

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

Лабораторная работа №2
по курсу «Параллельные и распределенные вычисления»

Обработка изображений на GPU. Фильтры

Выполнил: *Буловятов А.М.*
Группа: *М8О-310*
Преподаватель: А.Ю. Морозов

Москва, 2023

Условие

Цель работы. Научиться использовать GPU для обработки изображений. Использование текстурной памяти и двухмерной сетки потоков.

Вариант 5. Выделение контуров. Метод Робертса.

Программное и аппаратное обеспечение:

GPU:

Графическая карта: NVIDIA Tesla T4

Compute capability : 7.5

Глобальная память : 15835398144

Разделяемая память : 49152

Регистров в блоке : 65536

Макс. размер блока : (1024, 1024, 64)

Макс. размер сетки : (2147483647, 65535, 65535)

Константная память : 65536

Количество мультипроцессоров : 40

CPU: Intel(R) Xeon(R) CPU @ 2.30GHz

Метод решения: пользователем вводятся имена файлов с исходными данными и файлом для записи результата. После считывания файла с исходными данными происходит копирование данных на устройство для дальнейшего использования. Реализация ядра заключается в переводе данных пикселя из RGB формата в яркость, вычисляются производные по икс, игрек и находятся градиенты. Результат записывается в выходной массив.

Описание программы:

```
__global__ void roberts(uchar4 *out, int w, int h) {
    int idx = blockDim.x * blockIdx.x + threadIdx.x;
    int idy = blockDim.y * blockIdx.y + threadIdx.y;
    int offsetx = blockDim.x * gridDim.x;
    int offsety = blockDim.y * gridDim.y;
    int x, y;
    uchar4 p,p1,p2,p3;
    for(y = idy; y < h; y += offsety)
        for(x = idx; x < w; x += offsetx) {
            p = tex2D(tex, x, y);
            p1 = tex2D(tex, x+1, y+1);
            p2 = tex2D(tex, x+1, y);
            p3 = tex2D(tex, x, y+1);

            double Y = red * p.x + green * p.y + blue * p.z;
            double Y1 = red * p1.x + green * p1.y + blue * p1.z;
```

```

double Y2 = red * p2.x + green * p2.y + blue * p2.z;
double Y3 = red * p3.x + green * p3.y + blue * p3.z;
double Yres = sqrt((Y-Y1)*(Y-Y1) + (Y2-Y3)*(Y2-Y3));

out[y * w + x] = make_uchar4(Yres, Yres, Yres, p.w);
}
}

```

Результаты

Результаты работы ПО на изображении размером 4k-4k

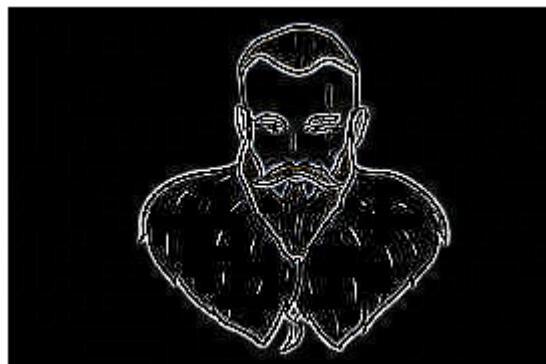
<1,1><1,32>	1.45 ms
<1,1><16,32>	1.32 ms
<1,1><32,32>	1.57 ms
<1,16><32,32>	2.13 ms
<16,16><32,32>	2.41 ms

Результаты работы <16,16><16,32> с разными входными данными.

1k-1k	1.54 ms
4k-4k	2.13 ms
5k-5k	2.48 ms

Результаты работы на CPU для изображения 2k-2k: 1.26 s.

Результат работы:



Выводы

В данной лабораторной работе реализовали фильтр Робертса с использованием технологии CUDA. Данная работа позволила научиться работать с изображениями а также бинарными файлами. Использование текстур позволило значительно упростить работу и сосредоточиться на реализации алгоритма.