

Reading and writing images for Project 2

Image files

Images are stored in files. There are multiple file formats in which images can be stored.

A common format in 2023 is PNG. Other formats you might encounter are BMP, JPEG, and GIF.

An image is a collection of pixels. Usually, each pixel would be represented as an RGB (red, green, blue) tuple, where the intensity of each of red, green, and blue is a number between 0 and 255. An RGB value of (0, 0, 0) means all the intensities are zero. That means the pixel is black. An RGB value of (255, 255, 255) means all the intensities are at their maximum. That corresponds to a white pixel. If all the intensities are approximately equal, the pixel will appear as a shade of gray, somewhere between completely black (0, 0, 0) and completely white (255, 255, 255). An RGB value of (255, 0, 0) corresponds to an intense red. More on Wikipedia, if you are interested https://en.wikipedia.org/wiki/RGB_color_model

Image files and C

There is not a standard way to read in common image formats in C. For that reason, we provided you `c_img.c`, which can read and write image files in a simple format. The simple format is named `bin`.

We also provide you with `png2bin.py`, which allows you to read, write, and display images in both formats such as PNG and BMP and in the custom `bin` format used by `c_img.c`.

What you need and don't need for Lab 7

- Need to be able to read a `bin` image using `c_img.c`, then modify the image using `get_pixel` and `set_pixel` from `c_img.c`, then write the modified image in `bin` format using `c_img.c`
- Need to understand the `rgb_img` struct in `c_img.h`

What you don't need for Lab 7

- Don't need to understand the `bin` format or how `c_img.c` writes and reads files
- Don't need to understand anything about image gradients
- Don't need to be able to read in a `bin` image written by `c_img.c` and display it using `png2bin.py`
- Don't need to be able to read and display PNG images using `png2bin.py`
- Don't need to be able to read a PNG image and write it to `bin` format using `png2bin.py`