BMP(Image) Processing

DSA2 Final Project

REPORT

French-Azerbaijan University

Authors :
Xaliq Agakerimov

Nezir Ahmedli

January 13, 2019

January 13, 2019

This is the report about the final DSA2 project where the information about the user manual is given together with the implementation part (not deep details).

1 User Manual

1.1 Compilation and Execution

The project has 3 source files: **bmp.c** **resize.c** **stitch.c** and one header file: **bmp.h** In **bmp.c** file there are some basic functions that are used in our project like reading a bmp file, the function that deals with argument problems, etc. The prototypes and the description of each function can be found in **bmp.h** header.

Apart of them the project provides the user with the makefile called ***Makefile*** which has some targets that are needed and can be used. Typing "make resize" in the terminal (or just make):

```
$ make resize
```

or

```
$ make
```

the user can create an executable file **resize** (also some object files that corresponds to the compiled source files). Pay attention: This is the program that deals with scaling of a bmp file. So the name of a bmp file should be provided as an argument when the user tries to run this program. Ex:

```
$ ./resize AzCroppedPot25.bmp
```

This will produce a new bmp file under the name *AzCroppedPot25Resized.bmp* which is just the scaled (resized) version of the previous bmp file. Pay attention: $ here denotes the prompt and is not typed by the user. Also, you should have gcc and make installed (they are installed by default in most GNU/Linux OS'es so you should be good to go). If there's a problem with arguments (few or more arguments than needed, a non-existing file, not providing a bmp file, etc.), then an appropriate error message will be displayed and the program will terminate. But if you don't even have any information about how to run this program, about the parameters or if you forget anything, you can just run the program **resize** with an option -h or –help.We can do the same thing will all the other strips (but since it will be too long for 360 images there's a script to do that for you; the script will be discussed later).

Besides we have an executable **extractStrip** file in the project folder which is the program of the last year's final project (extracting strip of a certain width from center of the given bmp file). To run the program you should provide the arguments as width and filename. Ex:

```
$ ./extractStrip -82 AzCroppedPot25Resized.bmp
```

As a result a new bmp file under the name *AzCroppedPot25Resized_S82.bmp* will be created. Again the similar help option, as well as the error messages exist. Be aware that this file is simply provided to the user as the executable one: there's no target for it in our ***Makefile***. For the purposes of project, this program should be run on resized files with strip width of 82 that we choose as the ideal one.

The another target in our ***Makefile*** is stitch. Typing

```
$ make stitch
```

you can create the executable stitch file. This program's objective as its name says is about stitching all the resized strip images that we obtain after the execution of previous programs. You can run this program just typing

```
$ ./stitch
```

Warning: The executable file stitch should be in the same directory as the files that need to be stitched (resized strips). You can also place the stitch file to the folder of these strips and run the program there, that would also work fine. From the 1st time, the program may not run. The reason can be that, you don‘t have the appropriate files (filenames), or in other words make sure that you run **resize** and **extractStrip** programs before trying to run stitch program.

The last target in our Makefile is clean which just removes all the object files and executable files in the project folder (except executable extractStrip file). One can run:

$ make clean

1.2 Batch Processing

Since the project is about resizing 360 bmp images of a pot rotated by 1 degree, extracting some strip from these 360 resized files and at the end stitching them, it takes a lot of time to do everything manually. Instead it's really a nice idea to write a small script that will do the same process. So that's why in the project folder, there's a script called **execute.sh** that does all the procedure for the user. To run it, you should provide only one argument: that's the name of a directory where all the strips located:

```
$ ./execute.sh allStrips/
```

First of all, to be sure, this script will create the resize and stitch executable files using make once more even if they were created before, because they are definitely needed in the further executions. After that the script will run the ***resize*** program on all 360 bmp images of the pot, extract strips of width 82 from these resized files using ***extractStrip*** and finally running stitch program will glue all the 360 images. As a result we will obtain an image of a flattened pot under the filename *flattenedPot.bmp* which will be in the same directory provided as an argument to the script.

2 Implementation

The main goal of this project was about a good resizing algorithm and to do that the main point was on determining the Pixel Scaling Factor for each line correctly. In order to determine that, we found an approximative mathematical function that takes the height of an image as an input and returns back the PSF of this height as an output. The function looks like:

$$F(height) = 74/(\sqrt{height} + 25)$$

where 74 stands for the maximum pixel shift and 25 was just the best number in order to get the PSF values close to their best/real values. Knowing these PSF values, the width of a current line and then using the weighted addition its pixels are determined.

3 Results

Below you can see some images where (a) corresponds to AzCroppedPot25.bmp, (b) corresponds to its resized version, (c) corresponds to the extracted strip of (b) and (d) is the flattened image obtained from 16 strips. If you want to see the image obtained from the whole 360 images you can go to link or you can just obtain it by running the script that we provide:

```
$ ./execute.sh allStrips/
```

But please be aware that you should have all the images downloaded to your computer inside some folder in the same directory as your project. It takes a few minutes (around 3-4) to run.

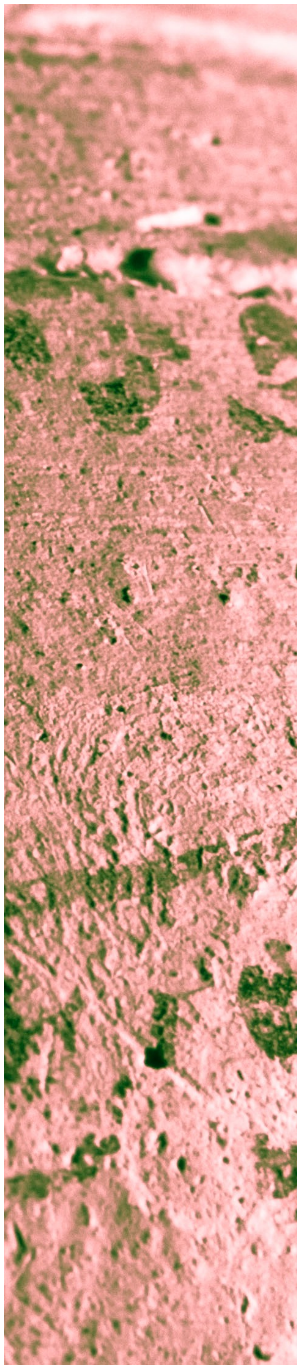

Figure 1: (a) AzCroppedPot25

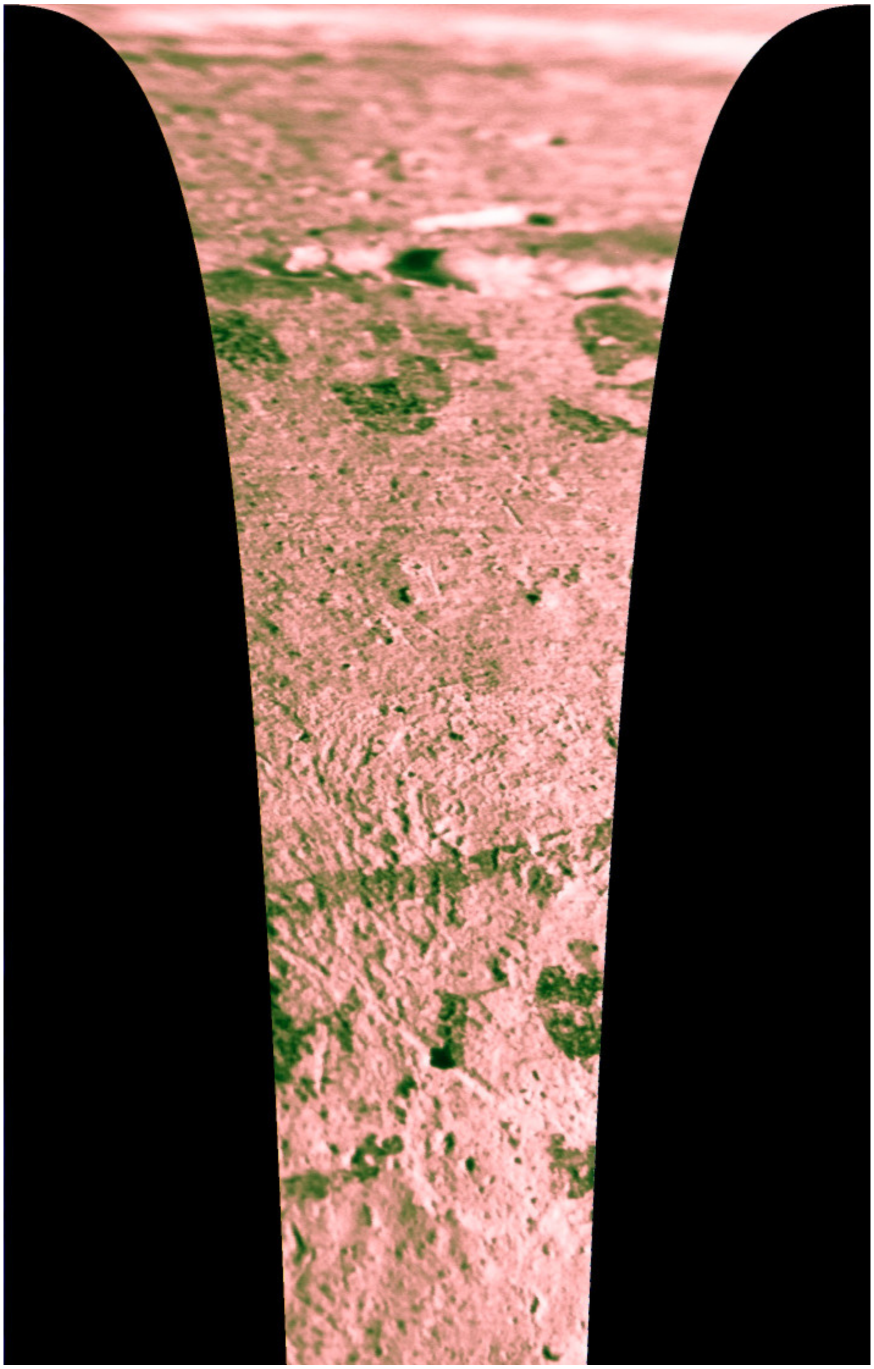

Figure 2: (b) AzCroppedPot25Resized

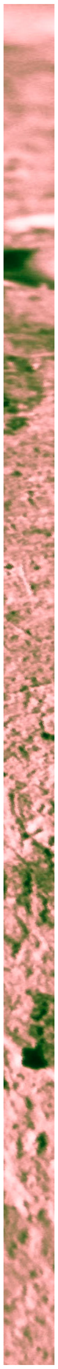

Figure 3: (c) AzCroppedPot25Resized_S82

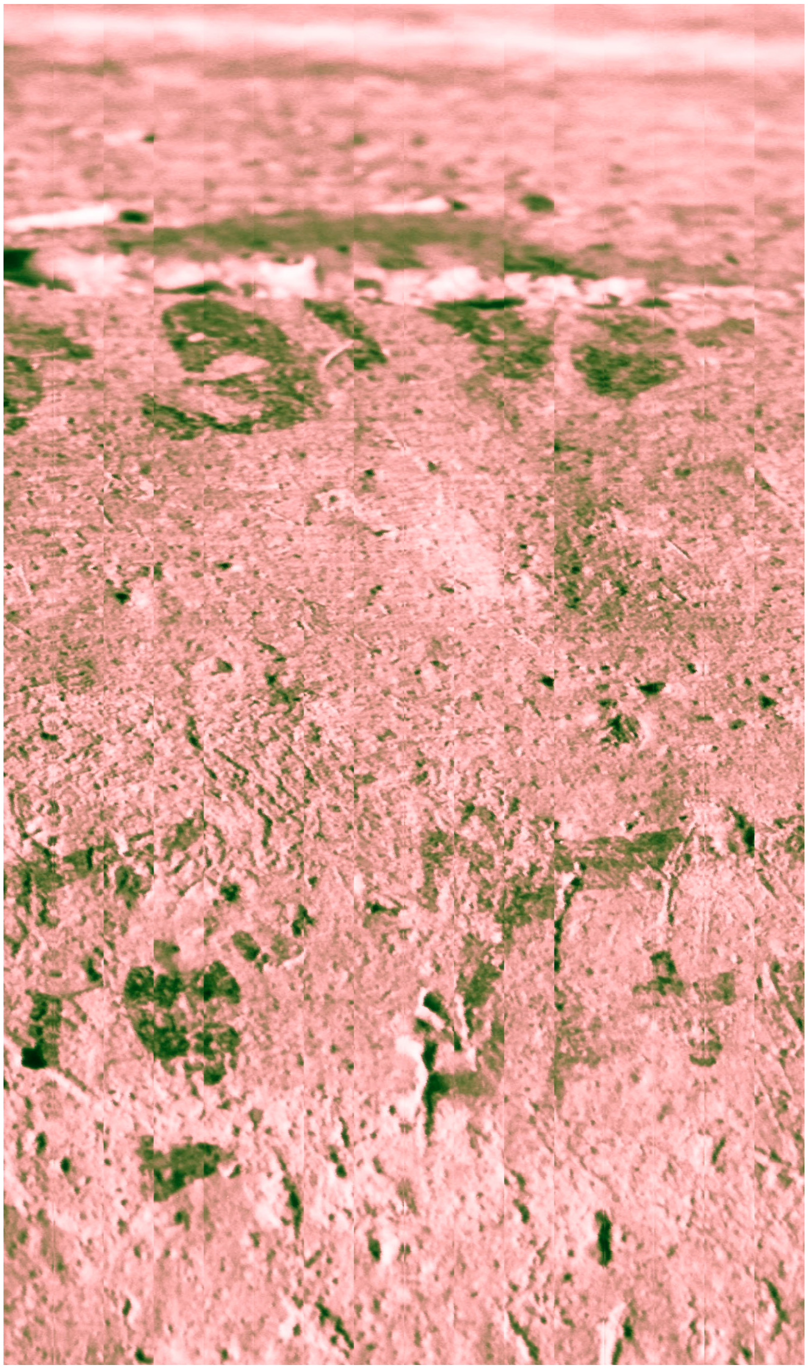

Figure 4: (d) flattenedPot