УДК 004.046

СПОСОБ КУБИЧЕСКОГО ПРЕОБРАЗОВАНИЯ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ СЖАТИЯ ДАННЫХ

Левенец А. В. – канд. техн. наук, доц. кафедры «Автоматика и системотехника», тел.: 8-914-191-3339, e-mail: levalvi@bk.ru; *Федяев А. Ю.* – ст. преп. кафедры «Автоматика и системотехника», тел.: 8-914-770-2693, e-mail: futgik@gmail.com ; *Чье Ен Ун* – д-р техн. наук, проф. завкафедрой «Автоматика и системотехника», тел.: 8-924-109-0898, e-mail: chye@ais.khstu.ru (ТОГУ)

Рассматривается способ предварительного преобразования данных на базе кубической структуры для упорядочивания их битовой структуры с целью эффективности их сжатия.

The way of preliminary data transformation based on the cubic structure to order their bit structure for making data compression more efficacious is considered.

Ключевые слова: сжатие данных, предварительная обработка данных, подсистема передачи данных.

Одной из важных задач, стоящих перед разработчиками различного рода информационно-измерительных систем, является сокращение объема передаваемых данных, что обусловлено как желанием снизить стоимость системы за счет применения низкоскоростных каналов связи, так и стремлением снизить потребляемую системой мощность. Известно, что для повышения эффективности работы процедур сжатия часто применяют некоторую предварительную обработку данных, приводя их параметры к некоторой оптимальной модели, которая заложена в алгоритме сжатия, т. е. обеспечивает их более коррелируемую структуру. На данный момент существует ряд способов предварительного преобразования, например [1, 2]. Эти способы могут использоваться при разработке новых алгоритмов сжатия, однако проведение исследований в данной области все еще необходимы для улучшения некоторых критических параметров: скорости преобразования, требований к системным ресурсам, структуры выходной последовательности.

Предварительное кубическое преобразование. В общем виде рассматриваемое предварительное преобразование может быть описано с помощью следующей последовательности шагов:

– данные переводятся в удобную для преобразования систему счисления;

– выбирается структура с определенной на ней группой операций, которые позволяют изменять состояние структуры (операции модификации);

– данные отображаются на эту структуру;

– происходит их преобразование с использованием последовательности операций;

– подбирается такая последовательность операций модификации (манипуляция h), при которой свойства преобразованных данных максимально приближены к некоторому оптимальному для сжатия варианту.

Для проведения обратного преобразования необходимо иметь совокупность проделанных преобразований, полученную в результате преобразований последовательность l_t и структуру, с помощью которой было выполнено преобразование. Выполнение манипуляции в обратном порядке с использованием обратных операций даст исходную (несжатую) последовательность l.

Наиболее простой формирующей структурой можно считать куб. В этом случае каждая грань куба разделяется на девять элементов, причем для отображения данных используется только восемь. Таким образом, все шесть граней куба позволяют отображать 48 бит данных, образующих один блок данных (рис. 1).

Для куба может быть определена группа из 18-ти операций модификации. В дальнейшем будут использоваться обозначения таких операций, сформированных из трех символов по следующему принципу. Первый символ означает способ отсчета слоя: U – отсчет слоя сверху, F – отсчет слоя спереди, L – отсчет слоя слева. Второй символ указывает номер слоя (первый, второй или третий). Третий символ указывает направление поворота: «'» – поворот по часовой стрелки, «”» – поворот против часовой стрелки. На рис. 2 приведены примеры обозначений и реализации различных операций модификации.

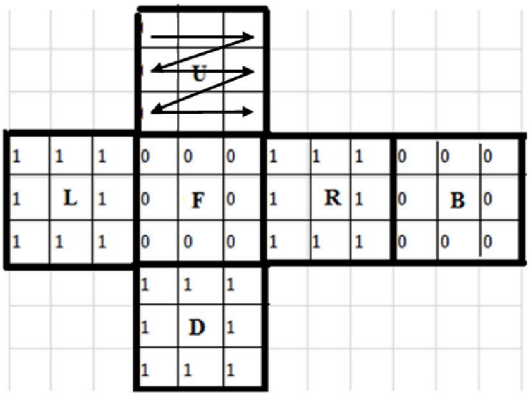

Рис. 1. Развертка куба с обозначением граней и загруженными данными для пяти граней и указанием способа загрузки данных на примере одной грани

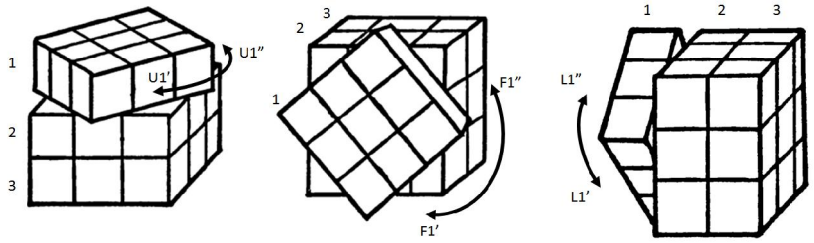

Рис. 2. Обозначение операций модификации

Функция предварительного преобразования для последовательности l определяется следующим образом:

$$f(l,h) = \overset{k}{\underset{i=1}{P}} f_i(l_i,h_i) = l_{\text{t}},$$

где $l = \sum_i^k l_i$; $h = \sum_i^k h_i$; l_i – i-й блок данных, h_i – манипуляция для i-го блока данных; $k = n/6$ – количество блоков данных; $f_i(l_i, h_i) = l_{ti}$ – функция преобразования одного блока данных; $\overset{k}{\underset{i=1}{P}}$ – правило получения l_{t} из совокупности l_{ti}.

Выбор критерия. Критерий оценки оптимальности преобразования существенно зависит от применяемого алгоритма сжатия [2]. Так, для методов RLE необходимо учитывать порядок следования символов, а для статистических методов важным является количество одинаковых символов во всей исходной последовательности данных. На рис. 3 приведены две последовательности *s1* и *s2*, оптимальные для RLE-метода и статистических методов соответственно, которые в дальнейшем будут использоваться как тестовые.

Оценить степень приближения свойств преобразованных данных к теоретически наиболее эффективным с точки зрения процедур сжатия можно с помощью коэффициента подобия K_{p}, вычисляемого следующим образом:

$$K_{\text{p}} = \frac{c_{\text{t}}}{c_{\text{p}}},$$

где c_{t} – максимально возможная (теоретическая) оценка критерия, означает наилучшее сжатие последовательности алгоритмом сжатия; c_{p} – оценка критерия текущей последовательности (практическая оценка).

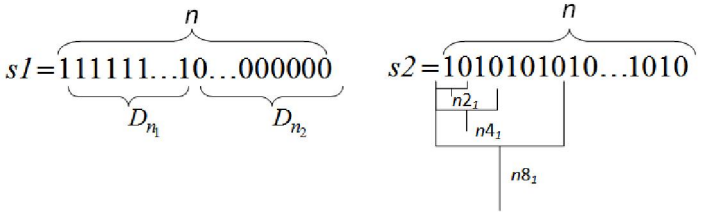

Рис. 3. Тестовые последовательности

Для критерия длины повторов необходимо добиться, чтобы функция преобразования $f(l, h)$ давала последовательность l_{rlt} максимально прибли-

женную к последовательности *s1*. Для получения такой последовательности предлагается следующий способ.

Функция преобразования одного блока $f_i(l_i, h_i) = l_{\text{rlt}i}$ стремится преобразовать каждый блок к виду *s1*. Далее данные в каждой из 48 позиций по каждому блоку считываются в порядке сверху вниз и слева направо (рис. 4). В идеале получается последовательность l_{rlt}, совпадающая по виду с последовательностью *s1*.

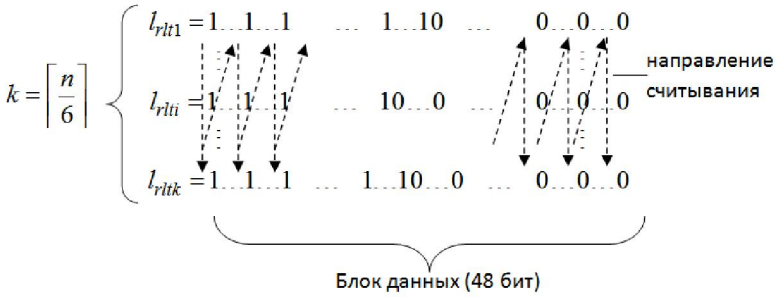

Рис. 4. Принцип формирования последовательности l_{rlt}

Для функции преобразования одного блока $f_i(l_i,h_i)$ критерий c_{rli} вычисляется по следующей формуле:

$$c_{rli} = \sum_{i=0}^{11}(1 - e_{47-i}) \cdot 2^i + \sum_{i=12}^{23} e_{23-i} \cdot 2^i,$$

где e_i – значение бита в *i*-й позиции блока.

Такая формула позволяет учитывать приоритет и позицию битов в блоке.

Для оценки степени похожести результирующей последовательности l_{rlt} к последовательности *s*2 предлагается использовать коэффициент

$$K_{\text{rlp}} = \frac{c_{\text{rlp}}}{c_{\text{rlt}}},$$

где $c_{\text{rlp}} = \sum_{i=1}^{k_c} D_{\text{p}_i}^2$ – оценка по критерию длины повторов; k_c – количество сменяющих друг друга единичных и нулевых блоков; p_i – единичный или нулевой *i*-й блок; $D_{\text{p}i}$ – длина n_i-го блока; $c_{\text{rlt}} = D_{\text{p1}}^2 + D_{\text{p2}}^2$ – теоретическая оценка критерия длины повторов; D_{p1} – длина первого блока; D_{p2} – длина второго блока.

Для другого критерия – блочного – необходимо добиться, чтобы функция преобразования $f(l, h)$ давала последовательность l_{b}, максимально приближенную к последовательности *s2*, содержащую как можно большее число одинаковых подблоков длины два, четыре, восемь бит.

Для оценки близости последовательности l_{b} к последовательности вида *s2* используется коэффициент K_{bp}

$$K_{bp} = \frac{c_{bp}}{c_{bt}},$$

где c_{bp}, c_{bt} – практическая и теоретическая оценки критерия соответственно.

Практическая оценка критерия рассчитается следующим образом:

$$c_{\text{bp}} = \sum_{i=1}^{kb_{\text{n}2}} N_{\text{n}2_i}^2 + \sum_{i=1}^{kb_{\text{n}4}} N_{\text{n}4_i}^2 + \sum_{i=1}^{kb_{\text{n}8}} N_{\text{n}8_i}^2,$$

где $n2_i$, $n4_i$, $n8_i$ – i-я разновидность подблоков длины два, четыре, восемь соответственно; $kb_{\text{n}2}$, $kb_{\text{n}4}$, $kb_{\text{n}8}$ – количество различных подблоков соответствующей длины; $N_{\text{n}2i}$, $N_{\text{n}4i}$, $N_{\text{n}8i}$ – количество появлений i-й разновидности подблоков соответствующей длины.

Теоретическая последовательность рассчитывается исходя из равенства числа нулевых и единичных бит следующим образом:

$$c_{\text{bt}} = \left(\frac{n}{2}\right)^2 + \left(\frac{n}{4}\right)^2 + \left(\frac{n}{8}\right)^2, \quad (1)$$

где $n/2$, $n/4$, $n/8$ – число блоков длины два, четыре, восемь.

Для функции преобразования одного блока $f_i(l_i, h_i)$ в качестве критерия $c_{\text{b}i}$ используется соотношение c_{bp} (1). Эта формула применяется к последовательности, составленной путем конкатенации текущей и всех ранее преобразованных последовательностей.

Алгоритм преобразования текущего блока данных. Для преобразования текущего блока данных $f_i(l_i, h_i)$ предлагается использовать следующий алгоритм:

1. Рассматриваемая последовательность li из шести символов накладывается на куб, полученное состояние рассматривается как исходное s.

2. Для каждой внешней итерации j из числа заданных n_{j} выполняется следующая ниже последовательность действий. При превышении числа заданных итераций следует выход из алгоритма.

3. Каждая операция из установленного набора операций модификации n_{o} применяется к исходному состоянию s. Полученная в результате модификации последовательность оценивается с помощью критерия. Результат, состоящий из манипуляции и оценки по критерию, помещается в буфер выборки $b1$. Вставка происходит таким образом, чтобы сохранялась сортировка буфера в порядке убывания оценок, вычисленных исходя из критерия. Если число итераций превышает заданное значение, происходит переход к шагу 10.

4. Для каждой внутренней итерации k из числа заданных n_{k} выполняется следующая ниже последовательность действий. Если число итераций превышает заданное значение, происходит переход к шагу 8.

5. Для каждой строки r из буфера выборки $b1$ выполняется следующая последовательность.

6. Каждая операция из установленного набора операций модификации no добавляется в конец манипуляции h, хранящейся в строке r, применяется к

исходному состоянию *s*. Результат, состоящий из манипуляции и оценки по критерию, помещается в буфер выборки *b*2. Запись происходит таким образом, чтобы сохранялась сортировка буфера в порядке убывания оценок, вычисленных исходя из критерия.

7. Переход к шагу 4 алгоритма.
8. Буфер выборки *b*2 становится текущим *b*1 (*b*1 = *b*2).
9. Переход к шагу 3 алгоритма.
10. Манипуляция *h*, взятая из первой строки буфера выборки *b*1, подается в поток вывода. *h* применяется к исходному состоянию куба. Полученное состояние куба рассматривается как исходное.
11. Буфер выборки отчищается.
12. Переход к шагу 1 алгоритма.

Так как по ходу заполнения буфера выборки его размер быстро увеличивается, что, в конечном счете, приводит к значительному увеличению времени преобразования, размер буфера ограничивается предельным значением. Уменьшение времени преобразования также достигается путем промежуточной отчистки буфера, записи промежуточного результата и повторения шагов сначала (пункты алгоритма 2, 10, 11, 12).

Для выполнения пунктов алгоритма 1 и 8 необходимо рассмотреть порядок и способ наложения и снятия данных. Учитывая отсутствие привязки алгоритма к определенным типам сжимаемых данных, можно считать, что для совокупности блоков вероятности появления нулей и единиц, в определенной позиции блока, являются равными. Исходя из этого наложение блоков на грани куба может быть произвольным. Следует установить единый способ загрузки данных для всех граней «слева – направо», «сверху – вниз», как это показано на рис. 1. Порядок наложения данных внутри блока будет следующим: верхняя (U), левая (L), центральная (F), правая (R), задняя (B), нижняя (D) грань.

Для установления порядка считывания полученных в результате преобразования данных был проведен ряд исследований. Проводилось несколько преобразований различных типов данных с одинаковыми параметрами алгоритма, со всеми возможными способами считывания результата. В качестве критерия для оценки результата использовался критерий длин повторов, т. к. для блочного критерия порядок считывания данных не имеет значения. На рис 5. представлен пример зависимости порядка считывания данных с граней в зависимости от коэффициента подобия для текстового файла. На рисунке по оси абсцисс условно, в виде индекса, изображен порядок рассмотрения граней, по оси ординат – коэффициент подобия. Вид графика не дает однозначного ответа о порядке считывания данных. Однако для различных типов файлов выявились последовательности считывания, которые показывают результат выше среднего. Одной из таких последовательностей является D, R, T, B, L, F. На рисунке такая последовательность выделена прямоугольником. Эта последовательность выбрана для дальнейшего использования.

Способ считывания данных установлен таким же, как и для загрузки, т. е. для всех граней «слева – направо», «сверху – вниз». Данный способ считывания выбран, так как результаты выполнения преобразований со всеми возможными способами и для различных видов данных не дают однозначных результатов.

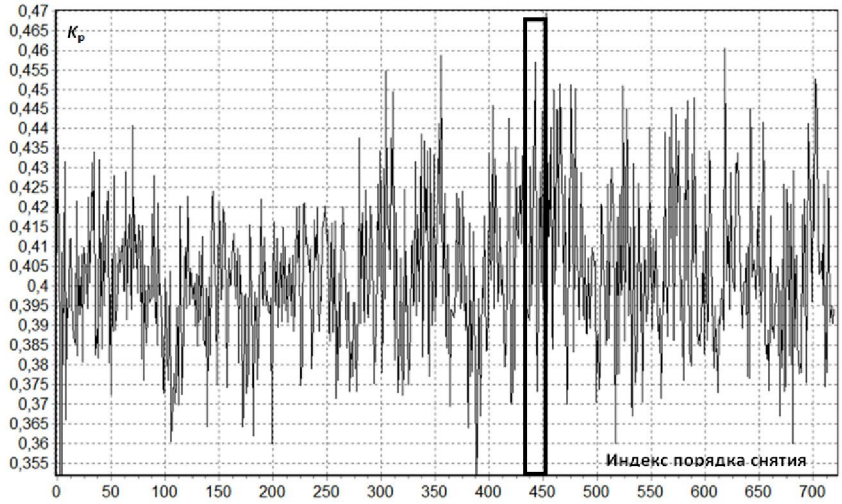

Рис. 5. Пример зависимости коэффициента подобия от порядка снятия значений для данных текстового типа

Определение параметров алгоритма преобразования. Для выполнения описанного выше алгоритма необходимо установить следующие его параметры: число итераций в рамках внешнего цикла (шаг 2 алгоритма) n_j, число итераций внутреннего цикла (шаг 4 алгоритма) n_k, размер буфера выборки n_b, размер списка n_o и список используемых операций модификаций. При выборе данных значений необходимо руководствоваться величиной коэффициента подобия, требованием к размеру манипуляции, временем для осуществления преобразования. Параметры n_j, n_k влияют на все три критерия, в то время как n_b и n_o на размер манипуляции влияния не оказывают. Методика нахождения оптимальных параметров алгоритма состоит в следующем. Для еще неустановленных параметров выбирается произвольное значение. Устанавливается диапазон значений искомого параметра. Для каждого из значений параметра производится преобразование нескольких блоков данных с использованием выбранного критерия. Для результирующей последовательности l_t подсчитывается коэффициент подобия K_p. Для каждого из значений строится график зависимости коэффициента подобия K_p от значения искомого параметра. Выбирается оптимальное значение.

После нахождения оптимального значения для нового параметра производится проверка результатов для ранее установленных параметров, путем повторения опытов с использованием найденного значения.

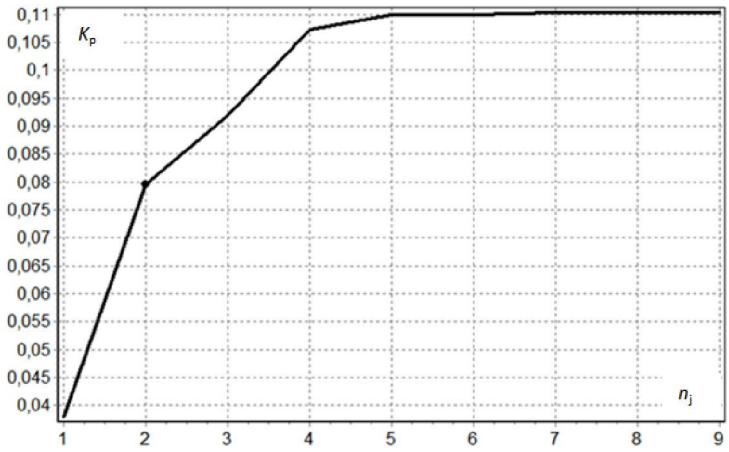

Рис. 6. Зависимость коэффициента подобия K_p от числа итераций внешнего цикла n_j

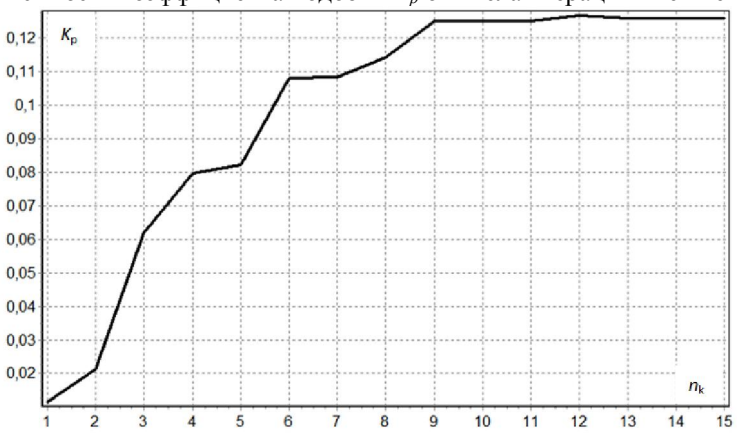

Рис. 7. Зависимость коэффициента подобия K_p от числа операций внутреннего цикла n_k

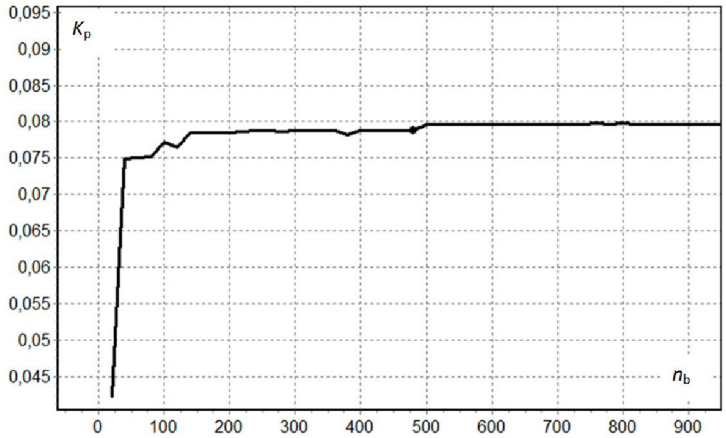

Рис. 8. Зависимость коэффициента подобия K_p от размера буфера выборки n_b

Полученные результаты показывают, что независимо от выбора критерия характер зависимостей не изменяется. Усредненные графики заключительных этапов установки оптимальных параметров для критерия длин повторов для значений n_i, n_k, n_b представлены на рис. 6–8.

Если исходить из того, что в результате работы алгоритма сжатия размер последовательности уменьшиться в два раза, т. е. $n_{tc} = n/2$, то для обеспечения условия сжатия $n_{htc} \leq n_c$ необходимо, чтобы размер n_h манипуляции составлял не более $n/2$. Таким образом, учитывая, что размер блока данных составляет 48 бит, для манипуляции h можно выделить не более чем 24 бита. Для кодирования любой операции модификации необходимо пять бит. Следовательно, модификация может состоять не более чем из четырех операций. С использованием того же числа бит можно сократить число операций в силу увеличения числа манипуляций для лучшего упорядочивания структуры. В качестве примеров таких сочетаний можно привести следующие: использование 16 операций, при этом для их кодирования используется четыре бита, модификация может включать шесть операций; использование восьми операций, при этом для их кодирования используется три бита, модификация может включать восемь операций.

Основываясь на полученных данных, можно считать, что оптимальным является использование восьми операций. При этом число внешних итераций выбирается равным двум, число внутренних итераций равным четырём, а размер буфера выборки ограничивается 500-ми значениями.

Выбор собственно операций так же оказывает влияние на эффективность преобразования. Для выбора восьми операций из общего числа определенных предлагается использовать следующее правило: из списка не следует выбирать операции <U2’, L2’ ,F2’, U2”, L2”, F2”> т. к. любая из них может быть воспроизведена при помощи оставшихся. Из оставшихся 12-ти операций итоговый список должен содержать все отсчеты U1, U3, L1, L3, F1, F3 как минимум по одному из направлений поворота. Исходя из перечисленных требований для проведения дальнейших исследований был выбран следующий набор операций: <U1’, U3’, L1’, F1’, U1”, L3”, F1”, F3”>.

Исследование алгоритма преобразования. Исследование алгоритма предварительного преобразования проводилось на данных различного типа, в том числе использовались текстовый файл формата txt, файл формата xml, файл исходного кода на языке C, файлы формата gif и jpg, а также архив 7zip.

Использовался описанный выше алгоритм со следующими параметрами. Способ занесения данных U, L, F, R, B, D; способ считывания данных D, R, T, B, L, F; порядок занесения, считывания данных «слева – направо», «сверху – вниз». Число итераций внешнего цикла $n_j = 2$, число операций внутреннего цикла $n_k = 4$, объем выборки $n_b = 500$, используемые для преобразования операции <U1’, U3’, L1’, F1’, U1”, L3”, F1”, F3”>. Оценка качества преобразования данных проводилась с использованием двух критериев: критерия длины повторов и блочного критерия. Для каждого из преобразован-

ных типов файлов помимо оценки критерия, вычислялась оценка энтропии. Результаты проведенных исследований приведены в таблице.

Результаты предварительных преобразований различных типов данных

	$N_{исх.}$	$N_{служ.}$	$K_{rlp\ исх.}$	$K_{bp\ исх.}$	$K_{rlp\ рез.}$	$K_{bp\ рез.}$	$H_{исх.}$	H_{RL}	H_B
7zip архив	12828	6414	$3{,}0\cdot10^{-4}$	0,20	0,03	0,36	8,0	7,0	6,3
gif файл	11646	5823	$3{,}0\cdot10^{-3}$	0,29	0,33	0,48	7,8	6,8	6,2
jpg файл	13062	6531	$3{,}0\cdot10^{-4}$	0,22	0,02	0,33	7,9	7,0	6,2
txt файл	16992	8496	$5{,}0\cdot10^{-4}$	0,56	0,06	0,79	4.5	6,9	6,1
xml файл	10794	5397	$6{,}0\cdot10^{-5}$	0,41	0,02	0,67	5,0	6,9	6,1
C файл	11784	5892	$1{,}09\cdot10^{-3}$	0,21	0,05	0,32	5,3	6,9	6,1

Примечание: $N_{исх.}$ – размер исходной последовательности в байтах; $N_{служ.}$ – размер служебной информации в байтах; $K_{rlp\ исх.}$, $K_{bp\ исх.}$ – коэффициенты подобия $K_{rlp,}$ K_b для исходных последовательностей; $K_{rlp\ рез.}$, $K_{bp\ рез.}$ – коэффициенты подобия $K_{rlp,}$ K_b для преобразованных последовательностей; $H_{исх.}$, H_{RL}, H_B – энтропии посчитанные для исходной, преобразованной с использованием критерия длины повторов и блочного критерия последовательностей

Анализ результатов показывает, что применение предложенного преобразования к данным, содержащим текстовую информацию, не приводит к снижению энтропии, несмотря на существенное повышение значений предложенных критериев. Этот факт объясняется тем, что, несмотря на увеличение значимости конкретного блока в результате преобразования, увеличивается количество самих блоков.

Следует отметить также существенное снижение энтропии для бинарных данных, что позволяет говорить о потенциальных возможностях предлагаемого метода предварительной обработки для повышения степени сжатия данных такого типа, причем, учитывая случайный характер таких данных, можно также прогнозировать хорошие перспективы предлагаемого преобразования для сжатия измерительных данных.

Библиографические ссылки

1. *Левенец А. В., Нильга В. В.* Структурное упорядочение данных для задач сжатия в информационно-измерительных системах // Вестник Тихоокеанского государственного университета. 2009. № 2(13).

2. *Федяев А. Ю.* Способ предварительной обработки измерительных сигналов для задач сжатия // Молодежь и современные информационные технологии: сб. трудов VI Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. Томск, 2008.

3. *Методы сжатия данных.* Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. М., 2003.