

# Численное вейвлетное преобразование с использованием архитектуры Nvidia CUDA для обработки медицинских сигналов

А.А. Вахтин, email: alvahtin@gmail.com  
Я.А. Туровский, email: yaroslav\_turovsk@mail.ru

Воронежский Государственный Университет

**Аннотация.** Рассматривается алгоритм численного вейвлетного преобразования (ЧВП) с использованием архитектуры Nvidia CUDA. Время выполнения ЧВП с использованием параллельных вычислений в CUDA, существенно сокращается, по сравнению с реализацией этого преобразования для CPU.

**Ключевые слова:** Вейвлетное преобразование, параллельные вычисления, CUDA.

## Введение

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

В данной статье 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].

## 1. Алгоритм вейвлетного анализа

Непрерывное вейвлетное преобразование осуществляется по формуле:

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

где  $f$  – исследуемая функция,  $\psi$  – функция вейвлет [1],  $s$  – масштаб вейвлетного преобразования,  $t_0$  – параметр сдвига.

При численной реализации вейвлетного преобразования значения функции  $f$  берутся с одинаковым интервалом времени  $\Delta t$ :

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

Формула численного вейвлетного преобразования будет:

$$W_{s,n} = \sum_{k=0}^{N-1} x_k \psi \frac{k - n \Delta t}{s}$$

Для реализации алгоритма численного вейвлетного преобразования можно воспользоваться одним важным свойством: фурье-образ вейвлетного преобразования это произведение фурье-образа функции  $f$  и комплексно сопряженного фурье-образа вейвлетной функции  $\psi^*$  [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  $\psi$  на заданном масштабе  $s$ .
3. Вычислить вейвлетное преобразование как обратное фурье-преобразование:

$$W_{s,n} = \sum_{k=0}^{N-1} x_k \psi^* s \omega_k e^{-i\omega_k n \Delta t}$$

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

При этом:

1. Существуют алгоритмы быстрого преобразования Фурье, позволяющие сократить количество 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 преобразование Фурье с заданным числом  $N$  и масштабом  $s$  и сохранить его для последующих вычислений, например, в файле.

Приведенный 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.

## **2. Численное вейвлетное преобразование с использованием параллельных вычислений CUDA**

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

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]. Этот стиль программирования является обычным для многих научных задач (в том числе, расчет вейвлетного преобразования).

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

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

$$w_n^k = \overline{2\pi k} x_n \psi^* \frac{2\pi m k}{N},$$

где  $x_n$  – фурье-коэффициенты исследуемого сигнала, полученные в п. 1 ( $n = 0, 1, \dots, N - 1$ ),  $\psi^*$  – 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  $w_0^k, w_1^k, \dots, w_{N-1}^k$  вычисляется обратное преобразование Фурье. Для этого используется библиотека CUFFT [3].

### **Заключение**

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 [4-5]. Извлечение информации о изменении или работе органов человека, применимой для формирования команд управления внешними устройствами, основано на локализованном спектральном анализе матриц вейвлет-коэффициентов. 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:

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

Несмотря на то, что вейвлет WAVE уступает вейвлету Морле в разрешении по частоте [1], его преимущество связано со значительно меньшей временной задержкой, вызванной лучшей его локализацией во временном пространстве.

Для испытаний использовался сигнал, поступающий 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  $1,05 \times 10^5$  отсчетов по каналам ЭЭГ. Результаты времени вычисления вейвлетного преобразования приведены в таблице 1. Очевидно, что алгоритм вейвлетного преобразования с использованием вычислений на CUDA более чем в тысячу раз быстрее, чем алгоритм для CPU.

Таблица 1.

*Время вычисления вейвлетного преобразования (сек)*

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

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 нейрокомпьютерные интерфейсы нового поколения.

Работа выполнена при финансовой поддержке гранта 19-29-01156 мк.

### **Список литературы**

1. Короновский, А.А. Непрерывный вейвлетный анализ и его приложения / А.А. Короновский, А.Е. Храмов. – М.: ФИЗМАТЛИТ, 2003. – 176 с.
2. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свертки / Г. Нуссбаумер. – М.: Радио и связь, 1985. – 248 с.
3. Сандерс Д. Технология CUDA в примерах. Введение в программирование графических процессоров / Д. Сандерс, Э. Кэндрот. – М.: МДК Пресс, 2017. – 232 с.
4. Туровский Я.А. Системы интерфейсов человек-компьютер на основе анализа спектральных особенностей биомедицинских сигналов и гибридного интеллекта / Я.А. Туровский // Сборник трудов XIII Всероссийского совещания по проблемам управления ВСПУ-2019. Институт проблем управления им. В.А. Трапезникова РАН. 2019. – М.: ИПУ РАН, 2019. – С. 1649-1652.
5. Исследование ЭЭГ-корреляторов субъективных временных шкал / Я.В. Булгакова [и др.] // Нейронаука для медицины и психологии. Труды конгресса. – М.: МАКС Пресс, 2020. – С. 119-120.