EE 569: Homework #1

Issued: 1/13/2017 Due: 11:59PM, 2/5/2017

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) Mirroring, Resizing and Compositing (Basic: 15%)

Conduct the following operations to insert the dog on the beach use the *Dog* and *Beach* images in Figure 1.

1. Dog Mirroring
   Write a program that mirrors the *Dog* image across the vertical axis. Please show the mirrored dog image.

2. Dog Resizing
   Apply the bilinear interpolation to the mirrored *Dog* image to three sizes: 200x200, 400x400 and 600x600. Please show the results of the resized images and comment on the quality of the resized images and discuss some ideas for further quality improvements.

3. Image Compositing
   Replace the blue background in *Dog* image with the *Beach* image. As the background color may not be uniformly distributed, please set a color range for background removal and discuss how it will affect the quality of the dog, especially for the boundary. Discuss the challenges you encounter. Place the mirrored *Dog* image in the *Beach* image with its top-left corner at coordinates (1100, 400) of the beach image. Please show the final composite image.

Figure 1: Dog and Beach images.

(b) Color Space Transformation (Basic: 15%)
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 *Parrot* and *Building* 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: Parrot and Building 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 *Cat* and *Dolphin* 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: Cat and Dolphin images.

3. **Sepia Filter** – The Sepia toning technique was used in old days to give black-and-white photos a warm tone and a special visual effect. The Sepia filter is defined as

$$\begin{bmatrix} S_F(R) \\ S_F(G) \\ S_F(B) \end{bmatrix} = \begin{bmatrix} 0.393 & 0.769 & 0.189 \\ 0.349 & 0.686 & 0.168 \\ 0.272 & 0.534 & 0.131 \end{bmatrix} \begin{bmatrix} I \\ I \\ I \end{bmatrix} \quad (4)$$

Please convert the *Beach* image in Figure 1 to a gray level image I using the following formula. Then, apply the Sepia Filter to it. Show both the intermediate and final results.

$$I = 0.21\mathrm{R} + 0.72\mathrm{G} + 0.07\mathrm{B}$$

(c) Layer Blending Mode Implementation (Bonus: 5%)

Layer blending techniques, such as Screen, Darken, Lighten, Overlay, Hard Light, Soft Light etc. are widely used in photo editing software to generate special artistic effects. Please implement one of the layer blending modes using the two images in Figure 4. Please explain how the blending function can achieve the desired special effect. Please refer to the website https://en.wikipedia.org/wiki/Blend_modes for more details. You only need to implement ONE blending mode.

Figure 4: Top Layer (left) and Bottom Layer (right) images.

Problem 2: Histogram Equalization (40 %)

(a) Histogram Enhancement for Grayscale Images (Basic: 10%)

1. Implement two histogram enhancement methods
   - Method A: the linear transfer-function-based histogram enhancement method
   - Method B: the cumulative-probability-based histogram equalization method

Enhance the contrast of images *Tulip_dark* and *Tulip_bright* in Figure 5 using both methods. Describe the procedure and show final results. Please write down the transfer function in Method A. Please compare and discuss the performance differences. (You can use either CDF mapping or bucket filling implementation in Method B.)

Plot histograms of all images (2 input and 4 output images) and their transfer functions (4 functions).

2. Apply your implemented Method A and Method B to *Tulip_mix* in Figure 5 and show the result. Can you get similar result as obtained in Problem 2(a).1? If not, propose and test one way to modify your implementation? Please justify your proposed method.

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

(a) Tulip_dark **(b) Tulip_bright** **(c) Tulip_mix**

Figure 5: Dark and bright Beach images.

(b) Histogram Equalization for Color Images (Basic: 16%)

Apply both histogram equalization methods (methods A and B) to the *Bedroom* image in Figure 6 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. Can you propose another color image histogram equalization method? (Hint: How to preserve both hue and saturation histogram distribution?) Compare the results and discuss your observation.

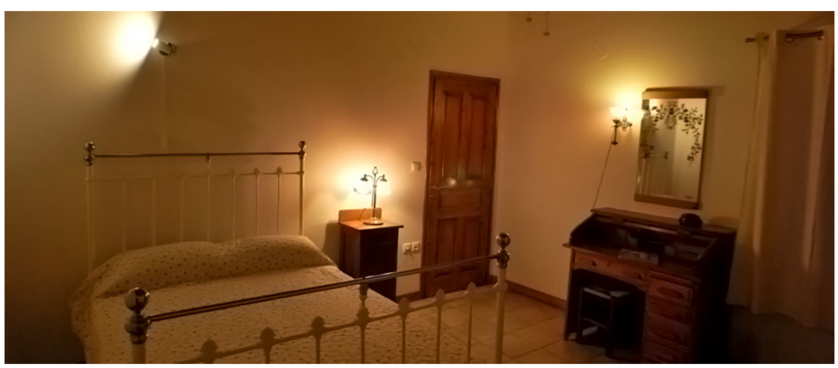

Figure 6: Bedroom image.

(c) 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 in form of

$$p\left(\frac{x}{\mu}, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{5}$$

where x is a channel intensity value. Implement an algorithm to match the histograms of *Forest_1* and *Forest_2* images in Figure 7 to the Gaussian distribution function in (4) with $\mu = 70$ and $\sigma = 20$. 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 red, green and blue channels of the two images as shown in Figure 7. Histogram transform can be performed separately on the red, green and blue channels for color images. Show and discuss their resulting images after histogram transform.

Figure 7: Forest_1 and Forest_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 Pepper images are shown in Figure 8.

Figure 8: The original and noisy Pepper 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 Pepper_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 one other testing image (Figure 9): sailboat_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 sailboat image.

(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: Pepper, and sailboat (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

Dog.raw	300x300	24-bit	color(RGB)
Beach.raw	1914x808	24-bit	color(RGB)
Parrot.raw	512x384	24-bit	color(RGB)
Building.raw	789x600	24-bit	color(RGB)
Cat.raw	600x398	24-bit	color(RGB)
Dolphin.raw	640x480	24-bit	color(RGB)
Top_layer.raw	890x900	24-bit	color(RGB)
Bottom_layer.raw	890x900	24-bit	color(RGB)

Problem 2: Histogram Equalization

Tulip_dark.raw	400x366	8-bit	gray
Tulip_bright.raw	400x366	8-bit	gray
Tulip_mix.raw	400x366	8-bit	gray
bedroom.raw	940x400	24-bit	color(RGB)
Forest_1.raw	960x600	24-bit	color(RGB)
Forest_2.raw	550x413	24-bit	color(RGB)

Problem 3: Noise Removal

Pepper.raw	512x512	24-bit	color(RGB)
Pepper_noisy.raw	512x512	24-bit	color(RGB)
Sailboat.raw	512x512	24-bit	color(RGB)
Sailboat_noisy.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] 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/