МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

Лабораторная работа №3
по курсу «Программирование графических процессоров»

Классификация и кластеризация изображений на GPU. Вариант 2. Метод расстояния Махаланобиса.

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

Москва, 2021

Условие

Цель работы.
Научиться использовать GPU для классификации и
кластеризации изображений. Использование константной памяти.

Вариант 2. Метод расстояния Махаланобиса.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. На следующей строке, число nc -- количество классов. Далее идут nc строчек описывающих каждый класс. В начале j-ой строки задается число npj—количество пикселей в выборке, за ним следуют npj пар чисел -- координаты пикселей выборки. nc , npj ≤ 32 ≤ 2 , w*h .

Формула определения класса

$$jc = arg\ max_j \left[-(p - avg_j)^T * cov_j^{-1} * (p - avg_j) \right]$$

Программное и аппаратное обеспечение

Nvidia GeForce GTX 660
Compute capability: 3.0
Графическая память: 2048MB
Регистров на блок: 65536
Нитей на блок: 1024
Мультипроцессоров: 5
Всего ядер: 960

Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz
Тактовая частота: 3.4 GHz
Кэш-память: 6 MB

Оперативная память
Объём: 8 GB

Жёсткий диск
Объём: 2 TB

Программное обеспечение
OS: Windows 10
IDE: Visual Studio 2019
CUDA: v10.2 nvcc

Метод решения

Рассчитываем на процессоре средние значения и обратные ковариационные матрицы, сохраняем их в константную память. В ядре для кадого пикселя по формуле определяем класс.

Описание программы

1. Макрос CSC
   Проверяет, с каким статусом завершаются CUDA-операции, и в, случае ошибки, выводит на стандартный поток ошибок debug-информацию.

2. determineClass
   Функция для исполнения на GPU. Принимает на вход пиксель и число классов, для каждого класса рассчитывает значение по формуле, затем выбирает первое максимальное значение класса и возвращает номер этого класса.

3. Ядро.
   Основная функция, работающая на устройстве. Принимает указатель на выходной вектор, размерности картинки и число классов для определения. В цикле для каждого пикселя вызывает функцию, определяющую класс пикселя, и обновляет значение класса этого пикселя.

4. Main
   Считывает данные из файла, рассчитывает по этим данным средние значения и обратные матрицы ковариации, выделяет под все эти данные память на GPU, запускает ядро.

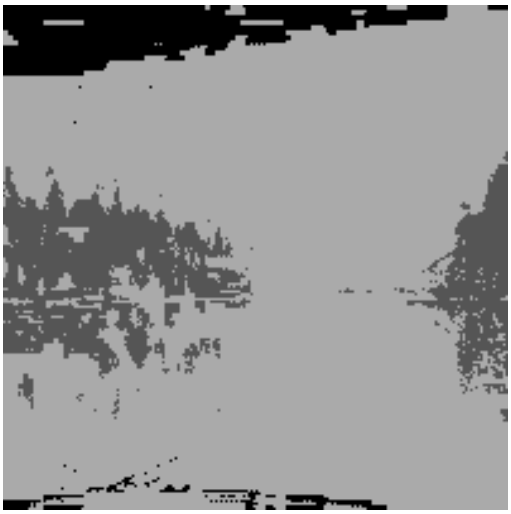
Результаты

Размер теста	<<<1, 32>>>	<<<32, 32>>>	<<<1024,1024>>>	CPU
128 на 128	190.37мс	6.02мс	1.59мс	51мс
420 на 420	1840.61мс	65.43мс	13.91мс	203мс
1024 на 1024	Видеокарта отказывается ждать так долго и убивает ядро	366.15мс	81.05мс	516мс

Демонстрация работы

Лог запуска:

```
PS C:\vs_projects\lab3\x64\debug> ./lab3.exe
200.data
out.data
3
4 10 10 15 17 28 14 5 3
4 10 89 20 100 196 75 199 100
4 50 50 45 45 60 57 57 60
```

Выводы

Проделав работу, я изучил принципы устройства константной памяти GPU, освежил свои знания по линейной алгебре за первый курс. Реализованный мною алгоритм может применяться при анализе картографических изображений, например с целью выделения различных природных зон или элементов рельефа.