Joshua Lai
804-449-134

1) The median, ξ, of a set of numbers is such that half the values in the set are less than or equal to ξ, and half are greater than or equal to ξ . For example, the median of the set of values {2, 3, 8, 20, 21, 25, 31} is 20. Show that an operator applied to the set of images (matrices) of the same dimension, that computes the median, is nonlinear.

   Let A be a matrix defined as

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

   and let B be a matrix defined as

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

   where both matrices live in the matrix vector space. Using the median operator, we can see that $\xi(A) = 0$ and $\xi(B) = 5$. Using the properties of matrix addition, we can see that

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

   where $\xi(A+B) = 6$. Thus, we can see that $\xi(A+B) \neq \xi(A)+\xi(B)$ and therefore, the median operator is nonlinear.

2) Write a computer program that will denoise an image using the 3×3 median filter. Apply your algorithm to the X-Ray image of circuit board corrupted by salt-and-pepper noise (Fig3.37(a).jpg). You should turn in the details of the method, your computer program, the input and output images. Perform your calculations only for interior pixels, not for boundary pixels. Explain your result and compare it with the output obtained in the previous homework on the same image (using a linear average filter).

   Please see attached pages for images. Code is shown on next page.

   In general, the image generated with the code below is significantly better than the image generated with code from the previous assignment (see attached images for reference). With this technique of using the median of a mask that had values generated from the original image (in comparison to hard-coding values), the image quality increased. Although there is still some residual salt-and-pepper noise in the image, it is a great improvement from the original image and from the linear average filtered image.

Joshua Lai
804-449-134

```
%Import image
A = imread('original_image_problem_2.jpg');

%Create seperate matrix to duplicate and show image
B = zeros(size(A));

%Loop through interior pixels of image
for i = 2:size(A,1) - 2
    for j = 2:size(A,2)-2

        %Create the filter with zeros and counter
        filter = [0 0 0 0 0 0 0 0 0];
        count = 1;

        %Loop through the filter
        for x = 1:3
            for y = 1:3
                %Replace values of filter with those of the image
                filter(count) = A(i + x - 1, j + y - 1);
                count = count + 1;
            end
        end

       %Sort array to find median value
        for a = 1: 50
            for b = 1:8
                if filter(b) > filter(b + 1)
                        %Swap values
                        temp = filter(b);
                        filter(b) = filter(b + 1);
                        filter(b + 1) = temp;
                end
            end
        end

        %Grab median (index 5)
        B(i,j)=filter(5);
    end
end

%Convert image to uint8 and then display
C = uint8(B);
imshow(C);
```

3) (Composite Laplacian Mask). Write a computer program that implements the operation $g(x, y) = f(x, y) - \nabla^2 f(x, y)$ in the form of a spatial linear filter with a 3×3 mask. Give the form of the mask and apply the program to the image of the North Pole of the moon (Fig3.40(a).jpg). You should turn in the details of the method, the computer program, the input and output images. Perform your calculations only for interior pixels, not for boundary pixels. Explain your result.

Please see attached pages for images. Code is shown on next page.

After applying the Laplacian filter to the image, we can see that the image does become sharper, in terms that we can see more details. The Laplacian filter is designed to detect edges better than other filters, which is why we can see more details and distinction in the image. Although it might be hard to see it on paper (since I only have a black-and-white printer), there is an improvement in the image.

Joshua Lai
804-449-134

```
%Import image
A = im2double(imread('original_image_problem_3.jpg'));

%Hard-code Laplacian filter
laplacian_filter = [-1 -1 -1 ; -1 9 -1 ; -1 -1 -1];

%Create dummy matrix to hold output image
B = zeros(size(A));

%Loop through image
for i = 2:size(A,1)-1
    for j = 2:size(A,2)-1

        %Apply filter to pixels
        temp = A(i-1:i+1,j-1:j+1) .* laplacian_filter;

        %Store filtered pixels in new matrix
        B(i,j) = sum(temp(:));
    end
end

imshow(B);
```

Optional Problem

4) Recall that the finite differences formula $\frac{f(x+h)-f(x-h)}{2h}$ is a second order approximation of the first-order derivative $f'(x)$.

a) Using $h = 1$, apply this formula to approximate the gradient map

$$g(x, y) = |\nabla f|^2(x, y) = \left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2.$$

b) Download Fig5.26a and plot an image negative of its gradient map g (edges will appear black, while homogenous regions will appear white; rescaling may be necessary). Explain the steps taken. Ignore the pixels on the boundary of the image for simplicity, when computing the discrete gradient.