UDC 004.62 https://doi.org/10.20535/2786-8729.4.2024.292118

SOME ASPECTS OF IMPROVEMENT OF THE RUN LENGTH ENCODING COMPRESSION METHOD

Viktor Poriev
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-5250-1454

The article analyzes the possibilities of further improvement of the RLE compression method. An extended set of code sequences is offered to increase compression. It is proposed to improve compression by automated search for optimal code parameters for individual pieces of data. Bitmap encoding packaging methods for 4, 8, 16, and 24 bit per pixel formats are proposed. Experimental studies based on synthetic tests for compression of high-resolution raster were performed to compare the proposed coding methods with known implementations of the RLE method.

Keywords*:* compression methods, raster images, run length encoding.

1. Introduction

The information compression method called "*Run Length Encoding*" (RLE) has been known for a long time. This is a very simple compression method, in which each sequence of identical values is encoded by a pair (number of repetitions, value). This method has gained wide popularity for recording images in various file formats. The main known implementations of the RLE method are the PackBits method used in TIFF, TGA, and other formats [1], as well as a version of the RLE method for the PCX file format [2].

Despite the age of the RLE method, it continues to be used as an element of solutions in various fields of data storage and transmission [3].

The main advantages of RLE

– in simple implementations of this method the highest packing speed is achieved;

– simplicity and high speed of unpacking (decoding);

– in simple implementations of the RLE method no additional memory is required (for example, for a dictionary)

The disadvantage of known implementations of the RLE method is a lower degree of compression than dictionary LZ-like data compression methods, in particular the LZW method, which is used in various raster image formats [4].

However, not everything is so clear-cut. if the task is to save images of sufficiently large sizes, for example, rasters of the order of tens and hundreds of thousands of pixels horizontally and vertically, then it may be necessary to provide quick direct access to individual image fragments. But dictionary LZxx methods provide a high degree of compression after the accumulation of previously found code sequences in the dictionary – therefore when archiving small independent blocks for quick direct access, the compression may be even lower than in RLE.

One of the advantages of the RLE method over dictionary compression methods is that no decoding history (dictionary) needs to be accumulated. This allows encoding independent raster fragments without loss of compression.

The ability to independently encode individual rows, columns, or other blocks creates the prerequisites for:

– the ability to organize fast direct access to any parts of the image without unpacking previous blocks;

– parallel (multi-threaded) organization of encoding-decoding.

Such capabilities are useful in applications that can use raster data of significant sizes, in particular, in geographic information systems [5].

It would be desirable to find out solution that would allow increasing the degree of RLE compression while preserving the main advantages of this method. The further search for a solution for high-speed decoders is considered relevant.

2. Literature review and problem statement

In [6] a modification of RLE coding by modifying codewords is proposed, but this is only for binary images.

To increase the level of lossless image compression, some authors suggested combining certain compression methods with Huffman coding [7]. So, in particular, in [8] it is proposed to first use Huffman coding and then to process the result using the RLE method. However, the increase in compression was not very significant.

It seems, that may be more effective to use prefix codes instead of RLE code pairs (number of repetitions, values), which start with a special prefix and then contain codes for the number of repetitions and a color index.The prefix can take into account the popularity of a certain code according to Huffman or it can mean the type of code sequence.

For the sake of completeness, several special combinations of the RLE method with other methods can be pointed out. In [9] proposed combining RLE with adaptive arithmetic coding for video data compression.

To improve RLE compression, in [5] proposed special prefix codes for representing color values in raster images, codes for the length of pixel chains, as well as independent encoding of individual raster fragments with optimal codes for these fragments. This version of the RLE compression method is called RLE-BP. The developed RLE-BP adaptive coder made it possible to increase compression by 1.5–2 times compared to PackBits, and PCX implementations while maintaining a high decompression speed. This made it possible to compete with more powerful dictionary LZ-like compression methods when implementing a geoinformation system [5].

3. The aim and objectives of the study

The aim of this study is to increase the compression of the RLE method. The main provisions regarding increasing the degree of compression of image coding by the RLE_BP method were formulated in [10]. It was recommended to achieve this by automatically adjusting the parameters of codewords for individual fragments of images, and several methods of encoding with variable parameters were proposed. And all this for palette images. Let's consider some aspects of coding images.

4. The study materials and methods research of raster's packaging

4.1. The object and hypothesis of the study

The object of the proposed research is a modified RLE raster's packaging.

For a raster format, the main characteristic is the color depth or number of bits per pixel (B), which ensures the use of a maximum of $C = 2^B$ colors. For example, 8 bits per pixel has a maximum of 256 colors. Palette images are usually images with no more than 256 colors, since a table is created for them, each row of which contains a triplet of values, for example, RGB. The row number of such a table is the color index, and such a table itself is called a palette. Sometimes not all colors of the palette may actually be used in specific images. To take this fact into account, let's denote the number of bits representing the actual number of used colors by M. Maybe M ≤ B. At the beginning of the packaging, the RLE_BP encoder sorts the actually used colors of the palette of a particular image in the order of their popularity. The most popular color gets an index of 0, the next most popular – 1 and so on. Indices of all colors are coded with binary codes – let's denote them as c..c. In addition, codes for a certain number ($C1$) of the most popular colors can be separately distinguished – let's call them the main colors. A simple binary code for the main colors can consist of $M1$ bits.

Packaging method 1. Three types of codewords are used.

0c...c (a total of M bits) – for single pixels in which the most significant color bit is 0. Single pixels of colors 1c...c must be coded already as chains of length 1 using codeword with prefix 11.

The following codewords are used to code pixel chains:

10n...nc...c – first the prefix (bits 10), then $N1$ bits of the chain length code (bits n). The codeword is completed by $M1$ bits c, which means the index of the main color. This codeword is used for pixel chains of the most popular colors in a given image.

$M1$ is the number of bits of the main colors and $M1 < M$. For example, if only one color is used as the main color, then $M1=0$, and the codewords for the pixel chains of this color have the form 10n...n

11n...nc...c – first prefix (11), then $N2$ bits of chain length, and then M bits of color. $N2$ is the number of bits of the maximum length of pixel chains of any color.

The parameters of this packing method are the values of M, $M1$, $N1$, and $N2$:

Packaging method 2. Codewords of two types are used. Unlike the previous method, the coding of single pixels is performed here with no more than (M+1)-bit code regardless of color. This takes into account the probability of a large number of single pixels of secondary colors.

0c...c – for single pixels of any color. First, a prefix (0) followed by M color bits.

1c...cnn...n – for main color pixel chains. First the prefix (1), then $M1$ bits of the index of the main color (that is, only 2^{M1} of the most popular colors can be encoded this way). The codeword is completed by the chain length bits (nn...n). Each i-th main color corresponds to its maximum chain length and Ni number of bits n. Below is an example of encoding the main color chains for $M1 = 2$.

100nnn – $N0 = 3$ for the most popular color (index cc=00);

101nnnnnnn – $N1 = 7$ for the main color with index cc=01;

110nnnn – $N2 = 4$ for the main color with index cc=10;

111nnnnn – $N3 = 5$ for main color with index cc=11.

The characteristics of packaging method 2 are determined by the set of parameters M, $M1$, and Ni.

In method 2, the number of initial bits (1c...c) is the same for chain codes of all main colors. In the following method, it is proposed to encode colors with prefixes of different numbers of bits, and this number is the inverse of the popularity of the color. It can be said that it is a combination of RLE and the Huffman method.

Packaging method 3. Codewords of two types are used.

0c...c – for single pixels of any color. First prefix (0), then M color bits.

1p...pnn...n – for pixel chains. First the prefix (1p...p), then the chain length bits (nn...n). Each i-th main color corresponds to its own prefix and individual number (Ni) of bits n. Prefixes according to the Huffman tree in relation to the popularity of the main colors. An example of a set of codewords for the four main colors (i.e. C1=4)

10nn...n – chain of main color 0 ($N0$ bits n);

110nn...n – chain of main color 1 ($N1$ bits n);

1110nn...n – chain of main color 2 ($N2$ n bits);

1111nn...n is a chain of main color 3 ($N3$ bits of n).

In this example, the length of the chain code for color 0 is shorter than for method 2, however, for colors 2 and 3, the chain codes are longer. The parameters of packaging method 3 are the M, $C1$, and the set of Ni values. The following method provides advanced capabilities for encoding chains of repeating pixels for a wide range of lengths of such chains.

Packaging method 4. Code sequences of two types are used.

0c...c – for single pixels of any color. First prefix (0), then M color bits.

1c...cxx...x – for main color pixel chains. First the prefix (1), then $M1$ bits for the main color index. The code sequence is completed by the chain length bits (xx...x). Each value of the index of the main color (bits c...c) corresponds to its format of the chain length code, which is chosen from the following three formats (a, b, c):

format a:

1c...cnn...n – $N1$ bits of n chain length ($N1$ from 0 to 15);

format b:

1c...c0nn...n – $N1$ bits of n ($N1$ from 0 to 15),

1c...c1nn...n – $N2$ bits of n ($N2$ from $N1$+1 to $N1$+16);

format c:

1c...c0nn...n – $N1$ bits of n ($N1$ from 0 to 7);

1c...c10nn...n – $N2$ bits of n ($N2$ from $N1$+1 to $N1$+8);

1c...c11nn...n – $N3$ bits of n ($N3$ from $N2$+1 to $N2$+8).

The code parameters for method 4 are the values of $C1$, $N1$, $N2$, $N3$, and the length code format type.

Theoretically, varieties of code formats in the form of 1c...cy..ynn...n can be developed and expanded by increasing the number of bits y..y. But in practice, the presence of three formats (a, b, c) for method 4 may be quite enough to ensure efficient encoding of all possible chains with lengths of up to tens of thousands of pixels.

For each line (or column) of the raster, the RLE_BP coder finds the optimal values of parameters $C1$, and Ni at which the minimum bit of the packed code is reached.

Further development of the RLE_BP method is to provide the ability to encode 4-bit (16 colors), 16-bit (HighColor), and 24-bit (TrueColor) rasters. To provide image compression capabilities in these formats, it is proposed to add the following packaging methods (5–7) to the set of RLE_BP methods

Packaging method 5. This packaging method is similar to method 1, but the encoding of single pixels is slightly different.

Three types of codewords are used:

0 c..c (a total of 1+B bits) – for single pixels of any color (c..c)

10 n..n c..c (a total of 2 + $N1$ + $M1$ bits) – first the prefix (bits 10), then $N1$ bits of the chain length code (bits n). The codeword is completed by $M1$ bits c, which means the index of the main color. This codeword is used for pixel chains of the most popular colors in a given image.

11 n..n c..c (a total of 2 + $N2$ + B bits) – first prefix (11), then $N2$ bits of chain length, and then B bits of any color.

Packaging method 6. Two types of codewords are used:

0 c..c (a total of 1+$M1$ bits) – for a single pixel of the main color

10 n..n c..c (a total of 2 + $N1$ + $M1$ bits) – first the prefix (bits 10), then $N1$ bits of the chain length code (bits n). The codeword is completed by $M1$ bits c, which means the index of the main color.

11 n..n c..c (a total of 2 + $N2$ + B bits) – first prefix (11), then $N2$ bits of chain length, and then B bits of any color. For a single pixel $N2$=0 and the codeword has the form 11 c..c

Packaging method 7. Two types of codewords are used:

0 n..n c..c (a total of 1+$N1$+$M1$) – for a single pixel or chain of pixels of the main color. For a single pixel, $N1$=0 and the codeword has the form 0 c..c

1 n..n c..c (a total of 1+$N2$+B bits) – for a single pixel or a chain of pixels of any color. For a single pixel $N2$=0 and the codeword has the form 1 c..c

For all packaging methods 5–7, the following value ranges are recommended for the code length parameters B, $M1$, $N1$, and $N2$:

B = 4 (16 colors), 16 (HighColor), 24 (TrueColor) – color depth of the raster

$M1$ = 1...8 – number of bits to represent the main color index

$N1$ = 0...10 – the number of bits to represent the length of the main color chain

$N2$ = 0...5 – the number of bits to represent the length of a chain of any color

To see further possible areas of improvement, it is necessary to make some generalizations and create an appropriate model for the analysis of RLE coding.

4.2. The generalized model of the RLE codewords

In general, it seems that for all known varieties of the RLE method, the following coding structure is used. Suppose that some packed bitstream contains codewords (Fig. 1).

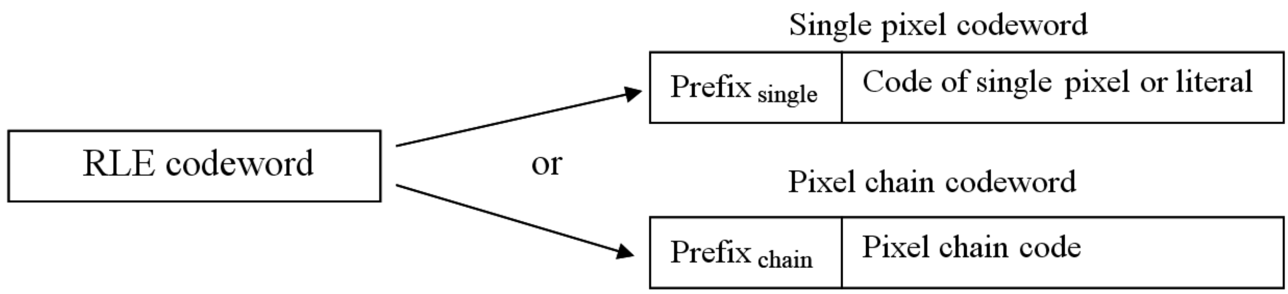

Fig. 1. Simplified traditional RLE codeword structure

The main purpose of the prefixes here is to distinguish between codewords for single pixels and pixel chain codes. For example, in the RLE PackBits method, both prefixes are single-bit. The first bit 0 of each codeword is an indication for $Prefix_{single}$ and that the code of a single pixel or the codes of a set of unique pixels (literal) is written next. If the first bit = 1, then it indicates $Prefix_{chain}$, and then there is a chain code of the same pixels.

The RLE PCX method uses two-bit prefixes and their role is a bit more complicated. In some single pixels, $Prefix_{single}$ = 00, 01, 10 simultaneously act as the two uppermost bits of the color code, and the value of the prefix for the chain codeword $Prefix_{chain}$ = 11.

The RLE_BP method of the current revision uses the one-bit $Prefix_{single}$ = 0 for single pixels. This bit can also identify certain color indices. The value 1 of the first bit of the codeword is a sign of pixel chain coding. In other words $Prefixc_{hain}$ bits = 1p..p. The number of bits ‘p’ can be different for different codewords of chains. In addition, the lower-order bits of the chain prefix can act as color index bits.

The code of a single pixel can be a normal binary color code, or it can be in the form of a prefix code, for example, Huffman. In order not to build a full Huffman tree, the color indices can be divided into intervals. Accordingly, there can be a combination of prefix and regular codes, for example, this:

0 c. . . c – m1 bits of c (in the 1st interval)
10 c. . . c – m2 bits of c (in the 2nd interval)
110 c. . . c – m3 bits of c (in the 3rd interval)
1110 c. . . c – m4 bits of c (in the 4th interval)
1111 c. . . c – m5 bits of c (in the 5th interval)

where: c. . . c are the bits of the normal binary code of the color indices in the intervals.

It is possible to propose to improve in this way, in particular, the Packaging method 1 considered above in the part of encoding single pixels – thereby leaving the format of code words for pixel chains. Let's call it Packaging method 8. Such an improvement of the method allows in certain cases to slightly improve the RLE compression of images with many single pixels.

Now about the pixel chain codeword formats. Such a codeword must somehow contain bits identifying the length of the chain (n) and bits of the pixel color (c). For example, for the RLE PackBits, method the 16-bit pixel string codeword is '1nnnnnnncccccccc', and for RLE PCX it is '11nnnnnncccccccc'. The sequence of elements in the codeword is fixed in the format: (prefix – length – color). But, as it seems, the sequence of elements can be different, and more flexible. So, for example, the packaging methods considered above, which are part of RLE_BP, can be described by at least two types of formats: (prefix – length – color) and also (prefix – color – length). So, for example, the codeword for packing method 1 has the format 'ppn..nc..c', 'pc...cnn...n' for method 2, and ‘1p...pnn...n’ for method 3. Moreover, the prefix code bits (p) and the color code bits (c) can be distributed in the codeword, for example, in the form of ‘pc...cppnn...n’ in packing method 4. Also, the color and/or length codes can, in turn, be represented by prefix codes. This is done, in particular, in packing method 2 – color indices are represented by prefix codes according to Huffman (Fig. 2).

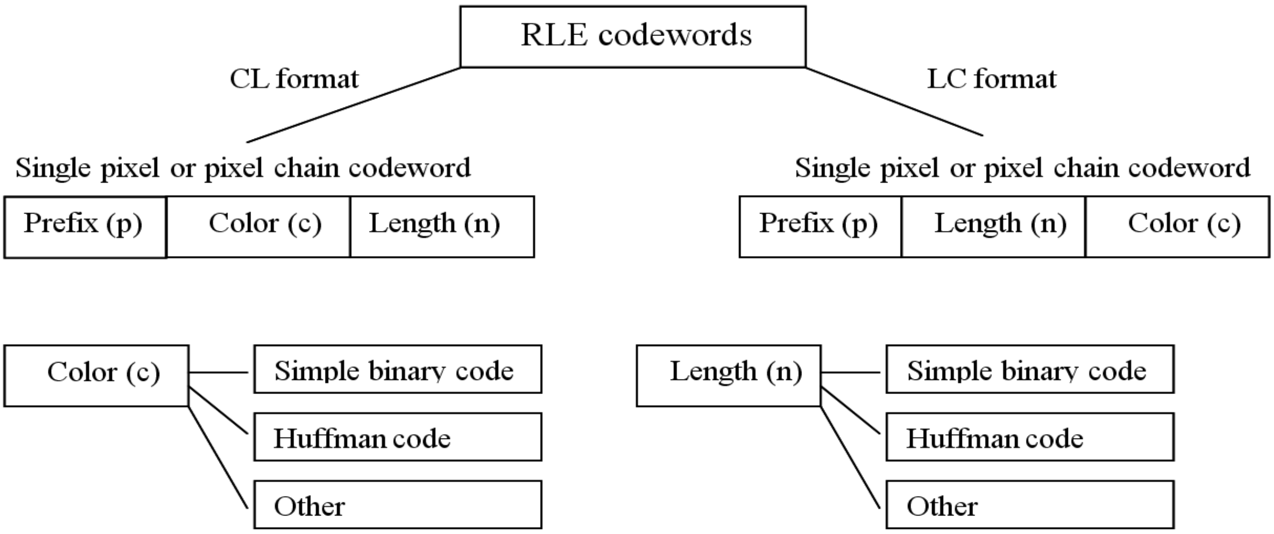

Fig. 2. Generalized structure of RLE codewords

Let's make some general estimates about the number of bits needed to encode a particular image. If each pixel is encoded separately, without taking into account the relationship to some neighboring pixels, and if all colors are encoded with the same number of bits, then

$$B_{bitmap\ total} = B \cdot V_{pixels\ total}, \qquad (1)$$

where B is the number of bits per pixel (color depth), $V_{pixels\ total}$ is the total number of all pixels of the raster (bitmap). Obviously, $V_{pixels\ total}$ = width × height of the raster. For example, a 256-color ($B = 8$) image with dimensions of 1000×2000 pixels requires 8×1000×2000 = 16000000 bits.

If pixels of different colors are coded differently, then the total number of bits needed to code the image (B_{code}) is usually no longer equal to B_{bitmap} in (1)

$$B_{code\ total} = \sum_{c} B(c) \cdot V(c), \qquad (2)$$

where $B(c)$ is the number of bits to represent color c, $V(c)$ is the number of pixels of this color in a specific image.

When coding using the RLE method, the total number of code bits in (2) consists of the number of code bits of all available single pixels ($B_{single\ total}$) and the number of code bits of available chains ($B_{chains\ total}$) in a specific image

$$B_{RLE\ total} = B_{single\ total} + B_{chains\ total}, \qquad (3)$$

If we clearly distinguish the dependence of the number of bits on colors, then

$$B_{RLE\ total} = \sum_{c} B_{single}(c) \cdot V_{single}(c) + \sum_{c} B_{chain}(c) \cdot V_{chain}(c), \qquad (4)$$

or, which is identical to the following

$$B_{RLE\ total} = \sum_{c} \left(B_{single}(c) \cdot V_{single}(c) + B_{chain}(c) \cdot V_{chain}(c)\right), \qquad (5)$$

where $B_{single}(c)$ is the number of bits for a single pixel of color c, $B_{chain}(c)$ is the number of bits for the pixel chain code of this color, $V_{single}(c)$ is the number of single pixels, and $V_{chain}(c)$ is the number of chains in a particular image.

And now about the sorting of colors (or their indices). If in a certain image, there are more pixels of some colors than others, then it is advisable to assign values of color indices in the order of decreasing popularity. For example, assign index 0 to the most popular color, index 1 to the next less popular color, and so on – the least popular color in the binary code will receive index cmax. What does it give? If the color indices are coded with binary codes, in which the number of bits decreases for indices of more popular colors and (or) increases for indices of less popular ones, the total amount of the code may decrease if the distribution of pixels by color is significantly uneven. This is the basic idea of Huffman coding. The RLE PCX method has a somewhat similar approach, which consists in the fact that single color pixels with indices 00cccccc, 01cccccc and 10cccccc are coded with one byte, and single pixels with color indices 11cccccc are coded with two bytes – as strings of unit length. In the latter case, color sorting is required to prevent a significant increase (in the worst case by a factor of two) of the code size when having a two-bit prefix to indicate pixel chain codes. Thus, in the general case, the functions $B_{single}(c)$ and $B_{chain}(c)$ can vary in a sufficiently wide range depending on the color index c.

To characterize the methods of encoding single pixels and the presence of such pixels in specific images, the concept of the average number of bits per single pixel ($B_{one\_single\_pixel}$) can be introduced

$$B_{one\ single\ pixel} = \frac{1}{V_{single\ pixels\ total}} \sum_{c} B_{single}(c) \cdot V_{single}(c), \qquad (6)$$

where $V_{single\ pixels\ total}$ is the total number of pixels that are coded exactly as single pixels.

Similarly, the average number of RLE codeword bits per pixel chain can also be considered

$$B_{one\ chain\ pixel} = \frac{1}{V_{pixels\ in\ chains\ total}} \sum_{c} B_{chain}(c) \cdot V_{chain}(c), \qquad (7)$$

where $V_{pixels\ in\ chains\ total}$ is the total number of pixels that form chains.

The total number of bits for RLE encoding the entire raster based on such averaging can be written as follows

$$B_{RLE\ total} = B_{onesingle\ pixel} \cdot V_{single\ pixels\ total} + B_{one\ chain\ pixel} \cdot V_{pixels\ in\ chains\ total}. \qquad (8)$$

For compression, it is necessary that $B_{RLE\ total} < B_{bitmap\ total}$, i.e

$$B_{onesingle\ pixel} \cdot V_{single\ pixels\ total} + B_{one\ chain\ pixel} \cdot V_{pixels\ in\ chains\ total} < B \cdot V_{pixels\ total}. \qquad (9)$$

The presence of a prefix in the RLE codeword, which is intended to distinguish single pixel codes from chain codes, means that the average number of bits per single pixel is usually greater than the color depth (B).So, for example, when encoding images with 8 bits per pixel using the RLE PCX Bone single pixel method in the range 8×(1..2), and for the PackBits method $B_{onesingle\ pixel}$= 8×(129/128 .. 2). A reduction in the average number of bits per single pixel $B_{onesingle\ pixel} < B$ may be achieved by Huffman coding, but this can only happen if one color is significantly more popular than others. But, if the pixels of the most popular colors form chains, then the main effect of compression is already achieved by methods of encoding the chains themselves.

5. Results of investigations of raster's packaging

To assess the possible increase in the degree of compression of the proposed methods, comparative tests were conducted on specific image samples. Both real images and specially synthesized tests should be used for testing. As the simplest tests for evaluating the capabilities of different RLE-encoding methods, it is possible to recommend images of chains of pixels of the same color compatible with a plurality of single pixels of different colors. Let's consider some of these tests.

Test 1. A set of pixel chains of the same length for all colors (Fig. 3).

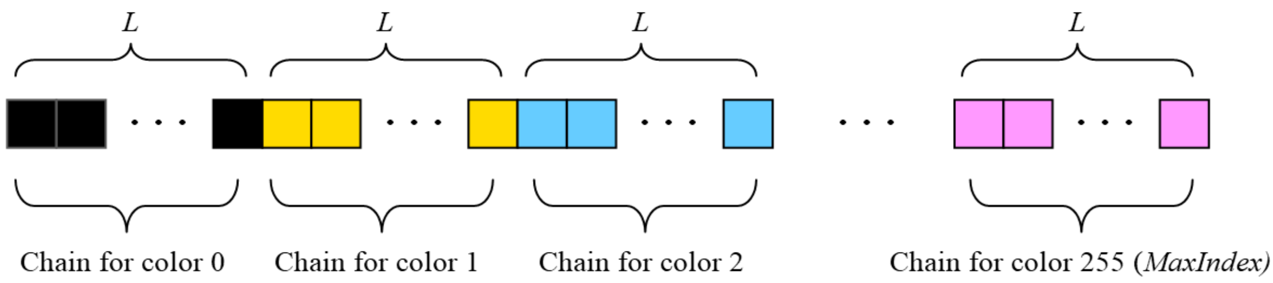

Fig. 3. One line raster for Test 1

Next, the software application – the test generator creates sequentially lines of appropriate length in the range L from 1 to 1000 and encodes these lines by several different methods. The compression ratio values are shown below in Fig 4.

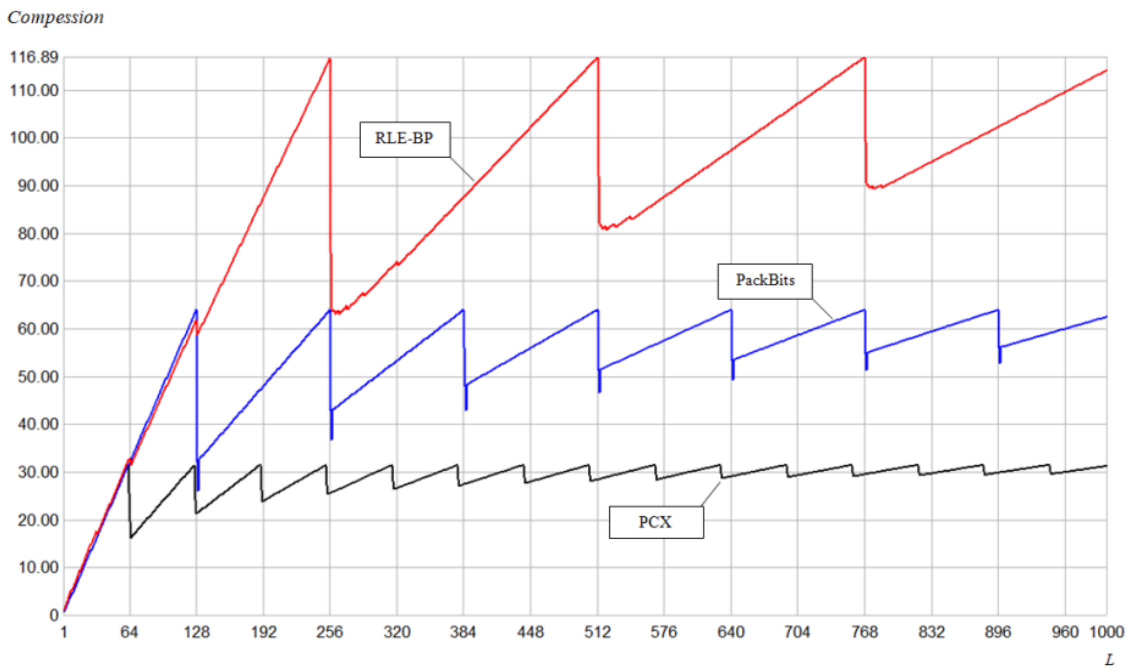

Fig. 4. Comparison of the degree of compression of known RLE methods (PackBits, PCX) and the proposed RLE_BP in Test 1

In Test 1, RLE_BP shows higher compression compared to the PackBits method when processing strings longer than 128 pixels. And compared to PCX, the gains start at chain sizes above 64.

Test 2. Each line contains chains of length from 1 to L. To the left of each chain is a single black pixel (Fig. 5, 6).

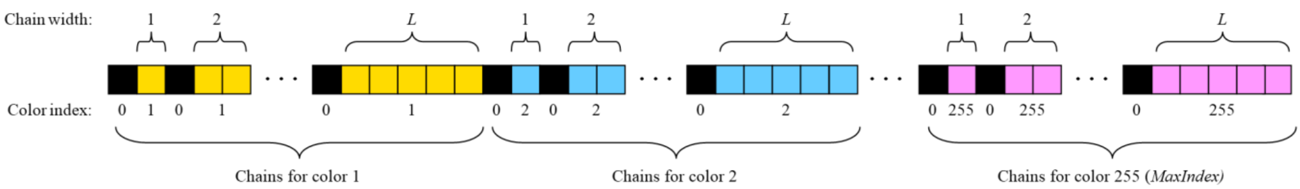

Fig. 5. One line raster for Test 2.

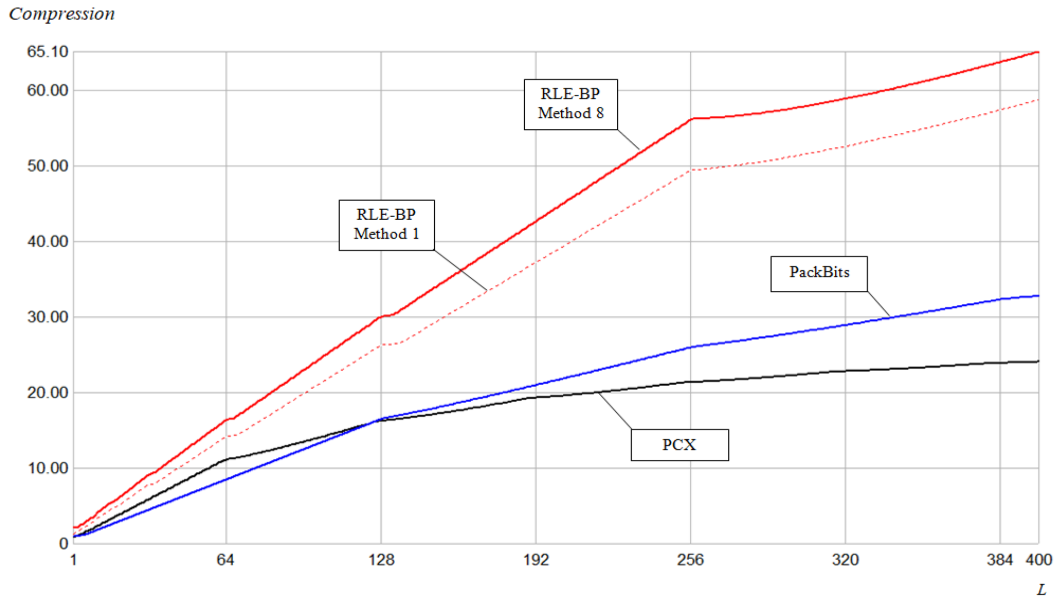

Fig. 6. Comparison of the degree of compression of known RLE methods (PackBits, PCX) and the proposed RLE_BP packaging methods 1 and 8 in Test 2

The execution of Test 2 demonstrates the advantages of RLE_BP packaging methods over PackBits and PCX in terms of greater perfection in encoding single pixels and chains of different lengths written together in each line of the raster. So, in particular, the packaging method 8 proposed in this work demonstrates a further increase in compression not only compared to PackBits and PCX, but also prevails over the packaging method 1 from the RLE_BP set.

Test 3. This test is similar to test 2, but here single pixels of all colors are present (Fig. 7, 8).

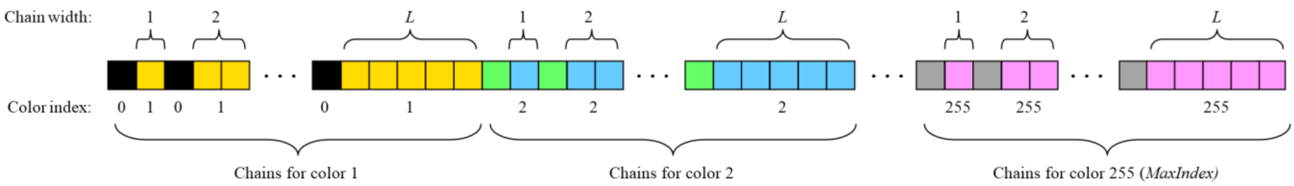

Fig. 7. One line raster for Test 3

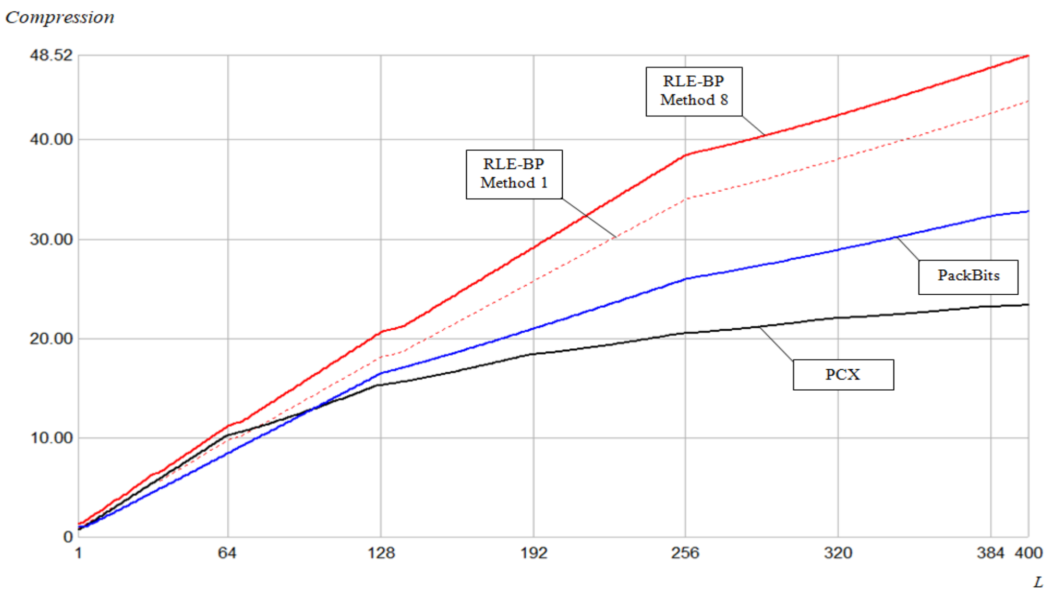

Fig. 8. Comparison of the degree of compression of known RLE methods (PackBits, PCX) and the proposed RLE_BP packaging methods 1 and 8 in Test 3

Test 4. Modeling the uneven distribution of chain lengths for several main colors. In the test, four chains are formed for each main color: (LW_i), ($2LW_i$), ($3LW_i$), ($4LW_i$), where W_i is the chain length coefficient for the i-th main color. Also, in the test, a set of single pixels is added to the chains. An example of the structure of the test line of the raster for the three main colors (i.e. $C1$=3) is shown below in Fig 9.

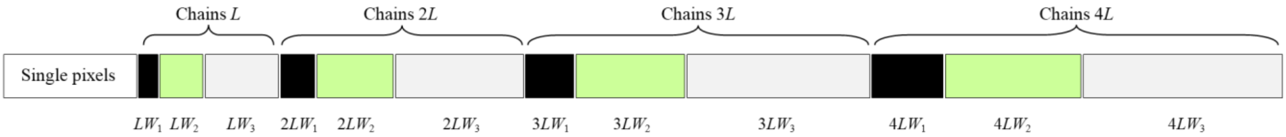

Fig. 9. An example of one raster line of the Test 4

When forming such a test, it is possible to take into account to some extent the uneven distribution of lengths of pixel chains of different colors. So, in particular, the chain of the background color can have the largest length – then the largest value of the $W_i$coefficient will be set for this color. The value of L can mean the overall scale – the resolution of the raster image.

As an example, consider the results of performing such a test for 256 colors of single pixels and chains for four main colors ($C1$ = 4) with coefficient parameters W_i = {1, 5, 10, 20} (Fig.10).

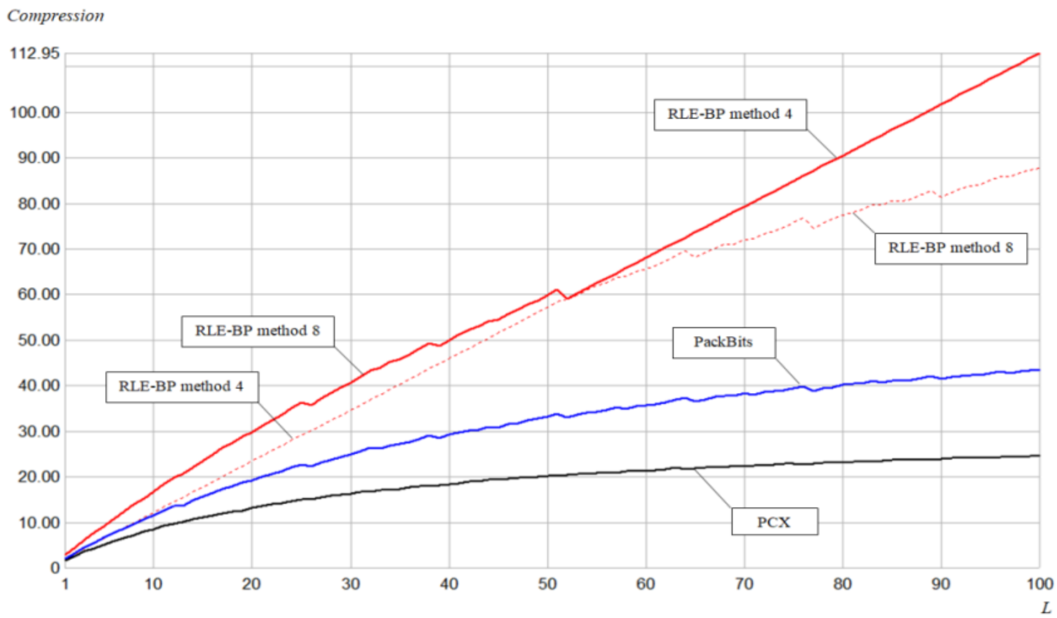

Fig. 10. Comparison of the degree of compression of known RLE methods (PackBits, PCX) and the proposed RLE_BP in Test 4. The limits of effective application of packaging methods 4 and 8 in Test 4 are highlighted

It should be noted that although this test is performed in the L range from 1 to 100, since the actual length of the chains is multiplied by $W_i$to determine the length, in fact, at L=100, in this test for the fourth main color with W_3 = 20, four chains of lengths are formed (LW_i), ($2LW_i$), ($3LW_i$), ($4LW_i$) that is 2000, 4000, 6000 and 8000 pixels, respectively.

The execution of this test illustrates the feasibility of using both method 4 and method 8 – but for different resolutions of the raster according to the value (L) of the base width of the chains.

6. Analysis of the obtained results of the raster's packaging

Even the small number of tests presented here seems to attest to the fact that it is probably inappropriate to try to use one encoding method for all images.

The main idea of RLE_BP is that the coder analyzes each specific line of the image and automatically chooses the coding method among the set of available ones, and moreover, finds such parameters of the codewords that provide the minimum number of bits of the output stream.

The main field of use of the RLE method is the encoding of raster images such as drawings, diagrams, and maps. Improving this method by inventing more complex and flexible coding methods, in particular, RLE_BP in combination with an automatic optimizing encoder allows to somewhat increase the degree of compression while maintaining the main advantage – high decoding speed and the convenience of providing direct access to high-resolution rasters.

Further studies may be devoted to the development of RLE_BP regarding the mathematical, algorithmic and technical aspects of choosing and implementing the optimal organization of saving raster data in information systems.

7. Conclusion

The problems of improving the method of RLE based on the methods of optimal encoding of bit sequences of individual fragments of raster images are investigated. The comparative testing of the realization of the modified method with known versions of RLE implementations is carried out. As a result of performing several tests, it has been proven that for the proposed RLE_BP coding methods, an increase in compression is achieved compared to known implementations of the RLE method.

References

[1] Wikipedia The Free Encyclopedia. “PackBits.” Accessed: Nov. 1, 2023. [Online]. Available: https://en.wikipedia.org/wiki/PackBits

[2] FileFormat.Info. “PCX File Format Summary.” Accessed: Nov. 1, 2023. [Online]. Available: https://www.fileformat.info/format/pcx/egff.htm

[3] C. Wang, R. L. Chen, and L. Gu, “Improving Performance of Virtual Machine Covert Timing Channel Through Optimized Run-Length Encoding,” *J. Comput. Sci. Technol.*, vol. 38, pp. 793–806, Jul. 2023, doi: 10.1007/s11390-021-1189-z.

[4] T. A. Welch, “Technique for High-Performance Data Compression,” *Computer*, vol. 17, issue 6, pp. 8-19, 1984, doi: 10.1109/MC.1984.1659158.

[5] T. A. Blinova and V. M. Poriev, “Some methods for encoding rasters in geographic information systems,” (in Russian), *Electronic Modeling*, vol. 30, no. 1, pp. 119-128, 2008, https://www.emodel.org.ua/images/em/all-pdf/08-1.pdf.

[6] T. Kathirvalavakumar and R. Palaniappan, "Modified Run-Length Encoding Method and Distance Algorithm to Classify Run-Length Encoded Binary Data," in *Proc. ICLICC 2011: Control, Computation and Information Systems*, 2011, pp. 271–280, doi: 10.1007/978-3-642-19263-0_33.

[7] A. Rahman and M. Hamada, “A prediction-based lossless image compression procedure using dimension reduction and Huffman coding,” *Multimedia Tools Appl.*, vol. 82, pp. 4081–4105, 2022, doi:10.1007/s11042-022-13283-3.

[8] A. Ibrahim, "Plain Text Encoding/Decoding Technique Using a Combination of Huffman and Run-Length Algorithms," *British Journal of Applied Science & Technology*, vol. 6(2), pp. 1-10, 2016, doi: 10.9734/bjast/2016/25712.

[9] V. Barannik et al., "Adaptive integer arithmetic coding with RLE-transform," (in Ukrainian), *Communication, informatization and cybersecurity systems and technologies,* issue3, pp. 5-13, 2023, doi: 10.58254/viti.3.2023.01.05.

[10] V. M. Poriev, “About modifications of the method of run length encoding,” (in Ukrainian), *Visnyk KPI. Informatics, Control and Computer Engineering*, issue 65, pp. 82-89, 2017, https://ela.kpi.ua/handle/123456789/22160.

УДК 004.62 https://doi.org/10.20535/2786-8729.4.2024.292118

ДЕЯКІ АСПЕКТИ ВДОСКОНАЛЕННЯ МЕТОДА КОДУВАННЯ RLE

Віктор Порєв
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5250-1454

Об’єктом дослідження, представленим у цій статі, є метод RLE та його застосування для компресії растрових зображень.

Метою даного дослідження є винайдення більш досконалих форматів кодових слів для пакування ланцюжків повторюваних пікселів сумісно з кодуванням одиночних пікселів зображення задля підвищення ступеня компресії методом RLE.

Для того, щоб досягти цієї мети, виконано узагальнення відомих форматів кодових слів у вигляді відповідної структурної моделі. Це дозволило знайти деякі напрямкі вдосконалення RLE кодування. Запропоновано декілька нових способів пакування ланцюжків пікселів разом з одиночними пікселами для підвищення ступеня компресії зображень порівняно з вже відомими версіями RLE. Вказані новітні способи включено до множини способів упаковки з назвою RLE_BP. Кодер RLE_BP автоматично шукає оптимальні параметри кодових слів задля досягнення мінімально можливого обсягу двійкового коду для конкретного зображення.

Виконані експериментальні дослідження компресії растрів на основі синтетичних тестів для порівняння запропонованих способів кодування з відомими реалізаціями метода RLE. Запропоновані способи кодування дозволяють досягти більшої компресії окремих категорій растрових зображень високої роздільної здатності порівняно з відомими.

Результати виконаного дослідження можуть бути використані для побудови широкого класу програмно-апаратних засобів.

Ключові слова: методи компресії, растрові зображення, кодування довжин повторів.