

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
КАФЕДРА СУПЕРКОМПЬЮТЕРОВ И КВАНТОВОЙ ИНФОРМАТИКИ



КАФЕДРАЛЬНЫЙ ПРАКТИКУМ

ЗАДАНИЕ 1: BASIC IMAGE CONVOLUTION ON NVIDIA GPU USING CUDA

Выполнил:
Алёшин Никита Александрович
группа 423

Москва 2021

Постановка задачи.

Необходимо реализовать программу на C++ и CUDA, которая:

1. Получает входные параметры командной строки (типа используемого фильтра и входных данных);
2. Загружает с диска необходимые изображения;
3. Преобразует изображения в линейные массивы (развертка матрицы в линейный массив);
4. Копирует эти массивы в память GPU;
5. Запускает CUDA-ядра, которые применяют к изображениям необходимый фильтр;
6. Выгружает результат в память CPU;
7. Выводит два времени работы: только CUDA-ядер и CUDA-ядер с копированием данных;
8. Сохраняет полученные после фильтрации изображения на диск (также в виде изображений, которые можно потом посмотреть).

Необходимо выбрать два фильтра размера 3x3 и один фильтра размера 5x5. Тип используемого фильтра должен задаваться параметром командной строки.

В качестве входных данных необходимо использовать данные двух видов:

1. Одно изображение большого размера (например, 2000x2000);
2. Много изображений маленького размера (300x300).

Необходимый тип входных данных (использовать большие или маленькие изображения) следует указывать в качестве второго параметра командной строки.

Описание структуры программы.

Программа для преобразования изображения в линейный массив (развертки матрицы в линейный массив) и сохранения полученных после фильтрации изображений на диск использует библиотеку `lodepng`.

Копирование данных на GPU с CPU и в обратную сторону происходит с помощью функции `cudaMemcpy()`.

Реализованные фильтры.

	<i>Gaussian Blur</i>	<i>Sharpen</i>	<i>Edge Detection</i>
$\frac{1}{256}$	$\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

Примеры фильтрации.

Оригинал:



Gaussian Blur:



Sharpen:



Edge Detection



Оригинал:



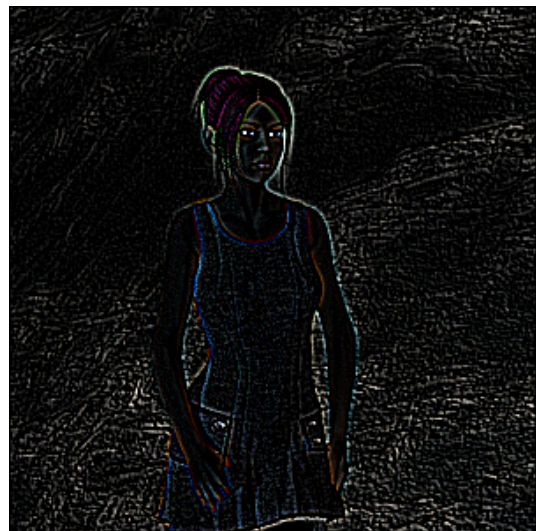
Gaussian Blur:



Sharpen:



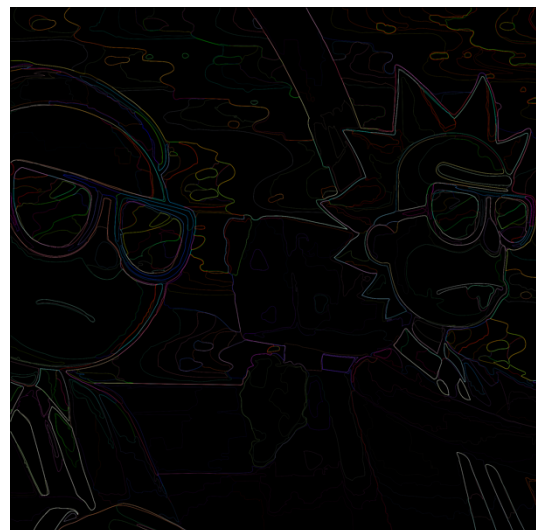
Edge Detection



Оригинал:



Edge Detection



Время выполнения программы:

Приведены средние результаты замеров по 5 экспериментам:
изображения 300x300:

фильтр 5x5:

ядра: 0.0964224 мс

ядра + копирование: 0.652774 мс

фильтр 3x3:

ядра: 0.0513536 мс

ядра + копирование: 0.609318

изображения 2000x2000:

фильтр 5x5:

ядра: 2.76345 мс

ядра + копирование: 5.81028 мс

фильтр 3x3:

ядра: 1.12634 мс

ядра + копирование: 4.18483 мс

Вывод.

Среднее время работы программы на фильтрах 5x5 больше, чем на фильтрах 3x3 и на маленьких, и на больших изображениях. Копирование данных на GPU и с GPU в данной задаче занимает гораздо больше времени, чем непосредственно вычисления.