

УДК 004.932.72

## СЖАТИЕ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРЬ НА ОСНОВЕ КОДИРОВАНИЯ ДЛИН СЕРИЙ

Х.К. АЛЬ-БАХДИЛИ, Е.Г. МАКЕЙЧИК, В.Ю. ЦВЕТКОВ, В.К. КОНОПЕЛЬКО

*Белорусский государственный университет информатики и радиоэлектроники  
П. Бровка, 6, Минск, 220013, Беларусь*

*Поступила в редакцию 16 ноября 2015*

Предложены модифицированные алгоритмы кодирования длин серий для сжатия полутоновых изображений без потерь, отличающиеся от базового алгоритма учетом вероятности повтора значений пикселей изображений и обеспечивающие уменьшение размера кодированных данных.

*Ключевые слова:* сжатие изображений, кодирование длин серий.

### Введение

Для сжатия изображений без потерь в настоящее время широко используются алгоритмы JPEG [1] и JPEG 2000 [2], основанные на энтропийном кодировании (Хаффмана и арифметическом) коэффициентов дискретно-косинусного и вейвлет-преобразований, а также алгоритм Rar и входящие в состав архиватора Zip алгоритмы Deflate и LZMA, кодирующие значения пикселей [3]. Данные алгоритмы позволяют сжимать изображения без потерь примерно до 2 раз, однако их использование требует значительных вычислительных ресурсов и времени на буферизацию всего изображения и реализацию преобразований. В тех случаях, когда временные и вычислительные ресурсы ограничены необходимо использовать более простые алгоритмы эффективного кодирования. К ним относится алгоритм кодирования длин серий RLE (Run-Length Encoding) [4], основанный на учете повторов символов. Он может использоваться для сжатия без потерь изображений с малым числом резких перепадов яркости (мультипликационных, медицинских, квантованных и сегментированных). Недостатком данного алгоритма является отсутствие учета вероятности повтора значений пикселей изображений.

Целью работы является разработка модифицированных алгоритмов кодирования длин серий, основанных на учете вероятности повтора значений пикселей изображений.

### Алгоритм RLE

Алгоритм RLE основан на подсчете числа повторов значений следующих друг за другом символов, может иметь блочную и поточную реализацию. Блочная реализация алгоритма RLE предполагает предварительное накопление кодируемых данных для их анализа и выбора параметров кодирования. Анализ производится на основе таблицы I/N-длин серий, в которой одному или нескольким одинаковым по значению и следующим друг за другом символам  $I$  ставится в соответствие их количество  $N$  (табл. 1).

Таблица 1. Таблица длин серий

$I$	$i(0)$	$i(1)$	...	$i(s)$	...
$N$	$n(0)$	$n(1)$	...	$n(s)$	...

На основе таблицы длин серий определяются битовая глубина  $BD_I$  изображения и битовая глубина значений длин серий  $BD_N$  с помощью выражений:

$$BD_I = \left\lceil \log_2 \left( \max(i(s))_{(s=0, S-1)} \right) \right\rceil$$

$$BD_N = \left\lceil \log_2 \left( \max(n(s))_{(s=0, S-1)} \right) \right\rceil$$

где  $i(s)$  – значение кодируемого  $s$ -го символа из таблицы длин серий;  $n(s)$  – число повторов  $s$ -го символа (длина серии);  $S$  – число кодируемых символов (число строк в таблице длин серий).

Затем осуществляется кодирование длин серий в результате формирования на выходе кодера пар  $\{i(s), n(s)\}$ . Алгоритм RLE сжатия изображений на основе кодирования длин серий и структура закодированных данных представлены на рис. 1. Блочная реализация алгоритма RLE позволяет получить минимальный объем кода. Ее недостатком является задержка при кодировании, необходимая для накопления данных и построения таблицы длин серий.

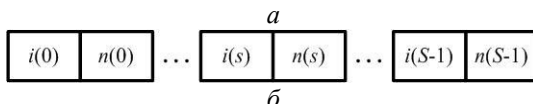
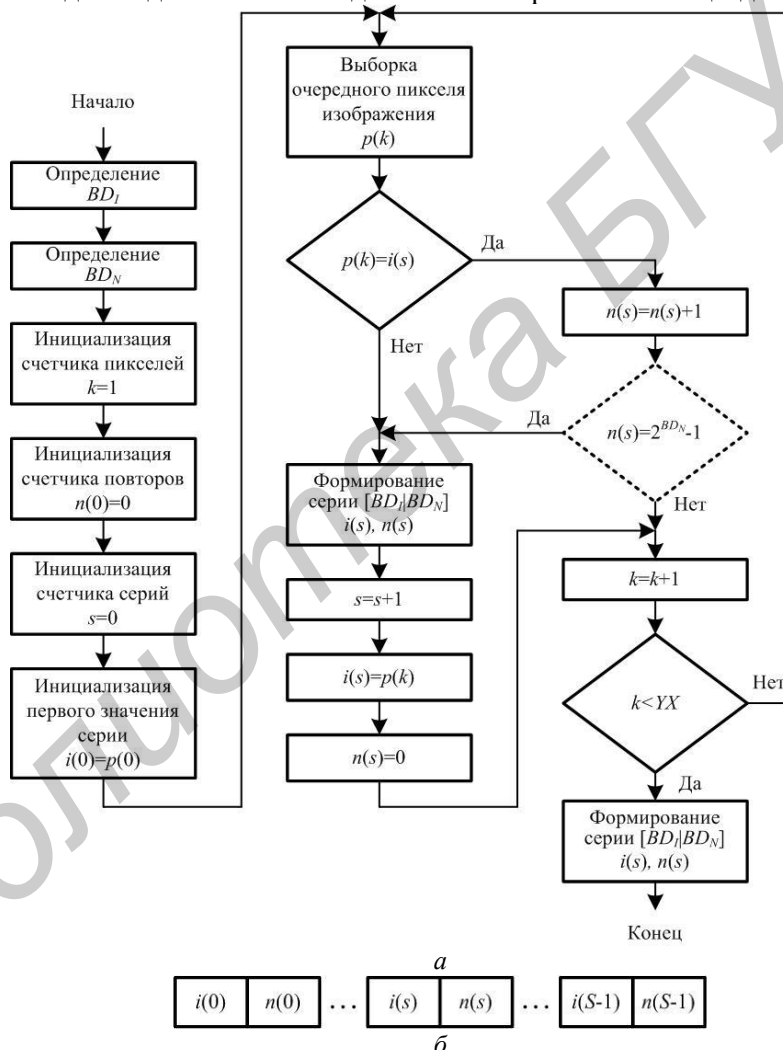


Рис. 1. Кодирование длин серий: а – алгоритм RLE; б – структура закодированных данных

При поточной реализации алгоритма RLE таблица длин серий не строится, а подсчет числа символов осуществляется по мере их поступления. Это позволяет существенно повысить скорость кодирования. Структуры алгоритма и закодированных данных при поточной реализации алгоритма RLE такие же, как при блочной. Значения битовой глубины  $BD_I$  изображения и битовой глубины значений длин серий  $BD_N$  выбираются независимо от поступающих данных. Они могут быть выбраны с избытком или недостатком (при

переполнении  $n(s)$  формируется новая серия  $\{i(s), n(s)\}$ . В любом случае это приводит к увеличению объема кода, что является недостатком поточной реализации алгоритма RLE.

При кодировании изображений размер  $R_{I/N}$  (бит) кода, коэффициент сжатия  $CR$  и вычислительная сложность  $C_{I/N}$  для алгоритма RLE определяются с помощью выражений:

$$R_{I/N} = S(BD_I + BD_N),$$

$$CR = 8YX/R_{I/N},$$

$$C_{I/N} = YX + 4S,$$

где  $YX$  – размер изображения, определяющий число операций на буферизацию и формирование таблицы  $I/N$  размером  $S$  записей;  $4S$  – дополнительные операции на поиск максимального значения по  $I$  ( $S$  операций), поиск максимального значения по  $N$  ( $S$  операций), кодирование и передачу ( $2S$  операций).

### Алгоритм кодирования длин серий I/2B/N

Предлагается модификация I/2B/N алгоритма кодирования длин серий RLE для сжатия полутонных изображений, основанная на учете вероятности повтора значений пикселей в строках. В процессе выполнения алгоритма строится таблица  $N/P$  вероятностей повторов (табл. 2), в которой каждому значению длины серии  $n$  ставится число таких серий  $p_n(n)$ .

Таблица 2. Таблица вероятности повторов

$N$	1	2	...	$n$	...
$P$	$p_n(1)$	$p_n(2)$	...	$p_n(n)$	...

Для изображений характерно постепенное убывание функции  $p_n(n)$  при увеличении значения  $n$ . При кодировании длины серии по алгоритму I/2B/N сначала формируется символ  $I$ . Затем, если символ  $I$  повторяется, то формируется бит  $b1(s) = 1$ , иначе  $b1(s) = 0$  (первый символ повтора). Если символ  $I$  повторяется снова, то формируется бит  $b2(s) = 1$ , иначе  $b2(s) = 0$  (второй символ повтора). Если символ  $I$  повторяется снова, то формируется символ  $n(s)$ , учитывающий число повторов (изначально  $n(s) = 0$ , если символ  $I$  повторяется снова, то  $n(s) > 0$ ). В результате могут формироваться серии вида:

$$\{i(s), b1(s) = 0\}, \{i(s), b1(s) = 1, b2(s) = 0\}, \{i(s), b1(s) = 1, b2(s) = 1, n(s)\}.$$

Структуры алгоритма I/2B/N кодирования длин и закодированных им данных серий приведены на рис. 2. Алгоритм I/2B/N кодирования длин серий также, как алгоритм RLE, может иметь блочную и поточную реализации.

Размер  $R_{I/2B/N}$  (бит) кода для алгоритма I/2B/N определяется с помощью выражения

$$R_{I/2B/N} = S(BD_I + 1) + \sum_{s=0}^{S-1} b1(s) + BD_N \sum_{s=0}^{S-1} b2(s).$$

Для определения вычислительной сложности используется выражение

$$C_{I/2B/N} = YX + 4S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s). \quad (1)$$

Выражение (1) учитывает  $YX$  операций на буферизацию и формирование таблицы  $I/N$ -длин серий размером  $S$  записей,  $S$  операций на поиск максимального значения по  $I$ ,  $S$  операций на поиск максимального значения по  $N$ ,  $2S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s)$  операций на кодирование и передачу.

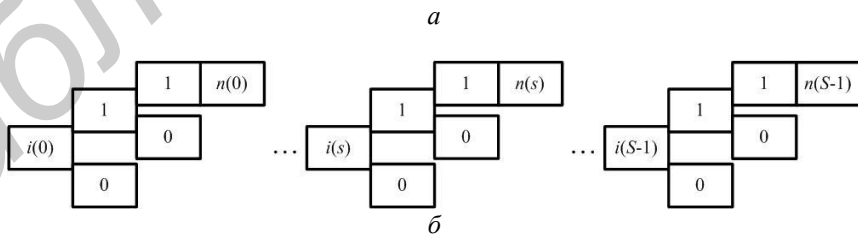
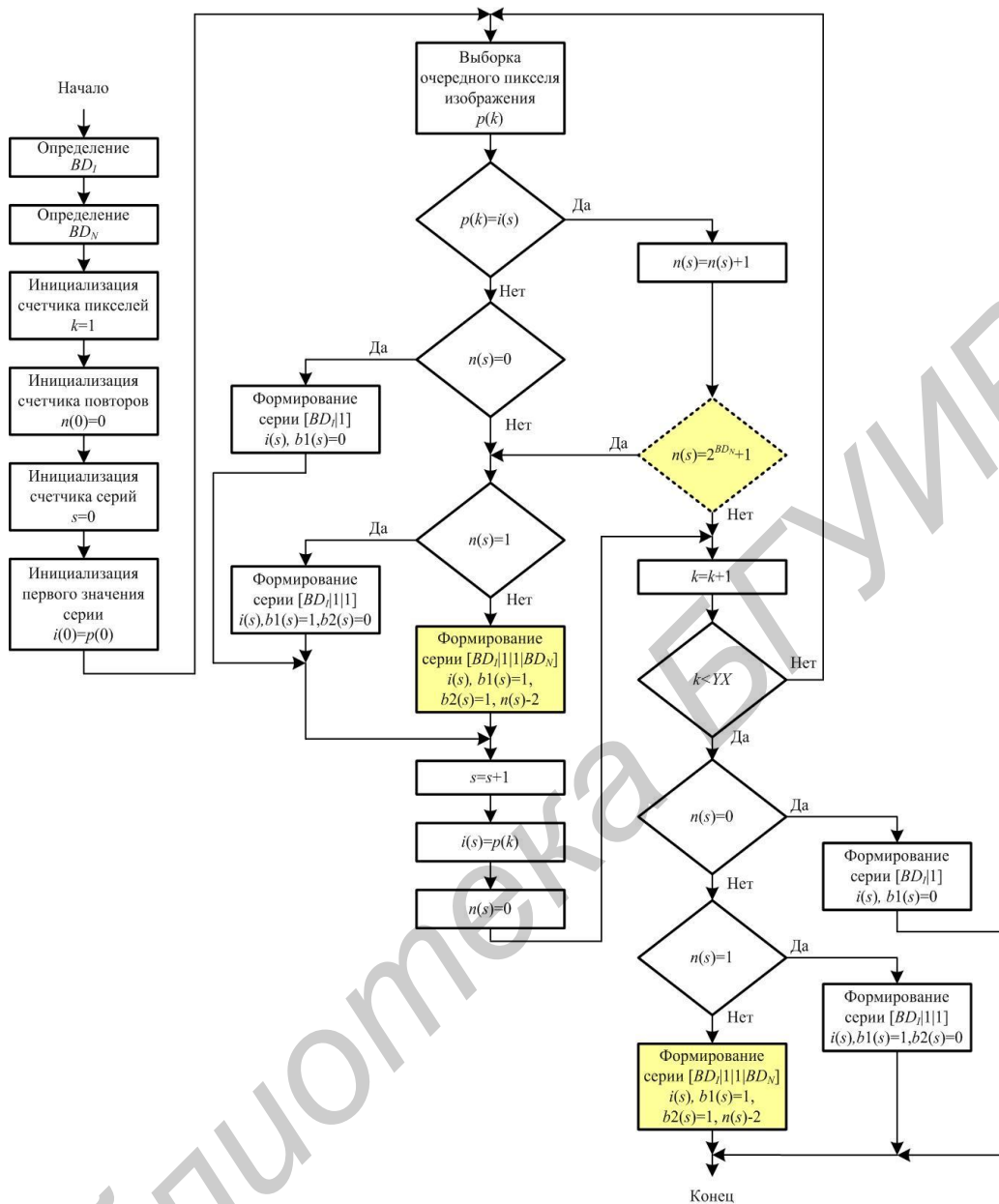


Рис. 2. Модифицированное I/2B/N кодирование длин серий: а – алгоритм I/2B/N кодирования длин серий; б – структура закодированных данных по алгоритму I/2B/N

### Алгоритм кодирования длин серий I/3B/N

Предлагается модификация I/3B/N алгоритма кодирования длин серий RLE для сжатия полутоновых изображений, основанная на учете вероятности повтора значений пикселей в строках. Алгоритм I/3B/N отличается от алгоритма I/2B/N использованием дополнительного символа  $b3(s)$ , который принимает значение 1, если символ  $I$  повторяется в

третий раз подряд, и принимает значение 0, если третий повтор символа  $I$  отсутствует. Структура закодированных данных в соответствии с алгоритмом I/3B/N приведена на рис. 3.

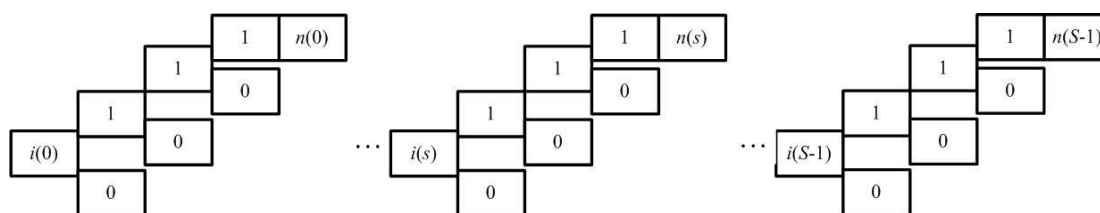


Рис. 3. Структура закодированных по алгоритму I/3B/N данных

Размер  $R_{I/3B/N}$  (бит) кода для алгоритма I/3B/N определяется с помощью выражения

$$R_{I/2B/N} = S(BD_I + 1) + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + BD_N \sum_{s=0}^{S-1} b3(s).$$

Вычислительная сложность алгоритма I/3B/N оценивается с помощью выражения

$$C_{I/2B/N} = YX + 4S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + \sum_{s=0}^{S-1} b3(s). \quad (2)$$

Выражение (2) учитывает  $YX$  операций на буферизацию и формирование таблицы  $I/N$  длин серий размером  $S$  записей,  $S$  операций на поиск максимального значения по  $I$ ,  $S$  операций на поиск максимального значения по  $N$ ,  $2S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + \sum_{s=0}^{S-1} b3(s)$  операций на кодирование и передачу.

### Другие модификации алгоритма кодирования длин

Для сжатия полутоновых изображений кроме алгоритмов I/2B/N и I/3B/N разработаны другие модифицированные алгоритмы кодирования длин серий, основанные на учете вероятности повтора значений пикселей в строках, отличающиеся использованием различного числа дополнительных символов для кодирования повторов символов (табл. 3).

Таблица 3. Модифицированные алгоритмы кодирования длин серий

№	Алгоритм	Особенность структуры закодированных данных
1	I/S/N	переменный размер поля длины серии
2	I/B/N	дополнительный символ повтора
3	I/B/S/N	дополнительный символ повтора и переменный размер поля длины серии
4	I/2B/N	два дополнительных символа повтора
5	I/2B/S/N	два дополнительных символа повтора и переменный размер поля длины серии
6	I/3B/N	три дополнительных символа повтора
7	I/3B/S/N	три дополнительных символа повтора и переменный размер поля длины серии
8	2I/N	повтор кодируемого символа
9	2I/S/N	повтор кодируемого символа и переменный размер поля длины серии
10	2I/B/N	повтор кодируемого символа и дополнительный символ повтора
11	2I/B/S/N	повтор кодируемого символа, дополнительный символ повтора и переменный размер поля длины серии
12	2I/2B/N	повтор кодируемого символа и два дополнительных символа повтора
13	2I/2B/S/N	повтор кодируемого символа, два дополнительных символа повтора и переменный размер поля длины серии
14	2I/B/2N(L/R)	повтор кодируемого символа, дополнительный символ повтора и двухсегментное поле длины серии

## Оценка эффективности использования алгоритмов кодирования длин серий для сжатия полутоновых изображений

Для тестовых изображений, представленных на рис. 4, в табл. 4 приведены размеры кода  $R_X$ , полученные для алгоритма RLE, алгоритмов из табл. 3, а также алгоритма Хаффмана, архиваторов Zip и Rar, где  $X$  – обозначение алгоритма в табл. 3.



Рис. 4. Тестовые изображения: *a* – M1 (128×128 пикселей);  
*б* – M2 (256×256 пикселей); *в* – M3 (512×12 пикселей)

Таблица 4. Размеры кода при сжатии тестовых изображений без потерь

Алгоритм	Размер кода (байт) для изображений			Алгоритм	Размер кода (байт) для изображений		
	M1	M2	M3		M1	M2	M3
RLE	18620	77896	242604	2I/S/N	34484	135364	261921
I/S/N	18739	72412	143162	2I/B/N	32599	127870	222123
I/B/N	21504	91397	303235	2I/B/S/N	32826	128820	225640
I/B/S/N	20957	81407	165901	2I/2B/N	32713	128345	223002
<b>I/2B/N</b>	14694	57583	106871	2I/2B/S/N	32825	128753	225460
I/2B/S/N	17689	68649	118177	2I/B/2N(L/R)	30046	118028	208692
<b>I/3B/N</b>	14577	56701	103512	Zip	13070	44603	76470
I/3B/S/N	17687	68581	117997	Rar	12505	43062	75887
2I/N	30714	120377	208857	Хаффмана	14361	58281	113091

Из табл. 4 следует, что минимальный размер кода обеспечивают алгоритмы Rar и Zip. По сравнению с ними алгоритм I/3B/N показывает примерно на 26 % худший результат, но превосходит на 57 % и 8 % алгоритмы RLE и Хаффмана соответственно. При этом алгоритмы RLE, I/2B/N, I/3B/N обеспечивают по сравнению с алгоритмами Rar и Zip уменьшение времени кодирования в 31,2 и 53,0 раз соответственно при размере изображения 256×256 пикселей, в 18,5 и 16,6 раз соответственно при размере изображения 1024×1024 пикселей, в 10,0 и 13,5 раз соответственно при размере изображения 8192×8192 пикселей (рис. 5).

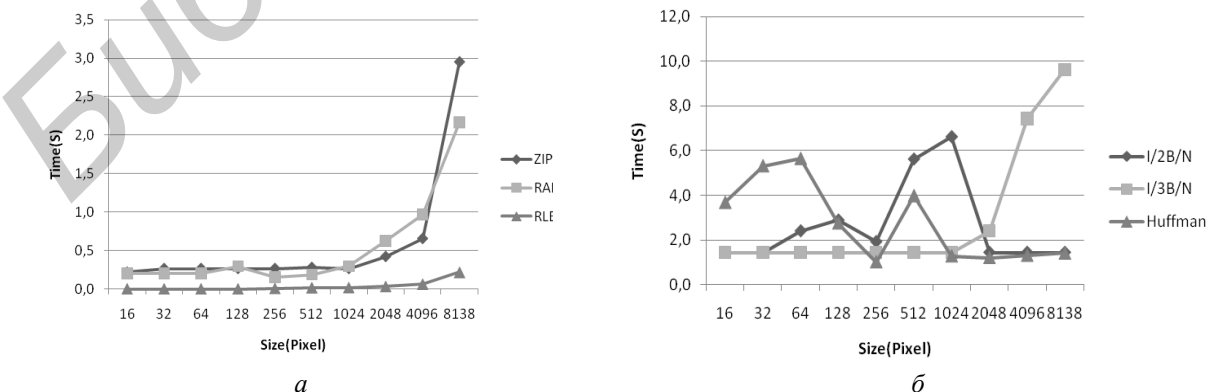


Рис. 5. Зависимости времени кодирования изображений от их размера:  
*a* – для алгоритмов Zip, Rar, RLE и его предложенных модификаций в операционной системе Windows 7;  
*б* – для алгоритмов Хаффмана, I/2B/N и I/3B/N в среде Matlab 2012

## Заключение

Предложены модифицированные алгоритмы кодирования длин серий для сжатия без потерь полутоновых изображений, отличающиеся от базового алгоритма RLE учетом вероятности повтора значений пикселей в строках изображения. Показано, что предложенные алгоритмы обеспечивают уменьшение размера закодированных данных на 57% и 8% по сравнению с алгоритмами RLE и Хаффмана соответственно; уменьшение времени кодирования в 10–31 раз и в 13–53 раза при изменении размера изображений от 256×256 пикселей до 8192×8192 пикселей по сравнению с алгоритмами Rar и Zip соответственно.

## LOSSLESS COMPRESSION OF THE HALFTONE IMAGES BASED ON RUN-LENGTH ENCODING

H.K. AL-BAHADILY, E.G. MAKEICHIK, V.Yu. TSVIATKOU, V.K. KANAPELKA

### Abstract

The modified algorithms length coding to compress grayscale images without losses that are different from the basic algorithm taking into account the likelihood of the occurrence of pixel values and images that reduce the size of the encoded data.

*Keywords:* images compression, run-length encoding.

### Список литературы

1. *Pennebaker W.B., Mitchell J.L.* JPEG Still Image Compression Standard. New York, 1993.
2. *Ebrahimi T.* // Proc. of the SPIE. San Diego, July–August 2000. Vol. 4115. P. 446–454.
3. *Ватолин Д.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М., 2003.
4. *Golomb S.W.* // IEEE Transactions on Information Theory. 1966. July. P. 399–401.