Computer Science
&
Data Structure and Algorithms 2

UFAZ DSA2 2021-22 mini-project

Professors:
Prof: Pierre Collet

Students:
Nicat Aghali
Khumar Huseynova
Arif Ahmadli

1 Introduction

The main goal of this project is to develop a program that allows us to automatically adjust the brightness and contrast of a BMP image. There are several steps so that required results can be obtained. In order to change the brightness of the image, we need to subtract the minimum values of all RGB pixels. Moving to maximum contrast, in order to obtain that, we need to multiply all the pixels by maximizing factor. (*Maximizing factors can be obtained by dividing max. brightness by max. RGB values.)

2 What is BPM image?

Files having extension .BMP represent Bitmap Image files that are used to store bitmap digital images, and they are also called device independent bitmap (DIB) file format. The BMP file format can store data as two-dimensional digital images in both monochrome as well as color format with various colour depths. BPM images have two types of headers:

* **BMP header** - 14 bytes - To store general information about the bitmap image file.

* **DIB header** - 40 bytes - To store detailed information about the bitmap image and define the pixel format

3 Implementation

To manipulate BMP images, following functions are written.

loadBMP - read the bytes which are distributed accordingly:

reading BMP Header...

- 2 Bytes - 'BM' file format;
- 4 Bytes - The size of the file;
- 4 Bytes - reserved for application(unused);
- 4 Bytes - the offset which tells at which byte the image starts;

reading DIB Header...

- 4 Bytes for size of the DIB header;
- 4 Bytes for image width;
- 4 Bytes for image height;
- 2 Bytes for number of color planes;
- 2 Bytes for number of bits per pixels;
- and the remaining bytes which are not important;

Then there exists several loops for finding minimum value, applying brightness and contrast rules.

4 User manual

So finally moving to exact usage of the application. In order to visually represent resulting image there are several steps of compiling which must be conducted.

User has to compile autoadjust.c file in following matter:

- gcc autoadjust.c -o autoadjust

In order to get help about the usage of the application and correct syntax of the input user can type following command:

- ./autoadjust -h

Program automatically checks the input for its correctness and prints error messages if something does not satisfy the requirements.

So the proper way of running is:

- ./autoadjust source.bmp > destination.bmp

or by using "-o" flag

- ./autoadjust source.bmp -o destination.bmp

5 Analyzing result

Example 1

By looking at the resulting image, we can observe that nothing

happened to it. So we can say that that lowest and highest RGB values of the pixels are already equal to valules of pixels in brightness and contrast.

Example 2 So here is another example of application output.As we can see image became more contrast and less bright.

Example 3 And here is the representation of how the application works on colored images.