EE420 – Digital Image Processing – Homework 2

Question 1 - Grayscale image denoising using Bilateral filter:

1. The MATLAB code is attached to this PDF in a '/code' folder.

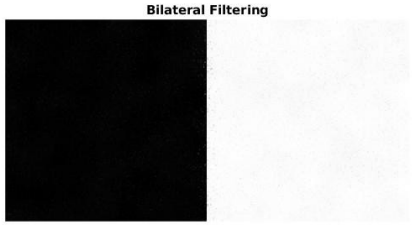
2. Key steps in MATLAB implementation:
   gray_bilateral_filter.m:
   This function is the implementation of the gray bilateral filter. The function gets the image, the space and the range parameters (sigmas) and computes a Gaussian filtered and Bilateral filtered images (implementation from https://www.mathworks.com/matlabcentral/fileexchange/12191-bilateral-filtering)

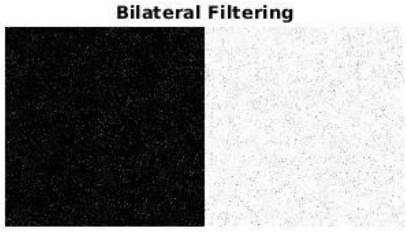
   - I defined 20X20 symmetric image borders to the original image (line 30).
   - I defined the filter size using the input space sigma (line 33).
   - I computed the Gaussian spatial filter mask (line 37-38).
   - I applied the Gaussian filter on the image using convolution (line 44).
   - I cropped the convoluted image back to it original size (line 45).
   - I applied bilateral filter on the image of the same dimensions as the Gaussian filter (line 48-58).
   - I calculated the Gaussian intensity kernel H (line 61).
   - I multiplied the Gaussian spatial filter kernel and the Gaussian intensity kernel to get the bilateral filtered response (line 64-65).
   - I returned the bilateral filtered image and the Gaussian filtered image (line 68-69).

   HWK2_GrayScaleDenoising_using_BLFilter.m:
   - I read the original image (line 5).
   - I defined the space and range parameters (line 13-14) – you can fine the reasons for choosing those values on questions 4-5.
   - I called the gray_bilateral_filter function in order to get the filtered images (line 16).
   - I showed the images (line 22-37).

3. The pixels near the image boundary being handled by applying reflection across edges (add a symmetric 20X20 pixels border to the image, apply the filter and crop to the original image size).

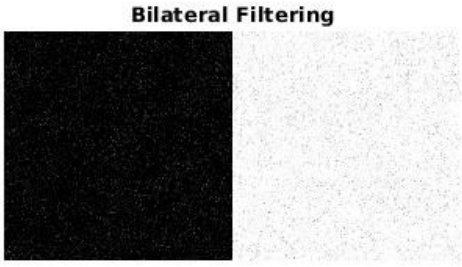
4. I chose sigma_s to be 6.7, because as we learn in class, the spatial parameter could be chosen proportional to image size - 2% of the image diagonal (sqrt(150^2 + 300^2) = 335 → 335*0.02 = 6.7). Moreover, lower sigma_s didn't remove the black spots from the white part of the image. For example, in the next 2 images, you can see the output image for 2 cases – the top image uses sigma_s = 6.7 and the lower image uses sigma_s = 3:

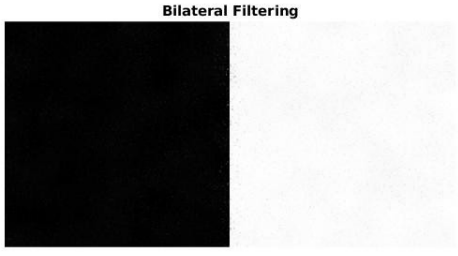


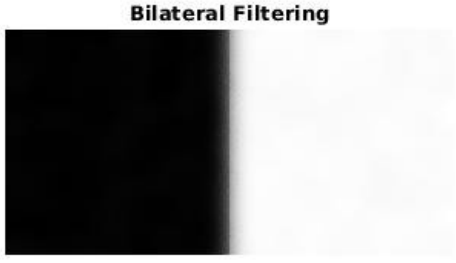


You can see that for lower sigma_s the filter doesn't work as well.

5. sigma_r is the "minimum" amplitude of an edge. I chose sigma_r to be 0.3, because it's the minimum value of sigma_r that removes the noise from the image, but it is not high enough so the image becomes blurry. Below there are 3 images – the first is the output image when using sigma_r = 0.1 (noisy), the second is the output image when using sigma_r = 0.4 (denoised), and the third is the output image when using sigma_r = 1.0 (blurry):







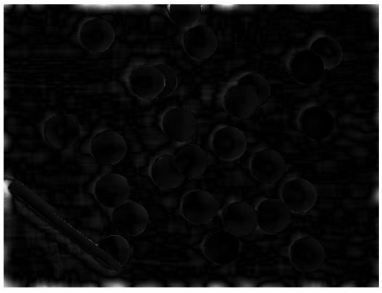
6. The original image and the denoised result using Bilateral and Gaussian filtering:

Question 2 - Color image processing using Bilateral filter:

1. The MATLAB code is attached to this PDF in a '/code' folder.

2. Upon bilateral filtering using the parameter values σ_r, σ_s specified in the starter code, the texture of the wooden table become smoother, and we can't really notice the original texture. The output filtered image is shown below:

3. I can see a difference between your implementation and MATAB's built-in implementation in the generated difference image:

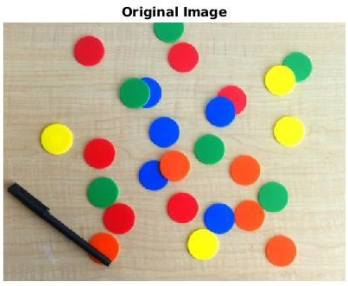


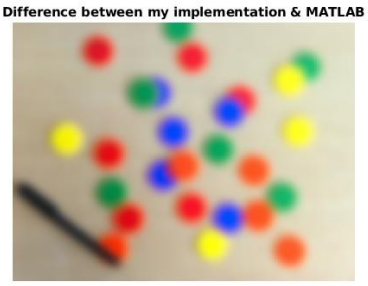
In the difference image, we can see the edges of the coins and bright spots on the edges of the image. The reason for that difference probably comes from a different way of solving the near edge problem – while I used the symmetric reflection solution, the MATLAB implementation probably uses a different solution.

4. For $\sigma_s=5$ and $\sigma_r=4*\text{patchStd}$, we can start noticing the table texture. The different between $\sigma_s=7$ is subtle, so I'll show the filtered image for $\sigma_s=3$ (on the right):

We can see that for smaller space parameter (sigma_s), we are getting a smaller space smoothing.

5. The original image, result of Bilateral filtering and Gaussian filtering:





Question 3 - Image enhancement by joint/cross bilateral filtering:

1. The MATLAB code is attached to this PDF in a '/code' folder.

2. The values of σ_s I chose for the joint Bilateral filter and the standard Bilateral filter are 1.2 for the joint Bilateral filter and 5.7 for the standard Bilateral filter. In the same way as question 1 section 4, I calculated the space sigma for the standard Bilateral filter as 2% of the image diagonal (sqrt(202^2 + 202^2) = 285 → 285*0.02 = 5.7).
   Fot the joint Bilateral filter, I chose a smaller sigma_s (sigma_s = 1.2), to smooth the image a little bit less.

3. A screenshots of various steps in the workflow organized as instructed: