EE 569: Homework #1

Issued: 8/29/2014 Due: 11:59PM, 9/21/2014

General Instructions:

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

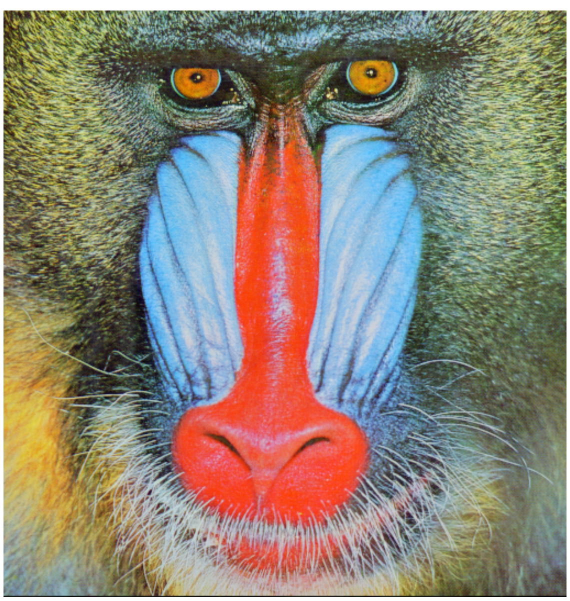
Problem 1: Simple Image Manipulation (30%)

In this problem, you are required to do a series of manipulations on grayscale and color images so that you learn how to access, modify and output image data.

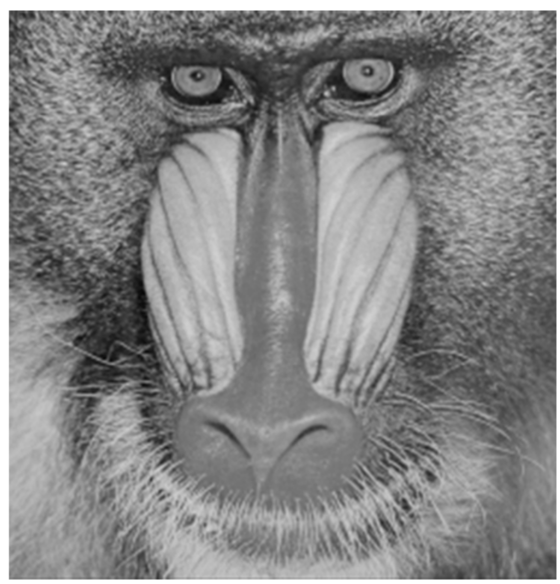
(a) Color-to-Grayscale Conversion (Basic: 12%)

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 average method simply averages the R, G, B values by Y = (R + G + B) / 3. The most popular one is the luminosity method [1], which is a weighted average to account for human perception. An example is given in Figure 1 which shows the color and the grayscale mandril images.
Convert the color Pepper image in Figure 2(a) to its gray scale image using the luminosity method with the following formula:

$$Y = 0.21\ R + 0.72\ G + 0.07\ B$$



(a) Color mandril image



(b) Gray-scale mandril image

Figure 1: Color and gray-scale mandril images.

(b) Image Resizing via Bilinear Interpolation (Advanced: 18%)

Use the bilinear interpolation to re-size the input color Mandril image of size 512x512 in Figure 1(a) to an output image of size 700x700. An image re-sizing example for the Pepper image is given in Figure 2.

(a) Pepper image of size 512x512 (b) resized pepper image of size 700x700

Figure 2: **Original and resized pepper images.**

Problem 2: Histogram Equalization and Image Filtering (40 %)

(a) Histogram Equalization (Basic: 12%)

Implement two histogram equalization techniques (the equalized "cumulative" histogram method and the equalized histogram method) to enhance the contrast of images in Figures 3 and 4. Histogram equalization can be performed separately on the red, green and blue channels for color images. Compare and comment on the output images and equalized histograms of these methods. Plot the histograms of all images (one input and 2 output images for both cases) and the corresponding transfer function.

Note that MATLAB users CANNOT use functions like imhist(), hist(), and other similar functions from the Image Processing Toolbox. The use of imshow() is however permitted.

Figure 3: Girl.raw

Figure 4: desk.raw

(b) Image Filtering – Creating Oil Painting Effect (Advanced: 14%)

An exemplary oil-painting effect on color image scene1_256 is shown in Figure 5. One can create an oil-painting style effect using the following two steps.

Step 1: Quantize all colors of the input color image, denoted by I_0, into an image containing only 256 colors, denoted by I_1.

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.

Perform the oil painting effect filter on images scene2_256.raw and Trojan_256.raw as shown in Figure 6 and Figure 7. Since both images contain 256 colors only, Step 1 in above can be omitted so that you can focus on Step 2. Try different values of *N*. Which *N* gives a better oil painting effect? Discuss your results.

Figure 5: scene1_256.raw and its oil painting effect

Figure 6: scene2_256.raw

Figure 7: Trojan_256.raw

(c) Image Filtering – Creating Film Special Effect (Advanced: 14%)

Design an algorithm to achieve the film special effect. An example is shown in Figure 8. Describe your algorithm step by step and apply your algorithm to test image chat.raw given in Figure 9. Show the film special effect result in your report.

Hint: Investigate the color channel relationship between the input and output images in Figure 8, which will give you ideas about the film special effect.

Figure 8: An exemplary film special effect: the input (left) and the output (right).

Figure 9: chat.raw

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

(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 10.

Figure 10: The original and noisy Lena images.

1) Identify noise types in the noisy Lena 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 like use to remove mixed noise?
   d) Can you cascade these filters in any order (justify your answer).
   e) Discuss the effect of different filter window sizes.
2) Get the best results in removing mixed noise and have the following in your report.
   a) Describe your method and show its results.
   b) Discuss its short-comings.
   c) Give some suggestions to improve its performance.

Use the PSNR (peak-signal-to-noise-ratio) quality metric to judge which method is better. 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 Image of size NxM

Y : Filterd Image of size NxM

Max : Maximum possible pixel intensity = 255

(b) Bilateral Filtering (Advanced: 15%)

In most low-pass linear filters, we often see degradation of edges. However, using some non-linear filters, we can preserve the edges. One example is the bilateral filter. A bilateral filter, in its most general form, can be given by [2]:

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

where $c(\xi,\mathbf{x})$ and $s(\mathbf{f}(\xi),\mathrm{f}(\mathbf{x}))$ 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 normalization factor. The discrete version of a bilateral filter is given by:

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

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

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

where $\|\cdot\|_2^2$ denotes the square 2-norm (the Euclidean 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_mixed.raw (shown in Figure 10) for each channel separately.

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 Ω_X).

3) Does this filter perform better than linear filters? Justify your answer in words.

4) The bilateral filter has an interesting property. That is, applying it to an image multiple times will result in a cartoon effect. Please verify this claim by applying it to the original Lena image.

(c) Non-Local Means (NLM) Filtering (Bonus: 10%)
The non-local-means (NLM) filtering technique [3] is a powerful denoising tool.
1) Describe and implement the NLM filter. DO NOT use any code from the Internet or other sources, as it would be considered as plagiarism.
2) Apply the NLM filter to Lena_mixed.raw (Figure 10). Try several filter parameters and discuss their effect on the filtering process. State your final choice of parameters in your report.
3) Compare the performance of NLM with those 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 [2, 3] or any other online sources. Explain in your own words. Reports and source codes are subject to verification for any plagiarism.

Appendix:

Problem 1: Simple Image Manipulation			
mandril.raw	512x512	24-bit	color(RGB)
Pepper.raw	512x512	24-bit	color(RGB)
Pepper_scale.raw	700x700	24-bit	color(RGB)
Problem 2: Histogram Equalization and Image Filtering			
Girl.raw	256x256	24-bit	color(RGB)
desk.raw	400x300	24-bit	color(RGB)
scene1_256.raw	410x365	24-bit	color(RGB)
scene1_oil.raw	410x365	24-bit	color(RGB)
scene2_256.raw	360x360	24-bit	color(RGB)
Trojan_256.raw	384x384	24-bit	color(RGB)
film.raw	1000x750	24-bit	color(RGB)
original.raw	1000x750	24-bit	color(RGB)
chat.raw	481x321	24-bit	color(RGB)
Problem 3: Noise Removal			
Lena.raw	512x512	24-bit	color(RGB)
Lena_mixed.raw	512x512	24-bit	color(RGB)

Reference Images

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

References

[1] John D. Cook. Three Algorithms for Converting Color to Grayscale. The Blog of John D. Cook, Aug. 24, 2009. http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/
[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *in Computer Vision, 1998*. Sixth International Conference on. IEEE, 1998, pp. 839–846.
[3] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2. IEEE, 2005, pp. 60–65.
[4] [Online] http://images.google.com/
[5] [Online] http://sipi.usc.edu/database/