EE 569: Homework #1

Issued: 8/26/2016 Due: 11:59PM, 9/18/2016

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: Basic Image Manipulation (30%)

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

(a) Cropping and Resizing (Basic: 12%)

Conduct the following operations to enlarge the face region of the *Anna* and *Rebel* images in Figure 1.

1. Face Cropping
   Write a program that crops the face area by entering two image coordinates (representing the top-left and the bottom-right corners of the output rectangle). For instance, if you choose two image coordinates, (55,85) and (184, 214), as the top-left and the bottom-right corners for *Anna*, respectively, the output will be her face image of size 130x130. Please show two cropped face images for Anna and Rebel. One is of the squared-shape and the other is of the rectangular shape.

2. Face Resizing
   Apply the bilinear interpolation to the cropped squared face images to three sizes: 100x100, 200x200 and 300x300. Show the results for both Anna and Rebel. Comment on the quality of the resized images and discuss some ideas for further quality improvements.

Figure 1: Anna and Rebel images

(b) 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. **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 *Clownfish* and *Octopus* 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.

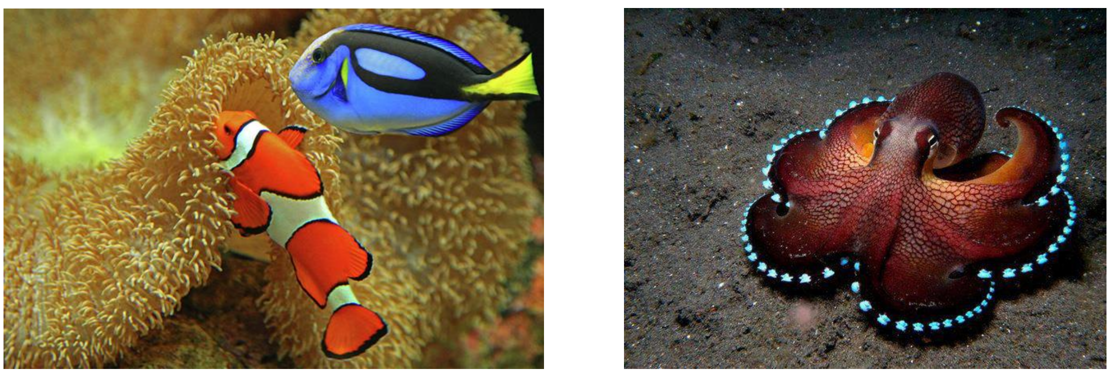

Figure 2: Clownfish and Octopus images

2. **HSL Color Space** – The Hue-Saturation-Lightness (HSL) color space is commonly used in computer graphics and image editing. Each of the HSL channels describes a certain property of the color image. Hue and Saturation describe the chromatic attributes while lightness defines the brightness of a pixel. HSL is a cylindrical-coordinate color model. Hue, an angle between 0 and 360 degrees, corresponds to the actual color of the pixel on the hue scale. Saturation, a percentage number between 0 and 100%, corresponds to the purity of the color. Lightness, a percentage between 0 and 100%, corresponds to the actual luminance of the pixel. Use the formulas below to write a function, which converts a RGB image to the HSL color space. Apply the formulas to *Turtle* and *Jellyfish* in Figure 3.

$$\begin{aligned} M &= \max(R, G, B) \\ m &= \min(R, G, B) \\ C &= M - m \end{aligned} \tag{2}$$

$$H = \begin{cases} 0 & C = 0 \\ 60\left(\dfrac{G-B}{C} \bmod 6\right) & M = R \\ 60\left(\dfrac{B-R}{C} + 2\right) & M = G \\ 60\left(\dfrac{R-G}{C} + 4\right) & M = B \end{cases}$$

$$L = \frac{M+m}{2} \tag{3}$$

$$S = \begin{cases} 0, \ L = 0 \\ \dfrac{C}{2L}, 0 < L < 0.5 \\ \dfrac{C}{2-2L}, otherwise \end{cases}$$

Write a program that generates 3 individual grayscale images for an input color image - one for each of the HSL channels. You should renormalize each of the channels to the range 0-255 for the display purpose. Show the results in the report, and comment on the property that each of the three channels attempt to capture from the input image.

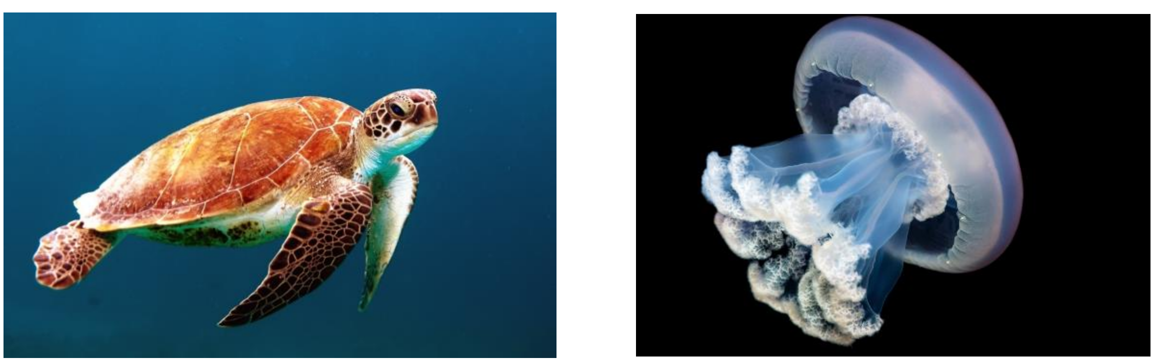

Figure 3: Turtle and Jellyfish images

Problem 2: Histogram Equalization (40 %)

(a) Histogram Equalization for Grayscale Images (Basic: 8%)

Implement two histogram equalization techniques:

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

Enhance the contrast of images *Beach_dark* and *Beach_bright* in Figure 4 using both methods. Describe the procedure and show final results. Please compare the performance with discussion.
Plot histograms of all images (2 input and 4 output images) and also their transfer functions (4 functions).

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

(a) Beach_dark **(b) Beach_bright**

Figure 4: Dark and bright Beach images.

(b) Histogram Equalization for Color Images (Basic: 8%)

Apply both histogram equalization techniques (methods A and B) to the *Skyline* image in Figure 5 for contrast enhancement. Histogram equalization can be performed separately on the red, green and blue channels for color images. Compare and comment on the input image and the histogram-equalized output images.

Figure 5: Skyline image

(c) Special Effect via Contrast Manipulation (Advanced: 10%)

The *Skyline* image produces a special effect of movie films in old days. Develop an algorithm to achieve such a special effect for any input image. Describe your algorithm in detail step by step. Implement your program and apply it to the *Trojan* and *Park* images in Figure 6. Show the results in your report. Discuss the color channel relationship between the *Skyline* image and your obtained results.

Figure 6: Trojan and Park images

(d) Histogram Transform (Advanced: 14%)

The histogram equalization is a special form of histogram matching. You can match the histogram of an input image to a pre-defined distribution. For example, we can make the histogram to match the Gaussian function with mean μ and standard deviation σ is given by

$$p\left(\frac{x}{\mu},\sigma\right)=\frac{1}{\sqrt{2\pi\sigma}}exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4}$$

where x is a channel intensity value. Implement an algorithm to match the histogram of *Student_1* and *Student_2* images in Figure 7 to the Gaussian distribution function in (4) with μ = 125 and σ = 40. We need to truncate the pdf outside [0,255] so that renormalization has to be performed. Propose and implement a renormalization procedure. Plot the histogram of the two images as shown in Figure 6. Show and discuss their resulting images after histogram transform.

Figure 7: Student_1 and Student_2.

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 8.


Figure 8: The original and noisy Lena images.

1. Identify noise types in the noisy peppers 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) Non-local means (NLM) filter (Advanced: 15%)

The non-local means (NLM) filtering technique is one of the state-of-the-art denoising tools. In this part, please read carefully the reference paper [1].

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_noisy.raw (Figure 8). Try several filter parameters and discuss their effect on the filtering process. State your final (best) choice of parameters in your report.
3. Repeat the same experiments for Part (a) and NLM filter with two other testing images (Figure 9): buildings_noisy.raw and trees_noisy.raw. Compare the performance of the NLM filter with those of filters used in part (a). Please explain what are the advantages of NLM filter 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.


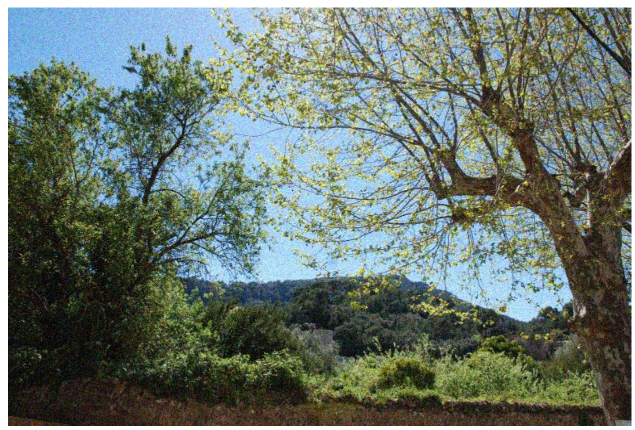

Figure 9: Noisy buildings and trees images.

(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 images: Lena, buildings and trees (Figure 8, 9). 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 [2]. Their code is written in MATLAB; so it is okay to use MATLAB for this part (You would still qualify for 5% bonus points if you have used C/C++ everywhere else).

2. Both the low-pass filters (LPF) and NLM filters are spatial domain filters. How would you classify BM3D - spatial domain, frequency domain, or both? Justify your answer.
3. Conduct qualitative performance comparison between the algorithms developed for Problem 3(b) and BM3D.

Appendix:

Problem 1: Simple Image Manipulation

Anna.raw	250x300	24-bit	color(RGB)
Rebel.raw	200x300	24-bit	color(RGB)
Clownfish.raw	600x400	24-bit	color(RGB)
Octopus.raw	500x374	24-bit	color(RGB)
Turtle.raw	400x250	24-bit	color(RGB)
Jellyfish.raw	500x333	24-bit	color(RGB)

Problem 2: Histogram Equalization

Beach_dark.raw	500x375	8-bit	gray
Beach_bright.raw	500x375	8-bit	gray
Skyline.raw	600x400	24-bit	color(RGB)
Trojan.raw	450x300	24-bit	color(RGB)
Park.raw	259x194	24-bit	color(RGB)
Student_1.raw	500x332	8-bit	gray
Student_2.raw	500x375	8-bit	gray

Problem 3: Noise Removal

Lena.raw	512x512	24-bit	color(RGB)
Lena_noisy.raw	512x512	24-bit	color(RGB)
Buildings.raw	512x512	24-bit	color(RGB)
Buildings_noisy.raw	512x512	24-bit	color(RGB)
Trees.raw	774x518	24-bit	color(RGB)
Trees_noisy.raw	774x518	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] 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.

[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/