

## Julia Set Rendering in Real-time

### Introduction

OpenGL is a graphics library with a wide range of APIs to render 2D and 3D graphics on the screen. It provides more than 200 functions for easily rendering vector graphics on GPU, including primitive function, attribute functions, viewing functions, input functions, and control functions [1]. CUDA is a parallel computing platform. It enables software developers to use GPU efficiently [2]. CUDA-OpenGL Interoperability enables the calculation of OpenGL can be done in parallel to achieve better performance. Even it is possible to finish all visualization without leaving GPU [3].

### Implementation

Julia set is a set of complex numbers. The Julia set consists of values in which a small perturbation can cause drastic changes in the sequence of iterated function values [4]. The family of complex quadratic gives a very popular complex dynamical system. The Julia set can be expressed as the function:

$$f(z) = z^2 + c$$

$c$  is a complex number. For this iteration, the Julia set is a fractal. The calculation of Julia set is computing intensive when  $c$  is updating. Thus, we use CUDA for the calculation.

The Julia is set as

$$f(z) = z^2 + 0.578e^{ia}$$

This formula is equal to

$$f(z) = z^2 + 0.578 \cos(a) + i * 0.578 \sin(a)$$

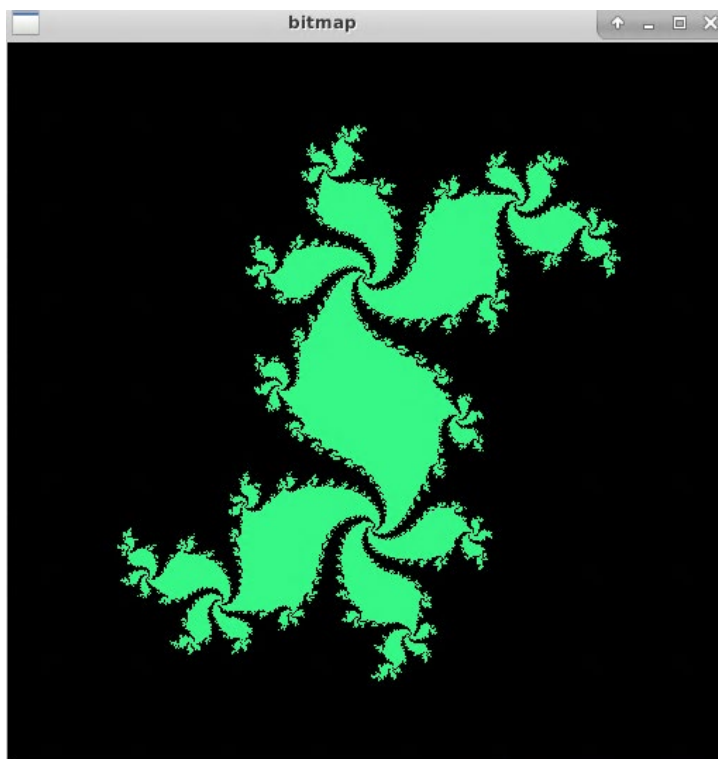
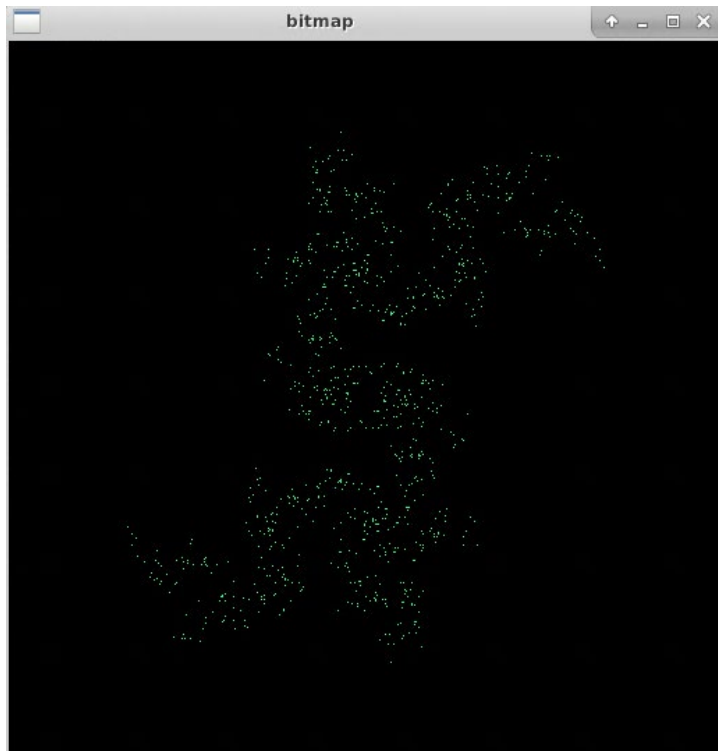
The  $a$  has a range updated from 0 to  $2\pi$ .

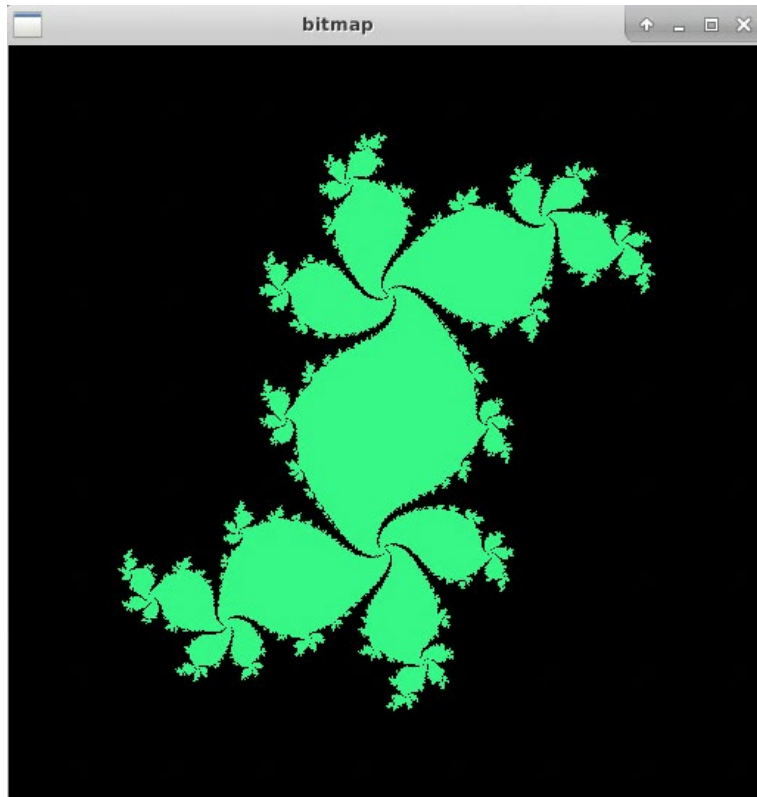
The OpenGL is used to render the graphics. Majority calculation is finished on CUDA.

### Result

The result of the Julia set when  $a$  updating.

GPGPU Final project  
Aileen Wei





## Discussing

The CUDA-OpenGL Interoperability enables the Julia set can be displayed in real-time. The OpenGL provides an easy way to render graphics. The CUDA-based GPU accelerates the calculation for rendering real-time graphics.

## Reference

- [1] Woo, Mason, et al. OpenGL programming guide: the official guide to learning OpenGL, version 1.2. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [2] Nvidia, C. U. D. A. "Nvidia cuda c programming guide." Nvidia Corporation 120.18 (2011): 8.
- [3] Sanders, Jason, and Edward Kandrot. CUDA by example: an introduction to general-purpose GPU programming. Addison-Wesley Professional, 2010.
- [4] Aarts, Jan M., and Lex G. Oversteegen. "The geometry of Julia sets." Transactions of the American Mathematical Society 338.2 (1993): 897-918.