

РЕАЛИЗАЦИЯ ЧИСЛЕННОГО ВЕЙВЛЕТНОГО ПРЕОБРАЗОВАНИЯ НА ГРАФИЧЕСКИХ АДАПТЕРАХ АРХИТЕКТУРЫ NVIDIA CUDA

А. А. Вахтин, Я. А. Туровский

Воронежский государственный университет

Поступила в редакцию 30.04.2012 г.

Аннотация. Рассматривается алгоритм численного вейвлетного преобразования с использованием архитектуры Nvidia CUDA. Благодаря возможности параллельных вычислений в CUDA, время выполнения численного вейвлетного преобразования существенно сокращается, по сравнению с реализацией этого преобразования для CPU.

Ключевые слова: вейвлетный анализ, CUDA, алгоритмы, параллельные вычисления.

Annotation. An algorithm for the numerical wavelet transform using NVidia CUDA architecture is considered. Due to the possibility of parallel computing of CUDA, run-time numerical wavelet transform is significantly reduced than the implementation of this transform for the CPU.

Key words: wavlet analiz, CUDA, algorithms, parallel computing.

Введение. Вейвлетный анализ является мощным средством анализа динамики систем и привлекает все большее внимание исследователей. Достаточно обширна его практическая область применения: исследование нестационарных сигналов, распознавание образов, сжатие аудио и видеoinформации, решение многих задач в радиотехнике, связи, электронике, ядерной физике, сейсмоакустике, метеорологии, биологии, экономике, медицине и других областях науки и техники [1].

В данной статье будет рассмотрен алгоритм и метод реализации вейвлетного преобразования, но не будет рассмотрен физический смысл вейвлетного анализа и особенности выбора вейвлетов, так как это зависит от задачи, для которой осуществляется вейвлетное преобразование и это не входит в тему данной статьи. Данные вопросы рассматриваются, например, в [1].

Алгоритм вейвлетного анализа для CPU. Непрерывное вейвлетное преобразование осуществляется по формуле:

$$W(s, t_0) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(t) \psi \left(\frac{t - t_0}{s} \right)$$

где f – исследуемая функция, ψ – функция вейвлет [1], s – масштаб вейвлетного преобразования (ширина вейвлета), t_0 – параметр сдвига.

При численной реализации вейвлетного преобразования значения функции f берутся с одинаковым интервалом времени Δt :

$$x_n = f(\Delta t \cdot n), \text{ где } n = 0, 1, \dots, N - 1.$$

Формула численного вейвлетного преобразования будет:

$$W(s, n) = \sum_{i=0}^{N-1} x_n \psi \left(\frac{(i - n)\Delta t}{s} \right)$$

Для реализации алгоритма численного вейвлетного преобразования можно воспользоваться одним важным свойством: Фурье-образ вейвлетного преобразования это произведение Фурье-образа функции \hat{f} и комплексно сопряженного фурье-образа вейвлетной функции $\hat{\psi}^*$ [1]. Таким образом, алгоритм численного вейвлетного преобразования сводится к следующим пунктам:

1. Выполнить преобразование Фурье для значений исследуемой функции x_n ($n = 0, 1, \dots, N - 1$).

2. Получить преобразование Фурье для вейвлетной функции ψ на заданном масштабе s .

3. Вычислить вейвлетное преобразование как обратное Фурье-преобразование:

$$W(s, n) = \sum_{k=0}^{N-1} \hat{x}_k \hat{\psi}^* (s\omega_k) e^{-i\omega_k n \Delta t}$$

$$\text{где } \omega_k = \begin{cases} \frac{2\pi k}{N\Delta t} & \text{? } \mathcal{B} \text{ } k \leq \frac{N}{2}, \\ -\frac{2\pi k}{N\Delta t} & \text{? } \mathcal{B} \text{ } k > \frac{N}{2}. \end{cases}$$

При этом:

1. Существуют алгоритмы быстрого преобразования Фурье, позволяющие сократить ко-

личество вычислений дискретного преобразования Фурье с $O(N^2)$ до $O(N \cdot \log(N))$ [2].

2. Существует набор вейвлетных функций, для которых были проведены исследования и получено аналитическое представление их Фурье-образов [1], что позволяет не использовать вычисление преобразования Фурье для вейвлетной функции, а получить ее значение по заданной формуле.

3. Если для вейвлетной функции не существует Фурье-образа и получить его в аналитическом виде не возможно, достаточно один раз вычислить ПФ с заданным числом N и масштабом s и сохранить его для последующих вычислений, например, в файле.

Данный алгоритм позволяет существенно повысить расчетное время вейвлетного преобразования, но, как видно из табл. 1 этого не достаточно, чтобы осуществить вейвлетный анализ для большого объема данных в режиме реального времени.

Применение графических адаптеров для параллельных вычислений. Чтобы повысить скорость вычислений можно воспользоваться технологией параллельных вычислений. Для решения данной задачи авторы воспользовались графическим адаптером NVIDIA GeForce GTX 465, который поддерживает технологию параллельных вычислений CUDA [3, 4].

CUDA (англ. Compute Unified Device Architecture) – это разработанная компанией NVIDIA программно-аппаратная архитектура параллельных вычислений, предназначенная для разработки параллельных программ и позволяющая существенно увеличить вычислительную производительность благодаря использованию графических процессоров (GPU).

В отличие от обычных процессоров CPU, процессоры GPU предназначены для параллельных вычислений с большим количеством арифметических операций. Поэтому значительно большее число транзисторов GPU работает по прямому назначению – обработка массивов данных, а не управляет исполнением (flow control) немногочисленных последовательных вычислительных потоков. Кроме того, в видеокартах NVIDIA основной блок – это мультипроцессор с восемью-десятью ядрами и сотнями ALU в целом, несколькими тысячами регистров и небольшим количеством разделяемой общей памяти. Видеокарта содержит быструю глобальную память с доступом к ней всех мультипро-

цессоров, локальную память в каждом мультипроцессоре, а также специальную память для констант (рис. 1). Поэтому вычисление на графических процессорах GPU происходит быстрее [4].

Технология программирования для CUDA заключается в предоставлении ядрам одновременно одни и те же инструкции и разные потоки данных. Такой подход позволяет увеличить количество исполнительных блоков за счёт их упрощения [3, 4]. Этот стиль программирования является обычным для графических алгоритмов и многих научных задач (в том числе, расчет вейвлетного преобразования).

Язык программирования для CUDA является расширением стандартного языка C (CUDA C). Управление вычислениями на графическом адаптере осуществляется из программы, работающей на процессоре CPU. В данной программе необходимо реализовать передачу данных в память графического адаптера и вызов соответствующих процедур, которые будут выполняться на GPU.

Для CUDA разработаны специальные библиотеки, в которых решается ряд графических и математических задач. Так, например, в библиотеке CUFFT реализован алгоритм быстрого преобразования Фурье [3, 4]. Данная библиотека была использована авторами для реализации алгоритма вейвлетного преобразования.

Алгоритм численного вейвлетного преобразования на архитектуре CUDA. Разработанный авторами алгоритм состоит из следующих этапов:

1. Используя библиотеку CUFFT [3] вычисляется преобразование Фурье для заданных значений x_n ($n = 0, 1, \dots, N - 1$).

2. Запускается на графическом адаптере процедура вычислений

$$\hat{\omega}_n^k = \sqrt{2\pi k} \hat{x}_n \hat{\psi} * \left(\frac{2\pi nk}{N} \right),$$

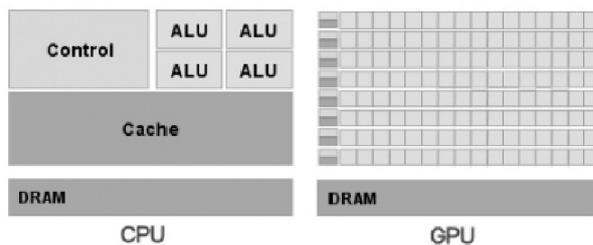


Рис. 1. Архитектура процессоров CPU и GPU

где \hat{x}_n – фурье-коэффициенты исследуемого сигнала, полученные в п. 1 ($n = 0, 1, \dots, N - 1$), $\hat{\psi}^*$ – комплексно-сопряженный Фурье-образ вейвлета, k – величина масштаба.

Значения n и k в каждом потоке, выполняемом на GPU, задаются в соответствии с их порядковым номером.

3. Для каждой полученной последовательности $\hat{\omega}_0^k, \hat{\omega}_1^k, \dots, \hat{\omega}_{N-1}^k$ вычисляется обратное преобразование Фурье. Для этого используется библиотека CUFFT.

Апробация. Из табл. 1 видно, что алгоритм вейвлетного преобразования с использованием вычислений на CUDA более чем в тысячу раз быстрее, чем алгоритм для CPU.

С помощью средств разработки программ основанных на технологии CUDA была разработана динамически компонуемая библиотека, в которой реализована процедура вейвлетного преобразования. Это позволяет использовать данную библиотеку в разработке прикладных программ, в которых необходимо вейвлетное преобразование данных в режиме реального времени.

Данная библиотека была использована для создания биомедицинских систем на основе нейро-компьютерного интерфейса, работающих в режиме реального времени. Извлечение информации о изменении или работе органов человека, применимой для формирования команд управления внешними устройствами, основано на локализованном спектральном анализе матриц вейвлет-коэффициентов [5-9]. Для проведения численного вейвлетного преобразования был выбран вейвлет WAVE, полученный из первой производной Гаусса [1]:

$$\psi(t) = te^{-\frac{t^2}{2}}$$

с Фурье-образом:

$$\hat{\psi}(t) = i \cdot te^{-\frac{t^2}{2}}$$

Несмотря на то, что вейвлет WAVE уступает вейвлету Морле в разрешении по частоте [1], его преимущество связано со значительно мень-

шей временной задержкой, вызванной лучшей его локализацией во временном пространстве.

Сигналы для обработки поступали в режиме реального времени по 21 каналу от электроэнцефалографа «Нейрон-Спектр 4-ВП» (производитель – ООО «Нейрософт»). Наибольшая частота дискретизации каждого канала – 5 кГц, разрядность АЦП – 16 бит. Таким образом, максимально в течение 1 с от прибора поступало 1.05×10^5 отсчетов по каналам ЭЭГ, 5×10^3 отсчетов по каналу дыхания, 2×10^4 отсчетов по четырем полиграфическим каналам, всего 1.3×10^5 отсчетов по 26 каналам. Помимо этого, было зарезервировано ещё 4 канала с той же частотой дискретизации для получения других сигналов от испытуемого.

Заключение. В ходе вычислительных экспериментов проведена оценка времени, необходимого для расчета матрицы вейвлет-коэффициентов с применением технологии CUDA. Представленные в настоящей статье результаты могут быть использованы для совершенствования систем параллельной обработки сигналов, характеризующих активность органов или системы органов человека, что позволит создать системы биологической обратной связи и нейро-компьютерные интерфейсы нового поколения.

СПИСОК ЛИТЕРАТУРЫ

1. Короновский А. А., Храмов А. Е. Непрерывный вейвлетный анализ и его приложения. – М.: ФИЗМАТЛИТ, 2003. – 176 с.
2. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток. – М.: Радио и связь, 1985. – 248 с.
3. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров. – М.: ДМК Пресс, 2011. – 232 с.
4. Берилло А. NVIDIA CUDA – неграфические вычисления на графических процессорах (<http://www.ixbt.com/video3/cuda-1.shtml>)
5. Изменение ритма сердца взрослого человека и плода в антенатальном периоде как самоподобный процесс / Я. А. Туровский, Е. В. Дорохов, С. В. Шамарин [и др.] // Фракталы и прикладная синерге-

Таблица 1

Время вычисления вейвлетного преобразования (сек)

	число элементов число \times масштабов					
	250 \times 128	500 \times 256	1000 \times 512	2500 \times 1024	3000 \times 2048	5000 \times 2048
CPU	4,934	8,341	23,989	85,899	135,526	235,460
CUDA	0,004	0,008	0,023	0,047	0,128	0,202

тика : тр. Первого междисциплинарного семинара. – М.:, 1999. – С. 68-69.

6. Динамика вагусных влияний на вариабельность сердечного ритма в покое / Я. А.Туровский, Е. В. Дорохов, М. В. Сысоев [и др.] // Прикладные информационные аспекты в медицине. – Воронеж, 2001. – Т. 2, № 3. – С. 25–31.

7. Вейвлетный анализ временных рядов вариабельности сердечного ритма / Я.А. Туровский, В. В. Мишин, Л.А. Битюцкая [и др.] // Известия вузов. Прикладная нелинейная динамика. – 2002. – Т. 10, № 6. – С. 16 – 23.

Вахтин Алексей Александрович – кандидат физико-математических наук, доцент кафедры программирования и информационных технологий факультета компьютерных наук; тел.: (473) 2-208-470, e-mail: alvahtin@gmail.com

Туровский Ярослав Александрович – кандидат медицинских наук, доцент кафедры цифровых технологий, руководитель лаборатории информационных технологий в медицине факультета компьютерных наук; тел.: (473) 2-208-384, e-mail: yaroslav_turovsk@mail.ru

8. Вейвлет-анализ энцефалограмм / С. Д. Кургалин, Я. А. Туровский, А. В. Максимов [и др.] // Информационные технологии в проектировании и производстве. – 2010. - № 1. – С. 89-95.

9. Моделирование медико-биологической информации на основе непрерывного вейвлет-преобразования / С.Д. Кургалин, Я.А. Туровский, А. В. Максимов [и др.] // Информатика : проблемы, методология, технологии : материалы X Международ. науч.-метод. конф., Воронеж, 11-12 февр. 2010 г. – Воронеж, 2010. – Т. 1. – С. 411–415.

Vakhtin A. A. – Candidate of sciences (physics and mathematics). Associate Professor of the Department of Programming and Information Technologies. Voronezh State University. E-mail: alvahtin@gmail.com

Turovsky Yaroslav Aleksandrovich – Candidate of Medical Science, Associate Professor of the Digital Technologies Department, Head of the Digital Medical Technologies Laboratory of Computer Science Faculty; tel.: (473) 2-208-384, e-mail: yaroslav_turovsk@mail.ru