Problem 3: Noise Removal (35%)

3(a) Mix noise in color image (Basic: 15%)

In this part, you will perform noise removal on a color image corrupted with "mix" type of noise. The original and noisy pepper images are shown in Figure 9. Since the image has mixed noise, you will need more than one noise removal filters.

(a) The Pepper image

(b) The noisy Pepper image (peppers_mix.raw)

Figure 9

1) Please identify noise types in the image and answer the following questions: a) do all channels have the same noise type? b) Should you perform filtering on individual channels separately for both noise types? c) What filters would you use to remove mixed noise? d) Can you cascade these filters in any order (justify your answer). e) Please discuss the affect of different filter window sizes.
2) Try to get the best results in removing mixed noise: a) describe your method and discuss its short-comings. b) Give some suggestions to improve its performance. There is no need to implement but explain them along with your motivation. You can use suitable plots/images to support your arguments.

3(b) Bilateral Filtering (Basic: 8%)

In most low-pass linear filters, we often see degradation of edges. However, using some non-linear filters, we can preserve the edges. Bilateral filters are one such kind of filters. A bilateral filter, in its most general form, can be given by [1]:

$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \mathbf{f}(\xi)\ c(\xi, \mathbf{x})\ s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) \quad d\xi \tag{3}$$

where $c(\xi, \mathbf{x})$ and $s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))$ measures, respectively, the spatial closeness and the pixel intensity closeness (*photometric* similarity) between point $\mathbf{x}$ and its nearby point ξ and

$$k(\mathbf{x}) = \int_{-\infty}^{\infty} c(\xi, \mathbf{x})\ s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) \quad d\xi$$

is a normaliztion factor. The discrete version of a bilateral filter is given by:

$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \sum_{\xi \in \Omega_{\mathbf{x}}} \mathbf{f}(\xi)\ c(\xi, \mathbf{x})\ s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) \tag{4}$$

Prob 3(b) continued on next page...

where $\Omega_{\mathbf{x}}$ is a finite window centered around $\mathbf{x}$. The Gaussian function is often chosen for both $c(.)$ and $f(.)$ so that we get

$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \sum_{\xi \in \Omega_{\mathbf{x}}} \mathbf{f}(\xi) \ \exp\left(-\frac{\|\mathbf{f}(\xi) - \mathbf{f}(\mathbf{x})\|_2^2}{2\sigma_c^2}\right) \ \exp\left(-\frac{\|\xi - \mathbf{x}\|_2^2}{2\sigma_s^2}\right), \tag{5}$$

where $\|a\|_2^2$ denotes the square 2-norm (Eucledean norm), and σ_c and σ_s are spread parameters (analogous to standard deviation in Eq. (1)).

1) Implement the Bilateral filter and apply it to lena.raw as shown in Figure 10.

2) Explain the role of functions c and s. Discuss the change in filter's performance with respect to the values of σ_c, σ_s and the window size (*i.e.* size of $\Omega_{\mathbf{x}}$).

3) Does this filter perform better than linear filters you implemented in Problem 3(a)? Justify your answer in words.

Figure 10: Noisy Lena (lena.raw)

3(c) Non-Local Means (NLM) Filtering (Advanced: 12%)

For this part, you may refer to [2].

1) Briefly explain the Non-Local Means Algorithm (NLM) and implement it. Please DO NOT use any code from the Internet or other sources, as it would be considered as plagiarized.

2) Apply NLM filtering to lena.raw (Figure 10). Try several filter parameters and discuss their effect on filtering process. Clearly state your final choice of parameters in your report.

3) Compare the performance of NLM with filters used in Problem 3(a) and Problem 3(b). Can you deduce a relationship between the NLM filter and the bilateral filter (with respect to s and c in Eq. (4))?

Note: DO NOT quote statements directly from [1, 2] or any other online source(s). Explain in your own words. Reports and source codes are subject to verification for any plagiarism.

3(d) Block Matching 3-D (BM3D) Transform Filter (Bonus: 10%)

In this part, you would get familiar with a state-of-the-art filtering method proposed in [3].

1) Please explain the BM3D algorithm in your own words, and implement the BM3D filter (Write your own code or use any available online source code but include the source in your reference) to denoise lena.raw

(Figure 10). It is recommended that you use the code provided by the authors on their website [4]. Their code is written in MATLAB; so it is okay to use MATLAB for this part, even if you have been coding on C/C++ platform (You would still qualify for 5% bonus points if you have used C/C++ everywhere else). Discuss the effects of several tunable parameters on the denoising result.
2) Both NLM and bilateral filter are spatial domain filters. How would you classify BM3D:- spatial domain, frequency domain, or both? Justify your answer.
3) Conduct qualitative performance comparison between NLM, bilateral filtring, and BM3D.