Report Labwork 3

Antoine Maffeis

December 2022

1 Introduction

Grayscale images are monochromatic images that use shades of gray to represent different intensities of light. Grayscale images are commonly used in fields such as computer vision and image processing, as they are often easier to analyze and process than color images.

In this report, we present two simple functions for converting an image to grayscale using Python : one is a GPU-accelerated function and the other one is using the CPU. The function using CPU is implemented from scratch and does not rely on any external libraries or dependencies. The function using GPU is implemented from scratch but rely on CUDA (Numba).

2 Methodology

The grayscale conversion function takes an image as input and checks if it is already grayscale (i.e., it has only one channel). If it is already grayscale, the function returns the image as is. If the image has three channels (i.e., it is a color image), the function converts it to grayscale using the luminosity formula, which weights the different color channels differently to better approximate the way the human eye perceives color.

The luminosity formula used in the function is as follows:

$$gray = 0.3 * R + 0.72 * G + 0.07 * B$$

Where R, G, and B are the red, green, and blue channels of the image, respectively, and $gray$ is the grayscale value of the pixel.

3 CPU Function

The CPU function grayscale conversion function convert the image using the luminosity formula explained in the Methodology seciton

4 GPU-accelerated function

The GPU-accelerated grayscale conversion function consists of two parts: a CUDA kernel called ‘gpu_grayscale_kernel‘ that will convert the image using the luminosity formula , and a wrapper function called ‘gpu_grayscale‘.