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**

### «Повышение контраста полутонового изображения посредством линейной растяжки гистограммы»

**Выполнил:** студент группы 381506-1

Никифорова Екатерина Александровна

Нижний Новгород

2018

### Содержание

[Постановка задачи 3](#_Toc501828223)

[Метод решения 4](#_Toc501828224)

[Схема распараллеливания 5](#_Toc501828225)

[Описание программной реализации 6](#_Toc501828226)

[Подтверждение корректности 9](#_Toc501828227)

[Результаты экспериментов и оценка масштабируемости 10](#_Toc501828228)

### Постановка задачи

Контраст-это разность максимального и минимального значений яркости на изображении. Слабый контраст является одним из наиболее часто встречаемых дефектов изображений.

Гистограмма — это график, который показывает распределение тонов на фотографии. Допустим, что имеется изображение в оттенках серого, интенсивность пикселей которого изменяется в пределах значений от 0 до 255. Для изображения можно построить гистограмму со столбцами, отвечающими количеству пикселей определенной интенсивности. Такого рода гистограмма позволяет представить распределение оттенков на изображении.

Существует три основных метода повышения контраста изображения:

* линейная растяжка гистограммы (линейное контрастирование),
* нормализация гистограммы,
* выравнивание (линеаризация или эквализация, equalization) гистограммы.

В данной работе будет реализован, распараллелен и протестирован на корректность первый алгоритм.

### Метод решения

В начале работы программы мы получаем минимальное и максимальное значение интенсивности. По ним генерируется изображение в оттенках серого. Линейная растяжка сводится к присваиванию новых значений интенсивности каждому пикселю изображения. Если интенсивности исходного изображения изменялись в диапазоне от m до n, тогда необходимо линейно «растянуть» указанный диапазон так, чтобы значения изменялись от 0 до 255. Для этого достаточно пересчитать старые значения интенсивности для всех пикселей согласно формуле , где коэффициенты просто вычисляются, исходя из того, что граница m должна перейти в 0, а n – в 255.

Другими словами:

* По изображению высчитывается гистограмма;
* По гистограмме происходит поиск минимального\максимального значения интенсивности, присутствующего на изображении;
* Каждое значение пикселя меняется по формуле для получения результата.

Важно! Из алгоритма следует, что, если на изображении представлены 0 и 255 интенсивности, то растяжка произведена не будет.

### Схема распараллеливания

В OpenMP и в TBB версиях будет распараллелено одно и то же:

* Для рассчета гистограммы изображение делится между потоками и каждый поток увеличивает значение, соответствующее той, или иной интенсивности.
* Далее происходит поиск минимума\максимума для работы формулы.
* Последним этапом изображение вновь делится между потоками, которые на этот раз меняют значение каждого пикселя по формуле.

### Описание программной реализации

**Руководство пользователя**

Для генерации теста следует запустить generator.exe и ввести несколько аргументов:

* Номер теста;
* Число потоков для распараллеливания;
* Название входного файла;
* Название выходного файла;
* Нижняя граница яркости изображения;
* Верхняя граница яркости изображения;

После запуска будет выведена гистограмма сгенерированного изображения, далее будет выведена гистограмма после работы алгоритма и время работы алгоритма.

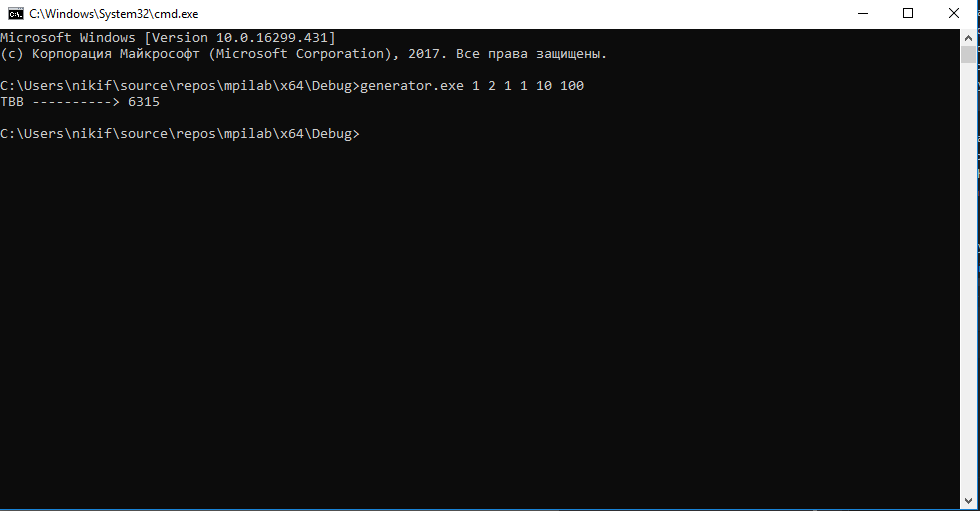
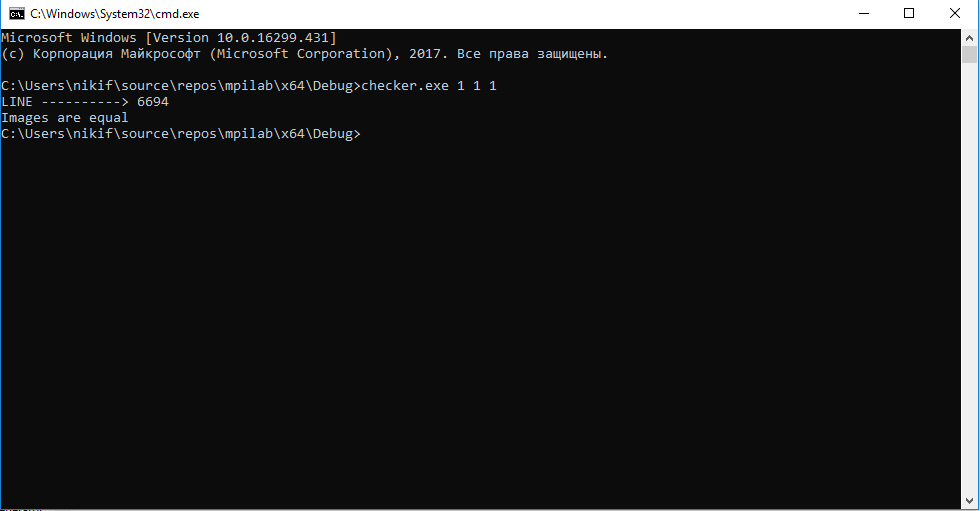


Рис. 1. Пример работы программы.

Для тестирования программы следует запустить chеcker.exe и ввести несколько аргументов:

* Номер теста;
* Название входного файла;
* Название выходного файла;

Как результат, получаем гистограмму до и после работы последовательного алгоритма (эталона), время его работы и совпадает\не совпадает ли результат параллельной версии с эталоном.

Рис. 2. Пример работы программы.

**Руководство программиста**

Для удобства тестирования программа разбита на несколько частей:

* generator - генерация тестового изображения, прогонка параллельной версии и запись результатов в бинарный файл.

|  |  |
| --- | --- |
| int main(int argc, char\* argv[]) | Параметры argc, argv – параметры, передаваемые в приложение через консоль.  Основная функция в программе. Отвечает за вызов функций, вывод результатов работы. |
| Mat generatepic(int a, int b) | Функция генерации изображения, принимает минимальное и максимальное значения интенсивности. |
| void clamp(int a, int b) | Функция преобразования введённых данных для корректной генерации. |
| bool typer(Mat pic, string buf) | Запись в бинарном формате. |

* mpilab – реализация последовательного алгоритма.

|  |  |
| --- | --- |
| void linears(Mat source, Mat result) | Сам последовательный алгоритм |
| void painthist(Mat source) | Отрисовка гистограммы |

* checker – проверка совпадения последовательной и параллельной версии.

|  |  |
| --- | --- |
| void main(int argc, char\*\* argv) | Параметры argc, argv – параметры, передаваемые в приложение через консоль.  Основная функция в программе. Отвечает за вызов функций, вывод результатов работы. |
| bool compareres(Mat fir, Mat sec) | Функция сравнения результатов. |

### Подтверждение корректности

Для проверки корректности работы алгоритма добавлена отрисовка гистограммы:

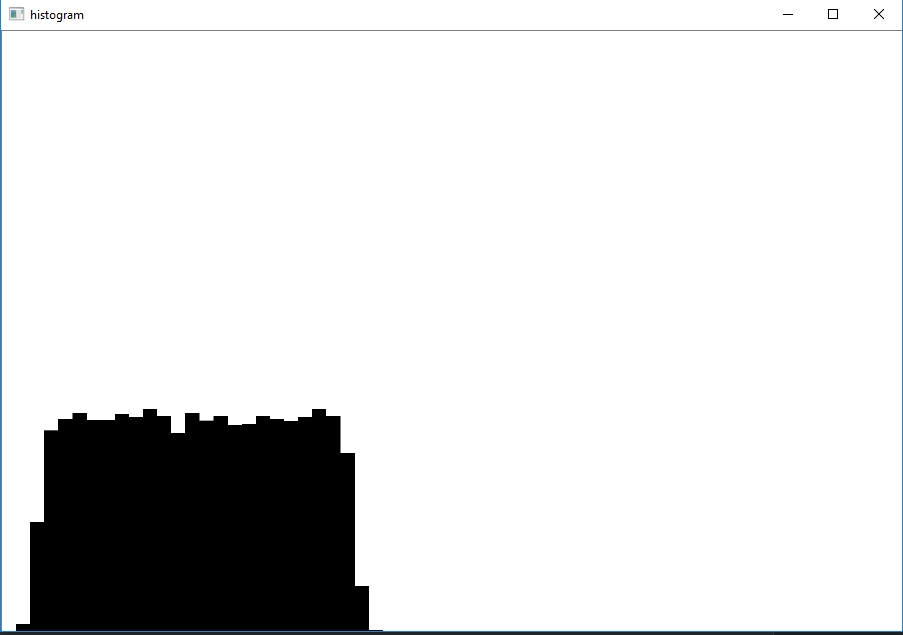


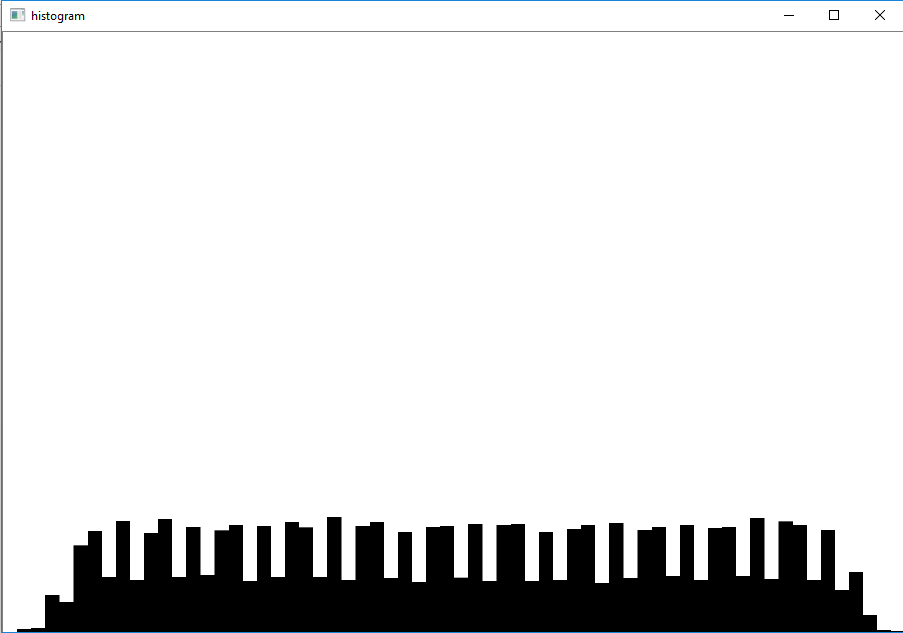
Рис. 3. Пример работы программы.

Рис. 4. Пример работы программы.

### Результаты экспериментов и оценка масштабируемости

**OpenMP**

График зависимости времени работы от количества процессов и объёма данных (ширина картинки).

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
|  |  | | | |  |  |  |  |

**OpenMP**

График зависимости времени работы от количества процессов и объёма данных (ширина картинки).

По результатам понятно, что большее ускорение достигается на бОльших по размеру изображениях.