EE 569: Homework #1

Issued: 1/12/2018 Due: 11:59PM, 2/4/2018

General Instructions:

1. Read *Homework Guidelines* and *MATLAB Function Guidelines* for the information about homework programming, write-up and submission. If you use the C/C++ programming language, you will get 5% of your total score as the bonus points.
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: Basic Image Manipulation (30%)

In this problem, you will conduct a series of simple manipulations on grayscale and color images to get familiar with image data access, processing and output.

(a) Color Space Transformation (Basic: 18%)

Different color spaces are adopted for different applications. In this problem, you will examine a couple of color spaces that are employed in image processing.

1. **Color-to-Grayscale Conversion**
   Each color pixel of an image is described by a triple (R, G, B) for red, green, and blue color intensities. There are many ways to convert a color image to its grayscale image. The **lightness** method averages the most prominent and least prominent colors: (max(R, G, B) + min(R, G, B)) / 2. The **average** method simply averages the values: (R + G + B) / 3. The **luminosity** method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. The formula for luminosity is 0.21 R + 0.72 G + 0.07 B. Convert the color *Tiffany* image in Figure 1 to its gray scale images using the above three methods. Show the three gray scale images and discuss which method works best overall.

Figure 1: Color Tiffany image

2. **CMY(K) Color Space** - The Cyan-Magenta-Yellow-(Black) (CMY(K)) color space is frequently used in image printing. It is defined by:

$$\begin{cases} C = 1 - R \\ M = 1 - G \\ Y = 1 - B \end{cases} \tag{1}$$

What is the CMY representation for test images *Bear* and *Dance* in Figure 2? For each input color image, produce 3 output grayscale images, corresponding to the cyan, magenta, yellow three channels and show these three grayscale images in your report.

Figure 2: Bear and Dance images

(b) Image Resizing via Bilinear Interpolation (Advanced: 12%)

Use the bilinear interpolation to re-size the input color image *airplane* of size 512x512 in Figure 3 to an output image of size 650x650. An image re-sizing example for the *House* image is given in Figure 4.

Figure 3: Airplane

(a) house image of size 512x512

(b) resized house image of size 650x650

Figure 4: **Original and resized House images.**

Problem 2: Histogram Equalization (40 %)

(a) Histogram Equalization (Basic: 12%)

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 *Desk* image in Figure 5 below.

(1) Plot the histograms of the red, green and blue channels 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 for each channel.

(3) Apply Method B the original image and show the enhanced image. Plot the cumulative histogram for each channel.

(4) Discuss your observations on these two enhancement results. Do you have any idea to improve the current result?

Figure 5: Desk image

(b) Image Filtering – Creating Oil Painting Effect (Advanced: 16%)

An exemplary oil-painting effect for the *Barn* image is shown in Figure 7. This effect can be implemented as a filter with the following two steps.

(a) Barn **(b) Barn with a reduced color set**

Figure 6: Barn images with a full and a reduced color sets

Step 1: Quantize all colors of the input color image, denoted by I_0, into an image containing only 64 colors, denoted by I_1. In other words, each channel should have only 4 values. The original and truncated *Barn* images are shown in Figs. 6 (a) and (b), respectively.

Step 2: For each pixel of image I_1, select the most frequent color in its *N*x*N* neighborhood (*N* is an odd number, usually ranging from 3 to 11), as the representative color for this pixel in another output image denoted by I_2. The result of N = 5 is shown in Figure 7.

Figure 7: Barn with the Oil-Painting Effect

Implement and apply the oil-painting filter to the *Star_Wars* and *Trojans* images as shown in Figs. 8 and 9.

(1) Show the 64-color version of both images by following Step 1. Discuss how you choose the threshold.
(2) Implement the oil painting process as described in Step 2 with several different values of *N*. Which *N* gives a better result? Discuss your observations.
(3) What happens to this filter when the input image has 512 colors instead? Show the corresponding results for both images and explain your observation.

Figure 8: Star_Wars

Figure 9: Trojans

(c) Image Filtering – Creating Film Special Effect (Advanced: 12%)

Design an algorithm to achieve the film special effect. An example is shown in Figure 10. Describe your algorithm step by step and apply your algorithm to test image *Girl* given in Figure 11. Show the film special effect result in your report.

Hint: investigate the color channel relationship between the input and output images in Figure 10, which will give your ideas about the film special effect.

Figure 10: An exemplary film special effect: the input (left) and the output (right).

Figure 11: Girl

Problem 3: Noise Removal (30 % + 10 %)

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

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

Perform noise removal on a color image corrupted by a "mix" type of noise. The original and noisy *Lena* images are shown in Figure 12.

Figure 12: The original and noisy Lena images.

1. Identify noise types in the noisy *Lena* image, and answer the following questions.

(1) Do all channels have the same noise type?
(2) Should you perform filtering on individual channels separately for both noise types?
(3) What filters would you like use to remove mixed noise?
(4) Can you cascade these filters in any order? Justify your answer.
(5) Discuss the effect of different filter window sizes.

2. 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.

(b) Principal component analysis (PCA) (Advanced: 15%)

Principal component analysis (PCA) is a dimensionality reduction algorithm, yet it can also be used as a tool for noise filtering. In this part, please read carefully the reference paper [1].

1. Explain why PCA can be used as a filtering approach for noisy data. What do the components mean and how to choose the number of components in image denoising.
2. Describe and implement the patch-based local PCA (PLPCA) in [1]. **DO NOT** use any code from the Internet or other sources, as it would be considered as plagiarism.
3. Apply the PLPCA to *House_noisy.raw* (Gaussian noise, $\sigma = 25$) in Figure 13. Optimize different parameters, e.g. patch size, the number of components, etc., and discuss their effect on the denoising process. State the best PSNR and your optimized parameters in your report.
4. Apply the approach in part (a) to *House_noisy.raw* and compare the performance of the PCA approach with those of filters used in part (a). Please explain the advantages of the PCA approach and why.
   DO NOT quote statements directly from [1] or any other online source. Try and explain in your own words. Reports and source codes are subject to verification for any plagiarism.

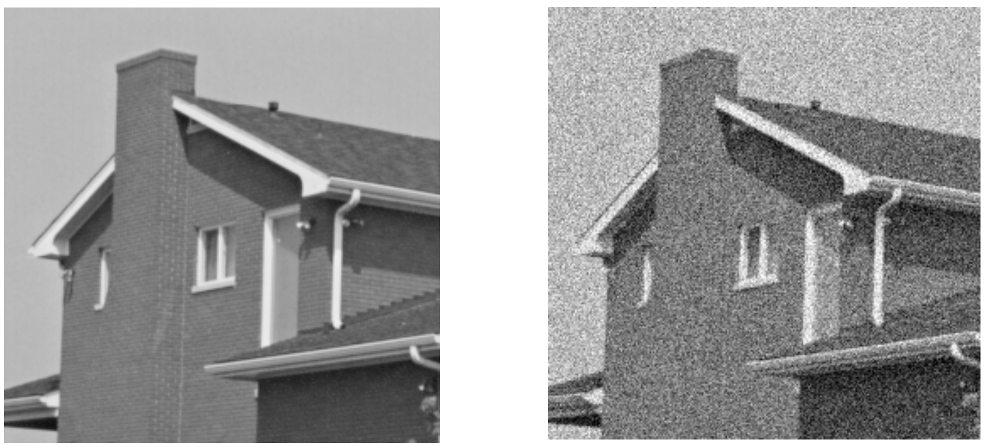

Figure 13: The grey-level original and noisy House images

(c) Block matching and 3-D (BM3D) transform filter (Bonus: 10%)

In this part, you will get familiar with another state-of-the-art denoising algorithm proposed in [2].

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 the noisy image: *House* (Figure 13). Discuss the effects of several tunable parameters on the denoising result.

Note: It is recommended that you use the code provided by the authors on their website [3]. Their code is written in MATLAB; so that it is okay to use MATLAB for this part (You still qualify for 5% bonus points if you have used C/C++ everywhere else).

2. Why does the author utilize block matching instead of other clustering methods like k-means? What is the motivation of step 2 (block matching and Wiener filtering) and compare the denoising performance with and without step 2.
3. How would you classify BM3D - spatial domain filter, frequency domain filter, or both? Justify your answer.
4. Conduct qualitative performance comparison between the algorithms developed for Problem 3(b) and BM3D.

Appendix:

Problem 1: Simple Image Manipulation

Tiffany.raw	512x512	24-bit	color(RGB)
Bear.raw	854x480	24-bit	color(RGB)
Dance.raw	854x480	24-bit	color(RGB)
Airplane.raw	512x512	24-bit	color(RGB)

Problem 2: Histogram Equalization

Desk.raw	400x300	24-bit	color(RGB)
Barn.raw	380x275	24-bit	color(RGB)
Barn_64.raw	380x275	24-bit	color(RGB)
Barn_oil.raw	380x275	24-bit	color(RGB)
Star_Wars.raw	600x338	24-bit	color(RGB)
Trojans.raw	1800x1200	24-bit	color(RGB)
Original.raw	1000x750	24-bit	color(RGB)
Film.raw	1000x750	24-bit	color(RGB)
Girl.raw	256x256	24-bit	color(RGB)

Problem 3: Noise Removal

Lena.raw	512x512	24-bit	color(RGB)
Lena_mixed.raw	512x512	24-bit	color(RGB)
House.raw	256x256	8-bit	grey
House_noisy.raw	256x256	8-bit	grey

Reference Images

All images in this homework are from Google images [4] or the USC-SIPI image database [5].

References

[1] Deledalle, Charles-Alban, Joseph Salmon, and Arnak S. Dalalyan. "Image denoising with patch based PCA: local versus global." BMVC. Vol. 81. 2011.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” Image Processing, IEEE Transactions on, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] [Online]. Available: http://www.cs.tut.fi/~foi/GCF-BM3D/

[4] [Online] http://images.google.com/

[5] [Online] http://sipi.usc.edu/database/