

# 数字图像处理-课堂作业5

计创18-连月菡-181002222-8

对课件中的图像数据做如下处理 手算--可以只做2, 6) 或者编程 :考虑一个图像):

1.扩展为9\*9, 复制外圈数据

2.加入10%的随机椒盐噪声 (5个噪声点)

3.用以下方法分别进行去噪声处理, 并分析哪种方法效果好。

10均值滤波; 20中值滤波; 30最大值滤波;

40最小值滤波; 50中点值滤波;

60自适应中值滤波, 最大5\*5。

*Original Image*

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

*Image f(x, y)*

1.

```
1 a=[54,52,57,55,56,52,51;50 49 51 50 52 53 58;51 204 52 52 0 57 60;48 50 51 49
53 59 63;49 51 52 55 58 64 67;50 54 57 60 63 67 70;51 55 59 62 65 69 72]
2 B = padarray(a,[1 1],'replicate','both')%复制外圈数据
```

B =

54	54	52	57	55	56	52	51	51
54	54	52	57	55	56	52	51	51
50	50	49	51	50	52	53	58	58
51	51	204	52	52	0	57	60	60
48	48	50	51	49	53	59	63	63
49	49	51	52	55	58	64	67	67
50	50	54	57	60	63	67	70	70
51	51	55	59	62	65	69	72	72
51	51	55	59	62	65	69	72	72

2.

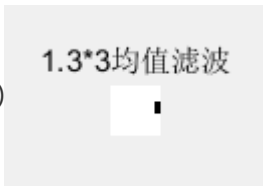


```
1 | c=imnoise(B,'salt & pepper',0.1);%添加10%的椒盐噪声
```

3.

(1)

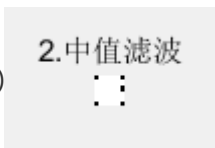
1.3\*3均值滤波



```
1 | d=imfilter(c,[3,3])%均值滤波
```

(2)

2.中值滤波



```
1 | d=medfilt2(c) %中值滤波
```

(3)

3.最大值滤波



```
1 | d=ordfilt2(c,9,ones(3,3))%最大值滤波
```

(4)

4.最小值滤波



```
1 | d=ordfilt2(c,1,ones(3,3))%最小值滤波
```

(5)

## 5.中点值滤波



```
1 | d=ordfilt2(c,4,ones(3,3))%中点值滤波
```

(6)

## 自适应中值滤波



```
1 | a=[54,52,57,55,56,52,51;50 49 51 50 52 53 58;51 204 52 52 0 57 60;48 50 51
2 | 49 53 59 63;49 51 52 55 58 64 67;50 54 57 60 63 67 70;51 55 59 62 65 69
3 | 72];
4 | B = padarray(a,[1 1],'replicate','both');%复制外圈数据
5 | img=imnoise(B,'salt & pepper',0.1);%添加10%的椒盐噪声
6 | [m ,n]=size(img);
7 |
8 | Nmax=5; %确定最大的滤波半径
9 |
10 | %下面是边界扩展，图像上下左右各增加Nmax像素。
11 | imgn=zeros(m+2*Nmax+1,n+2*Nmax+1);
12 | imgn(Nmax+1:m+Nmax,Nmax+1:n+Nmax)=img;
13 |
14 | imgn(1:Nmax,Nmax+1:n+Nmax)=img(1:Nmax,1:n); %扩展上边界
15 | imgn(1:m+Nmax,n+Nmax+1:n+2*Nmax+1)=imgn(1:m+Nmax,n:n+Nmax); %扩展右边界
16 | imgn(m+Nmax+1:m+2*Nmax+1,Nmax+1:n+2*Nmax+1)=imgn(m:m+Nmax,Nmax+1:n+2*Nmax+1); %扩展下边界
17 | imgn(1:m+2*Nmax+1,1:Nmax)=imgn(1:m+2*Nmax+1,Nmax+1:2*Nmax); %扩展左边界
18 | re=imgn;
19 | for i=Nmax+1:m+Nmax
20 |     for j=Nmax+1:n+Nmax
21 |         r=1; %初始滤波半径
22 |         while r<=Nmax
23 |             w=imgn(i-r:i+r,j-r:j+r);
24 |             w=sort(w);
25 |             Imin=min(w(:));
26 |             Imax=max(w(:));
27 |             Imed=w(uint8((2*r+1)^2/2));
28 |             if Imin<Imed && Imed<Imax %如果当前邻域中值不是噪声点，那么就用
29 |                 break;
30 |             else
31 |                 r=r+1; %否则扩大窗口，继续判断
32 |             end
33 |         end
```

```

34
35         if Imin<imgn(i,j) && imgn(i,j)<Imax           %如果当前这个像素不是噪声，
原值输出
36             re(i,j)=imgn(i,j);
37         else                                           %否则输出邻域中值
38             re(i,j)=Imed;
39         end
40
41     end
42 end
43
44 figure;
45 imshow(re(Nmax+1:m+Nmax,Nmax+1:n+Nmax),[]);

```

由上可知, 添加椒盐噪声前的原图和消除椒盐噪声后的结果进行对比,可以看出均值滤波效果最好,中值滤波,最大值滤波,自适应中值滤波效果一般,中点值滤波效果较差,最小值滤波效果最差。

应该使用其他多张较大图片进行多次比较。