МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Факультет «Информационные технологии и прикладная математика»
Кафедра «Вычислительная математика и программирование»

Лабораторная работа №3
по курсу «Параллельная обработка данных»

Классификация и кластеризация изображений на GPU

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

Москва, 2019

Условие

Цель работы: Науиться использовать GPU для классификации и кластеризации изображений. Использование константной памяти.
Вариант 4: Метод спектрального угла.

$$jc = argmax_j [p^T * \frac{avg_j}{||avg_j||}]$$

Программное и аппаратное обеспечение

- GPU: Geforce 940MX
  - Compute capability : 5.0
  - Total Global Memory : 2147483648
  - Shared memory per block : 49152
  - Registers per block : 65536
  - Max threads per block : (1024, 1024, 64)
  - Max block : (2147483647, 65535, 65535)
  - Total constant memory : 65536
  - Multiprocessors count : 3
- CPU: Intel Core i5-6200U 2.30GHz
- RAM: 4GB
- Software: Windows 10, Visual Studio Code, nvcc

Метод решения

Классификация методом спектрального угла используется для сравнения спектральных характеристик изображения со спектральными характеристиками эталонов. Иными словами принадлежность пикселя к тому или иному классу определяется углом между ним и эталонным представителем класса. Этот угол должен быть наименьшим.
В данной лабораторной работе мы сначала ищем средний элемент каждого класса на основе предложенной выборки, а затем нормируем его. После этого мы начинаем обработку изображения, мы считаем скалярные произведения классифицируемого пикселя и средних значений каждой выборки. Чем больше значение скалярного произведения, тем меньше спектральный угол с данным классом.

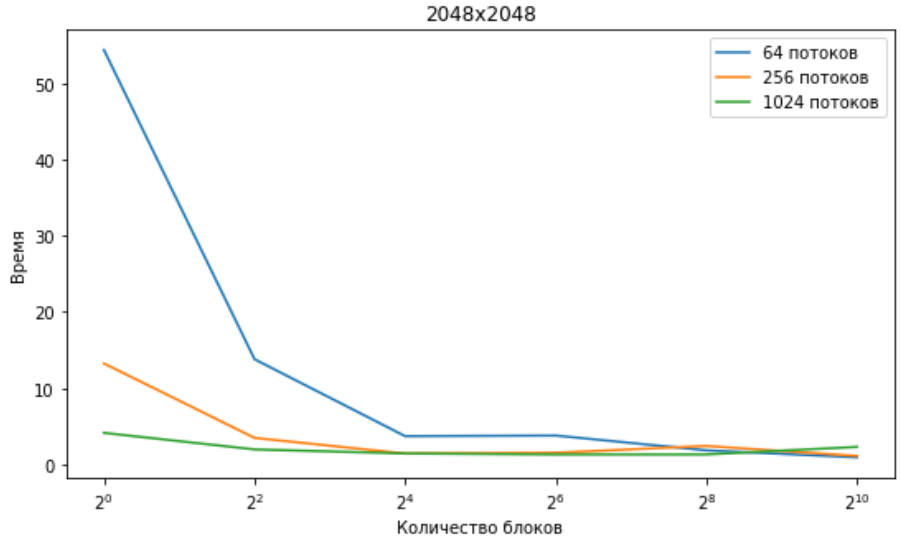
Описание программы

- void main() - функция, выполняемая хостом: считывание исходных данных, подготовка константной памяти, копирование с и на gpu, запуск ядра, вывод результатов.
  - char *inname, *outname – имя входного и выходного файлов
  - int w, h – размер изображения
  - uchar4 *data – указатель на изображение
  - float3 NormalizedMeanClassPixelCPU – среднее значение пикселя каждого класса
  - nc – кол-во классов
  - np – размер выборки

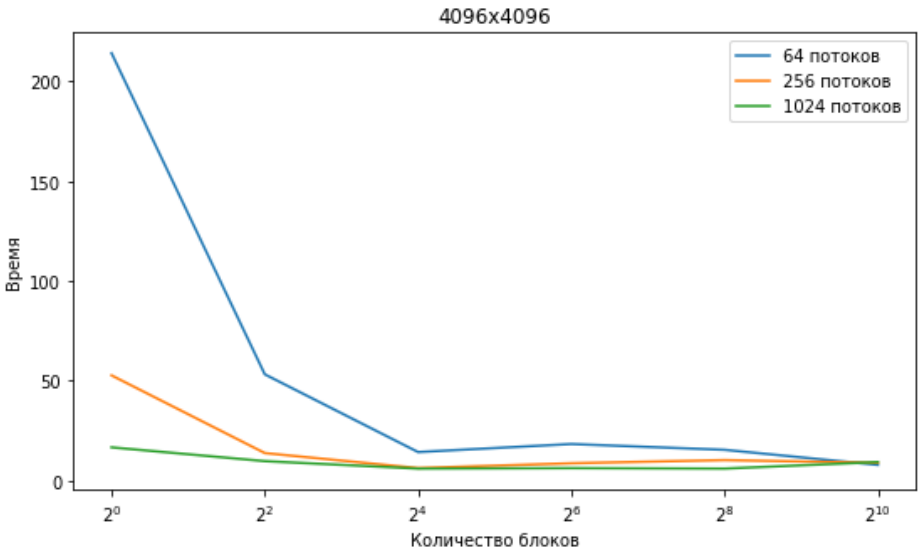
- void write_data(char *outname, uchar4 *data, int w, int h) – запись изображения data в файл с именем outname
- void read_data(char *inname, uchar4 **data, int *w, int *h) – считывание изображения из файла inname в массив data
- __global__ void kernel(uchar4 *dev_arr, int w, int h, int nc) – ядро, запускает поток для каждого пикселя изображения.
- __device__ uchar get_class(uchar4 p, int c) – определяет класс пиксля p.

Результаты

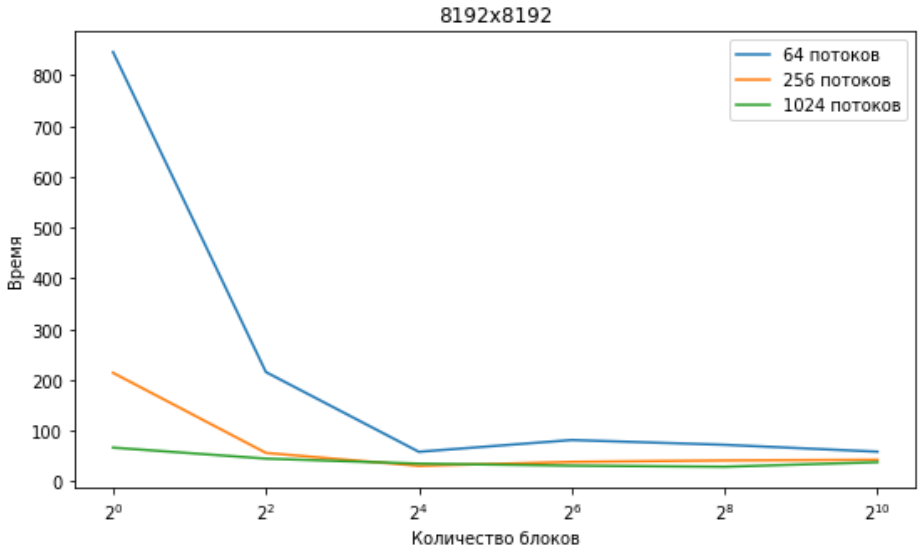
Посмотрим, как размеры входного изображения влияют на производительность:



Изображение 2048x2048



Изображение 4096x4096



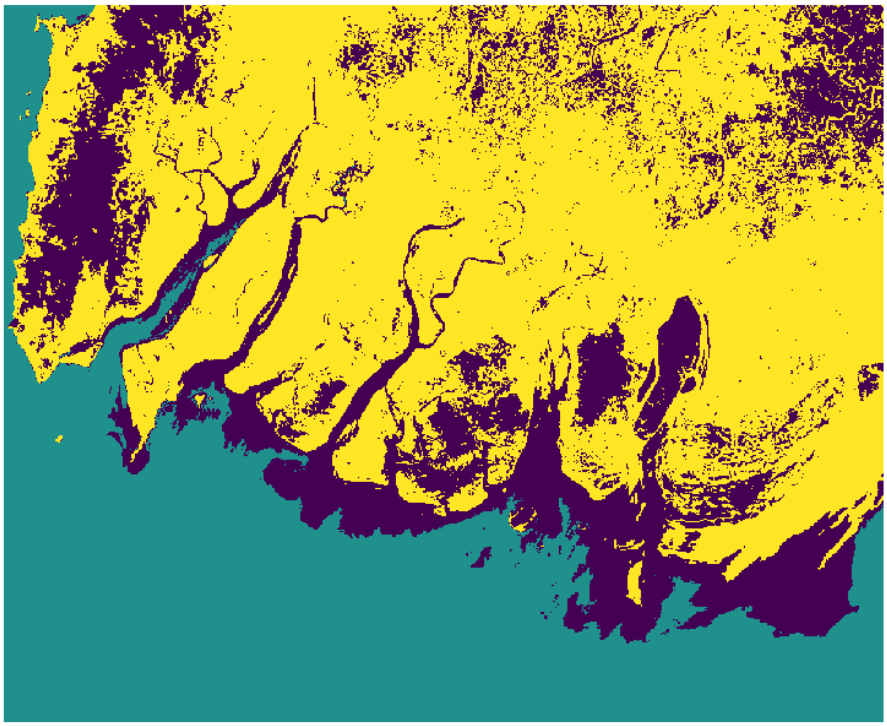
Изображение 8192x8192

Как видно наилучшие результаты достигаются при конфигурации ядра <<<dim3(4, 4), dim3(16, 16)>>>.

При использовании cpu, скорость, как и ожидается, заметно снижается:

Размеры	2048x2048	4096x4096	8192x8192
Время	717	2826	11445

Пример входного изображения и его обработанной версии на основе выходных данных:



Выводы

Реализованный алгоритм вычисления спектрального угла может применяться в задачах классификации. Одно из его конкретных применений — классификация земной поверхности по спутниковым снимкам. Такой алгоритм легко распраллелить и он легок в реализации. Оценить результаты его работы можно по изображению выше, если не обращать внимания на прибрежную полосу, то алгоритм достаточно точно определил лес, пустыню и море. Возможно, точность была бы еще выше, если увеличить размер выборки и число классов(добавить прибрежную зону).