

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
 Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №2
по курсу «Программирование графических процессоров»**

Обработка изображений на GPU. Фильтры.

Выполнил: О.А. Мезенин
Группа: 8О-406Б
Преподаватель: А.Ю. Морозов

Москва, 2024

Условие

Цель работы: Научиться использовать GPU для обработки изображений.

Использование текстурной памяти и двухмерной сетки потоков.

Вариант 5. Выделение контуров. Метод Робертса.

Формат изображений. Изображение является бинарным файлом, со следующей структурой. В первых восьми байтах записывается размер изображения, далее построчно все значения пикселей r, g, b, a.

Входные данные. На первой строке задается путь к исходному изображению, на второй путь к конечному изображению. $w \cdot h \leq 10^8$.

Программное и аппаратное обеспечение

Характеристики графического процессора:

- Compute capability: 8.9
- Name: NVIDIA GeForce RTX 4070 SUPER
- Total Global Memory: 12584550400
- Shared memory per block: 49152
- Registers per block: 65536
- Warp size: 32
- Max threads per block: (1024, 1024, 64)
- Max block: (2147483647, 65535, 65535)
- Total constant memory: 65536
- Multiprocessors count: 56

Характеристики центрального процессора:

- Version: AMD Ryzen 7 7700
- Max Speed: 5389 MHz
- Core Count: 8
- Thread Count: 16

Характеристики оперативной памяти:

- Size: 2x16 GB
- Type: DDR5
- Speed: 4800 MT/s

Характеристики жесткого диска:

- Device Model: ADATA SU635
- User Capacity: 480 103 981 056 bytes [480 GB]
- Sector Size: 512 bytes logical/physical
- SATA Version is: SATA 3.2, 6.0 Gb/s (current: 6.0 Gb/s)

Программное обеспечение:

- OS: Linux
- Kernel: 6.10.13-3-MANJARO
- Текстовый редактор: Kate
- Компилятор: nvcc V12.4.131

Метод решения

Пройдёмся по матрице изображения окном 2x2. Для каждой позиции преобразуем значения пикселей в чёрно-белый формат, затем найдем отклик фильтрации путём применения операции свёртки к заданному окну просмотра по следующим формулам:

- M_x – ядро свёртки для подсчёта G_x
- M_y – ядро свёртки для подсчёта G_y
- W – окно просмотра

$$G_x = W \circ M_x$$

$$G_y = W \circ M_y$$

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}$$

где ядро свёртки для метода Робертса выглядит следующим образом:

$$M_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$M_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

И запишем отклик фильтрации в текущую точку изображения.

Описание программы

Вся программа описана в одном файле main.cu. Есть следующие функции:

- int main() — входная точка программы, которая считывает массивы, настраивает объекты (например, текстурную память) для передачи их в ядро, запускает ядро и записывает результат;
- void readData(std::string& filename, int& w, int& h, uchar4** data) — открывает файл filename, считывает оттуда размеры изображения и записывает их в w и h, после чего считывает данные изображения и записывает их в data;
- void writeData(std::string& filename, int w, int h, uchar4* data) — записывает изображение data размером w на h в файл filename;
- `__global__ void kernel(cudaTextureObject_t tex, uchar4 *out, int width, int height)` — ядро проходится по изображению (текстуре) tex размера width на height, преобразует значения пикселей в чёрно-белый формат, затем применяет

операцию свёртки к окну и записывает результат в соответствующую точку нового изображения out.

Результаты

Изображения для тестов будут следующих размеров: 512x512, 2048x2048, 8192x8192.

Далее приведены замеры времени работы ядер с тремя конфигурациями, а также замеры работы без использований технологий CUDA (на CPU):

1. Конфигурация <<<dim3(2, 2), dim3(4, 4)>>>

n	Время (в мс)
512x512	5.05277
2048x2048	49.2094
8192x8192	776.413

2. Конфигурация <<<dim3(8, 8), dim3(16, 16)>>>

WxH	Время (в мс)
512x512	0.198048
2048x2048	1.28579
8192x8192	19.7406

3. Конфигурация <<<dim3(16, 16), dim3(32, 32)>>>

n	Время (в мс)
512x512	0.171968
2048x2048	0.861216
8192x8192	11.7556

4. CPU

n	Время (в мс)
512x512	10.496
2048x2048	160.813
8192x8192	2209.319

Примеры результата работы программы



Рис 1 — Исходное изображение примера 1



Рис 2 — Выходное изображение примера 1

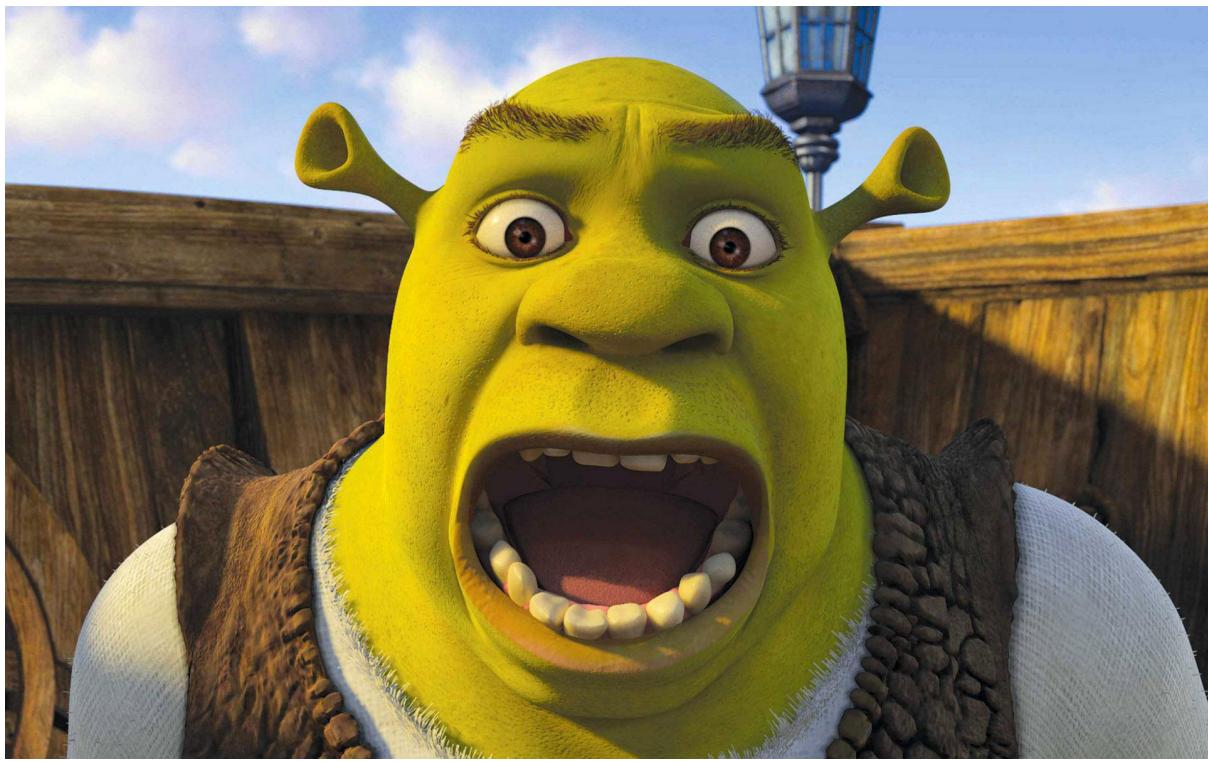


Рис 3 — Исходное изображение примера 2



Рис 4 — Выходное изображение примера 2

Выводы

В ходе лабораторной работы ознакомился с фильтрами обработки изображений, а также со способом обработки изображений на GPU через текстурную память.

Реализовал метод Робертса выделения контуров и провёл сравнительный анализ работы алгоритма с использованием технологий CUDA и без них.

Из результатов сравнения можно увидеть, что конфигурации с **большим** количеством блоков и потоков выигрывают во всех тестах, причем иногда на порядок, хотя и не исключено, что на более маленьких изображениях разница во времени между конфигурациями будет меньше. При этом даже конфигурация с наименьшем количеством блоков и потоков выигрывает у CPU, но если выбрать ещё меньше блоков и потоков, то CPU, вероятно, выиграет у такой конфигурации. Из этого можно сделать вывод, что GPU может работать с изображениями на порядки быстрее, чем CPU.