EE 569: Homework #1

Issued: 01/10/2022 Due: 11:59PM, 01/30/2022

General Instructions:

1. Read *Homework Guidelines* and *MATLAB Function Guidelines* for the information about homework programming, write-up and submission.
2. If you make any assumptions about a problem, please clearly state them in your report.
3. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.

Problem 1: Image Demosaicing and Histogram Manipulation (45%)

(a) Bilinear Demosaicing (10%)

To capture color images, digital camera sensors are usually arranged in form of a color filter array (CFA), called the Bayer array, as shown in Figure 1. Since each sensor at a pixel location only captures one of the three primary colors (R, G, B), the other two colors have to be re-constructed based on their neighbor pixel values to obtain the full color. Demosaicing is the process of translating this Bayer array of primary colors into a color image that contains the R, G, B values at each pixel.

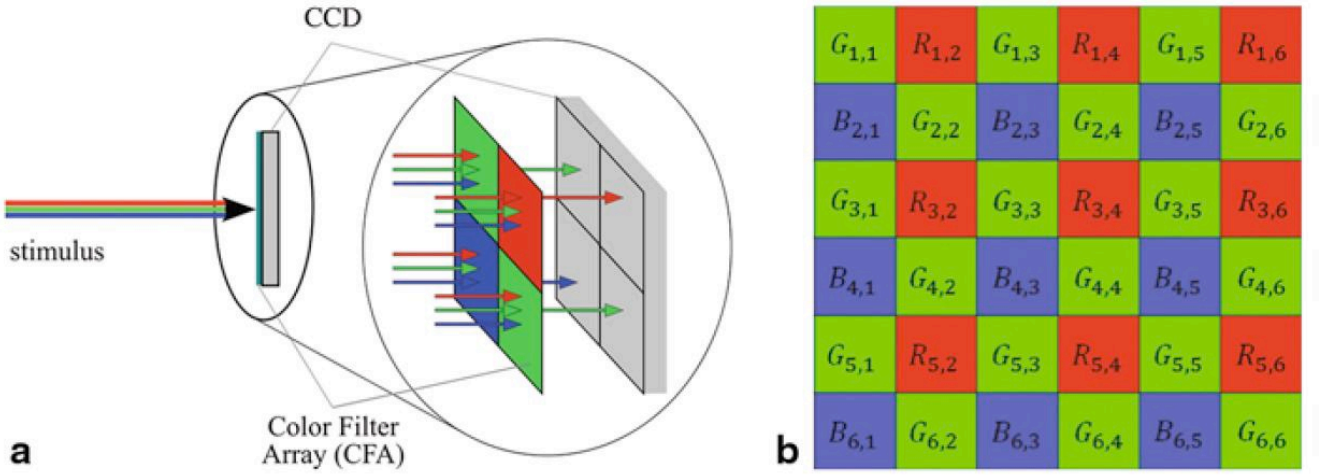

Figure 1: (a) Single CCD sensor covered by a CFA and (b) Bayer pattern [1].

Implement the simplest demosaicing method based on bilinear interpolation. Exemplary demosaicing results are given in Figure 2. With this method, the missing color value at each pixel is approximated by bilinear interpolation using the average of its two or four adjacent pixels of the same color. To give an example, the missing blue and green values at pixel $R_{3,4}$ are estimated as:

$$\hat{B}_{3,4} = \frac{1}{4}(B_{2,3} + B_{2,5} + B_{4,3} + B_{4,5})$$

$$\hat{G}_{3,4} = \frac{1}{4}(G_{3,3} + G_{2,4} + G_{3,5} + G_{4,4})$$

As for pixel $G_{3,3}$, the blue and red values are calculated as:

$$\hat{R}_{3,3} = \frac{1}{2}(R_{3,2} + R_{3,4})$$
$$\hat{B}_{3,3} = \frac{1}{2}(B_{2,3} + B_{4,3})$$

(a) (b)

Figure 2: (a) The original image and (b) the demosaiced image by bilinear interpolation.

Figure 3: The *House* image before demosaicing.

(1) Apply the bilinear demosaicing to the *House* image in Figure 3 and show your results.
(2) Compare your demosaiced image with the color image *House_ori* which is obtained from a more advanced demosaicing algorithm. Do you observe any artifacts? If yes, explain the cause of the artifacts and provide your ideas to improve the demosaicing performance.

(b) Histogram Manipulation (20%)

Implement two histogram equalization techniques:

- Method A: the transfer-function-based histogram equalization method,
- Method B: the cumulative-probability-based histogram equalization method

to enhance the contrast of the *Hat* image in Figure 4 below.

(1) Plot the histograms of the original image. The figure should have the intensity value as the x-axis and the number of pixels as the y-axis.
(2) Apply Method A to the original image and show the enhanced image. Plot the transfer function.
(3) Apply Method B to the original image and show the enhanced image. Plot the cumulative histograms before and after enhancement.
(4) Discuss your observations on these two enhancement results. Which one do you think is better and why?

Note that MATLAB users CANNOT use functions from the Image Processing Toolbox except displaying function like imshow().

Figure 4: Hat image

(c) Contrast Limited Adaptive Histogram Equalization (15%)

Instead of histogram equalization using global statistics from the entire image, adaptive histogram equalization breaks one image into several regions (tiles) and finds the histogram in each separately. Contrast Limited Adaptive Histogram Equalization (CLAHE) is one of the most popular algorithms. Please read the paper [2] carefully to learn about CLAHE.

In this problem, you will apply the CLAHE to do image haze removal, where you remove the fog in the provided image taken in a bad weather. Figure 5 shows an example before and after the haze removal. The process can be done through the following 3 steps.

Step-1: Transform the image from *RGB* color space to *YUV* color space. The transformation can be found in the Appendix.
Step-2: Apply a histogram equalization algorithm on the *Y* channel to get *Y'*.
Step-3: Combine *Y'* with *U* and *V*, transform them back to RGB color space. The resulted image is your haze-removed image.

(1) Explain CLAHE in your own words.

(2) Perform the above three steps on *Taj_Mahal.raw* by applying the two histogram equalization methods in Problem 1(b) in step-2. Show the resulted images for both methods.

(3) Repeat the three steps but apply the CLAHE in step-2. Here, you are allowed to use open-source code for CLAHE. For C++, you can use ***OpenCV***. For Matlab, you can check function ***adapthisteq***. Tune the

hyperparameters including the number of tiles and the clip limit. Show the resulted image that you think is the most pleasant subjectively.

(4) Compare your results between (2) and (3). Discuss your observations.

(a) Original foggy image **(b) Defogged image**

Figure 5: An examples of image haze removal.

Problem 2: Image Denoising (40 %)

In this problem, you will implement a set of denoising algorithms to improve image quality. You can use the PSNR (peak-signal-to-noise-ratio) quality metric to assess the performance of your denoising algorithm. The PSNR value for R, G, B channels can be, respectively, calculated as follows:

$$\text{PSNR (dB)} = 10\log_{10}\left(\frac{\text{Max}^2}{\text{MSE}}\right)$$

$$\text{where MSE} = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(Y(i,j)-X(i,j)\right)^2$$

X : Original Noise-free Image of size $N \times M$

Y : Filterd Image of size $N \times M$

Max: Maximum possible pixel intensity = 255

Remove noise in the image in Figure 6(b), compare it with the original image in Figure 6(a), and answer the following questions:

(a) Basic denoising methods (10%)

(1) What is the type of embedded noise in Figure 6(b)? Justify your answer.

(2) Apply a linear filter to the noisy image. Compare the performance of two choices of the filter parameters – the uniform weight function and the Gaussian weight function under different filter sizes.

(a) Original image **(b) Noisy image**

Figure 6: The original and noisy Flower images.

(b) Bilateral Filtering (10%)

In most low-pass linear filters, we often see degradation of edges. However, using some nonlinear filters, we can preserve the edges. Bilateral filters are one such kind of filters. A discrete bilateral filter is given by:

$$Y(i,j) = \frac{\sum_{k,l} I(k,l) w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_c^2} - \frac{\|I(i,j) - I(k,l)\|^2}{2\sigma_s^2}\right)$$

where (k,l) is the neighboring pixel location within the window centered around (i,j), I is the image with noise, Y is the filtered image. σ_c and σ_s are two spread parameters.

(1) Implement the bilateral denoising filter and apply it to the noisy image.
(2) Explain the roles of σ_c and σ_s. Discuss the change in filter's performance with respect to the values of σ_c and σ_s.
(3) Does this filter perform better than linear filters you implemented in Problem 2(a)? Justify your answer in words.

(c) Non-Local Means (NLM) Filtering (10%)

The non-local mean filter utilizes the pixel value from a larger region rather the mean of a local window centered around the target pixel. A discrete non-local mean filter with Gaussian weighting function is as follows:

$$Y(i,j) = \frac{\sum_{k,l} I(k,l) w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = \exp\left(-\frac{\|I(N_{i,j}) - I(N_{k,l})\|_{2,a}^2}{h^2}\right)$$

where I, Y are the noisy and filtered images respectively, $N_{x,y}$ is the window centered around location (x, y), and h is the filtering parameter, $N' \leq N$ and $M' \leq M$ denote the window size of your choice.
The Gaussian weighted Euclidian distance between window $I(N_{i,j})$and $I(N_{k,l})$ is defined as:

$$\left\|I(N_{i,j}) - I(N_{k,l})\right\|_{2,a}^{2} = \sum_{n_1,n_2 \in \aleph} G_a(n_1,n_2)\left(I(i-n_1, j-n_2) - I(k-n_1, l-n_2)\right)^2$$

$$G_a(n_1,n_2) = \frac{1}{\sqrt{2\pi}a}\exp\left(-\frac{n_1^2+n_2^2}{2a^2}\right)$$

where $\aleph$ denotes the local neighborhood centered at the origin, $n_1, n_2 \in \aleph$ denotes the relative position in the neighborhood window. $a > 0$ is the standard deviation of the Gaussian kernel.

(1) Apply the NLM filter (using any open-source code, e.g. C++ can use ***OpenCV***, Matlab can check function ***imnlmfilt***) to the noisy image. Try several filter parameters and discuss their effect on filtering process. Clearly state your final choice of parameters in your report.
(Note that there are four parameters to discuss: the big search window size $\aleph$, the small neighbor window size N', the Gaussian smoothing parameter for the search window h, the Gaussian smoothing parameter for the neighbor window a. If the open source code doesn't provide ways to adjust a certain parameter, just analyze that parameter theoretically.)
(2) Compare the performance of NLM with filters used in Problem 2(a) and Problem 2(b).

(d) Mixed noises in color image (10%)

Figure 7 (b) is a noisy color image corrupted with mixed types of noises. Please identify noise types in the image and answer the following questions:

(1) What types of noises are there? Justify your answer.
(2) What filters would you like use to remove mixed noise? Can you cascade these filters in any order? Justify your answer.
(3) Get the best results in removing mixed noise. Include the following in your report:
1. Describe your method and show its results
2. Discuss its shortcomings.
3. Give some suggestions to improve its performance.

(a) (b)

Figure 7: (a) the original Flower image (b) the Flower image with mixed noises.

Problem 3: Special Effect Image Filters: Creating Frosted Glass Effect (15%)

An exemplary frosted glass effect for the *Lake* image is shown in Figure 8.

Figure 8 An example of frosted glass effect

This effect can be created by simply **replacing the color at each pixel with the color at a random pixel in its local neighborhood of size NxN**. For example, considering N=3, the color of the center pixel at location 0 can be replaced by the color at any of the location 0-8 (including the location 0 which means unchanged).

1	2	3
8	0	4
7	6	5

(1) Implement the frosted glass filtering with N = 5 or 7 and apply it to the *Flower.raw* image. Note that you can still formulate the process as a 2-D image filtering problem.
(2) Repeat the process for the noisy image *Flower_noisy.raw*. Describe your observations and discuss whether the noise leads to any difference.
(3) Considering the Bilateral filtering denoising algorithm. Try the following two processes:
    a. First, perform denoising on *Flower_gray_noisy.raw*. Then, apply the frosted-glass filtering.
    b. First, apply the frosted-glass filtering on *Flower_gray_noisy.raw.* Then, perform denoising.

    Compare your results and discuss your observations.

Appendix:

Problem 1: Image Demosaicing and Histogram Manipulation

House.raw	768x512	8-bit	gray
House_ori.raw	768x512	24-bit	color(RGB)
Hat.raw	256x256	8-bit	gray
Taj_Mahal.raw	600x400	24-bit	color(RGB)

Problem 2: Image Denoising & Problem 3

Flower.raw	768x512	24-bit	color(RGB)
Flower_noisy.raw	768x512	24-bit	color(RGB)
Flower_gray.raw	768x512	8-bit	gray
Flower_gray_noisy.raw	768x512	8-bit	gray

Note: "768x512" means "width=768, height=512".

Convert from RGB to YUV color space

$$Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16$$
$$U = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128$$
$$V = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128$$

Convert from YUV to RGB color space

$$R = 1.164(Y - 16) + 1.596(V - 128)$$
$$G = 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128)$$
$$B = 1.164(Y - 16) + 2.018(U - 128)$$

Reference Images

All images in this homework are from Google images [3], USC-SIPI image database [4], Kodak image dataset [5] or others [6].

References

[1] M. E. Celebi et al. (eds.), Color Image and Video Enhancement.
[2] Zuiderveld, Karel. "Contrast Limited Adaptive Histograph Equalization." Graphic Gems IV. San Diego: Academic Press Professional, 1994. 474–485.
[3] [Online] http://images.google.com/
[4] [Online] http://sipi.usc.edu/database/
[5] http://r0k.us/graphics/kodak/
[6] https://comedytravelwriting.com/5-best-photos-of-the-taj-mahal-in-fog/