

## 作业 2 报告

### 一、问题描述

将常见的滤波器实现一遍，理解不同滤波器的作用原理。并对图片进行加噪声处理，使用滤波器进行复原，观察并比较不同滤波器的效果。

### 二、实现步骤

#### 1. 算数均值滤波器

原理：算术均值滤波器的原理是在任意点 $(x, y)$ 处复原图像就是用  $S_{xy}$  定义区域的像素计算出来的算术均值。

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$$

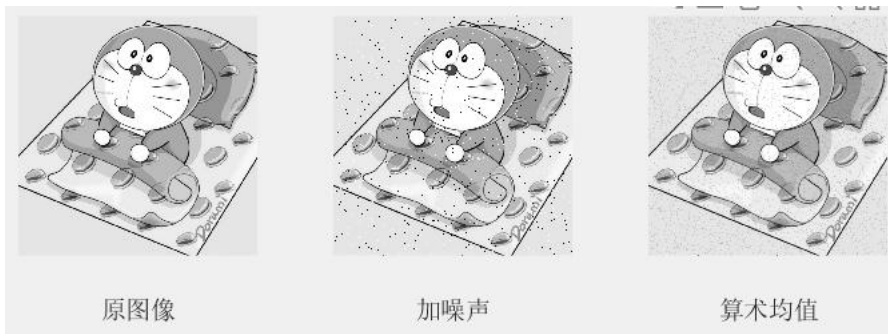
实现过程：选择定义的区域为  $3 \times 3$  的矩阵。

```
f= I_noisel;  
[w,h]=size(f);  
image1=I;  
image2= f;  
fsize=3;  
fssize=(fsize-1)/2;
```

```
figure();  
subplot(1,3,1);  
imshow(image1);  
xlabel('原图像');  
subplot(1,3,2);  
imshow(image2);  
xlabel('加噪声');  
resultImage = image2;
```

```
for x=1+fssize:1:w-fssize  
    for y=1+fssize:1:h-fssize  
        is=f(x-fssize:1:x+fssize,y-fssize:1:y+fssize);  
        resultImage(x,y)=sum(is(:))/numel(is);  
    end  
end  
  
subplot(1,3,3);  
imshow(resultImage);  
xlabel('算术均值');
```

实现效果：测试时首先对图片添加椒盐噪声，再与原图对比图像复原的效果。



#### 2. 几何均值滤波器

原理：每一个被复原像素由子图像窗口中像素点的乘积并自乘到  $1/mn$  次幂得到。

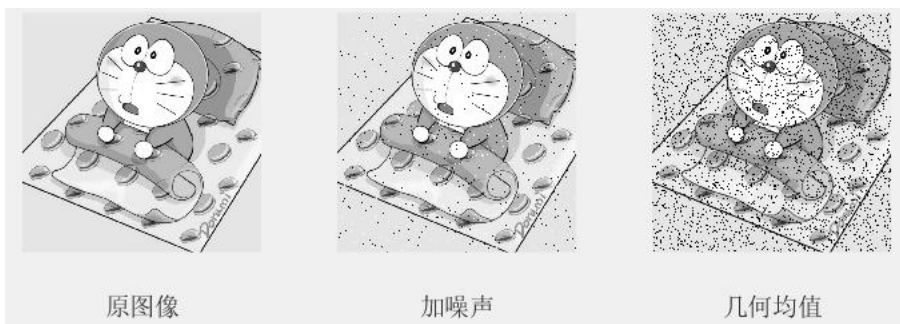
$$\hat{f}(x,y)=[\prod_{(s,t)\in S_{xy}}g(s,t)]^{\frac{1}{mn}}$$

实现过程：

```
figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');
resultImage = image2;

for x=1+fssize:l:w-fssize
    for y=1+fssize:l:w-fssize
        is=f(x-fssize:l:x+fssize,y-fssize:l:y+fssize);
        resultImage(x,y)=prod(prod(is(:)))^(1/numel(is));
    end
end
```

实现效果：



### 3. 谐波均值滤波器

原理：

$$\hat{f}(x,y)=\frac{mn}{\sum_{(s,t)\in S_{xy}}\frac{1}{g(s,t)}}$$

实现过程：

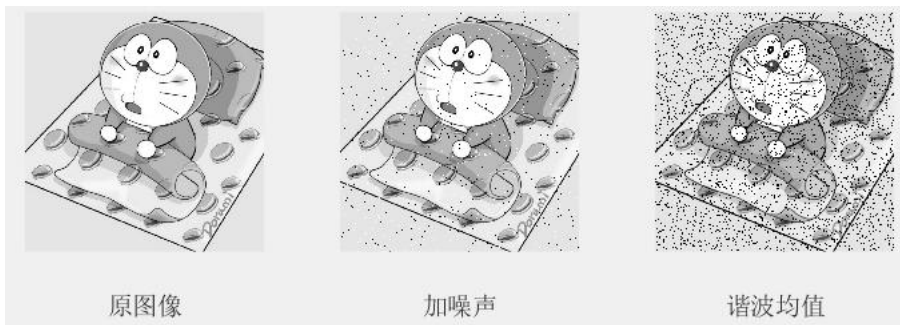
```

figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');
resultImage = image2;

for x=l+fssize:l:w-fssize
    for y=l+fssize:l:w-fssize
        is=f(x-fssize:l:x+fssize,y-fssize:l:y+fssize);
        is=1./is;
        resultImage(x,y)=numel(is)/sum(is(:));
    end
end
end

```

实现效果：



#### 4. 逆谐波均值滤波器

原理：

$$\hat{f}(x,y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^Q}$$

实现过程：

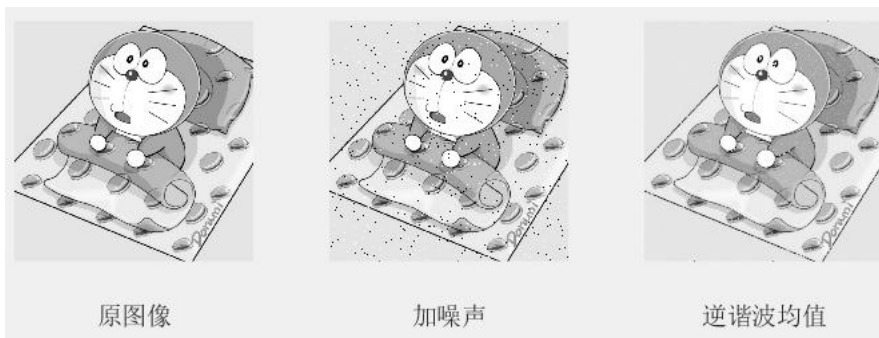
```

figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');
Q1 = 1.5;
resultImage = image2;

for x=l+fssize:l:w-fssize
    for y=l+fssize:l:w-fssize
        is=f(x-fssize:l:x+fssize,y-fssize:l:y+fssize);
        resultImage(x,y) = sum(is(:).^(Q1+1))/sum(is(:).^(Q1));
    end
end
end

```

实现效果：



## 5. 中值滤波器

原理：用该像素的相邻像素的灰度中值来替代该像素的值。

$$\hat{f}(x,y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s,t)\}$$

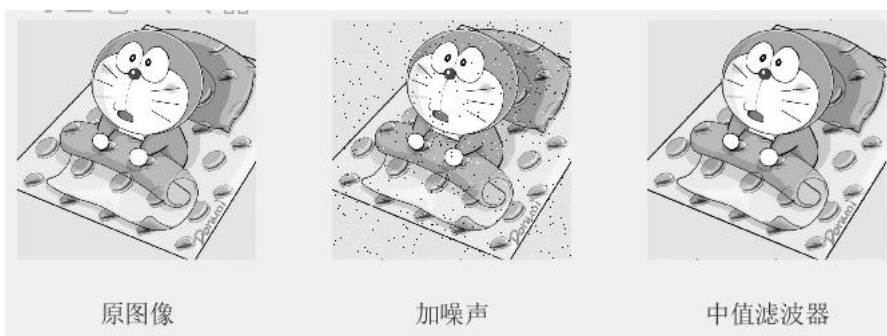
实现过程：

```
figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');

resultImage = image2;

for x=1+fssize:l:w-fssize
    for y=1+fssize:l:h-fssize
        is=f(x-fssize:l:x+fssize,y-fssize:l:y+fssize);
        temp = sort(is(:));
        resultImage(x,y)= temp((numel(temp)-1)/2);
    end
end
```

实现效果：



## 6. 最大值滤波器

原理：

$$\hat{f}(x,y) = \max_{(s,t) \in S_{xy}} \{g(s,t)\}$$

实现过程：

```
figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');

resultImage = image2;

for x=1+fssize:l:w-fssize
    for y=1+fssize:l:h-fssize
        is=f(x-fssize:l:x+fssize,y-fssize:l:y+fssize);
        temp = is(:);
        resultImage(x,y)= max(temp);
    end
end
```

实现效果：



## 7. 最小值滤波器

原理：

$$\hat{f}(x,y) = \min_{(s,t) \in S_{xy}} \{g(s,t)\}$$

实现过程：

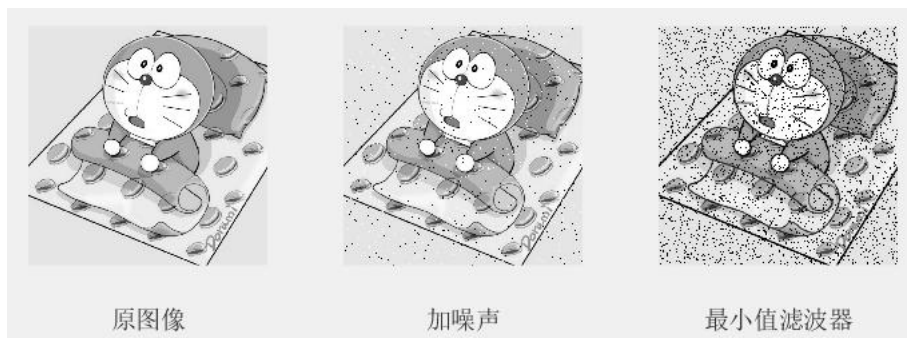
```

figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');
resultImage = image2;

for x=1+fssize:1:w-fssize
    for y=1+fssize:1:h-fssize
        is=f(x-fssize:1:x+fssize,y-fssize:1:y+fssize);
        temp = is(:);
        resultImage(x,y)= min(temp);
    end
end

```

实现效果：



## 8. 中点滤波器

原理：

$$\hat{f}(x,y) = \frac{1}{2} \left[ \max_{(s,t) \in S_{xy}} \{g(s,t)\} + \min_{(s,t) \in S_{xy}} \{g(s,t)\} \right]$$

实现过程：

```

figure();
subplot(1,3,1);
imshow(image1);
xlabel('原图像');
subplot(1,3,2);
imshow(image2);
xlabel('加噪声');

resultImage = image2;

for x=1+fssize:1:w-fssize
    for y=1+fssize:1:h-fssize
        is=f(x-fssize:1:x+fssize,y-fssize:1:y+fssize);
        temp = is(:);
        resultImage(x,y)= (max(temp) + min(temp))/2;
    end
end
end

```

实现效果：

