MATH 155 HWK 3

Shiqi Liang 305117507

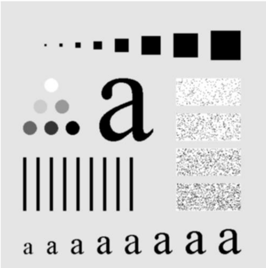
January 28, 2021

Question 1

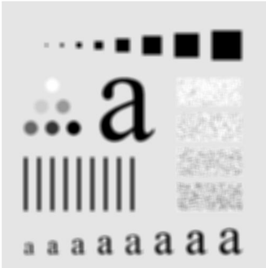
My matlab code is shown below:

```
function [after] = avgfilter(image,m)
a=(m-1)/2;
mm=m.^2;
output=uint8(zeros(size(image)));
%since we leave the borders unchanged, so we can leave out some ...
    borders
for i=(1+a):(size(image,1)-a)
    for j=(1+a):(size(image,2)-a)
        sum=0;
        for x=i-a:i+a
            for y=j-a:j+a
                tmp=image(x,y);
                tmp=int32(tmp);
                sum = sum + tmp;
                sum=int32(sum);
            end
        end
        output(i,j)=ceil(sum/mm);
    end
end

%now we replace the border values

for r=1:a
    for c=1:size(image,2)
    output(r,c)=image(r,c);
    end
end

for r=size(image,1)-a:size(image,1)
    for c=1:size(image,2)
        output(r,c)=image(r,c);
    end
end

for r=a+1:size(image,1)-a-1
    for c=size(image,2)-a:size(image,2)
        output(r,c)=image(r,c);
```

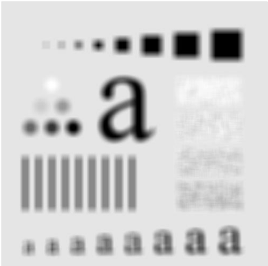
```
    end
end

for r=a+1:size(image,1)-a-1
    for c=1:a
        output(r,c)=image(r,c);
    end
end

after=uint8(output);
end
```



m=3



m=9



m=15

Question 2

To show it is a linear transformation, we need to show it is closed under addition and scalar multiplication. In other words, we just need to show that

$$H(cm+n) = cH(m) + H(n) \tag{1}$$

So we do tha by

$$H(cm+n) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)(cm+n)(x+s, y+t) \tag{2}$$

$$\Rightarrow \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)(cm)(x+s,y+t) + w(s,t)(n)(x+s,y+t) \tag{3}$$

$$\Rightarrow \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)(cm)(x+s,y+t) + \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)(n)(x+s,y+t) \tag{4}$$

$$\Rightarrow c \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)m(x+s,y+t) + \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)n(x+s,y+t) \tag{5}$$

$$\Rightarrow cH(m) + H(n) \tag{6}$$

The condition is satisfied, so it is indeed a linear transformation.

Question 3

A simple way to show this is to show that the operation is not closed under addition. Consider the image that is represented by the matrix A

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$\xi(A) = 0$. We also have a matrix B, which is defined as

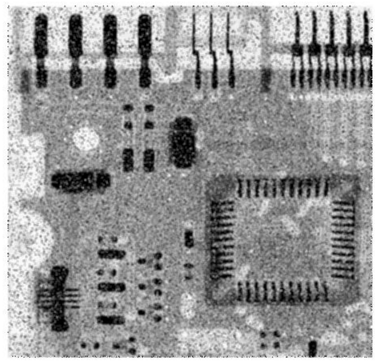
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$\xi(B) = 5$. However, The matrix A+B gives us

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 8 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

and subsequently $\xi(A+B) = 6$, and $\xi(A+B) \neq \xi(A) + \xi(B)$ This shows us that the operation is not linear.

Question 4



See code above in question 1.

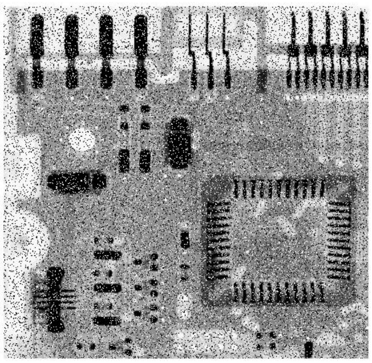
Question 5

The intuition for this question is to question is to first find all the values in the neighborhood, then sort out all the values in the order, find the median and substitute it in. By that logic, we have that:

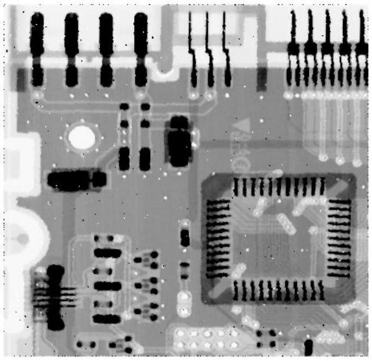
```
function [after] = medfilter(image)
output = zeros(size(image));
for i = 2:size(image,1) - 2
    for j = 2:size(image,2)-2
        list = [0 0 0 0 0 0 0 0 0];
        count = 1;


        for x = 1:3
            for y = 1:3
                list(count) = image(i + x - 2, j + y - 2);
                count = count + 1;
            end
        end



        output(i,j)=median(list);
    end
end

%substitute the og values back in the borders
    for c=1:size(image,2)
    output(1,c)=image(1,c);
    end

```

```
    for c=1:size(image,2)
        output(size(image,1),c)=image(size(image,1),c);
    end

    for r=1:size(image,1)
        output(r,1)=image(r,1);
    end

    for r=1:size(image,1)
        output(r,size(image,2))=image(r,size(image,2));
    end

after = uint8(output);

end
```

*note: since latex can not read tif images, I had to convert it into a jpg image first. The results are shown below:



Before



After

We can see that the image is a lot smoother and the salt and pepper noises aren't as obvious, but the median filter is obviously not as smooth as the average filter. Some of those noises are still very "jumpy".