

# Використання технології CUDA при реалізації обчислень частотно-часових перетворень

## Using CUDA Technology In Implementation Of Time-Frequency Transforms

Євчук О.В.

кафедра комп'ютерних технологій в системах  
управління та автоматики  
Івано-Франківський національний технічний  
університет нафти і газу  
Івано-Франківськ, Україна

Olga Yevchuk

dept. of computer technologies in control systems  
Iv.-Frankivsk National Technical University of Oil and Gas  
Ivano-Frankivsk, Ukraine

Ровінський В.А.

кафедра інформатики  
Прикарпатський національний університет ім. В.  
Стефаніка  
Івано-Франківськ, Україна

Victor Rovinskiy

dept. of informatics  
Vasyl Stefanyk Precarpathian National University  
Ivano-Frankivsk, Ukraine

**Анотація** — Наведено результати дослідження часу виконання перетворення Вігнера-Вілля з використанням технології CUDA. Визначено критичні по часу етапи перетворення та наведено рекомендації щодо його практичної реалізації.

**Ключові слова** — частотно-часові перетворення, перетворення Вігнера-Вілля, програмування CUDA

**Abstract**—The results of execution time exploring for Wigner-Ville transform are presented. Time critical stages of transform are defined and relevant recommendations about its practical implementation are given.

**Keywords**—time-frequency transform, Wigner-Ville transform, CUDA programming

### I. ВСТУП

Частотно-часові перетворення, та зокрема перетворення класу Коена [1] – це потужний засіб аналізу нестационарних сигналів, затосування якого активно досліджуються в останні десятиліття. Частинним випадком частотно-часових перетворень є короткочасове перетворення Фур'є (Short-time Fourier transform, STFT), яке часто застосовується в різних програмних продуктах (зокрема, для обробки звуку) для візуалізації зміни спектру в часі. Основним недоліком STFT є вимушений компроміс між роздільною здатністю по часу та частоті, що не дозволяє з достатньою точністю виявити і оцінити швидкі зміни характеру сигналу у вузькому частотному діапазоні. Перетворення класу Коена, або квадратичні частотно-часові перетворення, дозволяють значно краще оцінювати

локальні нестационарності, однак потребують суттєво більших обчислювальних затрат.

Використання графічних процесорів, і зокрема технології CUDA, з метою підвищення швидкості обчислення частотно-часових перетворень, є перспективним напрямком, що дозволить виконувати великі обсяги обчислень, пов'язані з оцінкою технічного стану механізмів і машин, в реальному часі або з прийнятною швидкістю.

### II. ЛІТЕРАТУРНИЙ ОГЛЯД

В [2] наведено опис реалізації алгоритму обчислення частотно-часових перетворень класу Коена за допомогою технології OpenCL та наведено дослідження швидкості обчислень від кількості відліків та степені згладжування функції ядра. Всі обчислення виконано для фрагменту сигналу довжиною 4096 відліків. Стверджується, що запропонований підхід дозволяє підвищити швидкодію від 20 до 1000 разів порівняно з використанням CPU, однак деталі CPU-реалізації, що використовувалась для порівняння, не наведено, тому не виключено, що таке значне підвищення зумовлено використанням неефективної CPU-реалізації. Для прикладу, в [3] наведено порівняння швидкодії обчислення згортки методом накладання-додавання з використанням CPU (застосовано бібліотеку FFTW для обчислення перетворень Фур'є), CUDA та OpenCL. Для сигналу довжиною близько 17млн. відліків та ядра згортки довжиною 96000 відліків збільшення швидкодії склало

близько 1.5 разів. Підвищення швидкодії вдалося досягти для довжин сигналу більших за  $2^{17}$  для CUDA та  $2^{18}$  для OpenCL.

Згідно даних корпорації NVIDIA [4], вигреш у швидкодії при обчисленні швидкого перетворення Фур'є (на яке припадає основне обчислювальне навантаження при обчисленні частотно-часових перетворень) з використанням бібліотеки cuFFT складає в середньому 10-15 разів порівняно із реалізацією для центрального процесора в бібліотеці Intel MKL (порівняння швидкодії проведено для графічного процесора Tesla C2070 та центрального процесора Intel Quad Core i7-940 1333, 2.93Ghz). Слід зазначити, що Tesla C2070 належить до лінійки графічних процесорів, якими оснащуються професійні високопродуктивні відеокарти, а для відеокарт та центральних процесорів середнього цінового діапазону вигреш у швидкодії, скорше всього, буде суттєво меншим.

В [5] показано, як за допомогою різних способів оптимізації коду, що враховують особливості архітектури CUDA, можна досягти збільшення швидкодії паралельних алгоритмів до 200 разів порівняно з "наївною" реалізацією. Слід зауважити, що в розглянутому прикладі в "наївній" реалізації лише 27% часу займали обчислення, тому вдалося суттєво підвищити ефективність за рахунок збільшення кількості обчислень на ядро, оптимізації пересилок між графічним прискорювачем та основною оперативною пам'яттю, тощо.

### III. АЛГОРИТМ РЕАЛІЗАЦІЇ ПЕРЕТВОРЕННЯ ВІГНЕРА-ВІЛЛЯ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ CUDA

Перетворення Вігнера-Вілля деякого аналітичного сигналу  $x(t)$  описується формулою

$$W(t, f) = \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) \cdot x^*\left(t - \frac{\tau}{2}\right) \cdot e^{-j2\pi f \tau} d\tau \quad (1)$$

де  $x^*(t)$  – комплексно-спряжене значення  $x(t)$ . Аналітичне представлення сигналу  $x(t)$  може бути отримане шляхом застосування перетворення Гільберта до дійсного сигналу.

Це перетворення належить до частотно-часових перетворень класу Коена, що в загальному випадку описуються формулою

$$TF(t, f) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \phi_{t-f}(t-u, f-v) \cdot W(u, v) dudv,$$

де  $\phi(t, f)$  – функція, що являє собою перетворення Фур'є від т.зв. функції ядра, а  $W(t, f)$  – функція, що описується формулою (1). Перетворення Вігнера-Вілля отримується, якщо функцію ядра покласти тотожно рівною одиниці. Будь-яке інше перетворення класу Коена можна, таким чином, трактувати як згладжування перетворення Вігнера-Вілля по часу та по частоті за допомогою двовимірного фільтра з імпульсною характеристикою  $\phi(t, f)$ .

Алгоритм обчислення перетворення Вігнера-Вілля для сигналу, заданого послідовністю дискретних відліків  $\{x_i\}$ ,  $i \in [0; N-1]$ , передбачає наступні кроки:

- обчислюється перетворення Гільберта вхідного сигналу. Результатом перетворення є аналітичний сигнал - масив комплексних чисел  $\{x a_i\}$ ,  $i \in [0; N-1]$ , де дійсна частина співпадає з вихідним сигналом, а уявна являє собою його аналітичне доповнення;
- для кожного відліка  $x a_i$  обчислюється масив добутоків  $K_j = x a_{N/2+i+j} \cdot x a_{N/2+i-j}^*$ ,  $j \in [0; N-1]$  та виконується швидке перетворення Фур'є для масиву  $\{K_j\}$ . Дійсна частина результату є  $j$ -м стовпцем матриці результату перетворення.

Оскільки при обчисленні  $K_j$  для всіх відліків, крім відліка з індексом  $N/2$ , необхідно використовувати відліки за межами інтервалу  $[0; N-1]$ , такі відліки приймаються рівними нулю. Щоб зменшити явище розтікання спектру, зумовлене обмеженою довжиною вікна, слід в такому випадку застосовувати функцію вікна, відмінну від прямокутної.

Оскільки в практичних застосуваннях

При використанні технології CUDA доцільно скористатися бібліотекою cuFFT для обчислення швидкого перетворення Фур'є. Додатковими кроками алгоритму будуть копіювання масиву даних із оперативної пам'яті в пам'ять графічного прискорювача перед виконанням перетворень та в зворотньому напрямку – після виконання.

### IV. ДОСЛІДЖЕННЯ ШВИДКОДІЇ

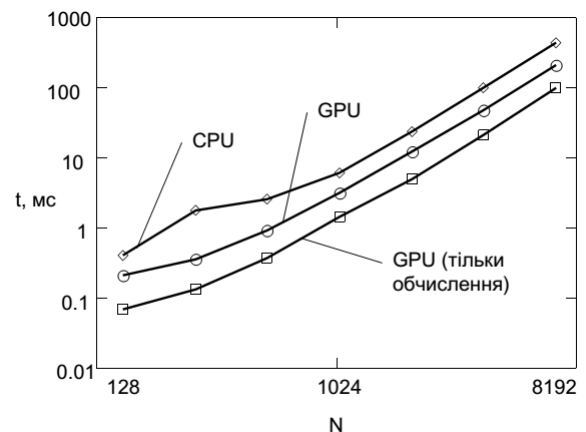


Рис. 1. Час виконання перетворення Вігнера-Вілля на центральному та графічному процесорі

### V. ВИСНОВКИ

Проведені дослідження свідчать, що при роботі з великими обсягами вхідних даних практично половина часу витрачається на пересилку даних між основною оперативною пам'яттю та пам'яттю графічного прискорювача. За таких умов максимальне підвищення

швидкодії для плати GT 525M складає близько 5 разів (для довжини фрагменту 256), середнє – близько двох разів. Якщо враховувати лише час на виконання обчислень, то максимальнє підвищення швидкодії складає 13 разів, середнє – близько 5. Якщо основною метою обчислень є візуалізація результату для його подальшого аналізу, розмірність даних все рівно потребує зменшення для масштабування зображення перед виводом на дисплей. Доцільно в такому разі виконувати масштабування (а також перетворення амплітуди в значення кольорових компонент для відображення, тощо) одразу на графічному процесорі і передавати до основної оперативної пам'яті, якщо це необхідно, готове зображення. При необхідності можна зберігати повний результат в пам'яті графічного прискорювача та формувати його фрагменти в різних масштабах по запиту користувача.

## ЛІТЕРАТУРА

- [1] Time Frequency Signal Analysis and Processing. A Comprehensive Reference / Edited by Boualem Boashash. Elsevier, 2003. 743pp.
- [2] K. Konopko. An implementation of the Cohen's class time-frequency distributions on a massively parallel processor. *Przegląd elektrotechniczny (Electrical Review)*, R. 88 NR 9b/2012: pp.289-291, 2012.
- [3] D.A. Mauro. Audio convolution by the mean of GPU: CUDA and OpenCL implementations. *Proceedings of the Acoustics 2012 Nantes Conference, 23-27 April 2012, Nantes, France*: pp.2863-2868, 2012.
- [4] CUDA Toolkit 4.0 Performance Report. June 2011. - [http://developer.download.nvidia.com/assets/cuda/files/CUDA\\_4\\_0\\_Math\\_Libraries\\_Performance\\_6\\_14.pdf](http://developer.download.nvidia.com/assets/cuda/files/CUDA_4_0_Math_Libraries_Performance_6_14.pdf)
- [5] Tarjan, D., K. Skadron, and P. Micikevicius, "The art of performance tuning for CUDA and manycore architectures," *Birds-of-a-feather session at SC'09, 2009*.