EE 5356 - DIGITAL IMAGE PROCESSING - PROJECT 4

NON-LINEAR FILTERING

By

PAVAI ARCHIMEDES

1001233996

Read any 256x256 or 512x512 grayscale image. Add the following types of noise to it to generate 4 noisy images:

1. Gaussian noise
2. Poisson noise
3. Salt & pepper noise
4. Speckle noise

Apply the following spatial filters to the noisy images:

1. Arithmetic mean
2. Geometric mean
3. Harmonic mean
4. Contra-harmonic mean
5. Median filter
6. Min
7. Max
8. Mid-point
9. Alpha trimmed mean filter

Submit the following with your code:

1. Print
   1. the original image,
   2. the noisy images, and
   3. the results of all the filters on each noisy image.
2. Determine which type of filtering worked well for each type of noise.

References:

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, III edition, Prentice Hall, pages 322-325, 2008.
2. Gonzalez, Woods and Eddins, “Digital Image Processing with MATLB”, I edition, Prentice Hall, pages 160-164, 2009.
3. Practical image and video processing using MATLAB by Marques, Oge

PROGRAM:

clc;

clear all;

close all;

Im\_img = imread('C:\Users\PAVAI ARCHIMEDES\Desktop\goldhill256.BMP');

figure(1);

imshow(Im\_img);

title('orig img');

dimension = size(Im\_img);

I = double(Im\_img);

Mi\_gauss = 0;

Vm\_gauss = 0.01;

salt\_pep = 0.05;

spk = 0.04;

ga\_noise = imnoise(Im\_img,'gaussian',Mi\_gauss,Vm\_gauss);

poi\_noise = imnoise(Im\_img,'poisson');

salt\_pepp\_noise = imnoise(Im\_img,'salt & pepper',salt\_pep);

spk\_noise = imnoise(Im\_img,'speckle',spk);

gau =double(ga\_noise);

poi = double(poi\_noise);

salt\_pepper = double(salt\_pepp\_noise);

spk = double(spk\_noise);

figure(2);

subplot(2,2,1);

imshow(uint8(gau));

title('gaussian noise');

subplot(2,2,2);

imshow(uint8(poi));

title('poison noise');

subplot(2,2,3);

imshow(uint8(salt\_pepper));

title('salt nd pepper noise');

subplot(2,2,4);

imshow(uint8(spk));

title('speckle noise');

ga\_arith = arithmetic\_filter(ga\_noise);

poi\_arith = arithmetic\_filter(poi\_noise);

salt\_arith = arithmetic\_filter(salt\_pepp\_noise);

spk\_arith = arithmetic\_filter(spk\_noise);

figure(3);

subplot(2,2,1);

imshow(uint8(ga\_arith));

title('gaussian noise arithmetic fil');

subplot(2,2,2);

imshow(uint8(poi\_arith));

title('poisson noise arithmetic fil');

subplot(2,2,3);

imshow(uint8(salt\_arith));

title('salt nd pepper noise arithmetic fil');

subplot(2,2,4);

imshow(uint8(spk\_arith));

title('speckle noise arithmetic fil');

ga\_geo = geometric\_filter(ga\_noise);

poi\_geo = geometric\_filter(poi\_noise);

salt\_geo = geometric\_filter(salt\_pepp\_noise);

spk\_geo = geometric\_filter(spk\_noise);

figure(4);

subplot(2,2,1);

imshow(uint8(ga\_geo));

title('gaussian noise geometric fil');

subplot(2,2,2);

imshow(uint8(poi\_geo));

title('poisson noise geometric fil');

subplot(2,2,3);

imshow(uint8(salt\_geo));

title('salt nd pepper noise geometric fil');

subplot(2,2,4);

imshow(uint8(spk\_geo));

title('speckle noise geometric fil');

ga\_har = harr\_filter(ga\_noise);

poi\_har = harr\_filter(poi\_noise);

salt\_har = harr\_filter(salt\_pepp\_noise);

spk\_har = harr\_filter(spk\_noise);

figure(5);

subplot(2,2,1);

imshow(uint8(ga\_har));

title('gaussian noise harmonic fil');

subplot(2,2,2);

imshow(uint8(poi\_har));

title('poisson noise harmonic fil');

subplot(2,2,3);

imshow(uint8(salt\_har));

title('salt and pepper noise harmonic fil');

subplot(2,2,4);

imshow(uint8(spk\_har));

title('speckle noise harmonic fil');

ga\_cont = cont\_filter(ga\_noise);

poi\_cont = cont\_filter(poi\_noise);

salt\_cont = cont\_filter(salt\_pepp\_noise);

spk\_cont = cont\_filter(spk\_noise);

figure(6);

subplot(2,2,1);

imshow(uint8(ga\_cont));

title('gaussian noise contra harmonic fil');

subplot(2,2,2);

imshow(uint8(poi\_cont));

title('poisson noise contraharmonic fil');

subplot(2,2,3);

imshow(uint8(salt\_cont));

title('salt nd pepper noise contraharmonic fil');

subplot(2,2,4);

imshow(uint8(spk\_cont));

title('speckle noise contraharmonic fil');

ga\_med = med(ga\_noise,[3 3]);

poi\_med = med(poi\_noise,[3 3]);

salt\_med = med(salt\_pepp\_noise,[3 3]);

spk\_med = med(spk\_noise,[3 3]);

figure(7);

subplot(2,2,1);

imshow(uint8(ga\_med));

title('gaussian noise median fil');

subplot(2,2,2);

imshow(uint8(poi\_med));

title('poisson noise median fil');

subplot(2,2,3);

imshow(uint8(salt\_med));

title('salt nd pepper noise median fil');

subplot(2,2,4);

imshow(uint8(spk\_med));

title('speckle noise median fil');

ga\_min = ordfilt2(ga\_noise,1,ones(3,3));

poi\_min = ordfilt2(poi\_noise,1,ones(3,3));

salt\_min = ordfilt2(salt\_pepp\_noise,1,ones(3,3));

spk\_min = ordfilt2(spk\_noise,1,ones(3,3));

figure(8);

subplot(2,2,1);

imshow(uint8(ga\_min));

title('gaussian noise min fil');

subplot(2,2,2);

imshow(uint8(poi\_min));

title('poisson noise min fil');

subplot(2,2,3);

imshow(uint8(salt\_min));

title('salt nd pepper noise min fil');

subplot(2,2,4);

imshow(uint8(spk\_min));

title('speckle noise min fil');

ga\_max = ordfilt2(ga\_noise,9,ones(3,3));

poi\_max = ordfilt2(poi\_noise,9,ones(3,3));

salt\_max = ordfilt2(salt\_pepp\_noise,9,ones(3,3));

spk\_max = ordfilt2(spk\_noise,9,ones(3,3));

figure(9);

subplot(2,2,1);

imshow(uint8(ga\_max));

title('gaussian noise max fil');

subplot(2,2,2);

imshow(uint8(poi\_max));

title('poisson noise max fil');

subplot(2,2,3);

imshow(uint8(salt\_max));

title('salt nd pepper noise max fil');

subplot(2,2,4);

imshow(uint8(spk\_max));

title('speckle noise max fil');

ga\_alp = alpha\_filter(ga\_noise);

poi\_alp = alpha\_filter(poi\_noise);

salt\_alp = alpha\_filter(salt\_pepp\_noise);

spk\_alp = alpha\_filter(spk\_noise);

figure(10);

subplot(2,2,1);

imshow(uint8(ga\_alp));

title('gaussian noise alphatrimd fil');

subplot(2,2,2);

imshow(uint8(poi\_alp));

title('poisson noise alphatrimd fil');

subplot(2,2,3);

imshow(uint8(salt\_alp));

title('salt nd pepper noise alphatrimd fil');

subplot(2,2,4);

imshow(uint8(spk\_alp));

title('speckle noise alphatrimd fil');

ga\_mid = midpt\_filter(ga\_noise);

poi\_mid = midpt\_filter(poi\_noise);

salt\_mid = midpt\_filter(salt\_pepp\_noise);

spk\_mid = midpt\_filter(spk\_noise);

figure(11);

subplot(2,2,1);

imshow(uint8(ga\_mid));

title('gaussian noise mid-pt fil');

subplot(2,2,2);

imshow(uint8(poi\_mid));

title('poisson noise mid-pt filter');

subplot(2,2,3);

imshow(uint8(salt\_mid));

title('salt nd pepper noise mid-pt fil');

subplot(2,2,4);

imshow(uint8(spk\_mid));

title('speckle noise mid-pt fil');

FUNCTIONS USED:

**midpt\_filter.m**

function B = midpt\_filter(noise\_image)

FUN = @(x) m\_mean(x(:));

B = nlfilter(double(noise\_image),[3 3],FUN);

end

**m\_mean.m**

function f = m\_mean(x)

dimension = size(x);

f = (1/2)\*(max(max(x))+min(min(x)))

end

**harr\_filter.m**

function B = harr\_filter(noise\_image)

FUN = @(x) h\_mean(x(:));

B = nlfilter(double(noise\_image),[3 3],FUN);

end

**h\_mean.m**

function f = h\_mean(x)

dimension = size(x);

sum =0;

for i = 1: dimension(1)

for j = 1:dimension(2)

sum = sum + 1/x(i,j);

end

end

f = (dimension(1)\*dimension(2))/sum;

end

**geometric\_filter.m**

function B = geometric\_filter(noise)

FUN = @(x) g\_mean(x(:));

B = nlfilter(double(noise),[3 3],FUN);

end

**g\_mean.m**

function f =g\_mean(x);

dimension = size(x);

prod =1;

for i = 1: dimension(1)

for j = 1:dimension(2)

prod = prod\*x(i,j);

end

end

f = (prod)^(1/(dimension(1)\*dimension(2)));

end

**cont\_filter.m**

function B = cont\_filter(noise\_image)

FUN = @(x) c\_mean(x(:));

B = nlfilter(double(noise\_image),[3 3],FUN);

end

**c\_mean.m**

function f = c\_mean(x)

dimension = size(x);

Q =1;

sum1 =0;

sum2 =0;

for i = 1: dimension(1)

for j = 1:dimension(2)

sum1 = sum1 + x(i,j)^(Q+1);

sum2 = sum2 + x(i,j)^(Q);

end

end

f = sum1/sum2;

end

**at\_mean.m**

function f = at\_mean(x)

dimension = size(x);

sum =0;

d = 0;

for i = 1: dimension(1)

for j = 1:dimension(2)

sum = sum + x(i,j);

end

end

f = sum\*(1/(dimension(1)\*dimension(2) - d));

end

**arithmetic\_filter.m**

function B = arithmetic\_filter(noise\_image)

FUN = @(x) a\_mean(x(:));

B = nlfilter(double(noise\_image),[3 3],FUN);

end

**alpha\_filter.m**

function B = alpha\_filter(noise\_image)

FUN = @(x) at\_mean(x(:));

B = nlfilter(double(noise\_image),[3 3],FUN);

end

**a\_mean.m**

function f = a\_mean(x)

dimension = size(x);

sum =0;

for i = 1: dimension(1)

for j = 1:dimension(2)

sum = sum + x(i,j);

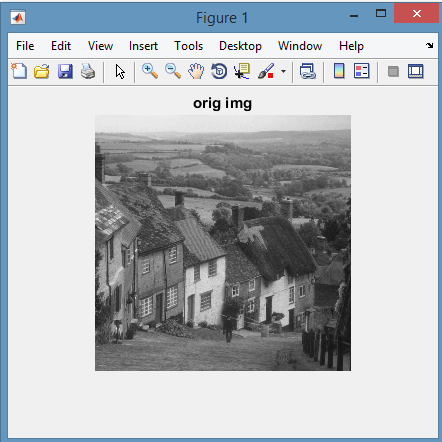
end

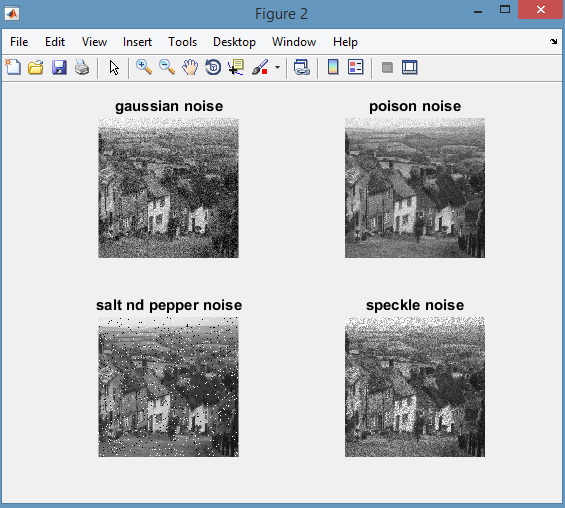
end

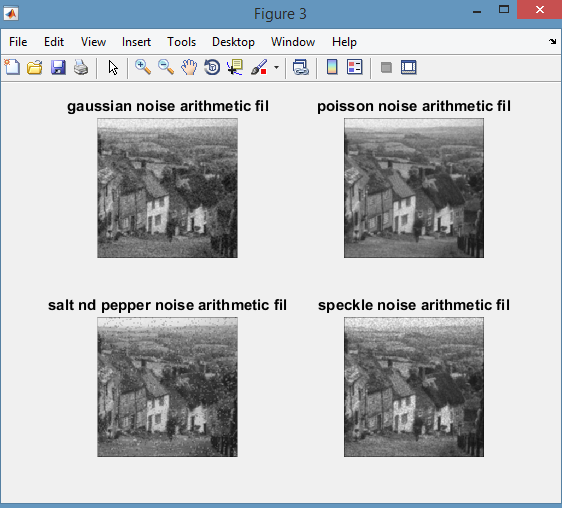
f = sum/(dimension(1)\*dimension(2));

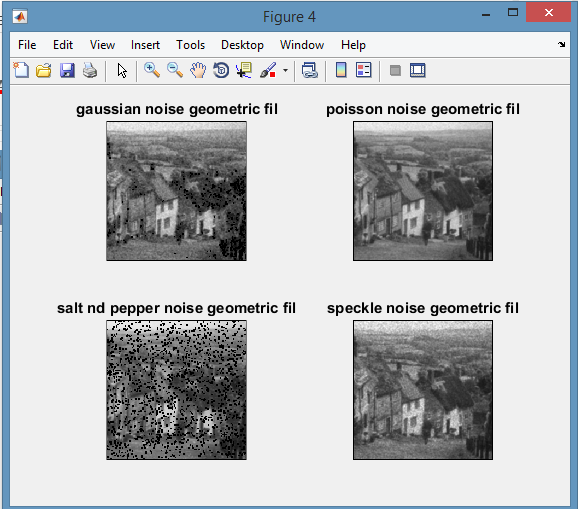
end

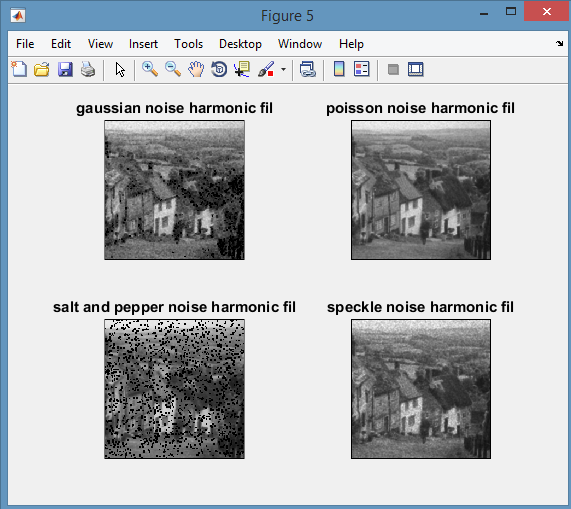
OUTPUT:

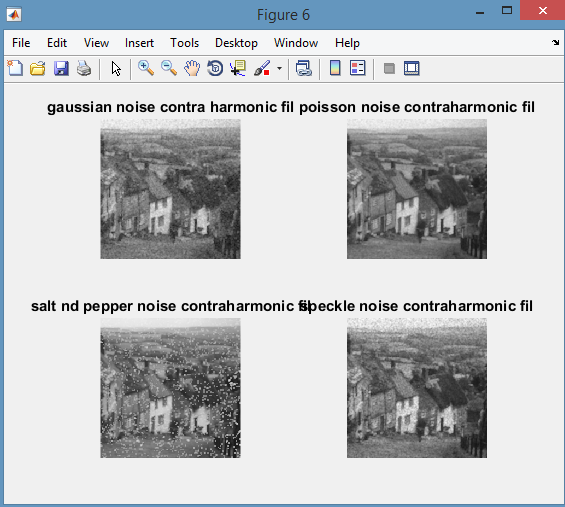


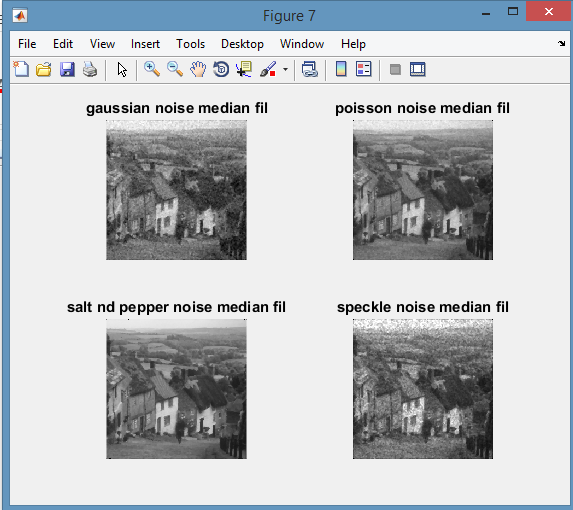


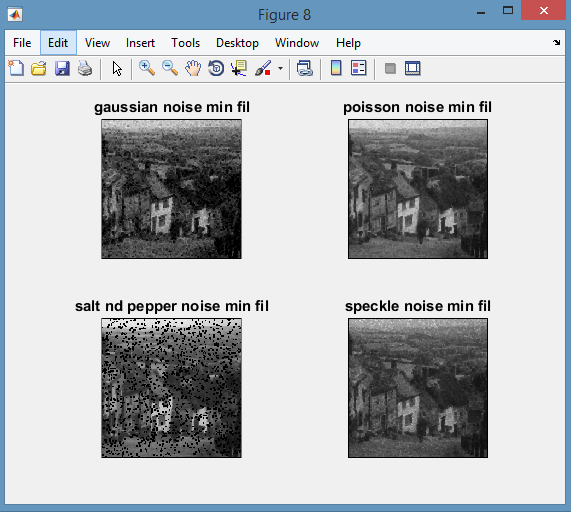


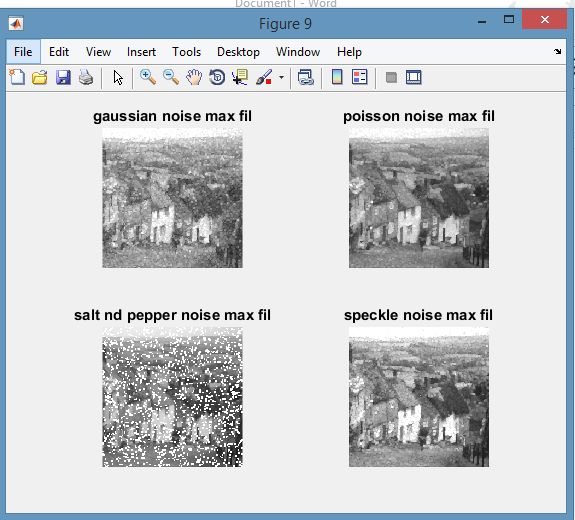


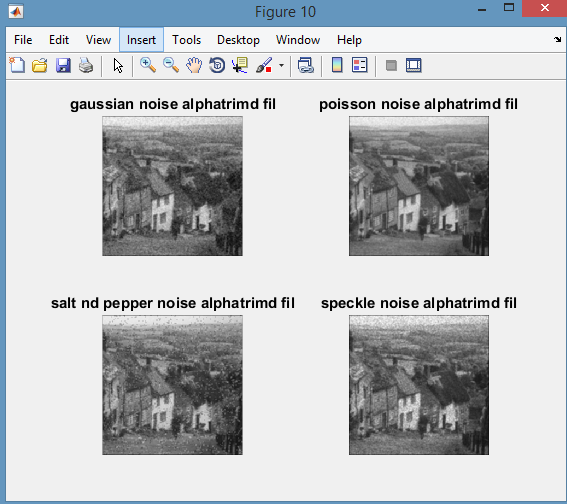


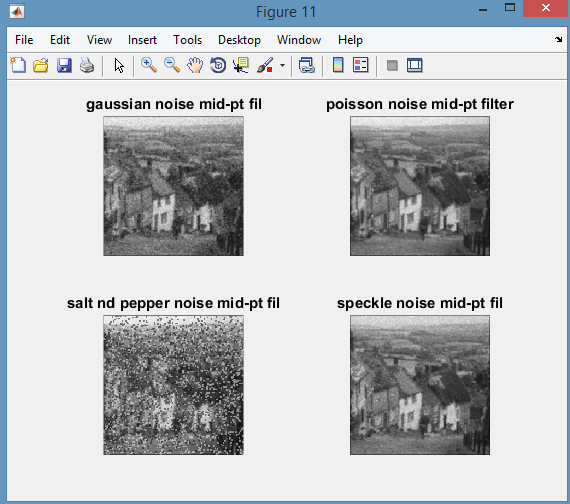












CONCLUSION:

The arithmetic mean filter works well on gaussian noise , geometric mean filter for speckle noise and poisson noise and median filter for salt and pepper noise.