

О параллельном вэйвлетно-сплайновом сжатии на локально квазиравномерной сетке*

Ю.К. Демьянович, О.М. Косоголов

В работе рассматриваются вэйвлетные разложения системы вложенных пространств сплайнов. На их основе строятся и исследуются алгоритмы обработки (в т.ч. сжатия) потоков числовой информации, которые целесообразно применять к потокам с резко меняющимися характеристиками. Предлагается программная реализация алгоритма. Рассматривается параллельный алгоритм, позволяющий полностью загрузить двухъядерную или двухпроцессорную архитектуру. Алгоритмы восстановления позволяют провести восстановление с потерей (осуществив сжатие) или (при необходимости) полностью восстановить исходный числовой поток.

1. Введение

1.1. Общие сведения

Известно, что на двукратно измельчающейся сетке полиномиальные В-сплайны образуют телескопическую систему пространств, на основе которой строятся вэйвлетные разложения (см. [1–5]); для равномерной сетки это устанавливается с помощью преобразования Фурье (см., например, [3]), а для неравномерной сетки - использованием специального дробно-рационального тождества (см. [6–8]). Применение неравномерной сетки позволяет улучшить приближение функций без усложнения вычислений. Отметим, что для дальнейшего улучшения приближения могут понадобиться различные степени измельчения сетки в разных частях рассматриваемого промежутка (двукратное измельчение недостаточно). Особый интерес представляет вэйвлетное разложение в случае неравномерной сетки, т.к. обычно применяемое на равномерной сетке преобразование Фурье в условиях неравномерной сетки использовать затруднительно. Использование биортогональной системы функционалов позволяет построить вэйвлетные разложения при произвольном измельчении сетки.

Рассмотрим вэйвлетное разложение телескопической системы пространств В-сплайнов второй степени при произвольном способе измельчения сетки (вывод формул см. в работе [8]). Первоначально с каждой сеткой связываются В-сплайны второй степени. Если новая сетка получена из исходной удалением одного узла, то пространство минимальных сплайнов, построенных для крупной сетки, содержится в пространстве аналогичных сплайнов, построенных для мелкой сетки. Использование биортогональной (к В-сплайнам) системы функционалов позволяет получить прямое сплайн-вэйвлетное разложение и соответствующие формулы реконструкции и декомпозиции. Последовательное удаление или добавление узлов в исходную сетку приводит к телескопическим системам пространств рассматриваемых сплайнов и к прямому разложению исходного пространства. От исходной сетки можно перейти к любой более мелкой сетке добавлением любой конечной совокупности узлов, не совпадающих с узлами расширяемой сетки (между любыми двумя узлами исходной сетки число добавляемых узлов может быть различным в зависимости от выбранной пары соседних узлов). Формулы декомпозиции и реконструкции получаются использованием соответствующих биортогональных систем функционалов. Все результаты верны в случаях конечного отрезка и бесконечного интервала.

1.2. Декомпозиция и реконструкция

Построение вэйвлетного разложения определяется формулами декомпозиции. Пусть $C = \{c_i\}$ – исходный поток, каждое значение c_i которого поставлено в соответствие узлу x_i сетки $X = \{x_i\}$. Результатом разложения является основной поток $A = \{a_i\}$ и вэйвлетные коэффи-

*Работа частично поддержана грантами РФФИ 07-01-00269 и 07-01-00451

циенты $B = \{b_i\}$. Рассмотрим алгоритм отбрасывания одного узла x_k и соответствующего ему значения c_k . При этом поток A существенно отличается от потока C лишь одним значением, а вэйвлетные коэффициенты все равны нулю, за исключением коэффициента b_k . Отбрасывание большого количества узлов можно получить последовательным применением приводимого алгоритма.

$$\begin{aligned}
 a_i &= c_i && \text{при } i \leq k-2 \\
 a_{k-1} &= \frac{x_{k+2} - x_{k-1}}{x_{k+1} - x_{k-1}} \cdot c_{k-1} - \frac{x_{k+2} - x_{k+1}}{x_{k+1} - x_{k-1}} \cdot c_{k-2} \\
 a_i &= c_{i+1} && \text{при } i \geq k \\
 b_j &= 0 && \text{при } j \neq k \\
 b_k &= \frac{x_{k+3} - x_{k+1}}{x_{k+1} - x_{k-1}} \cdot \left(\frac{x_{k+2} - x_{k+1}}{x_{k+3} - x_k} \cdot c_{k-2} - \frac{x_{k+2} - x_{k-1}}{x_{k+3} - x_k} \cdot c_{k-1} \right) + c_k - \frac{x_{k+1} - x_k}{x_{k+3} - x_k} \cdot c_{k+1}
 \end{aligned}$$

2. Сжатие модельных числовых потоков

Программная реализация декомпозиции и реконструкции показала быструю и эффективную работу алгоритма на ряде модельных примеров. На языке C++ написана программа, реализующая сжатие и восстановление модельных числовых потоков. Некоторые результаты работы этой программы приводятся в Таблице 1.

Таблица 1. Некоторые численные результаты сжатия

	$N = 100; w = 1050; r < 10^{-8}$			$N = 1000; w = 10050; r < 10^{-8}$		
$u(t)$	Q	r_1	i_0	Q	r_1	i_0
t^2	0.1562	$< 10^{-10}$	13	0.1393	$5 \cdot 10^{-10}$	116
$\frac{1}{1+t^2}$	0.1676	$11 \cdot 10^{-4}$	14	0.1441	$6 \cdot 10^{-6}$	120
$\sin t$	0.1562	$10 \cdot 10^{-4}$	13	0.1369	$12 \cdot 10^{-7}$	114
e^t	0.1562	$14 \cdot 10^{-3}$	13	0.1453	$13 \cdot 10^{-6}$	121

Здесь $u(t)$ - функция, генерирующая входной поток; N - количество узлов исходной сетки; w - объём входного потока (в байтах на диске); r - разность между значениями входного потока и полностью реконструированного потока (согласно теории, реконструкция даёт в точности исходный поток, так что здесь представлена ошибка округления: она весьма мала - менее 10^{-8}). Далее, i_0 - число узлов полученной крупной сетки; если v - объём основного выходного потока (в байтах на диске), то $Q = \frac{v}{w}$ - коэффициент сжатия входного потока; r_1 - максимальное значение разности между погрешностями аппроксимации сплайнами на исходной сетке и на сетке укрупнённой.

3. Распараллеливание вычислений

Рассмотрим распараллеливание вычислений при построении вэйвлетного разложения. Представим порядок вычислений при использовании двух параллельных процессов. Распараллеливание демонстрируется с помощью параллельных форм, в которых ярусы нумеруются последовательно; на каждом ярусе указывается: две очередные вычислительные операции, используемая память (в числе единиц хранимых данных) и число необходимых транзакций (указывается сумма числа пересылок между процессами и числа пересылок при поступлении данных).

Таблица 2. Распараллеливание формул декомпозиции

Ярус	I процесс	II процесс	Пам	Тр
1	$x_{k+2} - x_{k+1}$	$x_{k+3} - x_k$	6	4
2	$x_{k+2} - x_{k-1}$	$x_{k+1} - x_{k-1}$	8	3
3	$\frac{x_{k+2}-x_{k+1}}{x_{k+3}-x_k}$	$\frac{x_{k+2}-x_{k-1}}{x_{k+3}-x_k}$	8	2
4	$u_k = \frac{x_{k+2}-x_{k+1}}{x_{k+3}-x_k} \cdot c_{k-2}$	$v_k = \frac{x_{k+2}-x_{k-1}}{x_{k+3}-x_k} \cdot c_{k-1}$	11	2
5	$u_k - v_k$	$x_{k+3} - x_{k+1}$	9	1
6	$x_{k+1} - x_k$	$\frac{x_{k+3}-x_{k+1}}{x_{k+1}-x_{k-1}}$	7	1
7	$\frac{x_{k+1}-x_k}{x_{k+3}-x_k}$	$w_k = u_k - v_k \cdot \frac{x_{k+3}-x_{k+1}}{x_{k+1}-x_{k-1}}$	5	1
8	$\frac{x_{k+1}-x_k}{x_{k+3}-x_k} \cdot c_{k+1}$	$c_k + w_k$	7	2
9	$\frac{x_{k+2}-x_{k-1}}{x_{k+1}-x_{k-1}}$	$\frac{x_{k+2}-x_{k+1}}{x_{k+1}-x_{k-1}}$	4	2
10	$p_k = \frac{x_{k+2}-x_{k-1}}{x_{k+1}-x_{k-1}} \cdot c_{k-1}$	$q_k = \frac{x_{k+2}-x_{k+1}}{x_{k+1}-x_{k-1}} \cdot c_{k-2}$	3	2
11	$a_{k-1} = p_k - q_k$	$b_k = (c_k + w_k) - \frac{x_{k+1}-x_k}{x_{k+3}-x_k} \cdot c_{k+1}$	2	2

В приведённой параллельной форме ставилась задача минимизировать количество транзакций (в первую очередь при пересылке данных между процессорами) при минимально возможном количестве ярусов вычислений.

Поясним данные, приведённые в таблице 2. Пусть для каждого процессора выделено определённое количество ячеек памяти. Будем считать, что на каждом ярусе (итерации) соблюдается следующий порядок действий: 1) поступление данных на процессор 2) запись в память 3) пересылки с процессора на другой процессор 4) стирание из памяти 5) запись результатов вычислений с процессора в память. Также положим, что если результат вычислений на процессоре нужен только для вычислений на этом же процессоре на следующей итерации, то он может не записываться в память, т.е. будем считать эту информацию "уже поступившей" на процессор.

Для наглядности представим состояние двухпроцессорной системы на каждом n -ном ярусе параллельных вычислений таблицей следующего вида (и проиллюстрируем таким образом данные распараллеливания: см. таблицы 4 – 14):

Рассмотрим второй столбец данной таблицы, иллюстрирующий первый процесс. В клетке, отмеченной цифрой 1 данной таблицы указываются данные, поступающие извне для вычислений на первом процессоре. Это данные, которые не используются далее, поэтому они не записываются в память, а поступают непосредственно на процессор. В клетке 2 указываются данные, поступающие извне для вычислений на первом процессоре. Это данные, которые понадобятся для дальнейших вычислений и записываются в память, отведённую для первого процессора (в скобках указан номер ячейки памяти, в которую производится запись). В клетке 3 указываются данные, пересылаемые со второго процессора (или из памяти, отведённой для второго процессора) на первый (если эти данные записываются в память, то в скобках указывается номер ячейки памяти; если же эти данные не понадобятся для вычислений на следующих итерациях, то в скобках написано "на пр"). В клетке 4 указываются данные, которые более не нужны для дальнейших вычислений и стираются из памяти (в скобках указан номер освобождаемой ячейки памяти). В клетке 5 указывается результат вычислений на первом процессоре на данной итерации. Он может записываться в

Таблица 3. Состояние системы на ярусе n

	I процесс	II процесс
поступление данных:		
(строка 1) сразу на процессор	1	
(строка 2) с записью в память	2	
(строка 3) пересылки	3	
(строка 4) стираемые данные:	4	
(строка 5) данные с процессора:	5	
(строка 6) свободные ячейки:	6	

память (тогда в скобках указывается номер занимаемой ячейки памяти); если же результат вычислений на первом процессоре нужен только для вычислений на этом же процессоре на следующей итерации, то он в память не записывается (тогда в скобках написано "на пр"). В клетке 6 указываются номера ячеек памяти, свободных перед следующей итерацией. Всё сказанное выше аналогично для второго процесса (которому соответствует третий столбец данной таблицы). Для экономии пространства условимся также пропускать в таблицах такого вида пустые строки.

В данном случае на каждом ярусе нам понадобится не более шести ячеек памяти, отведённых для каждого из двух процессоров.

Таблица 4. Состояние системы на ярусе 1

(2) с зап	x_{k+2} (1) x_{k+1} (2)	x_{k+3} (1) x_k (2)
(5) с проц	$x_{k+2} - x_{k+1}$ (3)	$x_{k+3} - x_k$ (3)
(6) своб яч	4 5 6	4 5 6

Таблица 5. Состояние системы на ярусе 2

(1) на пр	x_{k-1}	x_{k-1}
(3) перес		x_{k+1} (4)
(4) стир	x_{k+2} (1)	
(5) с проц	$x_{k+2} - x_{k-1}$ (1)	$x_{k+1} - x_{k-1}$ (5)
(6) своб яч	4 5 6	6

4. Заключение

Предлагается алгоритм, который целесообразно использовать для обработки (разложения на составляющие) больших потоков числовой информации с резко меняющимися харак-

Таблица 6. Состояние системы на ярусе 3

(3) перес	$x_{k+3} - x_k$ (4)	$x_{k+2} - x_{k-1}$ (на пр)
(4) стир		$x_{k+3} - x_k$ (3)
(5) с проц	$\frac{x_{k+2}-x_{k+1}}{x_{k+3}-x_k}$ (на пр)	$\frac{x_{k+2}-x_{k-1}}{x_{k+3}-x_k}$ (на пр)
(6) своб яч	5 6	3 6

Таблица 7. Состояние системы на ярусе 4

(2) с зап	c_{k-2} (5)	c_{k-1} (3)
(5) с проц	$u_k = \frac{x_{k+2}-x_{k+1}}{x_{k+3}-x_k} \cdot c_{k-2}$ (на пр)	$v_k = \frac{x_{k+2}-x_{k-1}}{x_{k+3}-x_k} \cdot c_{k-1}$ (6)
(6) своб яч	6	нет

Таблица 8. Состояние системы на ярусе 5

(3) перес	$v_k = \frac{x_{k+2}-x_{k-1}}{x_{k+3}-x_k} \cdot c_{k-1}$ (на пр)	
(4) стир		x_{k+3} (1); x_{k+1} (4); v_k (6)
(5) с проц	$u_k - v_k$ (6)	$x_{k+3} - x_{k+1}$ (на пр)
(6) своб яч	нет	1 4 6

Таблица 9. Состояние системы на ярусе 6

(3) перес	x_k (на пр)	
(4) стир	x_{k+1} (2)	x_k (2)
(5) с проц	$x_{k+1} - x_k$ (на пр)	$\frac{x_{k+3}-x_{k+1}}{x_{k+1}-x_{k-1}}$ (на пр)
(6) своб яч	2	1 2 4 6

Таблица 10. Состояние системы на ярусе 7

(3) перес		$u_k - v_k$ (на пр)
(4) стир	$x_{k+3} - x_k$ (4); $u_k - v_k$ (6)	
(5) с проц	$\frac{x_{k+1}-x_k}{x_{k+3}-x_k}$ (на пр)	$w_k = u_k - v_k \cdot \frac{x_{k+3}-x_{k+1}}{x_{k+1}-x_{k-1}}$ (на пр)
(6) своб яч	2 4 6	1 2 4 6

теристиками. Алгоритм восстановления позволяет провести восстановление с потерей (т.о. сжав информацию) или (при необходимости) полностью восстановить исходный числовой поток.

Программная реализация декомпозиции и реконструкции показала быструю и эффек-

Таблица 11. Состояние системы на ярусе 8

(1) на пр	c_{k+1}	c_k
(5) с проц	$\frac{x_{k+1}-x_k}{x_{k+3}-x_k} \cdot c_{k+1}$ (2)	$c_k + w_k$ (1)
(6) своб яч	4 6	2 4 6

Таблица 12. Состояние системы на ярусе 9

(3) перес	$x_{k+1} - x_{k-1}$ (на пр)	$x_{k+2} - x_{k+1}$ (на пр)
(4) стир	$x_{k+2} - x_{k-1}$ (1); $x_{k+2} - x_{k+1}$ (3)	$x_{k+1} - x_{k-1}$ (5)
(5) с проц	$\frac{x_{k+2}-x_{k-1}}{x_{k+1}-x_{k-1}}$ (на пр)	$\frac{x_{k+2}-x_{k+1}}{x_{k+1}-x_{k-1}}$ (на пр)
(6) своб яч	1 3 4 6	2 4 5 6

Таблица 13. Состояние системы на ярусе 10

(3) перес	c_{k-1} (на пр)	c_{k-2} (на пр)
(4) стир	c_{k-2} (5)	c_{k-1} (3)
(5) с проц	$p_k = \frac{x_{k+2}-x_{k-1}}{x_{k+1}-x_{k-1}} \cdot c_{k-1}$ (на пр)	$q_k = \frac{x_{k+2}-x_{k+1}}{x_{k+1}-x_{k-1}} \cdot c_{k-2}$ (2)
(6) своб яч	1 3 4 5 6	3 4 5 6

Таблица 14. Состояние системы на ярусе 11

(3) перес	$q_k = \frac{x_{k+2}-x_{k+1}}{x_{k+1}-x_{k-1}} \cdot c_{k-2}$ (на пр)	$\frac{x_{k+1}-x_k}{x_{k+3}-x_k} \cdot c_{k+1}$ (на пр)
(4) стир	$\frac{x_{k+1}-x_k}{x_{k+3}-x_k} \cdot c_{k+1}$ (2)	$c_k + w_k$ (1) q_k (2)
(5) с проц	$a_{k-1} = p_k - q_k$ (1)	$b_k = (c_k + w_k) - \frac{x_{k+1}-x_k}{x_{k+3}-x_k} \cdot c_{k+1}$ (1)
(6) своб яч	2 3 4 5 6	2 3 4 5 6

тивную работу алгоритма на ряде модельных примеров. На языке C++ написана программа, реализующая сжатие и восстановление модельных числовых потоков.

Также рассматривается возможность распараллеливания предлагаемого алгоритма. Предлагается параллельная версия, позволяющая полностью загрузить двухъядерную или двух-процессорную архитектуру и допускающая эффективную программную реализацию.

Литература

1. Петухов А.П. Введение в теорию базисов всплесков. СПб. 1999. 132 с.
2. Добеши И. Десять лекций по вейвлетам. Москва-Ижевск. 2004. 404 с.
3. Чуи К. Введение в вэйвлеты. М.: Мир, 2001. 412 с.
4. Малла С. Вэйвлеты в обработке сигналов. М. 2005. 671 с.

5. Новиков И.Я., Стечкин С.Б. Основы теории всплесков // Успехи математических наук. 1998. Т. 53, № 6(324). С. 53–128.
6. Демьянович Ю. К. Калибровочное соотношение для В-сплайнов на неравномерной сетке // Математическое моделирование. 2001. Т. 13, № 9. С. 98–100.
7. Демьянович Ю. К. Всплесковые разложения в пространствах сплайнов на неравномерной сетке // Докл. РАН. 2002. Т. 382, № 3. С. 313–316.
8. Демьянович Ю. К. Локальный базис всплесков на неравномерной сетке // Записки научных семинаров ПОМИ. 2006. Т. 334, С. 84–110.