CENG 391 – Introduction to Image Understanding
Homework 1

November 6, 2021

Due Date: November 18, 2021

Download and extract the contents of `ceng391_02T_image_formation.tar.gz`.

Exercise 1 Color Support for the `ceng391::Image` Class

- Add an integer member field `m_n_channels` and a corresponding parameter to the constructor that stores the number of channels per pixel. The constructor should only accept the values 1, 3, or 4 for this parameter (grayscale, RGB, or RGBA data) and it should terminate the program with an error message if a different value is given.
- Write RGB and RGBA versions for the `set` and `set_rect` functions that takes more channels instead of a single value. Also modify the existing version to work with RGB and RGBA images (it should copy the given `value` to RGB channels and assume a default alpha value of 255).
- Modify the contents of the `Image::write_pnm` function so that instead of writing an error message for RGB or RGBA images, it saves the image contents in the binary PPM format. *Hint:* You should only write the RGB values into the file. The alpha values should not be saved. This means you will sometimes need two loops instead of one.

Exercise 2 Loading PNM Images

Write a new member function `Image::read_pnm` that takes a `std::string` argument named `filename`. The function should try to open the file with the given name and read its contents if its contents are in the PGM or PPM binary formats. When reading color images, remember to create a four channel RGBA image. You should read the RGB values from the file and initialize all alpha values to 255.

Exercise 3 Color Conversion

Write three new member functions `Image::to_rgb`, `Image::to_grayscale`, and `Image::to_rgba`, that converts the image contents from current number of channels to the desired format. If the image is already of the target format the functions should return immediately.

To convert from grayscale and RGB to RGBA, just copy the gray or color value to all the color channels and set alpha to 255. To convert from RGB and RGBA to grayscale, you can use the following set of formulas:

$$I_{Gray}(x, y) = I_{Red}(x, y) * 0.3 + I_{Green}(x, y) * 0.59 + I_{Blue}(x, y) * 0.11;$$

Before writing back the grayscale value, you should check for underflow and overflow and you should discard the alpha values.