CSB100: Project-I

8-BIMP: Bitmap Image Processor

Submitted By:

Sunke Manikanta - 221210103

Tanishq Kumar Toliya - 221210107

Tanmay Mittal - 221210108

Vaibhav Goel – 221210116

Branch: CSE 2

Semester: 2nd Sem

Group: 2

Submitted To: Dr. Chandra Prakash

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2023

TABLE OF CONTENTS

INTRODUCTION

The purpose of the report is to document the work of project focused on image processing. The project involved the development of functions to perform various operations on images, including copying of an image, creating a negative image, and rotating an image. These functions were implemented using appropriate image processing techniques and algorithms. The project was carried out without the use of external libraries, using built in functions and algorithms available in c language.

In this project, our main objective was to apply different changes and processes to an image. Image processing is a very important task performed on a daily basis to almost all the images for different purposes like copying, rotating, etc. In the project, we had to solve this problem using c language.

MOTIVATION

Image processing plays a crucial role in various fields, including photography, editing, computer vision, and medical imaging. The ability to manipulate images programmatically provides flexibility and opens up possibilities for enhancing, analysing, and transforming visual data. The motivation behind developing this program was to provide a simple yet effective tool for performing basic image processing operations using the c programming language.

There are several motivations to learn image processing. Some key reasons are:

1. Practical applications: image processing is widely used technology with numerous practical applications in various fields like
   Healthcare, entertainment, surveillance, etc.

2. Career opportunities: data in any form is in huge demand and image is one of the most popular form of data and hence processing of image creates a large number of opportunities.

3. Advancements in technology: with the technology in a rise, newer technology such as artificial intelligence,

machine learning are emerging and these technologies rely heavily on image processing techniques.

4. Research and innovation: image processing is a vibrant field with many ongoing research and innovation. By learning image processing, we can contribute to advancements in the field, explore new algorithms, develop new techniques, and solve challenging problems.

5. Creative expression: image processing gives a boost to the creative mindset of the developer and hence also helps in personal growth. Learning image processing can be intellectually stimulating and personally rewarding.

METHODOLOGY

ENVIRONMENT AND TOOLS

The project was implemented using c programming language without the use of external libraries.

The choice of editor was Visual studio on windows.

- **BITMAP IMAGE**

In the project, we have used bitmap image format for every function since it provides features such as-

1. High raw data

2. Lossless compression

3. structured storage of data

Bitmap image has the bare minimum information that one need in order to get started with image processing programs using c.

An image is structured in the following way:

Image header-> colour table (if any)-> image data

Similarly, the bmp image is structured in the same way. It has a 54-byte image header, 1024-byte colour table if present, and the rest is the image data.

IMAGE HEADER :

We extract the required information from the image header,

Width: it is stored in the 18 bit.

Height: it is stored in the 22 bit.

Bitdepth: it is stored in the 28 bit.

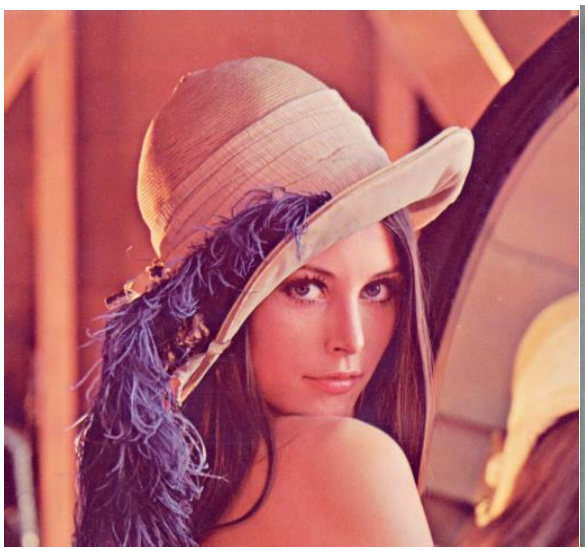
Then we read the colour table, and allocate the block of memory as per the image size, to read the image information.

Gray scale image:

For this project, we are using gray scale image for the function. In a Gray scale image, the pixel value would be between 0-255. For example, to find the complement, we just subtract the read image value from 255. The basic equation is

New_pixel_data = 255 – old_pixel_data;

We are using Lena image for displaying of the output

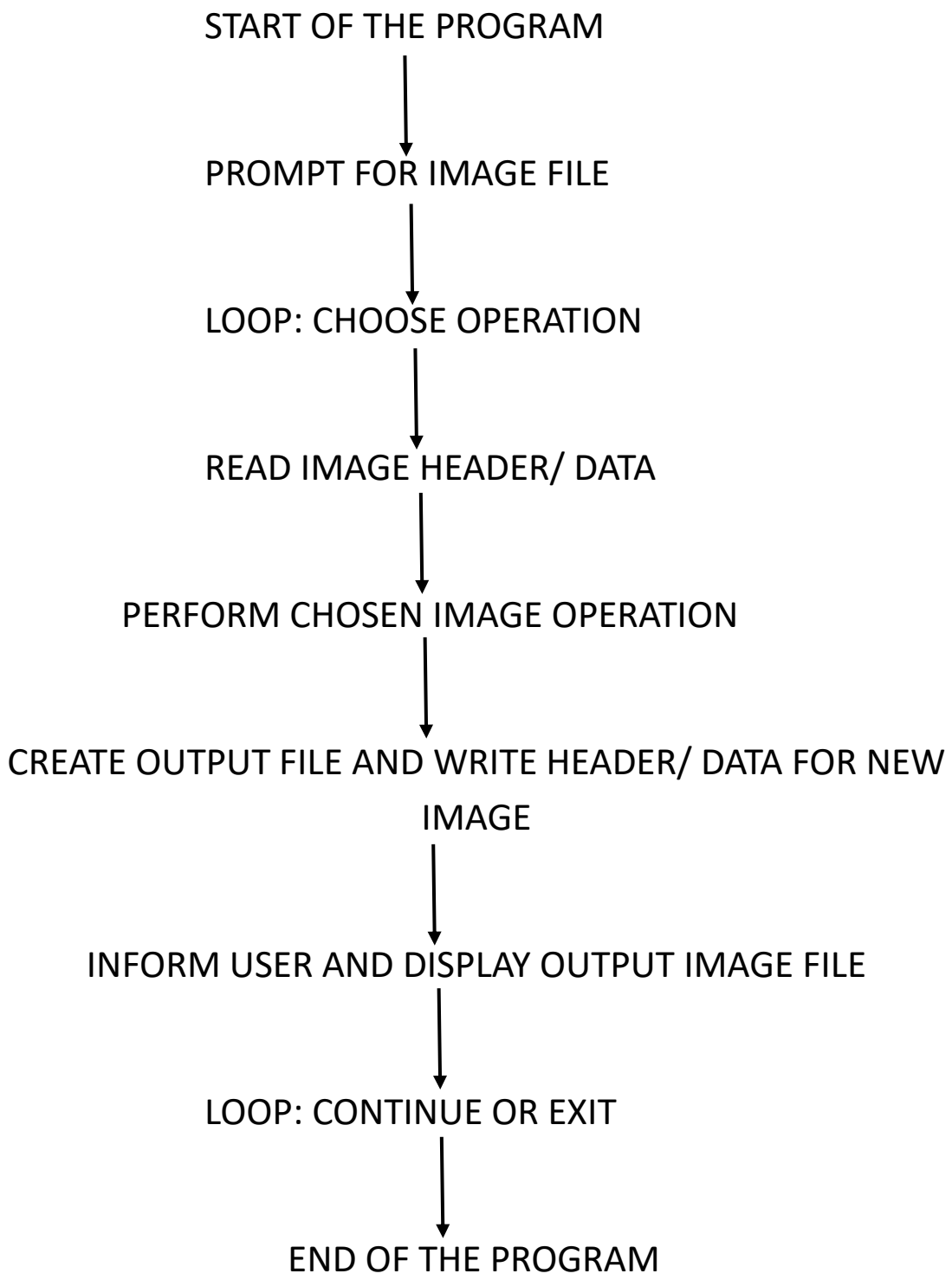


Coloured image

grayscale image

FLOWCHART

THE FLOWCHART BELOW ILLUSTRATES THE FLOW OF THE PROGRAM:



NEGATIVE TRANSFORMATION FUNCTION

The negative transformation function creates a negative version of an input image.

input image

output image

In the code below, using the nested for loop, we increment the value of height and the width

And we subtract old image data from 255 to obtain the negative image.

```
for (i = 0; i < height; i++)
{
    for (j = 0; j < width; j++)
    {
        newimageData[i * width + j] = 255 - imageData[i * width + j];
    }
}
```

IMAGE ROTATION CLOCKWISE FUNCTION

The image rotation function rotates an input image by 90 degrees angle in the clockwise direction.

input image output image

In the code below, using the nested for loop, we increment the height and the width to rotate the image clockwise.

```
for (i = 0; i < height; i++)//to rotate image 90 degree CLOCKWISE
{
    for (j = 0; j < width; j++)
    {
        newimageData[i * width + j] = imageData[j * width - i];
    }
}
```

IMAGE ROTATION ANTICLOCKWISE FUNCTION

The image rotation function rotates an input image by 90 degrees angle in the anticlockwise direction.

Input image output image

In the code below, using the nested for loop, we increment the height and the width but here there is a difference in the addition and subtraction of incremented variables causing differences in data stored in the new image.

```
for (i = 0; i < height; i++)// to rotate the image ANTICLOCKWISE
{
    for (j = 0; j < width; j++)
    {
        newimageData[i * width - j] = imageData[j * width + i];
    }
}
```

IMAGE FLIP FUNCTION OR MIRROR IMAGE FUNCTION

In this function, we mainly output the mirror image of the input image.

Input image output image

In the code below, using the nested for loop, we increment the height and the width and we only change the data stored in width because we are looking for a mirror image.

```
for(i = 0; i < height; i++){
  for(j = 0; j < width; j++){
     newimageData[i*width + j] = imageData[i*width - j];
   }
}
```

BRIGHTNESS INCREASE FUNCTION

By this function, we increase the brightness of the image.

Input image

output image

In the code below, the user is asked to input the value of x between -30 to +30. If the user enters a positive value, the brightness of the image increases.

```
for (i = 0; i < height; i++)
{
    for (j = 0; j < width; j++)
    {
        newimageData[i * width + j] = imageData[i * width + j]+x;
    }
}
```

DARK FUNCTION

By this function, we darken the background of the image.

Input image output image

The code for darkening of the image is similar to that of the one used for increasing the brightness of the image. The only difference is the input value of x. If the input value of x is negative, the image gets darken.

```
for (i = 0; i < height; i++)
{
    for (j = 0; j < width; j++)
    {
        newimageData[i * width + j] = imageData[i * width + j]+x;
    }
}
```

ROTATE 180 FUNCTION

In this function, we rotate the image by 180 degrees.

Input image output image

In the code below, using the nested for loop, we performed the rotate by 90 two times to achieve the result. It required opening of two files

RESULTS/ WORKING

This project allows users to choose from a set of image processing operations. These operations include turning the image negative, rotating the image 90 degrees clockwise, anticlockwise, even 180 degrees rotation, adjusting the brightness, and mirroring/ flipping the image. The program reads an input image file, performs the selected operation, and creates an output file with the modified image. The user is then informed about the creation of the output image, and the file is displayed using the system command. This provides users with a visual representation of the processed image.

REFERENCES:

During the development of this program, the following references were consulted:

1. https://abhijitnathwani.github.io/blog/2017/12/21/Negative-Image-using-C
2. Image processing in C, Dwayne Phillips

CONCLUSION

In conclusion, this report discussed a c program for image processing, covering its introduction, motivation, tools used, flow chart, results/working, and references. The program provides users with a set of image processing operations and allows them to modify image files using the selected operations.

We have also combined all the operations in a single code so that all operations can be performed by compiling the code once. Also we were able to display the image after the operation.

The main objective of the project was to perform different image processing operations like copying, creating a negative version of the image, rotating, flipping, changing the brightness of the image which has been accomplished in the project.

EXPECTED ERRORS

1) The User may try to run the program on a non bitmap image format.
2) The user may use a different bit depth bmp (e.g. 24-bit BMP)
3) The user may use a colour BMP instead of grayscale BMP, in that case, the functions that perform the manipulation of orientation, dimensions etc will work but the brightness control functions may produce unexpected outputs.