

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

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

**Лабораторная работа № 3
по курсу «Программирование графических процессоров»**

Классификация и кластеризация изображений на GPU.

Выполнил: Ф.М. Шавандрин
Группа: 8О-408Б
Преподаватель: А.Ю. Морозов

Условие

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

В результирующем изображении, на месте альфа-канала, должен быть записан номер класса (кластера), к которому был отнесён соответствующий пиксель. Если пиксель можно отнести к нескольким классом, то выбирается класс с наименьшим номером. Формат изображений соответствует формату, описанному ниже.

Цель работы: Научиться использовать GPU для классификации и кластеризации изображений. Использование *константной памяти и одномерной сетки потоков*.

Вариант 1. Метод максимального правдоподобия.

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 данной задачи необходимо сначала вычислить вектор средних и ковариационной матрицы по следующим формулам:

$$avg_j = \frac{1}{np_j} \sum_{i=1}^{np_j} ps_i^j$$

$$cov_j = \frac{1}{np_j - 1} \sum_{i=1}^{np_j} (ps_i^j - avg_j) * (ps_i^j - avg_j)^T,$$

где $ps_i^j = (r_i^j \ g_i^j \ b_i^j)^T$ - i -ый пиксель j -ой выборки.

Затем для каждого пикселя p , номер класса jc считается следующим образом:

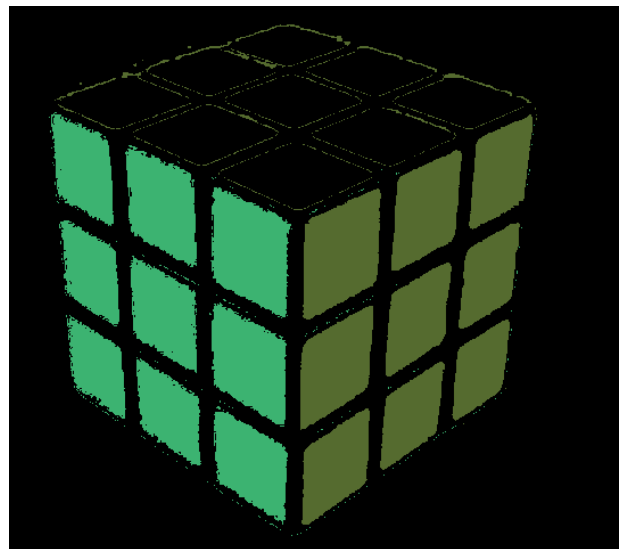
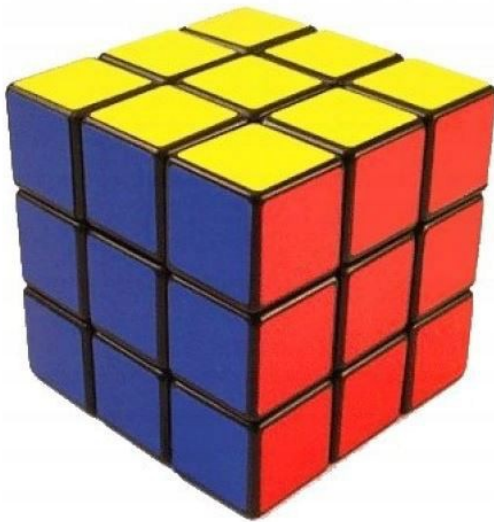
$$jc = \operatorname{argmax} \left[- (p - avg_j)^T * cov^{-1} * (p - avg_j) - \log(|\det(cov_j)|) \right]$$

Описание программы

Вектор средних, ковариационная и обратная ковариационная матрицы и детерминант считаются на ЦПУ. В функции *kernel* каждый пиксель обрабатываемого изображения классифицируется с помощью формулы, описанной выше.

Пример работы программы

В качестве примера выбрал изображение 3 граней кубика рубика: синей, жёлтой и красной. В качестве тестовой выборки выбрал 5 пикселей для каждого цвета, определив, что классу 0 соответствует синий цвет, классу 1 - жёлтый и классу 2 — красный.



Результаты

Для тестирования программы на GPU, будем сравнивать её с программой на CPU, обрабатывающей изображения с разрешением 736*736, 1280*720, 2975*2980, 3264*2724 соответственно.

CPU:

| Разрешение | Время работы, мс |
|------------|------------------|
| 736*736 | 1.4078 |
| 1280*720 | 1.9215 |
| 2975*2980 | 10.8620 |
| 3264*2724 | 11.1214 |

<1, 32>

| Разрешение | Время работы, мс |
|------------|------------------|
| 736*736 | 28.4864 |
| 1280*720 | 48.4596 |
| 2975*2980 | 465.7523 |
| 3264*2724 | 466.9933 |

<32, 32>

| Разрешение | Время работы, мс |
|------------|------------------|
| 736*736 | 2.2897 |
| 1280*720 | 3.8695 |
| 2975*2980 | 37.0149 |
| 3264*2724 | 37.0729 |

<128, 128>

| Разрешение | Время работы, мс |
|------------|------------------|
| 736*736 | 1.6004 |
| 1280*720 | 2.7009 |
| 2975*2980 | 25.8122 |
| 3264*2724 | 25.8861 |

Выводы

В ходе данной лабораторной работы были освоены навыки работы с классификацией изображений с помощью различных методов. Был реализован метод максимального правдоподобия для кластеризации обрабатываемых изображений с использованием константной памяти и одномерной сетки потоков. На практике кластеризация изображений нужна для того, чтобы определить, например, какие пиксели относятся к Ferrari, а какие — к Porsche, если рассматривается задача, связанная с автомобилями. Глобально, суть кластеризации состоит в том, чтобы объекты, попавшие в одну группу, имели сходные характеристики, в то время как у объектов из разных групп эти характеристики должны значительно отличаться.