

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(:));

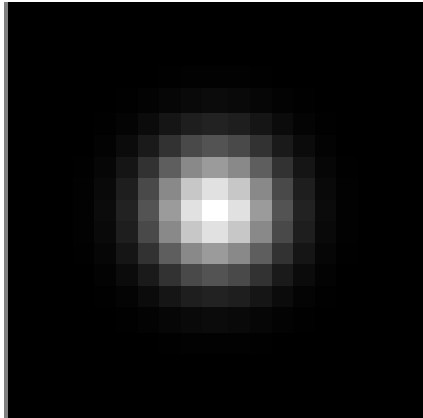
        final_image(i,j)=sum(sum(filter_matrix.*double(cropped_image)))/W;
```

```
end  
end  
rmsd=sqrt(norm(final_image-orig_image)^2/(row*col));  
  
%imwrite(final_image,'../images/final_image.png');
```

2. Final result:



3. Gaussian space filter mask:



4. Optimal values:
Sigma for range filter= 0.11
Sigma for space filter= 2
Optimal RMSD value= 0.0066
5. 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