

А. С. Матвеев¹, **В. В. Никитин**², **А. А. Дучков**^{1,3}, **А. А. Романенко**³

¹ ИИГТ СО РАН
пр. Академика Коптюга, 3, Новосибирск, 630090, Россия

² MAX IV Laboratory
Фотонгатан 2, 225 92, Лунд, Швеция

³ Новосибирский государственный университет
ул. Пирогова, 1, Новосибирск, 630090, Россия

alexey.matveev.9@gmail.com, vnikitin90@gmail.com
DuchkovAA@ipgg.sbras.ru, romanenko.alexey@gmail.com

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ПАРАБОЛИЧЕСКОГО ПРЕОБРАЗОВАНИЯ РАДОНА НА ОСНОВЕ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ НА НЕРЕГУЛЯРНЫХ СЕТКАХ *

Приводится параллельный алгоритм для вычисления параболического преобразования Радона – суммирование двумерной функции вдоль парабол с разной кривизной и вертикальным смещением вершины. Основу вычислительного алгоритма составляет параллельная реализация быстрого преобразования Фурье на нерегулярных сетках, учитывающая особенности современных центральных процессоров. Дается описание оптимизации алгоритмов и сравнение с существующими аналогами. Приводится применение параболического преобразования Радона при обработке сейсмических данных.

Ключевые слова: обработка сейсмических данных, преобразование Радона, преобразование Фурье на нерегулярных сетках, параллельный алгоритм.

Введение

Преобразования Радона широко используются в обработке сигналов в различных приложениях. Так, линейное преобразование Радона используется при обработке данных компьютерной томографии [1]. Параболическое и гиперболическое преобразование Радона широко применяется при обработке сейсмических данных [2].

В случае реализации преобразования Радона простым суммированием вдоль соответствующих кривых вычислительная сложность алгоритма в двумерном случае составляет $O(N^3)$ в предположении, что двумерный входной набор данных имеет по N отсчетов в каждом направлении. Для ускорения вычислений часто оказывается выгодно реализовать преобразования в области Фурье. Такой подход дает заметное преимущество, если удастся использовать алгоритм быстрого преобразования Фурье (БПФ или FFT), для которого вычислительная сложность составляет $O(N^2 \log N)$. В настоящее время алгоритмы БПФ реализованы в боль-

* Работа выполнена при поддержке РФФИ (грант № 14-05-00862-а).

шинстве вычислительных библиотек [3–5] и оптимизированы для выполнения на различных вычислительных архитектурах.

Стоит отметить, что все перечисленные стандартные реализации БПФ (FFT) предполагают, что входные и выходные функции заданы на регулярных сетках. Однако в случае преобразования Радона оно может быть реализовано путём выполнения серии одномерных дискретных преобразований Фурье и двумерного обратного преобразования Фурье на нерегулярных сетках. Если первый этап может быть реализован в форме стандартного БПФ, то преобразование Фурье на нерегулярных сетках требует использования отдельных алгоритмов. Сравнительно недавно были разработаны быстрые алгоритмы для вычисления данного преобразования (так называемые Unequally-Spaced Fast Fourier Transform, или USFFT) [6, 7], которые обеспечивают тот же порядок вычислительной сложности, что и БПФ.

Библиотеки, реализующие эти алгоритмы, уже начали появляться [8, 9], но не все они имеют параллельную версию. Так что задача их оптимизации для современных вычислительных платформ еще далека от завершения.

Алгоритмы быстрого преобразования Фурье на нерегулярных сетках

Рассмотрим ДПФ на нерегулярных сетках, заданные следующим образом:

$$F_k = \sum_{j=0}^{N-1} f_j e^{-2\pi i k x_j}, \quad (1)$$

$$g_j = \sum_{k=-N/2}^{N/2-1} G_k e^{-2\pi i k x_j}. \quad (2)$$

Здесь рациональные числа $\{x_j\}_{j=0}^{N-1}$, для которых $|x_j| < 1/2$, задают нерегулярную сетку, а целые числа $k = -N/2, \dots, N/2 - 1$ задают регулярную сетку. Значения f_j, g_j, F_k, G_k принадлежат множеству комплексных чисел. Вычисление сумм в формулах (1) и (2) напрямую выполняется за $\mathcal{O}(N^2)$ операций, в случаях двумерного и трехмерного аналогов данного преобразования вычислительная сложность равна $\mathcal{O}(N^4)$ и $\mathcal{O}(N^6)$ соответственно. Рассмотрим эвристическое описание работы алгоритмов USFFT для быстрого вычисления преобразования (1), более подробная формулировка и оценки точности представлены в [6].

Начнём с того, что экспоненциальные множители внутри сумм могут быть представлены в виде свёртки с дельта-функцией,

$$e^{-2\pi i k x_j} = \int_{-\infty}^{\infty} e^{-2\pi i k x} \delta(x - x_j) dx. \quad (3)$$

Тогда сумма в (1) может быть представлена в следующем виде:

$$\sum_{j=0}^{N-1} f_j \cdot e^{-2\pi i k x_j} = \int_{-\infty}^{\infty} \sum_{j=0}^{N-1} f_j \delta(x - x_j) e^{-2\pi i k x} dx. \quad (4)$$

Также введём гауссиан в форме $\phi(x) = e^{-\lambda x^2}$, для некоторого параметра λ , контролирующего ширину гауссиана. Умножим выражение (4) с обеих сторон на преобразование Фурье от гауссиана, обозначенное как $\hat{\phi}(k)$, тогда

$$\hat{\phi}(k) \sum_{j=0}^{N-1} f_j \cdot e^{-2\pi i k x_j} = \int_{-\infty}^{\infty} \sum_{j=0}^{N-1} f_j \phi(x - x_j) e^{-2\pi i k x_j} dx. \quad (5)$$

Интеграл в (5) может быть аппроксимирован при помощи правила трапеции, потому что получаемая под интегралом функция является гладкой за счёт свёртки с гауссианом. Поэтому достаточно точные результаты аппроксимации интеграла при помощи суммы могут быть получены на регулярной сетке x с повышенной частотой дискретизации (в соответствии с фактором ν , называемым oversampling factor): введём $\left\{\frac{l}{\nu N}\right\}_{l=-\frac{\nu N}{2}}^{\frac{\nu N}{2}}$ в качестве регулярной сетки для x .

Таким образом, алгоритм преобразования Фурье на нерегулярных сетках может быть представлен как последовательное применение операции дискретной свертки с гауссианом, БПФ и операции деконволюции (операция обратная свертке, необходима для получения спектра исходного сигнала, который был модифицирован в результате свертки). Стоит отметить тот факт, что имеется возможность вычислять значение функции $\phi\left(\frac{l}{\nu N} - x_j\right)$ только для таких значений аргумента $\left(\frac{l}{\nu N} - x_j\right)$, которые находятся в небольшой (относительно 1) окрестности точки x_j , так как она является быстроубывающей, и её можно рассматривать, как имеющую ограниченный численный носитель. Причем размер окрестности зависит только от наперед заданной точности вычислений ε . В таком случае операция дискретной свертки становится эквивалентной операции взвешенного рассеивания.

В итоге формула аппроксимации для (1) выглядит следующим образом:

$$\sum_{j=0}^{N-1} f_j \cdot e^{-2\pi i k x_j} \approx \frac{1}{\hat{\phi}(k)} \frac{1}{\nu N} \sum_{l=-\frac{\nu N}{2}}^{\frac{\nu N}{2}-1} \left(\sum_{j=0}^{N-1} f_j \phi\left(\frac{l}{\nu N} - x_j\right) \right) e^{-2\pi i \frac{kl}{\nu N}}, \quad (6)$$

где сумма по переменной l состоит из $L = 2M + 1 \ll N$ (параметр M зависит от требуемой точности вычислений) ненулевых элементов в виду ограниченного численного носителя функции $\phi(x)$.

Отметим, что на практике используется oversampling factor $\nu = 2$, так как он является оптимальным с точки зрения вычислительных ресурсов и требуемой точности.

Алгоритмы (1) и (2) составлены в соответствии с формулой (6). Отметим, что операторы (1) и (2) будут сопряжёнными, если в одном из них заменить x_j на $-x_j$. Поэтому алгоритм для преобразования (2) может быть построен аналогичным образом, заменив операцию взвешенного рассеивания на взвешенный сбор и выполняя шаги алгоритма (1) в обратном порядке.

Алгоритм 1

1. Взвешенное рассеивание значений точек нерегулярной сетки на точки регулярной сетки в шаре радиуса $M = \lceil \sqrt{2.25/\pi \log \varepsilon} \rceil$ [10], где $\lceil x \rceil$ – округление x к наименьшему большему целому, ε – требуемая точность вычислений, с центром в точке $v_j = \lfloor 2N x_j \rfloor$, где $\lfloor x \rfloor$ – округление x к ближайшему целому. Весовая функция – гауссиан $\phi(x) = e^{-\lambda x^2}$, где $\lambda = \frac{-2\pi^2 N^2}{\log \varepsilon}$.

2. Дискретное преобразование Фурье на регулярной сетке размера $2N$, на которую производилось взвешенное рассеивание. Может быть выполнено с помощью алгоритма БПФ.

3. Корректирующий фильтр к шагу 1, т.е. умножение на $1/\hat{\phi}(k)$.

Алгоритм 2

1. Корректирующий фильтр к шагу 3, значение параметра λ вычисляется так же, как в преобразовании с нерегулярной сетки на регулярную.

2. Дискретное преобразование Фурье на исходной регулярной сетке размера $2N$. Может быть выполнено с помощью алгоритма БПФ.

3. Взвешенный сбор значений точек регулярной сетки, находящихся в шаре радиуса M с центром в точке $v_j = [2Nx_j]$. Весовая функция идентична аналогу из операции взвешенного рассеивания.

Операции взвешенного рассеивания и сбора имеют вычислительную сложность $O(MN)$, преобразование Фурье имеет сложность $O(N \log N)$ и корректирующий фильтр может быть выполнен за $O(N)$ операций. В итоге вычислительная сложность каждого из алгоритмов равна $O(MN + N \log N)$, что для $M \ll N$ аппроксимируется как $O(N \log N)$.

Оптимизация последовательной программы

В ходе профилирования последовательной версии преобразований обоих типов было установлено, что до 95 % времени вычисления преобразований обоих типов занимают этапы взвешенного рассеивания и взвешенного сбора. Эти этапы характеризуются малым количеством вычислений на каждое обращение в память, что приводит к большой зависимости времени вычисления от эффективности использования кэша.

Так как в общем случае точки нерегулярной сетки распределены случайным образом, доступ к точкам регулярной сетки, участвующим в сборе или рассеивании, является хаотическим. Следствием этого является большое количество кэш-промахов. Для того чтобы их уменьшить, необходимо организовать обход точек нерегулярной сетки таким образом, чтобы каждая последующая точка нерегулярной сетки имела как можно больше общих точек регулярной сетки совместно с предыдущей. Для достижения этого точки нерегулярной сетки были отсортированы в порядке возрастания координат. При многомерном преобразовании происходит последовательная сортировка сначала по одной координате, затем по остальным.

Данная оптимизация позволила уменьшить время взвешенного сбора в 2 раза при двумерном преобразовании и в 5 раз в трёхмерном. В случае одномерного преобразования накладные расходы по сортировке оказались больше выигрыша полученного в результате оптимизации попаданий в кэш.

Также было проверено то, насколько может возрасти производительность при такой организации доступа в память, при которой количество кэш-промахов было бы минимальным. Тестовый пример, в котором все точки нерегулярной сетки имели одну координату, что гарантировало присутствие в кэше задействованных в преобразовании данных, показал, что в идеальном случае количество кэш-промахов для преобразования может быть уменьшено в 36 раз (см. таблицу) (данные приведены для двумерной версии преобразования с регулярной на нерегулярную сетку размера $2^{12} \times 2^{12}$ для процессора Intel i7-4770).

Количество промахов в кэш последнего уровня

| Размер сетки | Без сортировки | С сортировкой | Идеальный случай |
|-----------------------------|----------------|---------------|------------------|
| 2^{24} | 4,6e+06 | 2,9e+07 | 1,1e+07 |
| $2^{12} \times 2^{12}$ | 9,5e+08 | 3,4e+07 | 2,6e+07 |
| $2^8 \times 2^8 \times 2^8$ | 7,6e+09 | 1,4e+08 | 5,4e+07 |

Параллельный алгоритм USFFT

Аналогично этапу оптимизации последовательной версии программы, основное внимание было уделено этапам взвешенного сбора и рассеивания. При выполнении взвешенного сбора цикл по точкам нерегулярной сетки возможно естественным образом распараллелить, ис-

пользуя технологию OpenMP. Каждый поток использует точки нерегулярной сетки только для чтения, поэтому распараллеливание в этом случае не представляет сложности.

При этом установлено, что на количество попаданий в кэш последнего уровня прямым образом влияет политика планирования выполнения потоков. Это особенно актуально для трёхмерного сбора, который характеризуется обработкой большого количества точек регулярной сетки, приходящихся на каждую точку нерегулярной.

Планирование выполнения необходимо организовать таким образом, чтобы в каждый момент времени различные потоки обрабатывали как можно более локальные данные для предотвращения конфликта в кэше последнего уровня, который является общим для всех ядер в рамках одного процессора.

Для этого необходимо разбить итерации цикла на блоки небольшого размера и установить динамическое распределение блоков между потоками. Оптимальный размер такого блока напрямую зависит от характеристик конкретного вычислительного устройства. Установлено, что, например, для системы с двумя процессорами Intel Xeon E5-2690 оптимальными являются блоки размера от 4 до 32 элементов.

Тестирование масштабируемости показало, что такой подход к распараллеливанию является достаточно эффективным – при использовании максимального количества доступных вычислительных ядер эффективность распараллеливания достигает 0,87 (ускорение в 13,9 раз на 16 ядрах) в случае двумерного преобразования, см. кривую с квадратами на рис. 1.

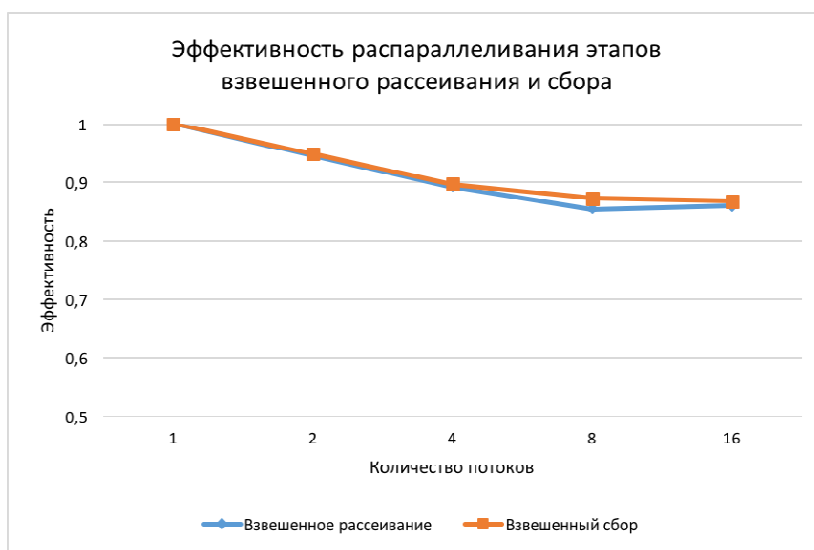


Рис. 1. Эффективность распараллеливания этапов взвешенного рассеивания и сбора на CPU

Однако такое распараллеливание цикла по точкам нерегулярной сетки невозможно при выполнении взвешенного рассеивания. Это связано с тем, что возможна ситуация, когда два потока будут производить запись в один и тот же участок памяти, что приведет к ошибке (race condition). Пример такой ситуации изображен на рис. 2. Видно, что существуют точки регулярной сетки, принадлежащие к окрестностям (области, ограниченные красным пунктиром) двух одновременно обрабатываемых точек нерегулярной сетки (красные крестики).

Для решения данной проблемы было предложено разбить множество точек регулярной сетки на группы блоков. Так, чтобы блоки, принадлежащие к одной группе, были расположены на расстоянии $2M + 1$ друг от друга, где M – радиус рассеивания. Тогда, для корректного и эффективного выполнения взвешенного рассеивания необходимо распараллелить цикл обработки блоков, принадлежащих одной группе. Здесь, под обработкой блока понимается проведение операции взвешенного рассеивания для всех точек нерегулярной сетки принадлежащих этому блоку.

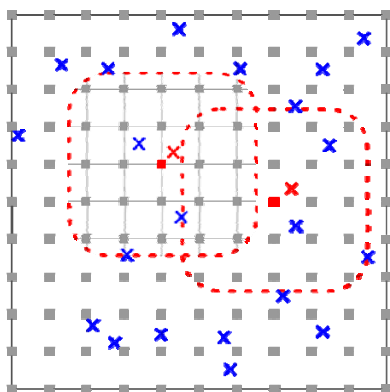


Рис. 2. Пример пересечения областей взвешенного рассеивания (пунктирные контуры)

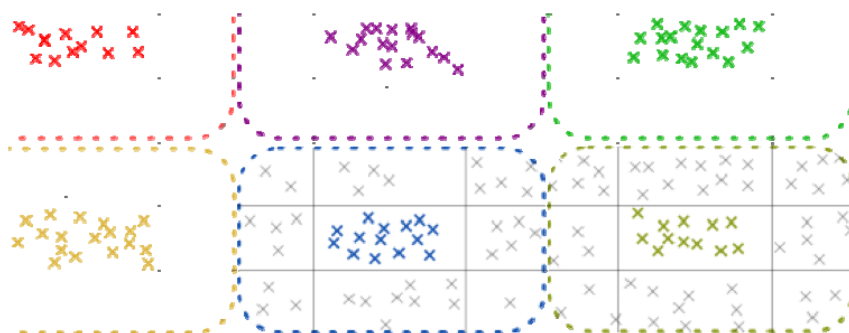


Рис. 3. Пример разбиения двумерной сетки на вычислительные блоки (обозначены пунктирные контуры)

На рис. 3 изображен пример такого разбиения. Цветными крестиками обозначены точки нерегулярной сетки, принадлежащие одной группе блоков, различные цвета этих крестиков обозначают принадлежность к различным блокам. Цветные пунктирные линии обозначают участки памяти, куда может быть произведена запись при обработке блока соответствующего цвета. Видно, что эти области не пересекаются, а значит, различные блоки могут быть обработаны независимо.

При равномерной плотности распределения точек нерегулярной сетки данный подход в двумерном случае достигает эффективности распараллеливания, равной 0,86 (ускорение в 13,7 раза на 16 потоках); эти результаты показаны кривой с ромбами на рис. 1. Меньшая, по сравнению с операций взвешенного рассеивания, эффективность распараллеливания связана в первую очередь с необходимостью барьерной синхронизации после обработки группы блоков.

Сравнение с существующими аналогами

Важным этапом при разработке параллельного алгоритма и реализации на его основе библиотеки (далее обозначена как библиотека USFFT), нацеленной на высокие показатели производительности, является её сравнение с существующими аналогами. В настоящее время наиболее распространенной реализацией преобразования Фурье на нерегулярных сетках является библиотека NFFT. Сравнение производилось на двухsocketном сервере с установленными на нем процессорами Intel Xeon E5-2690. В качестве тестов использовалась двумерная задача, со случайно распределенными точками нерегулярной сетки, количество регулярных и нерегулярных точек совпадает. Как видно из графиков на рис. 4 и 5 преобразование обоих типов из библиотеки USFFT в два-три раза быстрее своих аналогов из библиотеки NFFT.

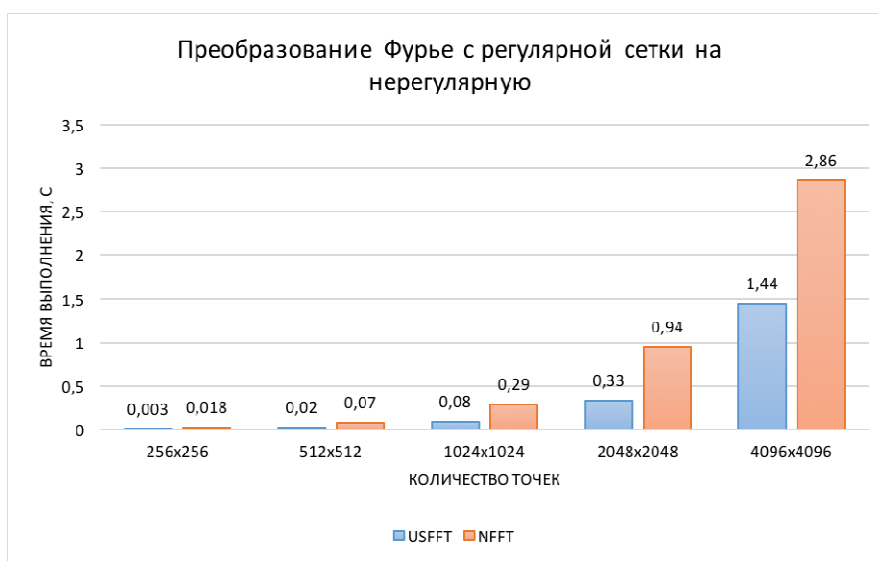


Рис. 4. Время выполнения двумерного преобразования Фурье с регулярной сетки на нерегулярную с использованием библиотек USFFT и NFFT для сеток разного размера

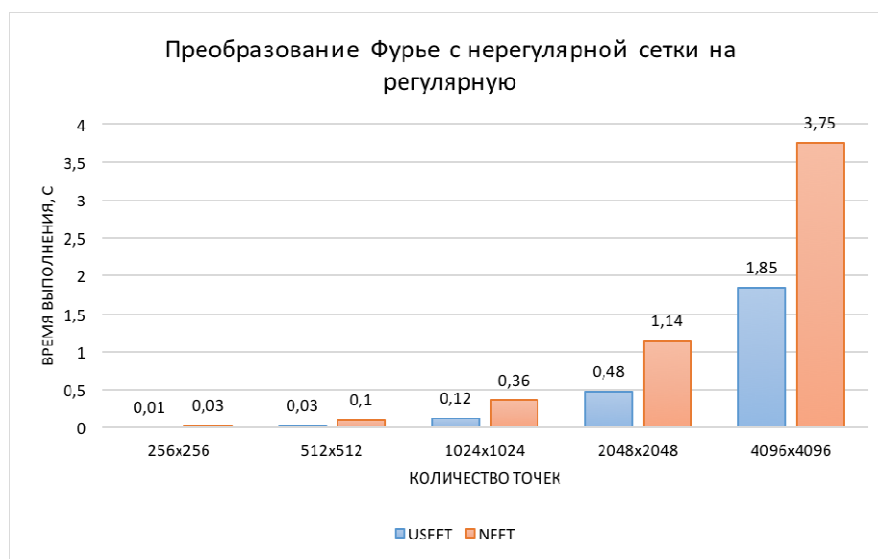


Рис. 5. Время выполнения двумерного преобразования Фурье с нерегулярной сетки на регулярную с использованием библиотек USFFT и NFFT для сеток разного размера

Параболическое преобразование Радона

Процесс сейсмической разведки может быть описан следующим образом. Специальные источники генерируют сейсмические волны, которые распространяются в геологической среде. При этом, резкие (по сравнению с длиной волны) изменения механических свойств среды выступают в роли внутренних границ и вызывают отражение волн, после отражения волны распространяются обратно на поверхность, где они регистрируются приемниками. Записанный сигнал может быть использован для построения изображения границ геологической среды, которое в свою очередь может быть использовано для разведки полезных ископаемых. Сигнал для каждого приемника называется сейсмотрассой, а отсортированная комбинация сейсмотрасс – сейсмограммой.

Отдельное внимание при проведении сейсморазведочных работ заслуживает этап предварительной обработки сейсмограмм, который подразумевает применение следующих процедур: увеличение отношения сигнал-шум, удаление кратных волн, регуляризация и интерполяция данных. Этот этап очень важен, так как в противном случае последующие этапы обработки могут иметь высокие ошибки, а в некоторых случаях могут быть неприменимы.

Существует множество методов для удаления кратных волн с различной степенью качества и сложности [11]. Как правило, проблема решается с помощью параболического или гиперболического преобразования Радона, так как параболы и гиперболы хорошо подходят для приближений сейсмических волн.

Параболическое преобразование Радона может быть определено несколькими способами. Определение, используемое при обработке сейсмических данных, пожалуй, является наиболее простым для понимания [12]. Преобразованием Радона непрерывной функции двух переменных называется функция

$$g(\tau, p) = \mathcal{R}_p f(p, \tau) = \int_{-\infty}^{\infty} f(\tau + px^2, x) dx \quad (7)$$

Это преобразование имеет простой геометрический смысл – это интеграл от функции вдоль параболы с вершиной в точке τ и кривизной p . Пример функции f и результаты применения к ней параболического преобразования Радона представлены на Рис. 6. Видно, что количество локальных максимумов функции g соответствует количеству парабол на изображении f .

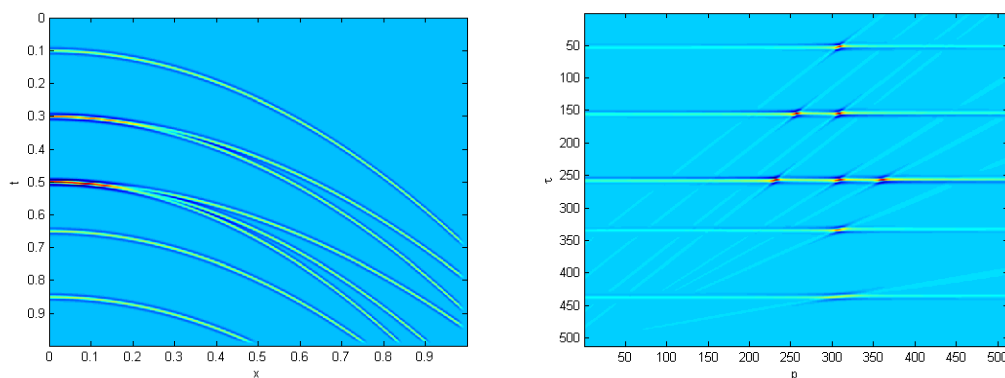


Рис. 6. Пример начальной функции f (слева) и результаты ее параболического преобразования Радона $\mathcal{R}_p f$ (справа)

Вычислительная сложность такого преобразования весьма высока и составляет $\mathcal{O}(N^3)$ (размерности входных и выходных данных совпадают и равны N), что приводит к высоким вычислительным затратам при большом размере входных данных.

Для более эффективного вычисления данного преобразования воспользуемся методом Фурье-синтеза [13]. Для начала, запишем определение непрерывного параболического преобразования Радона с помощью дельта-функции:

$$\mathcal{R}_p f(p, \tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, x) \delta(t - \tau - px^2) dt dx \quad (8)$$

Затем выполним прямое и обратное преобразование Фурье по переменной t исходной функции и воспользуемся свойством преобразования Фурье от дельта-функции:

$$\begin{aligned}
\mathcal{R}_p f(\tau, p) &= \mathcal{R}_p \left(\mathcal{F}_{\eta \rightarrow t}^{-1} \mathcal{F}_{t \rightarrow \eta} f(t, x) \right) (\tau, p) \\
&= \int \left(\iint \left(\int f(t', x) e^{-2\pi i \eta t'} dt' \right) e^{2\pi i \eta t} \delta(t - \tau - px^2) dt d\eta \right) dx \\
&= \iint \left(\int f(t', x) e^{-2\pi i \eta t'} dt' \right) e^{2\pi i \eta \tau + 2\pi i (\eta x^2) p} d\eta dx \\
&= \mathcal{F}_{(\eta, \eta x^2) \rightarrow (\tau, p)}^{-1} \mathcal{F}_{t \rightarrow \eta} f(t, x)
\end{aligned} \tag{9}$$

Таким образом, параболическое преобразование Радона может быть вычислено, как последовательное применения серии одномерных преобразований Фурье по координате t и обратного двумерного преобразования Фурье с нерегулярной сетки $(\eta, \eta x^2)$ на регулярную сетку (τ, p) . Соответствующее ему сопряжённое преобразование \mathcal{R}_p^* , в таком случае, может быть вычислено как двумерное преобразование Фурье с регулярной сетки на нерегулярную сетку $(\eta, \eta x^2)$ с последующим применением серии обратных одномерных преобразований Фурье.

Заключение

Предложены параллельные алгоритмы прямого и сопряжённого параболического преобразования Радона, а также проведена оптимизация скорости их исполнения на современных центральных процессорах.

Основное внимание при оптимизации было уделено наиболее сложным, с вычислительной точки зрения, этапам данных преобразований – преобразованиям Фурье на нерегулярных сетках (USFFT) различных типов. Так, были разработаны и реализованы соответствующие параллельные алгоритмы USFFT. Тестирование показало, что полученные реализации преобразований USFFT до трёх раз быстрее своих аналогов из библиотеки NFFT.

Проведено тестирование полученного алгоритма параболического преобразования Радона на синтетических данных, показана его применимость для задач обработки сейсмических данных.

Список литературы

1. Herman G. T., Louis A. K., Natterer F. (ed.). *Mathematical methods in tomography: proceedings of a conference held in Oberwolfach, Germany, 5–11 June, 1990*. Springer, 2006.
2. Yilmaz O. *Seismic data analysis*. Tulsa: Society of exploration geophysicists, 2001. Vol. 1. 2065 p.
3. Intel Math Kernel Library (Intel MKL). Адрес доступа: <https://software.intel.com/en-us/intel-mkl> (дата обращения 07.10.2016).
4. cuFFT | NVIDIA Developer. Адрес доступа: <https://developer.nvidia.com/cufft> (дата обращения 07.10.2016).
5. FFTW Home page. Адрес доступа: <http://www.fftw.org> (дата обращения 07.10.2016).
6. Dutt A., Rokhlin V. Fast Fourier transforms for nonequispaced data // *SIAM Journal on Scientific computing*. 1993. Vol. 14. No. 6. P. 1368–1393.
7. Beylkin G. On the fast Fourier transform of functions with singularities // *Applied and Computational Harmonic Analysis*. 1995. Vol. 2. No. 4. P. 363–381.
8. NUFFT page. Адрес доступа: <http://www.cims.nyu.edu/cmcl/nufft/nufft.html> (дата обращения 07.10.2016).
9. NFFT – TU Chemnitz. Адрес доступа: [https://www-user.tu-chemnitz.de/~sim\\$potts/nfft/](https://www-user.tu-chemnitz.de/~sim$potts/nfft/) (дата обращения 07.10.2016).
10. Andersson F. Algorithms for unequally spaced fast Laplace transforms // *Applied and Computational Harmonic Analysis*. 2013. Vol. 35. No. 3. P. 419–432.
11. Verschuur D. J. *Seismic multiple removal techniques: past, present and future*. EAGE publications, 2006. 191 p.
12. Toft P. A., Sørensen J. A. *The Radon Transform – Theory and Implementation*. Kgs. Lyngby, Denmark: Technical University of Denmark, 1996. 327 p.

13. Schonewille M. A., Duijndam A. J. W. Parabolic Radon transform, sampling and efficiency // Geophysics. 2001. Vol. 66. No. 2. P. 667–678.

Материал поступил в редколлегию 15.07.2016

A. S. Matveev¹, **V. V. Nikitin**², **A. A. Duchkov**^{1,3}, **A. A. Romanenko**³

¹ *IPGG SB RAS*

³ *Academician Koptyug, Novosibirsk, 630090, Russian Federation*

² *MAX IV Laboratory*

² *Fotogatan, 225 92, Lund, Sweden*

³ *Novosibirsk State University*

³ *1 Pirogov Str., Novosibirsk, Russian Federation*

alexey.matveev.9@gmail.com, vnikitin90@gmail.com

DuchkovAA@ipgg.sbras.ru, romanenko.alexey@gmail.com

PARALLEL IMPLEMENTATION OF PARABOLIC RADON TRANSFORM BASED ON UNEQUALLY-SPACED FAST FOURIER TRANSFORM

The article presents parallel algorithm for computing parabolic Radon transformation – summation of 2D function along parabolas with varying curvature and vertical shift of its apex. The basis of the computational algorithm is a parallel implementation of the Unequally-Spaced Fast Fourier Transform optimized for architecture of modern CPUs. We give a description of optimizations applied to the algorithm and the results of testing and comparison with alternative existing libraries. We also present an application of the parabolic Radon transform to seismic data processing.

Keywords: seismic data processing, Radon transform, Unequally-Spaced Fast Fourier Transform, parallel algorithm.