Manipulating Bitmap (BMP) images

Kanan Mikayilov
Shoykyat Sharafyabi

Computer Science L2
French-Azerbaijani University

January 4, 2020

Contents

1 Introduction

Nowadays, digital signatures are more and more used to sign documents. This is problematic because anyone can copy a signature from a document to paste it on a new document, so digital signatures are not meaningful anymore. The aim of this project is to have you create a program that will take a signature stored in BMP format and modify it to “watermark” the signature.

2 User Manual

2.1 How to compile

The program can be compiled by writing simple command in the terminal in the directory of program:

$ make

It will compile the program, but to run the program, you should close and reopen the terminal

2.2 How to launch

As told in the ”How to compile” part, after you compile the program, you can launch it, there are several methods to do it:

$ wm inputfile -text some text -date -color some color -pos x,y -o outputfile

Example:

$ wm example.bmp -text Hello -date -color FF3245 -pos 20,10 -o modsign.bmp

If you do not want to write any text, just date, you can skip it. You can skip also date. In this case initial file will just be copied to output file. You can write text without giving offset position, so text will be written from 0,0 position. And finally you can skip -color part also, in this case the text will be written in black color. The simplest launching is:

$ wm example.bmp -o modsign.bmp

Which will just copy the input image into output file

In case of getting an error message from terminal like:

error: corrupted size vs prev size ...

Just try to recompile the program, or try not to rewrite to the same image
The program works, just sometimes the user might get this kind of message

2.3 Error messages

If the program experiences some error during execution, it will print an error message. The possible error messages are:

1. **The original file must be provided** - you should provide the program a file which the program will change
2. **The original file must be bmp file** - the provided file should be of BMP extention
3. **Incorrect color** - color should be of even number of digits and should not be consisted of more than 6 digits
4. **The output file must be bmp file** - the file you provide to write the image to, should be of BMP extention
5. **Invalid variables given** - you should write the arguments in right order, the right order of arguments was mentioned in the user manual

3 C Code

3.1 C structure

The bmp structure consists of:

- width of image
- height of image
- pixel array
- header array
- size of header
- bytes per pixel

3.2 C functions

The program consists of 3 main functions:

- BMP readBMP(char *fileName) - which reads a file and write information to a BMP object
- void writeBMP(BMP img, char *fileName) - which takes information from BMP object and writes to a new file
- void changePixel(int x, int y, BMP* img, RGB color) - which changes the color of the pixel found by taking x position, y position and calculating the index of pixel by formula:

  pixel = ((currentrow * rowSize) + (x * bytes per pixel) + 16 + image data);

4 Results

4.1 The output

Now, the time has come to show the result:
I launched the program by writing:

$ wm example.bmp -text mister718 -date -color FF1236 -pos 20,10 -o modsign.bmp

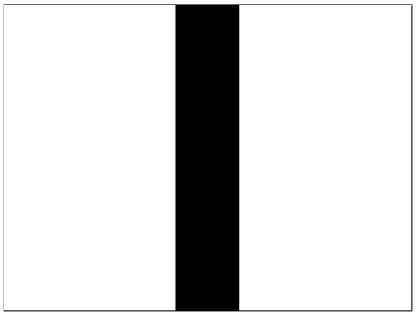

Figure 1: example.bmp

Figure 2: modsign.bmp

And you can see on the right image at the top of image the red line.

4.2 Which skills we acquired

- Understood what is a file format.
- Learnt to deal with bytes and not integers

- Learnt about how to use big little-endian values
- Learnt how to open / write files