Министерство образования РФ

ФГАУВО Нижегородский государственный университет им. Н. И. Лобачевского

Институт Информационных Технологий Математики и Механики

Отчет

По лабораторной работе №1

**Повышение контраста полутонового изображения посредством линейной растяжки гистограммы**

Выполнила:

Студентка группы

381506-3

Батанина Л. К.

Проверил:

Доцент

кафедры МОСТ

Сысоев А. В.

Н. Новгород

2018

Содержание

[Введение 3](#_Toc514399321)

[Постановка задачи 4](#_Toc514399322)

[Метод решения 5](#_Toc514399323)

[Последовательный алгоритм 5](#_Toc514399324)

[Параллельный алгоритм OpenMP 6](#_Toc514399325)

[Параллельный алгоритм TBB 7](#_Toc514399326)

[Checker 7](#_Toc514399327)

[Generator 7](#_Toc514399328)

[Результаты экспериментов 8](#_Toc514399329)

[Заключение 10](#_Toc514399330)

[Список литературы 11](#_Toc514399331)

# Введение

Распространенным недостатком ряда технических изображений является их малый контраст. Под контрастом обычно понимают разность максимального и минимального значений яркости. Примером таких изображений являются: рентгеновские изображения, изображения отпечатков пальцев, в некоторых случаях изображения получаемые с космических аппаратов и т.п. Причинами низкого контраста таких изображений могут быть: природа снимаемых объектов, неудовлетворительные условия освещения, ошибки, допущенные при определении экспозиции, высокий уровень шума, а также ряд других. Путем цифровой обработки качество изображения можно улучшить, изменяя яркость каждого элемента и увеличивая диапазон яркостей.

Данная работа содержит описание реализации алгоритма линейной растяжки гистограммы для полутонового изображения.

# Постановка задачи

Цель данной работы – реализация параллельного алгоритма повышения контраста полутонового изображения с использованием технологий OpenMP и TBB. Результатом работы программы должно стать изображение, полученное из заданного с помощью растяжки гистограммы. Для этого необходимо разработать последовательный и параллельный алгоритм повышения контраста. Затем сравнить время выполнения на нескольких потоках и определить ускорение. Программа должна использовать интерфейс OpenMP и TBB для эффективной работы на системах с общей памятью.

**Выполнение задачи включает:**

1. Освоение темы (постановка задачи).
2. Изучение метода решения.
3. Разработку тестовой версии, включающей solver, generator, checker и набор тестов для проверки корректности параллельных версий.
4. Разработку OpenMP версии.
5. Разработку TBB версии.
6. Проведение экспериментов на 1, 2, 4 и 8 потоках.
7. Подготовку отчета с анализом результатов экспериментов.

**В программе необходимо:**

1. Выполнить проверку совпадения результатов последовательной и параллельной реализаций (checker).
2. Продемонстрировать корректность работы алгоритма на задаче/задачах малой размерности.
3. Обеспечить генерацию данных для задач произвольной размерности (generator).
4. Вывести время работы последовательного и параллельного алгоритмов. Ускорение параллельного алгоритма.

# Метод решения

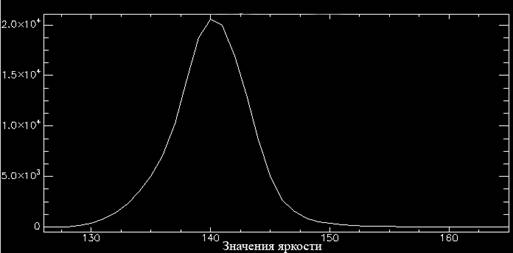
## Последовательный алгоритм

Последовательный алгоритм л*инейной растяжки гистограммы* сводится к присваиванию новых значений интенсивности каждому пикселю изображения. Если интенсивности исходного изображения изменялись в диапазоне от до , 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 должна перейти в 0, а – в 255.

Рассмотрим пример. Исходное изображение с минимальным значением яркости – 126 и максимальным – 165.

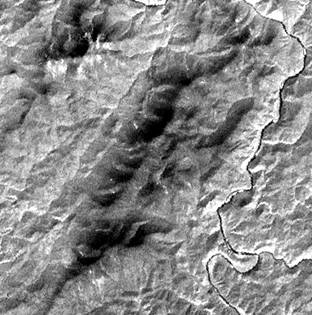


1. Исходное изображение.

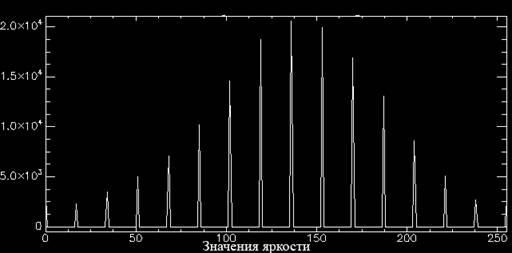


1. Гистограмма исходного изображения.

После линейного растяжения гистограммы получим изображение с диапазоном яркости от 0 до 255:



1. Полученное изображение.



1. Гистограмма полученного изображения.

## Параллельный алгоритм OpenMP

В параллельной реализации алгоритма производится распараллеливание функции по повышению контраста изображения void ompLinearCorection(const cv::Mat& input, cv::Mat& output, int num\_threads = 1). Сначала устанавливается количество потоков: omp\_set\_num\_threads(num\_threads). Затем картинка делится между потоками и обрабатывается параллельно, для этого используется директива #pragma omp parallel for, которая сообщает, что при выполнении цикла for в параллельном регионе итерации цикла должны быть распределены между потоками группы. При проведении численных экспериментов было выяснено, что наибольшая производительность достигается при управляемом планировании: schedule(guided). Для всех потоков общими данными являются: количество строк и столбов входного изображение, входное и выходное изображения: shared(rows, cols, input, output).

## Параллельный алгоритм TBB

В параллельной реализации алгоритма производится распараллеливание функции по повышению контраста изображения void tbbLinearCorection(const cv::Mat& input, cv::Mat& output, int num\_threads = 1). Сначала устанавливается количество потоков: tbb::task\_scheduler\_init init(num\_threads). При выполнении обхода изображения число итераций заранее определено, поэтому для распараллеливания цикла for используется шаблонная функция tbb::parallel\_for. Используется двумерное итерационное пространство tbb::blocked\_range2d, задающее двумерный полуинтервал [0, rows)×[ 0, cols).

## Generator

Берется одно из заранее подготовленных изображений и создаётся 2 бинарных файла, хранящих:

* 1. исходное изображение;
  2. результат применения последовательного алгоритма.

## Checker

К входному бинарному файлу применяется параллельный алгоритм. Затем полученные данные сравниваются с выходным бинарным файлом.

# Результаты экспериментов

Эксперименты проводились на 64-разрядной 4-ядерной системе с процессором Intel Core i5.

|  |  |  |  |
| --- | --- | --- | --- |
| **Количество потоков** | **Последовательная версия** | **OpenMP версия** | **TBB версия** |
| **1** | 171 ms | - | - |
| **2** | - | 95 ms | 94 ms |
| **4** | - | 60 ms | 43 ms |
| **8** | - | 55 ms | 38 ms |

* 1. Время работы программы при проведении эксперимента на изображении 14МБ.

|  |  |  |
| --- | --- | --- |
| **Эффективность:** | **OpenMP версия** | **TBB версия** |
| **p = 2** | 90,0% | 91,0% |
| **p = 4** | 71,3% | 99,4% |
| **p = 8** | 38,9% | 56,3% |

* 1. Эффективность схем распараллеливания для изображения 14МБ.

|  |  |  |  |
| --- | --- | --- | --- |
| **Количество потоков** | **Последовательная версия** | **OpenMP версия** | **TBB версия** |
| **1** | 85 ms | - | - |
| **2** | - | 49 ms | 50 ms |
| **4** | - | 40 ms | 32 ms |
| **8** | - | 33 ms | 30 ms |

* 1. Время работы программы при проведении эксперимента на изображении 9МБ.

|  |  |  |
| --- | --- | --- |
| **Эффективность:** | **OpenMP версия** | **TBB версия** |
| **p = 2** | 86,7% | 85,0% |
| **p = 4** | 53,1% | 66,4% |
| **p = 8** | 32,2% | 35,4% |

* 1. Эффективность схем распараллеливания для изображения 9МБ.

Проведя серию экспериментов, можно сделать вывод, что TBB версия работает быстрее, чем OpenMP версия. Любая параллельная версия работает быстрее последовательной. Из-за того, что эксперименты проводились на изображениях 6000×4000, то наибольшая эффективность использования параллельного алгоритма наблюдается на 2 и 4 потоках.

# Заключение

По результатам проведенных экспериментов можно сделать следующий вывод. Для повышения контраста полутонового изображения посредством линейной растяжки гистограммы эффективнее всего использовать TBB версию. На достаточно больших изображениях эффективность TBB версии с 4 потоками становится близкой к теоретической в наилучшем случае. Также на время работы программы влияют входные данные. Если в исходном изображении минимальная яркость – 0, а максимальная – 255, то изображение не будет изменено.

# Список литературы

1. Введение в разработку мультимедийных приложений с использованием библиотек OpenCV и IPP: [ https://www.intuit.ru/studies/courses/10621/1105/info].
2. Улучшение изображения путем изменения контраста: [http://nrcgit.ru/aster/methods/contrast.htm].