

Абрамов И.А., Кравченко Е.Н. Многопоточная реализация локальной фильтрации изображений. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XVI Междунар. научно-техн. конф. – Пенза: ПДЗ, 2016. – С. 129-132.

УДК 004.42

МНОГОПОТОЧНАЯ РЕАЛИЗАЦИЯ ЛОКАЛЬНОЙ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ

И.А. Абрамов, Е.Н. Кравченко

MULTI-THREADED IMPLEMENTATION OF LOCAL IMAGE FILTERING

I.A. Abramov, E.N. Kravchenko

Аннотация. Рассматривается один из подходов ускорения фильтрации изображения за счёт её многопоточного выполнения, приводятся показатели ускорения параллельной программы фильтрации для некоторых размеров графического изображения.

Ключевые слова: фильтрация изображений, потоки, многопоточное программирование, ускорение алгоритма.

Abstract. The article deals with one of the approaches accelerate the filtration image by its multi-threaded, shows the performance speedup of a parallel program filtering for some sizes of graphics.

Keywords: image filtering, threads, multithreaded programming, acceleration of the algorithm.

В процессе обработки графической информации одним из широко используемых методов является цифровая фильтрация изображения, цель которой – выявление плохо различимых деталей или подчёркивание интересующих характеристик на исходном изображении.

Двумерная природа изображений позволяет представлять их в виде матрицы элементов с неотрицательными значениями, каждый из которых соответствует значению яркости пиксела. Для фильтрации изображения используются двумерные фильтры, отвечающие двумерной природе самого изображения. Выбирается небольшой прямоугольный участок изображения, и на нём определяется некоторая функция. Этот участок называют апертурой, а заданную в нём функцию – весовой функцией.

Чтобы можно было однозначно указать на центральный элемент, линейные размеры апертуры, как правило, берутся нечётными (3×3 , 5×5 , ...). В процессе фильтрации происходит перемещение апертуры по изображению, и в каждом положении выполняются однотипные действия, которые и определяют отклик фильтра. В каждом положении апертуры весовая функция поэлементно умножается на значение яркости соответствующих пикселов изображения в области, покрытых апертурой, после чего произведения суммируются. Вычисленное таким образом значение делится на нормирующий коэффициент, представляющий собой сумму всех элементов весовой функции, после этого полученная величина присваивается соответствующему пикселу нового изображения, который соответствует положению центра апертуры.

В случае апертуры с размерами $Mp \times Mq$ дискретных элементов можно положить, что $Mp = 2a + 1$ и $Mq = 2b + 1$, где a и b – неотрицательные числа. Обозначение $w(p, q)$ будет представлять собой текущий элемент апертуры, где $p = -a \dots a$ – текущая строка; $q = -b \dots b$ – текущий столбец [1].

Текущее положение условного центра на исходном изображении f можно обозначить через $f(i, j)$. Отклик фильтра присваивается той же точке $g(i, j)$ нового, профильтрованного поля g . Фильтрация изображения f , имеющего размеры $N \times M$, с помощью апертуры размером $Mp \times Mq$ задаётся выражением общего вида

$$g(i, j) = \sum_{p=-a}^a \sum_{q=-b}^b w(p, q) f(i + p, j + q),$$

где $a = (Mp - 1)/2$, $b = (Mq - 1)/2$.

Фильтрация изображения требует выполнения множества однотипных операций, определяющих отклик фильтра. Этот факт свидетельствует о наличии параллелизма по данным, и как результат, распараллеливание может свестись к распределению данных между потоками, каждый из которых сможет независимо выполнить фильтрацию своей части изображения. Для этого исходное изображение предполагается разбить на горизонтальные полосы и выделить каждому потоку равное количество строк [2]. После этого в процессе многопоточной фильтрации изображения вычислительная нагрузка распределится между потоками программы поровну, при этом каждый поток в случае записи результата работы будет обращаться только к выделенным ему строкам.

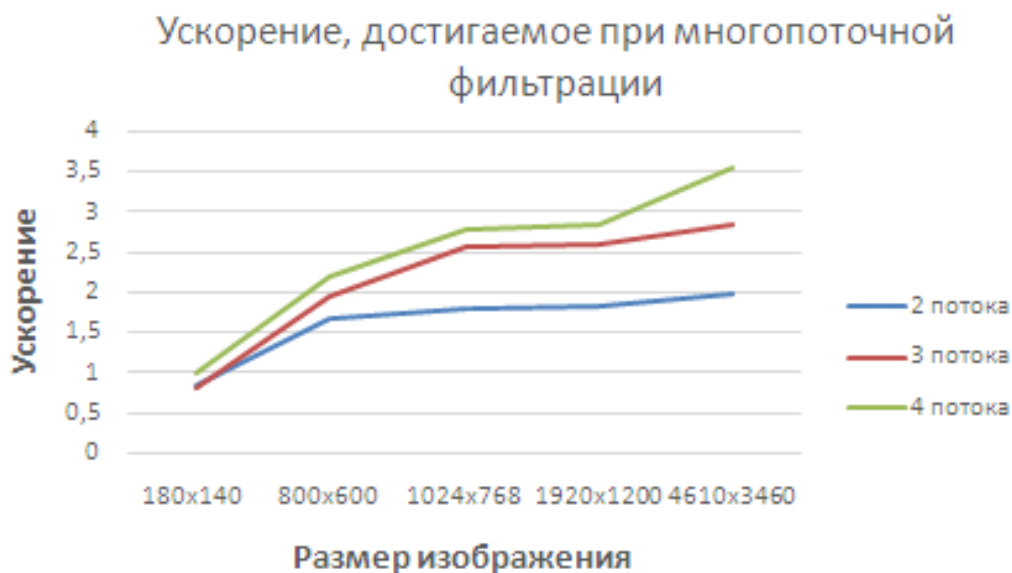
Рассматриваемый алгоритм обладает хорошей «локализацией вычислений», то есть каждый поток использует данные только выделенной ему полосы, между потоками отсутствует обмен данными и не возникает необходимость в синхронизации доступа к данным.

В процессе программной реализации алгоритма была использована стандартная поддержка многопоточности языка C++, которая появилась в стандарте C++11 [3]. Разработанная программа реализована на языке C++ и скомпилирована с помощью компилятора gcc5.4.0. Эксперименты с программой проводились на компьютере на базе процессора IntelCorei7, 3,5 ГГц, 8 Гб ОЗУ под управлением операционной системы Ubuntu 16.04. В эксперименте были исследованы однопоточный, двухпоточный, трёхпоточный и четырёхпоточный варианты выполнения программы.

В процессе эксперимента были обработаны с помощью апертуры 3×3 изображения размерами 180×140 , 800×600 , 1024×768 , 1920×1200 , 4610×3460 . Результаты вычислительных экспериментов для фильтрации изображения, где осуществляется обработка границ в соответствии с P -схемой, приведены в таблице и в графическом виде представлены на рисунке. Время выполнения фильтрации указано в миллисекундах.

Результаты вычислительных экспериментов для многопоточной фильтрации изображения

Размер изображения	Последовательный алгоритм, время	Параллельный алгоритм					
		2 потока		3 потока		4 потока	
		Время, мс	Ускорение	Время, мс	Ускорение	Время, мс	Ускорение
180×140	1,74	2,10	0,832	2,16	0,835	1,74	1,00
800×600	37,44	22,44	1,667	19,14	1,956	17,04	2,197
1024×768	62,43	34,89	1,790	24,28	2,571	22,54	2,770
1920×1200	173,55	94,71	1,833	66,27	2,619	61,16	2,838
4610×3460	1240,14	628,51	1,973	434,67	2,853	348,76	3,556



Зависимость ускорения многопоточной фильтрации от размера изображения

Как видно из представленных результатов, максимальное ускорение программы достигается при фильтрации изображения размером 4610×3460. Также несложно заметить, что при фильтрации изображения размером 180×140 ускорение, достигаемое при использовании четырёх потоков, отсутствует, а в случаях двух и трёх потоков лучшая эффективность наблюдается у последовательного варианта программы. Такие результаты являются следствием большой доли временных затрат на организацию параллелизма по отношению ко времени вычислений. С увеличением размеров изображения число вычислительных операций возрастает, и дополнительные накладные расходы в меньшей мере вносят вклад в общее время выполнения программы, при этом больше проявляется эффект от параллельного выполнения фильтрации.

Исходя из полученных в ходе эксперимента результатов можно заключить, что многопоточный подход к реализации алгоритма локальной фильтрации эффективен при обработке изображений больших размеров, где временные вычислительные затраты значительно превышают накладные расходы, связанные с организацией многопоточности.

Библиографический список

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
2. Гергель В.П. Современные языки и технологии параллельного программирования: учебник / предисл.: В. А. Садовничий. М.: Изд-во Московского университета, 2013. 408 с.
3. Уильямс Э. Параллельное программирование на С++ в действии. Практика разработки многопоточных программ / пер. с англ.: А.А. Слинкин. М.: ДМК Пресс, 2012.

Абрамов Игорь Анатольевич
Пензенский государственный
университет, г. Пенза, Россия

Кравченко Елена Николаевна
Пензенский государственный
университет, г. Пенза, Россия
E-mail: kravchencoll@yandex.ru

Abramov I.A.
Penza State University,
Penza, Russia

Kravchenko E.N.
Penza State University,
Penza, Russia