МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**ЛАБОРАТОРНАЯ РАБОТА**

**«****Линейная фильтрация изображений (блочное разбиение). Ядро Гаусса 3x3.»**

**Выполнил:** студент группы 381506-1

Мишутин Андрей Сергеевич

Нижний Новгород

2018

**Содержание**

Оглавление

[Постановка задачи 3](#_Toc524076447)

[Метод решения 4](#_Toc524076448)

[Схема распараллеливания 5](#_Toc524076449)

[Описание программной реализации 6](#_Toc524076450)

[Подтверждение корректности 8](#_Toc524076451)

[Результаты экспериментов и анализ масштабируемости 9](#_Toc524076452)

# Постановка задачи

Фильтр Гаусса – пример матричного фильтра и является фильтром размытия. Такие фильтры используют в своем алгоритме матрицу свертки. Матрица свёртки – это матрица коэффициентов, которая «умножается» на значение пикселей изображения для получения требуемого результата. Для фильтра Гаусса используется матрица, заполненная по Гауссову закону распределения.

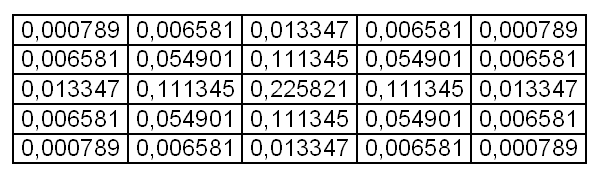


Рис. 1 Нормированная матрица свертки для фильтра Гаусса 5x5

В данной работе будут реализованы 3 способа фильтрации изображения:

* Последовательный алгоритм
* Параллельный алгоритм с использованием Open MP 2.0
* Параллельный алгоритм с использованием Intel(R) Threading Building Blocks 2018 Update 2

# Метод решения

Для начала заполняется матрица свертки. Значения ее элементов зависят от 2х параметров: sigma и радиус матрицы. Эти параметры выбираются в зависимости от того, насколько сильно необходимо размыть изображение. Чем больше радиус матрицы, тем сильнее изображение размывается. Для фильтрации необходимо пройти по каждому пикселю изображения и произвести над ним вычисления, используя ранее составленную матрицу.

Таким образом, возникает возможность для распараллеливания алгоритма.

# Схема распараллеливания

Библиотеки OpenMP и TBB используют для распараллеливания программы потоки, для которых используемая память является общей. Таким, образом каждому из потоков будет выделен блок пикселей, над которым он будет производить вычисления.

# Описание программной реализации

Руководство пользователя.

Для выполнения последовательной фильтрации изображения необходимо запустить PP\_Gauss.exe и указать параметры командной строки:

1. Номер изображения
2. Параметр sigma
3. Радиус ядра матрицы свертки

Для выполнения OMP версии программы – запустить OMP\_Gauss.exe и указать параметры:

1. Номер изображения
2. Параметр sigma
3. Радиус ядра матрицы свертки
4. Количество потоков

Для выполнения TBB версии программы = запустить TBB\_Gauss.exe и указать параметры:

1. Номер изображения
2. Параметр sigma
3. Радиус ядра матрицы свертки
4. Количество потоков

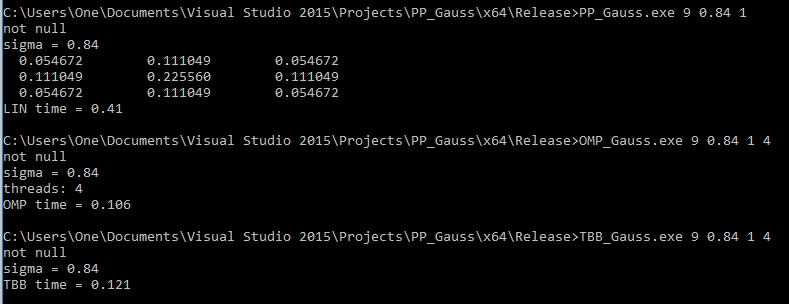


Рис. 2. Пример работы программы

Для проверки результата запускаем Checker.exe и указываем параметры:

1. Номер изображения
2. Параметр sigma

В этой программе сравнивается результат работы стандартного фильтра из библиотеки OpenCV и написанных алгоритмов с учетом небольшой погрешности. Радиус ядра по умолчанию принимается равным 3.

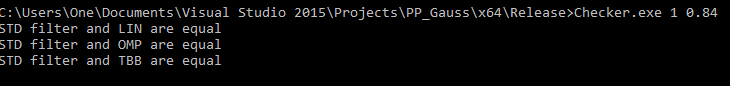


Рис. 3 Пример проверки алгоритмов

Руководство программиста.

void fillKernel(double \*\*kernel, int radius, double sigma) – заполнение ядра матрицы

int clamp(int coord, int minValue, int maxValue) – функция для предотвращения выхода за границы изображения

void gaussFilter(Mat& source, Mat& output1, int radius, double sigma) – функция фильтрации

int main(int argc, char\*\* argv) точка входа программы

bool checkRes(Mat &output, Mat &stdoutput) – функция проверки результата

# Подтверждение корректности

Для подтверждения корректности программы существует Checker.exe а также, все изображения:

* Source
* Liner filter
* OMP filter
* TBB filter
* STD filter

находятся в папке imgs в папке проекта.

# Результаты экспериментов и анализ масштабируемости

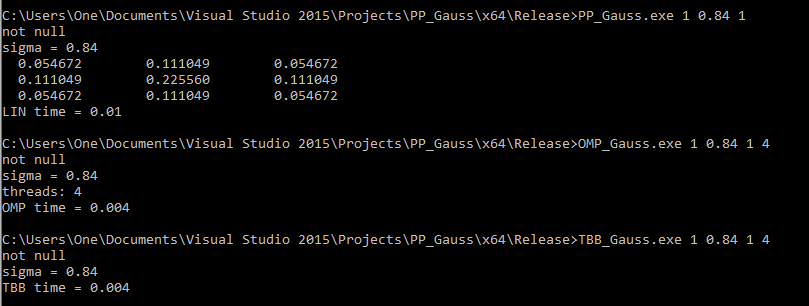


Рис. 4 Ускорение работы программы

На небольших изображениях порядка 500х500 ускорение работы программы на 4х потоков составляет всего 50%.

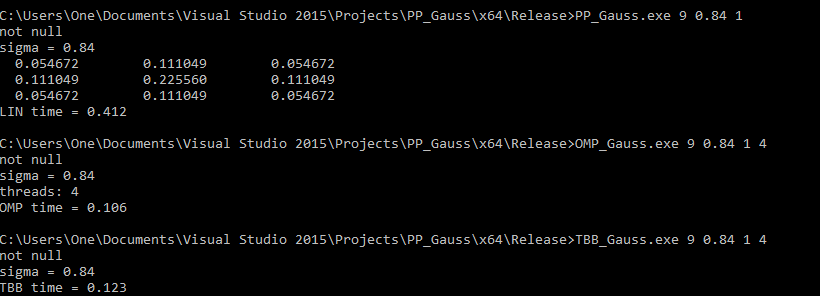


Рис. 5 Ускорение работы программы

На больших изображениях порядка 4000х400 программа ускоряется в 4 раза. Это обусловлено тем, что бОльшая часть времени занята непосредственными распределенными вычислениями.