Weifeng Jin

UID: 904429456
MATH 155 HW #4
Date: 02/09/2017

Problem 1

Suppose H is the median filter operator, f, g are two images. We want to show that for some (x, y) that $H(f(x,y) + g(x,y)) \neq H(f(x,y)) + H(g(x,y))$.

Suppose for some pixel (x, y) in f, we have surrounding pixels as the matrix, where the median is 4, so $H(f(x,y)) = 4$:

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}$$

Also, for the same pixel (x, y) in g, we have surrounding pixels as the matrix, where the median is 5, so $H(g(x,y)) = 5$, and $H(f(x,y)) + H(g(x,y)) = 9$

$$\begin{bmatrix} 4 & 4 & 4 \\ 5 & 4 & 7 \\ 6 & 7 & 8 \end{bmatrix}$$

By matrix addition, the corresponding pixels in matrix $f + g$ should be the matrix below, where the median is 8, so $H(f(x,y) + g(x,y)) = 8$

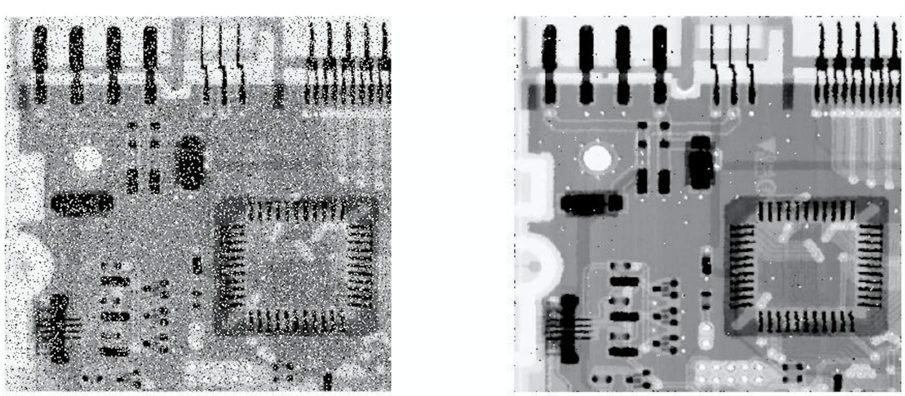
$$\begin{bmatrix} 4 & 4 & 4 \\ 7 & 8 & 12 \\ 12 & 14 & 16 \end{bmatrix}$$

Since $H(f(x,y) + g(x,y)) \neq H(f(x,y)) + H(g(x,y)) = 9$, we have proved that the median operator is not linear.

Problem 2

```
% the code for the median filter is shown below
% the method is explained in detail in the code comment
% the median filter creates a good-quality and clear denoising image
% for this picture, the median filter does a better job than the average
% filter, since the median filter has less blurring of the edges.
```

```
% read the original image
I=imread('Fig3.37(a).jpg');
% B is the output of the image
B=I;
% Since we want the boundary pixels to be unchanged, we adjust the size
% of our loop to be M/N-2 and start with (2,2)
for i= 1:size(I,1)-2
    for j=1:size(I,2)-2
        % create a mask for the median filter
        mask=zeros(9,1);
        % create an iterator for the mask
        inc=1;
        % we are using a 3*3 mask here
        for x=1:3
            for y=1:3
                % assin values in the sequence by the original image
                mask(inc)=I(i+x-1,j+y-1);
                inc=inc+1;
            end
        end
        % sort the mask sequence
        % median is the fifth element is a sorted list
        median=sort(mask);
        B(i,j)=median(5);
    end
end
B=uint8(B);
subplot(1,2,1)
imshow(uint8(I))
subplot(1,2,2)
imshow(uint8(B))
```



Problem 3

```
% the code for the composite Laplacian Mask is shown below
% the composite Laplacian mask makes the image sharper and clearer
% read the original image
I=imread('Fig3.40(a).jpg');
% mask for the composite Lapalacian mask
w=[0 -1 0; -1 5 -1;0 -1 0];
% if we use the builtin convolution function to create the output image
% img = conv2(double(I),w,'same');
[x,y]=size(I);
g=zeros(x+2,y+2);
%then, store f within g
for i=1:x
    for j=1:y
        g(i+1,j+1)=I(i,j);
    end
end
%traverse through the matrix, keep the boundaries unchanged
for i= 2:size(I,1)-2
    for j=2:size(I,2)-2
        img(i,j)=g(i,j)*w(1,1)+g(i+1,j)*w(2,1)+g(i+2,j)*w(3,1)
        + g(i,j+1)*w(1,2)+g(i+1,j+1)*w(2,2)+g(i+2,j+1)*w(3,2)
        + g(i,j+2)*w(1,3)+g(i+1,j+2)*w(2,3)+g(i+2,j+2)*w(3,3);
    end
end
subplot(1,2,1)
imshow(uint8(I))
subplot(1,2,2)
imshow(uint8(img))
```