МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

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

# Обработка изображений на GPU. Фильтры.

Выполнил: М.А.Трофимов
Группа: 8О-408Б
Преподаватели: К.Г. Крашенинников,
А.Ю. Морозов

Москва, 2021

## Условие

**Цель работы.** Научиться использовать GPU для обработки изображений. Использование текстурной памяти. **Вариант 1:** Гауссово размытие.

## Программное и аппаратное обеспечение

Характеристики GPU "NVIDIA GeForce GTX 950"

CUDA Driver Version / Runtime Version 11.4 / 11.4
CUDA Capability Major/Minor version number: 5.2
Total amount of global memory: 1997 MBytes (2094137344 bytes)
(006) Multiprocessors, (128) CUDA Cores/MP: 768 CUDA Cores
GPU Max Clock rate: 1278 MHz (1.28 GHz)
Memory Clock rate: 3305 Mhz
Memory Bus Width: 128-bit
L2 Cache Size: 1048576 bytes
Maximum Texture Dimension Size (x,y,z) 1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
Maximum Layered 1D Texture Size, (num) layers 1D=(16384), 2048 layers
Maximum Layered 2D Texture Size, (num) layers 2D=(16384, 16384), 2048 layers
Total amount of constant memory: 65536 bytes
Total amount of shared memory per block: 49152 bytes
Total shared memory per multiprocessor: 98304 bytes
Total number of registers available per block: 65536
Warp size: 32
Maximum number of threads per multiprocessor: 2048
Maximum number of threads per block: 1024
Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
Max dimension size of a grid size (x,y,z): (2147483647, 65535, 65535)
Maximum memory pitch: 2147483647 bytes
Texture alignment: 512 bytes

Характеристики CPU Intel i5-4460

# of Cores 4
# of Threads 4
Processor Base Frequency 3.20 GHz
Max Turbo Frequency 3.40 GHz
Cache 6 MB Intel® Smart Cache
Bus Speed 5 GT/s
Intel® Turbo Boost Technology 2.0 Frequency 3.40 GHz
TDP 84 W

Характеристики RAM
Total 15 Gi
Swap 2 Gi
Операционная система: Ubuntu 20.04 LTE
IDE Sublime Text 3
Compiler nvcc for cuda 11.4

## Метод решения

Общая идея в том, что мы получаем на вход радиус фильтра, предподсчитываем одномерный фильтр, который будем прикладывать. Прикладываем сначала к каждому фильтру, “размывая” по оси X, сохраняя результаты в отдельный, промежуточный массив, потом к полученным результатам прикладываем фильтр, “размывая” по Y, сохраняя полученный ответ.

## Описание программы

Всё писалось одним файлом main.cu, в котором 2 функции ядра, main и вспомогательная функция, для построения фильтра. Доступ к входным данным функций ядра производится через текстурные ссылки и куда массивы. Первая функция ядра прикладывает фильтр по X, вторая - по Y. Промежуточные значения сначала сохраняются в обычный массив на девайсе, потом копируют в куда массив, для работы через текстурную ссылку.

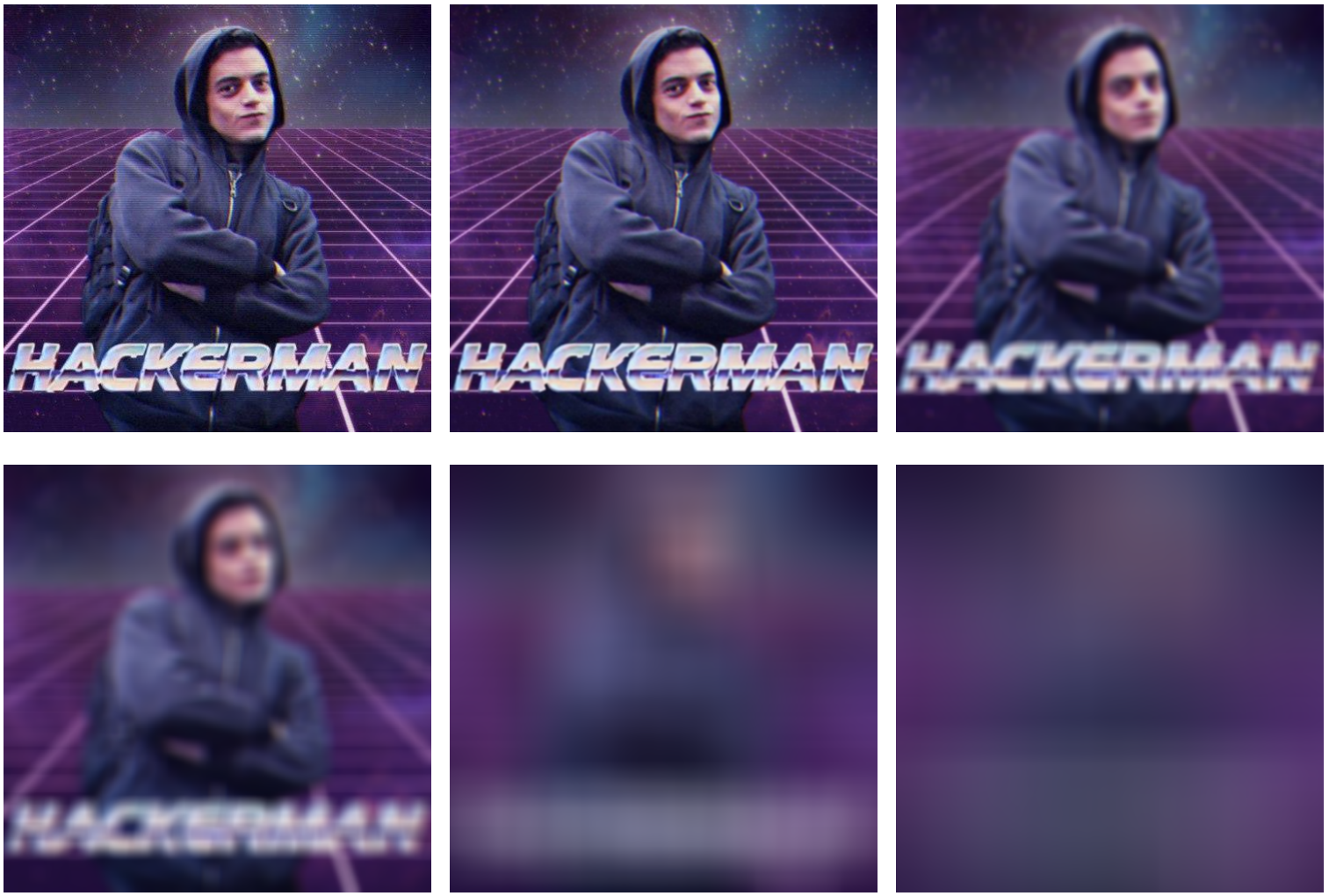
## Результаты

| Конфигурация | радиус 10 | | | радиус 100 |
|---|---|---|---|---|
| | тест картинки 320x320 | тест картинки 500x500 | тест картинки 889x906 | тест картинки 889x906 |
| На CPU | 387.94ms | 934.45ms | 3027.26ms | 30304.59ms |
| 1,32 | 79.49ms | 160.06ms | 515.81ms | 4779.61ms |
| 1,1 32, 32 | 6.76ms | 16.27ms | 63.21ms | 531.32ms |
| 1,32 32,32 | 1.47ms | 3.42ms | 12.18ms | 115.19ms |
| 32,32 1,32 | 3.22ms | 7.44ms | 23.76ms | 214.11ms |
| 32,32 32,32 | 1.49ms | 3.21ms | 11.73ms | 107.86ms |

Ядровые функции занимали в среднем 89% и 7% процентов времени работы GPU. Остальное время было занято копированием памяти.

## Демонстрация работы

Размытие картинки 500x500 с радиусом фильтра 0, 1, 5, 10, 50 и 100 соответственно.




## Выводы

Как видно, параллелизация алгоритма свёртки оказалось весьма эффективным. Получаемое ускорение в 300 раз, что весьма хорошо для больших данных. Написание программы оказалось достаточно лёгким, особенно благодаря использованию текстурной памяти, не считая собственных глупых ошибок.