实验1图像的基本操作

（1）

I1=imread('football.jpg');

I2=imread('cameraman','tif');

I3=imread('onion.png');

figure,

subplot(1,3,1),imshow(I1),title('football');

subplot(1,3,2),imshow(I2),title('cameraman');

subplot(1,3,3),imshow(I3),title('onion');

（2）

I1=imread('rice.png');

I2=imread('cameraman','tif');

I3=imadd(I1,I2);

imwrite(I3,'rice\_cameraman.tif')

figure,

subplot(1,3,1),imshow(I1),title('rice');

subplot(1,3,2),imshow(I2),title('cameraman');

subplot(1,3,3),imshow(I3),title('rice\_cameraman');

（3）

I1=imread('cameraman','tif');

I2=I1(1:2:end,1:2:end);

I3=I1(1:4:end,1:4:end);

figure,

subplot(1,3,1),imshow(I1),title('cameraman原图')

subplot(1,3,2),imshow(I2),title('4倍采样')

subplot(1,3,3),imshow(I3),title('16倍采样')

（4）

I1=imread('cameraman','tif');

I2=(I1/4)\*4;

I3=(I1/8)\*8;

I4=(I1/32)\*32;

figure,

subplot(2,2,1),imshow(I1),title('cameraman原图')

subplot(2,2,2),imshow(I2),title('64级灰度图')

subplot(2,2,3),imshow(I3),title('32级灰度图')

subplot(2,2,4),imshow(I4),title('8级灰度图')

实验2图像的空域与几何运算

（1）

[im,map]=imread('canoe.tif'); %此为索引图

im1=ind2rgb(im,map); %将索引图转换为真彩图（0-1）

im2=imadd(im1,30/255); %注意数据范围

im3=ind2gray(im,map); %将索引图转为灰度图（0-255）

im4=imadd(im3,30); %注意数据范围

figure,imshow(im,map);

figure

subplot(1,2,1),imshow(im2);

subplot(1,2,2),imshow(im4);

（2）

im1=imread('office\_1.jpg');

im2=imread('office\_2.jpg');

im3=imdivide(im1,im2);

im4=imdivide(im2,0.5);

figure,

subplot(1,2,1), imshow(im1);title('office\_1原图');

subplot(1,2,2), imshow(im2);title('office\_2原图');

figure,

imshow(im3);title('office\_1除以office\_2.jpg的图像');

figure,

imshow(im4);title('office\_2除0.5的图像');

（3）

a=imread('pillsetc.png');

b=im2bw(a,0.4);

x(:,:,1)=b; x(:,:,2)=b; x(:,:,3)=b;

y=~x;

c=immultiply(x,a);

d=immultiply(y,a);

figure,

subplot(2,2,1); imshow(a); title('原图');

subplot(2,2,2); imshow(b); title('二值图像');

subplot(2,2,3); imshow(c); title('目标图');

subplot(2,2,4); imshow(d); title('背景图');

（4）

a=imread('pout.tif');

b=imresize(a,0.5,'nearest');

c=imresize(a,[150,200],'nearest');

d=imresize(a,1.5,'nearest');

e=imresize(a,[450,300],'nearest');

figure;

subplot(1,3,1); imshow(a); title('原图');

subplot(1,3,2); imshow(b); title('按比例缩小图');

subplot(1,3,3); imshow(c); title('不按比例缩小图');

figure;

subplot(1,3,1);imshow(a); title('原图');

subplot(1,3,2);imshow(d); title('按比例放大图');

subplot(1,3,3);imshow(e); title('不按比例放大图');

（5）

im=imread('pout.tif');

figure(1); subplot(3,3,1); imshow(im);title('原图');

[x,y]=size(im);

a=zeros(x,y);

a(21:end,21:end)=im(1:end-20,1:end-20);

figure(1); subplot(3,3,2); imshow(uint8(a)); title('平移图');

b=imrotate(im,25,'bilinear');

figure(1);subplot(3,3,3); imshow(b);title('旋转25度图');

c=imresize(im,[291,120],'nearest');

figure(1); subplot(3,3,4); imshow(c); title('水平压缩一倍图');

d=imresize(im,[146,240],'nearest');

figure(1);subplot(3,3,5); imshow(d); title('垂直压缩一倍图');

b=zeros(x,y);

b(1:end,y:-1:1)=im(1:end,1:end);

figure(1); subplot(3,3,6); imshow(uint8(b)); title('水平镜像图');

b1=zeros(x,y);

b1(x:-1:1,1:end)=im(1:end,1:end);

figure(1); subplot(3,3,7); imshow(uint8(b1)); title('垂直镜像图');

h=imrotate(im,30,'bilinear');

[p,q]=size(h);

z=zeros(p,q);

z(1:end,q:-1:1)=h(1:end,1:end);

figure(2); imshow(uint8(z)); title('组合图片');

实验3 图像增强-空域

（1）

I=imread('D:/pout.tif');

J=imadjust(I,[0.1 0.5],[0 1],0.4);

K=imadjust(I,[0.1 0.5],[0 1],4);

figure,

subplot(131),imshow(uint8(I)),title('原图');

subplot(132),imshow(uint8(J)),title('提亮图');

subplot(133),imshow(uint8(K)),title('变暗图');

（2）

I = imread('pout.tif');

figure,subplot(221),imshow(I),title('原图')

subplot(222),imhist(I),title('原图的直方图')

Ieq = histeq(I, 256);

subplot(223),imshow(Ieq),title('均衡化后图')

subplot(224),imhist(Ieq),title('均衡化后的直方图')

（3）

b=imread('pout.tif');

f0=0; g0=0;

f1=60; g1=20;

f2=180; g2=220;

f3=255; g3=255;

figure,plot([f0,f1,f2,f3],[g0,g1,g2,g3]);

r1=(g1-g0)/(f1-f0); %第一段扩展/压缩系数

r2=(g2-g1)/(f2-f1); %第二段扩展/压缩系数

r3=(g3-g2)/(f3-f2); %第三段扩展/压缩系数

h=double(b);

g=(r1\*(h-f0)+g0).\*((h>=f0).\*(h<=f1))+(r2\*(h-f1)+g1).\*((h>=f1).\*(h<=f2))+(r3\*(h-f2)+g2).\*((h>=f2).\*(h<=f3));

figure,imshow(b);title('原图')

figure,imshow(uint8(g));title('处理后的图')

（4）

I=imread('tower.jpg'); I=rgb2gray(I); I=im2double(I);

J=imnoise(I,'gaussian',0,0.02);

a=fspecial('average',3);

x=imfilter(J,a,'symmetric');

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

figure(1),

subplot(1,4,1), imshow(I),title('原图')

subplot(1,4,2), imshow(J), title('高斯加噪图')

subplot(1,4,3),imshow(x), title('3\*3滤波模板均值滤波图')

subplot(1,4,4), imshow(y), title('3\*3滤波模板中值滤波图')

K=imnoise(I,'salt & pepper');

b=fspecial('average',5);

z=imfilter(J,b,'symmetric');

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

figure(2)

subplot(1,4,1), imshow(I), title('原图')

subplot(1,4,2), imshow(J), title('高斯加噪图')

subplot(1,4,3), imshow(z), title('5\*5滤波模板均值滤波图')

subplot(1,4,4), imshow(p), title('5\*5滤波模板中值滤波图')

q=imfilter(K,a,'symmetric');

w=medfilt2(K,[3 3]);

figure(3)

subplot(1,4,1), imshow(I), title('原图')

subplot(1,4,2), imshow(K), title('椒盐加噪图')

subplot(1,4,3), imshow(q), title('3\*3滤波模板均值滤波图')

subplot(1,4,4),imshow(w), title('3\*3滤波模板中值滤波图')

t=imfilter(K,b,'symmetric');

l=medfilt2(K,[5 5]);

figure(4)

subplot(1,4,1),imshow(I), title('原图')

subplot(1,4,2), imshow(K), title('椒盐加噪图')

subplot(1,4,3), imshow(t), title('5\*5滤波模板均值滤波图')

subplot(1,4,4), imshow(l), title('5\*5滤波模板中值滤波图')

(5)

I=imread('rice.png'); I=im2double(I);

Q=imread('raw\_egg1.png'); Q=rgb2gray(Q);

h=[0,1,0;1,-4,1;0,1,0];

J=conv2(I,h,'same');

K=conv2(Q,h,'same');

figure

subplot(221),imshow(I),title('rice原图')

subplot(222),imshow(mat2gray(J)),title('rice锐化图')

subplot(223),imshow(Q),title('egg原图')

subplot(224),imshow(mat2gray(K)),title('egg锐化图')

（6）

I=imread('rice.png');

J=imread('raw\_egg1.png');

roberts=[-1,0;0,1];

sobel=[-1,0,1;-2,0,2;-1,0,1];

Prewitt=[-1,0,1;-1,0,1;-1,0,1];

roberts1=[0,-1;1,0];

sobel1=[-1,-2,-1;0,0,0;1,2,1];

Prewitt1=[-1,-1,-1;0,0,0;1,1,1];

a=abs(imfilter(I,roberts))+abs(imfilter(I,roberts1));

b=abs(imfilter(I,sobel))+abs(imfilter(I,sobel1));

c=abs(imfilter(I,Prewitt))+abs(imfilter(I,Prewitt1));

figure(1)

subplot(141),imshow(I),title('rice原图')

subplot(142),imshow(a),title('rice-roberts锐化图')

subplot(143),imshow(b),title('rice-sobel锐化图')

subplot(144),imshow(c),title('rice-prewitt锐化图')

d=abs(imfilter(J,roberts))+abs(imfilter(J,roberts1));

e=abs(imfilter(J,sobel))+abs(imfilter(J,sobel1));

f=abs(imfilter(J,Prewitt))+abs(imfilter(J,Prewitt1));

figure(2)

subplot(141),imshow(J),title('egg原图')

subplot(142),imshow(d),title('egg-roberts锐化图')

subplot(143),imshow(e),title('egg-sobel锐化图')

subplot(144),imshow(f),title('egg-prewitt锐化图')

实验4 图像增强-频域

（1）

I=imread('coins.png');

If=fft2(I);

a=fftshift(If);

b=log(abs(a));

out=ifftshift(a);

out=ifft2(out);

out=out/max(out(:));

subplot(1,3,1),imshow(I),title('原图');

subplot(1,3,2),imshow(b,[8,10]),title('移动到图像中心的频谱图');

subplot(1,3,3),imshow(out),title('逆变换回空域的图像');

（2）

I=imread('coins.png');

figure(1);

subplot(1,3,1);imshow(I);title('原图');

I2=im2double(I);

[M,N]=size(I2);

u=-M:M;

v=-N:N;

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

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

D0=30;

H=exp(-D.^2/(2\*D0.^2));

J=fftshift(fft2(I2,size(H,1),size(H,2)));

K=J.\*H;

L=ifft2(ifftshift(K));

L1=L(1:M,1:N);

subplot(1,3,2);mesh(U,V,H);title('高斯低通滤波器三维图');

subplot(1,3,3);imshow(L1);title('高斯滤波后硬币图');

（3）

I=imread('coins.png');

I=im2double(I);

M=2\*size(I,1);

N=2\*size(I,2);

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));

J=fftshift(fft2(I,size(H,1),size(H,2)));

K=J.\*H;

L=ifft2(ifftshift(K));

L=L(1:size(I,1),1:size(I,2));

figure,

subplot(221),imshow(I),title('原图');

subplot(223),mesh(H),title('巴特沃斯低通滤波器三维图');

subplot(222),imshow(L),title('巴特沃斯滤波后硬币图（空域）');

（4）

im=(imread('coins.png'));

I=im2double(im);

[M,N]=size(I);

u=-M:M;

v=-N:N;

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

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

D0=20;

n=1;

H=1./(1+D0./D).^(2\*n);

J=fftshift(fft2(I,size(H,1),size(H,2)));

K=J.\*H;

L=ifft2(ifftshift(K));

L1=L(1:M,1:N);

figure,

subplot(2,2,1),imshow(im),title('原图');

subplot(2,2,2),mesh(H),title('巴特沃斯高通滤波器三维图');

subplot(2,2,3),imshow(mat2gray(L1)),title('高通滤波后硬币图');

subplot(2,2,4),imshow(mat2gray(I+L1)),title('高频增强图');

实验5

（1）

a=imread('copper.png');

a=rgb2gray(a);

IF=fftshift(fft2(a)); %傅里叶变换后将低频移到中心

IFV=log(1+abs(IF)); %为了显示，进行频谱范围压缩

imwrite(mat2gray(IFV), 'copper\_feq.png');

%一定要重新存为图像，观察频谱中亮点离中心点的距离，以判断带阻滤波器的中心频率

freq=8; %通过观察确定中心频率为8（频谱亮点距离中心8个像素距离）

width=3; %带阻滤波器宽度

[p,q]=size(a); %生成带阻滤波器

ff = ones(p,q);

for i=1:p

for j=1:q

ff(i,j)=1-exp(-0.5\*((((i-p/2)^2+(j-q/2)^2)-freq^2)/(sqrt((i-p/2)^2+(j-q/2)^2)\*width))^2);

end

end

out = IF.\* ff; %频域带阻滤波

out = ifftshift(out); %低频移回四个角

out = ifft2(out); %逆傅里叶变换

out = abs(out); %取模

out = out/max(out(:)); %归一化

figure,

subplot(221),imshow(a),title('原图');

subplot(222),imshow(IFV,[]),title('原图频谱');

subplot(223),mesh(ff),title('带阻滤波器三维图');

subplot(224),imshow(out,[]),title('去噪后图');

（2）

RGB=imread('saturn.png');

a=rgb2gray(RGB);

b=imnoise(a,'gaussian',0,0.025);

c=imfilter(b,fspecial('average',3));

d=wiener2(b,[5 5]);

figure,

subplot(221); imshow(a); title('原图');

subplot(222); imshow(b); title('噪声图');

subplot(223); imshow(c); title('均值滤波图');

subplot(224); imshow(d); title('维纳滤波图');

（3）选做

（4）

a=im2double(imread('cameraman.tif'));

LEN=21; THETA=45;

b=fspecial('motion',LEN,THETA);

c=imfilter(a,b,'conv','circular');

NSR1=0; L1=deconvwnr(c,b,NSR1);

NSR2=0; L2=deconvwnr(c,b,NSR2);

figure,

subplot(221);imshow(a);title('原图');

subplot(222);imshow(c);title('运动模糊图');

subplot(223);imshow(L1);title('逆滤波');

subplot(224);imshow(L2);title('维纳滤波');

（5）

a=im2double(imread('cameraman.tif'));

LEN=21;

THETA=45;

PSF=fspecial('motion',LEN,THETA);

b=imfilter(a,PSF,'conv','circular');

noise\_mean=0;

noise\_var=0.0001;

c=imnoise(b,'gaussian',noise\_mean,noise\_var); %运动+噪声退化图像

NSR2=noise\_var/var(a(:));

p=deconvwnr(c,PSF, 0); %inverse filter

q=deconvwnr(c,PSF,NSR2); %wiener filter

figure,

subplot(221),imshow(a),title('原图');

subplot(222),imshow(b),title('运动模糊噪声图');

subplot(223),imshow(p),title('逆滤波结果');

subplot(224),imshow(q),title('维纳滤波结果');

（6）

逆滤波对无噪声的图像具有较好的恢复能力，但存在噪声时，滤波效果很差。维纳滤波对噪声有一定的抑制能力，但是需要知道图像的噪信比。

实验6 图像编码

（1）

p=[0.4,0.18,0.1,0.1,0.07,0.06,0.05,0.04];

entropy=sum(-(log(p+0.00001)).\*p);

xi=1:8;

[dict, avglen]=huffmandict(xi,p);

eta=entropy/avglen; %编码效率

（2）

I=imread('cameraman.tif');

I1=I(:);

[M,N]=size(I);

p=imhist(I)/(numel(I));

entropy=sum(-(log(p+0.00001)).\*p);

xi=0:255;

[dict, avglen]=huffmandict(xi,p);

enco=huffmanenco(I1,dict);

avglen2=numel(enco)/(numel(I));

eta=entropy/avglen; %编码效率

deco=huffmandeco(enco,dict); Ideco= reshape(deco,[M,N]);

figure,

subplot(1,2,1);imshow(I);title('原图');

subplot(1,2,2);imshow(uint8(Ideco));title('解码后图像');

（3）

i=imread('boy.png');

j=rgb2gray(i);

[M,N]=size(j);

dcti1=dct2(j);

mask=(abs(dcti1)<30);

dcti1(mask)=0;

im=mat2gray(idct2(dcti1));

psnr=PSNR(j,im); %信噪比

ratio=sum(~mask(:))/(M\*N); **%总系数的百分比（非零系数个数/图像总像素数量）**

figure,

subplot(1,3,1);imshow(j);title('原图');

subplot(1,3,2);imshow(log(abs(dcti1)),[]);title('D0=20,DCT变换图')

subplot(1,3,3);imshow(im);title('DCT逆变换图')

function [output]=PSNR(i,im) % i: 原图； im:退化图

i=double(i);

im=double(im);

[M,N]=size(i);

i=255\*(i-min(i(:)))./(max(i(:))-min(i(:))); % i数据归一化，double（0-255）

im=255\*(im-min(im(:)))./(max(im(:))-min(im(:)));% im数据归一化， double（0-255）

MSE=sum(sum((i-im).^2))/(M\*N);

X1=max(max(i));

output=20\*log10(X1/(MSE^(1/2)));

end

D0=20时，DCT逆变换的峰值信噪比为： 30.8 , 总系数的百分比（非零系数个数/图像总像素数量）：0.0757

D0=30时，DCT逆变换的峰值信噪比为： 27.8 , 总系数的百分比（非零系数个数/图像总像素数量）：0.0474

实验7 图像分割

（1）

clear

clc

I=imread('rice.png');

I=im2double(I);

[J1, thresh1]=edge(I, 'roberts', 20/255);

[J2, thresh2]=edge(I, 'prewitt', 20/255);

[J3, thresh3]=edge(I, 'sobel', 20/255);

[J4, thresh4]=edge(I, 'log', 1/255);

[J5, thresh5]=edge(I, 'canny',[0.1 0.3]);

figure;

subplot(2,3,1); imshow(I);title('原图')

subplot(2,3,2); imshow(J1);title('roberts')

subplot(2,3,3); imshow(J2);title('prewitt')

subplot(2,3,4); imshow(J3);title('sobel')

subplot(2,3,5); imshow(J4);title('log')

subplot(2,3,6); imshow(J5);title('canny')

（2）

clear

clc

I=imread('rice.png');

I=im2double(I);

J=imnoise(I, 'gaussian', 0, 0.01);

[J1, thresh1]=edge(J, 'roberts', 20/255);

[J2, thresh2]=edge(J, 'prewitt', 20/255);

[J3, thresh3]=edge(J, 'sobel', 20/255);

[J4, thresh4]=edge(J, 'log', 1/255);

[J5, thresh5]=edge(J, 'canny',[0.1 0.3]);

figure;

subplot(2,3,1); imshow(I);title('原图')

subplot(2,3,2); imshow(J1);title('roberts')

subplot(2,3,3); imshow(J2);title('prewitt')

subplot(2,3,4); imshow(J3);title('sobel')

subplot(2,3,5); imshow(J4);title('log')

subplot(2,3,6); imshow(J5);title('canny')

（3）

roberts算子：边缘定位精度较高，但容易丢失部分边缘，不具备抑制噪声的能力。对具有陡峭边缘且含噪声少的图像效果较好。

Prewitt算子和Sobel 算子：对图像先进行加权平滑操作（平滑部分的权值有差异），再进行微分；对噪声都有一定的抑制能力，但检测出的边缘容易出现多像素宽度。

Laplacian算子：对图像中存在的阶跃型边缘点定位准确。对噪声非常敏感，造成不连续的检测边缘。

Log算子：抗噪能力较强，但可能将原有的比较尖锐的边缘平滑掉。高斯函数中方差越大，尖锐边缘会被平滑掉；反之方差越小，可以检测出图像更高频率的细节，但对噪声的抑制能力相对下降。

Canny算子：同样采用高斯函数对图像进行平滑，具有较强的噪声抑制能力。优点在于，使用两种不同的阈值分别检测强边缘和弱边缘，并且当弱边缘和强边缘相连时，才将弱边缘包含在输出图像中。

（4）

clear

clc

I=imread('car.png');

I=rgb2gray(I);

figure(1);

subplot(1,3,1); imshow(I);title('原图')

subplot(1,3,2); imhist(I);title('直方图')

J=I<120;

subplot(1,3,3); imshow(J);title('全局阈值分割图')

（5）

clear

clc

I=imread('car.png');

I=im2double(I);

I=rgb2gray(I);

T=graythresh(I);

%J=im2bw(I, T);

J=I<T;

figure;

subplot(1,3,1); imshow(I);title('原图')

subplot(1,3,2); imhist(I);title('直方图')

subplot(1,3,3); imshow(J);title('二值化图')