CSE 273: Computational Photography, Spring 2023 – Assignment 2

Instructor: Ben Ochoa

Due: Wednesday, April 19, 2023, 11:59 PM

Instructions

- Review the academic integrity and collaboration policies on the course website.
- This assignment must be completed individually.
- This assignment may be completed in the programming language of your choice.
- You may use third party libraries/packages for basic linear algebra, basic image processing, and image file I/O. But, you may not use third party libraries/packages that directly solve the problem. If you are uncertain about using a specific library/package, then please ask the instructional staff whether or not it is allowable.
- You must prepare a report as a pdf file. The report must describe the problems, and your solutions and results. Math must be done in Markdown/LaTeX.
- Additionally, you must create a zip file containing all of your source code, along with an automated build method (e.g., a makefile) and a `readme` file with clear and concise directions on how to build and execute your program.
- The zip file must also contain any output image files.
- You must submit both files (.pdf and .zip) on Gradescope. You must mark each problem on Gradescope in the pdf.
- It is highly recommended that you begin working on this assignment early.

Problems

When completing these problems, refer to the spectral data at
`https://files.cie.co.at/204.xls`

1. (5 points) Develop a function/method named `spd_D65` that outputs the relative spectral power distribution (SPD) function of CIE standard illuminant D65 at a given wavelength λ in nanometers. In your report, include a plot of the SPD function over the wavelengths 300 nm–830 nm. Develop a function/method named `colorMatchingFunctions` that outputs the values of the CIE 1931 standard colorimetric observer color matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ at a given wavelength λ in nanometers. In your report, include a plot of the color matching functions (on a single plot) over the wavelengths 380 nm–780 nm.

2. (15 points) Develop a function/method named `transformationMatrixHyperspectralToXYZ_D65` that calculates the 3-by-n color space transformation matrix $\mathtt{M}_{\text{HSIToXYZ}}$ that maps from hyperspectral to XYZ with CIE standard illuminant D65 using the CIE color matching functions, given the n wavelengths

(in nanometers) associated with the n channels of a hyperspectral image. Develop a function/method named `transformationMatrixRGBToXYZ` that calculates the 3-by-3 transformation matrix that maps linear RGB values in a color space with a specified set of chromaticities to XYZ (hint: reuse code from assignment 1). Using `transformationMatrixRGBToXYZ`, calculate the transformation matrix $\mathtt{M}_{\mathrm{RGBToXYZ}}$ that maps from sRGB chromaticities to XYZ. Calculate the 3-by-31 transformation matrix $\mathtt{M}_{\mathrm{HSIToRGB}} = \mathtt{M}_{\mathrm{RGBToXYZ}}^{-1}\mathtt{M}_{\mathrm{HSIToXYZ}}$ for the wavelengths 410 nm–710 nm, inclusive, at 10 nm increments. Include the numerical values of the 31-by-3 matrix $\mathtt{M}_{\mathrm{HSIToRGB}}^{\top}$ in your report with sufficient precision such that it can be evaluated.

3. (15 points) Using $\mathtt{M}_{\mathrm{HSIToRGB}}$, map the hyperspectral images contained in `scene4.zip`, `scene5.zip`, and `scene7.zip` to linear RGB color space 32 bit floating-point per sample images with sRGB chromaticities. Write the resulting linear RGB color space 32 bit floating-point per sample, 3 channel image directly to the OpenEXR files `scene4_rgb.exr`, `scene5_rgb.exr`, and `scene7_rgb.exr`, respectively. Note unless chromaticities are explicitly set when writing the OpenEXR file, sRGB chromaticities are implicitly set. Develop a function/method named `linearTosRGB` that converts a linear RGB color space 32 bit floating-point per sample image with sRGB chromaticities to a nonlinear sRGB color encoded 8 or 16 bit unsigned integer per sample image. Using `linearTosRGB`, convert the linear RGB color space 32 bit floating-point per sample, 3 channel images with sRGB chromaticities to a nonlinear sRGB color encoded 8 bit unsigned integer per sample, 3 channel images. Write the resulting nonlinear sRGB color encoded 8 bit unsigned integer per sample, 3 channel images directly to the PNG files `scene4_rgb.png`, `scene5_rgb.png`, and `scene7_rgb.png`, respectively. Briefly discuss your results.