



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

The Sparse Fourier Transform: Theory & Practice

Разреженное преобразование Фурье: теория и практика

by

Haitham Al Hassanieh,
Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

Часть 1

- Введение
- Историческая справка
- Текущая ситуация
- Основные вопросы исследования
- Теоретический вклад работы
- Практический вклад работы
- Постановка задачи
- Структура алгоритма:
 - Frequency Bucketization,
 - Оценка частоты
 - Разрешение конфликтов

Преобразование Фурье – один из фундаментальных инструментов вычисления частотного представления сигналов.

Оно играет центральную роль в обработке сигналов связи, сжатии аудио видео, медицинской визуализации, геномике, астрономии, обработке сейсмических данных, а также во многих других областях.

Быстрые алгоритмы вычисления преобразования Фурье могут использоваться повсеместно, приносить пользу большому количеству приложений.



Fast Fourier Transform (FFT, БПФ) – самый быстрый алгоритм вычисления преобразования Фурье, который работает в почти линейном режиме.

Это делает его незаменимым инструментом для многих приложений.

БПФ – один из самых мощных алгоритмов в новейшей истории .

Впервые программная реализация алгоритма БПФ была осуществлена в начале 60-х годов Джоном Кули в вычислительном центре IBM под руководством Джона Тьюки, а в 1965 году они же опубликовали статью об алгоритме быстрого преобразования Фурье (Cooley–Tukey FFT).

Изобретённый в 1965 году Кули и Тьюки, БПФ вычисляет преобразование Фурье сигнала размера n в $O(n \log n)$ времени.

Однако сегодня алгоритм БПФ выполняется уже недостаточно быстро, особенно, когда речь идёт о больших данных.

Проблемы возникают в следующих случаях:

- обрабатываемые наборы данных превышают терабайты →
→ FFT выполняется слишком долго
- процедура сбора данных дорогостоящая или трудозатратная →
→ нельзя собрать достаточное количество параметров для вычисления БПФ

Одним из центральных вопросов в теории алгоритмов:

существование алгоритмов преобразования Фурье быстрее чем БПФ.

За последние 20 лет достигнуты значительные успехи в разработке алгоритмов для разреженных сигналов:

1. Появился первый подобный алгоритм для преобразования Адамара
2. Обнаружены ещё несколько сублинейных алгоритмов для комплексных входных данных.

Достоинства:

- + эти алгоритмы превосходят время исполнения FFT для разреженных сигналов. Для очень разреженных сигналов самый быстрый алгоритм имеет время исполнения $O(k \log c(n) \log(n/k))$ для некоторого $c > 2$. Этот алгоритм превосходит FFT для любого k , меньшего $\Theta(n / \log^a n)$ для некоторого $a > 1$.

Недостатки:

- ни один из существующих алгоритмов не уменьшает время исполнения FFT для всего диапазона разреженных сигналов, т. е. $k = o(n)$
- вышеупомянутые алгоритмы осложнены большой константой «Big-Oh», которая на практике приводит к длительной работе
- в практические системы эти алгоритмы сложно включить

Два основных вопроса исследования

- Как можно использовать разреженность для разработки более быстрых алгоритмов преобразования Фурье?
- Как создавать программные и аппаратные системы, которые адаптируют эти алгоритмы к приложениям из различных областей (с целью достижения практической пользы)?

• Оптимизация сложности исполнения

В диссертации разработаны алгоритмы разреженного преобразования Фурье с самой низкой на сегодняшний день сложностью исполнения.

- для точно разреженных сигналов представлен алгоритм, который работает в $O(k \log n)$, где k – число больших частотных коэффициентов (т. е. разреженности), а n – размер сигнала. Этот алгоритм оптимален, если алгоритм FFT является оптимальным.
- для приблизительно разреженных сигналов разработан алгоритм, который исполняется за $O(k \log n \log(n/k))$ время, которое является $\log n$ фактором от оптимального.

Оба алгоритма улучшают FFT для любой разреженности $k = o(n)$ и имеют небольшие константы. В результате на практике они часто бывают быстрее чем FFT и быстро запускаются на очень больших наборах данных.

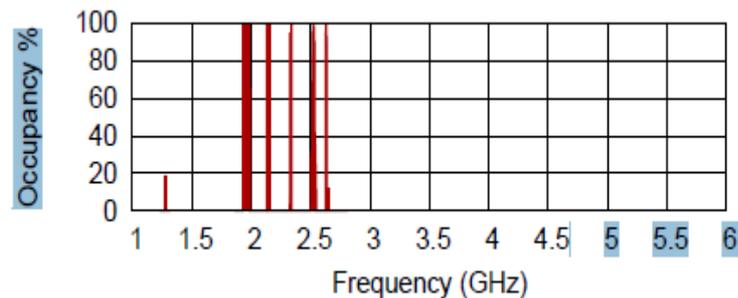
• Оптимизация сложности выборки

В диссертации разработаны алгоритмы разреженного преобразования Фурье с оптимальной сложностью выборки для средних входных данных, т. е. этот алгоритм использует минимальное количество выборок входных данных, которые приведут к правильному ответу. Следовательно, они снижают затраты на приобретение, пропускную способность и накладные расходы ввода-вывода, необходимые для сбора, передачи и хранения данных. В частности, эти алгоритмы требуют только выборок $O(k)$ для точно разреженных сигналов и $O(k \log n)$ для приблизительно разреженных сигналов.

Кроме того, алгоритмы естественно расширяются до многомерных разреженных преобразований Фурье без больших издержек.

Wireless Networks

Непрерывно растущий спрос на беспроводную связь привёл к тому, что было выпущено несколько новых диапазонов для совместного использования динамического спектра.

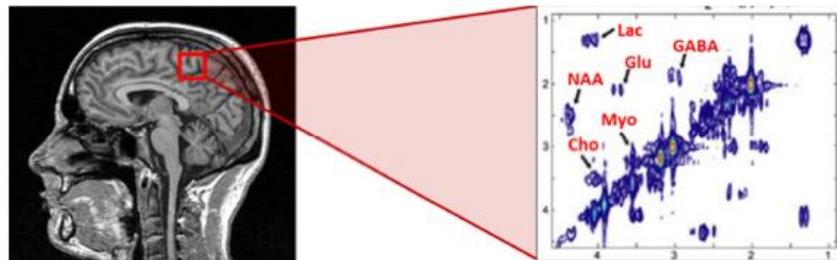


Однако основным техническим препятствием является потребность в приёмниках, которые смогут захватывать ГГц спектра в реальном времени, чтобы быстро идентифицировать незанятые полосы. Такие приёмники потребляют много мощности.

Для разрешения этой проблемы стоит обратить внимание на то, что беспроводной спектр используется редко и использует разреженное преобразование Фурье для создания приёмника, который сможет захватывать и восстанавливать ГГц спектра в режиме реального времени, а выборки – только в млн. выборок в секунду. Использование aliasing filters и метода чередования фаз для создания всего приёмника, позволяет использовать только дешёвые компоненты, с низким энергопотреблением, аналогичным тому, что сегодня используется Wi-Fi и LTE в каждом мобильном телефоне. Дизайн реализуется с использованием трёх программных радиостанций, каждая выборка которых составляет 50 мегапикселей / с, и созданием устройства, которое захватывает 0,9 ГГц – то есть в **6 раз большую цифровую пропускную способность**.

Medical Imaging

Одним из основных направлений в МРТ является магнитно-резонансная спектроскопия (МРС).



МРС позволяет масштабировать и обнаруживать биохимическое содержание каждого вокселя в головном мозге, а это можно использовать для обнаружения биомаркеров болезней и диагностировать ранние стадии рака, аутизма и болезни Альцгеймера.

Основные проблемы МРС:

- МРС-исследования занимают слишком много времени: пациент должен оставаться на МРТ-машине более 2 часов.
- изображения МРС подвержены множеству помех, которые могут маскировать некоторые биомаркеры болезней.

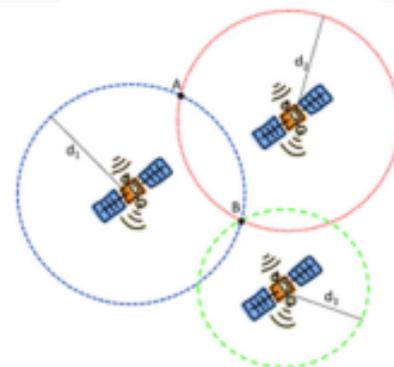
Таким образом, МРС-исследования далеко не всегда являются основанием для постановки клинического диагноза. Для решения этой проблемы может быть использована обработка данных МРС с применением разреженного преобразования Фурье, что позволяет:

- повысить качество изображения за счёт подавления помех
- сократить втрое время, которое пациент должен потратить в аппарате

Mobile Systems

GPS - одна из наиболее широко используемых беспроводных систем.

Для определения своего положения GPS-приёмник должен зафиксировать спутниковые сигналы, совместив полученный сигнал с кодом каждого спутника.



Для GPS необходимы масштабные вычисления, которые требуют много времени и энергии.

В результате запуск GPS, например, в телефоне может быстро разрядить аккумулятор .

В работе представлен новый GPS-приёмник, который минимизирует требуемые вычисления для блокировки сигнала спутника, тем самым уменьшая задержку локализации и потребление энергии. Разработан GPS-приёмник, который использует разреженное преобразование Фурье, чтобы быстро фиксировать спутниковый сигнал и определять своё местоположение.

Сравнение эмпирических результатов с реальными спутниковыми сигналами показывает, что новый GPS-приёмник **сокращает вычислительные затраты от 2 до 6 раз, что приводит к значительному снижению задержки локализации и потребления энергии.**

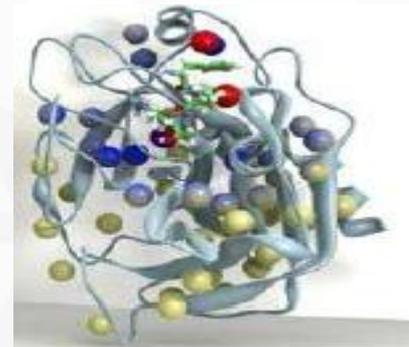
Ядерный магнитный резонанс (ЯМР)

ЯМР – это метод, который представляет детальные структурные свойства химических соединений, обеспечивающих 3D-структуру сложных белков и нуклеиновых кислот.

Сбор ЯМР-измерений – очень трудоёмкий и дорогостоящий процесс, который может занимать от нескольких дней до нескольких недель, и потому исследователи не могут проводить многомерные ЯМР-эксперименты, необходимые для анализа более сложных белковых структур.

ЯМР использует спектральный анализ для нахождения резонансных частот, которые соответствуют связям между разными атомами, а ЯМР-спектры являются редкими.

В работе показано, как с помощью преобразования SFT генерировать ЯМР-спектры путём подвыборок измерений ЯМР, как именно настроить разреженное преобразование Фурье для многомерного ЯМР и доказано, что оно может **сократить время ЯМР-эксперимента в 16 раз.**



Computer Graphics

Световая съемка - это активная область в графике, где 2D-массив камер или линз используется для захвата 4-го светового поля сцены. Это позволяет пользователю извлекать глубину 3D, переориентировать сцену на любую плоскость и изменять угол, из которого он просматривает сцену. Это важно для систем VR (Virtual Reality), а также для последующей обработки изображений и видео. Однако захват светлых полей является дорогостоящим, поскольку для съемки сцены с разных точек зрения требуется много камер или линз.



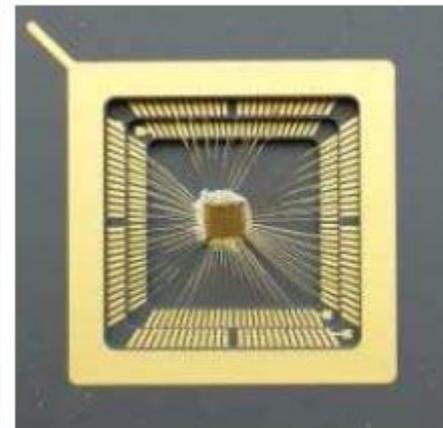
Таким образом, наша цель - снизить затраты на захват светового поля, используя лишь некоторые из камер в 2D-массиве и восстанавливая изображения с отсутствующих камер. Для этого мы используем тот факт, что преобразование Фурье светового поля является редким, и мы используем разреженное преобразование Фурье для под-выборки ввода и уменьшения количества камер. Это улучшает качество реконструкции, **уменьшая необходимое количество камер от 6 до 10 раз.**

Digital Circuits

Размер традиционных аппаратных реализаций БПФ был ограничен несколькими тысячами точек. Это связано с тем, что БПФ требует огромной пропускной способности ввода-вывода, потребляет много энергии и занимает большую площадь кремния.

Разреженное преобразование Фурье, безусловно, решает эти проблемы благодаря своим низким требованиям к вычислительной мощности и памяти, что позволяет производить очень большие преобразования Фурье.

Сегодня существует самый большой чип VLSI с почти миллионным преобразованием Фурье, который **потребляет в 40 раз меньше мощности**, чем предыдущие реализации FFT VLSI.



Рассмотрим сигнал x размера n , дискретное преобразование Фурье которого \hat{X} определено как:

$$\hat{x}(f) = \sum_{t=0}^{n-1} x(t) \cdot e^{-j2\pi ft/n}$$

\hat{X} является k -разреженным, если он имеет ровно k ненулевых частотных коэффициентов, а остальные n коэффициентов равны нулю.

Цель разреженного преобразования Фурье:

точное восстановление \hat{X} для нахождения частотных положений f и значений $\hat{x}(f)$ для k ненулевых коэффициентов.

Для общих сигналов разреженное преобразование Фурье вычисляют как ***k*-разреженное приближение** (*k*-sparse approximation) $\hat{\mathbf{x}}'$ от $\hat{\mathbf{x}}$

Лучшее ***k* разреженное приближение** $\hat{\mathbf{x}}$ можно получить, установив все, кроме наибольших ***k***, коэффициентов $\hat{\mathbf{x}}$ равными 0.

Целью является вычисление приближения $\hat{\mathbf{x}}'$, в котором ошибка приближения $\hat{\mathbf{x}}$ ограничена ошибкой на наилучшем ***k*-разреженном приближении**.

Формально $\hat{\mathbf{x}}$ должно удовлетворять следующим ℓ_2 / ℓ_2 условиям:

$$\|\hat{\mathbf{x}} - \hat{\mathbf{x}}'\|_2 \leq C \min_{k\text{-sparse } \mathbf{y}} \|\hat{\mathbf{x}} - \mathbf{y}\|_2$$

где **C** - некоторый фактор приближения, минимизация по точности ***k*-разреженных сигналов**

Три основных компоненты разряженного преобразования Фурье:

- Frequency Bucketization
- Оценка частоты
- Разрешение конфликтов

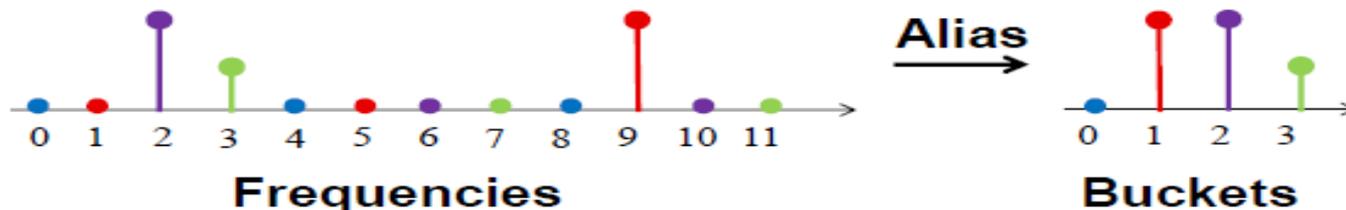
Frequency Bucketization

Разряженное преобразование Фурье начинается с хэширования частотных коэффициентов векторов \hat{X} , таких что значение корзины представляет собой сумму значений частотных коэффициентов, которые попали в эту корзину. Поскольку \hat{X} разрежен, многие корзины будут пустыми и могут быть просто отброшены. Затем алгоритм фокусируется на непустых корзинах и вычисляет позиции и значения больших частотных коэффициентов в этих корзинах, это называется *шаг оценки частоты*.

Процесс Frequency Bucketization осуществляется за счёт использования фильтров. Фильтр подавляет и обнуляет частотные коэффициенты, которые выпали из корзины при прохождении через частотные коэффициенты, которые попали в корзину.

В данной диссертации определено несколько эффективных фильтров, которые используют небольшое количество выборок и имеют минимальную или нулевую утечку в частотной области:

1. Flat Window Filter
2. Aliasing Filter
3. Fourier Projection Filter



Bucketization с использованием Aliasing фильтра

В частности, 12 частот разложены в 4 контейнера. Частоты, которые равномерно распределены на 4 (показаны одинаковым цветом), оказываются в одном и том же контейнере.

На этом этапе разреженное преобразование Фурье оценивает позиции и значения ненулевых частотных коэффициентов, которые создавали энергию в каждом из непустых контейнеров.

Так как \hat{X} разрежен, многие из непустых контейнеров, вероятно, будут иметь в себе один ненулевой частотный коэффициент, и только небольшое число контейнеров будет иметь столкновение нескольких ненулевых коэффициентов.

Сначала сосредоточимся на контейнерах с одним ненулевым частотным коэффициентом и определим значение и положение этой ненулевой частоты, т.е. $\hat{X}(f)$ и соответствующее f .

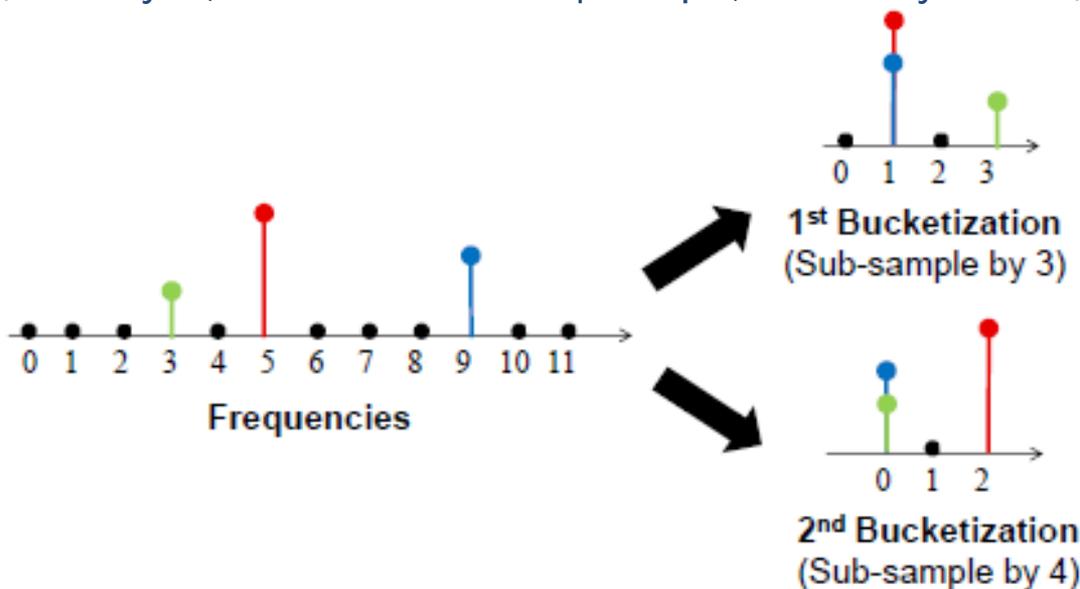
В отсутствии столкновений значение ненулевого частотного коэффициента представляет собой значение контейнера, которое оно хэширует, поскольку все остальные частоты, которые попали в этот же контейнер, имеют нулевые значения. Следовательно, можно легко найти значение ненулевого частотного коэффициента в контейнере. Тем не менее, всё ещё не известно его частотное положения f , так как *frequency bucketization* отображает несколько частот в одном и том же контейнере.

В диссертации использованы следующие подходы для оценки частоты:

- подход "Время-сдвиг / фаза-вращение":
- подход к голосованию

Когда ненулевые частоты сталкиваются в одном контейнере, невозможно их правильно оценить. Следовательно, для восстановления полного частотного спектра необходимо разрешить эти конфликты.

Для разрешения конфликтов необходимо повторить процесс *frequency bucketization* таким образом, чтобы гарантировать, что одни и те же ненулевые частоты не сталкиваются друг с другом каждый раз. Выбор способа, которым эта цель будет достигнута, зависит от типа фильтра, используемого для bucketization.



Частоты, которые сталкиваются в одном из фильтров, не будут сталкиваться во втором. Например, частоты 5 и 9 сталкиваются в первом фильтре. Но частота 5 не сталкивается во втором фильтре, что позволяет её оценить и вычесть.

Алгоритмические результаты

Chapter	Algorithm	Runtime / Sampling Complexity	Sparsity Range	Model & Guarantee	Analysis	Bucketization & Estimation Techniques
3	SFT 1.0*	$O(\log n \sqrt{nk \log n})$	$O(n / \log n)$	Approximate (ℓ_∞ / ℓ_2)	Worst Case	<ul style="list-style-type: none"> • Flat Window • Voting
	SFT 2.0	$O(\log n \sqrt[3]{nk^2 \log n})$	$O(n / \sqrt{\log n})$	Approximate (Heuristic)	Worst Case	<ul style="list-style-type: none"> • Flat Window & Aliasing • Voting
4	SFT 3.0	$O(k \log n)$	$O(n)$	Exact	Worst Case (Time Optimal)	<ul style="list-style-type: none"> • Flat Window • Phase Rotation
	SFT 4.0	$O(k \log n \log(n/k))$	$O(n)$	Approximate (ℓ_2 / ℓ_2)	Worst Case	<ul style="list-style-type: none"> • Flat Window • Phase Rotation
	SFT 4.1	$O(k \log^2 n \log(n/k))$	$O(n / \log n \log \log n)$	Approximate (ℓ_2 / ℓ_2)	Worst Case, 2D	<ul style="list-style-type: none"> • Flat Window • Phase Rotation
5	SFT 5.1	$O(k \log k)$ Time $O(k)$ Samples	$O(\sqrt{n})$	Exact	Average Case, 2D (Optimal)	<ul style="list-style-type: none"> • Projections & Aliasing • Phase Rotation
	SFT 6.0	$O(k \log^2 n)$ Time $O(k \log n)$ Samples	$\Theta(\sqrt{n})$	Approximate (ℓ_2 / ℓ_2)	Average Case, 2D (Sample Optimal)	<ul style="list-style-type: none"> • Projections • Phase Rotation

Основные обозначения

$\omega = e^{-2\pi i/n}$ используется как n -й корень из единицы, $\omega' = e^{-2\pi i/\sqrt{n}}$ как корень \sqrt{n} из единицы.
Для любого комплексного числа a используется $\varphi(a) \in [0, 2\pi]$ для обозначения **фазы** a .

Для вектора $x \in \mathbb{C}^n$ обозначено $\text{supp}(x) \subset [n]$.

$\|x\|_0$ используется для обозначения $|\text{supp}(x)|$, числа ненулевых координат x .
Его спектр Фурье обозначается через \hat{x} ,

$$\hat{x}_i = \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{ij} x_j$$

Для вектора длины n индексы должны интерпретироваться по модулю n , поэтому $x_{-i} = x_{n-i}$.
Это позволяет определить **свёртку**

$$(x * y)_i = \sum_{j \in [n]} x_j y_{i-j}$$

И покоординатное произведение $(x \cdot y)_i = x_i y_i$, так $\widehat{x \cdot y} = \hat{x} * \hat{y}$.

Определение Window Functions.

Мы определяем $a(\varrho, \delta, w)$ как стандартную Window Function симметричным вектором $F \in R^n$ при $\text{supp}(F) \subseteq [-w/2, w/2]$ таких, что $\hat{F}_0 = 1$, $\hat{F}_i > 0$ для всех $i \in [-\varrho n, \varrho n]$, и $|\hat{F}_i| < \delta$ для всех $i \notin [-\varrho n, \varrho n]$

Утверждение.

Для любого ϱ и δ , существует $(\varrho, \delta, O(1/\varrho \log(1/\delta)))$ standard window function.

Определение Flat Window Functions.

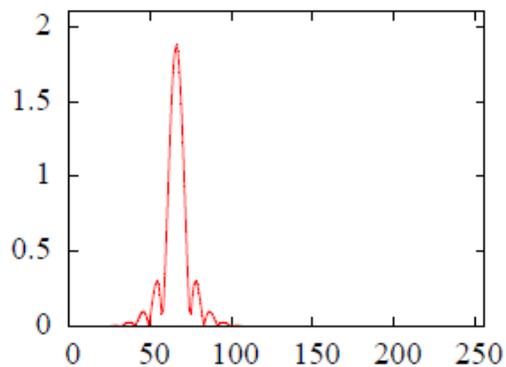
Определяется $a(\varrho, \varrho', \delta, w)$ flat window function симметричным вектором $F \in R^n$ при $\text{supp}(F) \subseteq [-w/2, w/2]$ таких, что $\hat{F}_i \in [1-\delta, 1+\delta]$ для всех $i \in [-\varrho' n, \varrho' n]$ и $|\hat{F}_i| < \delta$ для всех $i \notin [-\varrho n, \varrho n]$.

Утверждение.

Для любого ϱ, ϱ' , и δ при $\varrho' < \varrho$, существует $(\varrho, \varrho', \delta, O(1/(\varrho - \varrho') \log n / \delta))$ flat window function.

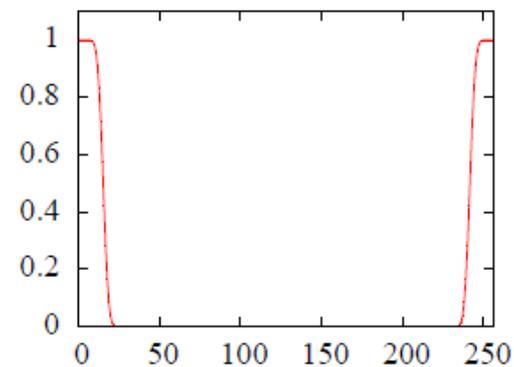
Пример flat window function для $n = 256$. Это сумма из 31 смежных $(1/22, 10^{-8}, 133)$ функций Дольфа-Чебышева, дающие функцию flat window function $(0.11, 0.06, 2 \times 10^{-9}, 133)$.

G (linear scale)



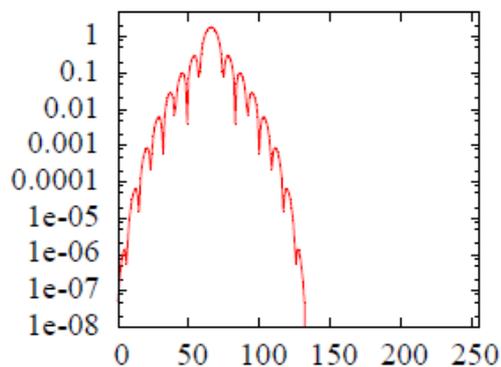
(a)

\widehat{G} (linear scale)



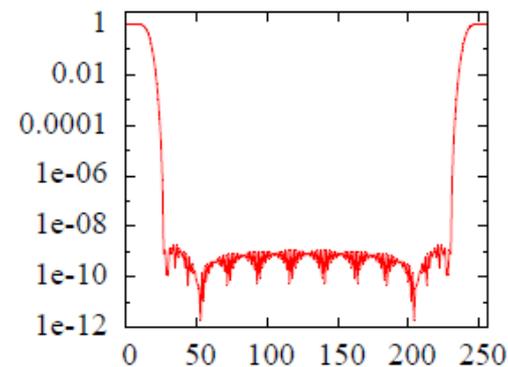
(b)

G (log scale)



(c)

\widehat{G} (log scale)



(d)

Определение

Пусть существует $\sigma \equiv 1 \pmod n$. Определим перестановку $P_{\sigma, a, b}$ как

$$(P_{\sigma, a, b} x)_i = x_{\sigma(i-a)} \omega^{abi}$$

Также определяем $\pi_{\sigma, a, b}(i) = \sigma(i - b) \pmod n$.

Утверждение. $\widehat{P_{\sigma, a, b} x}_{\pi_{\sigma, b}(i)} = \widehat{x}_i \omega^{a\sigma i}$.

Доказательство

$$\begin{aligned} \widehat{P_{\sigma, a, b} x}_{\sigma(i-b)} &= \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{\sigma(i-b)j} (P_{\sigma, a, b} x)_j \\ &= \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{\sigma(i-b)j} x_{\sigma(j-a)} \omega^{\sigma bj} \\ &= \omega^{a\sigma i} \frac{1}{\sqrt{n}} \sum_{j \in [n]} \omega^{i\sigma(j-a)} x_{\sigma(j-a)} \\ &= \widehat{x}_i \omega^{a\sigma i}. \end{aligned}$$

Лемма. Если $j \neq 0$, а n - степени два, а σ - равномерное случайное нечётное число в $[n]$, то $\Pr[\sigma_j \in [-C, C]] \leq 4C/n$.

Предположим, что есть $x \in C_n$ и параметр B делящийся на n , и необходимо вычислить

$$\hat{y}_i = \hat{x}_{i(n/B)} \text{ for } i \in [B].$$

Утверждение. \hat{y} - B -мерное преобразование Фурье $y_i = \sum_{j=0}^{n/B-1} x_{i+Bj}$.

Поэтому \hat{y} может быть выполнено за время $O(|\text{supp}(x)| + B \log B)$

Доказательство.

$$\begin{aligned} \hat{x}_{i(n/B)} &= \sum_{j=0}^{n-1} x_j \omega^{ij(n/B)} = \sum_{a=0}^{B-1} \sum_{j=0}^{n/B-1} x_{Bj+a} \omega^{i(Bj+a)n/B} \\ &= \sum_{a=0}^{B-1} \sum_{j=0}^{n/B-1} x_{Bj+a} \omega^{ian/B} = \sum_{a=0}^{B-1} y_a \omega^{ian/B} = \hat{y}_i, \end{aligned}$$

Простой и практичный алгоритм

Представленный алгоритм, имеет время исполнения

$$O\left(\log n \sqrt{nk \log n}\right)$$

где n – размер сигнала, а k – число ненулевых частотных коэффициентов.

Таким образом, алгоритм работает быстрее, чем FFT для k до $O(n / \log n)$.

Напротив, более ранние алгоритмы требовали асимптотически меньших оценок на k . Это асимптотическое улучшение отражается и на эмпирическом времени исполнения.

Пусть B -параметр, разделяющий n (будет определён позже).

Пусть $G - (1/B, 1/(2B), \delta, w)$ flat window function для некоторых δ и $w = O(B \log n/\delta)$.

Тогда $\delta \approx 1/n^c$, поэтому можно считать этот параметр ничтожно малым.

Существует две версии внутреннего цикла: *циклы определения местоположения* и *циклы оценки*:

- *цикл определения местоположения* описывается как процедура LOCATIONINNERLOOP: даётся параметр d и вывод множество $I \subset [n]$ из dkn/B координат, который содержит все большие коэффициенты с «хорошей» вероятностью
- *цикл оценки*, описывается как процедура ESTIMATIONINNERLOOP: даётся множество $I \subset [n]$ и оценка \hat{x}_I такая, что каждая из координат вычисляется также с «хорошей» вероятностью.

Можно вычислить \hat{z} за $O(w + B \log B) = O(B \log n/\delta)$ время.

Таким образом, *цикл определения местоположения* займёт $O(B \log n/\delta + dkn/B)$ времени, и *цикл оценки* займёт $O(B \log n/\delta + |I|)$ времени.

SFT 1.0: Неитеративное разреженное преобразование Фурье для $k = o(n / \log n)$

procedure LOCATIONINNERLOOP(x, k, B, d)

Choose σ, τ and b uniformly at random from $[n]$ such that σ is odd.

Compute $\hat{z}_i = \hat{y}_{in/B}$ for $j \in [B]$, where $y = G \cdot (P_{\sigma, \tau, b} x)$ \triangleright FFT of $z_i = \sum_{j=0}^{\lceil w/B \rceil - 1} y_{i+jB}$

$J \leftarrow$ indices of dk largest coefficients in \hat{z} .

$I = \{i \in [n] \mid h_{\sigma, b}(i) \in J\}$ where $h_{\sigma, b}(i) = \text{round}(\sigma(i - b)B/n)$ $\triangleright h_{\sigma, b} : [n] \rightarrow [B]$

return I

procedure ESTIMATIONINNERLOOP(x, B, I)

Choose σ, τ and b uniformly at random from $[n]$ such that σ is odd.

Compute $\hat{z}_i = \hat{y}_{in/B}$ for $j \in [B]$, where $y = G \cdot (P_{\sigma, \tau, b} x)$

$\hat{x}' \leftarrow 0$

for $i \in I$ **do**

$\hat{x}'_i = \hat{z}_{h_{\sigma, b}(i)} \omega^{-\tau \sigma i} / \widehat{G}_{o_{\sigma, b}(i)}$ where $o_{\sigma, b}(i) = \sigma(i - b) - h_{\sigma, b}(i)(n/B)$

$\triangleright o_{\sigma, b} : [n] \rightarrow [-n/(2B), n/(2B)]$

return \hat{x}'

procedure NONITERATIVESPARSEFFT(x, k, B, L, d)

$\hat{x}' \leftarrow 0$

for $r \in \{1, \dots, L\}$ **do**

$I_r \leftarrow$ LOCATIONINNERLOOP(x, k, B, d).

$I = I_1 \cup \dots \cup I_L$

for $i \in I$ **do**

$s_i = |\{r \mid i \in I_r\}|$

$I' = \{i \in I \mid s_i \geq L/2\}$

for $r \in \{1, \dots, L\}$ **do**

$\hat{x}'_{I'_r} \leftarrow$ ESTIMATIONINNERLOOP(x, B, I'_r)

for $i \in I'$ **do**

$\hat{x}'_i = \text{median}(\{\hat{x}'_{I'_r}\})$

return \hat{x}'

SFT 1.0: Неитеративное разреженное преобразование Фурье для $k = o(n / \log n)$

В части (a) алгоритм начинается с доступа во временной области к $P_{\sigma, \tau, b} x$, заданный как

$$(P_{\sigma, \tau, b} x)_i = x_{\sigma(i-\tau)} \omega^{\sigma b i}$$

Который переставляет спектр x , переставляя образцы в область времени.

В части (b) алгоритм вычисляет сигнал временной области

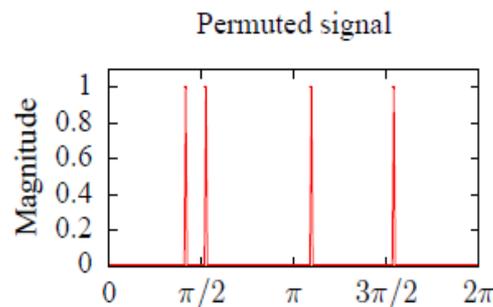
$$y = G \cdot P_{\sigma, \tau, b} x$$

Спектр y (на фото) большой вокруг больших координат $P_{\sigma, \tau, b} x$.

Алгоритм затем вычисляет \hat{z} , который, как показано на рисунке в части (c). Во время цикла estimationloop алгоритм вычисляет \hat{x}_i на основе значения ближайшей координаты \hat{z} , а именно

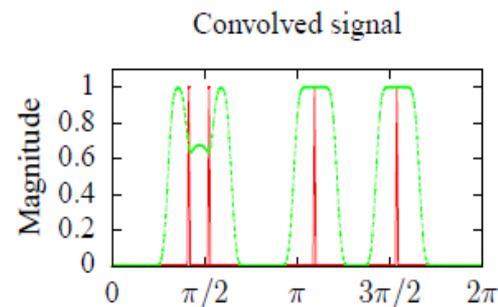
$$\hat{z}_{h_{\sigma, b}(i)}$$

Во время locationloops (часть (d)) алгоритм выбирает J , верхние dk (здесь 4) координаты z и выбирает элементы $[n]$, которые расположены ближе всего к этим координатам (затененные области изображения). На выходе набор I преобразованных этих элементов.



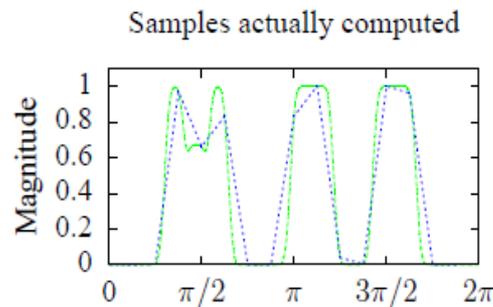
$$\widehat{P_{\sigma, \tau, b} x}$$

(a)



$$\hat{y} = G \cdot \widehat{P_{\sigma, \tau, b} x}$$

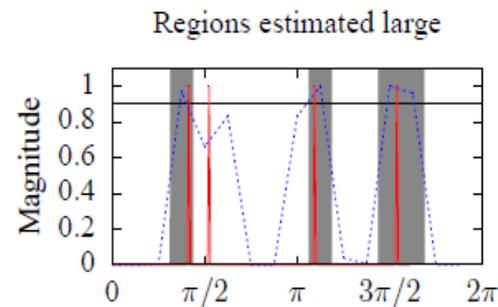
(b)



$$\hat{y} = G \cdot \widehat{P_{\sigma, \tau, b} x}$$

$$\hat{z}$$

(c)



Chosen region $\widehat{P_{\sigma, \tau, b} x}$

Sample cutoff \hat{z}

(d)

Алгоритм SFT 1.0, параметризуется с помощью ϱ и δ . Он прорабатывает $L = O(\log n)$ итераций внутреннего цикла с параметрами $B = O\left(\sqrt{\frac{nk}{\epsilon \log(n/\delta)}}\right)$

и $d = O(1/\varrho)$, а также δ .

Лемма. Алгоритм работает за время $O\left(\sqrt{\frac{nk \log(n/\delta)}{\epsilon}} \log n\right)$.

Теорема. Запуск алгоритма с параметрами $\varrho, \delta < 1$ дает \hat{x}' , удовлетворяющую

$$\|\hat{x}' - \hat{x}\|_{\infty}^2 \leq \frac{\epsilon}{k} \|\hat{x}_{-s}\|_2^2 + \delta^2 \|\hat{x}\|_1^2.$$

с вероятностью $1 - 1/n$ и временем работы $O\left(\sqrt{\frac{nk \log(n/\delta)}{\epsilon}} \log n\right)$

SFT 2.0: неитеративное разреженное преобразование Фурье с эвристикой для $k = o(n / \sqrt{\log n})$

procedure NONITERATIVESPARSEFFT2(x, k, B, L, d, M)

$\hat{x}' \leftarrow 0$

Choose $\tau \in [n]$ uniformly at random

$y_i = x_{i(n/M)+\tau}$

Compute \hat{y}

$T \leftarrow$ indices of the $2k$ largest elements of \hat{y}

$\triangleright T \subset [M]$

for $r \in \{1, \dots, L\}$ **do**

$J \leftarrow$ LOCATIONINNERLOOP(x, k, B, d).

$I_r = \{i \in J \mid i \bmod M \in T\}$

$I = I_1 \cup \dots \cup I_L$

for $i \in I$ **do**

$s_i = |\{r \mid i \in I_r\}|$

$I' = \{i \in I \mid s_i \geq L/2\}$

for $r \in \{1, \dots, L\}$ **do**

$\hat{x}_{I'}^r \leftarrow$ ESTIMATIONINNERLOOP(x, B, I')

for $i \in I'$ **do**

$\hat{x}'_i = \text{median}(\{\hat{x}_i^r\})$

return \hat{x}'

SFT 2.0 предполагает, что все «большие» коэффициенты j имеют $j \bmod M$ в T . Таким образом, множества I_r могут содержать только координаты i с $i \bmod M \in T$. Ожидается, что $|I_r| \approx \frac{2k}{M} dkn/B$, а не предыдущий dkn/B . Это означает, что приведённая эвристика улучшит время выполнения внутренних циклов от $O(B \log(n/\delta) + dkn/B)$ до $O(B \log(n/\delta) + \frac{k}{M} dkn/B + M + dk)$ за счёт предварительной обработки $O(M \log M)$

Утверждение. Эвристическое приближение, SFT 2.0 работает за

$$O((k^2 n \log(n/\delta)/\epsilon)^{1/3} \log n)$$

пока $k \leq \epsilon^2 n \log(n/\delta)$



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Спасибо за внимание!

101000, Россия, Москва, Мясницкая ул., д. 20

Тел.: (495) 621-7983, факс: (495) 628-7931

www.hse.ru