УДК 004:681.5:681.3

***В.Г. ИВАНОВ***, **канд. техн. наук, доцент, зав. кафедрой информатики и вычислительной техники Национальной юридической академии Украины имени Ярослава Мудрого**

# НЕ СИММЕТРИЧНЫЕ АЛГОРИТМЫ ПОЛУЧЕНИЯ ДАННЫХ В СИСТЕМЕ ХААРА

Показано, что методы сжатия данных (архивация данных) стали практически обязательной составляющей всех современных информационных технологий, начиная от бытовых мультимедийных систем и кончая географически удаленными инфраструктурами хранения и обработки распределенных данных. Предложен ассиметричный алгоритм получения данных в системе Хаара, заключающийся в анализе размеров двоичной записи номера отсчета восстанавливаемой функции и последующим суммированием коэффициентов Хаара с соответствующими знаками, предварительно вычислив промежуточные суммы с учетом постоянства функций Хаара на соответствующих временных интервалах их определения.

Методы сжатия данных (архивация данных) стали практически обязательной составляющей всех современных информационных технологий, начиная от бытовых мультимедийных систем и кончая географически удаленными инфраструктурами хранения и обработки распределенных данных /1,2,3/. Поэтому очень актуальной является задача создания эффективных в вычислительном отношении алгоритмов, позволяющих с высокой скоростью восстанавливать исходные данные (не симметричные алгоритмы). Под симметричностью мы будем понимать временные затраты процессов кодирования и декодирования данных.

Известен метод представления функций в ряд по системе ортогональных функций, например тригонометрических, заключающийся в суммировании коэффициентов ортогонального разложения, умноженных на значения функций базисной системы /4/.

Данный метод при реализации его на ЭВМ требует значительных вычислительных затрат.

Более экономичным, с точки зрения вычислений, является метод, предложенный в /5/ и заключающийся в анализе разрядов двоичной записи номера отсчета восстанавливаемой функции и последующим суммированием коэффициентов ортогонального преобразования Хаара с соответствующими знаками. Таким образом, при вычислении значений функции на ЭВМ, использующих двоичную систему, реализуется соотношение

$$Pn(x) = a_1 + \sum_{m=1}^{m_0} (-1)^{E_m} b_{m,j_m}, \qquad (1)$$

где, x= 0, $E_1$ $E_2$ … Es…, $E_1$ $E_2$ … $E_{m-1}$=$j_m$ – 1, $b_{m,j_m}$ – обобщенные коэффициенты Хаара.

Тогда при каждом m легко выделить цифры $E_1$ $E_2$ ... $E_{m-1}$. Для этого нужны только простейшие логические операции. По значениям m и $j_m$ можно сформировать адрес ячейки, содержащей $b_{m,j_m}$. Если следующая цифра в двоичной записи числа x (т.е. $E_m$) равна нулю, то $b_{m,j_m}$ прибавляется к накапливаемой сумме, а если следующая цифра равна 1, то $b_{m,j_m}$ вычитается:

$$Z_m = Z_{m-1} + (-1)^{E_m} b_{m,j_m}, \qquad (2)$$

где $m = 1, 2 \ldots m_0$.

Начав $Z_0 = a_1$, получим $Z_0 = P_n(x)$. В каждом цикле здесь производится одно сложение и несколько логических операций. Поэтому общее число элементарных операций, затрачиваемых на вычисление $P_n(x)$ равно $O(m_0)$ или $O(\log_2 n)$, где n – число отсчетов функции, а запись $O(\log_2 n)$ означает, что число вычислительных операций в одной точке стремится к значению двоичного логарифма размерности базиса Хаара и для восстановления исходной функции во всех «n» её точках требуется $n\log_2 n$ операций сложения - вычитания.

Предлагаемый ниже метод позволяет сократить общее число вычислительных операций при восстановлении всех «n» значений исходной функции до 2(n-1) и для его осуществления необходимо вычислить промежуточные суммы из одинаковых коэффициентов Хаара, входящих в суммы всего ряда дискретных точек, а в дискретных точках с нечетными номерами значение восстанавливаемой функции получают вычитанием удвоенного значения коэффициента Хаара из значения суммы, полученной в предыдущей дискретной точке с четным номером. Рассмотрим предлагаемый метод синтеза дискретных данных в базисе функций Хаара, например, для восстановления исходной функции в восьми ее точках, т.е. при n = 8.

Запишем сумму в каждой дискретной точке по способу, предложенному в /5/. Например, для точки x=0 или в двоичной системе x=0,000, при m =1 выделим (m-1=0) цифры x и т.к. следующая за этой цифрой стоит цифра ноль (x =0,000), то коэффициент $b_1$, в выражении (2) берут со знаком плюс. При m=2 выделяют (m-1=1) цифры аргумента x, т.е. (x =0,000) и анализируют следующий двоичный разряд и т.д. до $m=m_0$.

Тогда в точке x=0,000 по вышеуказанному правилу ряд $P_n(x)$ запишется как

$$P_0(x) = a_1 + b_{11} + b_{21} + b_{31}. \qquad (3)$$

Аналогично для остальных точек будем иметь:

$P_1(0{,}001) = a_1+b_{11}+b_{21}-b_{32}$ $\quad$ $P_2(0{,}010) = a_1+b_{11}-b_{21}+b_{32}$

$P_3(0{,}011) = a_1+b_{11}-b_{21}-b_{32}$ $\quad$ $P_4(0{,}100) = a_1-b_{11}+b_{22}+b_{33}$

$P_5(0{,}101) = a_1-b_{11}+b_{22}-b_{33}$ $\quad$ $P_6(0{,}110) = a_1-b_{11}-b_{22}+b_{34}$

$P_7(0{,}111) = a_1-b_{11}-b_{22}-b_{34}$ (4)

Для выражения (4) запишем промежуточные суммы

$$a_1+b_{11}=C_1, \qquad a_1-b_{11}=C_2$$

и (4) перепишем как

$$\begin{cases} P_0(x) = C_1+b_{21}+b_{31} \\ P_1(x) = C_1+b_{21}-b_{31} \end{cases} \qquad \begin{cases} P_4(x) = C_2+b_{22}+b_{33} \\ P_5(x) = C_2+b_{22}-b_{33} \end{cases}$$

$$\begin{cases} P_2(x) = C_1-b_{21}+b_{32} \\ P_3(x) = C_1-b_{21}-b_{32} \end{cases} \qquad \begin{cases} P_6(x) = C_2-b_{22}+b_{34} \\ P_7(x) = C_2-b_{22}-b_{34} \end{cases} \qquad (5)$$

Так как значение суммы в точке с нечетным номером отличается от значения суммы, вычисленной в предыдущей четной точке на удвоенное значение коэффициента с индексом максимального значения «m», то выражение (5) окончательно запишется:

$$\begin{array}{ll} P_0(x) = C_1+b_{21}+b_{31} & P_4(x) = C_2+b_{22}+b_{33} \\ P_1(x) = P_0(x) -2b_{31} & P_5(x) = P_4(x) -2b_{33} \\ P_2(x) = C_1-b_{21}+b_{32} & P_6(x) = C_2-b_{22}+b_{34} \\ P_3(x) = P_2(x) -2b_{32} & P_7(x) = P_6(x) -2b_{34} \end{array} \qquad (6)$$

Таким образом, число операций типа сложения-вычитания в выражении (6) будет равняться двенадцати и следует еще добавить две операции при вычислении промежуточных сумм $C_1$ и $C_2$, что дает общее число вычислительных операций равное четырнадцати, т.е. 2(n-1). Умножение на два в дискретных точках с нечетными номерами на ЭВМ с двоичной системой представляет собой простой сдвиг числа в регистре и это время можно не учитывать.

Предложенный метод позволяет существенно в $\left(\frac{n\log n}{2(n-1)}\right)$ раз сократить время восстановления исходной функции в дискретном ряде точек и например, при n=1024 этот выигрыш составит 5 раз.

Высвободившееся машинное время можно использовать для решения других задач инженерного или информационного содержания, что в целом значительно повышает производительность вычислительных систем обработки.

**Список литературы:** 1. Открытые системы. М., 2001, № 2. 2. Матеріали Міжнародної конференції з управління "Автоматика -2001", м. Одеса, Україна: в 2-х т.- Одеса, 2001. 3. Материалы 3 Международной конференции «Цифровая обработка сигналов и ее применение», г. Москва, 2000. 4. *Толстов Г.И.* Ряды Фурье. М., «Наука». 1980. 5. *Соболь И.М.* Многомерные квадратурные формулы и функции Хаара. М., «Наука». 1969.

*Поступила в редколлегию 25.03.02*