EE7374 – Digital Image Processing

Homework-2

Mingze Sun

47505501

1. Grayscale image denoising using Bilateral filter

Deliverables & Questions

1. Completed MATLAB code for `HWK2_GrayScaleDenoising_using_BLFilter.m` and `gray_bilateral_filter`. (10 points)
2. Description of key steps in MATLAB implementation with references to relevant line numbers. Be as detailed as possible in your explanation. (7 points)

   Gray_bilateral_filter.m

   Line 7: use gaussian_filter function implemented by myself

   8: identify r = (filter_size -1)/2 for filter_size = 2*ceil(3*sigma_s)+1

   9: identify size of image

   10: prepare the matrix for storing

   11: prepad image boundary by using replicate operation

   14: get the window to be filtered

   15: rotate 180 degree for convolution

   16: implement convolution

   17: store the sum of result to target matrix

   26: produce grids for gaussian filter

   27,28: generate gaussian filter

   30: prepad image boundary by using replicate operation

   34: get intensity of pixels

   35: calculate the intensity difference

   36: produce intensity gaussian filter kernel

   37: get weight by multiply the special and intensity gaussian kernels

   38: multiply the weight with intensity of pixels

   39: normalization

3. How are pixels near the image boundary being handled? (3 points)

I handled it by using replicate operation. I pad replicated r rows and columns pixel in both directions before doing operation, by this way, the result image will be same size with original image.

4. What value of σ_s did you choose and why? (1 point)

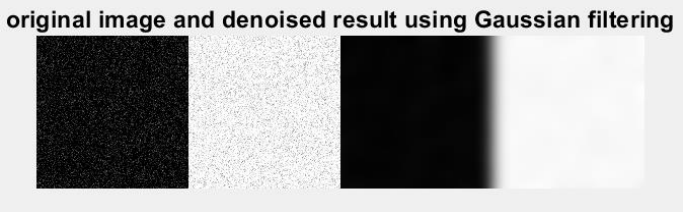
   I choose 7 because its smooth most of noise pixels and filter size is proper for intensity kernel.

5. What value of σ_r did you choose and why? (2 points)

   I choose 1 because the smaller σ_r, the edge will be clearer. But it can't be too small because the noise pixels will be sensitive to small σ_r and appear. And set it to 1 get the best result.

6. Screenshots of the original image and the denoised result using Bilateral and Gaussian filtering. Label each screenshot clearly. Failure to do so will result in deduction of points. (2 points)

   σ_s = 7 and σ_r = 1



7. Label each screenshot clearly. Failure to do so will result in deduction of points. (2 points)

2. Color image processing using Bilateral filter

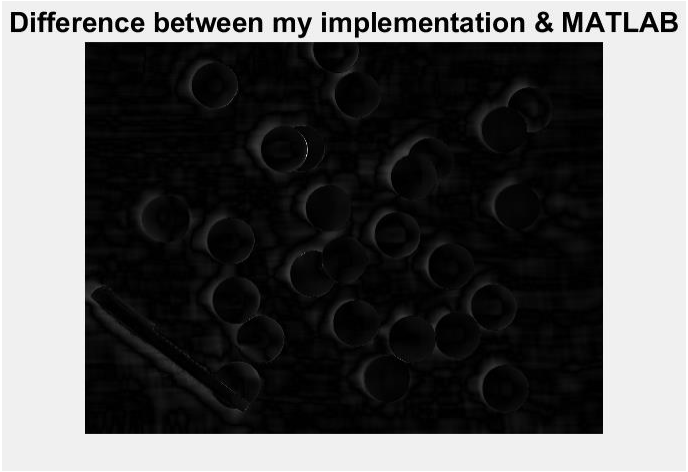
Deliverables & Questions

1. Completed MATLAB code for `HWK2_ColorImageFiltering_using_BLFilter.m` and `bilateral_filter`. (10 points)
2. What happens to the texture of the wooden table upon bilateral filtering using the parameter values σ_r, σ_s specified in the starter code? (2 points)

   The texture of the wooden table is smoothed.

3. Do you notice any difference between your implementation and MATAB's built-in implementation? If so where are the differences and speculate as to the reasons for the difference?

   Yes. Difference mainly appears in the edge of chips. The reason may be the method I used to implement the bilateral filter is linear while the implement method MATLAB used is nonlinear.



4. How does the appearance of the image change if you pick smaller values for σ_s and σ_r? Try picking $\sigma_s = 5$ and $\sigma_r = 4 * \texttt{patchStd}$ for example. (5 points)

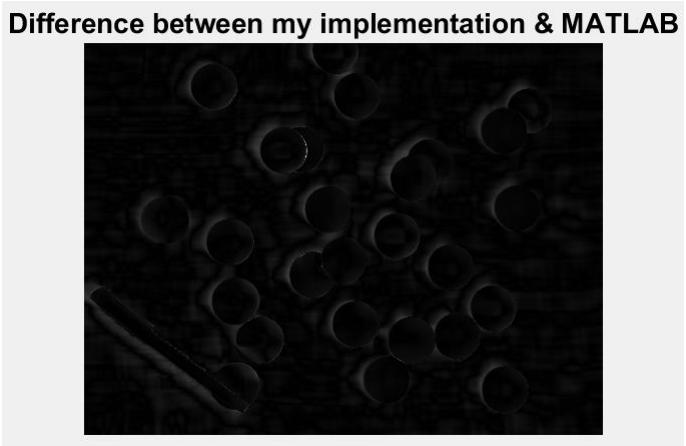
$\sigma_s = 7$

$\sigma_s = 5$

The texture becomes more visible.

5. Screenshots of the original image, result of Bilateral filtering and Gaussian filtering. Label each screenshot clearly. Failure to do so will result in deduction of points. (3 points)

$\sigma_s = 7$ and σ_r= `4*patchVar`



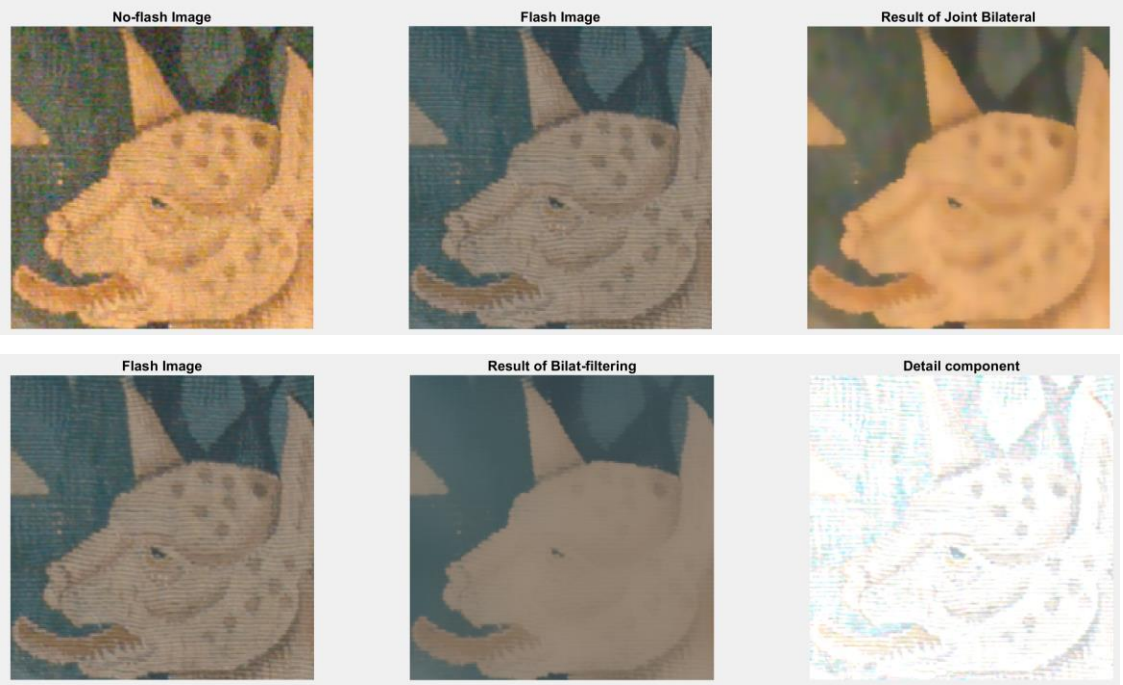
3. Image enhancement by joint/cross bilateral filtering

Deliverables & Questions

1. Completed MATLAB code for `HWK2_ImageEnhancement_using_CrossBLFilter.m` and `joint_bilateral_filter`. (15 points)
2. What value of σ_s did you choose for the **joint Bilateral filter** and the **standard Bilateral filter** and why? (5 points)

   I choose $\sigma_s = 3$ for the joint Bilateral filter because we want to keep as much as more details, so we don't want to smooth the texture too much. But we still want to smooth the noisy pixels. Set $\sigma_s = 3$ make us get the smallest filter size but keep the ability to smooth the noisy pixels.

   I choose $\sigma_s = 11$ for the standard Bilateral filter, because we want to smooth the texture as much as possible but when σ_s increases, the filter size will increase and influence intensity kernel to make edge clearer, which is not good for detail transfer. And with filter size increase, the time spend will also increase rapidly.
3. Screenshots of various steps in the workflow organized as follows: (5 points)




No-flash Image
Flash Image
Result of Enhenced image