

технологии. 2008. Т. 13. Специальный выпуск 3: Избранные доклады Международной конференции и школы молодых ученых по вычислительно-информационным технологиям для наук об окружающей среде, 14–25 июля 2007 г., Томск. – С. 64–69.

2. Ботыгин И.А., Попов В.Н. Архитектура распределенной файловой системы // Интернет-журнал «НАУКОВЕДЕНИЕ» 2014. № 6 <http://naukovedenie.ru/PDF/137TVN614.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/137TVN614.

3. Метеорологические коды. Режим доступа: <http://moryak.biz/modules.php?name=Content&pa=showpage&pid=375> (дата обращения 13.03.2015).

УДК 004

РЕАЛИЗАЦИЯ АЛГОРИТМА ЛИНЕЙНОЙ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ CUDA

М.В. Демешко, А.Ю. Дёмин

*Научный руководитель: А.Ю. Дёмин, к.т.н., доцент кафедры ИПС ИК ТПУ
Томский политехнический университет*

This article describes basic principles of linear image filtering and gives basic definitions of such area. There is an overview of existing methods of filtering. Article is generally devoted to the particular case of parallel realization of the algorithm of linear filtering using CUDA.

Keywords: CUDA, Nvidia, GPU, parallel programming, linear filtering.

Ключевые слова: линейная фильтрация, параллельное программирование, графический процессор.

CUDA – это архитектура параллельных вычислений фирмы Nvidia, с помощью которой достигается значительное увеличение вычислительной производительности. Эта технология основана на использовании графических процессоров (GPU). Платформа параллельных вычислений обеспечивает набор расширений для языков C и C++ [4].

Под фильтрацией изображений понимается уменьшение действия помех, влияющих на конечное изображение в процессе его формирования [1]. Природа помех зависит от различных факторов, однако, любые помехи препятствуют как визуальному анализу изображений, так и их машинной автоматизированной обработке. Поэтому фильтрация изображений зачастую является предварительным этапом различных способов обработки изображений.

Фильтрация изображений заключается в вычислении новых значений яркости или цветов для каждого пикселя исходного изображения. Таким образом, результатом фильтрации является изображение, имеющее размер исходного изображения, однако, отличающееся от него вследствие изменений, произведенных по определенным правилам. Эти правила определяются алгоритмом фильтрации. Необходимо отметить, что значения яркости или цвета результирующего изображения признаются в наименьшей степени искаженными.

Пространственная фильтрация имеет в своей основе окрестностную обработку изображений. Окрестностью точки изображения будем называть некоторое множество соседних для неё точек. В различных алгоритмах фильтрации используются разные типы окрестностей. Выбор используемой окрестности определяется используемой моделью изображения, моделью полезного сигнала и помех.

Различают казуальную и неказуальную фильтрацию изображений. Названные типы отличаются соотношениями положения текущей точки и точек, входящих в её окрестность. Если ни одна из координат всех точек окрестности не превышает соответствующей координаты текущей точки, то окрестность будет считаться казуальной. Окрестность, не удовлетворяющая условию казуальности, будет считаться неказуальной. Простейшим примером такой

окрестности будут восемь точек, являющиеся восьмисвязными для текущей точки. Именно такая окрестность используется в данной работе.

Если фильтр использует окрестность, состоящую из большого числа точек, то принцип рациональности фильтрации предписывает вводить понятие веса точки. Весом будем называть степень влияния точек, входящих в окрестность, на решения, принимаемые фильтром относительно будущего цвета текущей точки.

Линейная фильтрация сводится к применению линейного оператора к каждой точке изображения. Другими словами, значение цвета каждого пикселя, входящего в окрестность текущей точки, умножается на соответствующий весовой коэффициент, а затем полученные произведения суммируются. Полученная сумма считается значением цвета соответствующего пикселя результирующего изображения [2].

Очевидно, что число весовых коэффициентов определяется количеством точек в используемом типе окрестности. Так, например, рассматриваемая неказуальная окрестность размера $n \times n$ потребует n^2 коэффициентов. Весовые коэффициенты принято формировать в виде матрицы, которую называют маской или ядром фильтра [2].

Размерность маски также зависит от используемого алгоритма, однако, чаще всего используются маски нечетных размеров, так как они, в отличие от четноразмерных, имеют выраженную центральную точку. Маска размерности 1×1 считается тривиальной и не рассматривается. В данной работе используется маска 3×3 .

Самой простой разновидностью линейной фильтрации в пространственной области является реализация вычисления значения математического ожидания, вычисленного по всем точкам, входящим в окрестность [2]. Другими словами, для реализации такого фильтра с использованием ядра фильтра размера 3×3 необходимо установить значения всех элементов матрицы равными $1/9$. Необходимо отметить, что для такого случая все точки окрестности, включая непосредственно текущую точку, будут обладать одинаковым весом.

Непосредственно параллельная реализация алгоритма сводится к формированию трёх векторов, хранящих в себе информацию о компонентах цветовой составляющей исходного изображения. Затем вектора передаются на графический процессор с помощью базовых функций CUDA API, где с каждым из них производятся операции, соответствующие воздействию линейного фильтра.

Рассмотрим результат работы программы для изображения с зашумлением в 20 %.



Рис. 1. Фрагмент исходного изображения



Рис. 2. Фрагмент изображения после применения фильтра с ядром с элементами одинаковых весов

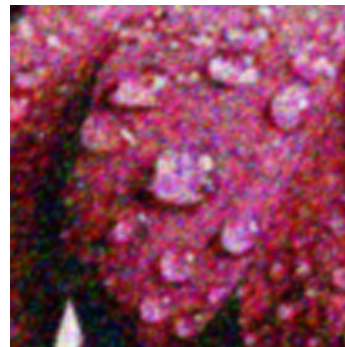


Рис. 3. Фрагмент изображения после применения фильтра с ядром с элементами различных весов

В примерах на рис. 2 и рис. 3 использовались, соответственно, следующие ядра:

$$M_1 = \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix}; M_2 = \begin{pmatrix} 1/28 & 1/7 & 1/28 \\ 1/7 & 2/7 & 1/7 \\ 1/28 & 1/7 & 1/28 \end{pmatrix}$$

Коэффициенты ядра M_2 позволяют точкам, являющимся четырехсвязными для текущей точки, оказывать на результат большее влияние, чем точкам, являющимся восьмисвязными. Сама же текущая точка оказывает наибольшее влияние на результат.

Использование ядра с различными весами позволяет сохранить изображение более резким, чем в случае с одинаковыми для всех точек весами. Так, визуальное сравнение полученных изображений позволяет увидеть, что на изображении, полученном с использованием одинаковых коэффициентов, яснее наблюдается эффект «расфокусировки».

Список литературы

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – Москва: Техносфера, 2012. – 110 с.
2. Дёмин А.Ю. Основы компьютерной графики: Учебное пособие – Томск: Издательство ТПУ, 2011. – 191 с.
3. Дёмин А.Ю., Дорофеев В.А. Распараллеливание алгоритма выделения границ объектов на основе структурно-графического представления // Известия Томского политехнического университета. – 2013. – № 5.
4. Parallel Forall. The Massively Parallel Programming Blog. URL: <http://devblogs.nvidia.com/parallelforall/> (дата обращения: 14.03.2015).
5. Siddhartha Mukherjee. Application of parallel algorithm approach for performance optimization of oil paint image filter algorithm. – Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.2, April 2014.

УДК 004

РАЗРАБОТКА КЛИЕНТА ДЛЯ МОНИТОРИНГА СЕТЕВОГО ОБОРУДОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА SNMP В WINDOWS 7

С.Г. Фролов, В.А. Дорофеев

Научный руководитель: В.А. Дорофеев, ст. преподаватель каф. ИПС ИК ТПУ

Simple Network Management Protocol – network protocol which allows to monitoring and managing network devices, such as routers, switches and computers with SNMP-agents. This article about prototype of program which uses SNMP to get some information about clients, such as system uptime, IP addresses, etc.

Keywords: SNMP, monitoring devices, network protocol, developing program.

Ключевые слова: SNMP, мониторинг оборудования, сетевой протокол, разработка приложения.

SNMP (*Simple Network Management Protocol*, рус. «Простой Протокол Сетевого Управления») – протокол сетевого управления, созданный с целью управлять большим количеством сетевых устройств. За время своего существования протокол набрал большую популярность и стал международным стандартом. Кроме управления устройствами, SNMP позволяет производить мониторинг оборудования, так как с помощью данного протокола можно получать различную информацию от любых сетевых устройств, будь то маршрутизатор, коммутатор или сетевой компьютер, в котором есть поддержка данного протокола и запущен SNMP-агент. Получаемое от устройств информация может быть очень разнообразна по своему составу [1].

В терминологии протокола SNMP-менеджер – это устройство, которое посылает запросы и получает ответы, а SNMP-агент – устройство, получающее запросы от менеджера, обрабатывающее их и отсылающее обратно необходимую информацию.

SNMP определяет всего пять типов сообщений, которыми обмениваются менеджер и агент: *get-request*, *get-next-request*, *set-request*, *get-response*, *trap* [2].