Министерство образования и науки Российской Федерации

Государственное образовательное учреждение высшего профессионального образования «Нижегородский государственный университет   
им. Н.И. Лобачевского»

**Институт информационных технологий, математики и механики**

**Кафедра: программная инженерия**

Специальность (направление): Программная инженерия

**Отчет**

по самостоятельной работе

по дисциплине «Параллельное программирование»

тема:

**«Выделение ребер на изображении с использованием оператора Собеля»**

**Выполнила:** студентка группы 381508

Виноградова Екатерина

Нижний Новгород  
2018

Оглавление

[**Введение** 3](#_Toc514689437)

[**Постановка задачи** 4](#_Toc514689438)

[**Вспомогательные элементы** 5](#_Toc514689439)

[Typer 5](#_Toc514689440)

[Viewer 5](#_Toc514689441)

[Generator 5](#_Toc514689442)

[Checker 5](#_Toc514689443)

[**Метод решения** 6](#_Toc514689444)

[Последовательный алгоритм 6](#_Toc514689445)

[Параллельный алгоритм 7](#_Toc514689446)

[**Результаты экспериментов** 8](#_Toc514689447)

[**Вывод** 9](#_Toc514689448)

# **Введение**

Основной целью выделения ребер на изображении является фиксация важных событий и изменений мира. Они могут отражать различные предположения о модели формирования изображения, изменения в яркости изображения могут указывать на:

* изменения глубины;
* изменения ориентации поверхностей;
* изменения в свойствах материала;
* различие в освещении сцены.

Границы имеют довольно важное значение во многих приложениях обработки изображений, особенно в системах машинного зрения, которые анализируют сцены искусственных объектов при фиксированном освещении.

Постановка задачи

В данной работе необходимо реализовать последовательный алгоритм и параллельные алгоритмы с использованием библиотек OpenMP и TBB, позволяющие выделить ребра на изображении, используя оператор Собеля.

**Для выполнения задачи необходимо**:

* Разработать тестовую версию, включающую в себя:
  + solver - решение задачи последовательным алгоритмом, по возможности, максимально простым;
  + generator - программа для генерации набора тестовых данных и их сохранения в бинарные файлы;
  + checker - программа для проверки корректности параллельных версий;
  + набор тестов (не менее десяти, из которых не менее пяти "большие") для проверки корректности параллельных версий.
* Разработать OpenMP версию
* Разработать TBB версию

Вспомогательные элементы

### **Typer**

Используя библиотеку OpenCV, создает массив, состоящий из целочисленных значений яркости пикселей изображения, а затем записывает этот массив в бинарный файл.

### **Viewer**

Читает массив из бинарного файла, а затем, используя OpenCV, генерирует изображение на основе данных из полученного массива.

### **Generator**

Генератор тестов. Создает бинарный файл и записывает в него тестовые данные.

Возможные входные параметры:

* <имя бинарного файла> <имя изображения> - с помощью typer преобразует указанное изображение в тестовые данные и записывает в указанный бинарный файл
* <имя бинарного файла> -<тип изображения> <размер изображения> <размер элемента> - генерирует изображение указанного размера с элементом указанного типа и размера (это может быть квадрат, линия или градиент)

### **Checker**

Проверяет корректность работы алгоритма, сравнивая результаты работы алгоритма на тестовых изображениях с готовыми ответами и выдает вердикт:

* AC = Accepted = Решение выдаёт корректный результат на данном тесте
* WA = Wrong Answer = Решение выдаёт некорректный результат на данном тесте
* PE = Presentation Error = Ошибка формата выходных данных

Метод решения

Оператор Собеля используется для выделения ребер на изображении. Оператор вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина ее изменения в этом направлении. Результат показывает, насколько “резко” или “плавно” меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы.

Точнее, оператор использует значения яркости только в окрестности 3×3 каждого пиксела для получения приближения соответствующего градиента изображения.

Формула оператора Собеля:

Gx и Gy — две матрицы, где каждая точка содержит приближенные производные по x и по y. Они вычисляются следующим образом путем умножения матрицы Gx и Gy и суммированием обоих матриц, в результате полученный результат записывается в текущие координаты x и y в новое изображение:

Матрицы Gx и Gy:

где А – исходное изображение.

### **Последовательный алгоритм**

Алгоритм перебирает каждый пиксель изображения. Для каждого пикселя он перебирает его окрестность и в каждой из точек окрестности прибавляет к результирующему значению пикселя яркость точки окрестности, умноженную на значение матрицы Gx. То же самое делается для матрицы Gy. Затем из двух полученных значений по вышеуказанной формуле вычисляется результирующее значение пикселя.

### **Параллельный алгоритм**

Исходный и результирующий массивы являются общими для всех потоков. Каждый поток обрабатывает свою часть массива, используя общие исходные данные и записывает результат в общий результирующий массив. Это необходимо, потому что для вычисления результата нужны значения соседних пикселей, не все из которых являются соседними в массиве.

Результаты экспериментов

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Библиотека | 1 поток, сек. | Параллельный алгоритм | | | | | | | |
| 2 потока | | 4 потока | | 8 потоков | | 16 потоков | |
| T, сек | ускорение | T, сек | ускорение | T, сек | ускорение | T, сек | ускорение |
| OpenMP | 7,925483 | 4,19303 | 1,8901565 | 2,478733 | 3,1973928 | 1,845617 | 4,2942187 | 1,857392 | 4,2669953 |
| TBB | 8,1912 | 4,16779 | 1,9653581 | 2,564049 | 3,1946347 | 1,82285 | 4,4936226 | 1,813678 | 4,5163474 |

Для вычисления ускорения параллельных алгоритмов было использовано тестовое изображение 20. Из полученных данных можно заключить, что параллельные алгоритмы работают гораздо быстрее последовательной версии. Наилучшее время работы достигается при распараллеливании на 16 потоков с помощью TBB.

Вывод

В результате работы была написана программа, в которой была реализована как последовательная, так и параллельная версия обработки изображения при помощи оператора Собеля.

В результате замера времени работы различных версий алгоритма было выяснено, что параллельные версии с использованием библиотек OpenMP и TBB работают быстрее последовательной версии во всех случаях.