

Применение параллельно-рекурсивных методов вейвлет-преобразования для обработки двумерных сигналов

Нго Хыу Фук

В статье рассмотрены современные параллельно-рекурсивные методы выполнения вейвлет-преобразований для одномерных и двумерных дискретных сигналов. Предлагаемый автором алгоритм параллельно-рекурсивного вейвлет-преобразования эффективным образом реализуется на параллельных вычислительных системах

Вейвлеты являются математическим инструментом для иерархического представления функций. Предлагаемый алгоритм предназначен для увеличения скорости выполнения дискретного вейвлет-преобразования, используемого при обработке сигналов.

В обоих случаях мы предполагаем, что базисные функции $\phi(x)$ и $\psi(x)$ компактно определены [1]. Это автоматически гарантирует финитность последовательностей коэффициентов h_n и g_n . Далее предположим, что сигнал, подвергаемый преобразованию, имеет длину $N = 2^d, d \in \mathbb{Z}^+$.

Рассмотрим сигнал – последовательность чисел $x = (x_0, x_1, \dots, x_{N-1})$ и матрицы H и G.

$$H_{i,j} := h_{j-2i} \tag{1}$$

где для $\{h_i\}$ должно выполняться условие $\sum_{i=0}^{n-1} h_i = 2$.

При длине фильтра $size=2$ и $N=8$, матрица H имеет вид:

$$H = \begin{bmatrix} h_0 & h_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_0 & h_1 \end{bmatrix}$$

Матрица G – фильтр детали:

$$G_{i,j} := g_{j-2i} \tag{2}$$

где $g_k = (-1)^k h_{n-1-k}$.

При длине фильтра $size = 2$ и $N=8$, матрица G имеет вид:

$$G = \begin{bmatrix} g_0 & g_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{bmatrix}$$

Матрицы H и G должны удовлетворять условиям:

$$H^T H + G^T G = I, \quad GH^T = HG^T = 0, \quad GG^T = HH^T = I \quad (3)$$

Матрицы H и G определены следующим образом:

$$H_{i,j} = \begin{cases} h_{j-2i} & \text{если } (j-2i \geq 0) \text{ и } (j-2i < \text{size}) \\ h_{N+j-2i} & \text{если } (j-2i < 0) \text{ и } (N+j-2i \geq 0) \text{ и } (N+j-2i < \text{size}) \text{ и } (2i < N) \\ 0 & \text{иначе} \end{cases} \quad (4)$$

$$G_{i,j} = \begin{cases} g_{j-2i} & \text{если } (j-2i \geq 0) \text{ и } (j-2i < \text{size}) \\ g_{N+j-2i} & \text{если } (j-2i < 0) \text{ и } (N+j-2i \geq 0) \text{ и } (N+j-2i < \text{size}) \text{ и } (2i < N) \\ 0 & \text{иначе} \end{cases} \quad (5)$$

Мы можем вычислять по процедуре:

$$h(i, j, N, \text{size}) = \begin{cases} h_{j-2i} & \text{если } (j-2i \geq 0) \text{ и } (j-2i < \text{size}) \\ h_{N+j-2i} & \text{если } (j-2i < 0) \text{ и } (N+j-2i \geq 0) \text{ и } (N+j-2i < \text{size}) \text{ и } (2i < N) \\ 0 & \text{иначе} \end{cases} \quad (4')$$

$$g(i, j, N, \text{size}) = \begin{cases} g_{j-2i} & \text{если } (j-2i \geq 0) \text{ и } (j-2i < \text{size}) \\ g_{N+j-2i} & \text{если } (j-2i < 0) \text{ и } (N+j-2i \geq 0) \text{ и } (N+j-2i < \text{size}) \text{ и } (2i < N) \\ 0 & \text{иначе} \end{cases} \quad (5')$$

При $\text{size}=4$ и $N=8$, матрица H имеет вид:

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \end{bmatrix}$$

Мы определим $c_{j,k}$ и $d_{j,k}$ - значения коэффициентов вейвлета уровня j ($0 \leq j \leq \log_2 n$). При уровне рекурсии 0, $x = \{c_{0,k}\}_0^{n-1}$.

Можно записать значения уровня j :

$$c_{j,k} = \sum_i h_{i-2k} c_{j-1,i}, \quad d_{j,k} = \sum_i g_{i-2k} c_{j-1,i} \quad (6)$$

При $\text{size}=4$ и $N=8$, можно вычислять значения уровня 1 по матрице H и G:

$$\begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \end{bmatrix} = \sqrt{2} \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{0,4} \\ c_{0,5} \\ c_{0,6} \\ c_{0,7} \end{bmatrix}$$

и

$$\begin{bmatrix} d_{1,0} \\ d_{1,1} \\ d_{1,2} \\ d_{1,3} \end{bmatrix} = \sqrt{2} \begin{bmatrix} g_0 & g_1 & g_2 & g_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & g_3 \\ g_2 & g_3 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{0,4} \\ c_{0,5} \\ c_{0,6} \\ c_{0,7} \end{bmatrix}$$

Алгоритм параллельно-рекурсивных вейвлетов основан на принципе «разделяй и властвуй» (рис. 1). Полное разбиение получим на вейвлет-преобразованиях Хаара. С другим вейвлет-преобразованием нам нужно переносить $size-2$ элементов для каждой обрабатываемой части. Однако, время коммуникации на каждом шаге составляет $\Theta(0)$.

Предполагаем, что часть номер D , имеет длину $D = 2^k, k \in Z^+$ и $D \leq N$.

Алгоритм обработки участка $\{c_{0,k}\}_{k=0}^{N-1}$:

- Элементы $\{c_{0,k}\}_{k=0}^{N-1}$ разделены на D частей, $\{c_{0,k}^0\}$, $\{c_{0,k}^1\}$, ..., $\{c_{0,k}^{D-1}\}$, если возможно. Эту операцию мы будем называть разделением (процедура *Divide*).
- Процедура выполнения вейвлетов рекурсивно вызывается для $\{c_{0,k}^0\}$, $\{c_{0,k}^1\}$, ..., $\{c_{0,k}^{D-1}\}$.
- В результате этой процедуры получим - $\{c_{1,k}\}$ и $\{d_{1,k}\}$.
- Повторяем процедуру разделения и процедуру вычисления вейвлетов для $\{c_{1,k}\}$.
- Количество уровней рекурсии - $\log_2(N) - \log_2(size) + 1$.

После этого сигнал $\{c_{0,k}\}$ обработан.

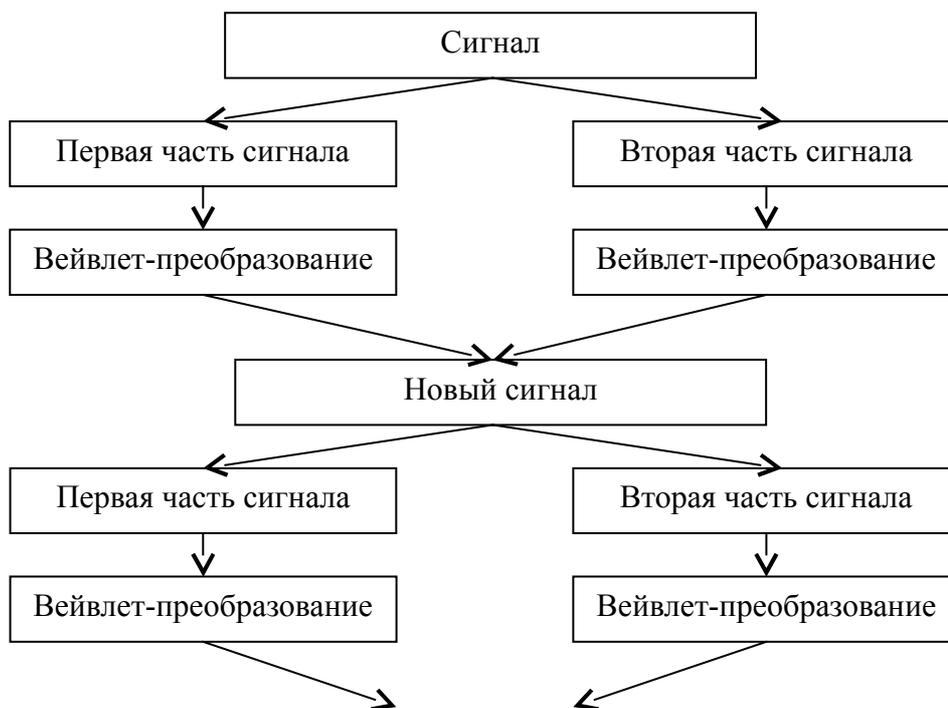


Рисунок 1: Схема алгоритма параллельно-рекурсивного вычисления вейвлетов, основанного на принципе «разделяй и властвуй».

Итак, процедура выполнения вейвлетов WAVEONE_D выглядит следующим образом:

WAVEONE_D(c,D)

1. T:=c

2. $N \leftarrow \text{length}(c)$

3. **for** k:=0 to $\log_2(N) - \log_2(\text{size})$ **do**

STemp \leftarrow 0

Temp \leftarrow 0

if $\text{length}(c)=D$ **then** $D:=D/2$

for i:=0 to D-1 **do**

$S^i \leftarrow \text{Divide}(T,i,D)$

$Temp^i \leftarrow \text{waveone}(S^i)$

end for

$K \leftarrow \text{length}(Temp^0)$

if $(K \leq 1)$ **then**

$T \leftarrow \text{Divide}(T,0,1)$

$T \leftarrow \text{waveone}(T)$

for i:=0 to D-1 **do**

```

         $c_i \leftarrow T_i$ 
    end for
else
T ← 0
for i:=0 to K/2 – 1 do
    for j:=0 to D-1 do
         $T_{i+a\frac{K}{2}} \leftarrow Temp_i^j$ 
    end for
end for
for i:=K/2 to K-1 do
    for j:=0 to D-1 do
         $c_{i+K(D-1)-a\frac{K}{2}} \leftarrow Temp_i^j$ 
    end for
end for
end if
4. Return c

```

где c – исходный сигнал; D – количество частей сигнала.

Основной шаг алгоритма – процедура Divide, которая разбивает сигнал $\{c_{0,k}\}$ на D частей.

Процедура выглядит следующим образом:

```

Divide(S,t,D)
1.  $n \leftarrow \text{length}(S)$ 
2. if size-2>0 then
    for i:=0 to size-2 do
         $S_{n+i} \leftarrow S_i$ 
    end for
3. for i:=0 to n/D -1 +(size-2) do
         $T_i \leftarrow S_{i+n/D}$ 
    end for
4. Return T

```

где t – индекс части ($0 \leq t \leq D - 1$); S – сигнал.

Работа процедуры Divide показана на рис. 2.2. Время работы этой процедуры, составляет $\Theta(n/D)$. Если мы используем только адреса значений сигнала, то время работы процедуры составляет $\Theta(0)$.

Процедура waveone выполняет вейвлет-преобразование (6). Процедура выглядит следующим образом:

waveone(S)

1. $n \leftarrow \text{length}(S) - (\text{size} - 2)$

2. $K \leftarrow \text{length}(S)$

3. **for** $i := 0$ **to** $n/2 - 1$

$$T_i \leftarrow \sum_j h(i, j, K) \cdot S_j$$

$$T_{n-i-1} \leftarrow \sum_j g(i, j, K) \cdot S_j$$

end for

4. **Return** T

где S – сигнал.

Время работы алгоритма параллельно-рекурсивных вейвлетов зависит от архитектуры компьютера. Для параллельной системы время работы алгоритма зависит от D (в этом случае, время коммуникаций между процессорами системы является маленьким). Процедура *waveone* занимает время $T(n)$

Пусть t_{start} – время, которое требуется, чтобы начать коммуникации.

t_{one_comm} – время, которое требуется, чтобы послать одно двойное число другому процессору.

t_{one_comp} – время, которое требуется, чтобы вычислить одну операцию с плавающей запятой на данном компьютере.

а. Временные затраты на вычисления: Пусть в системе D процессоров, на каждом шаге каждый процессор работает на протяжении времени $T(k)/D$, где $k: 0 \dots \log_2(N) - \log_2(\text{size}) + 1$.
Время работы алгоритма составляет:

$$t_{compute} = \frac{T(0)}{D} + \frac{T(1)}{D} + \dots + \frac{T(L)}{D} = \frac{T(N)}{D} * t_{one_comp} \quad (7)$$

где $L = \log_2(N) - \log_2(size) + 1$; $T(N)$ - время выполнения вейвлет-преобразования на одном процессоре.

б. Временные затраты на обмен данными между процессорами: время коммуникаций между процессорами системы составляет:

$$t_{communication} = O(\log N) \approx C * t_{one_comm} \quad (8)$$

где C – константа.

в. Временные затраты для полного преобразования составляют:

$$T_{full} = t_{compute} + L * (t_{communication} + t_{start}) \quad (9)$$

Модифицируем процедуру *Divide*, чтобы уменьшать время, затрачиваемое на обмен данными между процессорами. Модифицированная процедура выглядит следующим образом:

Divide($t, size, D, number$)

1. отклонение $\leftarrow size - 2$

2. $T_0 \leftarrow t \frac{number}{D}$

3. $T_1 \leftarrow (t + 1) \frac{number}{D} - 1 + \text{отклонение}$

4. **Return** T

где *number* – количество элементов сигнала для преобразования.

Использовать вейвлеты для преобразования значений двумерных дискретных можно двумя способами. Каждое из этих преобразований представляет собой двумерное обобщение вейвлет-преобразования, описанного выше.

Первое преобразование называется *стандартным разложением*. Чтобы получить стандартное разложение изображения, сначала мы применим одномерное вейвлет-преобразование к каждой строке значений пикселей. Эта операция даст нам среднее значение и уточняющие коэффициенты для каждой строки. Затем мы рассмотрим эти преобразованные строки так, как если бы они сами являлись исходным сигналом, и применим одномерное преобразование к каждому столбцу. Полученные в результате значения окажутся уточняющими коэффициентами, за исключением

единственного коэффициента, представляющего общее среднее значение. Ниже приведен алгоритм вычисления стандартного разложения.

STANDARDWAVE(M)

1. **For** i:=0 to rows(M)-1 **do**

$M^i \leftarrow \text{WAVEONE_D}(M^i, D)$

end for

2. $M \leftarrow M^T$

3. **For** i:=0 to cols(M)-1 **do**

$M^i \leftarrow \text{WAVEONE_D}(M^i, D)$

end for

4. **Return** M

где M – матрица сигнала.

Второй тип двумерного вейвлет-преобразования, так называемое *нестандартное разложение*, представляет чередование операций над строками и столбцами. Сначала мы выполняем один этап горизонтального попарного усреднения и нахождения разности значений сигнала в каждой строке матрицы сигнала. Затем мы применяем попарное усреднение и нахождение разностей к каждому получившемуся столбцу. Чтобы закончить преобразование, мы рекурсивно повторяем этот процесс только на квадратах, содержащих средние значения в обоих направлениях. Перед тем как записать процедуру нестандартного разложения, нужно переписать процедуру WAVEONE_D следующим образом:

nonWAVEONE_D(c,D,N)

1. T:=c

2. setlength(T):=N

3. Похоже WAVEONE_D

4. **Return** c

В отличие от процедуры WAVEONE_D, в этой процедуре нам нужно установить длину вектора T – N (на шаге 2), потому что в каждом шаге, размер матрицы в преобразовании уменьшается от N до N/2.

Процедура NONSTANDARDWAVEONE_D выглядит следующим образом:

NONSTANDARDWAVE(M)

1. $N \leftarrow \text{rows}(M)$
2. **for** $k:=0$ **to** $\log_2 N - \log_2 \text{size}$ **do**
3. **For** $i:=0$ **to** $N-1$ **do**
 $M^i \leftarrow \text{nonWAVEONE_D}(M^i, D, N)$
end for
4. $M \leftarrow M^T$
5. **For** $i:=0$ **to** $N-1$ **do**
 $M^i \leftarrow \text{nonWAVEONE_D}(M^i, D, N)$
end for
6. $N \leftarrow N/2$
end for
4. **Return** M

Оба метода разложения двумерного изображения приводят к коэффициентам, соответствующим двум различным множествам базисных функций. Стандартное разложение изображения дает коэффициенты для базиса, образованного в соответствии со стандартным построением двумерного базиса. Аналогично, нестандартное разложение изображения дает коэффициенты для нестандартного построенного базиса функций.

Стандартное построение двумерного базиса вейвлетов состоит во взятии всевозможных тензорных произведений функций одномерного базиса. Например, если вначале мы имеем одномерный базис для пространства V^2 , то мы получим двумерный базис для пространства V^2 . Заметим, что если мы применим стандартное конструирование к одномерному ортонормированному базиса, то в итоге получим двумерный ортонормированный базис.

Нестандартное построение двумерного базиса начинается с определения двумерных масштабирующих функций[2,3]

$$\phi\phi(x, y) = \phi(x)\phi(y) \quad (12)$$

и трех вейвлетов:

$$\phi\psi(x, y) = \phi(x)\psi(y) \quad (13)$$

$$\psi\phi(x, y) = \psi(x)\phi(y) \quad (14)$$

$$\psi\psi(x, y) = \psi(x)\psi(y) \quad (15)$$

Обозначим уровни масштабирования надстрочным знаком j , а горизонтальные и вертикальные сдвиги подстрочными индексами k и l , соответственно. Нестандартный базис состоит из

единственной масштабирующей функции, соответствующей грубому приближению $\phi\phi_{0,0}^0(x, y) = \phi\phi(x, y)$, а также трех масштабированных и сдвинутых вейвлетов $\phi\psi$, $\psi\phi$, $\psi\psi$:

$$\phi\psi_{kl}^j(x, y) = 2^j \phi\psi(2^j x - k, 2^j x - l) \quad (13')$$

$$\psi\phi_{kl}^j(x, y) = 2^j \psi\phi(2^j x - k, 2^j x - l) \quad (14')$$

$$\psi\psi_{kl}^j(x, y) = 2^j \psi\psi(2^j x - k, 2^j x - l) \quad (.15')$$

Постоянная 2^j нормирует вейвлеты для образования ортонормированного базиса.

Мы представили здесь два подхода к вейвлет-преобразованиям и базисным функциям – стандартный и нестандартный, - потому что у каждого из них имеются свои преимущества. Стандартное разложение изображения привлекательно простотой выполнения – нужно только произвести одномерное преобразование всех строк, а затем и всех столбцов. С другой стороны, чуть более эффективным представляется нестандартное разложение сигнала.

Например, в случае преобразования Хаара с длиной фильтра 2 для матрицы сигнала $N \times N$, стандартное разложение требует $4(N^2 - N)$ операций, тогда как нестандартное разложение требует $\frac{8}{3}(N^2 - 1)$ операций.

Еще один момент касается носителя каждой базисной функции, обозначающего участок области определения функции, на котором функция не равна нулю. Все функции нестандартно построенного базиса имеют квадратные носители, в то время как некоторые функции стандартно построенного базиса имеют неквадратные носители [4]. Каждый из этих вариантов может оказаться предпочтительнее другого, в зависимости от конкретного случая применения.

Литература:

1. Daubechies I. *Orthonormal Bases of Compactly Supported Wavelets*. //Communication on Pure Applied Mathematics.- 1988, vol.41.-p. 906-966.
2. Daubechies I. *The wavelet transform, time-frequency localization and signal analysis*. //IEEE Trans. Inform. Theory.- 1990, №5. -p.961-1005.
3. Daubechies I. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.
4. Астафьева Н.М. *Вейвлет-анализ: основы теории и некоторые приложения*. //Успехи физических наук.- 1996, №11.- с.1145-1170.

СВЕДЕНИЯ ОБ АВТОРЕ

Нго Хьу Фук, аспирант кафедры вычислительной математики и программирования Московского авиационного института (государственного технического университета)
e-mail: ngohuiphuc76@mail.ru