EE 5356

Assignment 4

Vasudevan Murali

1000723391

4. Non Linear Filtering:

1. Program :

%%%%%%%%%%%%%%%%%%% assign4\_1.m %%%%%%%%%%%%%%%%%%%

clear all

close all

clc;

imageOrg = imread('goldhill256.bmp');

[row col] = size(imageOrg);

figure(1), imshow(imageOrg),title('Original Image');

% gaussian noise

mean = 0;

v = 0.01;

gaussianImage = imnoise(imageOrg,'gaussian',mean,v);

modGaussImage = zeros(row+2,col+2);

modGaussImage(2:1:row+1,2:1:col+1) = gaussianImage(:,:);

modGaussImage(:,1) = modGaussImage(:,2);

modGaussImage(:,col+2) = modGaussImage(:,col+1);

modGaussImage(1,:) = modGaussImage(2,:);

modGaussImage(row+2,:) = modGaussImage(row+1,:);

%poisson noise

poissonImage = imnoise(imageOrg,'poisson');

modPoissImage = zeros(row+2,col+2);

modPoissImage(2:1:row+1,2:1:col+1) = poissonImage(:,:);

modPoissImage(:,1) = modPoissImage(:,2);

modPoissImage(:,col+2) = modPoissImage(:,col+1);

modPoissImage(1,:) = modPoissImage(2,:);

modPoissImage(row+2,:) = modPoissImage(row+1,:);

% salt & pepper noise

D = 0.05;

spImage = imnoise(imageOrg,'salt & pepper',D);

modSpImage = zeros(row+2,col+2);

modSpImage(2:1:row+1,2:1:col+1) = spImage(:,:);

modSpImage(:,1) = modSpImage(:,2);

modSpImage(:,col+2) = modSpImage(:,col+1);

modSpImage(1,:) = modSpImage(2,:);

modSpImage(row+2,:) = modSpImage(row+1,:);

% speckle noise

V = 0.05;

speckleImage = imnoise(imageOrg,'speckle',V);

modSpeckleImage = zeros(row+2,col+2);

modSpeckleImage(2:1:row+1,2:1:col+1) = speckleImage(:,:);

modSpeckleImage(:,1) = modSpeckleImage(:,2);

modSpeckleImage(:,col+2) = modSpeckleImage(:,col+1);

modSpeckleImage(1,:) = modSpeckleImage(2,:);

modSpeckleImage(row+2,:) = modSpeckleImage(row+1,:);

figure(2),

subplot(2,2,1),imshow(gaussianImage), title('Gaussian noisy imshow');

subplot(2,2,2), imshow(poissonImage), title('Poisson noisy imshow');

subplot(2,2,3), imshow(spImage), title('Salt & pepper noisy imshow');

subplot(2,2,4), imshow(speckleImage),title('Speckle noisy imshow');

m = 3; % window row size

n = 3; % window column size

% arithmetic filter

fun = @(x) arithmetic(x(:));

meanImage = nlfilter(double(modGaussImage),[m n],fun);

figure(3),

subplot(2,2,1),imshow(uint8(meanImage(2:1:row+1,2:1:col+1)))

meanImage = nlfilter(double(modPoissImage),[m n],fun);

subplot(2,2,2),imshow(uint8(meanImage(2:1:row+1,2:1:col+1)))

meanImage = nlfilter(double(modSpImage),[m n],fun);

subplot(2,2,3),imshow(uint8(meanImage(2:1:row+1,2:1:col+1)))

meanImage = nlfilter(double(modSpeckleImage),[m n],fun);

subplot(2,2,4),imshow(uint8(meanImage(2:1:row+1,2:1:col+1)))

% geometric filter

fun1 = @(x) geometric(x(:));

geoImage = nlfilter(double(modGaussImage),[m n],fun1);

figure(4),

subplot(2,2,1),imshow(uint8(geoImage(2:1:row+1,2:1:col+1)))

geoImage = nlfilter(double(modPoissImage),[m n],fun1);

subplot(2,2,2),imshow(uint8(geoImage(2:1:row+1,2:1:col+1)))

geoImage = nlfilter(double(modSpImage),[m n],fun1);

subplot(2,2,3),imshow(uint8(geoImage(2:1:row+1,2:1:col+1)))

geoImage = nlfilter(double(modSpeckleImage),[m n],fun1);

subplot(2,2,4),imshow(uint8(geoImage(2:1:row+1,2:1:col+1)))

%harmonic filter

fun2 = @(x) harmonic(x(:));

harmoImage = nlfilter(double(modGaussImage),[m n],fun2);

figure(5),

subplot(2,2,1),imshow(uint8(harmoImage(2:1:row+1,2:1:col+1)))

harmoImage = nlfilter(double(modPoissImage),[m n],fun2);

subplot(2,2,2),imshow(uint8(harmoImage(2:1:row+1,2:1:col+1)))

harmoImage = nlfilter(double(modSpImage),[m n],fun2);

subplot(2,2,3),imshow(uint8(harmoImage(2:1:row+1,2:1:col+1)))

harmoImage = nlfilter(double(modSpeckleImage),[m n],fun2);

subplot(2,2,4),imshow(uint8(harmoImage(2:1:row+1,2:1:col+1)))

%contraharmonic filter

fun3 = @(x) contraharmonic(x(:));

contraImage = nlfilter(double(modGaussImage),[m n],fun3);

figure(6),

subplot(2,2,1),imshow(uint8(contraImage(2:1:row+1,2:1:col+1)))

contraImage = nlfilter(double(modPoissImage),[m n],fun3);

subplot(2,2,2),imshow(uint8(contraImage(2:1:row+1,2:1:col+1)))

contraImage = nlfilter(double(modSpImage),[m n],fun3);

subplot(2,2,3),imshow(uint8(contraImage(2:1:row+1,2:1:col+1)))

contraImage = nlfilter(double(modSpeckleImage),[m n],fun3);

subplot(2,2,4),imshow(uint8(contraImage(2:1:row+1,2:1:col+1)))

% Median filter

medImage = medfilt2(gaussianImage,[m n],'symmetric');

figure(7),

subplot(2,2,1),imshow(uint8(medImage))

medImage = medfilt2(poissonImage,[m n],'symmetric');

subplot(2,2,2),imshow(uint8(medImage))

medImage = medfilt2(spImage,[m n],'symmetric');

subplot(2,2,3),imshow(uint8(medImage))

medImage = medfilt2(speckleImage,[m n],'symmetric');

subplot(2,2,4),imshow(uint8(medImage))

% Max filter

maxImage = ordfilt2(gaussianImage,m\*n,ones(m,n),'symmetric');

figure(8),

subplot(2,2,1),imshow(uint8(maxImage))

maxImage = ordfilt2(poissonImage,m\*n,ones(m,n),'symmetric');

subplot(2,2,2),imshow(uint8(maxImage))

maxImage = ordfilt2(spImage,m\*n,ones(m,n),'symmetric');

subplot(2,2,3),imshow(uint8(maxImage))

maxImage = ordfilt2(speckleImage,m\*n,ones(m,n),'symmetric');

subplot(2,2,4),imshow(uint8(maxImage))

% Min filter

minImage = ordfilt2(gaussianImage,1,ones(m,n),'symmetric');

figure(9),

subplot(2,2,1),imshow(uint8(minImage))

minImage = ordfilt2(poissonImage,1,ones(m,n),'symmetric');

subplot(2,2,2),imshow(uint8(minImage))

minImage = ordfilt2(spImage,1,ones(m,n),'symmetric');

subplot(2,2,3),imshow(uint8(minImage))

minImage = ordfilt2(speckleImage,1,ones(m,n),'symmetric');

subplot(2,2,4),imshow(uint8(minImage))

% mid-point filter

f1 = ordfilt2(gaussianImage, 1, ones(m,n), 'symmetric');

f2 = ordfilt2(gaussianImage, m\*n, ones(m,n), 'symmetric');

mpImage = imlincomb(0.5, f1, 0.5, f2);

figure(10),

subplot(2,2,1),imshow(uint8(mpImage))

f1 = ordfilt2(poissonImage, 1, ones(m,n), 'symmetric');

f2 = ordfilt2(poissonImage, m\*n, ones(m,n), 'symmetric');

mpImage = imlincomb(0.5, f1, 0.5, f2);

subplot(2,2,2),imshow(uint8(mpImage))

f1 = ordfilt2(spImage, 1, ones(m,n), 'symmetric');

f2 = ordfilt2(spImage, m\*n, ones(m,n), 'symmetric');

mpImage = imlincomb(0.5, f1, 0.5, f2);

subplot(2,2,3),imshow(uint8(mpImage))

f1 = ordfilt2(speckleImage, 1, ones(m,n), 'symmetric');

f2 = ordfilt2(speckleImage,m\*n, ones(m,n), 'symmetric');

mpImage = imlincomb(0.5, f1, 0.5, f2);

subplot(2,2,4),imshow(uint8(mpImage))

% alpha trimmed mean filter

d = 4;

atmImage = imfilter(double(gaussianImage), ones(m, n), 'symmetric');

for k = 1:d/2

atmImage = imsubtract(atmImage, ordfilt2(double(gaussianImage), k, ones(m, n), 'symmetric'));

end

for k = (m\*n - (d/2) + 1):m\*n

atmImage = imsubtract(atmImage, ordfilt2(double(gaussianImage), k, ones(m, n), 'symmetric'));

end

atmImage = atmImage / (m\*n - d);

figure(11),

subplot(2,2,1),imshow(uint8(atmImage))

atmImage = imfilter(double(poissonImage), ones(m, n), 'symmetric');

for k = 1:d/2

atmImage = imsubtract(atmImage, ordfilt2(double(poissonImage), k, ones(m, n), 'symmetric'));

end

for k = (m\*n - (d/2) + 1):m\*n

atmImage = imsubtract(atmImage, ordfilt2(double(poissonImage), k, ones(m, n), 'symmetric'));

end

atmImage = atmImage / (m\*n - d);

subplot(2,2,2),imshow(uint8(atmImage))

atmImage = imfilter(double(spImage), ones(m, n), 'symmetric');

for k = 1:d/2

atmImage = imsubtract(atmImage, ordfilt2(double(spImage), k, ones(m, n), 'symmetric'));

end

for k = (m\*n - (d/2) + 1):m\*n

atmImage = imsubtract(atmImage, ordfilt2(double(spImage), k, ones(m, n), 'symmetric'));

end

atmImage = atmImage / (m\*n - d);

subplot(2,2,3),imshow(uint8(atmImage))

atmImage = imfilter(double(speckleImage), ones(m, n), 'symmetric');

for k = 1:d/2

atmImage = imsubtract(atmImage, ordfilt2(double(spImage), k, ones(m, n), 'symmetric'));

end

for k = (m\*n - (d/2) + 1):m\*n

atmImage = imsubtract(atmImage, ordfilt2(double(spImage), k, ones(m, n), 'symmetric'));

end

atmImage = atmImage / (m\*n - d);

subplot(2,2,4),imshow(uint8(atmImage))

%%%%%%%%%%%%%%%%%%%%% arithmetic.m %%%%%%%%%%%%%%%%%%%

function v = arithmetic(A)

[M N] = size(A);

sum = 0;

for i = 1:M

for j = 1:N

sum = sum + A(i,j);

end

end

v = sum / (M \* N);

%%%%%%%%%%%%%%%%%%%%% geometric.m %%%%%%%%%%%%%%%%%%%

function v = geometric(A)

[M N] = size(A);

prod = 1;

for i = 1:M

for j = 1:N

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

end

end

v = prod ^ (1/(M \* N));

%%%%%%%%%%%%%%%%%%%% harmonic.m %%%%%%%%%%%%%%%%

function v = harmonic(A)

[M N] = size(A);

sum = 0;

for i = 1:M

for j = 1:N

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

end

end

v = (M \* N)/sum;

%%%%%%%%%%%%%%%%%% contraharmonic.m %%%%%%%%%%%%%%%%%

function v = contraharmonic(A)

[M N] = size(A);

Q = 1;

sum = 0;

sum1 = 0;

for i = 1:M

for j = 1:N

sum = sum + (A(i,j)^(Q+1));

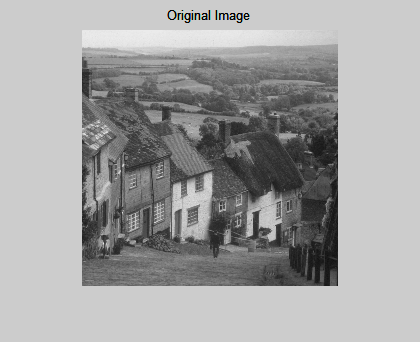
sum1 = sum1 + (A(i,j)^(Q));

end

end

v = sum/sum1;

1. Input :



1. Noisy Image :

****

1. Output :

Arithmetic filtering

Geometric filtering

Harmonic filtering

Contra-harmonic filtering with Q = 1

Median filtering

Max filtering

Min filtering

Mid-point filtering

Alpha trimmed mean filtering with d =4

1. Conclusion : best filters for each type of noise are listed below
   1. Gaussian – Arithmetic filter
   2. Poisson – Arithmetic filter
   3. Salt and Pepper – Alpha trimmed mean filter
   4. Speckle – Arithmetic filter