

УДК 681.004.6

АНАЛИТИЧЕСКИЙ ОБЗОР АЛГОРИТМОВ СЖАТИЯ ЦИФРОВОЙ ИНФОРМАЦИИ

В.В. Кириченко

Объединенный институт проблем информатики НАН Беларуси, Минск

ANALYTICAL REVIEW OF COMPRESSION ALGORITHMS OF DIGITAL INFORMATION

V.V. Kirichenko

United Institute of Informatics Problems of the National Academy of Sciences, Minsk

Приведен краткий обзор и анализ существующих алгоритмов сжатия цифровой информации. Рассматриваются преимущества и недостатки алгоритмов сжатия данных в зависимости от вида цифровой информации. Указываются программные продукты, реализованные на основе рассматриваемых алгоритмов.

Ключевые слова: информация, сжатие, сжатие данных, сжатие информации, сжатие изображений, алгоритм, алгоритм сжатия, архиватор, контекстное моделирование, кодирование информации.

A brief review and analysis of existing algorithms for compression of digital information are presented. The advantages and disadvantages of data compression algorithms depending on the type of digital information are considered. The software implemented on the basis of the considered algorithms is specified.

Keywords: information, compression, data compression, image compression, algorithm, compression algorithm, archive program, context modeling, encoding of information.

Введение

Развитие электроники, компьютерных и информационных технологий в наши дни идет высоким темпом. Создается инновационное оборудование, позволяющее проводить очень точные исследования в различных областях науки и медицины, которое оперирует огромным объемом данных. К примеру, архив снимков телескопа «Хаббл», накопленный за 15 лет, составляет примерно 50 терабайт, архив диагностических исследований в Минском консультационно-диагностическом центре, накопленный за 5 лет, составляет примерно 3,5 терабайт, а средний объем одного исследования компьютерной томографии в учреждениях здравоохранения Республики Беларусь составляет примерно 500 мегабайт. В медицине важную роль играют архивы исследований, которые позволяют проследить тенденцию развития заболевания конкретного пациента, изучить похожие случаи и т. п. Все это приводит к тому, что объемы данных, которые необходимо хранить и обрабатывать, растут в геометрической прогрессии.

Таким образом, проблема хранения и обработки больших объемов данных очень актуальна на сегодняшний день.

В данной статье рассматриваются универсальные и специфические алгоритмы сжатия данных, которые могут быть применены для сжатия любого вида информации. Рассматриваются преимущества и недостатки тех или иных алгоритмов сжатия в зависимости от типа данных.

1 Универсальные алгоритмы сжатия

1.1. Начало развития теории сжатия данных. Теория сжатия данных стала развиваться, как только ученые стали задумываться о количественной составляющей информации. Первые шаги в этом направлении сделал Ральф Вinton Лайон Хартли в 1928 году. Полученный им результат [1] можно сформулировать следующим образом: если во множестве, состоящем из N элементов, выделен некоторый элемент x , принадлежащий этому множеству, то для того, чтобы найти x , необходимо получить количество информации по формуле: $I = \log_2 N$. Эту формулу обычно называют формулой Хартли.

Однако основной вклад в изучение вопросов сжатия данных внес основатель теории информации Клод Шеннон. Он стал изучать вопрос взаимосвязи между семантикой и синтаксисом, т.е. каким образом можно уменьшить длину информации, сохраняя при этом ее значение. Исследования Шеннона в этом направлении привели к созданию теории кодирования информации [2]. В основе теории кодирования лежит следующая идея: если представить часто встречающиеся элементы короткими кодами, а редко встречающиеся коды длинными кодами, то для хранения такого блока данных требуется меньший объем памяти, чем если бы все элементы представить кодами одинаковой длины. Точная связь между вероятностями и кодами установлена в теореме Шеннон о кодировании источника: элемент s_i вероятность появления которого равняется

$p(s_i)$, выгоднее всего представлять $-\log_2 p(s_i)$ битами. Если при кодировании размер кода всегда в точности получается равным $-\log_2 p(s_i)$ битам, то в этом случае длина закодированной последовательности будет минимальна для всех возможных способов кодирования. Если распределение вероятностей $F = \{p(s_i)\}$ неизменно, и вероятность появления элементов независима, то мы можем найти среднюю длину кодов как среднее взвешенное:

$$H = -\sum_i p(s_i) \cdot \log_2 p(s_i).$$

Это значение также называется энтропией распределения вероятностей F или энтропией источника в заданный момент времени.

1.2. Алгоритм Хаффмана. Эта базовая теория лежит в основе исследований Дэвида Хаффмана [3], который перенес теоретические исследования в практическую область информационных технологий. Алгоритмы сжатия Хаффмана и его варианты оставались наиболее популярными методами сжатия данных вплоть до 1977 года, когда Абрахам Лемпел и Якоб Зив предложили совершенно иной подход к проблеме сжатия данных [4]. Они выдвинули идею формирования «словаря» общих последовательностей данных. При этом сжатие данных осуществляется за счет замены последовательности символов соответствующими кодами из словаря.

1.3. Алгоритмы семейства LZ. Результатом исследований Лемпеля и Зива стали алгоритмы сжатия данных LZ77 [4], [5] и LZ78 [5]. Однако, у этих алгоритмов был существенный недостаток: много времени тратится на создание эффективного словаря. Решение этой проблемы было предложено Терри Велчем, который расширил алгоритм LZ78, создав новый вариант, известный как LZW [6].

Алгоритмы семейства LZ получили широкое распространение благодаря своей универсальности. На сегодняшний день существует большое количество модификаций: LZMW (1984 г.) [18], LZB (1987 г.) [19], LZH (1987 г.) [7], LZFG (1987 г.) [20], LZBW (1991 г.) [21], LZRW1 (1991 г.) [22], LZR (1981 г.) [23], LZR (1995 г.) [24], LZ77-PM (1995 г.) [25], LZFG-PM (1995 г.) [25], LZW-PM (1995 г.) [25].

Одной из значимых модификаций алгоритмов семейства LZ была комбинация алгоритма LZ77 и алгоритма Хаффмана, которую предложил Фил Кац [7].

Архиваторы, использующие алгоритмы сжатия данных семейства LZ: 7-Zip, ACE, ARJ, ARJZ, CABARC, Imp, JAR, PKZIP, RAR, WinZip, Zip.

В таблице 1.1 [8] представлены результаты сравнения некоторых архиваторов по степени сжатия файлов набора CalgCC [17], являющегося международным эталоном для исследования алгоритмов сжатия данных

Таблица 1.1

Имя файла	ARJ	PKZIP	ACE	RAR	CABARC	7-Zip
Bib	3.08	3.16	3.38	3.39	3.45	3.62
Book1	2.41	2.46	2.78	2.80	2.91	2.94
Book2	2.90	2.95	3.36	3.39	3.51	3.59
Geo	1.48	1.49	1.56	1.53	1.70	1.89
News	2.56	2.61	3.00	3.00	3.07	3.16
Obj1	2.06	2.07	2.19	2.18	2.20	2.26
Obj2	3.01	3.04	3.39	3.38	3.54	3.96
Paper1	2.84	2.85	2.91	2.93	2.99	3.07
Paper2	2.74	2.77	2.86	2.88	2.95	3.01
Pic	9.30	9.76	10.53	10.39	10.67	11.76
Progc	2.93	2.94	3.00	3.01	3.04	3.15
Progl	4.35	4.42	4.49	4.55	4.62	4.76
Progp	4.32	4.37	4.55	4.57	4.62	4.73
Trans	4.65	4.79	5.19	5.23	5.30	5.56
Итого	3.47	3.55	3.80	3.80	3.90	4.10

1.4. Алгоритмы сжатия данных, основанные на методе контекстного моделирования. Одновременно с попытками улучшить эффективность алгоритмов семейства LZ, проводились исследования в поисках новых подходов к сжатию данных. Так, в 1971 году Риссанен и Лэнгдон предложили метод контекстного моделирования [9]. Предложенный метод опирается на парадигму сжатия с помощью универсального моделирования и кодирования. Свое развитие метод контекстного моделирования получил в исследованиях Клири Уиттена, который предложил использовать технику предсказания по частичному поведению (PPM) [10]. Дальнейшие исследования в рамках PPM были направлены на оценку вероятности ухода, адаптивные методы. Одной из первых попыток улучшения оценки вероятности ухода был метод Z, описанный Чарльзом Блумом [11], и получивший название PPMZ. Современным адаптивным методом является подход, предложенный Дмитрием Шкариным. Метод получил название PPMd [12].

Таблица 1.2

	CM	HA	PPMY	RKUC	PPMN	PPMd	PPMonstr
Bib	3.01	4.12	4.55	4.55	4.57	4.62	4.76
Book1	2.99	3.27	3.72	3.62	3.64	3.65	3.74
Book2	3.19	3.74	4.35	4.30	4.31	4.35	4.49
Geo	1.74	1.72	1.74	2.12	1.96	1.84	1.92
News	1.69	2.19	2.09	2.25	2.32	2.26	2.29
Obj1	2.37	3.39	3.59	3.51	3.59	3.65	3.74
Obj2	2.32	3.07	3.46	3.79	3.65	3.62	3.79
Paper1	2.37	3.39	3.59	3.51	3.59	3.65	3.74
Paper2	2.61	3.43	3.67	3.57	3.62	3.67	3.77
Pic	9.30	10.00	10.26	10.67	11.01	10.53	11.43
Progc	2.27	3.36	3.51	3.45	3.54	3.60	3.70
Progl	3.07	4.68	5.30	5.37	5.26	5.44	5.76
Progp	2.99	4.74	5.33	5.30	5.19	5.26	5.76
Trans	2.96	5.23	6.30	6.40	6.17	6.35	6.84
Итого	2.62	3.07	4.00	4.39	4.46	4.46	4.70

Архиваторы, использующие методы контекстного моделирования: HA, CM, PPMY, RKUC, PPMN, PPMd и PPMonstr.

В таблице 1.2 [8] представлены результаты сравнения некоторых архиваторов по степени сжатия файлов набора CalgCC [17].

1.5. Алгоритмы сжатия данных, основанные на предварительном преобразовании. Еще одним подходом к сжатию данных является подход, основанный на предварительном преобразовании информации для последующего более эффективного сжатия. Впервые такой подход был опубликован 10 мая 1994 г. [13]. Его авторами являются Дэвид Уиллер и Майк Барроуз. Алгоритм был назван преобразованием Барроуза-Уиллера (BWT). Особенностью преобразования является тот факт, что обычные алгоритмы сжатия, примененные на преобразованном массиве данных, работают не так эффективно, как алгоритмы, специально разработанные для сжатия преобразованных данных. Среди методов, разработанных для сжатия данных после преобразования Барроуза-Уиллера, можно выделить следующие: RLE (кодирование длин серий) [14], MTF (метод перемещения стопки книг) [15], DC (кодирование расстояний) [16], Метод Хаффмана [3], Арифметическое кодирование [26]–[33].

Таблица 1.3 – Последовательность применения методов, совместно с BWT

Шаг	Используемый алгоритм	
1	Кодирование длин серий (необязательно)	
2	Преобразование Барроуза-Уиллера	
3	Перемещение стопки книг	Кодирование расстояний
4	Кодирование длин серий (необязательно)	
5	Метод Хаффмана	Арифметическое кодирование

Таблица 1.4

Архиватор, версия и параметры	Р-р сжатого файла, байт	Время сжатия, с	Время разжатия, с
SBC 0.860 m3a	126,811	0.69	0.42
DC 0.99.298b	127,377	0.38	0.18
ARCb2	128,685	0.38	0.23
YBS 0.03-m256k	130,356	0.37	0.24
Compressiab256	131,737	0.61	0.40
BA1.01b5-24-r	132,651	0.41	0.30
Zzip 0.36-al	132,711	0.65	0.40
DC 0.99.298b-a	133,825	0.34	0.23
YBS 0.03e	133,915	0.37	0.25
BWC/PGCC 0.99 m600k	134,183	0.33	0.19
bzip2/ PGCC1.0b7-6	134,932	0.44	0.14
Szipl.12b21 o0	134,945	0.90	0.15
IMP1.10-2ul000	135,431	0.30	0.12
ICTUC1.0	136,842	0.41	0.29
BA1.01b5-24-z	137,566	0.49	0.31
Szipl.12b21o4	141,784	0.17	0.18

Архиваторы, использующие методы BWT: SBC, DC, ARC, YBS, Compressia, BA, Zzip, DC, YBS, BWC/PGCC, bzip2/PGCC, szip, IMP, ICTUC.

В таблице 1.4 [8] представлены результаты сравнения некоторых архиваторов по степени сжатия файла, содержащего смесь текстовых и бинарных данных, размером 427,520 байт.

1.6. Сравнительный анализ методов сжатия данных. В таблице 1.5 [8] приведен сравнительный анализ трех методов сжатия данных в зависимости от типа данных.

2 Алгоритмы сжатия изображений

Все существующие алгоритмы сжатия графических данных делятся на две категории [14]:

1. Сжатие с потерями информации. Т. е. компрессия достигается за счет удаления мало значимой информации, не искажающей общей картины. После сжатия с потерями, изображение не может быть восстановлено в исходное состояние.

2. Сжатие без потерь информации. Т. е. компрессия достигается с сохранением всей информации об изображении, а значит изображение может быть восстановлено в исходное состояние.

Алгоритмы сжатия с потерями имеют более широкое распространение, чем алгоритмы сжатия без потерь, поскольку:

- алгоритмы сжатия с потерями позволяют обеспечить более высокий коэффициент сжатия;

- определен перечень коэффициентов, при которых потерянная информация при сжатии не заметна на уровне человеческой интерпретации [26].

К недостаткам алгоритмов сжатия с потерями можно отнести следующие:

- алгоритмы сжатия с потерями не могут быть применены на информации, для которой основным критерием служит точность восстановления (текстовая информация, бинарные файлы, медицинские изображения и др.);

- с каждой следующей стадией компрессии и декомпрессии, качество изображения заметно ухудшается.

Далее рассмотрим алгоритмы сжатия графических данных без потерь.

2.1. RLE. Базовым методом сжатия графической информации без потерь является метод кодирования длин серий (RLE) [14]. Суть алгоритма в том, что повторяющиеся символы могут быть заменены цифрой, которая указывает количество повторов, и самим символом. Наилучший результат метод RLE достигает на изображениях, содержащих большие площади смежного цвета (например, монохромные изображения). Но на сложных цветных изображениях RLE может дать обратный результат, увеличив размер файла.

2.2. LZ. Также к изображениям могут быть применены алгоритмы семейства LZ и метод Хаффмана, которые были описаны выше. Эти алгоритмы относятся к группе алгоритмов сжатия без потерь информации.

Таблица 1.5 – Сравнительный анализ трех методов сжатия данных в зависимости от типа данных

Параметр	Метод	Однородные данные (текст)	Однородные данные с большой избыточностью (исходные код программ)	Неоднородные данные	Данные с малой избыточностью
Степень сжатия	Контекстное моделирование (PPM)	Высокая	Высокая	Высокая	Невысокая
	Кодирование (LZ)	Высокая	Высокая	Без фрагментирования – низкая	
	Преобразование BWT	Низкая	При большом количестве длинных повторов – высокая	Высокая	
Скорость кодирования	Преобразование BWT	Высокая	Средняя	Высокая	Высокая
	Контекстное моделирование (PPM)	При большом порядке модели – самая низкая; при небольшом – немного быстрее BWT	Если использовать сложное моделирование – высокая	Средняя	Низкая
	Кодирование (LZ)	средняя, а при малом словаре – самая высокая	средняя, а при малом словаре – самая высокая	Высокая	Высокая
Скорость декодирования	Кодирование (LZ)	Примерно в 10 раз выше скорости кодирования			
	Контекстное моделирование (PPM)	На 5–10% медленнее кодирования			
	Преобразование BWT	В 2–4 раза выше скорости кодирования			
Требуемый объем памяти при сжатии	Преобразование BWT	Постоянный при сжатии данных любого типа			
	Контекстное моделирование (PPM)	Варьируется в широких пределах, в зависимости от сложности моделирования и порядка модели; вырастает для очень неоднородных данных; в зависимости от структуры хранения контекстной информации может увеличиваться для мало избыточных данных			
	Кодирование (LZ)	Пропорционально размеру словаря			
Требуемый объем памяти при разжатии	Кодирование (LZ)	Минимальный			
	Контекстное моделирование (PPM)	Максимальный			
	Преобразование BWT	Средний			

2.3. CCITT.4 Group 3 и CCITT.6 Group 4. Другой группой алгоритмов сжатия графической информации без потерь являются алгоритмы CCITT.4 Group 3 [27] и CCITT.6 Group 4 [28]. Эти алгоритмы были предложены Международным консультативным комитетом по телефонии и телеграфии (СЦИТТ) в 1985 г. В основном они применяются для кодирования и сжатия 1-битных (монохромных) изображений (факсимильные сообщения). К достоинствам данных алгоритмов можно отнести простоту реализации [8] и достаточно высокие коэффициенты сжатия [29]. К недостаткам: узкая применимость (монохромные изображения).

2.4. JPEG. Одним из самых распространённых алгоритмов сжатия графической информации является алгоритм JPEG, разработанный в 1990 году объединенной группой Международной организации по стандартизации (ISO) и Международным консультативным комитетом по

телефонии и телеграфии (СЦИТТ). Основная спецификация JPEG описана в [30]. Основная область применения JPEG – 24-х битовые изображения. Изначально алгоритм JPEG относился к алгоритмам сжатия с потерей информации. Эффект компрессии достигался за счет частичного удаления незаметных для человеческого глаза данных изображения, используя метод дискретного косинусного преобразования (DCT) [31]. На последнем этапе применяется кодирование по методу Хаффмана [3]. Однако, в 1993 году было разработано расширение алгоритма «JPEG без потерь» [30]. Разработанный подход использует технику адаптивного предсказания значения текущего пикселя, анализируя соседние.

2.5. JPEG 2000. Следующим этапом развития алгоритма JPEG был формат JPEG 2000 [32], разработанный группой JPEG ISO в 2000 году. JPEG 2000 предусматривает сжатие, как с потерей информации, так и без потерь. Основные

отличия от алгоритма JPEG заключаются в следующем:

- лучшее качество изображения при сильной степени сжатия;
- поддержка кодирования отдельных областей с лучшим качеством;
- основной алгоритм сжатия заменен на wavlet-преобразование [41], [46];
- для повышения степени сжатия в алгоритме используется арифметическое сжатие;
- поддержка сжатия 1-битовых изображений;
- на уровне формата JPEG 2000 поддерживается прозрачность.

В итоге, JPEG 2000 на 5–100% превосходит по эффективности алгоритм JPEG [43], [44].

В 1996 году был разработан алгоритм сжатия без потерь PNG [33]. Сжатие в алгоритме PNG основано на методе Deflate и эффективен при сжатии изображений с глубиной цвета от 1бит до 48-бит.

2.6. Сравнение алгоритмов сжатия изображений. В таблицах 2.1 и 2.2 приведено сравнение параметров рассмотренных алгоритмов[8].

Таблица 2.1

Алгоритм	Особенности изображения, за счет которых происходит сжатие
RLE	Подряд идущие одинаковые цвета: 2 2 2 2 2 15 15 15
LZ	Одинаковые подцепочки: 2 3 15 40 2 3 15 40
Хаффмана	Разная частота появления цвета: 2232243222 4
СЦИТТ – 3	Преобладание белого цвет в изображении, большие области, заполненные одним цветом
JPEG	Отсутствие резких границ

Таблица 2.2

Алгоритм	Коэффициенты сжатия	Симметричность по времени	На что ориентирован
RLE	32, 2, 0.5	1	3,4-битовые
LZ	1000, 4, 5/7	1.2–3	1-8-битовые
Хаффман	8, 1.5, 1	1–1.5	8-битовые
СЦИТТ – 3	213, 5, 0.25	1	1-битовые
Lossless JPEG	20, 2, 1	1	24-битовые, 8-битовые
JPEG 2000	2–200	1–1.5	Полноцветные 24-битовые, 8-битовые, 1-битовые

3 Современные подходы к проблеме сжатия данных

В настоящее время исследования в области сжатия информации ведутся по трем основным направлениям:

- повышение эффективности сжатия;
- увеличение скорости работы алгоритма сжатия;
- поиск нового подхода к сжатию информации (новая система контекста).

Одним из таких подходов является использование больших словарей с синтаксической и семантической информацией, что позволит получить преимущество от имеющейся в тексте связанности. Так, Уолкери Эмслер [34] провел анализ 8 миллионов слов из New York Times News Service для Webster's Seventh New Collegiate Dictionary. Полученный результат свидетельствовал о том, что 64% слов отсутствовали в текущих словарях (около 1/4 из них – грамматические формы слов, еще 1/4 – собственные существительные, 1/6 – слова, написанные через дефис, 1/12 – напечатаны с орфографическими ошибками). Поэтому поиски методов улучшения сжатия текстов, вероятно, будут вестись в направлении исследований в области контекстного анализа.

Другой подход противоположен наукоемкому подходу контекстного анализа. Он заключается в поддержании полной адаптивности системы и исследований в поисках улучшений в существующих алгоритмах сжатия данных. Например, ведутся исследования в области выделения кодов ухода в частично соответствующих контекстуальных моделях, например, алгоритм Баума-Уэлча [35], направленный на определение скрытых моделей Маркова [36], и в последнее время вновь заинтересовавший исследователей [37].

Другим направлением исследований является увеличение скорости работы алгоритмов сжатия. Исследования в этой области вплотную связаны с развитием микроэлектроники. Так, ведутся работы по ускорению работы алгоритмов за счет большего использования памяти. Например, применение архитектуры RISC [38], разными путями воздействует на баланс между хранением и выполнением. Ведутся работы в более глубоком применении аппаратного кодирования. Например, Гонзалес-Смит и Сторер [39] разработали для сжатия LZ параллельные алгоритмы поиска, используя аппаратные возможности микросхем.

Последнее направление – разработка новых подходов к сжатию данных. Например, исследуется возможность объединения в единую систему ряда разработок из разных областей применения: текст-процессор MaxWrite [40], цифровой факсимильный аппарат [41], почту и новости. Так же существуют разработки в области объединения адаптированного алгоритма с аппаратным контроллером [42]. Развивается идея внедрения алгоритмов сжатия в телефонии и т. д.

Еще одной областью, которая может дать толчок в развитии алгоритмов сжатия, является область криптографии и защиты данных [43].

Сжатые данные зачастую уже становятся трудными для восприятия, тем самым придавая некую степень секретности.

Заключение

С момента первых исследований в области сжатия данных было разработано большое количество алгоритмов сжатия данных, которые имеют широкий спектр применения. Сжатие данных востребовано в современных программно-технических комплексах различного назначения.

Каждый год разрабатываются новые информационно-технические решения, которые требуют большого объема данных, к которому необходимо обеспечить оперативный доступ для обработки и последующих исследований. Это приводит к тому, что исследования в области сжатия данных играют все более важную роль.

Материалы, изложенные в статье, представляют интерес как для информационно-технических решений, так и для прикладного оборудования, которое создается с использованием достижений в области сжатия данных.

ЛИТЕРАТУРА

1. *Hartley, R.V.L.* Transmission of information / R.V.L. Hartley // *Bell System Technical Journal*. – 1928. – Vol. 7, iss. 3. – P. 535–563.
2. *Shannon, C.E.* A mathematical theory of communication / C.E. Shannon // *The Bell System Technical Journal*. – 1948. – Vol. 27. – P. 379–423.
3. *Huffman, D.A.* A method for the construction of minimum redundancy codes / D.A. Huffman // *J. Proc. of IRE*. – 1952. – Vol. 40. – P. 1098–1101.
4. *Ziv, J.* A universal algorithm for sequential data compression / J. Ziv, A. Lempel // *IEEE Transactions on Information Theory*. – May 1977. – Vol. 23, iss. 3. – P. 337–343.
5. *Ziv, J.* Compression of individual sequences via variable-rate coding / J. Ziv, A. Lempel // *IEEE Transactions on Information Theory*. – Sept. 1978. – Vol. 24, iss. 5. – P. 530–536.
6. *Welch, T.A.* A technique for high-performance data compression / T. A. Welch // *IEEE Computer*. – June 1984. – Vol. 17, iss. 6. – P. 8–19.
7. *IETF Tools* [Electronic resource]: DEFLATE Compressed Data Format Specification v.1.3 (RFC 1951). – Mode of access: <https://tools.ietf.org/html/rfc1951>. – Date of access: 28.09.2015.
8. *Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео* / Д. С. Ватолин [и др.]. – М.: ДИАЛОГ–МИФИ, 2002. – 384 с.
9. *Rissanen, J.J.* Universal modeling and coding / J.J. Rissanen, G.G. Langdon // *IEEE Transactions on Information Theory*. – Jan. 1981. – Vol. 27, iss. 1. – P. 12–23.
10. *Cleary, J.G.* Data compression using adaptive coding and partial string matching / J.G. Cleary, I.H. Witten // *IEEE Transactions on Communications*. – April 1984. – Vol. 3, iss. 4. – P. 396–402.
11. *California Institute of Technology* [Electronic resource]: Solving the problems of context modeling / C. Bloom. Mode of access: <http://www.cbloom.com/papers/ppmz.zip>. – Date of access: 28.09.2015.
12. *Шкарин Д.А.* Повышение эффективности алгоритма PPM / Д.Ф. Шкарин // *Проблемы передачи информации*. – 2001. – Т. 34 № 3. – С. 44–54.
13. *HP Labs* [Electronic resource]: A Block-sorting Lossless Data Compression Algorithm / M. Burrows, D.J. Wheeler // *SRC Research Report 124, Digital Systems Research Center, Palo Alto, May 1994*. Mode of access: <http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-124.pdf>. – Date of access: 29.09.2015.
14. *The national archives* [Electronic resource]: Digital preservation guidance note 5: image compression, 2003 / A. Brown. Mode of access: <https://www.nationalarchives.gov.uk/documents/selecting-file-formats.pdf>. – Date of access: 29.09.2015.
15. *Ryabko, B.Ya.* Data Compression by Means of a “Book Stack” / B.Ya. Ryabko // *Problems of Information Transmission*. – 1980. – Vol. 16, iss. 4. – P. 265–269.
16. *Fenwick, P.M.* Block sorting text compression: papers from the Australasian Computer Science Conference, ACSC'96, Melbourne, Australia, Feb 1996. – Melbourne, 1996. – P. 193–202.
17. *Bell, T.C.* Modeling for text compression II *ACM Computer Survey* / T.C. Bell, I.H. Witten, J.G. Cleary. – 1989. – Vol. 24, № 4. – P. 555–591.
18. *Miller, V.S.* Variations on a theme by Ziv and Lempel / V.S. Miller, M.N. Wegman // *In Combinatorial Algorithms on Words. NATO ASI Series: papers from the conference, Berlin, 1984; ed.: A. Apostolico [et al.]. – Springer-Verlag, 1984. – Vol. F12. – P. 131–140.*
19. *Bell, T.C.* A unifying theory and improvements for existing approaches to text compression: dissertation, Dept. of Computer Science / Ph. D. T.C. Bell; Univ. of Canterbury, New Zealand. – Canterbury, 1987. – P. 9.
20. *Fiala, E.R.* Data compression with infinite windows / E.R. Fiala, D.H. Greene // *Commun. ACM*. – Apr. 1989. – Vol 32, iss. 4. – P. 490–505.
21. *Bender, P.E.* New asymptotic bounds and improvements on the Lempel-Ziv data compression algorithm / P.E. Bender, J.K. Wolf // *IEEE Transactions on Information Theory*. – May 1991. – Vol. 37, iss. 3. – P. 721–727.
22. *Williams, R.N.* An Extremely Fast Ziv-Lempel Data Compression Algorithm / R.N. Williams // *Data Compression Conference: papers from the conference, Snowbird, Utah, 8–11 April, 1991. – P. 362–371.*
23. *Roden, M.* Linear algorithm for data compression via string matching / M. Roden, V.R. Pratt,

- S. Even // J. ACM. – Jan. 1981. – Vol. 28, iss. 1. – P. 6–24.
24. *C. Bloom* [Electronic resource]: New Techniques in Context Modeling and Arithmetic Encoding / C. Bloom. Mode of access: <http://www.cbloom.com/papers/context.pdf>. – Date of access: 29.09.2015.
25. *Hoang, D.T.* Multiple-dictionary compression using partial matching / D.T. Hoang, P.M. Long, J.S. Vitter // Proceedings of Data Compression Conference: papers from the conference, Washington, DC, USA, March 1995. – P. 272.
26. *Земляченко А.Н.* Сжатие изображений без визуально заметных искажений / А.Н. Земляченко, В.В. Лукин // Радиоэлектронные и компьютерные системы. – 2011. – № 3 – С. 73–79.
27. *ITU* [Electronic resource]: Standardisation of Group 3 Facsimile apparatus for document transmission. CCITT Recommendations. Fascicle VII.2. 1980. T.4 Mode of access: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-T.4-198811-S!!PDF-E&type=items. – Date of access: 30.09.2015
28. *ITU* [Electronic resource]: Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, Recommendation T.6, Volume VII, Fascicle VII.3. Mode of access: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-T.6-198811-I!!PDF-E&type=items. – Date of access: 30.09.2015.
29. *Нгуен, Д.М.* Разработка методики оценки качества сжатых изображений: дис. канд. техн. наук.: 25.00.35 // Нгуен Динь Минь. – М., 2006. – 163 с.
30. *Wallace, G.K.* The JPEG still picture compression standard / G.K. Wallace // Communication of ACM. – April 1991. – Vol. 34, № 4. – P. 30–44.
31. *Narasimha, M.* On the Computation of the Discrete Cosine Transform / M. Narasimha, A. Peterson // IEEE Transactions on Communications. – June 1978. – Vol. 26, iss. 6. – P. 934–936.
32. *Information technology – JPEG 2000 image coding system.* – Part 1: Core coding system: ISO/IEC 15444-1 2001.
33. *Portable Network Graphics* [Электронный ресурс] – Mode of access: <http://libpng.org/pub/png/> – Date of access: 30.09.2015.
34. *Walker, D.E.* The use of machine-readable dictionaries in sublanguage analysis / D.E. Walker, R.A. Amsler // In Analysis languages in restricted domains: Sublanguage description and processing: papers from the conference, Hillsdale, NJ, USA, 1986; ed.: R. Grishman [et al.]. – NJ, 1986. – P. 69–83.
35. *A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains* / L. E. Baum, T. Petrie, G. Soules, N. Weiss // Ann. Math. Stat. – 1970. – Vol. 41. – P. 164–171.
36. *Rabiner, L.R.* An Introduction to Hidden Markov models / L.R. Rabiner, B.H. Juang // IEEE ASSP Mag. – Jan. 1986. – P. 4–16.
37. *Levinson, S.E.* An introduction to the application of the theory of probabilistic function of a Markov process to automatic speech recognition / S.E. Levinson, L.R. Rabiner, M. Sondni // Bell Syst. Tech. J. – Apr. 1983. – Vol. 62, iss. 4. – P. 1035–1074.
38. *Jagger, D.* Fast Ziv-Lempel decoding using RISC architecture / D. Jagger // Dept. of Computer Science: Res. Rept., Univ. of Canterbury, New Zealand. – 1989.
39. *Gonzalez-Smith, M.E.* Parallel algorithms for data compression / M.E. Gonzalez-Smith, J.A. Storer // J. ACM. – 1985. – Vol. 32, iss. 2. – P. 344–373.
40. *Young, D.M.* MacWrite file formats / D.M. Young // Wheels for the mind: Newsletter of the Australian Apple University Consortium, University of Western Australia, Nedlands, WA 6009, Australia. – Nedlands, 1985. – P. 34.
41. *Hunter, R.* International digital facsimile coding standards / R. Hunter, A.H. Robinson // Proceedings of the IEEE. Jul. 1980. – Vol. 68, iss. 7. – P. 854–867.
42. *Auslander, M.* PCTERM: A terminal emulator using compression / M. Auslander, W. Harrison, V. Miller, M. Wegman // Proceedings of the IEEE. – 1985. – P. 860–862.
43. *Witten, I.H.* Picture coding and transmission using adaptive modelling of quad trees / I.H. Witten, J. Cleary // Proceeding of the International Electrical, Electronics conference 1: papers from the conference, Toronto, Canada, 1983. – Toronto, 1983. – P. 222–225.

Поступила в редакцию 01.10.15.