УДК 004.627

Н.В. Кожемякина, Н.Н. Пономаренко, А.А. Зеленский

Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ», Харьков

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ СЖАТИЯ ДАННЫХ ПРИ КОДИРОВАНИИ СИМВОЛОВ БОЛЬШИХ АЛФАВИТОВ

Проведены исследования сжатия информации в системах передачи данных. Представлен метод формирования тестовых данных, имеющих большой размер алфавита, что соответствует мультимедийной информации, передаваемой в современных телекоммуникационных сетях. Сформированы тестовые наборы данных с разными размерами алфавита, для которых точно можно рассчитать достижимую степень сжатия. Проведен анализ работы современных архиваторов. Предприняты меры по усовершенствованию формирования тестовых выборок, чтобы исследуемая эффективность сжатия архиватора зависела только от его способности работы с символами больших алфавитов. Показано, что используемые архиваторы недостаточно эффективно сжимают данные с большим размером алфавиты. Для всех архиваторов свойственно ухудшение показателей эффективности при увеличении размера алфавита.

Ключевые слова: *сжатие данных, тестовые наборы для сжатия, сжатие без потерь, большой алфавит, аналитические модели.*

Введение

Вследствие того, что объемы передаваемых по каналам связи данных растут быстрее, чем растет пропускная способность этих каналов, все более актуальной становится задача разработки эффективных методов сжатия трафика [1]. При этом основной объем в передаваемых данных занимает мультимедийная информация: изображения и видео. Типичными данными, которые можно при этом дополнительно сжать при передаче по каналам связи являются квантованные коэффициенты дискретного косинусного преобразования (ДКП) в блоках изображений или кадров видео. Эти блоки могут иметь размеры 8x8 пикселей в JPEG [2] или 16x16 пикселей в стандартах сжатия видео [3]. Многие современные методы сжатия изображений оперируют с блоками большего размера, например, 32x32 пикселя [4] или с блоками, для которых размер выбирается адаптивно (до 256x256 пикселей) [5].

Квантованные коэффициенты ДКП в каждом блоке изображения фактически в совокупности являются символом сверхбольшого алфавита (например, для блоков 8x8 пикселей один символ алфавита занимает 64 байта), так как между отдельными коэффициентами блока присутствует корреляция по амплитуде. Это наличие корреляции используется при разработке различных эвристических процедур сжатия квантованных коэффициентов ДКП, от относительно простых, как в стандарте JPEG, до сложных, как в [4]. Однако актуальной остается разработка универсальных методов, которые бы сжимали тексты со сверхбольшими алфавитами при априори неизвестной величине алфавита и функции плотности вероятности распределения символов в нем.

Различные методы сжатия данных без потерь информации сильно отличаются по эффективности сжатия текстов с большими размерами алфавитов. В частности, словарные методы сжатия, к которым относится, например, архиватор WinZip [6], работают в основном с восьмибитными алфавитами. Методы, использующие PPM кодирование, к которым относится, например, архиватор RK [7], способны учитывать зависимости между несколькими соседними символами текста, эффективно работая с 5-6 байтными алфавитами. Методы, основанные на преобразовании Бэрроуза-Уилера, к которым относится архиватор bbb [8], для текстов с большой избыточностью также способны учитывать символы достаточно больших алфавитов. И, наконец, предложенный нами метод рекурсивного группового кодирования (РГК) [9] был разработан специально для кодирования символов больших алфавитов.

Целью данной статьи является разработка метода формирования тестовых последовательностей данных с большим размером алфавита, для которых точно можно определить достижимую степень сжатия.

Чтобы сравнить эффективность сжатия разными методами текстов со сверхбольшими алфавитами, достаточно взять несколько изображений, разбить их на блоки и получить квантованные коэффициенты ДКП в этих блоках. При этом можно узнать, какой из этих архиваторов эффективнее сжимает такие данные. Однако останется неизвестно, был ли этим архиватором достигнут теоретически возможный предел сжатия или нет? Потому что неясно, каким образом вычислить величину этого предела для заданного текста.

В данной работе предлагается метод синтезирования тестовых выборок данных, который позво-

ляет генерировать текст с алфавитом заданной длины. При этом сгенерированная выборка будет обладать статистическими характеристиками, близкими к выборкам квантованных коэффициентов ДКП блоков изображений. Кроме того, предлагаемый метод позволяет вычислить теоретически достижимый (в соответствии с теоремой Шеннона [10]) предел сжатия этой выборки.

В подразделе 1 данной работы описывается предлагаемый метод формирования тестовых выборок. В подразделе 2 сформированные тестовые выборки используются для оценивания эффективности сжатия текстов сверхбольших алфавитов известными архиваторами.

1. Метод формирования тестовых выборок

Пусть N - длина алфавита в байтах, K - число символов этого алфавита в тексте, а rand(X) - равномерно распределенное целое случайное число в диапазоне от 0 до X-1. Тогда предлагаемый метод формирования тестовой выборки будет состоять из выполнения следующих шагов:

1. Пусть D - переменная, которая будет содержать количество бит, которое в соответствии с теоремой Шеннона достаточно для хранения сформированной тестовой выборки. Значение D инициализируется нулем.

2. Для каждого из K символов выборки вычисляются N его байт $\{Z_1, Z_2, \ldots, Z_N\}$ в соответствии со следующим выражением:

$$Z_0 = 256,\ R = \left\lfloor \sum_{m=0}^{i-1} Z_m \,/\, i \,/\, Q \right\rfloor,$$
$$Z_i = \mathrm{rand}(R),\ D = D + \log_2(R), \qquad (1)$$

где i = 1..N; Q - поправочный коэффициент, больше или равный единицы.

Значение Z_0 не сохраняется в выборке и лишь используется в процессе вычисления. Если R=0, то в соответствующее Z_i заносится ноль, а D не увеличивается.

3. Формируется массив A, заполненный последовательностью целых чисел от 0 до 255. Эти числа перемешиваются в массиве A случайным образом.

4. Представим полученную на шаге 2 выборку в виде массива из L=KxN байт (K символов по N байт) и обозначим его как B. Для каждого байта массива B выполняется табличная подстановка:

$$B_j = A(B_j), \qquad (2)$$

где j=1..L.

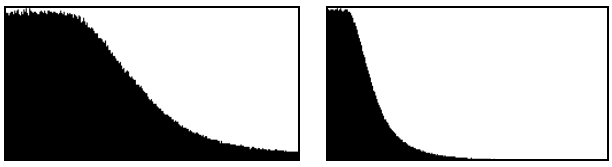
Здесь из ячейки массива A с номером B_j извлекается число и заносится в массив B вместо Bj.

Поясним подробнее шаги 2-4 метода.

Выражение (1) гарантирует, что R для каждого последующего байта многобайтного символа будет не больше, чем для предыдущего байта, и, таким образом, будет постепенно уменьшаться. Это соответствует свойствам квантованных коэффициентов ДКП блоков изображений. Низкочастотные коэффициенты (расположенные в начале выборки), как правило, больше по амплитуде, чем высокочастотные (расположенные в конце выборки). В то же время суммирование всех предыдущих байт при вычислении R обеспечивает то, значение каждого следующего байта в какой-то степени зависит от всех предыдущих байт, и, таким образом, случайные числа $Z_1..Z_N$ будут образовывать отдельный символ алфавита, а не N независимых случайных величин. В то же время соседние символы алфавита друг от друга никак не зависят.

Поправочный коэффициент Q позволяет регулировать степень сжатия тестовой выборки. При Q=1 полученная выборка будет обладать наименьшей статистической избыточностью и, следовательно, будет сжиматься хуже всего. Увеличение Q приведет к более быстрому уменьшению R для последующих байт символа и, следовательно, к возрастанию числа нулевых байтов в символе. Выборка при этом будет сжиматься тем лучше, чем выше значение Q.

На рис. 1, a приведена гистограмма распределения значений байт в сформированной на шаге 2 выборке при N = 8 и Q = 1, а на рис 1, b – при N = 64 и Q = 1.



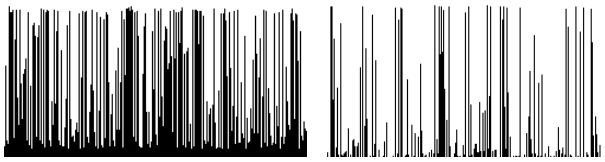
a b

Рис. 1. Гистограмма распределения значений байт в сформированной выборке после второго шага: а – для N = 8, б – для N = 64

Как видно из рис. 1, увеличение N также приводит к увеличению количества нулей в формируемой выборке. Таким образом при фиксированном Q тестовая выборка с большей длиной алфавита будет сжиматься лучше, чем выборка с меньшей длиной алфавита. Выборка с N=1 сжиматься вообще не будет, так как представляет собой равномерно распределенные случайные числа в диапазоне 0..255.

Шаги 3 и 4 выполняются для того, чтобы усложнить задачу архиваторов, сделав бесполезными часть применяемых в них эвристик. Выполняемая при этом процедура замены одних чисел на другие не приводит к изменению избыточности текста. Однако плотность распределения вероятности значений отдельных байт в тексте уже не будет гладкой

функцией, как на рис. 1. Гистограммы распределения значений байт тех же выборок, что и на рис. 1, но после выполнения шага 4, приведены на рис. 2.



a b

Рис. 2. Гистограммы распределения значений байт в сформированной выборке после четвертого шага: а – для N = 8, б – для N = 64

Теперь эффективность сжатия тестовой выборки тем или иным методом сжатия будет зависеть только от способности этого метода работать с символами больших алфавитов.

2. Анализ эффективности архиваторов

В работе были сформированы восемнадцать тестовых выборок с размерами алфавитов в 2, 4, 8, 16, 32 и 64 байта. Архив с соответствующими файлами можно скачать по ссылке:
http://ponomarenko.info/picst2014.rar.

В табл. 1 приведены теоретически достижимые при сжатии размеры этих тестовых файлов в байтах (полученные делением соответствующего числа D на 8). Как видно из данных таблицы, с увеличением размера алфавита, избыточность тестовых выборок возрастает.

Для анализа были выбраны следующие восемь архиваторов: 7-zip FM [11] (уровень сжатия: ультра, метод сжатия:LZMA2), WinRar [12] (compression method: best), WinAce [13], WinZip [6] (compression level: maximum), PAQ8px_v29 [14] (с ключом “-8”), RK [7] (с ключом “-mx”), bbb [8] и РГК [9].

В табл. 2 – 4 приведены результаты сжатия тестовых последовательностей с разными значениями поправочного коэффициента Q для каждого набора вышеперечисленными архиваторами.

Как видно из данных табл. 1 - 4, лучшие результаты показывает РГК и архиватор PAQ8px (очень медленный). Неплохие результаты показывают также RK и bbb, в то время как остальные архиваторы отстают более существенно.

Покажем в процентах, насколько размеры файлов, приведенные в табл. 2 превышают теоретически достижимые.

Чтобы не перегружать информацией рисунок (рис. 3), приведем графики только для РГК, RK и WinRar.

Как видно из приведенных графиков, с увеличением размера алфавита, избыточность кода всех архиваторов увеличивается. Причем, если для небольших размеров алфавита она составляет приемлемые 1-3%, то для 64-байтного размера алфавита избыточность достигает 8-15%.

Таблица 1

Теоретически достижимые при сжатии размеры тестовых файлов

Достижимый размер файла	Размер алфавита в байтах					
	2	4	8	16	32	64
Q=1	1019427	980343	932556	877989	818891	756039
Q=3	788972	721252	639426	546841	446543	340179
Q=9	570782	492813	397676	289647	168955	84453

Таблица 2

Результаты сжатия тестовых последовательностей исследуемыми архиваторами (Q = 1)

Архиватор	Размер алфавита					
	2	4	8	16	32	64
7-zip FM	1048756	1037988	1005776	962645	911736	854365
WinRar	1048648	1034111	999488	961587	915568	865798
WinAce	1048672	1039840	1007448	971373	929405	884209
WinZip	1048812	1032356	996785	957387	903611	854035
PAQ8px	1030342	1014923	980746	931526	871849	814162
RK	1044671	1026916	993270	946614	890133	825080
bbb	1045587	1025877	992023	945980	891038	828726
РГК	1032885	1016567	972103	927880	869245	814050

Таблица 3

Результаты сжатия тестовых последовательностей исследуемыми архиваторами (Q = 3)

Архиватор	Размер алфавита					
	2	4	8	16	32	64
7-zip FM	843126	800096	736488	660657	564360	444705
WinRar	843222	815192	771122	694165	600516	486186
WinAce	857934	835206	792214	706751	603575	488711
WinZip	829797	793756	730112	692734	597930	470679
PAQ8px	793410	736911	660519	569468	469057	361248
RK	843833	800674	739003	657215	564434	459393
bbb	819061	776263	709046	618848	511966	394197
РГК	797635	747460	677672	608841	515283	407527

Таблица 4

Результаты сжатия тестовых последовательностей исследуемыми архиваторами (Q = 9)

Архиватор	Размер алфавита					
	2	4	8	16	32	64
7-zip FM	627027	551144	462779	355798	235798	127142
WinRar	657541	591753	503685	397419	272991	144834
WinAce	678398	614158	526458	419891	280739	148323
WinZip	655517	608929	519550	410041	262389	136411
PAQ8px	573499	498279	405749	298113	177262	90073
RK	629558	578828	509691	410011	275547	142565
bbb	595720	532531	447809	334170	198245	100731
РГК	575207	498088	424364	335919	210970	105802

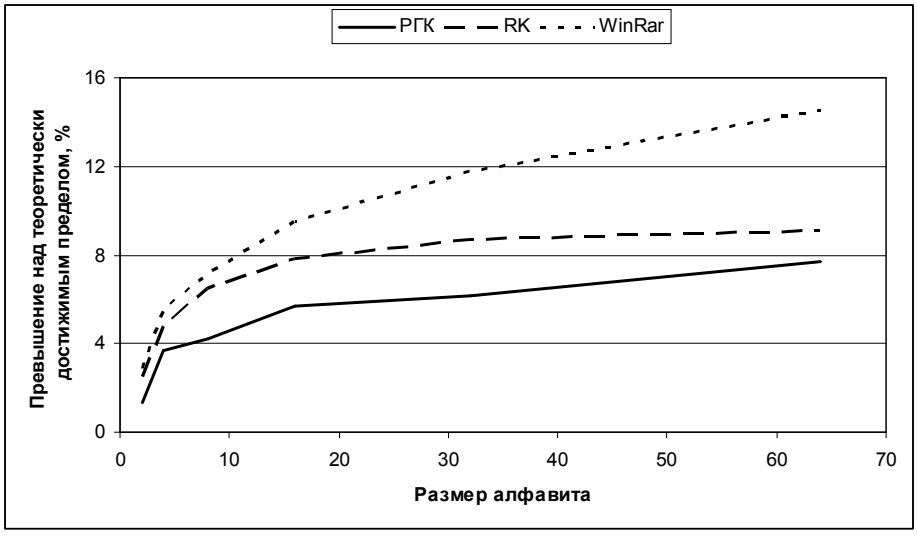


Рис. 3. Зависимости избыточности кода от размера алфавита для РГК, RK и WinRar

Это свидетельствует в пользу перспективности дальнейших исследований в сжатии текстов с алфавитами больших размеров.

Разработка новых более эффективных методов сжатия в этой области способна принести существенный эффект.

Заключение

В работе был предложен метод, который позволяет формировать тестовые последовательности данных с алфавитами больших размеров. При этом тестовые выборки формируются, соблюдая условие, что каждый следующий элемент зависит от предыдущих. Метод позволяет формировать тестовые последовательности, для которых можно точно вычислить теоретически достижимую степень сжатия. Было проведено исследование эффективности сжатия последовательностей с большим алфавитом рядом существующих архиваторов. Показано, что для всех архиваторов характерно увеличение избыточности с ростом размера алфавита: от 1-3% для 2-байтного и до 8-15% для 64-байтного алфавита. Наиболее эффективными для сжатия символов больших алфавитов оказались метод РГК и архиватор PAQ8px. Однако по скорости кодирования РГК значительно превосходит PAQ8px.

Список литературы

1. Salomon, D. Data Compression: The Complete Reference [Text] / D. Salomon. – Springer. : Springer, 2004. – 419 p.

2. Wallace, G. The JPEG Still Picture Compression Standard [Text] / G. Wallance // Communications of the ACM. – April, 1991. – Vol. 34. – P. 30–44.

3. Richardson, Iain E. G. H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia [Text] / Iain E. G. Richardson. – Chichester. – : John Wiley & Sons Ltd, 2003. – 306 p.

4. DCT based high quality image compression [Text] / K. Egiazarian, J. Astola, V. Lukin, N. Ponomarenko // Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA'2005). – June, 2005. – Vol. 3540. – P. 1177–1185.

5. Ponomarenko, N. ADCTC: A new high quality DCT based coder for lossy image compression [Electronic resource] / N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola. – 80 Min / 700 MB. CD ROM Proceedings of LNLA. – Switzerland, August – 2008, 6 p. – 1 electronic optical disc (CD-ROM).

6. The WinZip committee home page [Electronic resource]: Data compression programs, website. – Access mode: http://www.winzip.com. – Access date 05.06.2015. – Title by screen.

7. The WinRK committee home page [Electronic resource]: Data compression programs, website. – Access mode: http://www.msoftware.co.nz/products/winrk. – Access date 05.06.2015. – Title by screen.

8. Data compression programs. BBB [Electronic resource]: Data compression programs, website. – Access mode: http://www.mattmahoney.net/dc/. – Access date 05.06.2015. – Title by screen.

9. Fast recursive coding based on grouping of Symbols [Text] / N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola // Telecommunications and Radio Engineering. – 2009. – Vol. 68, N 20. – P. 1857–1863.

10. Shannon, C A Mathmetical Theory of Communication [Text] / C. Shannon // Bell System Technical Journal. – 1948. – Vol. 27, № 3. – P. 379–423.

11. 7-Zip format [Electronic resource]: Data compression programs, website. – Access mode: http://www.7-zip.org – Access date 05.06.2015. – Title by screen.

12. The WinRar committee home page [Electronic resource]: Data compression programs, website. – Access mode: http://www.rarlab.com – Access date 05.06.2015. – Title by screen.

13. The WinAce committee home page [Electronic resource]: Data compression programs, website. – Access mode: http://www.winace.com. – Access date 05.06.2015. – Title by screen.

14. The PAQ Data compression Programs [Electronic resource]: Data compression programs, website. – Access mode: http://www.mattmahoney.net/dc/paq.html – Access date 05.06.2015. – Title by screen.

Поступила в редколлегию 15.06.2015

Рецензент: д-р техн. наук, проф. В. В. Лукин, Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Харьков.

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЕФЕКТИВНОСТІ МЕТОДІВ СТИСНЕННЯ ДАНИХ ПРИ КОДУВАННІ СИМВОЛІВ ВЕЛИКИХ АЛФАВІТІВ

Н.В. Кожемякіна, М.М. Пономаренко, А.А. Зеленський

Проведено дослідження стиснення інформації в системах передачі даних. Наведено метод формування тестових даних, що мають великий розмір алфавіту, що відповідає мультимедійній інформації, що передається в сучасних телекомунікаційних мережах. Сформовані тестові набори даних з різними розмірами алфавіту, для яких точно можна розрахувати досяжну ступінь стиснення. Проведено аналіз роботи сучасних архіваторів. Вжито заходи щодо вдосконалення формування тестових вибірок, щоб досліджувана ефективність стиснення архіватора залежала тільки від його здатності роботи з символами великих алфавітів. Показано, що архіватори, які зараз використовуються, недостатньо ефективно стискають дані з великим розміром алфавіту. Для всіх архіваторів властиво погіршення показників ефективності при збільшенні розміру алфавіту.

Ключові слова: *стиснення даних, тестові набори для стиснення, стиснення без втрат, великий алфавіт, аналітичні моделі.*

COMPARATIVE ANALYSIS OF DATA COMPRESSION METHODS FOR ENCODING OF SYMBOLS OF LARGE ALPHABETS

N.V. Kozhemiakina, N.N. Ponomarenko, A.A. Zelensky

Several studies have been performed to analyze data compression in telecommunication systems. A method of test data forming (generation) which are large alphabet which corresponds to the transmitted multimedia information in modern telecommunication networks is proposed. Test sequences with different sizes of alphabet with can accurately calculating theoretically achievable compression ratio was generated. The compressed ratio of new modern archivers was analyzed. A method of forming test samples has been improved order to studied efficiency archiver compression ratio depends only on its ability to work with a data with large alphabets. It is shown for all archivers by increasing the size of the alphabet the its performance is decreased.

Keywords*: data compression, test data compression, analytical models, lossless compression, large alphabets.*