Smoothing Filters

Project #3

Submitted by:
Name: Bidur Bhurtel
DAWG: 856121420

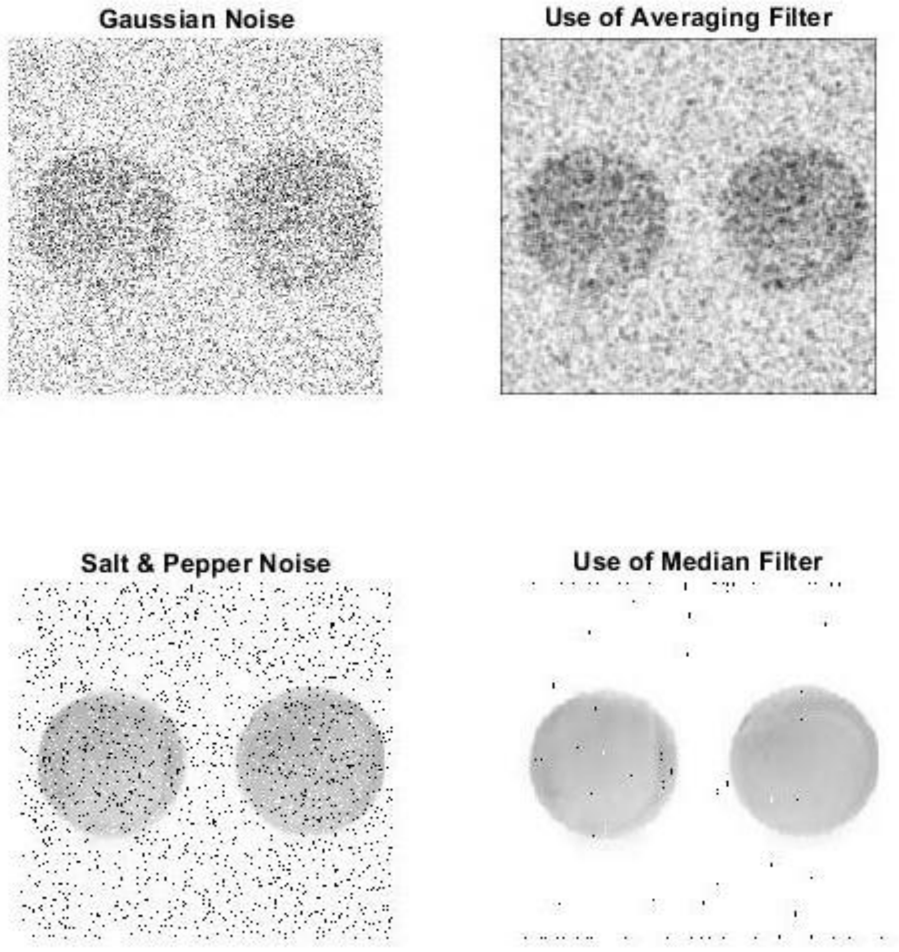
Submitted on November 04, 2019

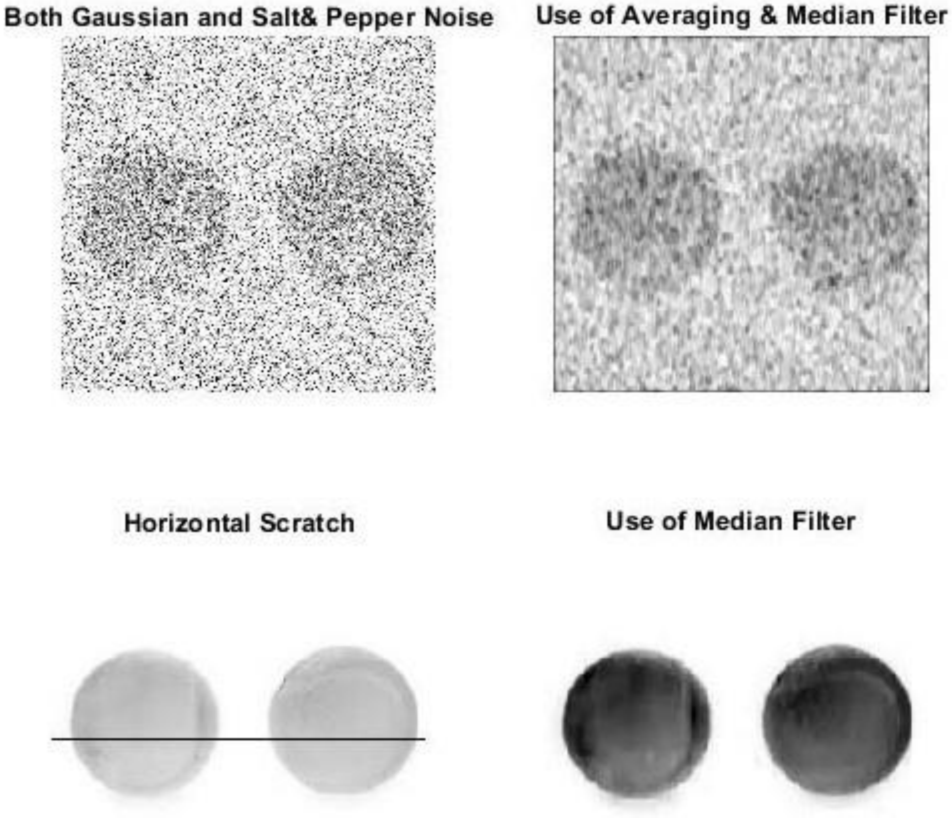
Objectives:

We are going to write a MATLAB program to design smoothing filters for different kind of noises and analyze the result by showing the images with the presence of noise and after use of the appropriate filters.

Original Image

Results




Both Gaussian and Salt& Pepper Noise
Use of Averaging & Median Filter
Horizontal Scratch
Use of Median Filter

Comments

We used box averaging filter of size 3X3 to smooth the gaussian noise in the image. The noise is spread out throughout the image. For the image with salt & pepper noise, we used median filter of size 5X5 and median filter of size 7X1 was used for the image with a horizontal scratch. The use of median filter almost removes all the salt & pepper noise. We could easily remove the remaining salt and pepper noises by increasing the filter size.

For the image with both gaussian and salt & pepper noise, first median filter of size 5X5 was used and then an averaging filter was applied to smooth the image.

MATLAB Code

```
image_RGB = rgb2gray(imread('two_balls.jpg'));

noisy_gauss = imnoise(image_RGB, 'gaussian', 0, 0.1);
noisy_salpep = imnoise(image_RGB, 'salt & pepper', 0.1);

noisy_gauss_salpep = imnoise(noisy_gauss, 'salt & pepper', 0.1);

%% for horizontal scratch
hor_scratch = image_RGB;
hor_scratch(112, :, :) = 0;

%% horizontal scratch filter
[row_size, col_size] = size(hor_scratch);
padded_hor_scratch = padarray(hor_scratch, [3,0]);
smoothed_hor_scratch = zeros(row_size, col_size);
for i = 4:(row_size+3)
    for j = 1:col_size
        filter_values = padded_hor_scratch(i-3:i+3, j);
        sorted_fil_val = sort(filter_values);
        smoothed_hor_scratch(i-3,j) = sorted_fil_val(4);
    end
end
hor_scratch_smoothed = mat2gray(smoothed_hor_scratch);

%% function call
noisy_gauss_smoothed = averaging_filter(noisy_gauss);
noisy_salpep_smoothed = median_filter(noisy_salpep);
noisy_salpep_gauss_smoothed =
median_filter(averaging_filter(noisy_gauss_salpep));
% hor_scratch_smoothed = median_filter(hor_scratch);

%% for displaying
% figure;
% subplot(4,2,1), imshow(hor_scratch);
% subplot(4,2,3), imshow(noisy_gauss);
% subplot(4,2,5), imshow(noisy_salpep);
% subplot(4,2,7), imshow(noisy_gauss_salpep);
% subplot(4,2,2), imshow(hor_scratch_smoothed);
% subplot(4,2,4), imshow(noisy_gauss_smoothed);
% subplot(4,2,6), imshow(noisy_salpep_smoothed);
% subplot(4,2,8), imshow(noisy_salpep_gauss_smoothed);

figure, imshow(image_RGB), title('Original Grayscale Image');
figure, subplot(1,2,1), imshow(noisy_gauss), title('Gaussian Noise');
subplot(1,2,2), imshow(noisy_gauss_smoothed), title('Use of Averaging
Filter');
figure, subplot(1,2,1), imshow(noisy_salpep), title('Salt & Pepper Noise');
subplot(1,2,2), imshow(noisy_salpep_smoothed), title('Use of Median Filter');
figure, subplot(1,2,1), imshow(noisy_gauss_salpep), title('Both Gaussian and
Salt& Pepper Noise')
subplot(1,2,2), imshow(noisy_salpep_gauss_smoothed),title('Use of Averaging &
Median Filter');
figure, subplot(1,2,1), imshow(hor_scratch), title('Horizontal Scratch');
subplot(1,2,2), imshow(hor_scratch_smoothed),title('Use of Median Filter');
```

```
%% for gaussian noise, using averaging filter of size 3X3
function[smoothed_img_gauss] = averaging_filter(noisy_gauss)
[row_size, col_size] = size(noisy_gauss);
% zero padding
noisy_gauss_padded = padarray(noisy_gauss, [1,1]);
smoothed_gauss = zeros(row_size, col_size);
for i = 2:(row_size+1)
    for j = 2:(col_size+1)
        smoothed_gauss(i-1,j-1) = sum(sum(noisy_gauss_padded(i-1:i+1, j-
1:j+1)))/9;
    end
end
smoothed_img_gauss = mat2gray(smoothed_gauss);
end

%% for salt and pepper noise, using median filter of size 5X5
function[smoothed_img_salpep] = median_filter(noisy_salpep)
[row_size, col_size] = size(noisy_salpep);
noisy_salpep_padded = padarray(noisy_salpep, [2,2]);
smoothed_salpep = zeros(row_size,col_size);
for i = 3:(row_size+2)
    for j = 3:(col_size+2)
        filter_values = noisy_salpep_padded(i-2:i+2, j-2:j+2);
        sorted_fil_val = sort(filter_values);
        smoothed_salpep(i-2,j-2) = sorted_fil_val(13);
    end
end
smoothed_img_salpep = mat2gray(smoothed_salpep);
end
```