

UNIVERSITY OF CALIFORNIA, SANTA BARBARA

Department of Electrical and Computer Engineering

ECE 178

Image Processing

Fall 2020

Homework Assignment #1

(Due on Wednesday 10/14/2020 by 6 pm)

**Problem # 1.** Image and video data requirements:

a) Calculate the data size (in bits) of the following:

- $1024 \times 1024$  color (RGB) image with 8 bits per pixel per color channel
- $512 \times 1024 \times 64$  microscopy volume, single channel, with intensity quantization to 4096 gray levels
- $1920 \times 1080$  color (RGB) video at 30 frames per second, at 8 bits per pixel per channel, running for 10 seconds.

b) How long will it take to transmit the above video sequence using a mobile phone whose transmission speed is  $2.5 \times 10^5$  bits/s?

c) Finally, consider this video running continuously (not restricted to 10 seconds), what transmission speed in bits/s would be required of your transmission channel to enable you to communicate this video in real time (without delay)?

**Problem # 2.** Contrast stretching with paper and pencil:

Consider the following image:

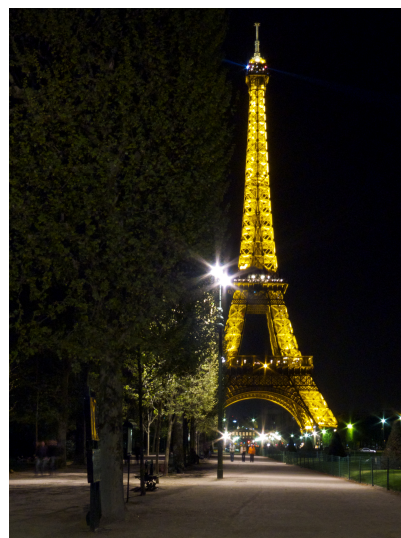
20	35	11
51	9	23
23	11	60

a) Find a linear pixel value transformation of the form  $g(r) = ar + b$  [or if you prefer the linear form  $g(r) = a(r - b)$ ], where  $r$  stands for the pixel value, such that when it is applied to all pixels in the image, the resulting image will have dynamic range of  $[0, 255]$ . Recall that this simply means that the lowest value is 0, and the highest is 255. Please explain your approach before jumping into “nitty gritty” calculations.

- b) Find the output image after applying the linear transformation.
- c) Suppose now that you wanted to apply a linear transformation that achieves full dynamic range as before, but at the same time creates a negative image (which means that the lowest value pixel before transformation is mapped to 255, and the highest pre-transformation value is mapped to 0.) Find this linear transformation. Again, first explain your approach before crunching numbers.
- d) This problem setting was contrived to avoid unnecessary trouble. In real life, which is not often as friendly as your ECE 178 homework, you will typically find that your approach results in non-integer values for the pixels. A natural solution would be to simply round the pixel values to the nearest integer. Explain why the overall transformation will then not be strictly linear, but only “approximately linear”.

**Problem # 3.** You were hired to install a fixed camera at appropriate distance from the base of the Eiffel tower in Paris, which is 300 meters tall. The camera lens has focal length of 20 mm and the custom made CCD sensor active area is 21 mm (height)  $\times$  16 mm (width), with  $4200 \times 3200$  pixels. (We assume “portrait” setting.)

- a) How far from the base of the tower do you need to position it so that the tower is about 3150 pixels tall on the array?
- b) The camera will also serve as a security camera, and given the pandemic, it has recently been tasked with verifying that social distancing of 2 meters is maintained by all visitors. This is done by checking the number of pixels of separation between persons standing at the base of the tower, as they appear on the sensor (i.e., on the image). How many pixels of separation would be required so as to not trigger the alarm?



**Problem # 4.** *A hands-on problem.*

Let us now revisit the problem of contrast stretching but with a real image. Consider the image posted with this homework: *peppers\_poor\_contrast.png*

- a) Stretching the image to full dynamic range: Write a program that finds and performs a linear transformation of the form discussed in Problem #2 (either  $g(r) = ar + b$ , or if you prefer the form  $g(r) = a(r - b)$ , where  $r$  stands for the pixel value) such that when it is applied to all pixels in the image, the resulting image will have dynamic range of  $[0, 255]$ . *Word of caution:* If working with matlab, images are loaded as uint8 by default when using *imread*. Cast the images to appropriate data type (e.g., double precision) before performing the contrast stretching, and then back to uint8 before saving the output image. Save and submit the resulting image in format: *YourName-HW1-P4A.png*
- b) Contrast stretching in the negative direction: apply another linear transformation to the original image but this time, the transformation should map the minimum pixel value in the image to 255 and maximum to 0. (Again, watch out for data type conversions when loading, transforming, and saving the image). Save and submit the resulting image in format: *YourName-HW1-P4B.png*
- c) Quantify the rounding error: referring to your response to Problem #2d, which concerned the need to round pixel values of contrast-stretched images to the nearest integer. For parts (a) and (b) of this problem, compute the root-mean-squared error between the images before and after this rounding operation.