

Лабораторная работа №2

Обработка изображений на GPU. Фильтры.

Цель работы. Научиться использовать GPU для обработки изображений.
Использование *текстурной памяти*.

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

Пример картинки размером 2 на 2, синего цвета, в шестнадцатеричной записи:

```
02000000 02000000 0000FF00 0000FF00 0000FF00 0000FF00
```

Студентам предлагается самостоятельно написать конвертер на *любом* языке программирования для работы с вышеописанным форматом.

В данной лабораторной работе используются только цветовые составляющие изображения (r g b), альфа-канал не учитывается. При расчетах значений допускается ошибка в ± 1 . Ограничение: $w < 2^{16}$ и $h < 2^{16}$. Во всех вариантах, кроме 2-го и 4-го, в пограничном случае, необходимо “расширять” изображение за его границы, при этом значения соответствующих пикселей дублируют граничные. То есть, для любых индексов i и j , координаты пикселя $[ip, jp]$ будут определяться следующим образом:

```
ip := max(min(i, h), 1)
jp := max(min(j, w), 1)
```

Вариант 1. Гауссово размытие.

Необходимо реализовать размытие по Гауссу с линейной сложностью от радиуса размытия (двухпроходный алгоритм).

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. На следующей строке, целое число r -- радиус размытия, $w \cdot h \leq 5 \cdot 10^7$, $0 \leq r < 1024$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data 1	03000000 03000000 00010200 03040500 06070800 08070600 05040300 02010000 00000000 00140000 00000000	03000000 03000000 02030300 03040400 04040500 03040300 03050200 02040200 01050100 01070000 00040000
in.data out.data 1	01000000 05000000 00000000 00000000 64646400 00000000 00000000	01000000 05000000 00000000 1B1B1B00 2D2D2D00 1B1B1B00 00000000
in.data out.data 5	03000000 03000000 00000000 00000000 00000000 00000000 FFFFFFF0 00000000 00000000 00000000 00000000	03000000 03000000 02020200 02020200 02020200 02020200 03030300 02020200 02020200 02020200 02020200

Вариант 2. Медианный фильтр.

Необходимо реализовать медианный фильтр для изображения. Медианным элементом считается элемент с номером $n/2$ в отсортированном массиве, для его нахождения использовать гистограмму и неполную префиксную сумму по ней.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. На следующей строке, целое число r -- радиус размытия, $w \cdot h \leq 5 \cdot 10^7$, $r \leq 100$, $w \cdot h \cdot r \leq 64 \cdot 10^7$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data 1	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	03000000 03000000 06050600 06050600 06050600 06050600 04050400 06050600 09080700 06050400 06050400
in.data out.data 2	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	03000000 03000000 04050400 04050400 04050400 04050400 04050400 04050400 04050400 04050400 04050400

Вариант 3. Билинейная интерполяция.

Необходимо реализовать изменение размера изображения, используя билинейную интерполяцию. Значения пикселей привязываются к их центрам, а не к углам. Использовать встроенную текстурную билинейную интерполяцию запрещается.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. На следующей строке, два числа w_p и h_p -- новые размеры изображения. $w \cdot h \leq 4 * 10^8$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data 2 2	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	02000000 02000000 03030400 05060600 05050500 04040400
in.data out.data 2 6	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	02000000 06000000 01020300 06070800 03030400 05060600 06060500 04030300 07060500 04030200 05050500 04040400 05050500 05050500
in.data out.data 3 5	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	03000000 05000000 01020300 04050600 07080900 04040400 04050500 05050500 09080700 06050400 03020100 03030200 0E0E0D00 01000000 00000000 14141400 00000000

Вариант 4. SSAA.

Необходимо реализовать избыточную выборку сглаживания. Исходное изображение представляет собой “экранный буфер”, на выходе должно быть сглаженное изображение, полученное уменьшением исходного.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. На следующей строке, два числа w_p и h_p -- размеры нового изображения, гарантируется, что размеры исходного изображения соответственно кратны им. $w \cdot h \leq 4 * 10^8$.

Пример:

Входной файл	hex: in.data	hex: out.data
--------------	--------------	---------------

in.data out.data 2 2	04000000 04000000 01020300 04050600 07080900 0A0B0C00 0D0E0F00 10111200 13141500 16171800 191A1B00 1C1D1E00 1F202100 22232400 25262700 28292A00 2B2C2D00 2E2F3000	02000000 02000000 08090A00 0E0F1000 20212200 26272800
in.data out.data 2 4	04000000 04000000 01020300 04050600 07080900 0A0B0C00 0D0E0F00 10111200 13141500 16171800 191A1B00 1C1D1E00 1F202100 22232400 25262700 28292A00 2B2C2D00 2E2F3000	02000000 04000000 02030400 08090A00 0E0F1000 14151600 1A1B1C00 20212200 26272800 2C2D2E00

Вариант 5. Выделение контуров. Метод Робертса.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. $w \cdot h \leq 10^8$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	03000000 03000000 04040400 03030300 07070700 0C0C0C00 12121200 03030300 1C1C1C00 1C1C1C00 00000000
in.data out.data	03000000 03000000 00000000 00000000 00000000 00000000 80808000 00000000 00000000 00000000 00000000	03000000 03000000 80808000 80808000 00000000 80808000 80808000 00000000 00000000 00000000 00000000

Вариант 6. Выделение контуров. Метод Превитта.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. $w \cdot h \leq 10^8$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	03000000 03000000 0D0D0D00 06060600 0B0B0B00 17171700 05050500 14141400 25252500 07070700 2C2C2C00

in.data	03000000 03000000	03000000 03000000
out.data	00000000 00000000 00000000	B5B5B500 80808000 B5B5B500
	00000000 80808000 00000000	80808000 00000000 80808000
	00000000 00000000 00000000	B5B5B500 80808000 B5B5B500

Вариант 7. Выделение контуров. Метод Собеля.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. $w \cdot h \leq 10^8$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data	03000000 03000000 01020300 04050600 07080900 09080700 06050400 03020100 00000000 14141400 00000000	03000000 03000000 14141400 0C0C0C00 11111100 13131300 15151500 18181800 39393900 14141400 3F3F3F00
in.data out.data	03000000 03000000 00000000 00000000 00000000 00000000 80808000 00000000 00000000 00000000 00000000	03000000 03000000 B5B5B500 FFFFFFF0 B5B5B500 FFFFFFF0 00000000 FFFFFFF0 B5B5B500 FFFFFFF0 B5B5B500

Вариант на “два”. Черно-белый фильтр.

Необходимо реализовать черно-белый фильтр на основе яркости.

Входные данные. На первой строке задается путь к исходному изображению, на второй, путь к конечному изображению. $w \cdot h \leq 10^8$.

Пример:

Входной файл	hex: in.data	hex: out.data
in.data out.data	03000000 03000000 0A141E00 28323C00 46506400 000A1400 001E2800 00323C00 00004600 00005000 00005A00	03000000 03000000 12121200 30303000 4F4F4F00 08080800 16161600 24242400 07070700 09090900 0A0A0A00