

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Параллельные алгоритмы»**  
**Тема: Знакомство с программированием гетерогенных систем в**  
**стандарте OpenCL**

Студент гр. 0304

\_\_\_\_\_

Люлин Д.В.

Преподаватель

\_\_\_\_\_

Сергеева Е.И.

Санкт-Петербург

2023

### **Цель работы.**

Изучить принцип работы с OpenCL, реализовать построение множества Мандельброта. Сравнить производительность вычислений на процессоре и с помощью OpenCL.

### **Здание.**

Реализовать расчёт фрактала Мандельброта на OpenCL. Визуализировать результат.

В отчёте: Произвести оценку производительности.

### **Выполнение работы.**

#### ***1. Реализация на процессоре***

Для построения фрактала был реализован алгоритм escape-time. Суть алгоритма заключается в том, что для каждого пикселя изображения производятся вычисления, сможет ли точка (координаты которой являются координатами пикселя) перейти границу уравнения. В зависимости от того, за сколько итераций вычисления точка сможет «сбежать», устанавливается цвет точки. Также устанавливается предельное количество итераций, после которых вычисления для точки завершаются. В данной работе максимум итераций принят за 1000. Для каждого пикселя вычисления производятся последовательно.

Отрисовка множества Мандельброта производилась с помощью OpenGL (см. рис. 1).

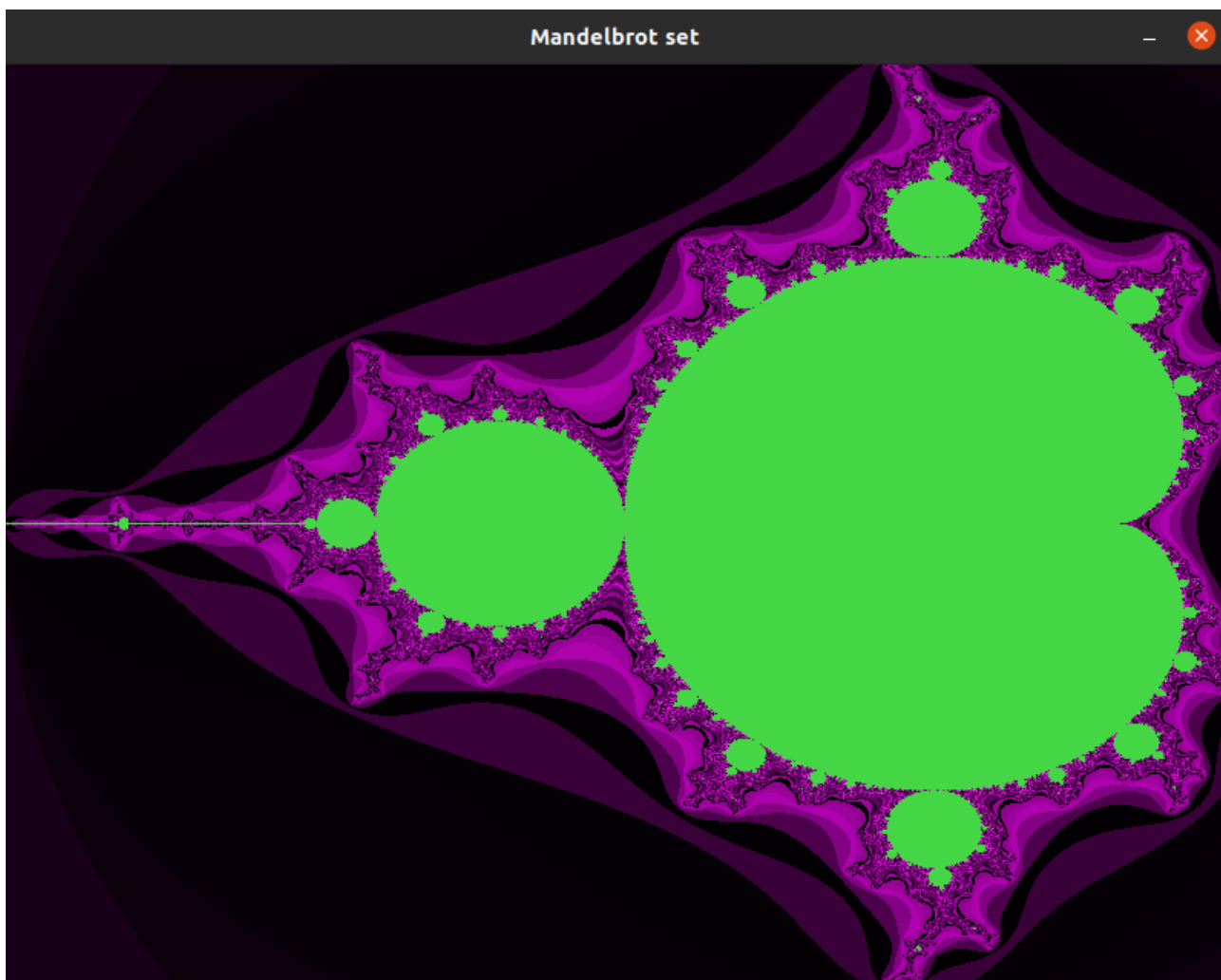


Рисунок 1. Визуализация множества Мандельброта.

## ***2. Реализация на OpenCL***

Для реализации алгоритма *escape-time* было написано ядро – программа, исполняющаяся на устройстве OpenCL. В качестве устройств могут выступать процессоры, видеокарты и др. Реализованная программа сначала пытается получить GPU, а в случае неудачи использует CPU.

Сначала создаётся контекст OpenCL, затем определяются рабочие устройства. Задача разбивается на равные части, по одной части на каждое устройство. Таким образом, достигается параллелизм.

## ***3. Оценка производительности***

В табл. 1 приведено сравнение производительности двух реализаций.

<b>Размер изображения</b>	<b>CPU, мс</b>	<b>OpenCL, мс</b>	<b>Разница, %</b>
<b>800x600</b>	1111.27	446.623	59.856
<b>1024x768</b>	1810.07	824.209	54.475
<b>1280x960</b>	2875.65	794.139	72.382
<b>1920x1440</b>	6407.06	1488.28	76.775

Реализация на OpenCL оказалась более производительной.

### **Выводы.**

В ходе работы был реализован алгоритм escape-time для построения множества Мандельброта. Было создано две реализации: с последовательным вычислением на процессоре и с параллельным вычислением на OpenCL.

Производительность реализация на OpenCL оказалась выше, чем производительность последовательной реализации. Время вычисления сократилось от 54% до 76%. С увеличением разрешения изображения в опыте разница производительности также увеличивалась.

## **ПРИЛОЖЕНИЕ А**

### **ИСХОДНЫЙ КОД ПРОГРАММЫ**

Исходный код программы доступен в репозитории  
[https://github.com/Astana-Mirza/parallel\\_algo/tree/master](https://github.com/Astana-Mirza/parallel_algo/tree/master).