Smoothing Filters

Project #3

**Submitted by:**

Name: Bidur Bhurtel

DAWG: 856121420

Submitted on November 04, 2019

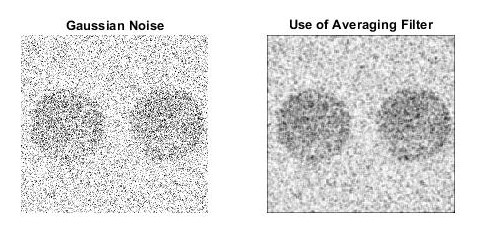
**Objectives:**

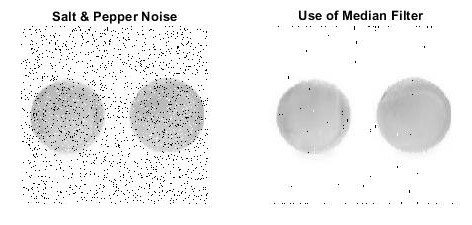
We are going to write a MATLAB program to design smoothing filters for different kind of noises and analyze the result by showing the images with the presence of noise and after use of the appropriate filters.

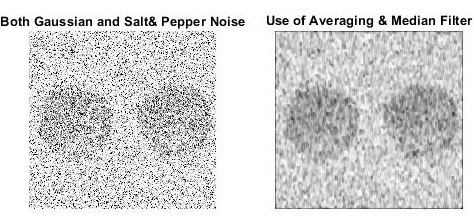
**Original Image**

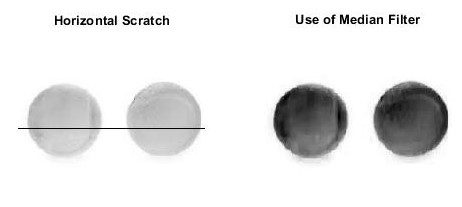
****

**Results**

****

****

****

****

**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