%主函数：

clear;clc;

%

im=imread('Lena.bmp');

im=double(im);

[M,C]=size(im);

count = 1; %¼ÆÊý

for i = 1:M/32

for j = 1:C/32

im0 = im((i-1)\*32+1 : i\*32, (j-1)\*32+1 : j\*32); %生成小图

[R,T]=size(im0);%计算尺寸

f1=DREAM(im0);%使用理想低通滤波器进行降质

%f1=lixiang(im0);% 使用理想低通滤波器进行降质

%f1=bas(im0);% 使用巴特沃斯低通滤波器进行降质

%f1=Gauss(im0,20);% 使用高斯低通滤波器进行降质

f1=f1(:);

N=length(f1);%N等于im的长度

H=eye(N);

%有约束最小二乘法

lambda=0.5;

f\_hat=(H'\*H+lambda\*eye(R\*T))\*H'\*f1;

im0=zeros(size(im0));

im0(:)=f\_hat;

%figure

%imshow(im0)

rebuild((i-1)\*32+1 : i\*32, (j-1)\*32+1 : j\*32) = im0; %重建原图

end

end

psnr=PSNR(im,rebuild);

figure;

subplot(121)

imshow(mat2gray(im))

title('Ô­Ê¼Í¼Ïñ')

subplot(122)

imshow(mat2gray(rebuild)); title('¸´Ô­ºóµÄÍ¼Ïñ')%显示重组后的图像

xlabel(['psnr=',num2str(psnr),'db']);

function psnr = PSNR(x,y)

x=double(x);

y=double(y);

mse=mean((x(:)-y(:)).^2);

psnr=10\*log10((255^2)/mse);

end

function [f1]=DREAM(im0)

%利用理想低通滤波器对图像进行滤波

im0=double(im0);

M=2\*size(im0,1); %滤波器的行数

N=2\*size(im0,2); %滤波器的列数

u=-M/2:(M/2-1);

v=-N/2:(N/2-1);

[U,V]=meshgrid(u,v);

D=sqrt(U.^2+V.^2);

D0=80; %截止频率为80

H=double(D<=D0); %理想低通滤波器

J=fftshift(fft2(im0,size(H,1),size(H,2))); %时域图像转换到频域

K=J.\*H; %滤波处理

f1=ifft2(ifftshift(K)); %傅里叶反变换

f1=f1(1:size(im0,1),1:size(im0,2));

end

function [f1] = lixiang( im0 )

im0=im2double(im0);

Fim0=fft2(double(im0));%傅里叶变换

Fim0=fftshift(Fim0);%将变换的原点移到频率矩形的中心

[M,N]=size(im0);

%半径为50的低通滤波器处理

z1=zeros(M,N);

for i=1:M

for j=i:N

if(sqrt(((i-M/2)^2+(j-N/2)^2))<50)

z1(i,j)=1;

end

end

end

g1=Fim0.\*z1;

g1=ifftshift(g1);

f1=real(ifft2(g1));

end

%利用巴特沃斯低通滤波器对图像进行处理

function [f1]=bas(im0)

M=2\*size(im0,1); %滤波的行数

N=2\*size(im0,2); %滤波的列数

u=-M/2:(M/2-1);

v=-N/2:(N/2-1);

[U,V]=meshgrid(u,v);

D=sqrt(U.^2+V.^2);

D0=50;

n=6;

H=1./(1+(D./D0).^(2\*n)); %构造巴特沃斯滤波器

J=fftshift(fft2(im0,size(H,1),size(H,2))); %转换到频域

K=J.\*H;

f1=ifft2(ifftshift(K)); %傅里叶反变换

f1=f1(1:size(im0,1),1:size(im0,2)); %改变图像大小

end

function [f1]=Gauss(im0,D0)

im0 =double(im0); %读取输入图像

%得到高斯低通滤波器

[r,c] = size(im0); %获取图像的行和列

D = zeros(r,c); %D(u,v)是距离频域矩形的中心距离

for i=1:r

for j=1:c

D(i,j)=sqrt((i-r/2)^2+(j-c/2)^2);

end

end

Hf=exp(-(D.^2)/(2\*D0\*D0)); %计算滤波器，得到高斯低通滤波器

a=fft2(im0,size(Hf,1),size(Hf,2)); %对原图像进行反傅里叶变换

a=fftshift(a); %对傅里叶变换后的F进行中心移位

b=ifft2(ifftshift(Hf.\*a)); f1=real(b); %从结果中获取幅度

end