CS663 : Digital Image Processing **Deadline: 12:00 AM, Aug-18 2021**

Assignment 2

1 Question #8

Consider the two images in the homework folder 'barbara256.png' and 'kodak24.png'. Add zero-mean Gaus- sian noise with standard deviation = 5 to both of them. Implement a bilateral filter and show the outputs of the bilateral filter on both images for the following parameter configurations: (s = 2,r = 2);(s = 0.1,r = 0.1);(s = 3,r = 15). Comment on your results in your report. Repeat when the image is corrupted with zero-mean Gaussian noise of = 10 (with the same bilaterial filter parameters). Com- ment on your results in your report. For the bilateral filter implementation, write a MATLAB function mybilateralfilter.m which takes as input an image and parameters r,s. Implement your filter using at the most two nested for-loops for traversing the image indices. For creating the filter, use functions like meshgrid and vectorization for more efficient implementation. Include all image ouputs as well as noisy images in the report.

First we will add guassian noise

```
J1 = imread("barbara256.png");
imshow(J1)
J1noise = imnoise(J1,'gaussian',0,(5/255)*(5/255));
imshow(J1noise)
J1noise = im2double(J1noise);

J2 = imread("kodak24.png");
imshow(J2)
J2noise = imnoise(J2,'gaussian',0,5/255*5/255);
imshow(J2noise)
J2noise = im2double(J2noise);
```

Now we will write mybilateralfilter function Firstly - We will write distance calculating term which would be constant for given filter size and spatial variance

```
function Imnew = mybilateralfilter(image,sds,sdr)
    numf=9;
    n0 = 5;
    n1 = 4;

    d1 = 1:numf;
    [X,Y] = meshgrid(d1,d1);
    X = X-n0;
    Y = Y-n0;
    Z= X.*X +Y.*Y;
    Ds = exp(-(Z)/(2*sds*sds))/(sds*sqrt(2*pi));
```

Secondly a function to calculate the radial term, it takes intensity difference and radial standard deviation and returns radial term

```
function val = gs1(I1,sd)
    val = 1/(sd*sqrt(2*pi))*exp(-(I1.*I1)/(2*sd*sd));
end
```

Finally for loop on all pixels

```
    [a,b] = size(image);
    new = zeros(a-numf,b-numf);
    for i = n0:a-n0
        for j = n0:b-n0
            I1 = image(i-n1:i+n1,j-n1:j+n1)-image(i,j);
            num =  Ds.*gs1(I1,sdr).*image(i-n1:i+n1,j-n1:j+n1);
            numsum = sum(sum(num));
            den =  Ds.*gs1(I1,sdr);
            densum = sum(sum(den));
            new(i-n0+1,j-n0+1) = numsum/densum;
        end
    end
    imshow(new)
    Imnew = new;
end
```

Results Barbara

```
I1 =mybilateralfilter(J1noise,2,2);
imwrite(I1,"1.png")
I2 =mybilateralfilter(J1noise,0.1,0.1);
imwrite(I2,"2.png")
I3 =mybilateralfilter(J1noise,3,15);
imwrite(I3,"3.png")
```

Results House

```
I1 =mybilateralfilter(J2noise,2,2);
imwrite(I1,"1J.png")
I2 =mybilateralfilter(J2noise,0.1,0.1);
imwrite(I2,"2J.png")
I3 =mybilateralfilter(J2noise,3,15);
imwrite(I3,"3J.png")
```

Figure 1: Barbara with Guassian Noise

Figure 2: House with Guassian noise

Figure 3: Barbara - sds =2 ,sdr =2

Figure 4: Barbara - sds =0.1 ,sdr =0.1

Figure 5: Barbara - sds =3 ,sdr =15

Figure 6: House - sds =2 ,sdr =2

Figure 7: House - sds =0.1 ,sdr =0.1

Figure 8: House - sds =3 ,sdr =15