CS 576 – Assignment 2
Instructor: Parag Havaldar

Assigned on Mon 09/30/24,
## Solutions due on Monday 10/14/24 by 5:00 pm

*Late Policy: **None**. Any assignment submitted after 5 pm will not be graded. If multiple assignments are submitted, the last one before the deadline will be graded.*

Theory Part (No need to submit, solutions to be provided)

Question1: Color Theory
One of uses of chromaticity diagrams is to find the gamut of colors given the primaries. It can also be used to find dominant and complementary colors –
Dominant color of a given color D (or dominant wavelength in a color D) is defined as the spectral color which can be mixed with white light in order to reproduce the desired D color. *Complementary colors* are those which when mixed in some proportion create the color white. Using these definitions and the understanding of the chromaticity diagram that you have, answer the following.
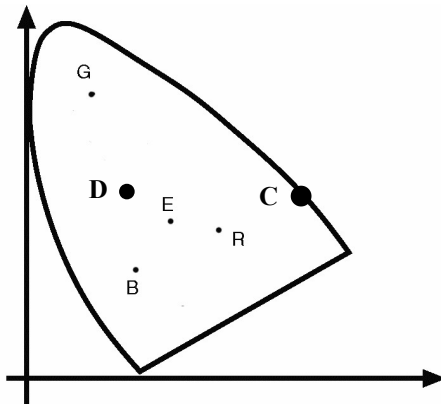
i. In the image alongside find the dominant wavelength of color D. Show this wavelength. *(2 points)*
ii. Do all colors have a dominant wavelength? Explain your reasoning. *(3 points)*
iii. Find the color which is complimentary to the color C. Show this color *(2 points)*
iv. What colors in RGB color space map to the equiluminous point E upon projection into the chromaticity space. *(3 points)*

Question 2: Color Theory
- The chromaticity diagram in (x, y) represents the normalized color matching functions X, Y and Z. Prove that *(2 points)*
$$Z = [(1-x-y)/y] Y$$
Here you are tasked with mapping the gamut of a printer to that of a color CRT monitor. Assume that gamuts are not the same, that is, there are colors in the printer's gamut that do not appear in the monitor's gamut and vice versa. So in order to print a

color seen on the monitor you choose the nearest color in the gamut of the printer. *(8 points)*

- Comment (giving reasons) whether this algorithm will work effectively? *(2 points)*
- You have two images – a cartoon image with constant color tones and a real image with varying color tones? Which image will this algorithm perform better – give reasons. *(2 points)*
- Can you suggest improvements rather than just choosing the nearest color? *(4 points)*

Question 3: Entropy Coding
Consider a communication system that gives out only two symbols X and Y. Assume that the parameterization followed by the probabilities are $P(X) = x^k$ and $P(Y) = (1-x^k)$

- Write down the entropy function and plot it as a function of x for k=2. *(1 points)*
- From your plot, for what value of x with $k=2$ does H become a minimum? *(1 points)*
- Your plot visually gives you the minimum value of x for $k=2$, find out a generalized formula for x in terms of k for which H is a minimum *(3 points)*.
- From your plot, for what value of x with $k=2$ does H become a maximum? *(1 points)*
- Your plot visually gives you the maximum value of x for $k=2$, find out a generalized formula for x in terms of k for which H is a maximum *(4 points)*.

Programming Part (*100 points*)
This assignment will help you gain a practical understanding of analyzing color channels especially as it pertains to image segmentation. Image segmentation is a challenging task in the field of image processing and computer vision. In order to obtain an accurate segmentation performance, user interaction is always used in practical image-segmentation applications. Here you will implement one such application where the image pixels need to be segmented based on two threshold values in the HSV color space, using the hue values as thresholds. All the pixels falling between these two hue thresholds will be displayed in the original color in the output image, whereas all the other pixels outside the threshold will be displayed in gray.

You will be given input images in the usual rgb format. You are free to use extend the display code sample given of assignment 1 to implement this project or you may write your own in any language of your choice (*no scripting languages such as MATLAB or python please!*).

Recall that the HSV color space (Hue, Saturation and Value) describes Hue as a number between 0 and 360, 0 being closer to red
The input to your program will take three parameters where

- The first parameter is the name of the image, which will be provided in an 8 bit per channel RGB format (Total 24 bits per pixel). You may assume that all images will be of the same size for this assignment of size 512 x 512
- The second parameter will be a number between 0-360. This will provide the first hue threshold **h1** for deciding your segmentation boundary.
- The third parameter will be a number between 0-360. This will provide the second hue threshold **h2** for deciding your segmentation boundary. This number will be greater than the previous number.

To invoke your program we will compile it and run it at the command line as

YourProgram.exe C:/myDir/myImage.rgb h1 h2

where *h1* and *h2* are the parameters as described above. To run your program you will need to convert your rgb values Example inputs are shown below and this should give you a fair idea about what your input parameters do and how your program will be tested.
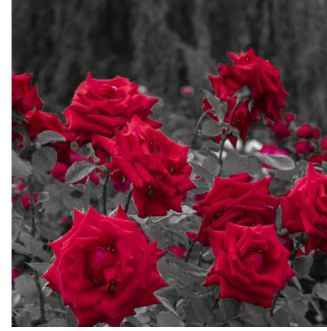
1. *YourProgram.exe roses_image1.rgb 0 359*

The left image is your input and the right image is the produced output. Here the thresholds h1 and h2 scan through all colors 0 to 359, so the output will be same as the input.
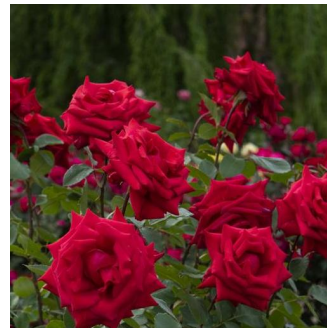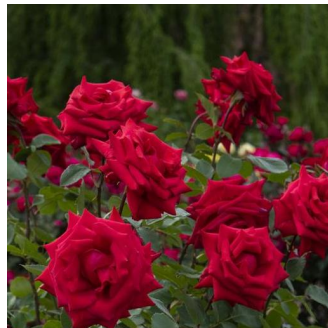

2. *YourProgram.exe roses_image1.rgb 320 359*

The left image is your input and the right image is the produced output. Here the thresholds h1 and h2 scan through 320 to 359 which is the red part of the HSV space. Correspondingly the output is shown below.
 input.

1. *YourProgram.exe roses_image1.rgb 60 120*

The left image is your input and the right image is the produced output. Here the thresholds h1 and h2 scan through 160 to 120 which falls in the green spectrum of the HSV space. Correspondingly the output is shown below.
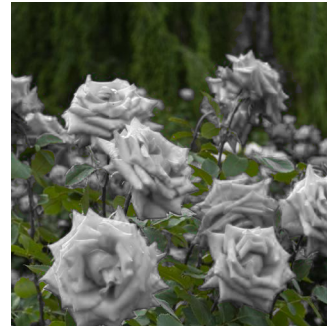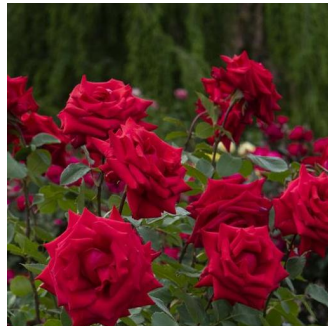
Now for the details - You will need to convert your image from an RGB representation to a HSV representation. Unlike other color space conversions, there is no matrix to linearly convert to the HSV color space. The HSV color space is represented as a cylindrical coordinate system with H varying from 0 (near violet/blue) to 360 (red), where as H and V both vary from 0 to 1. You can read about the color conversion here:
https://en.wikipedia.org/wiki/HSL_and_HSV
https://www.cs.rit.edu/~ncs/color/

Once in the HSV space, you can appropriately use the thresholds h1 and h2 to perform the assignment.

What should you submit?
- Your source code, and your project file or makefile, if any, using the submit program. ***Please do not submit any binaries or images***. We will compile your program and execute our tests accordingly.
- If you need to include a readme.txt file with any special instructions on compilation, that is fine too.