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: Р.С. Лисин

Группа: 8О-406Б

Преподаватели: К.Г. Крашенинников,

А.Ю. Морозов

Москва, 2023

**Условие**

**Цель работы**: Научиться использовать GPU для классификации и

кластеризации изображений. Использование константной памяти и одномерной

сетки потоков.

Формат изображений соответствует формату описанному в лабораторной

работе 2. Во всех вариантах, в результирующем изображении, на месте альфа-канала

должен быть записан номер класса (кластера), к которому был отнесен

соответствующий пиксель. Если пиксель можно отнести к нескольким классам, то

выбирается класс с наименьшим номером.

**Вариант 4. Метод спектрального угла.**:

**Входные данные**. На первой строке задается путь к исходному изображению,

на второй - путь к конечному изображению. На следующей строке задаётся число классов. Далее на каждой строке (число строк равно числу классов) задаётся число пикселей, у которых известен класс - номер строки в 0-индексации, и описываются две координаты пикселей.

**Программное и аппаратное обеспечение**

В качестве графического процессора использую видеокарту NVIDIA Tesla T4.

Compute capability : 7.5

Name : Tesla T4

Total Global Memory : 15835398144

Shared memory per block : 49152

Registers per block : 65536

Warp size : 32

Max threads per block : (1024, 1024, 64)

Max block : (2147483647, 65535, 65535)

Total constant memory : 65536

Multiprocessors count : 40

В качестве редактора кода использовался Jupyter Notebook в Google Colab.

**Метод решения**

На GPU реализуем метод спектрального угла для каждого пикселя, предварительно подсчитав на CPU пронормированную (поделили среднее значение на его норму) оценку средних значений для известных пикселей.

**Описание программы**

Создаётся один динамический двумерный массив data для хранения изображения. Он копируется на GPU. Также в программе используется массив normed\_avgs с типом данных RGB (структура с 3 double), в который записываются пронормированные оценки средних значений для известных пикселей. Далее этот массив копируется в массив device\_normed\_avgs, находящийся в константной памяти.

В функции ядра kernel производится нахождение argmax для метода спектрального угла - для определения их класса пиксели умножаются на пронормированные оценки средних значений по цветам, и эти произведения суммируются. Используется одномерная сетка потоков, так как нам не нужно смотреть на соседние пиксели. Результат записывается в одновременно входной и выходной двумерный массив arr.

**Результаты**

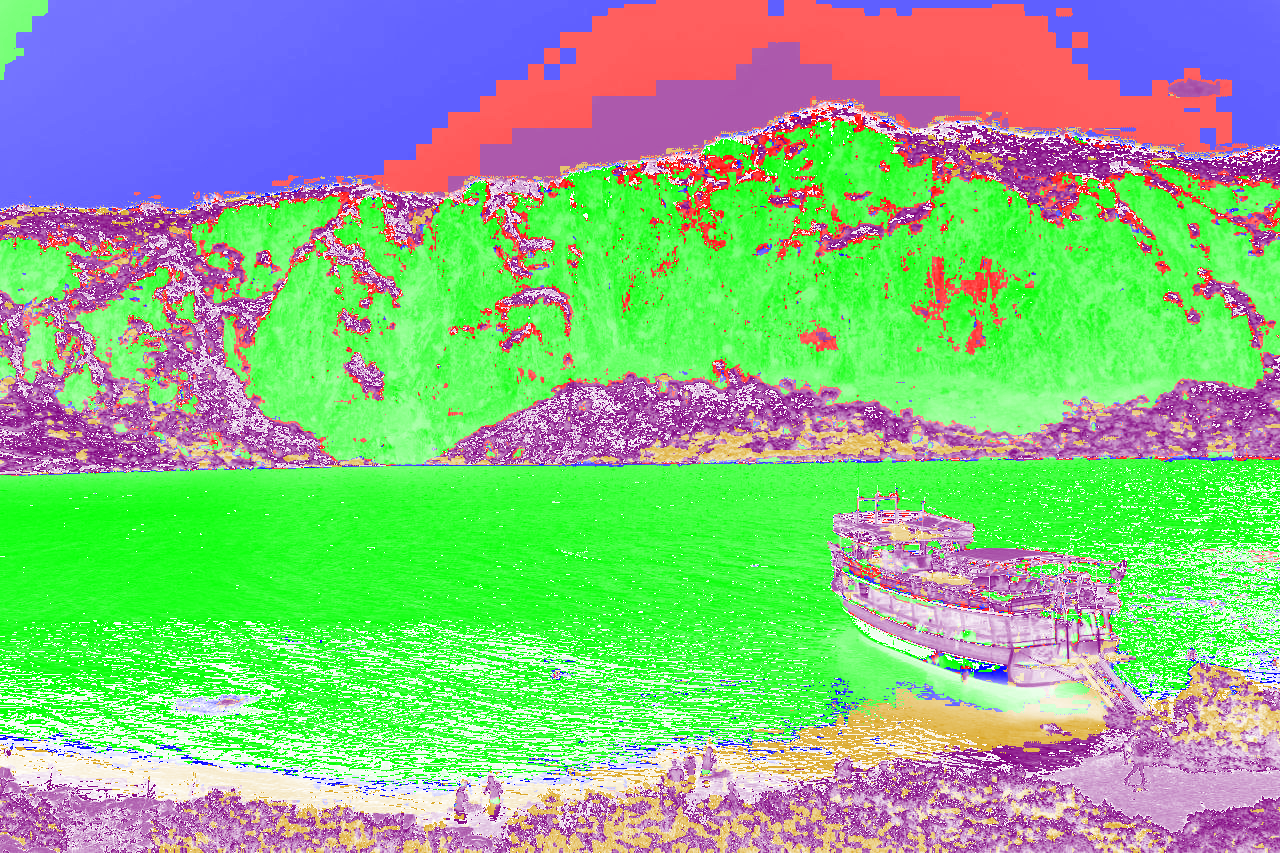
Рассмотрим время работы программы на различных тестах при различных размерах сетки и на CPU. Будем замерять непосредственно время работы алгоритма. В качестве тестов используется одна картинка с видом на море и количеством классов 5. Для разных тестов меняются её размеры. Результаты приведены в таблице ниже.

|  |  |  |  |
| --- | --- | --- | --- |
| Размер сетки ядра | 1300x1300 px, мс | 2000x2000 px, мс | 5000x5000 px, мс |
| CPU | 61.953000 | 147.449000 | 914.561000 |
| <<<1, 32>>> | 101.296417 | 217.312576 | 771.050171 |
| <<<32, 32>>> | 3.180576 | 7.493056 | 46.793758 |
| <<<128, 128>>> | 1.219904 | 2.856352 | 17.771423 |
| <<<256, 256>>> | 1.060448 | 2.493888 | 15.538880 |
| <<<512, 512>>> | 0.999968 | 2.324224 | 14.422400 |
| <<<1024, 1024>>> | 0.996640 | 2.282112 | 14.375840 |

Алгоритм на CPU справляется гораздо медленнее чем на GPU. Это безусловно связано с тем, что в данном случае распараллеливание в разы ускоряет работу алгоритма.

Пример работы программы для 5 классов с 3 пикселями на класс.





**Выводы**

Во третьей лабораторной работе я познакомился с методами кластеризации и классификации пикселей и использовал константную память. GPU позволяет очень быстро обрабатывать изображения, собственно поэтому он и называется графическим процессором. В примере классификатор отработал достаточно хорошо для такой сложной живой фотографии. Самое необычное, что подчеркнула классификация, - небо далеко не однотонное на данной картинке. Человеческим глазом это не сразу заметно.