

THE USE OF DIFFERENCE COLOUR MODELS WITH COMPONENT DIFFERENCE OFFSETS TO IMPROVE THE COMPRESSION OF PNG IMAGES

Shportko O. V. – PhD, Associate Professor, Doctorant of the Postgraduate and Doctoral Studies Centre, National University of Water and Environmental Engineering, Rivne, Ukraine. ROR: <https://ror.org/00ppfcz28>. ORCID: <https://orcid.org/0000-0002-4013-3057>.

Bomba A. – Dr. Sc., Professor, Professor of the Department of Computer Sciences and Applied Mathematics of the National University of Water and Environmental Engineering, Rivne, Ukraine. ROR: <https://ror.org/00ppfcz28>. ORCID: <https://orcid.org/0000-0001-5528-4192>.

ABSTRACT

Context. Today, static difference colour models with integer coefficients are used to improve the efficiency of lossless image compression in graphic formats and archivers. These models improve compression, but do not consider the level of cross-correlation between different pairs of colour components of pixels in each image. Therefore, the development of methods for the formation and use of difference colour models for individual images in order to improve their compression by intercomponent decimation is currently an urgent scientific task.

Objective. To develop methods and algorithms for the transition to difference colour models with integer coefficients and difference offsets to reduce compression ratios in the process of lossless compression of RGB images in modern graphic formats, in particular in PNG format.

Method. Depending on the coding time constraints, the paper proposes to use 4, 16, 19 or 49 alternative difference colour models with difference offsets to select the most efficient model for each image. Prediction of the compression efficiency due to the use of the next alternative difference colour model is performed using entropy. The differences in the colour models are shifted so that the centre of the interval with the maximum number of these differences is shifted to the middle of the range of possible values. The effectiveness of three methods of determining the centre of this interval is investigated: without considering the deviations of component brightnesses, using the difference in component medians, and by determining the centre of the interval with the maximum number of component differences after their sequential search.

Results. Our experiments have shown that, for example, applying difference colour models with integer coefficients to whole images in the process of sequential lossless compression, in particular, in the PNG graphic format we modified, allows reducing compression ratios of photorealistic images of the ACT set by 0.19–1.06 bpb. Shifting the differences to the differences of the medians of individual components or centring the intervals of component differences provides an additional 0.01–0.02 bpb compression ratio reduction on average. Thus, difference colour models with integer coefficients and difference offsets can significantly increase the compression efficiency of lossless three-component photorealistic images in formats that use predictors and therefore can be implemented in the next versions of these formats at the standard level.

Conclusions. In graphic formats, to reduce the lossless image compression ratio, in addition to decorrelation of individual component data, it is advisable to perform intercomponent decorrelation by switching to difference colour models with integer coefficients with difference offsets, which provide fast decoding. To maximise the reduction in compression ratio due to the application of the selected colour model, the midpoints of the difference intervals of the basic components R , G , B should be shifted to the middle of the range of possible values. When, for photorealistic images, due to strict limitations on encoding time or encoder size, it is impossible to select a difference colour model from among 49, 19 or 16 alternative ones, this choice should be made among three models: $G - R + 128$, $G, G - B + 128$; $R, R - G + 128$, $B - G + 128$, or $R - G + 128$, $B - G + 128$, B . For synthesised images, it is not advisable to switch to difference colour models with integer coefficients.

KEYWORDS: lossless image compression, PNG graphic format, difference colour models with integer coefficients.

ABBREVIATIONS

ACT is an Archive Comparison Test;
ARIC is an Arithmetic Coding;
CM is a Colour Models;
CR is a Compression Ratio;
HUFF is a Huffman Coding;
JPEG is a Joint Photographic Experts Group;
LZ is a Lempel-Ziv algorithm;
PNG is a Portable Network Graphics;
RAM is a Random Access Memory;
RAR is a Roshal Archive;
RGB is a Red, Green, Blue;
YCrCb is a Luminance, Chrominance blue, Chrominance red.

NOMENCLATURE

B is a discrete value of the brightness of the blue component of an image pixel;
 $brightness_{uv}$ is a brightness of the next component of the image pixel located in row u and column v ;
 G is a discrete value of the brightness of the green component of an image pixel;
 H is an information entropy of the source;
 L is a length of the entropy code of the data block;
 l_i is a predicted length of the code of element i in the data block.
 $med(X)$ is a median value of the component X ;

$middle(X - Y)$ is a middle of the interval with the maximum number of differences between components X and Y ;

N is a number of elements in the data block;

n_i is a frequency of element i in the data block.

$numberCM$ is a number of the selected colour model for the next image from the list of alternatives;

p_i is a probability of occurrence of element i in the data block;

$predict_{uv}$ is a predicted brightness value of the next component of the image pixel located in row u and column v ;

R is a discrete value of the brightness of the red component of an image pixel;

Δ_{uv} is a deviation of the brightness of the next component of the image pixel located in row u and column v from the value predicted by the predictor.

INTRODUCTION

In modern information systems, data transmitted via communication channels and stored in computer memory most often belong to one of four main classes: texts, images, audio and video data. The size of such files varies considerably: while a page of unformatted text takes only a few kilobytes to store, a photorealistic raster uncompressed image measuring 9×13 cm takes several megabytes, a music recording takes tens of megabytes, and a full-length film takes several gigabytes. Obviously, processing, transmitting and storing such large amounts of information requires significant resources.

At the same time, it is known that most multimedia data is characterised by a high level of redundancy, which creates the prerequisites for efficient compression. Eliminating redundancy can significantly reduce file sizes, reduce bandwidth requirements and save disc space.

Existing image compression methods are divided into two main classes: lossy and lossless. Lossy algorithms, such as JPEG [1; 2], fractal and wavelet transforms [3], can reduce the volume of images by a factor of ten with little loss of quality. In this case, for photorealistic images with high resolution, the loss of some information is almost imperceptible. However, in the case of low-resolution graphics with clear lines, discrete tone transitions, logos or textures, even minor distortions become noticeable and impair visual perception. Medical, technical and scientific images (e.g., X-rays) are a separate category, where any distortion is unacceptable. In these cases, lossless image compression methods are used [4–5]. The algorithms of these methods ensure complete preservation of the quality of the original data, but their compression ratio (the ratio of the sizes of compressed to uncompressed image files, expressed in bpb, hereinafter CR) is much lower than that of lossy methods. Today, the development of this group of methods and algorithms does not stop, new approaches appear that, by means of decorrelation of individual pixel brightnesses (in particular, by switching to alternative colour models [6, 10]),

affect the levels of redundancy of various types and thus reduce CR.

Therefore, in the age of the information society, when huge amounts of graphic data are transmitted and stored daily, the task of developing new methods and improving the efficiency of lossless image compression algorithms remains relevant. It is essential both for applied fields (medicine, archiving, digitalisation of librarianship) and for the development of modern technologies for processing and storing information.

Today, one of the most effective and widely used formats for lossless image compression is the PNG graphic format [6], and the images themselves are most often represented using the True Color method in the RGB colour model with a sampling resolution of 8 bits for the red, green, and blue colour components of each pixel. Therefore, we tested our proposed alternative difference colour models for modifications of this graphic format by compressing RGB images with 24-bit colour depth.

Thus, **the object of this study** is methods and algorithms for decreasing pixel brightnesses in the process of lossless image compression. **The subject of the study** is the possibility of using difference colour models with integer coefficients to improve the compression of RGB images in PNG format. **The aim of the study** is to develop methods and algorithms for transition to difference colour models with integer coefficients and difference offsets to reduce CR in the process of lossless compression of RGB images.

1 PROBLEM STATEMENT

It is well known that data compression is possible by reducing redundancies. Lossless image compression consists of two consecutive independent stages: transition to an alternative image representation (display), during which inter-element redundancy is reduced, and element-by-element encoding of the resulting data to eliminate code redundancy.

Suppose that in the next data block, each of the values of i occurs n_i times, forming a sequence of length

$$N = \sum_i n_i.$$

According to the fundamental provision of information theory, in the process of element-by-element coding, to minimise the code length of the sequence, each element with a probability of occurrence of

$$p_i = \frac{n_i}{N}.$$

should be encoded with $l_i = -\log_2 p_i$ bits (this length is called “entropy”). The average length of the element code after applying such coding cannot be less than the information entropy of the source, which is calculated by the Shannon formula [7]:

$$H = -\sum_i p_i \times \log_2 p_i . \quad (1)$$

Therefore, the total length of the code after elemental coding is not less than the length of the entropy code of the sequence [10], which, given formula (1), is

$$L = N \times H = N \log_2(N) - \sum_i n_i \log_2(n_i). \quad (2)$$

To reduce inter-element redundancy in the PNG format, predictors [6, p. 241] and the LZ77 dictionary algorithm [8] are used within each colour component, and Huffman codes (HUFF) [9] are used to eliminate code redundancy. However, to date, the PNG format does not perform intercomponent decorrelation to reduce entropy [1].

The fact is that the different components of True Colour images (*R*, *G* and *B*) reflect objects that are quite similar in geometric and spatial structure [10]. The correlation coefficients between pairs of components in the RGB colour model for different images can differ significantly, and one of the components of any such pair with a strong correlation can be replaced **by a difference** with another component [10; 11], if this reduces the entropy (1) in the process of further elemental coding. The use of component differences instead of component brightnesses of the RGB colour model actually performs intercomponent decorrelation, just as predictors implement decorrelation between the brightnesses of adjacent pixels. Currently, modern archivers and image compression formats process pixel brightnesses mainly in a fixed colour model (e.g., PNG format – in the *R, G, B* model; BMP format – in the *B, G, R* model; JPEG format [1; 2] – in the *YCbCr* model; RAR archiver format [12] – in the *R-G, G, B-G* model) and do not use the possibility of choosing an effective colour model for each image that minimises entropy (1) through intercomponent decoration [10]. In particular, in the popular *YCbCr* colour model [6], the differences of the *R, G*, and *B* components are used in the two chromatic components *Cb* and *Cr*:

$$\begin{aligned} Y &= \left(\frac{77}{256}\right)R + \left(\frac{150}{256}\right)G + \left(\frac{29}{256}\right)B; \\ Cb &= -\left(\frac{44}{256}\right)R - \left(\frac{87}{256}\right)G + \left(\frac{131}{256}\right)B + 128; \\ Cr &= \left(\frac{131}{256}\right)R - \left(\frac{110}{256}\right)G - \left(\frac{21}{256}\right)B + 128, \end{aligned} \quad (3)$$

but when compressed, this model is applied to all images [1, 2]. In this article, we will investigate the possibilities of choosing effective colour models with component difference offsets **for each image** in order to reduce CR in PNG format.

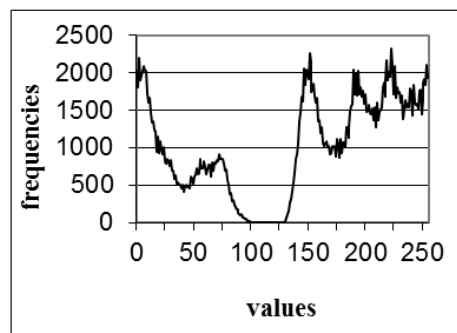
2 REVIEWS OF THE LITERATURE

In [10], we presented 49 difference colour models with integer coefficients and offsets of component differences by the opposite median differences of these components and the middle of the range of possible values that can be effective in lossless image compression (hereinafter, such offsets will be abbreviated as “by the difference of medians”). Therefore, let us describe in more detail the principles of applying preprocessing and compression algorithms used in the PNG format [13] and analyse the impact of difference colour models on their efficiency:

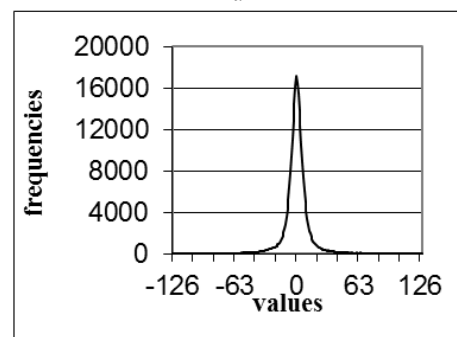
1. During the sequential traversal of the image pixels, predictors predict the brightness value of each component, recorded as integers in separate bytes, using the brightness values of the same components of previously processed adjacent pixels [6, p. 241], since these brightnesses have the highest correlation with each other. At the same time, the deviation Δ_{uv} of the brightness value of the next pixel component *brightness_{uv}* from the value predicted by the selected predictor *predict_{uv}* is calculated and further encoded:

$$\Delta_{uv} = \text{brightness}_{uv} - \text{predict}_{uv} \quad (4)$$

(*u* and *v* run through all rows and columns of image pixel components, respectively). Adjacent image pixels most often have similar colours (similar brightness values of the corresponding components), so the prediction often coincides with the brightness value of the next component, most often is close to this value, and rarely differs significantly from it.



a



b

Figure 1 – A division of frequencies of values of green components of image *Lena.bmp*: a – before the use of the predictor ($H=7.59$ of bpb); b – after the use of *Left*-predictor ($H=5.34$ of bpb) with [14]

That is, most of the Δ_{uv} values are close to zero (Fig. 1b). In this way, the use of predictors often increases the unevenness of the probability distribution of luminance values and, as a result, reduces entropy (1). Predictors take into account changes in luminance relative to adjacent pixels within each component. The brightness of pixels in different components can vary unequally. Therefore, the use of predictors reduces the intercomponent correlation, negatively affecting the effectiveness of difference colour models. Therefore, these models should be applied before predictors.

On the other hand, when calculating the luminance differences of the R , G or B components, in cases where the values of the negative component's luminance are slightly higher than the values of the positive component's luminance, negative values of the differences are obtained, which, when stored in unsigned 8-bit components of the colour model, are transformed into values close to 256 due to overflow. And when the brightness values of the negative component are slightly less than the brightness values of the positive component, the differences obtain values close to zero. For example, the frequencies of the differences of the R and G components near zero for the Lena.bmp image are non-zero (Figure 2a). It is clear that such a scattering of colour model difference values negatively affects the predictor's prediction accuracy, disperses the differences (4) and, as a result, increases the entropy (1).

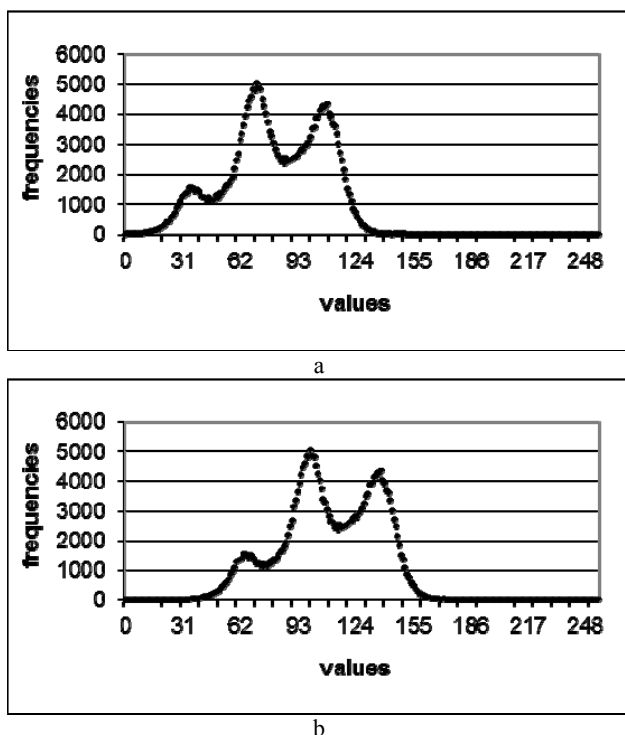


Figure 2 – Frequency distribution of brightness differences of individual components of image Lena.bmp:
 a – difference $R - G$; b – shifted difference $R - G + 29$ obtained from [11]

To avoid such a scattering of differences of close component brightness values, it is customary to shift them to the middle of the range of possible values. For example, in the YCbCr colour model (3), for the most common 8-bit sampling, the difference values are shifted by 128 (in this article, by shift by 128 we mean the shift to the middle of the range of possible values). In [10], we shifted the differences of the RGB colour model components by 128 and by the opposite differences of the medians of these components (Figure 2b), expecting that most component brightnesses are concentrated around their medians, but then the predictors' prediction of the brightnesses of component differences that are close to 128 from this median difference worsens.

2. The LZ77 dictionary algorithm [8] is used in many graphic formats and archivers because it provides the fastest decoding. Describing dictionary algorithms, a fixed number of previously encoded indivisible elements (literals) of the input stream is called a “dictionary”, and the subsequent unencoded elements are called a “buffer”. The LZ77 algorithm is based on replacing a subsequence of consecutive buffer literals in the output stream with a reference to a similar subsequence of dictionary literals in the form of a pair of numbers $\langle \text{length}; \text{offset} \rangle$ from the end of the dictionary. In the absence of a similar subsequence of literals in the dictionary, the first literal of the buffer is transferred to the output stream without change. After that, the encoded literals are transferred from the beginning of the buffer to the end of the dictionary and the encoding continues in the same way until the end of the input stream literals. For example, the stream of luminance codes of pixel components 36 38 35 35 36 38 35 36 36 38 35 38 36 38 35 35 28 is written in encoded form as 36 38 35 35 $\langle 3; 4 \rangle$ 36 $\langle 3; 4 \rangle$ 38 $\langle 4; 12 \rangle$ 28. When decoding LZ77 algorithm codes, individual literals are copied to the output stream without change. Pairs $\langle \text{length}; \text{offset} \rangle$ are decoded by sequentially copying the required number of literals from the end of the source stream at the specified offset to the end of the source stream.

It is clear that the more $\langle \text{length}; \text{offset} \rangle$ pairs can be found and the longer their lengths are, the more efficient the compression by the LZ77 algorithm is. However, photorealistic images and drawings with smooth colour changes contain a small number of identical subsequences of pixel brightnesses, so this algorithm is effective primarily for discrete-tone drawings.

Difference colour models transform the brightness of the components of individual pixels and can reduce or even break identical subsequences. Therefore, it is advisable to search for identical subsequences of the LZ77 algorithm in the input RGB image, and apply difference colour models to individual literals.

That is, for each image, you should choose a difference colour model from the list of alternatives, taking into account the pixels that are not included in the long substitutions of the LZ77 algorithm.

3. Context-independent Huffman coding [9] assigns a fixed prefix code to each element depending on its probability (frequency of occurrence). This coding is most

often performed in the following sequence: calculate the probabilities (frequencies) of individual items; order the items in descending order of probability; iteratively combine the two items with the lowest probabilities (most often the last ones in the list) to obtain one item, assigning a code 0 to the first item and a code 1 to the second item; sum the probabilities of the selected items to calculate the probability of the resulting item and insert this item into the sorted list of probabilities; generate HUFF codes by writing the generated codes in reverse order – from the top to each item. The average length of a HUFF code coincides with the source entropy (1) only if the lengths of their optimal codes $-\log_2 p_i$ are the same for all elements i . In addition, the length of the HUFF code, even for the most probable element, cannot be less than one bit. Therefore, in our modifications of the PNG format, we replaced HUFF encoding with arithmetic coding (ARIC) [15; 16].

Difference colour models are used to reduce the entropy (1) and therefore improve the CR of the context-independent algorithm. In this paper, we show how to further reduce CR by using not only differences but also component difference offsets in colour models.

3 MATERIALS AND METHODS

We will shift the component differences so that the maximum number of these differences is concentrated around the middle of the range of possible values (128). Let's denote the middle of the interval with the maximum number of differences between components X and Y by $middle(X - Y)$ (we will discuss the choice of this interval later). Then, to shift this middle to the middle of the range of possible values, you need to subtract $middle(X - Y)$ from the component difference and add 128. Since in difference colour models, at most two components can be replaced by component differences and they must provide unambiguous decoding, then, by analogy with [10], the following difference colour models with integer coefficients and shifted differences are alternative for compressing each image:

0. $R, G, B;$
1. $R, G, R - B - middle(R - B) + 128;$
2. $R, G, G - B - middle(G - B) + 128;$
3. $R, R - G - middle(R - G) + 128, B;$
4. $R, B - G - middle(B - G) + 128, B;$
5. $G - R - middle(G - R) + 128, G, B;$
6. $B - R - middle(B - R) + 128, G, B;$
7. $R, G, B - R - middle(B - R) + 128;$
8. $R, G, B - G - middle(B - G) + 128;$
9. $R, G - R - middle(G - R) + 128, B;$
10. $R, G - B - middle(G - B) + 128, B;$
11. $R - G - middle(R - G) + 128, G, B;$
12. $R - B - middle(R - B) + 128, G, B;$
13. $R, R - G - middle(R - G) + 128, R - B - middle(R - B) + 128;$
14. $R, G - R - middle(G - R) + 128, G - B - middle(G - B) + 128;$
15. $R, B - G - middle(B - G) + 128, B - R - middle(B - R) + 128;$
16. $R, G - R - middle(G - R) + 128, B - R - middle(B - R) + 128;$
17. $R, R - G - middle(R - G) + 128, B - G - middle(B - G) + 128;$
18. $R, G - B - middle(G - B) + 128, R - B - middle(R - B) + 128;$

19. $R - G - middle(R - G) + 128, G, R - B - middle(R - B) + 128;$
20. $G - R - middle(G - R) + 128, G, G - B - middle(G - B) + 128;$
21. $B - R - middle(B - R) + 128, G, B - G - middle(B - G) + 128;$
22. $G - R - middle(G - R) + 128, G, B - R - middle(B - R) + 128;$
23. $R - G - middle(R - G) + 128, G, B - G - middle(B - G) + 128;$
24. $R - B - middle(R - B) + 128, G, G - B - middle(G - B) + 128;$
25. $R - B - middle(R - B) + 128, R - G - middle(R - G) + 128, B;$
26. $G - R - middle(G - R) + 128, G - B - middle(G - B) + 128, B;$
27. $B - R - middle(B - R) + 128, B - G - middle(B - G) + 128, B;$
28. $B - R - middle(B - R) + 128, G - R - middle(G - R) + 128, B;$
29. $R - G - middle(R - G) + 128, B - G - middle(B - G) + 128, B;$
30. $R - B - middle(R - B) + 128, G - B - middle(G - B) + 128, B;$
31. $R, G - R - middle(G - R) + 128, R - B - middle(R - B) + 128;$
32. $R, R - G - middle(R - G) + 128, G - B - middle(G - B) + 128;$
33. $R, G - B - middle(G - B) + 128, B - R - middle(B - R) + 128;$
34. $R, R - G - middle(R - G) + 128, B - R - middle(B - R) + 128;$
35. $R, G - R - middle(G - R) + 128, B - G - middle(B - G) + 128;$
36. $R, B - G - middle(B - G) + 128, R - B - middle(R - B) + 128;$
37. $G - R - middle(G - R) + 128, G, R - B - middle(R - B) + 128;$
38. $R - G - middle(R - G) + 128, G, G - B - middle(G - B) + 128;$
39. $R - B - middle(R - B) + 128, G, B - G - middle(B - G) + 128;$
40. $R - G - middle(R - G) + 128, G, B - R - middle(B - R) + 128;$
41. $G - R - middle(G - R) + 128, G, B - G - middle(B - G) + 128;$
42. $B - R - middle(B - R) + 128, G, G - B - middle(G - B) + 128;$
43. $B - R - middle(B - R) + 128, R - G - middle(R - G) + 128, B;$
44. $R - G - middle(R - G) + 128, G - B - middle(G - B) + 128, B;$
45. $R - B - middle(R - B) + 128, B - G - middle(B - G) + 128, B;$
46. $R - B - middle(R - B) + 128, G - R - middle(G - R) + 128, B;$
47. $G - R - middle(G - R) + 128, B - G - middle(B - G) + 128, B;$
48. $B - R - middle(B - R) + 128, G - B - middle(G - B) + 128, B.$

Let us denote these difference colour models with the centre of the intervals with the maximum number of differences by $CM_j, j = 0, 48$ (CM is an abbreviation for colour models). It is clear that among the alternative difference colour models (5) for each image, $CM_{numberCM}$ should be chosen, which, after applying the predictors (4), provides the smallest predicted entropy code length (2) of pixels that are not included in the long substitutions of the LZ77 algorithm:

$$L(\Delta(CM_{numberCM})) = \min_j L(\Delta(CM_j)). \quad (6)$$

After applying the predictors, the length of the entropy code of the component difference (2) is close to the length of the entropy code of the opposite difference, because it is based not on the values of the elements, but on their frequencies. However, the length of the entropy code after applying difference colour models with opposite differences and predictors (4) may slightly decrease primarily due to an increase in cross-correlation between components. Therefore, to speed up the selection of a difference colour model with interval centring with the maximum number of elements, we will select 16 basic models out of 49 alternative models (5) [11]:

0. $R, G, B;$
1. $R, R - G - middle(R - G) + 128, R - B - middle(R - B) + 128;$
2. $R, R - G - middle(R - G) + 128, B - G - middle(B - G) + 128;$
3. $R, G - B - middle(G - B) + 128, R - B - middle(R - B) + 128;$
4. $G - R - middle(G - R) + 128, G - B - middle(G - B) + 128;$

5. $G - R - middle(G - R) + 128, G, B - R - middle(B - R) + 128;$
6. $R - B - middle(R - B) + 128, G, G - B - middle(G - B) + 128;$
7. $B - R - middle(B - R) + 128, G - R - middle(G - R) + 128, B;$
8. $R - G - middle(R - G) + 128, B - G - middle(B - G) + 128, B;$
9. $B - R - middle(B - R) + 128, B - G - middle(B - G) + 128, B;$
10. $R, G, G - B - middle(G - B) + 128;$
11. $R, G, R - B - middle(R - B) + 128;$
12. $R, R - G - middle(R - G) + 128, B;$
13. $R, B - G - middle(B - G) + 128, B;$
14. $G - R - middle(G - R) + 128, G, B;$
15. $B - R - middle(B - R) + 128, G, B.$

From (5), alternative difference colour models whose difference components contain the same decreasing or increasing factors of the same sign, which increases their cross-correlation, are selected in list (7). It is advisable to form an alternative difference colour model using the list (7) in two stages: at the first stage, select among them according to (6) such a difference colour model that provides the smallest predicted length of the entropy code (2) after applying the predictors, and at the second stage, compare the effectiveness of the selected model with a maximum of three alternative difference colour models with opposite differences (with two opposite differences, with the first opposite difference, and with the second opposite difference). For example, if after the first stage, the smallest predicted entropy code length is provided by the difference colour model $G - R - middle(G - R) + 128, G, G - B - middle(G - B) + 128$, then at the second stage, it will be necessary to compare its effectiveness only with the following alternative difference colour models: $R - G - middle(G - R) + 128, G, B - G - middle(B - G) + 128; R - G - middle(R - G) + 128, G, G - B - middle(G - B) + 128; G - R - middle(G - R) + 128, G, B - G - middle(B - G) + 128$. This approach reduces the number of analysed alternative colour models from 49 to a maximum of 19 and therefore speeds up the selection of an alternative colour model by 2.5 times, simplifies the implementation of the encoder and, when performing only the first stage, provides the ability to compactly store the number of the selected one from 16 basic alternative difference colour models (4 bits are enough for this).

In the following, we will investigate the effectiveness of three methods for determining the centre of an interval with a maximum number of differences:

1. To speed up the determination of difference colour models and in cases where the components have close or even the same brightness (like grayscale images), the deviation of the brightnesses of the component differences from each other can be ignored by setting

$$middle_1(X - Y) = 0. \quad (8)$$

Then the component differences will be shifted by 128 (the middle of the range of possible values).

2. If most of the brightnesses of the pixel components are concentrated around their medians, i.e. they have a small dispersion, then we shift the difference of the medi-

ans of these components to the middle of the range of possible values:

$$middle_2(X - Y) = med(X) - med(Y). \quad (9)$$

This method of determining the centre of the interval with the maximum number of differences is implemented by first determining the medians of the three components.

3. By selecting the interval with the maximum number of component differences among all intervals of width $256 - b$ by sequentially searching them (here b is the number of values that are not included in the interval from the entire range of possible values). In the following, these intervals are called “selected”. For example, to determine the centre of an interval with a maximum number of differences with a range of 241 values ($b = 15$) from an array of frequencies of differences, we used the following C++ function:

```
UBYTE1 middleMaxSumFreq(UBYTE4* masFreq)
{UBYTE4 maxSumFreq = 0, sumFreq;
 UBYTE1 middle = 0; short i, j;
 for (i = 0; i <= 255; i++) // possible midpoints
  // counting of the sum of frequencies
  // of the current interval
  sumFreq = 0;
  for (j = i - 120; j <= i + 120; j++)
   sumFreq += masFreq[(UBYTE1)j];
  if (sumFreq > maxSumFreq)
   {maxSumFreq = sumFreq;
    middle = i; }
 return middle; }
```

The execution of this function can be accelerated if, instead of a nested loop, when moving from the previous to the next interval, you remove the frequency of the previous interval on the left, which is no longer included in the next interval, and add the frequency of the next interval on the right, which has just entered it, but this will reduce its visibility.

4 EXPERIMENTS

We tested the effectiveness of the proposed lists of difference colour models with component difference offsets (5) and (7) with the three methods of determining the centre of intervals with the maximum number of differences for their centering by compressing images of the well-known ACT test set [17], modifying the encoder/decoder codes [6] of the PNG format [13] in the C++ programming language. Reduced copies of the images of this set are shown in Figure 3. This particular test set was chosen due to the availability in open sources of the results of testing algorithms by other researchers and the diversity of its images: no. 1, 2 and 7 are artificial, but the first of them is continuous-tone, and the second and third are discrete-tone, all other images are photorealistic with different sizes and numbers of objects. We conducted the testing on a computer with an Intel Pentium 4 processor with a clock speed of 3 GHz and 4Gb RAM.

5 RESULTS

First, let's compare the effect on the CR of the ACT set images in the modified PNG format (Table 1) of individual colour models from the 16 main alternative ones (7) with difference offsets of 128 (the middle of the range of possible values (8), the first method of determining the centre of the interval). From the data in this table, we can see that different images are affected differently

by individual colour models. In particular, different colour models do not significantly affect the synthesised images or worsen their compression relative to the RGB model. **Therefore, for each image, you need to choose the difference colour model that reduces CR (6) as much as possible or revert to the RGB model if there is no reduction.**

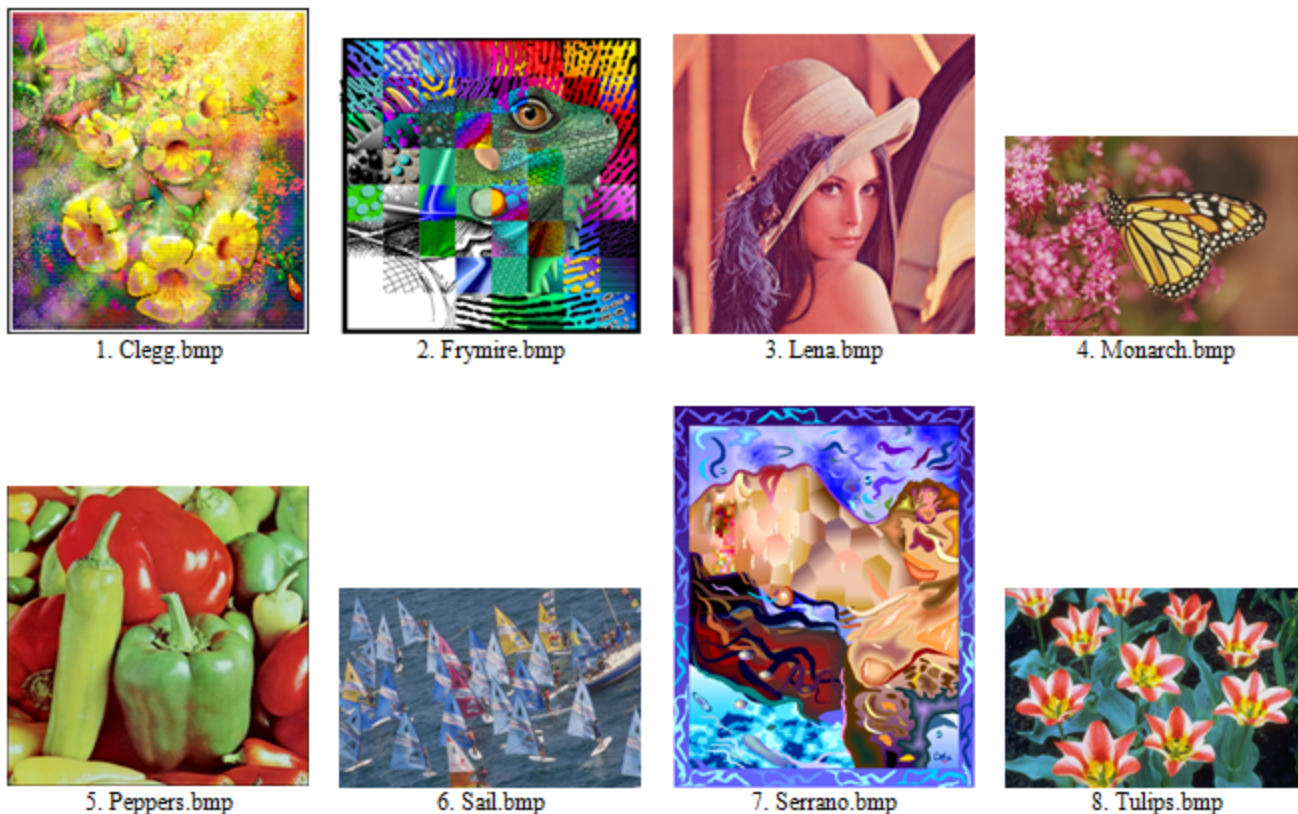


Figure 3 – Image of the ACT kit

Table 1 – Compression ratios of ACT set images in modified PNG format after applying individual difference colour models from 16 main alternative ones (7) with difference offsets by 128 (8), bpp

Colour model	file no.								Average CS
	1	2	3	4	5	6	7	8	
0. R, G, B	1.68	0.539	4.82	4.16	4.31	5.26	0.56	4.60	3.24
1. $R, R-G+128, R-B+128$	1.69	0.557	4.82	3.56	3.86	4.32	0.57	4.05	2.93
2. $R, R-G+128, B-G+128$	1.71	0.557	4.63	3.55	4.15	4.23	0.57	3.88	2.91
3. $R, G-B+128, R-B+128$	1.72	0.557	4.84	3.55	3.86	4.37	0.57	3.99	2.93
4. $G-R+128, G, G-B+128$	1.75	0.557	4.75	3.56	3.82	4.20	0.56	3.99	2.90
5. $G-R+128, G, B-R+128$	1.74	0.559	4.93	3.57	3.87	4.30	0.57	4.16	2.96
6. $R-B+128, G, G-B+128$	1.77	0.559	4.96	3.57	3.88	4.35	0.56	4.10	2.97
7. $B-R+128, G-R+128, B$	1.72	0.557	4.98	3.61	3.89	4.30	0.56	4.20	2.98
8. $R-G+128, B-G+128, B$	1.73	0.554	4.81	3.59	3.84	4.21	0.56	4.03	2.92
9. $B-R+128; B-G+128, B$	1.75	0.554	4.99	3.60	3.90	4.35	0.56	4.14	2.98
10. $R, G, G-B+128$	1.71	0.546	4.70	3.89	4.07	5.00	0.57	4.24	3.09
11. $R, G, R-B+128$	1.70	0.548	4.88	3.89	4.12	5.06	0.57	4.39	3.14
12. $R, R-G+128, B$	1.67	0.548	4.75	3.93	4.09	4.97	0.57	4.34	3.11
13. $R, B-G+128, B$	1.69	0.546	4.76	3.92	4.09	5.01	0.57	4.28	3.11
14. $G-R+128, G, B$	1.71	0.548	4.87	3.94	4.10	4.95	0.56	4.43	3.14
15. $B-R+128, G, B$	1.73	0.548	5.06	3.93	4.15	5.05	0.56	4.52	3.19

The best results for compression of the ACT image set are, on average, provided by the colour models $G-R+$

$128, G, G-B+128$ and $R, R-G+128, B-G+128$. The colour model $R-G+128, B-G+128, B$ is slightly

worse. An analogue of the first of these models without difference offsets is used in the WinRAR archiver [12] to compress RGB images. On average, these colour models in the ACT set improve compression relative to the RGB model by 0.32–0.34 bpb for photorealistic images. Each of these three models has a different carrier component, so we recommend using them by default in graphic formats as the main alternative to the RGB model when, due to strict restrictions on encoding time or encoder size, it is impossible to select a differential colour model from the lists of alternatives (5) or (7).

Differential colour models reduce the brightness differences in the difference components (see, for example, Figure 4 in comparison with Figure 3) and therefore increase the efficiency of the predictors (4) and, as a result, reduce the length of the entropy code of the sequence (2).

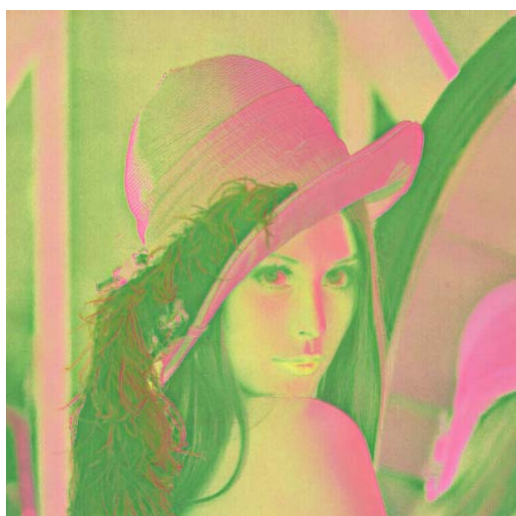


Figure 4 – Image Lena.bmp in a differential colour model
 $R, R - G + 128, B - G + 128$

Now let's analyse the effect on the sequential compression performance of the ACT set images in the modified PNG format (Table 2 – Table 4) of the differential colour models selected according to (6) from the list of alternative (5) with shifts of component differences by the inverse of their median differences (9) (the second way to determine the centre of the interval). In these experiments, we returned to the RGB colour model when the predictors were not applied to at least 75% of the image lines. This is typical for cumbersome discrete-tone images.

As shown in these tables, as a result of applying difference colour models with integer coefficients and with differences in the medians of the components, the CR of photorealistic images of the ACT set decreased from 0.2 bpb to 1.06 bpb. On average, the set CR decreased by 0.39 bpb (first and second rows of Table 2), which correlates with the effect of these models on CR in the process of progressive hierarchical traversal [10; 11] with almost the same average decoding time (Table 4). **That is, the reduction in image CR due to the use of difference colour models depends not so much on the pixel traversal method as on the type of image and the brightness differences of its components.** At the same time, the contribution of shifts of component differences to the middle of the range of possible values (just adding 128) to the CR reduction is only 0.01 bpb on average (third and fourth rows of Table 2). The contribution of taking into account the differences in the medians of the components (9) to the reduction in CR is also 0.01 bpb on average (second and fourth rows of Table 2). In other words, under strict coding time constraints, it is sufficient to shift the component brightness differences by 128, as in the YCbCr colour model (3), without considering the component medians or other centring options.

Table 2 – Compression ratios of ACT set images in the modified PNG format after applying difference colour models with different variants of using differences of medians of components (9), bpb

Colour model	file number								Average CS
	1	2	3	4	5	6	7	8	
RGB	1.68	0.539	4.82	4.16	4.31	5.26	0.56	4.60	3.24
Difference of 49 alternatives with differences in component medians	1.67	0.541	4.62	3.55	3.81	4.20	0.56	3.89	2.85
Difference from 49 alternatives without difference shifts	1.67	0.539	4.70	3.55	3.82	4.22	0.56	3.89	2.87
Difference of 49 alternative medians without differences with difference shifts of 128	1.67	0.539	4.63	3.55	3.81	4.20	0.56	3.88	2.86

Table 3 – Time for encoding ACT image files into a modified PNG format using difference colour models with differences in component medians (9), s

Colour model	File number								Average time
	1	2	3	4	5	6	7	8	
RGB	5.42	7.17	2.19	4.53	2.70	3.31	3.06	3.80	4.02
Difference from 49 alternatives	5.69	7.69	2.24	4.99	2.98	4.08	3.22	4.44	4.42

Table 4 – Decoding time of ACT set image files from the modified PNG format encoded using difference colour models with differences in component medians (9), s

Colour model	file number								Average time
	1	2	3	4	5	6	7	8	
RGB	0.58	0.84	0.38	0.41	0.33	0.56	0.22	0.50	0.48
Difference from 49 alternatives	0.56	0.80	0.38	0.48	0.30	0.52	0.34	0.49	0.48

In these experiments, we determined the difference colour models for each image based on the results of applying (6) and the linear predictor *LeftPredict* [14], for which, for example, $L(\Delta(R - G)) = L(\Delta(G) - \Delta(R))$, which allowed us to take into account the results of applying this predictor only for individual components without the need to calculate the frequencies of its use to the differences of the components and without taking into account the offset. Such simplifications reduce the encoding slowdown by up to 10 % on average due to the need to select the variant of the difference colour model (Table 3).

Let us also analyse the effect of difference colour models with the centre of selected intervals of component differences (the third method of determining the centre of the interval) on the performance of sequential compression of ACT images in the modified PNG format (Table 5). The encoding and decoding times of images with these models are similar to those shown in Table 3 – Ta-

Table 5 – Compression ratios of ACT set images in the modified PNG format after applying different variants of forming difference colour models with the centre of selected intervals of component differences, bpb

Colour model with centring of selected intervals of component differences	file number								Average CS
	1	2	3	4	5	6	7	8	
Difference from 19 or 16 main alternatives (7)	1.67	0.539	4.62	3.55	3.81	4.20	0.56	3.87	2.85

6 DISCUSSION

In addition to studying the effectiveness of applying difference colour models with component difference offsets, we are also studying the feasibility of using different difference colour models for individual image fragments. To date, we have investigated the possibility of using two types of fragments to solve this problem:

1. Application of differential colour models with integer coefficients to individual MCUs (Minimum Coding Unit, blocks of 8 x 8 pixels, which are used, among other things, in JPEG format [1]) **with the possibility of switching to a single differential colour model for the entire image.** The use of such fragments additionally reduces CR by an average of only 0.02 bpb, but at the same time slows down encoding by 2.88 times and decoding by 3.39% and therefore is not recommended for use in graphic formats. It should be noted that the same luminance differences in different colour models of different MCUs can cause unequal results of the predictors' application and therefore lead to scattering of element values after their application, and, as a result, to an increase in entropy. In addition, when applying difference colour models to individual MCUs, you also need to store the numbers of the selected models for each MCU. Therefore, for the next individual image, both a single difference colour model and a set of different colour models for individual MCUs may be more effective.

2. Splitting images into rectangular fragments of non-overlapping pixels covering the entire image, with the definition of different difference colour models for two adjacent rectangles at each splitting. The use of such fragments, for example, for the ACT set, reduced CR by an average of only 0.01 bpb, while slowing down the encoding by 6.18 times. Therefore, just like for MCU, we

do not recommend splitting images into rectangular pixel fragments with different difference colour models with integer coefficients in graphic formats. Differential colour models should be applied to individual MCUs or rectangular fragments in archivers to ensure maximum lossless image compression.

Therefore, the issue of finding an effective image fragmentation for applying different difference colour models to individual fragments remains open at this time.

CONCLUSIONS

1. It is possible to reduce the CR of images in three-component colour models not only by decreasing the data of individual components, but also by intercomponent decreasing by switching to difference colour models. It is advisable to perform intercomponent decoration in such a way as to enhance the image properties used by preprocessing algorithms and direct compression of the selected graphic format, for example, to minimise the predicted length of the entropy code (2). To evaluate the effectiveness of preprocessing algorithms during compression, it is advisable to calculate this length only for pixels that should be processed by a context-independent algorithm.

2. For lossless image compression in graphic formats, it is advisable to use difference colour models with integer coefficients with difference offsets, as they provide fast decoding, although their determination slows down encoding. In order to maximise the reduction of CS due to the application of the selected colour model, the midpoints of the difference intervals of the basic components *R*, *G*, *B* should be shifted to the middle of the range of possible values (for example, in colour models with a sampling resolution of 8 bits – up to 128).

3. The application of difference colour models with integer coefficients to whole images in the process of lossless sequential compression, for example, can reduce the CR of photorealistic images of the ACT set from 0.19 bpb to 1.06 bpb. Shifting the differences to the differences of the medians of individual components or centring the selected intervals of component differences can further reduce the CR by an average of 0.01–0.02 bpb. Thus, difference colour models with integer coefficients and difference offsets can significantly improve the compression efficiency of lossless three-component photorealistic images in formats that use predictors, and therefore can be implemented in the next versions of these formats at the standard level.

4. To ensure faster encoding, a difference colour model with integer coefficients for an entire image should be selected from a maximum of 19 alternatives (16 basic alternatives from the list (7) and additionally a maximum of three with symmetric differences to the one selected from the basic alternatives), rather than from all possible 49 models (5).

5. In graphic formats, when for photorealistic images, due to strict encoding time or encoder size limitations, it is impossible to select a difference colour model from 19 alternative ones, we recommend selecting a difference colour model among three: $G - R + 128$, G , $G - B + 128$; R , $R - G + 128$, $B - G + 128$, or $R - G + 128$, $B - G + 128$, B , as they provide on average better CRs than other alternative difference colour models. When this choice is not possible, we recommend using the $G - R + 128$, G , $G - B + 128$ colour model by default. For synthesised images, it is usually not advisable to switch to difference colour models with integer coefficients.

In the future, we plan to officially propose to the developers of the PNG format to introduce difference colour models with difference offsets in the next versions of this format.

ACKNOWLEDGEMENTS

This article is a development of the research work “Development of methods and graphic format of making progressive compression color images loss less” (state registration number 0113U001203), funded by the state budget, and the research work “Improving image compression loss less in modern graphics formats” (state registration number 0110U004001) of the Rivne State Humanitarian University.

DECLARATIONS

Conflict of interest: The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Authors' contributions: All authors contributed equally to the research, numerical experiments, data analysis, etc.

Data availability: The data will be made available on reasonable request from authors by email, then provide email ITShportko@gmail.com.

Software availability: The software will be made available on reasonable request by email, then provide email ITShportko@gmail.com.

Use of artificial intelligence tools: The authors confirm that they did not use artificial intelligence technologies in creating the materials and conducting the experiments of the submitted work.

REFERENCES

- Wallace G. The JPEG still picture compression standard. *Communication of ACM*, 1991, Vol. 34, № 4, pp. 30–44. DOI: 10.1145/103085.103089.
- Shehata O. Unraveling the JPEG [Electronic resource]. *Parametric Press*, 2019, Iss. 1. Access mode: <https://parametric.press/issue-01/unraveling-the-jpeg>.
- Guo T., Zhang T., Lim E., López-Benítez M., Ma F., Yu L., A review of wavelet analysis and its applications: Challenges and opportunities. *IEEE Access*, 2022, № 10, pp. 58869–58903. DOI: 10.1109/ACCESS.2022.3179517.
- Selomon D. A Guide to Data Compression Methods. Springer, New York, 2002, 295 p. DOI: 10.1007/978-0-387-21708-6.
- Kotha H. D., Tummanapally M., Upadhyay V. K. Review on Lossless Compression Techniques, *Journal of Physics*, 2019, Vol. 1228, 6 p. DOI: 10.1088/1742-6596/1228/1/012007.
- Miano J. Compressed Image File Format: JPEG, PNG, GIF, XBM, BMP, Addison Wesley. New York, 1999, 264 p. ISBN 0201604434.
- Shannon C. E. A Mathematical Theory of Communication. *Bell System Technical Journal*, 1948, Vol. 27, pp. 379–423, 623–656. DOI: 10.1002/j.1538-7305.1948.tb00917.x.
- Ziv J., Lempel A. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, May 1977, Vol. 23(3), pp. 337–343.
- Huffman D. A Method for the Construction of Minimum Redundancy Codes. *Proceedings of the IRE*, 1952, Vol. 40(9), pp. 1098–1101.
- Shportko A. V., Bomba A. Ya, Postolatii V. A. Programming the Formation of Difference Color Models for Lossless Image Compression. *Computational Linguistics and Intelligent Systems (COLINS 2023) : Proceedings of the 7th International Conference (Kharkiv, Ukraine, 20–21 april, 2023)*. CEUR Workshop Proceedings, 2023, Vol. 3403, pp. 53–68. Access mode: <http://ceur-ws.org/Vol-3403/paper5.pdf>.
- Shportko A. V., Bomba A. Ya. Formuvannja kolirnyh modelej z centrumannjam intervaliv riznych' komponentiv v procesi progresujuchogo ije-rarhichnogo stysnennja zobrazhen' bez vtrat. *Modeling, control and information technologies (MCIT-2023) : Proceedings of Vth International scientific and practical conference (Rivne, 9–11 november, 2023)*. Rivne, National university of water and environmental engineering, 2023, pp. 194–197. DOI: 10.31713/MCIT.2023.060.
- WinRAR download free and support, version 7.00 [Electronic resource], 2024. Access mode: <https://www.winrar.com/start.html?&L=0>.
- Boutell T. et. al. PNG Specification. Version 1.0. RFC 2083, Boutell. Com, inc. Mar. 1997, 102 p. DOI: 10.17487/RFC2083.

14. Shportko A. V., Postolatii V. A. Development of Predictors to Increase the Efficiency of Progressive Hierarchic Context-Independent Compression of Images Without Losses. *Computational Linguistics and Intelligent Systems (COLINS 2021) : Proceedings of the 5th International Conference (Kharkiv, Ukraine, 22–23 April 2021)*. CEUR Workshop Proceedings, 2021, Vol. 2870, pp. 1026–1038. Access mode: <http://ceur-ws.org/Vol-2870/paper77.pdf>.
15. Moffat A., Neal R. M., Witten I. H. Arithmetic coding revisited. *ACM Transactions on Information Systems*, 1998, Vol. 16(3), pp. 256–294. DOI:10.1145/290159.290162.
16. Bomba A. Ya., Shportko A. V., Postolatii V. A. Redistribution of the Compressed Data Between Modified DEFLATE-Blocks in the Image Compression Process Without Lossless. *Computational Linguistics and Intelligent Systems (COLINS 2024) : Proceedings of the 8th International Conference (Lviv, 12–13 Apr 2024)*. Volume II: Modeling, Optimization, and Controlling in Information and Technology Systems Workshop (MOCITSW). CEUR Workshop Proceedings, 2024, Vol. 3668, pp. 145–156. Access mode: <https://ceur-ws.org/Vol-3668/paper11.pdf>.
17. Repository links.uwaterloo.ca [Electronic resource]. Access mode: <https://links.uwaterloo.ca/Repository.html>.

Received 12.12.2025.

Accepted 15.04.2026.

Published 26.06.2026.

УДК 004.043

ВИКОРИСТАННЯ РІЗНИЦЕВИХ КОЛІРНИХ МОДЕЛЕЙ ЗІ ЗМІЩЕННЯМИ РІЗНИЦЬ КОМПОНЕНТІВ ДЛЯ ПОКРАЩЕННЯ СТИСНЕННЯ ЗОБРАЖЕНЬ У ФОРМАТІ PNG

Шпортко О. В. – канд. техн. наук, доцент, докторант центру аспірантури та докторантури Національного університету водного господарства та природокористування, Рівне, Україна. ROR: <https://ror.org/00ppfcz28>. ORCID: <https://orcid.org/0000-0002-4013-3057>.

Бомба А. Я. – д-р техн. наук, професор, професор кафедри комп'ютерних наук та прикладної математики Національного університету водного господарства та природокористування, Рівне, Україна. ROR: <https://ror.org/00ppfcz28>. ORCID: <https://orcid.org/0000-0001-5528-4192>.

АНОТАЦІЯ

Актуальність. На сьогодні для підвищення ефективності стиснення зображень без втрат в графічних форматах та архіваторах використовуються статичні різницеві колірні моделі з цілими коефіцієнтами. Ці моделі покращують стиснення, але не враховують рівні перехресної кореляції між різними парами компонентів кольорів пікселів кожного зображення. Тому розробка методів для формування та використання різницевих колірних моделей для окремих зображень з метою покращення їх компресії шляхом міжкомпонентної декореляції є на цей час актуальним науковим завданням.

Мета роботи – розробка методів та алгоритмів переходу до різницевих колірних моделей з цілими коефіцієнтами і зміщеннями різниць для зменшення коефіцієнтів стиснення в процесі компресії RGB-зображень без втрат у сучасних графічних форматах, зокрема у форматі PNG.

Метод. Залежно від обмежень на час кодування, в роботі запропоновано використовувати 4, 16, 19 чи 49 альтернативних різницевих колірних моделей зі зміщеннями різниць для вибору найефективнішої моделі для кожного зображення. Прогнозування ефективності стиснення внаслідок застосування чергової альтернативної різницевої колірної моделі виконується за допомогою ентропії. Зміщення різниць в колірних моделях виконується так, щоб центр інтервалу з максимальною кількістю цих різниць змістився до середини діапазону можливих значень. Досліджено ефективність трьох способів визначення центру цього інтервалу: без врахування відхилень яскравостей компонентів, з використанням різниці медіан компонентів та шляхом визначення центру інтервалу з максимальною кількістю різниць компонентів після їх послідовного перебору.

Результати. Наші експерименти показали, що, наприклад, застосування різницевих колірних моделей з цілими коефіцієнтами до цілих зображень в процесі послідовної компресії без втрат, зокрема, в модифікованому нами графічному форматі PNG, дає змогу зменшити коефіцієнти стиснення фотореалістичних зображень набору АСТ на 0.19–1.06 брб. Зміщення різниць на різниці медіан окремих компонентів чи центрування інтервалів різниць компонентів забезпечують додаткове зменшення коефіцієнта стиснення в середньому на 0.01–0.02 брб. Таким чином, різницеві колірні моделі з цілими коефіцієнтами та зміщеннями різниць дають змогу суттєво підвищити ефективність стиснення трикомпонентних фотореалістичних зображень без втрат у форматах, що використовують предиктори, і тому можуть бути впроваджені в наступні версії цих форматів на рівні стандартів.

Висновки. В графічних форматах для зменшення коефіцієнтів стиснення зображень без втрат крім декореляції даних окремих компонентів доцільно виконувати міжкомпонентну декореляцію шляхом переходу до різницевих колірних моделей з цілими коефіцієнтами зі зміщеннями різниць, які забезпечують швидке декодування. З метою максимізації зменшення коефіцієнта стиснення внаслідок застосування обраної колірної моделі середини інтервалів різниць базових компонентів R , G , B слід зміщувати до середини діапазону можливих значень. Коли для фотореалістичних зображень через жорсткі обмеження на час кодування або на розмір кодера неможливо виконати вибір різницевої колірної моделі з 49, 19 чи 16-ти альтернативних, то цей вибір слід виконувати серед трьох моделей: $G - R + 128$, $G, G - B + 128$; $R, R - G + 128$, $B - G + 128$, чи $R - G + 128$, $B - G + 128$, B . Для синтезованих зображень переходити до різницевих колірних моделей з цілими коефіцієнтами не доцільно.

КЛЮЧОВІ СЛОВА: стиснення зображень без втрат, графічний формат PNG, різницеві колірні моделі з цілими коефіцієнтами.

ЛІТЕРАТУРА

1. Wallace G. The JPEG still picture compression standard / G. Wallace // *Communication of ACM*. – 1991. – Vol. 34, № 4. – P. 30–44. DOI: 10.1145/103085.103089.
2. Shehata O. Unraveling the JPEG [Electronic resource] / O. Shehata // *Parametric Press*. – 2019. – № 1. Access mode: <https://parametric.press/issue-01/unraveling-the-jpeg>.
3. Guo T. A review of wavelet analysis and its applications: Challenges and opportunities / T. Guo, T. Zhang, E. Lim et al // *IEEE Access*. – 2022. – № 10. – P. 58869–58903. DOI: 10.1109/ACCESS.2022.3179517.
4. Selomon D. *A Guide to Data Compression Methods* / D. Selomon. – New York : Springer, 2002. – 295 p. DOI: 10.1007/978-0-387-21708-6.
5. Kotha H. D. Review on Lossless Compression Techniques / H. D. Kotha, M. Tummanapally, V. K. Upadhyay // *Journal of Physics*. – 2019. – Vol. 1228. – 6 p. DOI: 10.1088/1742-6596/1228/1/012007.
6. Miano J. *Compressed Image File Format: JPEG, PNG, GIF, XBM, BMP* / J. Miano. – New York: Addison Wesley, 1999. – 264 p. ISBN 0201604434.
7. Shannon C. E. A Mathematical Theory of Communication // *Bell System Technical Journal*. – 1948. – Vol. 27. – P. 379–423, 623–656. DOI: 10.1002/j.1538-7305.1948.tb00917.x.
8. Ziv J. A universal algorithm for sequential data compression / J. Ziv, A. Lempel // *IEEE Transactions on Information Theory*. – May 1977. – Vol. 23(3). – P. 337–343.
9. Huffman D. A Method for the Construction of Minimum Redundancy Codes / D. Huffman // *Proceedings of the IRE*. – 1952. – Vol. 40(9). – P. 1098–1101.
10. Shportko A. V. Programming the Formation of Difference Color Models for Lossless Image Compression / A. V. Shportko, A. Ya. Bomba, V. A. Postolatii // *Computational Linguistics and Intelligent Systems (COLINS 2023)* : Proceedings of the 7th International Conference (Kharkiv, Ukraine, 20–21 april, 2023). – CEUR Workshop Proceedings, 2023. – Vol. 3. – P. 53–68. Access mode: <http://ceur-ws.org/Vol-3403/paper5.pdf>.
11. Шпортко О. В. Формування колірних моделей з центруванням інтервалів різниць компонентів в процесі прогресуючого ієрархічного стиснення зображень без втрат / О. В. Шпортко, А. Я. Бомба // *Modeling, control and information technologies (MCIT-2023)* : Proceedings of Vth International scientific and practical conference (Rivne, 9–11 november, 2023). – Rivne : National university of water and environmental engineering, 2023. – P. 194–197. DOI: 10.31713/MCIT.2023.060.
12. WinRAR download free and support, version 7.00 [Electronic resource]. – 2024. – Access mode: <https://www.winrar.com/start.html?&L=0>.
13. Boutell T. et. al. PNG Specification. Version 1.0. RFC 2083 / T. Boutell et. al. – Boutell. Com, inc, Mar. 1997. – 102 p. DOI: 10.17487/RFC2083.
14. Shportko A. V. Development of Predictors to Increase the Efficiency of Progressive Hierarchic Context-Independent Compression of Images Without Losses / A. V. Shportko, V. A. Postolatii // *Computational Linguistics and Intelligent Systems (COLINS 2021)* : Proceedings of the 5th International Conference (Kharkiv, Ukraine, 22–23 April 2021). – CEUR Workshop Proceedings, 2021. – Vol. 2870. P. 1026–1038. Access mode: <http://ceur-ws.org/Vol-2870/paper77.pdf>.
15. Moffat A. Arithmetic coding revisited / A. Moffat, R. M. Neal and I. H. Witten // *ACM Transactions on Information Systems*. – 1998. – Vol. 16(3). – P. 256–294. DOI: 10.1145/290159.290162.
16. Bomba A. Ya. Redistribution of the Compressed Data Between Modified DEFLATE-Blocks in the Image Compression Process Without Lossless / A. Ya. Bomba, A. V. Shportko, V. A. Postolatii // *Computational Linguistics and Intelligent Systems (COLINS 2024)* : Proceedings of the 8th International Conference (Lviv, 12–13 Apr 2024). Volume II: Modeling, Optimization, and Controlling in Information and Technology Systems Workshop (MOCITSW). – CEUR Workshop Proceedings, 2024. – P. 145–156. Access mode: <https://ceur-ws.org/Vol-3668/paper11.pdf>.
17. Repository links.uwaterloo.ca [Electronic resource]. Access mode: <https://links.uwaterloo.ca/Repository.html>.