Практикум CUDA
Отчет No1

Плужников Иван, 423

2021 Г, МГУ, ВМК, СКИ

Задание: 1. получает входные параметры командной сроки (типы используемого фильтра и
входных данных — про них далее);
2. загружает с диска необходимые изображения;
3. преобразует изображения в линейные массивы (развертка матрицы в линейный массив)
4. копирует эти массивы в память GPU;
5. запускает CUDA-ядра, которые применяют к изображениям необходимый фильтр;
6. выгружает результат в память CPU;
7. выводит 2 времени работы: только CUDA-ядер, а также CUDA-ядер + копирований
данных;
8. сохраняет полученные после фильтрации изображения на диск (также в виде изображений, которые можно потом посмотреть).

Фильтры:
Необходимо выбрать 2 фильтра размера 3x3, и один фильтр размера 5x5.
Тип используемого фильтра должен задаваться параметром командной строки.
Входные данные
В качестве входных данных необходимо использовать данные двух видов:
1) одно изображение большого размера
изображения в интернете самостоятельно (чем больше — тем лучше)
2) много изображений маленького размера (300x300).

Реализация:
Реализовано 3 фильра: **Edge detection, Box blur, Unsharp masking**

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Изображение считано в вектор с помощью Lodepng.cpp, Lodepng.h

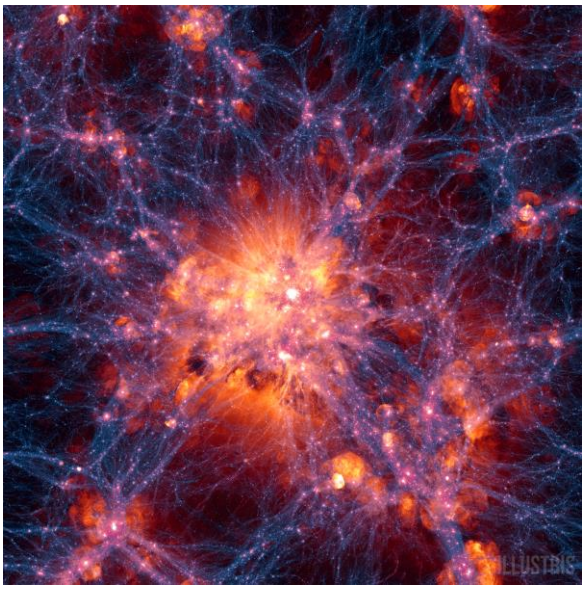
Далее, с помощью cudaMalloc на устройстве выделяется память для входного и выходного изображений, входное копируется.

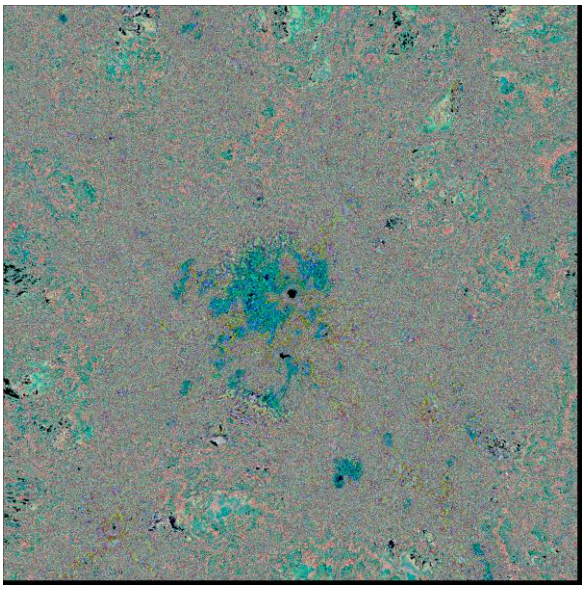
После этого на основе cudaGetDeviceProperties и параметров изображения, рассчитывается размер блока и грида, чтобы каждый пиксель рассчитывался на своем отдельном треде.
На следующем шаге ядру передаются входное и выходное изображение, параметры изображения и тип фильтра. В ядре фильтр применяется, данные записываются в выходное изображение.
По завершении работы ядра данные выходного изображения копируются на Host устройство.

Результаты:
Unsharp masking


Box Blur

Edge detection





Время :
маленькое изображение и ядро 5 на 5
caunting: 0.087264
all time 0.957216

маленькое изображение и ядро 3 на 3
caunting: 0.024416
all time 0.865664

большое изображение и ядро 5 на 5
caunting: 0.613632
all time 3.85574

большое изображение и ядро 3 на 3
caunting: 0.309024

all time 3.60499

Вывод:
На большем изображении и большем ядре программа работает дольше, что неудивительно. На копирование данных уходит значительно больше времени, чем на функционирование ядра.