> sig=leleccum(1:3920);

>> [c l]=wavedec(sig,5,'sym4');

>> d=detcoef(c,l);

>> plot(d);

>> d1=detcoef(c,l,1);

>> figure,plot(d1);

>> d2=detcoef(c,l,2);

>> figure,plot(d2);

Demo28

%demo28

load noisdopp;

Y=noisdopp(1:100);

[cA,cD] = dwt(Y,'sym4');

xrec=idwt(cA,[],'sym4');

% rec=upcoef('d',cA,'sym4',1);

% xrec2=idwt(xrec,[],'sym4');

figure(1);

for i=1:6

rec=upcoef('d',cA,'sym4',i);

essup=length(rec);

ax=subplot(6,1,i);

plot(rec(1:essup));

set(ax,'xlim',[1 3370]);

end

subplot(611);

title(['Approximation signals, obtained from a single'...

' coefficient at levels 1 to 6']);

Demo29

load sumsin;

s=sumsin;

[c l]=wavedec(s,5,'sym4');

a5=wrcoef('a',c,l,'sym4',5);

d5=wrcoef('d',c,l,'sym4',5);

subplot(311),plot(s),title('Original signals');

subplot(312),plot(a5),title('Reconstructed approximation at level 5 :a5');

subplot(313),plot(d5),title('Reconstructed detail at level 5 :d5');

练习，分解出1-5层的细节信号和近似信号。分别在5个figure中画出原始信号、近似信号和细节信号。

Demo29.5

clear;

load sumsin;

s=sumsin;

level=7;

[c,l]=wavedec(s,level,'sym4');

a=zeros(level,length(s));

d=a;

for i=1:level

a(i,:)=wrcoef('a',c,l,'sym4',i);

d(i,:)=wrcoef('d',c,l,'sym4',i);

figure(i);

subplot(311),plot(s),title('raw signal');

str1 = sprintf('approximation signal at level %d',i);

str2 = sprintf('detail signal at level %d',i);

subplot(312),plot(a(i,:)),title(str1);

subplot(313),plot(d(i,:)),title(str2);

end

Demo30

load sumsin;

s=sumsin;

[c,l]=wavedec(s,3,'db1');

subplot(311),plot(s),title('Original signals');

subplot(312),plot(c),title('wavelet decomposition structure,level 3'),

xlabel('Coefs for approx. at level 3 and for det.at level 3,2,1'),

[nc,nl]=upwlev(c,l,'db1');

subplot(313),plot(nc),title('Wavelet decomposition structure at level 2'),

xlabel('Coefs for approx. at level 2 and for det. at level2 2,1');

%[c2,l2]=wavedec(s,2,'db1');

%figure,plot(c2);

%nc-c2

练习，直接用wavedec进行2层分解，画出分解之后的c，并与nc对比。

load sumsin;

s=sumsin;

[c,l]=wavedec(s,3,'sym4');

subplot(511),plot(s),title('original signal');

subplot(512),plot(c),title('decomposition structure,level 3');

xlabel('Coefs for approx. at level 3 and detail level 3,2,1');

%up to level 2(1)

[nc,nl]=upwlev(c,l,'sym4');

subplot(513),plot(nc),title('analysis structure level 2');

xlabel('Coefs for approx. at level 2 and detail level2,2,1');

[nnc,nnl]=upwlev(nc,nl,'sym4');

subplot(514),plot(nnc),title('analysis structure level 1');

xlabel('Coefs for approx. at level 1 and detail level2,2,1');

[nnnc,~]=upwlev(nnc,nnl,'sym4');

subplot(515),plot(nnnc),title('analysis structure level 0');

xlabel('Coefs for approx. at level 1 and detail level2,2,1');

% directly to level2(2)

[cc,ll]=wavedec(s,2,'sym4');

%compare (1) and (2)

max(abs(nc-cc))

Demo31

%Demo31

%单层分解与重构

load leleccum;

s=leleccum(1:3920);

ls=length(s);

plot(s),title('Original signal');

[ca1,cd1]=dwt(s,'db1');

figure,plot(ca1),title('Appromiation signal coef at level 1');

figure,plot(cd1),title('detail signal coef at level 1');

a1=upcoef('a',ca1,'db1',1);

d1=upcoef('d',cd1,'db1',1);

figure,plot(a1),title('constructed Appromiation signal coef at level 1');

figure,plot(d1),title('constructed detail signal coef at level 1');

figure,plot(a1+d1),title('sum of Appromiation and detail signal coef at level 1 same as s ');

a0=idwt(ca1,cd1,'db1',ls);

figure,plot(a0),title('idwt reconstruction signal');% compare with a1+d1

demo32

%多层分解与重构

load leleccum;

s=leleccum(1:3920);

ls=length(s);

plot(s),title('Original signal');

[c,l]=wavedec(s,3,'db1');

%appcoef提取近似分量系数

ca3=appcoef(c,l,'db3',3);

figure,plot(ca3),title('the appromximation signal at level 3');

ca2=appcoef(c,l,'db3',2);

figure,plot(ca2),title('the appromximation signal at level 2');

%detcoef函数提取信号第3、2、1层的细节分量系数

cd3=detcoef(c,l,3);

cd2=detcoef(c,l,2);

cd1=detcoef(c,l,1);

figure,plot(cd3),title('detail signal coef at level 3');

figure,plot(cd2),title('detail signal coef at level 2');

figure,plot(cd1),title('detail signal coef at level 1');

%利用wrcoef函数重构信号第3层的近似信号

a3=wrcoef('a',c,l,'db1',3);

a2=wrcoef('a',c,l,'db1',2);

a1=wrcoef('a',c,l,'db1',1);

figure,plot(a3),title('approximated signal at level 3');

figure,plot(a2),title('approximated signal at level 2');

figure,plot(a1),title('approximated signal at level 1');

%利用wrcoef函数重构信号第3、2、1层的细节信号

d3=wrcoef('d',c,l,'db1',3);

figure,plot(d3'),title('the detail signal at level 3');

d2=wrcoef('d',c,l,'db1',2);

figure,plot(d2'),title('the detail signal at level 2');

d1=wrcoef('d',c,l,'db1',1);

figure,plot(d1'),title('the detail signal at level 1');

demo33

load woman;

imshow(X,map);

figure;

[cA,cH,cV,cD]=dwt2(X,'sym4');

subplot(221),imagesc(cV),title('Vertical detail image');

colormap(gray);

subplot(222),imagesc(cA),title('Lowpass Approximation');

subplot(223),imagesc(cH),title('Horizental detail image');

subplot(224),imagesc(cD),title('Diagonal detail image');

demo34

load woman;

imshow(X,map),title('Original Image');

figure;

[cA,cH,cV,cD]=dwt2(X,'sym4');

subplot(221),imagesc(cV),title('Vertical detail image');

colormap(gray);

subplot(222),imagesc(cA),title('Lowpass Approximation');

subplot(223),imagesc(cH),title('Horizental detail image');

subplot(224),imagesc(cD),title('Diagonal detail image');

figure;

A0=idwt2(cA,cH,cV,cD,'sym4');

imshow(A0,map);title('Reconstruct Image');

max(max(abs(X-A0)))

% A1=idwt2(A0,[],[],[],'sym4');

% figure,imshow(A1,map);

demo35

load woman;

[C,S]=wavedec2(X,2,'db1');

sizeX=size(X)

sizec=size(C)

S

sizeS=size(S)

思考，各个数字之间的关系。

Demo36

load woman;

[C,S]=wavedec2(X,2,'sym4');

a0=waverec2(C,S,'sym4');

imshow(X,map);title('origin image');

figure,imshow(a0,map);title('reconstructed image');

max(max(abs(X-a0)))

demo37

%读入图像数据，并对其颜色进行编码

load woman2;

nbcol=size(map,1);

colormap(pink(nbcol));

cod\_X=wcodemat(X,nbcol); %Extended pseudocolor matrix scaling

image(cod\_X);

axis('square');

%%单层小波变换

% [ca1,ch1,cv1,cd1]=dwt2(cod\_X,'db1');

% cod\_ca1=wcodemat(ca1,nbcol);

% cod\_ch1=wcodemat(cd1,nbcol);

% cod\_cv1=wcodemat(cv1,nbcol);

% cod\_cd1=wcodemat(cd1,nbcol);

% image([cod\_ca1,cod\_ch1;cod\_cv1,cod\_cd1]);

% axis('square'),title('approximated signal and 3 detail signals at level 1');

%

% %对图像进行二次分解

% [ca2,ch2,cv2,cd2]=dwt2(ca1,'db1');

% cod\_ca2=wcodemat(ca2,nbcol);

% cod\_ch2=wcodemat(ch2,nbcol);

% cod\_cv2=wcodemat(cv2,nbcol);

% cod\_cd2=wcodemat(cd2,nbcol);

% figure,image([cod\_ca2,cod\_ch2;cod\_cv2,cod\_cd2]);

% colormap(pink(nbcol));

% axis('square');

% title('approximated signal and 3 detail signals at 2th decomposition');

%

% %利用idwt2函数在第1层重构近似信号

% a0=idwt2(ca1,ch1,cv1,cd1,'db1',size(X));

% a0=wcodemat(a0, nbcol);

% figure,image(a0);colormap(pink(nbcol));title('construct on level 1');

% axis('square');

%

% %利用idwt2函数在第2层重构近似信号

% a1=idwt2(ca2,ch2,cv2,cd2,'db1',size(X));

% a1=wcodemat(a1, nbcol);

% figure,image(a1);colormap(pink(nbcol));

% title('construct on level 2');

% axis('square');

%%end of single-decomposition

%%wavedec2函数在第2层对图像进行近似分解

[c,s]=wavedec2(X,2,'db1');

%利用函数appcoef2和detcoef2从分解得到的结构[C,S]提取图像的第2层近似分量和细节分量

ca2=appcoef2(c,s,'db1',2);

ch2=detcoef2('h',c,s,2);

cv2=detcoef2('v',c,s,2);

cd2=detcoef2('d',c,s,2);

cod\_ca2=wcodemat(ca2,nbcol);

cod\_ch2=wcodemat(ch2,nbcol);

cod\_cv2=wcodemat(cv2,nbcol);

cod\_cd2=wcodemat(cd2,nbcol);

figure,colormap(pink(nbcol));

image([cod\_ca2,cod\_ch2;cod\_cv2,cod\_cd2]);

axis('square');

title('approximation amd detail coefficients at level 2');

%利用函数appcoef2和detcoef2从分解得到的结构[C,S]提取图像的第2层近似分量和细节分量

ca1=appcoef2(c,s,'db1',1);

ch1=detcoef2('h',c,s,1);

cv1=detcoef2('v',c,s,1);

cd1=detcoef2('d',c,s,1);

cod\_ca1=wcodemat(ca1,nbcol);

cod\_ch1=wcodemat(ch1,nbcol);

cod\_cv1=wcodemat(cv1,nbcol);

cod\_cd1=wcodemat(cd1,nbcol);

figure,colormap(pink(nbcol));

image([cod\_ca1, cod\_ch1;cod\_cv1, cod\_cd1]);

axis('square');

title('approximation amd detail coefficients at level 1');

%用wrcoef2函数在图像分解的第2层重构近似信号以及细节信号

a2=wrcoef2('a',c,s,'db1',2);

cod\_a2=wcodemat(a2,nbcol);

h2=wrcoef2('h',c,s,'db1',2);

v2=wrcoef2('v',c,s,'db1',2);

d2=wrcoef2('d',c,s,'db1',2);

cod\_h2=wcodemat(h2,nbcol);

cod\_v2=wcodemat(v2,nbcol);

cod\_d2=wcodemat(d2,nbcol);

figure,colormap(pink(nbcol));

image([cod\_a2,cod\_h2;cod\_v2,cod\_d2]);

axis('square');title('approximation amd detail coefficients at level 2 by wrcoef2');

%利用upwlev2来提取图像分解第1层的分解结构[C，S]

[C,S]=upwlev2(c,s,'db1');

%首先从第一层分解结构[C,S]中提取系数

ca1=appcoef2(C,S,'db1',1);

ch1=detcoef2('h',C,S,1);

cv1=detcoef2('v',C,S,1);

cd1=detcoef2('d',C,S,1);

%利用系数重构

siz=S(size(S,1),:);

a1=upcoef2('a',ca1,'db1',1,siz);

hd1=upcoef2('h',ch1,'db1',1,siz);

vd1=upcoef2('v',cv1,'db1',1,siz);

dd1=upcoef2('d',cd1,'db1',1,siz);

cod\_a1=wcodemat(a1,nbcol);

cod\_hd1=wcodemat(hd1,nbcol);

cod\_vd1=wcodemat(vd1,nbcol);

cod\_dd1=wcodemat(dd1,nbcol);

figure,colormap(pink(nbcol));

image([cod\_a1,cod\_hd1;cod\_vd1,cod\_dd1]);

axis('square');title('approximation amd detail coefficients at level 1 by upcoef2');

%从分解结构[C,S]中重构原始信号的近似分量

a0=waverec2(C,S,'db1');

cod\_a0=wcodemat(a0,nbcol);

figure,colormap(pink(nbcol));

image(cod\_a0);

axis('square');

%%end of muiltilevel decomposition