

Sushitha Hanumanthappa Rajeeva

200425621

CS825 – Assignment1

Q.1 Programming Write a program that reads a grayscale image in raw format from a file; resize the image to a specified resolution using the single point resampling method; and save the new image into a new file in raw format. Test your program with the following data:

For the question 1, I have shown how to execute program. I have saved my program as Q1.c. I have compiled my program by giving command `gcc Q1.c -o q1`. Here, q1 is my executable. Once it is compiled, I use

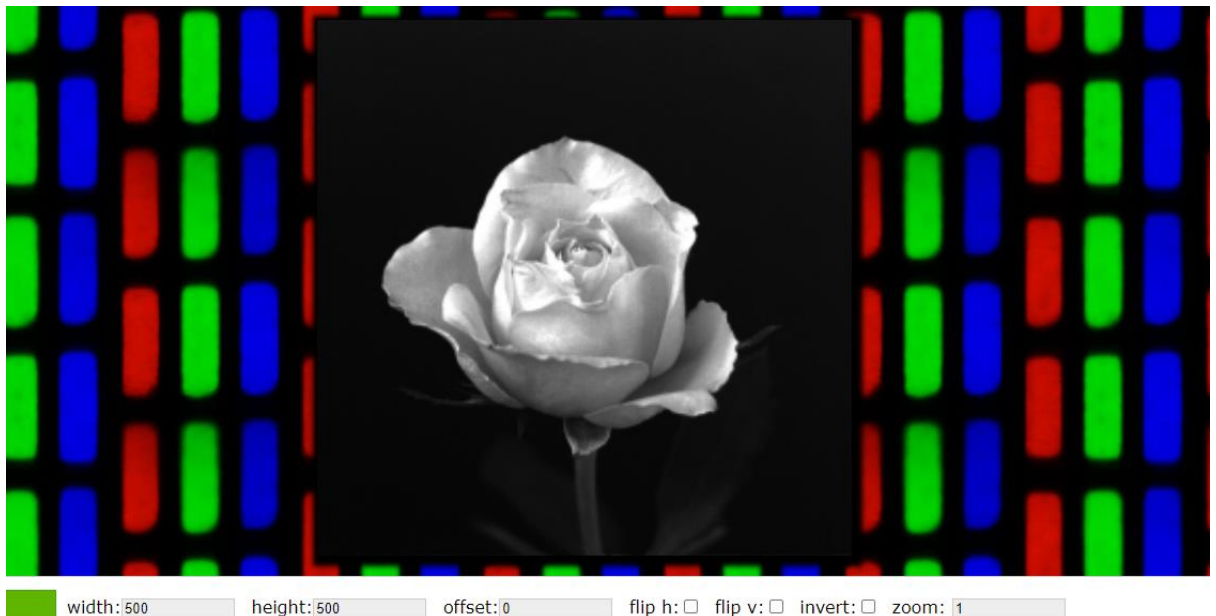
`./q1 "Path of input image which you want to process" "Path of output image where you want to save output image" In_cols In_rows Out_cols Out_rows` as shown below.

```
PS C:\Users\Sushitha Rajeev\OneDrive\Documents> gcc Q1.c -o q1
PS C:\Users\Sushitha Rajeev\OneDrive\Documents> ./q1 "C:\Users\Sushitha Rajeev\OneDrive\Desktop\museum.raw" "C:\Users\Sushitha Rajeev\OneDrive\Desktop\museum_9146.raw" 960 1280 480 640
... Load input image
1228800...Save the output image
```

Figure1: Commands to execute program

a. Input image filename: "rose.raw"

- Format: grayscale
- Original resolution: 256x256
- New resolution: 500x500



b. Input image filename: "rose.raw"

- Format: grayscale
- Original resolution: 256x256

- New resolution: 1000x1000



c. Input image filename: "museum.raw"

- Format: grayscale
- Original resolution: 1280x960
- New resolution: 640x480



d. Input image filename: "museum.raw"

- Format: grayscale

- Original resolution: 1280x960
- New resolution: 320x240



e. Input image filename: "museum.raw"

- Format: grayscale
- Original resolution: 1280x960
- New resolution: 160x120



Q.2 Repeat Q.1, but using the weighted average resampling method.

For the question 2, I have shown how to execute program. I have saved my program as Q2.c. I have compiled my program by giving command gcc Q2.c -o q2. Here, q2 is my executable. Once it is compiled, I use

./q2 "Path of input image which you want to process" "Path of output image where you want to save output image" In_cols In_rows Out_cols Out_rows as shown below.

```
PS C:\Users\Sushitha Rajeev\OneDrive\Documents> gcc Q2.c -o q2
PS C:\Users\Sushitha Rajeev\OneDrive\Documents> ./q1 "C:\Users\Sushitha Rajeev\OneDrive\Desktop\rose.raw" "C:\Users\Sushitha Rajeev\OneDrive\Desktop\Rose\rose_1.raw" 256 256 500 500
... Load input image
65536...Save the output image
```

Figure2: Commands to execute program

a. Input image filename: "rose.raw"

- Format: grayscale
- Original resolution: 256x256
- New resolution: 500x500



dth: 500 height: 500 offset: 0 flip h: ☐ flip v: ☐ invert: ☐ zoom: 1

b. Input image filename: "rose.raw"

- Format: grayscale
- Original resolution: 256x256
- New resolution: 1000x1000



c. Input image filename: "museum.raw"

- Format: grayscale
- Original resolution: 1280x960
- New resolution: 640x480



d. Input image filename: "museum.raw"

- Format: grayscale
- Original resolution: 1280x960
- New resolution: 320x240



e. Input image filename: "museum.raw"

- Format: grayscale
- Original resolution: 1280x960
- New resolution: 160x120



Q.3 Programming Write a program that reads "rose.raw" as the input image; change the intensity quantization level; and save the new image into a new file in raw format. The original image uses 8 bits per pixel, thus having 256 intensity levels. Test your program by converting intensity level to the following 4 new levels respectively:

For question3, we need to compile the program using command

```
gcc program_name.c -o q3
```

We need to execute the program by giving the command q2 "The path of input image" "The path of output image" Out_cols Out_rows Intensity_level. As input and output rows and columns are same, we just need give the command

```
PS C:\Users\Sushitha Rajeev\OneDrive\Documents> gcc Karthik.c -o q3
PS C:\Users\Sushitha Rajeev\OneDrive\Documents> ./q3 "C:\Users\Sushitha Rajeev\OneDrive\Desktop\rose.raw" "C:\Users\Sushitha Rajeev\OneDrive\Desktop\Rose\rose_16.raw" 256 256 16
... Load input image
Loading done
... Save the output image
```

Figure3:Commands to execute program

1. 128 levels (by setting the least significant bit of each pixel to 0) e.g. if a pixel's original value in binary is 10110101, the new value will be 10110100.



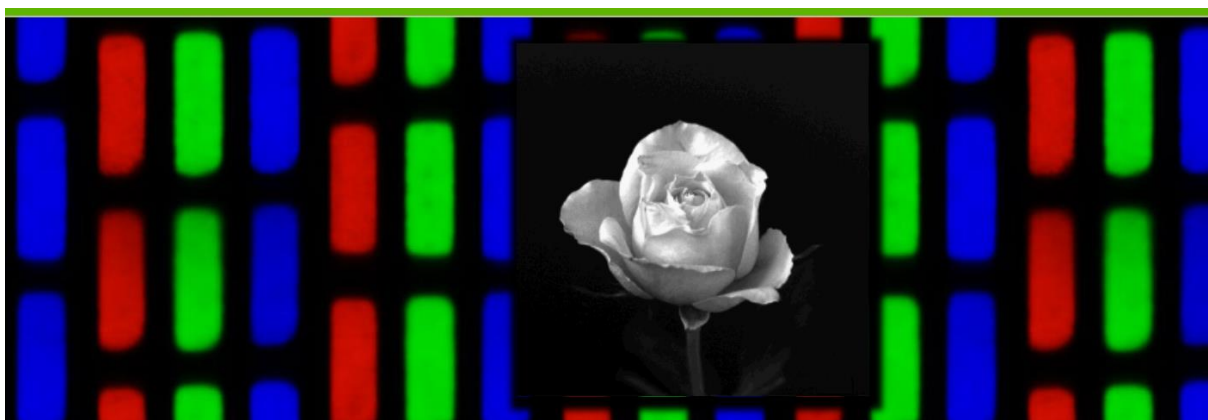
lth: 256 height: 256 offset: 0 flip h: ☐ flip v: ☐ invert: ☐ zoom: 1

2. 64 levels (by setting the two least significant bits of each pixel to 0's)



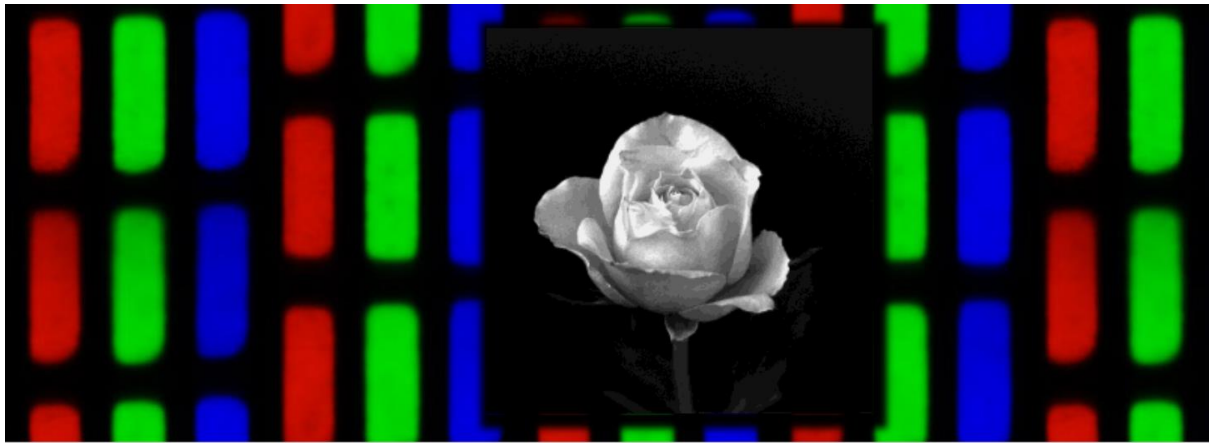
width: 256 height: 256 offset: 0 flip h: ☐ flip v: ☐ invert: ☐ zoom: 1

3. 32 levels (by setting the three least significant bits of each pixel to 0's)



width: 256 height: 256 offset: 0 flip h: ☐ flip v: ☐ invert: ☐ zoom: 1

4. 16 levels (by setting the four least significant bits of each pixel to 0's)



h: height: offset: flip h: ☐ flip v: ☐ invert: ☐ zoom: