冒泡排序函数文件bubbleSort.m：

function [f, g] = bubbleSort( n, varargin )

% 参数2~3个

% 参数2为要排序的数组

% 参数3伴随参数2的排序而排序

error(nargchk(2,3,nargin));

f = varargin{1};

if nargin==3

g= varargin{2};

end

for i=1:n-1

for j=2:n-i+1

if f(j-1)>f(j)

t=f(j);

f(j)=f(j-1);

f(j-1)=t;

if nargin==3

t=g(j);

g(j)=g(j-1);

g(j-1)=t;

end

end

end

end

end

主过滤器文件myFilter.m：

function g = myFilter( f, moduleLen, varargin )

% 参数3~4个

% 参数3为处理的函数句柄

% 参数4为参数3的附加参数

handle=varargin{1};

error(nargchk(3,4,nargin));

[m,n] = size(f);

g=f;

if m>=moduleLen && n>=moduleLen

t = floor(moduleLen/2);

for i=(t+1):(m-t)

for j=(t+1):(n-t)

%根据参数个数选择是否传入附带参数

if nargin==3

g(i,j)=handle(f, i-t, j-t, moduleLen);

else

g(i,j)=handle(f, i-t, j-t, moduleLen, varargin{2});

end

end

end

end

end

对每个模块求均值文件squareAverage.m：

function avg = squareAverage( f, left, top, len )

avg = 0;

for i=left:left+len-1

for j=top:top+len-1

avg = avg+double(f(i,j));

end

end

avg = avg/(len\*len);

end

对每个模块求中值文件squareMedian.m：

function median = squareMedian( f, left, top, len )

I = f(left:left+len-1,top:top+len-1);

% 二维转一维

I = I';

I = I(:);

I=I';

n = len\*len;

bbSort=@bubbleSort;

I=bbSort(n, I);

median = I((n+1)/2);

end

对每个模块求K近邻均值文件KNN.m：

function avg = KNN( f, left, top, len, k )

t=floor(len/2);

%取出模块转为一行

arr=f(left:left+len-1,top:top+len-1);

arr=arr(:);

arr=arr';

%计算差值绝对值

n=len\*len;

center=f(left+t, top+t);

diff=zeros(n);

for i=1:n

diff(i)=abs(arr(i)-center);

end

%按差值升序

bbSort=@bubbleSort;

[diff, arr]=bbSort(n, diff, arr);

sum=0;

% 第一个为原中心点center，跳过

for i=2:k+1

sum=sum+double(arr(i));

end

avg=sum/k;

end

对每个模块求西格玛均值文件mySigma.m：

function avg = mySigma( f, left, top, len, k )

t=floor(len/2);

%取出模块转为一行

arr=f(left:left+len-1,top:top+len-1);

arr=arr';

arr=arr(:);

%计算差值绝对值

n=len\*len;

center=f(left+t, top+t);

diff=zeros(n);

for i=1:n

diff(i)=abs(arr(i)-center);

end

%按差值升序

bbSort=@bubbleSort;

[diff, arr]=bbSort(n, diff, arr);

sum=0;

count=0;

for i=1:n

if diff(i)<=k

sum=sum+double(arr(i));

count=count+1;

else

break;

end

end

avg=sum/count;

end

主文件noise.m：

rows=4;

cols=2;

I = imread('img/Fig10.20(a).tif');

% 缩小I节省测试时间

% [m,n]=size(I);

% I = I(1:0.3m,1:0.3n);

f1 = imnoise(I, 'gaussian');

f2 = imnoise(I, 'salt & pepper');

figure();

subplot(rows,cols,1);

imshow(I, []);

title('原图');

subplot(rows,cols,3);

imshow(f1, []);

title('高斯噪声图');

subplot(rows,cols,4);

imshow(f2, []);

title('椒盐噪声图');

handle1=@squareAverage;

g1 = myFilter(f1, 3, handle1);

subplot(rows,cols,5);

imshow(g1, []);

title('高斯噪声图的均值滤波');

handle2=@squareMedian;

g2 = myFilter(f2, 5, handle2);

subplot(rows,cols,6);

imshow(g2, []);

title('椒盐噪声图的中值滤波');

handle3=@KNN;

k=5;

g3 = myFilter(f1, 3, handle3, k);

subplot(rows,cols,7);

imshow(g3, []);

title('高斯噪声图的K近邻均值滤波');

handle4=@mySigma;

k=50;

g4 = myFilter(f1, 5, handle4, k);

subplot(rows,cols,8);

imshow(g4, []);

title('高斯噪声图的西格玛均值滤波');

测试：

命令窗口输入noise:

