

**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: Ф.М. Шавандрин

Группа: 8О-408Б

Преподаватель: А.Ю. Морозов

Москва, 2022

## Условие

Написать конвертер программирования для работы с нижеописанным форматом.

**Формат изображений.** Изображение является бинарным файлом, со следующей структурой:

width(w)	height(h)	r	g	b	a	r	g	b	a	r	g	b	a	...	r	g	b	a	r	g	b	a
4 байта, int	4 байта, int	4 байта, значение пикселя [1,1]				4 байта, значение пикселя [2,1]				4 байта, значение пикселя [3,1]				...	4 байта, значение пикселя [w - 1, h]				4 байта значение пикселя [w,h]			

В первых восьми байтах записывается размер изображения, далее построчно все значения пикселей, где

- r -- красная составляющая цвета пикселя
- g -- зеленая составляющая цвета пикселя
- b -- синяя составляющая цвета пикселя
- a -- значение альфа-канала пикселя

## Цель работы:

Научиться использовать GPU для обработки изображений.

Использование текстурной памяти и двумерной сетки потоков.

**Вариант 5.** Выделение контуров. Метод Робертса.

## Программное и аппаратное обеспечение

### GPU:

- Название NVIDIA GeForce GT 545
- Compute capability: 2.1
- Графическая память: 3150381056
- Разделяемая память: 49152
- Константная память: 32768
- Количество регистров на блок: 32
- Максимальное количество нитей: (1024, 1024, 64)
- Максимальное количество блоков: (65535, 65535, 65535)
- Количество мультипроцессоров: 3

### Сведения о системе:

- Процессор: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz
- ОЗУ: 15 ГБ
- HDD 500 ГБ

### Программное обеспечение:

- OS: Ubuntu 16.04.6 LTS
- Текстовый редактор: Vim
- Компилятор: nvcc

## Метод решения

Для решения этой задачи нужно предварительно конвертировать все пиксели изображения из RGB представления в YUV с помощью формулы:

$$brightness = 0.299 * x + 0.587 * y + 0.114 * z,$$

где  $x, y, z$  – RGB-компоненты пикселя. Далее ищем первые частные производные для каждого пикселя:

$$G_x = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, G_y = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

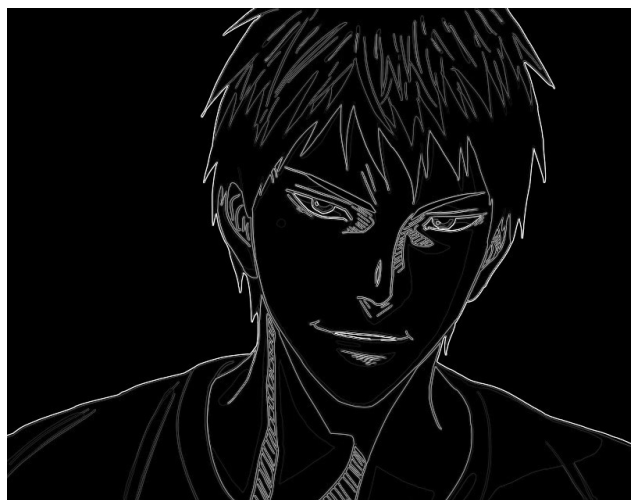
Норма градиента будет окончательным ответом для каждого пикселя, т. е.

$$|(g)| = \sqrt{(g_x^2 + g_y^2)}$$

## Описание программы

В функции *kernel* вычисляется яркость для каждого пикселя по формуле выше, затем находятся первые частные производные и берётся минимум из 255 и нормы градиента, т. к. при вычислениях норма градиента может превышать максимальное значение 255, что будет не совсем корректно, т.к значение 255 соответствует максимально насыщенному белому цвету.

## Пример работы программы



## Результаты

В качестве тестирования используются изображения с разрешением 736\*736, 1280\*720, 2975\*2980 и 3264\*2724 соответственно.

CPU:

Разрешение	Время работы, мс
736*736	1.2323
1280*720	1.8558
2975*2980	11.2803
3264*2724	11.3630

<<<(16, 16), (32, 32)>>>

Разрешение	Время работы, мс
736*736	1.3517
1280*720	2.1399
2975*2980	18.3572
3264*2724	18.4780

<<<(32, 32), (32, 32)>>>

Разрешение	Время работы, мс
736*736	1.5618
1280*720	2.4682
2975*2980	18.7678
3264*2724	18.9545

<<<(1, 16), (32, 16)>>>

Разрешение	Время работы, мс
736*736	1.2238
1280*720	2.0273
2975*2980	19.8591
3264*2724	19.9263

## Выводы

В ходе данной лабораторной работы были получены знания о контурах изображений и о методах выделения контуров. Был реализован метод Робертса для обработки изображений с использованием текстурной памяти и двухмерной сетки потоков. На практике выделение контуров могут указывать, например, на изменение глубины или изменения в свойствах материала. Таким образом, целью выделения контуров является фиксация важных событий и изменений мира.