French-Azerbaijani University

Data Structure

report on project

BMP Manipulating
part 2

Group:

CS-016

Authors:
Emin Sultanov
Sanan Najafov

May 13, 2019

Contents

1 Introduction

This report looks into the procedure of working the program that helps to stitch 360 images of `bmp` format into one to obtain a panorama picture from "Pot" photo of `bmp` format and gives the information about the steps of its creation. Purpose of this project, working process, the result will be also discussed in this report.

2 Description

The main idea of the project is to understand the phenomena of stereo-graphic projection and try to stitch the photos that are given from the beginning, while elongating and cropping them. The photos of `bmp` format are big and a total size of the photos in sum becomes very big, that is why it is supposed to work with 16 photos. As the pot does not have cylinder shape the stitching process becomes complicated.

3 Creation Process

3.1 Steps of the Creation

3.1.1 Algorithm

There are five steps that have to be done to obtain elongated photo for making stitching easier:

1. As it will take a lot of time to find right elongation for each line of the photo it is supposed to **divide photo for every 223 pixels and find the scaling factors of each part that will store in a float array with size of 10**. The Pixel Scaling Factor to apply to each pixel will be :

$$psf = \frac{1}{ScalingFactor}$$

2. The second step is **the space allocation for the elongated photo**. To find the right width for the photo it is supposed to multiply the width by the maximum of psf.
3. To work with images it is needed to **store information about the file**. It suggested to use read function which had been already used during the last year project.

4. After allocating the space and getting the data for elongation it is need to re-size each part of the photo. To re-size the photo it is necessary to multiply each pixel of the part (step) of the photo by the current psf. If the current psf is 1.75, then the re-sized pixel will contain previous one and the 0.75 of the current pixel to re-size. In this case one pixel is not filled completely. That is why the next re-sized pixel will fill the 0.25 not-filled part of the previous pixel, the next one pixel and 0.5 of the following pixel (0.25 + 1.0 + 0.5 = 1.75). **In this step it was supposed to determine the formulas to fulfill all the pixels and center the re-sized parts**.
5. The last step to do is to **save the re-sized photo** by taking the file name of the photo and the name of the file where the values of shifting of each part of the photo are located. The compiling of the program has to be: ./nameOfProgramm xxx.bmp psf.
   The saved photo should be saved in format xxxResized.bmp

3.1.2 C Code

The second step of solving the problem is the creation of the program.

Structure

Before starting to write the functions there should be created a structure of BMP files. The structure is the same as in the project of last year:

```
typedef struct {
    int w;//width
    int h;//height
    unsigned char *value;//size of the file
    char *header;//header of the file
    int sizeOfHeader;//size of header
    int byte;//byte per pixel
} BMP;
```

To make the program work in the right it was needed to have 3 functions

- `BMPImage read(char *fileName);`
- `void save(BMPImage img, char *fileName);`
- `BMPImage resize(BMPImage img, float *shifting, int shiftingNum);`

The first 2 functions was created and taken from the last year project.

Resize

This function is making all the work to change the header and data of the photo to re-size it. Firstly, it is important to find maximum shifting and psf from the array that is given initially with the number of dividing of the photo. Secondly, it is necessary to get the data of bmp image to modify its size and width. To store the modifications it is need to create the header of re-sized image. Afterwards, to store the data (pixels) of each part of the photo it is necessary to modify it firstly. As there are several parts of the photo, the modifications for them were different. The changes were done by two formulas :

1. If there is full pixel to fill (>= 1.0) :
   ```
   new_image.data[(current_positionY_ofStep * new_image.width
   + new_image_positionX_ofLine)*new_image.bytes_per_pixel +
   current_rgb] += pixel_to_fill*image.data[(y*image.width +
   image_postionX)*image.bytes_per_pixel + current_rgb];
   ```

   Where `current_positionY_ofStep * new_image.width + new_image_positionX_ofLine` is the current position of pixel and `pixel_to_fill` is equal to 1.0.

2. If there is part of pixel to fill (<1) :
   ```
   new_image.data[(current_positionY_ofStep * new_image.width
   + new_image_positionX_ofLine)*new_image.bytes_per_pixel +
   current_rgb] += cur_psf*image.data[(y*image.width + image_postionX)
   + current_rgb];
   ```

   Where `current_positionY_ofStep * new_image.width + new_image_positionX_ofLine` is the current position of pixel and `pixel_to_fill` is less than 1.0.

The function returns new modified photo.

Main

In order to save the new photo as ”xxxResized.bmp” it tooks the original name of the photo and adds ”Resized” with the help of `strcpy` and `strcat` functions. Also, to make modified photos appear more comfortable it is supposed to create new directory named ”Resized” and save all the modified photos there.

3.1.3 Error Cases

- If the file is not opened. It prints that the file is not opened.

- If the file is not in BMP format. It prints an error in format.
- If the arguments more than needed or less

3.1.4 Additional tasks and Compilation

It is better to use bash scripting to make compiling more user-friendly. To compile the program user can easily write **bash extract.sh** into the terminal, where it modifies all the bmp photos in this directory with the help of `resize.c` program and saves them in the ”Resized” directory, extracts strip of width of 480 pixels and saves the photos in the ”ResizedAndCropped” directory, at the end it glues them together. It is supposed to use the last year program `crop.c` to crop the photos. The program `glue.c` was created to glue them together. This program doesn't need any input from user, but it will do nothing if it will not find any files with end as ”Cropped.bmp”. The `glue` function in this program takes as input an array of bmp images with the names that end as ”Cropped.bmp” and glue them by extracting 70 pixels of width of each photo except the last one and place them in the right order. For the last image it places the whole photo on the last position. At the end of process it will save the result in the directory ”Glued” as the file `glue.bmp`. It is also possible to compile with two command (better to use if you want to modify certain image):

1. For resizing:
    - gcc resize.c -o resize
    - ./resize xxx.bmp psf
2. For cropping:
    - gcc crop.c -o crop
    - ./resize xxx.bmp
3. For gluing:
    - gcc glue.c -o glue
    - ./glue

Note:

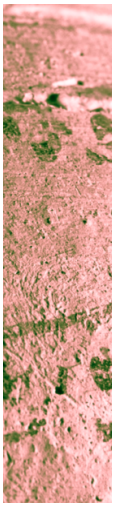
- After using programs without compiling bash code, the directories (Resized, ResizedAndCropped, Glued) will be created in the same directory as the photos. In another case, the directories will be created in consequent directories :

1. ./Resized
2. ./Resized/ResizedAndCropped
3. ./Resized/ResizedAndCropped/Glued

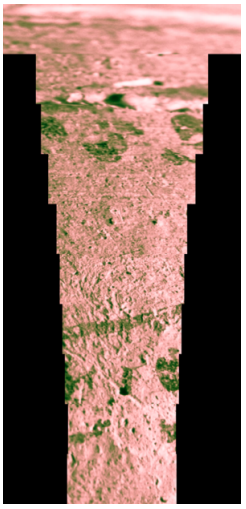
- Bash code and the program should be in the same directory as the initial photos to compile without errors.
- Bash code will not re-size and crop if there is no `bmp` files.
- Bash code will re-size only initial files and crop only the modified and at the end will glue only the modified photos.

4 Result

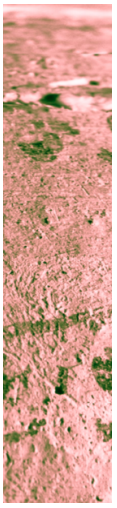
From the very beginning we were provided by extracted strip of each 360 photos, with a 1 degree of rotation between them. The first result of the project supposed to obtain the modified photo with elongated parts of it. The second result was the extraction of the middle strip with the width of 480 pixels. The final result supposed to be the glued variant of each photo to obtain **the flattened pot**



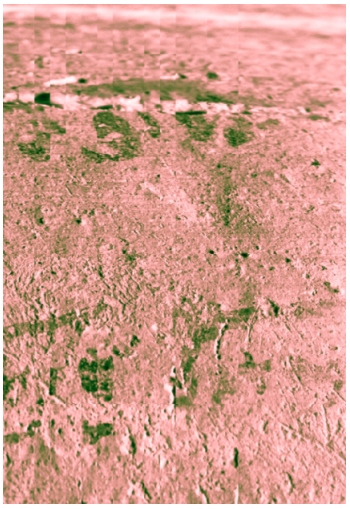
(a) Initial Strip



(b) Resized



(c) Cropped



(a) Glued

5 References

List of web-sites that contains useful information to do this project:

- https://www.tutorialspoint.com
- https://www.stackoverflow.com
- https://www.unix.stackexchange.com/
- https://www.wikipedia.org
- https://engineering.purdue.edu
- http://www.dimensions-math.org/Dim_regarder_E.htm
- https://stackexchange.com