

УДК 004.932

ОПТИМИЗАЦИЯ АЛГОРИТМОВ СЕГМЕНТАЦИИ С ПРИМЕНЕНИЕМ МЕТОДОВ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ СИСТЕМ АНАЛИЗА ИЗОБРАЖЕНИЙ

*Степанов В.Н., к.т.н., с.н.с. Института проблем управления им. В.А. Трапезникова РАН,
e-mail: vnstepanov@yandex.ru.*

SEGMENTATION ALGORITHMS OPTIMIZATION WITH PARALLEL COMPUTING TECHNIQUES FOR IMAGE ANALYSIS SYSTEMS

Stepanov V.N.

The paper presents an analysis of the General Processing on Graphical Processing Unit (GPGPU) in relation to image processing (segmentation) for analyzers of medical and biological micro-objects. Adaptation of processing algorithms for parallelization and increase in execution speed are shown.

Key words: parallel computing, image processing, segmentation, GPGPU.

Ключевые слова: параллельные вычисления, обработка изображений, сегментация, общие вычисления на графических процессорах.

Введение

За последнее десятилетие широкое распространение получила технология общих вычислений на графических процессорах (GPGPU – General Processing on Graphical Processing Unit). Развитие видеоадаптеров шло благодаря компьютерным играм и в определенный момент быстродействие достигло такого уровня, что позволило рассматривать видеоадаптер как самостоятельный вычислительный модуль, применяемый в любых задачах.

На данный момент второй по вычислительной мощности суперкомпьютер Cray Titan использует специализированные вычислительные блоки Tesla K20X, которые являются, по сути, бытовыми видеоадаптерами NVidia GeForce GTX Titan.

Работа с технологией GPGPU осуществляется с помощью аппаратно-зависимых и аппаратно-независимых реализаций. К первым относятся NVidia CUDA и AMD Firestream, рассчитанные на работу с видеоадаптерами соответственно NVidia и AMD. Эти реализации, как правило, позволяют достичь большего быстродействия, т.к. созданы самими разработчиками видеоадаптеров и используют все их нюансы. Поэтому аппаратно-зависимые реализации применяют обычно для очень ресурсоемких задач на уникальных аппаратных платформах, таких как суперкомпьютеры.

Для менее ресурсоемких, но более массовых задач, таких как обработка изображений, целесообразно использовать универсальные аппаратно-независимые реализации. В данной работе используется аппаратно-независимая реализация OpenCL [1, 2]. Эта реализация основана на языке C и разрабатывается некоммерческим консорциумом KhronosGroup, в который входят большинство основных производителей видеоадаптеров и центральных процессоров.

В данной работе исследования проводились на ап-

Представлен анализ технологии общих (неграфических) вычислений на графических процессорах (GPGPU) применительно к обработке изображений (сегментации) для систем анализа и обработки изображений медико-биологических микрообъектов. Приводится адаптация алгоритмов обработки для распараллеливания и повышения производительности.

паратной платформе следующей конфигурации: Процессор Intel Core i7-860, оперативная память DDR3 12Гб, видеоадаптер nVidia GeForce GTX 970. Программная среда EmbarcaderoRADStudio 10 Seattle.

Анализ эффективности технологии GPGPU для сегментации изображений на примере системы «Морфолог-Сеть»

Обработка изображений является удобной задачей для применения параллельных вычислений. Изображение состоит из пикселей и, как правило, при обработке каждый пиксель изменяется независимо от результата изменения предыдущих. То есть все пиксели можно изменять одновременно. В компьютерном анализаторе «Морфолог сеть» [3] проводится предварительная фильтрация изображения, сегментация и постобработка полученного семантического описания.

Наиболее ресурсоемкая операция – это сегментация. Уже существуют работы по применению параллельных вычислений к различным методам сегментации изображений. Например, в [4] автор адаптировал алгоритм Mean Shift к выполнению на GPU и получил прирост скорости обработки в 97 раз (на GPU NVidia GeForce GTX 470 относительно процессора Intel Core i5 750). Этот алгоритм применяется для сегментации изображений с неизвестным количеством классов. При анализе изображений медико-биологических препаратов обычно заранее известно количество классов и их признаки, поэтому алгоритм Mean Shift может привести к появлению ложных классов. В [5] показана теоретическая возможность распараллеливания генетического алгоритма сегментации трехмерных изображений. При анализе изображений медико-биологических препаратов обычно име-

ется большой объем исходных данных, помимо самого изображения, которые необходимо использовать в том числе и при сегментации. Это позволит повысить точность анализа. Поэтому целесообразно применение специализированных алгоритмов. В данной работе показана адаптация под параллельные вычисления интерактивного метода пороговой сегментации, разработанного для системы анализа изображений «Морфолог-сеть». Алгоритм интерактивной сегментации подразумевает сравнение каждого пикселя изображения с массивом опорных точек, с вычислением расстояний в трехмерных цветовых пространствах [6]. Интерактивность метода заключается не только в выборе опорных точек, принадлежащих конкретной области сегментации, но и в настройке коэффициентов расстояний, при одновременном наблюдении результатов сегментации. Для этого требуется достаточно большая частота обновления результата. Однопоточный алгоритм на центральном процессоре не может обеспечить достаточно быстрого действия и приходилось отображать результат в маленьком окне предварительного просмотра, что не очень удобно. Применение алгоритмов параллельной обработки на графическом процессоре позволит отображать результат с приемлемой частотой для всего изображения (или увеличить на порядок размер окна предварительного просмотра для очень больших изображений).

Для анализа быстродействия параллельного алгоритма сегментации был сгенерирован ряд синтетических изображений размерами 2000x2000, 4000x2000, 4000x4000 и 8000x4000 пикселей. Пример такого изображения приведен на рис. 1. Изображения создавались на основе реальных изображений препаратов предстательной железы [7].

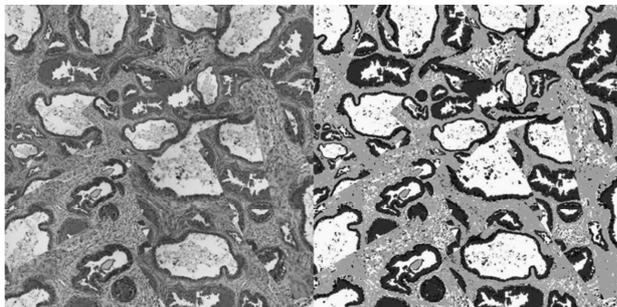


Рис. 1. Пример синтетического изображения на основе препаратов предстательной железы размером 2000x2000 пикселей для анализа производительности: а) исходное, б) сегментированное

В системе «Морфолог-сеть» все расчеты при сегментации могут вестись как в пространстве RGB , так и HSV (цвет, насыщенность, яркость). Пространство HSV лучше подходит для выделения областей на изображении, как показано в [6], но требует дополнительных вычислений.

На рис. 2 показан график увеличения времени сегментации на центральном процессоре в пространстве RGB и HSV в зависимости от размера изображения. При размере изображения 32 МРiх падение производительности достигает 24 %.

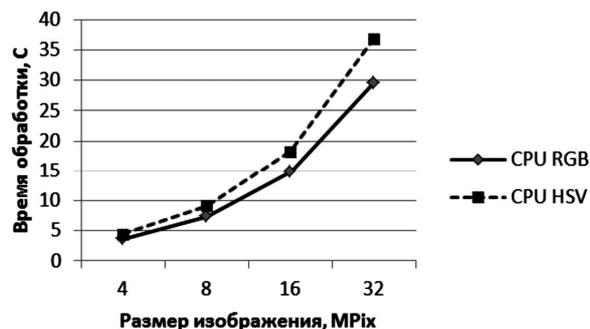


Рис. 2. График зависимости скорости расчета (с) на центральном процессоре от размера изображения (МРiх) для пространства RGB и HSV

При сегментации с помощью технологии GPGPU преобразование пространства цветов RGB к HSV производится по формулам:

$$R < B < G \left\{ \begin{array}{l} H = \frac{1}{3} - \frac{G-B}{6 \times (G-R)}; \\ S = \frac{G-R}{G}; \\ V = G; \end{array} \right.$$

$$G < R < B \left\{ \begin{array}{l} H = \frac{2}{3} - \frac{B-R}{6 \times (B-G)}; \\ S = \frac{B-G}{B}; \\ V = B; \end{array} \right.$$

$$G < B < R \left\{ \begin{array}{l} H = \frac{R-B}{6 \times (R-G)}; \\ S = \frac{R-G}{R}; \\ V = R; \end{array} \right.$$

$$B < R < G \left\{ \begin{array}{l} H = \frac{1}{3} + \frac{G-R}{6 \times (G-B)}; \\ S = \frac{G-B}{G}; \\ V = G; \end{array} \right.$$

$$B < G < R \left\{ \begin{array}{l} H = -\frac{R-G}{6 \times (R-B)}; \\ S = \frac{R-B}{R}; \\ V = R. \end{array} \right.$$

где R, G, B – соответственно красный, синий и зеленый компоненты цвета, а H, S, V – цвет, насыщенность и яркость. Все параметры могут принимать значения от 0 до 1. Это не оптимальные формулы с точки зрения математики, но они более удобны для вычисления специализированным графическим процессором по технологии GPGPU. Кроме того, при кодировании формул были

максимально широко использованы векторные операции [2], на которые графический процессор рассчитан. Это позволило сократить разрыв в быстродействии между алгоритмами с RGB и HSV до 5 %. На рис. 3 приведен аналогичный график (рис. 2), но с использованием графического процессора.

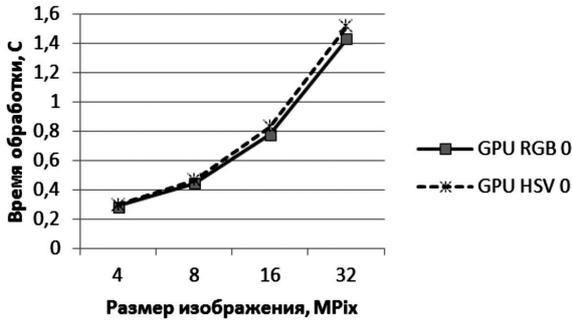


Рис. 3. График зависимости скорости расчета (с) на графическом процессоре от размера изображения (MPix) для пространства RGB и HSV

Расстояние между текущей точкой изображения и опорными точками для пространства RGB вычисляется по формуле:

$$D = \sqrt{(Rn - Ri)^2 + (Gn - Gi)^2 + (Bn - Bi)^2},$$

где Rn, Gn, Bn – красный, зеленый и синий компонент цвета текущей опорной точки, а Ri, Gi, Bi – компоненты цвета текущей точки изображения.

Для пространства HSV формула имеет вид:

$$D = \sqrt{(Xn - Xi)^2 + (Yn - Yi)^2 + (Vn - Vi)^2},$$

где Vn, Vi – значение яркости опорной точки и точки изображения, а Xn, Xi, Yn, Yi – координаты, преобразованные из полярных координат HS по формулам:

$$X = \cos(H) \times S;$$

$$Y = -\sin(H) \times S.$$

Для вычисления расстояния применена быстрая функция half_sqrt [2]. Эта функция обладает пониженной точностью, а именно 8192 ULP. ULP – это единица наименьшей точности (unit of least precision), то есть минимальное расстояние между соседними числами с плавающей точкой. Для 32 битных вычислений это 0,00000011920929. Таким образом, функция half_sqrt выдает результат с шагом $0,00000011920929 \times 8192 = 0,00097656250368$. Сравнение результатов сегментации с полной и пониженной точностью показало отсутствие различий.

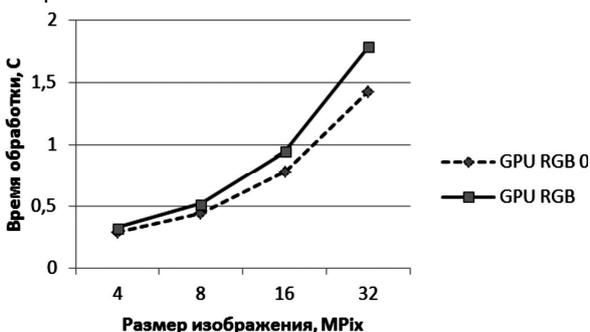


Рис. 4. График зависимости скорости расчета (с) от размера изображения (MPix) для быстрой функции (GPURGB0) и функции с полной точностью (GPURGB)

Применение быстрой функции позволяет увеличить быстродействие в среднем на 15,9 %, причем разница растет вместе с увеличением размера обрабатываемого изображения. Это объясняется тем, что при увеличении размера изображения уменьшается процент накладных расходов, связанных с подготовкой изображения, и увеличивается процент собственно вычислений.

В табл. 1 представлены все результаты изменений времени работы алгоритмов сегментации, для однопоточного алгоритма на центральном процессоре в пространстве RGB (CPURGB), параллельного алгоритма на графическом процессоре (GPURGB), быстрого параллельного алгоритма с пониженной точностью (GPURGB0), однопоточного алгоритма в пространстве HSV (CPUHSV) и быстрого параллельного алгоритма в пространстве HSV (GPUHSV 0).

Таблица 1.

	CPU RGB	GPU RGB 0	GPU RGB	CPU HSV	GPU HSV 0
4	3,658	0,292	0,329	4,567	0,301
8	7,387	0,444	0,522	9,18	0,466
16	14,772	0,778	0,947	18,402	0,83
32	29,646	1,429	1,782	36,92	1,513

Видно, что увеличения быстродействия от применения параллельных вычислений достигает 25 раз. На рис. 5 представлена информация из табл. 1 в графическом виде, три графика для GPU практически слились в один.

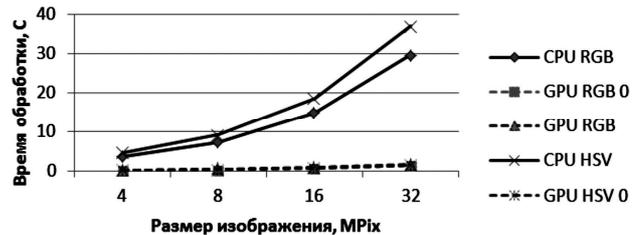


Рис. 5. График зависимости скорости расчета (с) от размера изображения (MPix)

Заключение

В результате проведенных исследований можно отметить, что обработка изображений является хорошо распараллеливаемой задачей. Применение параллельных алгоритмов с использованием технологии общих вычислений на видеоадаптере позволило ускорить сегментацию изображений до 25 раз.

Использование быстрых функций с пониженной точностью не сказывается на результате сегментации – попиксельное сравнение сегментированных изображений показало, что результат не отличается от такового при расчетах с полной точностью. Ускорение при этом составляет в среднем 15,9 %. Применение оптимизированного алгоритма преобразования пространства RGB в HSV позволило увеличить быстродействие еще на 19 %.

Таким образом, параллельная обработка на используемом в работе оборудовании (видеоадаптер NVidia GeForce 970) позволяет проводить сегментацию реальных изображений (обычно не более 1 MPix) со скоростью, приближенной к реальному времени. Задержка составляет 170 мс. Такая задержка позволяет отказаться от окна предварительного просмотра при настройке порогов сегментации, как было ранее, и выводить результат прямо на изображение.



Литература

1. AaftabMunshi; Benedict R. Gaster; Timothy G. Mattson; James Fung; Dan Ginsburg. OpenCLProgramming Guide. – Addison-WesleyProfessional, 2011.
2. AaftabMunshi. The OpenCL Specification 1.0.48. KynosGroup. 2009
3. Попова Г.М., Степанов В.Н., Дружинин Ю.О., Дятчина И.Ф. Многофункциональный информационно-вычислительный комплекс анализа и диагностики изображений // Информационные технологии и вычислительные системы. 2010. № 4. С. 25-37.
4. Димашова М.П. Реализация алгоритма сегментации изображений MeanShift на GPU // Сборник трудов X Всероссийской конференции «Высокопроизводительные параллельные вычисления на кластерных системах» 2010. Т.1 – с. 214
5. Дрынди́к Р.В., Привалов М.В. Распараллеливание эволюционных методов сегментации трехмерных медицинских изображений.// Информационные управляющие системы и компьютерный мониторинг. – Донецк :ДонНТУ, 2012. - С. 480 - 484.
6. Попова Г.М., Степанов В.Н., Дружинин Ю.О. Интерактивный метод обработки цветных изображений объектов биологической природы для получения их семантического описания // Системный анализ и управление в биомедицинских системах. 2009. Т. 8, № 3. С. 741-746.
7. Попова Г.М., Дружинин Ю.О., Степанов В.Н., Дятчина И.Ф., Чазова Н.Л., Берцанская А.М., Мельникова Н.В. Количественная диагностика онкологических заболеваний предстательной железы с использованием компьютерного анализатора «Морфолог-Сеть» // Системный анализ и управление в биомедицинских системах. 2006. Т. 5, № 4. С. 943-954.