BILATERAL FILTERING

1. CODE

function [new\_image,rmsd]=myBilateralFiltering(sigma\_s,sigma\_r)

a=load('../data/barbara.mat');

orig\_image=a.imageOrig;

orig\_image=orig\_image/255.0;

%imwrite(orig\_image,'../images/original\_image.png');

[row,col]=size(orig\_image);

image = imnoise(orig\_image,'gaussian',0.0004);

%imwrite(image,'../images/blurred\_image.png');

%Parameters

window=9;

[row,col]=size(image);

[x\_val y\_val]=meshgrid(-window:window,-window:window);

space\_filter=exp(-(x\_val.^2+y\_val.^2)/(2\*sigma\_s^2));

final\_image=image;

for i=1:row

for j=1:col

row\_min=max(i-window,1);

col\_min=max(j-window,1);

row\_max=min(i+window,row);

col\_max=min(j+window,col);

cropped\_image=image(row\_min:row\_max,col\_min:col\_max);

range\_filter=exp(-double(cropped\_image-image(i,j)).^2/(2\*sigma\_r^2));

filter\_matrix=range\_filter.\*space\_filter((row\_min:row\_max)+window+1-i, ...

(col\_min:col\_max)+window+1-j);

W=sum(filter\_matrix(:));

final\_image(i,j)=sum(sum(filter\_matrix.\*double(cropped\_image)))/W;

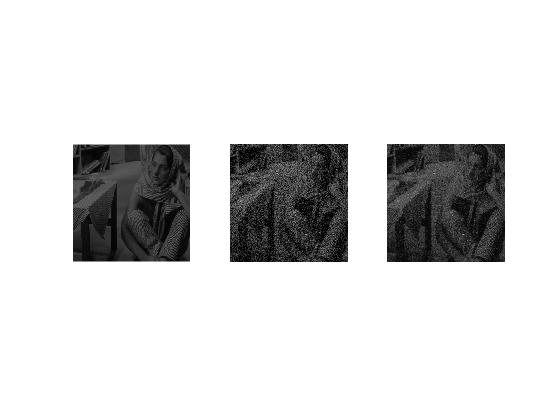
end

end

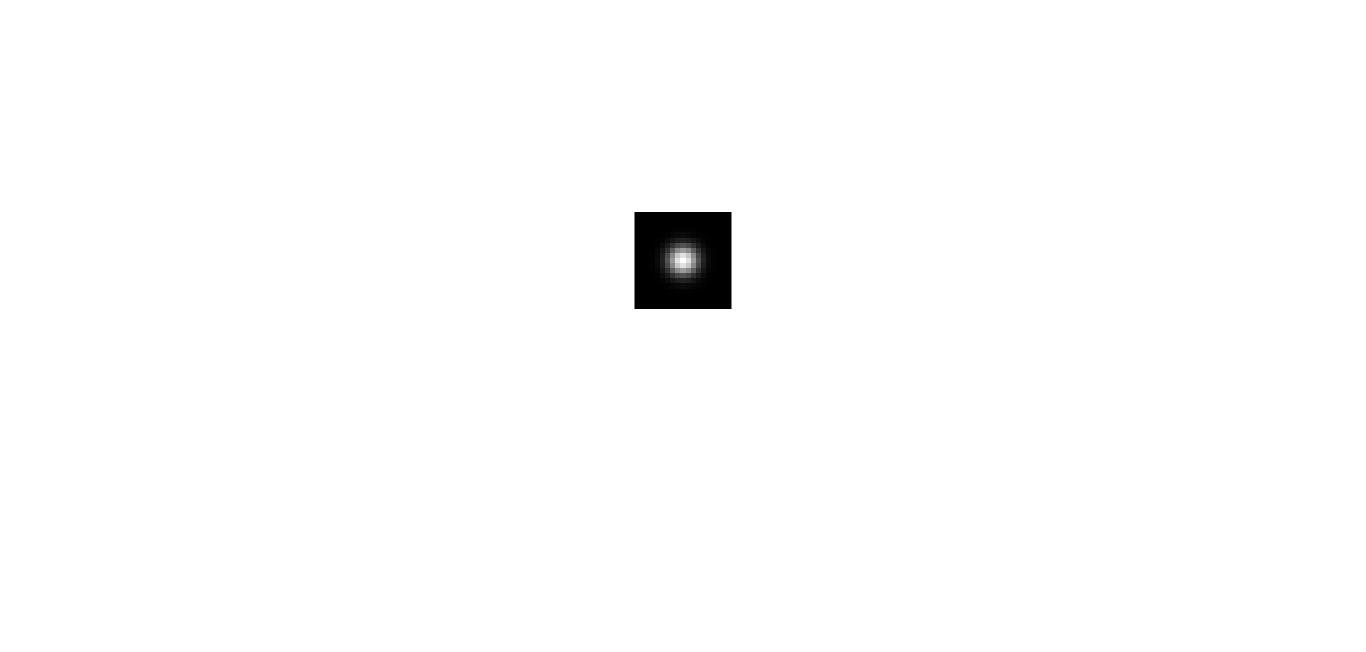
rmsd=sqrt(norm(final\_image-orig\_image)^2/(row\*col));

%imwrite(final\_image,'../images/final\_image.png');

1. Final result:



1. Gaussian space filter mask:



1. Optimal values:

Sigma for range filter= 0.11

Sigma for space filter= 2

Optimal RMSD value= 0.0066

1. A. RMSD value for sigma\_space=1.8 and sigma\_r=0.11 : 0.0068

B. RMSD value for sigma\_space=2.2 and sigma\_r=0.11 : 0.0068

C. RMSD value for sigma\_space=2 and sigma\_r=0.10 : 0.0069

D. RMSD value for sigma\_space=2 and sigma\_r=0.12 : 0.0069