

ВЕРОЯТНОСТНОЕ КОДИРОВАНИЕ ДЛИН СЕРИЙ С ПОДТВЕРЖДЕНИЕМ ПОВТОРА И ПРЕДВАРИТЕЛЬНОЙ СОРТИРОВКОЙ ДЛЯ СЖАТИЯ БИТОВЫХ ПЛОСКОСТЕЙ ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЙ

Х.К. АЛЬ-БАХДИЛИ, В.Ю. ЦВЕТКОВ, В.К. КОНОПЕЛЬКО

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 4 июля 2017

Аннотация. Предложен алгоритм вероятностного кодирования длин серий с подтверждением повтора и предварительной сортировкой. Исследована эффективность использования данного алгоритма для сжатия без потерь битовых плоскостей полутоновых изображений.

Ключевые слова: сжатие изображений, кодирование длин серий, вероятностное кодирование, сортировка.

Abstract. It's proposed an algorithm for probabilistic coding of the lengths of series with confirmation of repetition and preliminary sorting. The efficiency of this algorithm using for compression without loss of bit planes of halftone images is investigated.

Keywords: image compression, run length encoding, probabilistic coding, sorting.

Doklady BGUIR. 2017, Vol. 109, No. 7, pp. 12-19

Probabilistic coding of length of series with confirmation

of repetition and preliminary sorting for compression of halftone images bits

H.K. Al-Bahadily, V.Yu. Tsviatkou, V.K. Kanapelka

Введение

Метод RLE (Run-Length Encoding) кодирования длин серий является самым быстрым среди методов сжатия [1–3]. Он ориентирован для обработки строк данных с высокой вероятностью повтора символа. Поэтому его непосредственное использование для сжатия изображений, имеющих высокую вероятность коротких серий пикселей вдоль строк или столбцов, не эффективно по сравнению с другими методами (RAR, ZIP [4]), особенно специально для этого предназначенными (JPEG [1], JPEG 2000 [2]). В [5] показано, что такая статистика характерна для ландшафтных, космических и других изображений, формируемых в реальных условиях. Причем, если для старших битовых плоскостей характерен большой разброс вероятностей повтора бит, то в младших битовых плоскостях доминируют короткие серии, делающие использование метода RLE не эффективным. Модификации алгоритма RLE, предложенные в работах [5, 6], позволяют повысить коэффициент сжатия изображений за счет учета разброса значений вероятностей повтора значений пикселей с сохранением высокой скорости обработки, однако они не всегда приводят к сокращению объема данных и существенно уступают в коэффициенте сжатия алгоритмам RAR, ZIP, JPEG, JPEG 2000. Повышение эффективности использования алгоритма RLE для сжатия изображений возможно за счет учета особенностей функций распределения вероятностей повтора значений бит для различных разрядов пикселей и предварительной сортировки битовых плоскостей, позволяющей повысить вероятность повтора значения бита.

Цель работы – разработка быстрого алгоритма сжатия полутоновых изображений без потерь на основе вероятностного кодирования длин серий бит с предварительной сортировкой.

Алгоритм вероятностного кодирования длин серий бит с подтверждением повтора и предварительной сортировкой

Для сжатия битовых плоскостей полутоновых изображений предлагается алгоритм XmVN вероятностного кодирования длин серий бит с подтверждением повтора и предварительной сортировкой, основанный на формировании для каждой серии m символов подтверждения повтора бита и символа паритета, относящего соответствующий фрагмент

следующей по убыванию значимости битовой плоскости в зависимости от преобладания нулевых или единичных бит к одному из двух классов – нулевому или единичному, кодируемых далее отдельно. Алгоритм XmBN отличается от базового алгоритма RLE использованием для кодирования каждой серии от 1 до m (в зависимости от длины серии) дополнительных символов подтверждения повтора бита, одного дополнительного символа паритета, распределением фрагментов нижней по значимости битовой плоскости по двум векторам в зависимости от значения символа паритета, отдельным кодированием каждого вектора. Алгоритм XmBN вероятностного кодирования длин серий бит изображения с предварительной сортировкой состоит из следующих шагов.

1. Кодирование старшей $(L-1)$ -й битовой плоскости, где L – число битовых плоскостей изображения (совпадает с разрядностью пикселей).

1.1. Текущая l -я битовая плоскость размером $Y \times X$ бит преобразуется в строку и фиксируется значение $i(l, p)$ ее первого бита, где p – номер строки (для старшей битовой плоскости, соответствующей старшим $(L-1)$ -м разрядам пикселей, предусмотрена одна строка $l = L-1$ и $p = 0$).

1.2. Полученная строка разделяется на $S(l, p)$ серий повторяющихся бит.

1.3. Для каждой s -й серии ($s = \overline{0, S(l, p) - 1}$) определяется проекция на нижнюю по значимости $(l-1)$ -ю битовую плоскость (для старшей битовой плоскости это $(L-2)$ -я битовая плоскость) и выделяется таким образом соответствующая s -я проективная серия.

1.4. Для каждой s -й проективной серии оценивается число нулей и единиц. Если число нулей в серии больше, чем число единиц, то соответствующая серия относится к нулевому классу, иначе – к единичному классу. В зависимости от того, к какому классу отнесена проективная серия, символ сортировки $\tilde{O}(l, p, s)$ соответствующей серии l -й битовой плоскости устанавливается в ноль или единицу.

1.5. Осуществляется вложенное кодирование каждой s -й серии l -й битовой плоскости. При этом если значение текущего s -го бита $i(l, s)$ серии повторяется r раз при $r \geq m$, то формируются m бит повтора $b(l, p, k, s) = 1$ и символ $n(l, p, s) = r - m$ длины серии, учитывающий число повторов текущего s -го бита $i(l, s)$ более чем m раз, где $k = \overline{1, m}$, $s = \overline{0, S(l, p) - 1}$. Если значение текущего s -го бита $i(l, s)$ повторяется r раз при $0 < r < m$, то формируются r бит повтора $b(l, p, k, s) = 1$ и один бит повтора $b(l, p, k+1, s) = 0$, где $k = \overline{1, r}$.

1.6. Формируется код старшей битовой плоскости в соответствии с одним из вариантов (при $m = \overline{0, 3}$):

$\{n(L-1, 0, s)\}$ при $m = 0$;

$\{b(L-1, 0, 1, s) = 0\}$, $\{b(L-1, 0, 1, s) = 1, n(L-1, 0, s)\}$ при $m = 1$;

$\{b(L-1, 0, 1, s) = 0\}$, $\{b(L-1, 0, 1, s) = 1, b(L-1, 0, 2, s) = 0\}$,

$\{b(L-1, 0, 1, s) = 1, b(L-1, 0, 2, s) = 1, n(L-1, 0, s)\}$ при $m = 2$;

$\{b(L-1, 0, 1, s) = 0\}$, $\{b(L-1, 0, 1, s) = 1, b(L-1, 0, 2, s) = 0\}$,

$\{b(L-1, 0, 1, s) = 1, b(L-1, 0, 2, s) = 1, b(L-1, 0, 3, s) = 0\}$,

$\{b(L-1, 0, 1, s) = 1, b(L-1, 0, 2, s) = 1, b(L-1, 0, 3, s) = 1, n(L-1, 0, s)\}$ при $m = 3$.

2. Кодирование средних l -х битовых плоскостей, где $l = \overline{0, L-2}$.

2.1. l -я битовая плоскость (соответствующая l -м разрядам пикселей) размером $Y \times X$ бит разделяется на $S(l+1)$ проективных серий и преобразуется в $P(l) = 2^{(L-1)-l}$ строк в результате

сортировки проективных серий в зависимости от значений соответствующих символов сортировки $X(l+1, p, s)$, где $(p = \overline{0, P(l)-1})$.

2.2. Каждая p -я из полученных в результате сортировки $P(l)$ строк $(p = \overline{0, P(l)-1})$, включающая $S(l, p)$ серий, кодируется по алгоритму 1.1 – 1.5 независимо.

2.3. Формируется код каждой p -й строки l -й битовой плоскости в соответствии с одним из вариантов (при $m = \overline{0, 3}$, $s = \overline{0, S(l, p)-1}$):

$\{X(l+1, u, z)\}$, $\{n(l, p, s)\}$ при $m=0$;
 $\{X(l+1, u, z)\}$, $\{b(l, p, 1, s)=0\}$, $\{b(l, p, 1, s)=1, n(l, p, s)\}$ при $m = 1$;
 $\{X(l+1, u, z)\}$, $\{b(l, p, 1, s)=0\}$, $\{b(l, p, 1, s)=1, b(l, p, 2, s)=0\}$,
 $\{b(l, p, 1, s)=1, b(l, p, 2, s)=1, n(l, p, s)\}$ при $m = 2$;
 $\{X(l+1, u, z)\}$, $\{b(l, p, 1, s)=0\}$, $\{b(l, p, 1, s)=1, b(l, p, 2, s)=0\}$,
 $\{b(l, p, 1, s)=1, b(l, p, 2, s)=1, b(l, p, 3, s)=0\}$, $\{b(l, p, 1, s)=1, b(l, p, 2, s)=1, b(l, p, 3, s)=1, n(l, p, s)\}$
при $m = 3$,

где $u = \lfloor p/2 \rfloor$; $z = \overline{0, S(l+1, \lfloor p/2 \rfloor)-1}$; $\lfloor \rfloor$ – операция округления с недостатком.

3. Кодирование младших r -х битовых плоскостей. Для младших битовых плоскостей изображений кодирование длин серий бит не эффективно. Поэтому младшие битовые плоскости преобразуются в строку без кодирования.

На рис. 1 приведена схема, иллюстрирующая процесс вероятностного кодирования длин серий бит с подтверждением повтора и предварительной сортировкой в соответствии с алгоритмом XmBN.

Размер R_{XmBN} (бит) кода для алгоритма XmBN (при $m = \overline{0, 3}$) определяется с помощью выражений:

$$\begin{aligned}
R_{\overline{0}BN} &= 1 + BD_N(L-1, 0) \cdot S(L-1, 0) + \\
&+ \sum_{l=0}^{L-2} \sum_{p=0}^{P(l)-1} (S(l, p) + 1 + BD_N(l, p) \cdot S(l, p)) \text{ и } \delta \text{ } m = 0, \\
R_{\overline{1}BN} &= 1 + S(L-1, 0) + BD_N(L-1, 0) \sum_{s=0}^{S(L-1, 0)-1} b(L-1, 0, 1, s) + \\
&+ \sum_{l=0}^{L-2} \sum_{p=0}^{P(l)-1} \left[1 + 2S(l, p) + BD_N(l, p) \sum_{s=0}^{S(l, p)-1} b(l, p, 1, s) \right] \text{ и } \delta \text{ } m = 1, \\
R_{\overline{2}BN} &= 1 + S(L-1, 0) + \sum_{s=0}^{S(L-1, 0)-1} b(L-1, 0, 1, s) + BD_N(L-1, 0) \sum_{s=0}^{S(L-1, 0)-1} b(L-1, 0, 2, s) + \\
&+ \sum_{l=0}^{L-2} \sum_{p=0}^{P(l)-1} \left(1 + 2S(l, p) + \sum_{s=0}^{S(l, p)-1} b(l, p, 1, s) + BD_N(l, p) \sum_{s=0}^{S(l, p)-1} b(l, p, 2, s) \right) \text{ и } \delta \text{ } m = 2, \\
R_{\overline{3}BN} &= 1 + S(L-1, 0) + \sum_{s=0}^{S(L-1, 0)-1} b(L-1, 0, 1, s) + \\
&+ \sum_{s=0}^{S(L-1, 0)-1} b(L-1, 0, 2, s) + BD_N(L-1, 0) \sum_{s=0}^{S(L-1, 0)-1} b(L-1, 0, 3, s) + \\
&+ \sum_{l=0}^{L-2} \sum_{p=0}^{P(l)-1} \left(1 + 2S(l, p) + \sum_{s=0}^{S(l, p)-1} b(l, p, 1, s) + \right. \\
&\left. + \sum_{s=0}^{S(l, p)-1} b(l, p, 2, s) + BD_N(l, p) \sum_{s=0}^{S(l, p)-1} b(l, p, 3, s) \right) \text{ и } \delta \text{ } m = 3,
\end{aligned}$$

где $BD_N(l, p) = \lceil \log_2 (\max(n(l, p, s))_{(s=0, S(l, p)-1)}) \rceil$ – число разрядов, выделяемых для представления символа $n(l, p, s)$, $\lceil \cdot \rceil$ – операция округления с избытком.

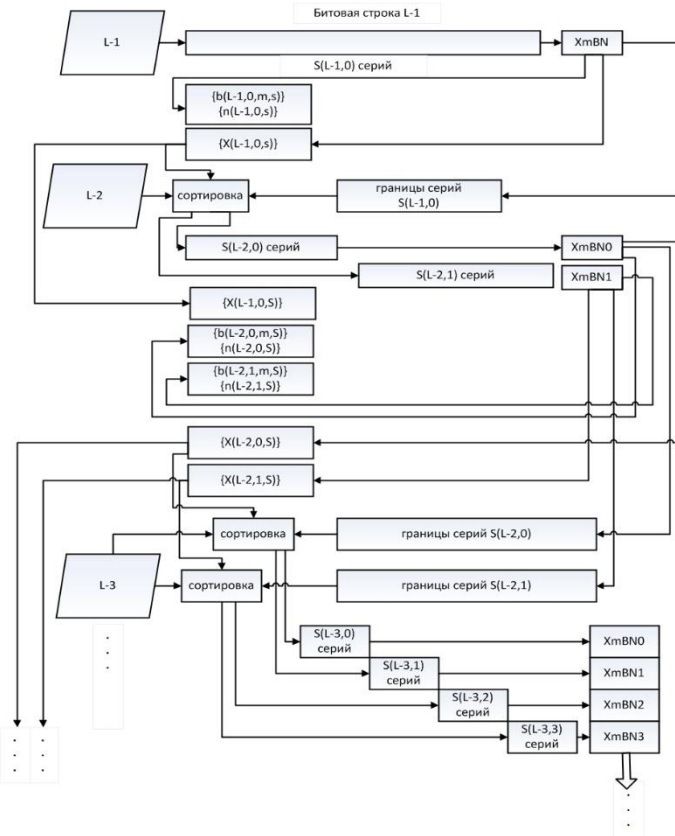


Рис. 1. Схема процесса вероятностного кодирования длин серий бит с подтверждением повтора и предварительной сортировкой по алгоритму XmBN

Вычислительная сложность C_{XmBN} алгоритма XmBN (при $m = \overline{0,3}$) оценивается с помощью выражений:

$$C_{X0BN}(L_0) = (L - L_0 + 3)YX + 11 \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l, p) \text{ при } m = 0,$$

$$C_{X1BN}(L_0) = (L - L_0 + 3)YX + 11 \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l, p) + 2 \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l, p)-1} b(l, p, 1, s) \text{ при } m = 1,$$

$$C_{X2BN}(L_0) = (L - L_0 + 3)YX + 11 \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l, p) + 2 \left(\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l, p)-1} b(l, p, 1, s) + \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l, p)-1} b(l, p, 2, s) \right) \text{ и } \text{д} \text{ } m = 2,$$

$$C_{X3BN}(L_0) = (L - L_0 + 3)YX + 11 \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l, p) + 2 \left(\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l, p)-1} b(l, p, 1, s) + \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l, p)-1} b(l, p, 2, s) + \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l, p)-1} b(l, p, 3, s) \right) \text{ и } \text{д} \text{ } m = 3,$$

где $(L - L_0 + 3)YX$ – число операций на буферизацию, формирование таблицы длин серий, включающей $\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l, p)$ записей, и сортировку; $11 \sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l, p)$ – число операций поиска

максимального значения по $n(l, p, s)$ (1 операция на серию), формирования и передачи бит подтверждения повтора и символов длин серий (2 операции на серию); определения значений бит

паритета $X(l+1, p, s)$ (1 операция на серию), границ серий (2 операции на серию), сортировки (5 операций на серию); L_0 – число старших битовых плоскостей, для которых алгоритм XmBN обеспечивает коэффициент сжатия больше единицы.

Оценка эффективности адаптивного вероятностного кодирования длин серий бит с предварительной сортировкой

В табл. 1 приведены коэффициенты сжатия тестовых изображений без потерь, представленных в работе [7], с помощью алгоритмов XmBN, mBN, ImBN [8] (при $m = 3$), RLE, RAR, ZIP.

Таблица 1. Коэффициенты сжатия тестовых изображений без потерь

Алгоритм	Битовая плоскость	Коэффициенты сжатия тестовых изображений							
		M1		M2		M3		M4	
		$R(q)$, бит	CR	$R(q)$, бит	CR	$R(q)$, бит	CR	$R(q)$, бит	CR
RLE	$q = 7$	225632	1,03	232974	1,01	289816	< 1	56580	1,19
	$q = 6$	242720		275778		422905		177366	
	$q = 5$	369964		309120		631659		213462	
	$q = 4$	602938		566797		670530		328833	
3BN	$q = 7$	180729	1,08	128078	1,19	220159	1,02	42949	1,38
	$q = 6$	191060		148324		285646		107312	
	$q = 5$	268028		170510		342267		141229	
	$q = 4$	335108		294340		330143		182134	
	$q = 3$	–		–		–		270390	
	$q = 2$	–		–		–		334980	
X3BN	$q = 7$	180729	1,23	128078	1,19	220159	1,04	42949	1,67
	$q = 6$	48351		61129		219057		91726	
	$q = 5$	163408		–		323470		102951	
	$q = 4$	303381		–		–		124243	
	$q = 3$	–		–		–		207565	
	$q = 2$	–		–		–		164790	
	$q = 1$	–		–		–		300233	
I3BN	–	–	< 1	–	< 1	–	< 1	–	1,10
ZIP	–	–	1,25	–	1,38	–	1,14	–	1,68
RAR	–	–	1,54	–	1,56	–	1,14	–	2,07

Из табл. 1 следует, что алгоритм XmBN эффективен для кодирования верхних (старших) l -х ($(L-1) \geq l > 0$) битовых плоскостей, для которых коэффициент сжатия больше единицы ($CR = YX/R_{\text{OmBN}}(l) > 1$). Причем он имеет различную эффективность для сжатия изображений, полученных с помощью видеокамеры и тепловизора. При сжатии изображений видимого диапазона алгоритм X3BN превосходит алгоритм RLE в 1,16 раза, алгоритм 3BN – в 1,05 раза, алгоритм I3BN – в 1,15 раза, но уступает в коэффициенте сжатия алгоритму RAR в 1,22 раза и алгоритму ZIP – в 1,1 раза (с усреднением по изображениям). Также из табл. 1 следует, что при сжатии тепловизионных изображений алгоритм X3BN превосходит алгоритм RLE в 1,4 раза, алгоритм 3BN – в 1,21 раза, алгоритм I3BN – в 1,52 раза, не уступает алгоритму ZIP и проигрывает алгоритму RAR в 1,24 раза.

Повышение коэффициента сжатия в алгоритме X3BN по сравнению с алгоритмом 3BN достигается за счет увеличения вычислительной сложности. В [8] показано, что вычислительная сложность алгоритма 3BN для одной l -й битовой плоскости определяется с помощью выражения

$$C_{3BN}(l) = YX + 3S(l) + 2 \left(\sum_{s=0}^{S(l)-1} b(l,1,s) + \sum_{s=0}^{S(l)-1} b(l,2,s) + \sum_{s=0}^{S(l)-1} b(l,3,s) \right), \text{ где } b(l,k,s) \text{ биты повтора } s\text{-й серии; } s=0, S(l)-1; S(l) \text{ – число серий.}$$

Таким образом, вычислительная сложность в алгоритме X3BN по сравнению с алгоритмом 3BN увеличивается в $C_{X3BN}(L-L_{X3BN}) / \left(\sum_{l=L-L_{3BN}}^{L-1} C_{3BN}(l) \right)$ раз, где L_{X3BN} , L_{3BN} – число старших битовых плоскостей, обеспечивающих коэффициент сжатия больше единицы для

соответствующего алгоритма. В табл. 2 приведены параметры кодирования алгоритма 3BN для старших битовых плоскостей тестовых изображений M1 ($L_{3BN} = 3$) и M4 ($L_{3BN} = 5$) размером 512×512 пикселей. Исходя из данных табл. 2, вычислительная сложность составляет для изображений M1 и M4 2659014 и 1783354 операций соответственно.

Таблица 2. Параметры кодирования тестовых изображений с помощью алгоритма 3BN

Тестовое изображение	$\sum_{l=L_{3BN}-2}^{L-1} S(l)$	$\sum_{l=L_{3BN}-2}^{L-1} \sum_{s=0}^{S(l)-1} b(l,1,s)$	$\sum_{l=L_{3BN}-2}^{L-1} \sum_{s=0}^{S(l)-1} b(l,2,s)$	$\sum_{l=L_{3BN}-2}^{L-1} \sum_{s=0}^{S(l)-1} b(l,3,s)$
M1	111396	82978	62842	50500
M2	139593	91397	62985	45408

В табл. 3 приведены параметры кодирования алгоритма X3BN для старших битовых плоскостей тестовых изображений M1 ($L_{X3BN} = 4$) и M4 ($L_{X3BN} = 7$) размером 512×512 пикселей. Исходя из данных табл. 3, вычислительная сложность составляет для изображений M1 и M4 4658121 и 5328578 операций соответственно. Таким образом, вычислительная сложность в алгоритме X3BN по сравнению с алгоритмом 3BN увеличивается в 1,75 и 3,0 раза (в среднем примерно в 2,5 раза).

Таблица 3. Параметры кодирования тестовых изображений с помощью алгоритма X3BN

Тестовое изображение	$\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} S(l,p)$	$\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l,p)-1} b(l,p,1,s)$	$\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l,p)-1} b(l,p,2,s)$	$\sum_{l=L_0}^{L-1} \sum_{p=0}^{P(l)-1} \sum_{s=0}^{S(l,p)-1} b(l,p,3,s)$
M1	409702	235964	155579	112914
M2	197255	117450	78992	56660

В табл. 4 приведено среднее время кодирования тестовых изображений размером 512×512 пикселей с помощью рассматриваемых алгоритмов на компьютере с процессором AMD A8 и тактовой частотой 3,1 ГГц. Время кодирования для алгоритма X3BN получено на основе данных для алгоритма 3BN, исходя из предположения о линейной зависимости скорости обработки от вычислительной сложности. Из табл. 4 следует, что алгоритм X3BN позволяет повысить скорость кодирования в 5,7 и 4,7 раз по сравнению с алгоритмами RAR и ZIP соответственно, но проигрывает в скорости в 2,5 раз алгоритму RLE.

Таблица 4. Время кодирования тестовых изображений 512×512 пикселей без потерь

Алгоритм	RLE	3BN	X3BN	I3BN	ZIP	RAR
Время кодирования, с	0,017	0,017	0,043	0,017	0,200	0,240

Заключение

Предложен алгоритм XmBN вероятностного кодирования длин серий бит с подтверждением повтора и предварительной сортировкой, отличающийся от базового алгоритма RLE использованием для кодирования каждой серии от 1 до m (в зависимости от длины серии) дополнительных символов подтверждения повтора бита, одного дополнительного символа паритета, распределением фрагментов нижней по значимости битовой плоскости по двум векторам в зависимости от значения символа паритета, отдельным кодированием каждого вектора. Данные отличия позволили повысить коэффициент сжатия по сравнению с алгоритмом RLE в среднем в 1,16 раза для изображений видимого диапазона и в 1,4 раза – для тепловизионных изображений за счет увеличения вычислительной сложности алгоритма и времени обработки в 2,5 раза. При этом выигрыш в коэффициенте сжатия для изображений видимого диапазона составил в среднем 1,22 и 1,1 раза по сравнению с алгоритмами RAR и ZIP соответственно. Установлено, что предложенный алгоритм более эффективен для сжатия тепловизионных изображений – выигрыш в коэффициенте сжатия по сравнению с алгоритмом RLE составляет 1,4 раза при проигрыше алгоритму RAR в 1,24 раза. При этом предложенный алгоритм не уступает в коэффициенте сжатия алгоритму ZIP.

Список литературы

1. Pennebaker W.B. Still Image Compression Standard. New York, 1993.
2. Ebrahimi T. JPEG 2000 still image coding versus other standards // Proc. of the SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII. USA, 2000. Vol. 4115. P. 446–454.

3. Golomb S.W. Run Length Encoding // IEEE Transactions on Information Theory. 1966. P 399–401.
4. Методы сжатия данных. Устройство архиваторов / Д. Ватолин [и др.]. М.: ДИАЛОГ-МИФИ, 2002. 384 с.
5. Grayscale Image Compression using Bit Plane Slicing and Developed RLE Algorithms / Albahadily H. [et al.] // International Journal of Advanced Research in Computer (IJARCCE). 2017. Vol. 6, No. 2. P 309–314.
6. Аль Бахдили Х. Сжатие изображений дистанционного зондирования Земли на основе вероятностного кодирования длин серий пикселей // Доклады БГУИР. 2017. № 1 (103). С. 48–54.
7. Аль-Бахдили Х.К., Цветков В.Ю., Конопелько В.К. Сжатие битовых плоскостей полутонных изображений посредством вероятностного кодирования длин серий бит с подтверждением повтора // Доклады БГУИР. 2017. № 6 (108). С. 65–69.
8. New Modified RLE Algorithms to Compress Grayscale Images with Lossy and Lossless Compression / H. Albahadily [et al.] // International Journal of Computer Science (IJACSA). 2016. Vol. 7, No. 7. P 250–255.

References

1. Pennebaker W.B. Still Image Compression Standard. New York, 1993.
2. Ebrahimi T. JPEG 2000 still image coding versus other standards // Proc. of the SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII. USA, 2000. Vol. 4115. P. 446–454.
3. Golomb S.W. Run Length Encoding // IEEE Transactions on Information Theory. 1966. P 399–401.
4. Metody szhatija dannyh. Ustrojstvo arhivatorov / D. Vatin [i dr.]. M.: DIALOG-MIFI, 2002. 384 s. (in Russ.)
5. Grayscale Image Compression using Bit Plane Slicing and Developed RLE Algorithms / Albahadily H. [et al.] // International Journal of Advanced Research in Computer (IJARCCE). 2017. Vol. 6, No. 2. P 309–314.
6. Al' Bahdily H. Szhatie izobrazhenij distancionnogo zondirovaniya Zemli na osnove verojatnostnogo kodirovaniya dlin serij pikselej // Doklady BGUIR. 2017. № 1 (103). S. 48–54. (in Russ.)
7. Al'-Bahdili H.K., Cvetkov V.Ju., Konopel'ko V.K. Szhatie bitovyh ploskostej polutonovyh izobrazhenij posredstvom verojatnostnogo kodirovaniya dlin serij bit s podtverzheniem povtora // Doklady BGUIR. 2017. № 6 (108). S. 65–69. (in Russ.)
8. New Modified RLE Algorithms to Compress Grayscale Images with Lossy and Lossless Compression / H. Albahadily [et al.] // International Journal of Computer Science (IJACSA). 2016. Vol. 7, No. 7. P 250–255.

Сведения об авторах

Аль-Бахдили Х.К., аспирант кафедры инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники.

Цветков В.Ю., д.т.н., доцент, заведующий кафедрой инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники.

Конопелько В.К., д.т.н., профессор, профессор кафедры инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники.

Information about the authors

Al-Bahadily H.K., PG student of department of infocommunication technologies of Belarusian state university of informatics and radioelectronics.

Tsviatkou V.Yu., D.Sci., associate professor, chief of department of infocommunication technologies of Belarusian state university of informatics and radioelectronics.

Kanapelka V.K., D.Sci., professor, professor of department of infocommunication technologies of Belarusian state university of informatics and radioelectronics.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, д. 6,
Белорусский государственный
университет информатики и радиоэлектроники
тел. +375-17-293-84-08;
e-mail: vtsvet@bsuir.by
Цветков Виктор Юрьевич

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovka st., 6,
Belarusian state university
of informatics and radioelectronics
tel. +375-17-293-84-08;
e-mail: vtsvet@bsuir.by
Tsviatkou Victor Yur'evich