Общие требования:

- методичка (CUDA) - черновик.pdf
- реализация CPU и GPU
  - CPU - эталон (качество кода не существенно)
  - GPU (*обязательный вариант - 2*)
    - вариант 1 - без разделяемой памяти
    - вариант 2 - с разделяемой памятью
  - **1 задание = 1 ядро**
- сравнение времени работы в едином формате
  - формат: микро-, миллисекунды и т.п.
  - измерение времени на GPU **через события** в CUDA
- полное сравнение результатов работы, т.е. **никаких проверок на подобии:**
  - через сумму элементов массива
  - вывод фрагмента на экран и т.п.
- допускается отличие результатов CPU/GPU на единицу
- входное изображение:
  - 5 л.р. - в градациях серого (1 пиксель = 1 байт)
  - 6 л.р. - RGB без альфа канала (1 пиксель = 3 байта)
  - размер изображения: от 3000x3000
- тип данных: unsigned char
- использование дополнительных массивов запрещено
- **использование текстурной памяти запрещено**

- рекомендации:
  - входной/выходной формат: PGM/PPM
    - читает/пишет - IrfanView
  - допускается использование helper_image.h:
    - bool __loadPPM(const char *file, unsigned char **data, unsigned int *w, unsigned int *h, unsigned int *channels)
    - bool __savePPM(const char *file, unsigned char *data, unsigned int w, unsigned int h, unsigned int channels)
  - OpenGL опционально
  - на ЦП увеличить границы изображения

Не Допускается:

- работать с квадратным изображением
- работать с изображением кратным размеру grid и block

Балл от 0 до 0,5. Меняется линейно согласно следующим требованиям:

- работать с изображениями разного размера - *0,1 баллов*
- оптимизация кода - *до 0,1 балла*:
  - разделяемая память + корректное обращение по банкам памяти
  - формирование транзакций
  - обоснование, почему невозможно достигнуть 100% выполнения условия

- оптимизация доступа через регистры - *до 0,1 балла*
- обработать изображение больше 300 MPx - *0,2 балла*

Низкочастотный фильтр:

$$H_1 = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad H_2 = \frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad H_3 = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Высокочастотный фильтр:

$$H_1 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Оператор Собеля(не выбираем):

$$H_1 = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \qquad H_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

$$S = \sqrt{P^2 + Q^2}$$

где P и Q – отклик ядер H1 и H2 (не желательно, т.к. корень)

Оператор Превитта:

$$H_1 = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \qquad H_2 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

max{P, Q}, где P и Q – отклик ядер H1 и H2.

Оператор Лапласа:

$$H_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

Min-фильтр:

В процессе фильтрации значение текущего пикселя заменяется на минимальное значение соседних пикселей

Max-фильтр:

В процессе фильтрации значение текущего пикселя заменяется на максимальное значение соседних пикселей

Min-Max-фильтр:

В процессе фильтрации значение текущего пикселя изображения сначала заменяется на минимальное значение соседних пикселей, а при повторном проходе на максимальное

Медианный фильтр:

Усредненное фильтрование использует значения элементов, содержащихся в области примыкания, для определения нового значения. Фильтр располагает элементы области примыкания в отсортированном порядке и отбирает среднее значение.
Размер матрицы:

1. 3x3
2. 5x5

Эффект тиснения:

$$H = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$