Universität Tübingen
Prof. Dr.-Ing. Hendrik P.A. Lensch
Lehrstuhl Computergrafik
Dennis Bukenberger (dennis.bukenberger@uni-tuebingen.de)
Zabiha Kehribar (zabiha.kehribar@student.uni-tuebingen.de)

11. January 2021

Graphische Datenverarbeitung
Assignment 8

Submission deadline for the exercises: 18. January 2021 6.00 am

Source Code Solutions

- Upload **only** the source code files listed in the description in Ilias.
- Upload them one by one and **don't** zip them.

The source code must run on `cgpool120[0-7].informatik.uni-tuebingen.de` by extracting your submitted `.tar.gz` file to a certain folder and running `scons`. You can log onto this machine via ssh by using your WSI account. From outside the WSI network, you may have to use a ssh gateway, e.g. `cgcontact.informatik.uni-tuebingen.de`
Attention: Do not use `cgcontact` for working, but only for ssh-ing to the `cgpool` machines.

The framework you get for completion already compiles and runs as requested above and you only have to modify source and header files - no files have to be created.

8.1 CIE XYZ images from spectral data (20 + 20 + 10 = 50 Points)

In this exercise, you will convert spectrally acquired photographs to the CIE Labs XYZ color space. The `fileio.*` files have been extended with a `load_image_pfm(...)` function which allows you to load the supplied set of high dynamic range (HDR) `pfm` files in the `spectral_roof` folder. Further, a `save_image_pfm(...)` function allows you you store your result as HDR image file.

Download the tables for the CIE XYZ color matching functions (CMF) from `http://cvrl.ioo.ucl.ac.uk/index.htm`. Select the *CIE physiologically-relevant LMS functions* and download the *Cone Fundamentals* (2-deg) with linear energy units and 5 nm stepsize. The csv format should be the easiest to understand and to import to your program (or to convert to C++ source code).
The downloaded file contains the response of the three types of human photoreceptors on certain wavelengths.

Your task is to generate a `pfm` file in CIE XYZ color space from the given `env_roof_l???.pfm` image files:

a) Load the downloaded CMF data or convert it to source code to be able to access it from your program somehow.

b) Load all `env_roof_l???.pfm` files and add the response that is caused by them to your result image.

c) Make your program saving the resulting image.

Hints: The input `pfm` files contain RGB values $r = g = b = L(\lambda)$ at each pixel.
You can display `pfm` files on linux using `pfsin image.pfm | pfsglview` from `pfstools`. On other OSes, search for an HDR image viewer that supports the PFM format.
Feel free to hard code wavelength loop boundaries, image sizes and so on.
You might want to visit http://www.cplusplus.com and have a look at

- `map<..., ...>` to organize the mapping from wavelengths to CIE XYZ responses.
- `sprintf(...)` for assembling filenames.
- `fscanf(...)` for reading formatted data from text files.

8.2 Supersampling 2 (20 + 20 + 10 = 50 Points)

A pixel actually corresponds to a square area. To calculate its value means to integrate over this area. It is not possible to solve these integrals analytically, so they have to be approximated numerically. This is done by computing weighted means of samples of the integrand. The actual image is reconstructed by sampling the integrand on selected points.
The placement of the sample points greatly influences the result. The approximation error may result in random noise or in aliasing artifacts in the image.

For this exercise implement the sampling strategies and test them with the given test functions.

a) Evaluate the given test functions:

$$Z : [0, 1)^2 \to [0, 1]$$

$$(x, y) \to \frac{1}{2}(1 + (1 - y)^3 \cdot sin[2\pi x e^{10x}]) \tag{1}$$

$$(x, y) \to \frac{1}{2}(1 + sin(1600 \cdot (x^2 + y^2))) \tag{2}$$

$$(x, y) \to \frac{1}{2}(1 + sin(60 \cdot 4\pi \cdot arctan\frac{x}{y})) \tag{3}$$

b) Place samples according to the following strategies:

- **Regular Sampling**: The pixel is subdivided into $n = m \times m$ equally sized regions, which are sampled in the middle:

$$(x, y) = \left(\frac{i + \frac{1}{2}}{m}, \frac{j + \frac{1}{2}}{m}\right)_{i,j=0}^{m-1}$$

- **Random Sampling**: The pixel is sampled by n randomly placed samples:

$$(x, y) = (\xi_{i,1}, \xi_{i,2})_{i=0}^{n-1}$$

  where $\xi_i \in_r [0, 1)$

- **Stratified Sampling**: This is a combination of regular and random sampling. One sample is randomly placed in each of the $n = m \times m$ regions with $\xi_i, \xi_j \in_r [0, 1)$:

$$(x, y) = \left(\frac{i + \xi_i}{m}, \frac{j + \xi_j}{m}\right)_{i,j=0}^{m-1}$$

c) Create some slides with the generated images and show, why stratified sampling is superior to regular sampling and why is it superior to random sampling. Submit the slides as PDF.

Hint: The (x,y) in b) is the position of the sample in the pixel region. This means you have to offset each pixel in the image with this values and than map to $[0, 1]$ and evaluate the given functions.