CS5310_assignment2

Chenxi Qu

May 2024

1 Image Structure

This part the code I writed is totally in C, because some of the functions in the assignment requirements cannot be implemented in C++, they do not meet the requirements. Image.h can be called with c, or with C++'s extern as below:

```
extern "C" {
#include "Image.h"
}
```

The result as below:

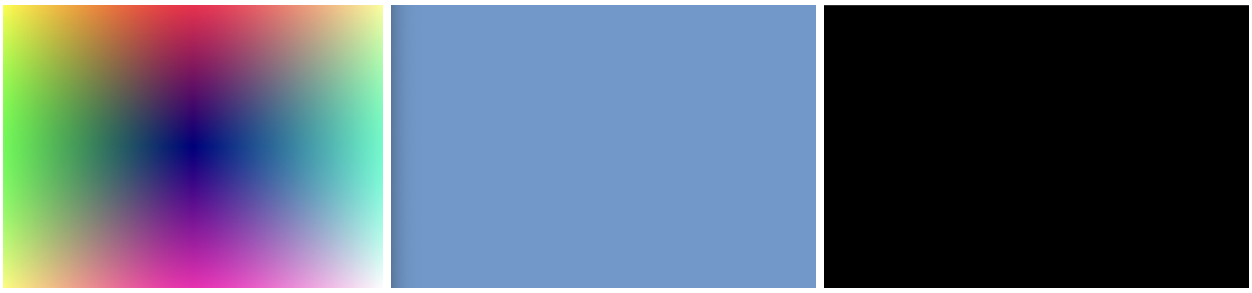

Figure 1: result of running imagetest.c

I used to set the max_value as float [0,1] I found the reason why this doesnot work and cause a crash. At last I found that PPM (P6) set max_value as int [0,255] not as I except.

2 Mandelbrot and Julia Sets

based on the requirement below:

```
A picture of the complete Mandelbrot set in an appropriate rectangle.
A picture of a Julia set defined by c = 0.7454054 + i*0.1130063 in an appropriate
    rectangle.
```

The assignment didn't involve considering colors, so the result as below.

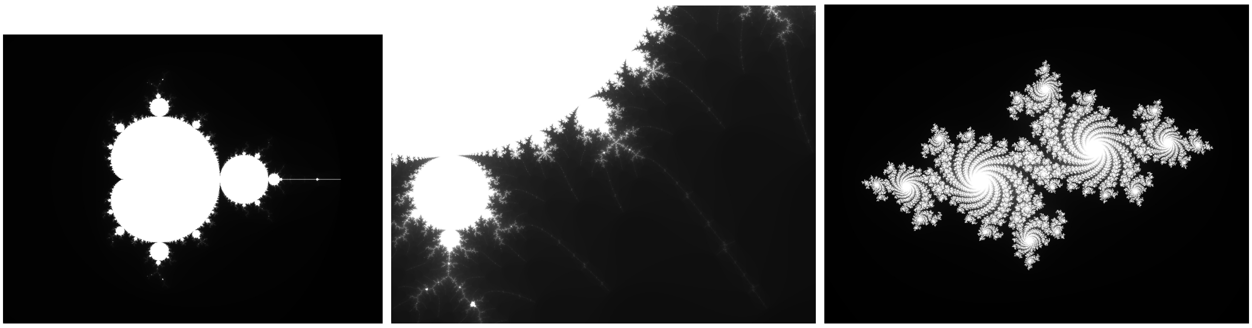

Figure 2: result of running mandeltest.cpp

Here are something I wanna mentioned:

1. I change the mandeltest.c to mandeltest.cpp, because the implantation of fractals.cpp/h I use c++.

2. Normal equation of this two method to update z is $z^2 + c$, but the lab instruction is $z^2 - c$. I choose to use the lab instruction.
3. To conform to the coordinate system required for the assignment, I inverted my y-axis.

2.1 add something I am interested in : ImageMagick

I wanna to see the movement of julia function like output a video, there are two method comes to my mind : 1. build a graphic pipeline or output lots of ppm and combine them as a JIF. I am too lasy to select the second method. Click the figure below can jump to the website I store the GIF.

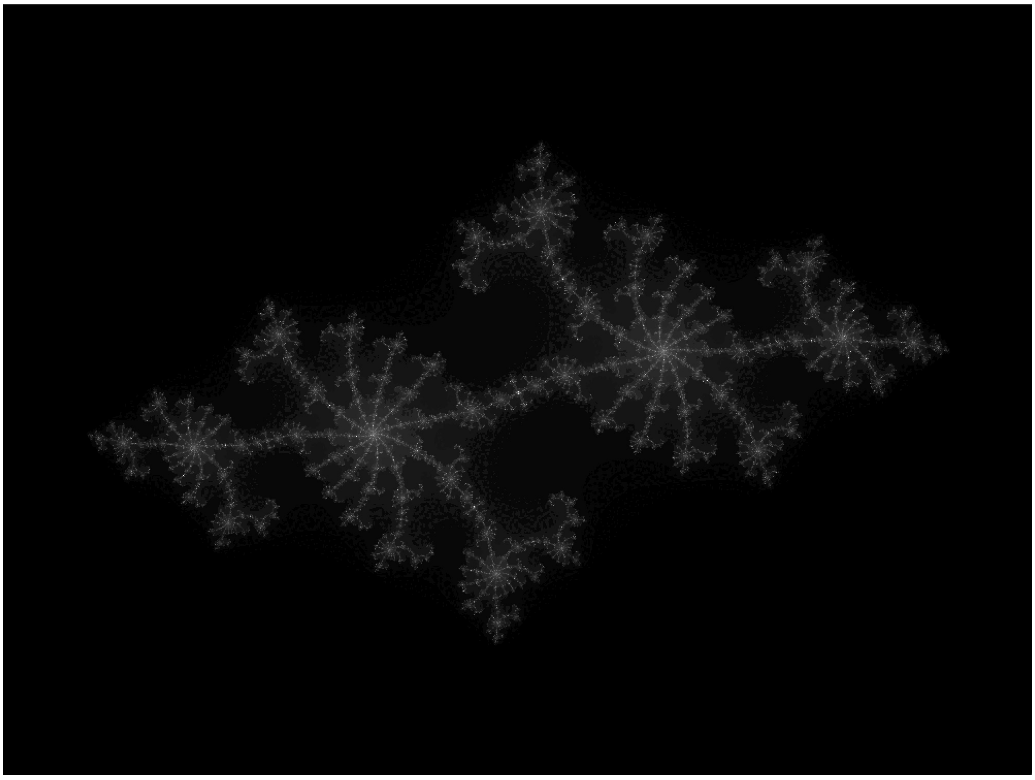

Figure 3: Julia set animation (click to view GIF)

3 Fractal Noise: Perlin

3.1 single-level Perlin

Generates a cloud pattern using a single-level Perlin noise algorithm and applies it to an image. In my main.cpp it is generateCloudsPerlin.

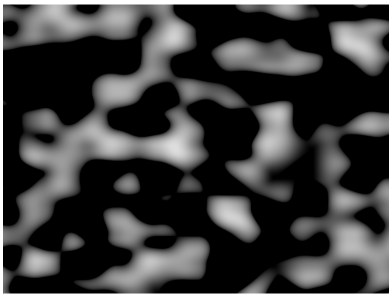

Figure 4: single-level Perlin

3.2 multi-level Perlin

Generates a cloud pattern using a multi-level Perlin noise algorithm and applies it to an image. In my main.cpp it is generateCloudsPerlin2.

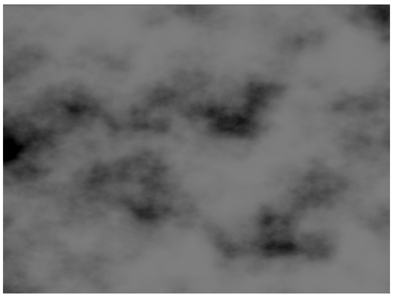

Figure 5: multi-level Perlin

3.3 animation multi-level Perlin

Generates a cloud pattern using a multi-level Perlin noise algorithm with time-based animation and applies it to an image. In my main.cpp it is generateCloudsPerlin3.

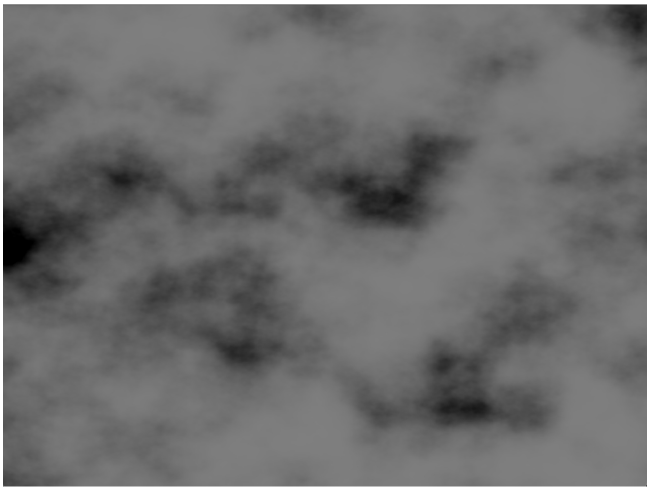

Figure 6: multi-level Perlin animation (click to view GIF)

3.4 animation of sky and cloud

Generates a cloud pattern using a multi-level Perlin noise algorithm with time-based animation and distinguishes between cloud and sky regions. In my main.cpp it is generateCloudsPerlin4.

Figure 7: my final result: animate sky(click to view GIF)

4 what I learned

1. how to compile C++ and C together.
2. The underlying principle of noise function.

References

[1] Shadertoy. Noise Generation and Visualization. Retrieved from https://www.shadertoy.com/view/lsf3WH

[2] Reputeless. Perlin Noise Implementation. Retrieved from https://github.com/Reputeless/PerlinNoise

[3] Solarian Programmer. Generating Perlin Noise in C++11. Retrieved from https://solarianprogrammer.com/2012/07/18/perlin-noise-cpp-11/

[4] Zhihu. Perlin Noise and its Application. Retrieved from https://zhuanlan.zhihu.com/p/61989717

[5] Tencent Cloud. Understanding and Implementing Perlin Noise. Retrieved from https://cloud.tencent.com/developer/article/1689991