МНОГОМЕРНОЕ ГИПЕРКОМПЛЕКСНОЕ ДПФ: ПАРАЛЛЕЛЬНЫЙ ПОДХОД

Алиев М.В.[1], *Чичева М.А.*[2]

[1]*Адыгейский государственный университет,*

[2] *Институт систем обработки изображений РАН*

Аннотация

В работе предложен способ параллельной реализации вычислений с гиперкомплексными числами в многомерном пространстве. В частности, предложен параллельный алгоритм вычисления многомерного дискретного гиперкомплексного преобразования Фурье (ГДПФ).

Введение

В работе [1] было дано определение многомерного ГДПФ вещественного сигнала:

$$F\left(m_1,...,m_d\right)=\sum_{n_1,...,n_d=0}^{N-1} f\left(n_1,...,n_d\right) W^{<\mathbf{m},\mathbf{n}>} ,$$

$$W^{<\mathbf{m},\mathbf{n}>}=\prod_{k=1}^{d} w_k^{m_k n_k} \ , w_k^N=1 , \tag{1}$$

предложены и исследованы алгоритмы параллельного вычисления этого преобразования в двумерном случае. Однако, во-первых, существует интерес к вычислению ГДПФ больших размерностей [2], [3], [4], а, во-вторых, проблема эффективных вычислений с гиперкомплексными числами не ограничивается быстрой реализацией ГДПФ [5, 6, 7, 8].

Таким образом, в статье рассматривается многомерная гиперкомплексная алгебра и способ параллельной реализации вычислений с элементами этой алгебры, основанный на ее структуре. Затем предлагается параллельный алгоритм многомерного ГДПФ, использующий разработанный способ вычислений.

Многомерная гиперкомплексная алгебра

Ниже приводятся некоторые сведения о структуре рассматриваемой многомерной гиперкомплексной алгебры (см., например, [9]).

Определение. Коммутативно-ассоциативной гиперкомплексной алгеброй $\mathbf{B}_d$ будем называть 2^d -мерную $\mathbf{R}$ -алгебру с базисом:

$$\Lambda=\left\{\prod_{i\in I}\varepsilon_i^{\alpha_i}, \quad \alpha_i\in\{0,1\} \ ; \ I=\{1,\ldots,d\} \right\},$$

где $\varepsilon_i^0=1$, $\varepsilon_i^1=\varepsilon_i$. Закон умножения базисных элементов алгебры индуцирован следующим правилом преобразования произведений базисных элементов пространства $\mathbf{V}$:

$$\varepsilon_i\varepsilon_j=\varepsilon_j\varepsilon_i, \quad \varepsilon_i^2=\beta_i, \quad i,j\in I, \ \beta_i=\pm1.$$

Произвольный элемент $g\in\mathbf{B}_d$ записывается в виде:

$$g=\xi_0 E_0+\ldots+\xi_{2^d-1}E_{2^d-1}=\sum_{t\in T}\xi_t E_t , \tag{2}$$

где $E_t=\prod_{i\in I}\varepsilon_i^{\alpha_i}$, $t=\sum_{i\in I}\alpha_i 2^{i-1}\in T=\left\{0,1,\ldots,2^d-1\right\}$.

В этой гиперкомплексной алгебре операция сложения выполняется покомпонентно. Умножение определяется правилами умножения базисных элементов:

$$E_t E_\tau=\Psi(t,\tau)E_{t\oplus\tau}, \quad \forall t,\tau\in T ,$$

где $\oplus$ – поразрядное сложение по модулю 2,

$$\Psi(t,\tau)=\prod_{i\in I}\beta_i^{h_i(t,\tau)} , \ h_i(t,\tau)=\alpha_i\alpha_i' ,$$

$$\tau=\sum_{i\in I}\alpha_i' 2^{i-1} .$$

Доказательство приведенных соотношений может быть найдено, например, в [9]. Ниже мы будем рассматривать 2^d -мерную гиперкомплексную алгебру

$$\mathbf{B}_d\cong\underbrace{\mathbf{C}\dotplus\mathbf{C}\dotplus\ldots\dotplus\mathbf{C}}_{2^{d-1}} ,$$

которая может быть представлена в виде прямой суммы комплексных алгебр.

В этом случае существует, по крайней мере, один элемент α_i, которому соответствует $\beta_i=-1$. Далее, без потери общности, будем считать, что $\beta_1=-1$ и $\beta_i=1$ для остальных значений $i\in I$.

Как показано в [9], такая структура обеспечивает наименьшее количество вещественных операций необходимых для выполнения умножения и сложения элементов алгебры $\mathbf{B}_d$. Кроме того, в этом случае существует возможность синтеза эффективного алгоритма распараллеливания арифметических операций над элементами алгебры.

Параллельные вычисления в алгебре $\mathbf{B}_d$

В работе [1] предложен алгоритм вычислений в четырехмерной гиперкомплексной алгебре $\mathbf{B}_2$ с распараллеливанием операции умножения на 2 ветви.

Пусть произвольный элемент $g\in\mathbf{B}_d$ определяется соотношением (2). Разобьем множество $\{E_t\}_{t\in T}$ на две части: $t\in T'$, если E_t не содержит сомножителя ε_1, и $t\in T''$ в противном случае. При выбранном способе нумерации элементов первое подмножество будет содержать E_t для четных *t*, а второе – для нечетных. Введем замену переменных:

$$\mathbf{U}_0=\mathbf{A}\,\mathbf{E}_0 , \ \mathbf{U}_1=\mathbf{A}\,\mathbf{E}_1 , \tag{3}$$

где

$$\mathbf{E}_0=\{E_t\}_{t\in T'} , \ \mathbf{E}_1=\{E_t\}_{t\in T''} ,$$

$$\mathbf{U}_0=\begin{pmatrix}u_0\\u_1\\\vdots\\u_{2^{d-1}}\end{pmatrix},\ \mathbf{A}=\begin{pmatrix}1&1&1&\cdots&1\\1&1&1&\cdots&-1\\1&1&-1&\cdots&1\\\vdots&\vdots&\vdots&\ddots&\vdots\\1&-1&1&\cdots&-1\end{pmatrix},\qquad(4)$$

Каждая строка и каждый столбец матрицы **A** содержат ровно 2^{d-2} отрицательных чисел из 2^{d-1} значений.

Правило умножения новых базисных элементов записывается в виде:

$$u_j^2=\begin{cases}pu_j, & \text{if } j<p,\\ -pu_{j-2^{d-1}} & \text{if } j\ge p,\end{cases}$$

$$u_ju_k=\begin{cases}pu_k, & \text{if } k=j+p,\\ 0, & \text{otherwise,}\end{cases}$$

где $p=2^{d-1}$. Заметим, что многие произведения равны нулю. Это позволяет нам представить произведение двух произвольных элементов алгебры $\mathbf{B}d$ следующим образом:

$$g\cdot s=\sum_{t\in T}\xi_tE_t\cdot\sum_{\tau\in T}\varsigma_\tau E_\tau=\sum_{t\in T}a_tu_t\cdot\sum_{\tau\in T}b_\tau u_\tau=$$
$$=\sum_{t\in T'}\left(a_tu_t+a_{t+p}u_{t+p}\right)\left(b_tu_t+b_{t+p}u_{t+p}\right),$$

так как остальные слагаемые будут включать произведение $u_ju_k=0$.

Таким образом, произведение двух произвольных элементов алгебры $\mathbf{B}d$ сводится к p независимым произведениям, каждое из которых требует трех вещественных умножений и трех вещественных сложений (по аналогии с умножением комплексных чисел).

В таком представлении вычисления могут быть распараллелены на p независимых ветвей, не требующих обмена данными. Так как замена переменных линейна, сложение остается покомпонентным.

Можно показать, что для произвольного $g\in\mathbf{B}_d$ переход к новому представлению требует 4^{d-1} вещественных сложений. Однако для вещественных и комплексных чисел не требуется выполнения каких-либо нетривиальных арифметических операций. Обратный переход к исходному представлению так же требует 4^{d-1} вещественных сложений.

Пример 1: d=2. См. [1].

Пример 2: d=3. Произвольный элемент восьмимерной гиперкомплексной алгебры $\mathbf{B}_3$ имеет вид:

$$z=\xi_0+\xi_1\varepsilon_1+\xi_2\varepsilon_2+\xi_3\varepsilon_1\varepsilon_2$$
$$+\xi_4\varepsilon_3+\xi_5\varepsilon_1\varepsilon_3+\xi_6\varepsilon_2\varepsilon_3+\xi_7\varepsilon_1\varepsilon_2\varepsilon_3,$$

где $\varepsilon_1^2=-1$, $\varepsilon_2^2=\varepsilon_3^2=1$. Замена переменных выглядит следующим образом:

$$u_0=1+\varepsilon_2+\varepsilon_3+\varepsilon_2\varepsilon_3,$$
$$u_1=1+\varepsilon_2-\varepsilon_3-\varepsilon_2\varepsilon_3,$$
$$u_2=1-\varepsilon_2+\varepsilon_3-\varepsilon_2\varepsilon_3,$$
$$u_3=1-\varepsilon_2-\varepsilon_3+\varepsilon_2\varepsilon_3,$$
$$u_4=\varepsilon_1+\varepsilon_1\varepsilon_2+\varepsilon_1\varepsilon_3+\varepsilon_1\varepsilon_2\varepsilon_3,$$
$$u_5=\varepsilon_1+\varepsilon_1\varepsilon_2-\varepsilon_1\varepsilon_3-\varepsilon_1\varepsilon_2\varepsilon_3,$$
$$u_6=\varepsilon_1-\varepsilon_1\varepsilon_2+\varepsilon_1\varepsilon_3-\varepsilon_1\varepsilon_2\varepsilon_3,$$
$$u_7=\varepsilon_1-\varepsilon_1\varepsilon_2-\varepsilon_1\varepsilon_3+\varepsilon_1\varepsilon_2\varepsilon_3.$$

тогда

$$z=\tfrac{1}{4}\big((\xi_0+\xi_2+\xi_4+\xi_6)u_0+(\xi_0+\xi_2-\xi_4-\xi_6)u_1$$
$$+(\xi_0-\xi_2+\xi_4-\xi_6)u_2+(\xi_0-\xi_2-\xi_4+\xi_6)u_3$$
$$+(\xi_1+\xi_3+\xi_5+\xi_7)u_4+(\xi_1+\xi_3-\xi_5-\xi_7)u_5$$
$$+(\xi_1-\xi_3+\xi_5-\xi_7)u_6+(\xi_1-\xi_3-\xi_5+\xi_7)u_7\big).$$

Правила умножения новых базисных элементов приведены в Таблице 1.

Таблица 1. Правило умножения базисных элементов (d =3)

	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7
u_0	$4u_0$	0	0	0	$4u_4$	0	0	0
u_1	0	$4u_1$	0	0	0	$4u_5$	0	0
u_2	0	0	$4u_2$	0	0	0	$4u_6$	0
u_3	0	0	0	$4u_3$	0	0	0	$4u_7$
u_4	$4u_4$	0	0	0	$-4u_0$	0	0	0
u_5	0	$4u_5$	0	0	0	$-4u_1$	0	0
u_6	0	0	$4u_6$	0	0	0	$-4u_2$	0
u_7	0	0	0	$4u_7$	0	0	0	$-4u_3$

Вместо вычисления произведения

$$\sum_{j=0}^{7}x_ju_j\sum_{k=0}^{7}y_ku_k$$

достаточно независимо вычислить четыре произведения:

$$(x_0u_0+x_4u_4)(y_0u_0+y_4u_4),$$
$$(x_1u_1+x_5u_5)(y_1u_1+y_5u_5),$$
$$(x_2u_2+x_6u_6)(y_2u_2+y_6u_6),$$
$$(x_3u_3+x_7u_7)(y_3u_3+y_7u_7).$$

Главным достоинством предложенного способа распараллеливания вычислений является существенное снижение вычислительной сложности арифметических операций в алгебре $\mathbf{B}_d$ по сравнению с оценками работы [9]:

– сложения гиперкомплексных чисел в 2^{d-1} раз;
– умножения - в $2^{d-1}(2d+1)/3$ раз.

Таким образом, представленный способ позволяет нам распараллелить любой линейный алгоритм обработки многомерных сигналов с небольшим количеством дополнительных операций и высокой эффективностью.

Параллельный алгоритм ГДПФ

Рассмотрим преобразование (1). Известно, что алгоритмы быстрого преобразования Фурье, построенные по аналогии с многомерно схемой Кули-Тьюки, требуют только операций сложения и умножения. В данном случае нам необходимо оперировать с гиперкомплексными числами. Структура быстрого алгоритма ГДПФ [9] такова, что позволяет нам полностью разделить вычисления на 2^{d-1} независимых ветвей с использованием представления (3)-(4). В этом случае алгоритм ГДПФ будет состоять из следующих основных шагов.

Шаг 1. Переход от исходного представления (2) к представлению (3)-(4) (*тривиально*, так как входные данные вещественные, а корни w_k – комплексные).

Шаг 2. Распределение данных по 2^{d-1} процессорам.

Шаг 3. Вычисление преобразования (1) на каждом процессоре с использованием алгоритмов типа Кули-Тьюки [9].

Шаг 4. Реконструкция гиперкомплексного спектра.

Одним из достоинств такого алгоритма (в дополнение к указанным в предыдущей части) является сохранение важнейшего качества последовательного алгоритма - использования свойств симметрии гиперкомплексного спектра вещественного сигнала (см., например, [9]). Однако использование симметрии требует обмена данными между процессорами, что несколько снижает общую эффективность распараллеливания.

Дополнительные возможности для снижения времени вычисления преобразования, по-прежнему, могут быть найдены во внутреннем параллелизме многомерной схемы Кули-Тьюки. На шаге 3 описанного алгоритма может быть выполнено дополнительное распараллеливание каждой ветви на 2^d процессоров. Однако, как показывают исследования, этот способ обладает низкой эффективностью [1], и применять его следует только в том случае, если снижение времени вычисления более актуально, чем эффективность использования вычислительных мощностей.

В работе [1] нами было показано, что в случае d=2 эффективность распараллеливания ГДПФ за счет структуры алгебры составляет около 90%, а эффективность дополнительного распараллеливания в рамках схемы Кули-Тьюки составляет 60-75%. Для сравнения отметим, что эффективность параллельных алгоритмов классического дискретного преобразования Фурье составляет от 20% до 60% (см., например, [10], [11]).

Заключение

В статье рассмотрен метод распараллеливания вычислений в многомерной гиперкомплексной алгебре достаточно общего вида.

Метод основан на особенностях структуры алгебры, а именно, на ее изоморфности прямой сумме комплексных алгебр. Главное достоинство - существенное снижение времени выполнения арифметических операций над элементами алгебры.

Предложенный подход использован для построения высокоэффективного параллельного алгоритма гиперкомплексного дискретного преобразования Фурье. Кроме того, предложенный способ вычислений может быть использован для решения других задач многомерной обработки сигналов.

Благодарности

Работа выполнена при поддержке российско-американской программы «Фундаментальные исследования и высшее образование» (BRHE); и Российского фонда фундаментальных исследований (РФФИ), проекты №№ 03-01-00736, 05-01-96501.

Литература

1. Алиев М.В., Белов А.М., Ершов А.В., Чичева М.А. Алгоритмы двумерного гиперкомплексного дискретного преобразования Фурье // Компьютерная оптика, 2004. № 26. С.101-104
2. Фурман Я.А., Кревецкий А.В., Передреев .К. Введение в контурный анализ; приложения к обработке изображений и сигналов // Под ред. Фурмана Я.А. М.: ФИЗМАТЛИТ, 2002.
3. Geometric Computing with Clifford Algebra // Sommer G. (Ed.). Berlin: Springer-Verlag, Springer Series in Information Sciences, 2001.
4. Vanwormhoudt M.C. Rings of hypercomplex numbers for NT Fourier transforms // Signal Processing, 1998. Vol. 67. P. 189–198.
5. Bülow T., Sommer G Hypercomplex signals - A novel extension of the analytic signal to the multidimensional case // IEEE Transactions on Signal Processing, IEEE Signal Processing Society, 2001. Vol. 49. No. 11. P. 2844-2852.
6. Chaitelin F., Meškauskas T. Computation with hypercomplex numbers // Nonlinear analysis, 2001. No. 47. P. 3391-3400.
7. Labunets E.V., Labunets V.G., Egiazarian K., Astola J. Hypercomplex moments application in invariant image recognition // Int. Conf. On Image Processing 98, 1998. P.256–261.
8. Sommer G. A geometric algebra approach to some problems of robot vision // Computational Noncommunicative Algebra and Applications, Kluwer Academic Publishers, J. Byrnes ed., NATO Science Series, 2004. No. 136. P. 309-338.
9. Алиев М.В. Быстрые алгоритмы d-мерного ДПФ вещественного сигнала в коммутативно-ассоциативных алгебрах 2^d размерности над полем действительных чисел // Компьютерная оптика, 2002. №24. С. 130-136.
10. Gupta A., Kumar V. The scalability of FFT on Parallel Computers // IEEE Transactions on Parallel and distributed systems, IEEE Computer Society, 1993. Vol. 4. No. 8. P. 922-932.
11. Inda M.A., Bisseling R.H. A simple and efficient parallel FFT algorithm using the BSP model // Parallel Computing, 2001. Vol. 27. No. 14. P. 1847-1878.