UDC 004.043

A. V. Shportko[1], O. V. Kapshii[2]
[1]Rivne State Humanitarian University,
[2]Karpenko Physico-Mechanical Institute of NASU

USING DIFFERENTIAL COLOR MODELS IN LOSSLESS RGB-IMAGES COMPRESSION TASKS

Запропоновано методи визначення оптимальної різницевої кольорової моделі у задачі безвтратного стиснення RGB-зображень із застосуванням предикторів, використання якої дає змогу покращити показники компресії. Наведено алгоритм розрахунку та переходу до різницевої кольорової моделі з цілими коефіцієнтами, результати перевірки роботи запропонованих методів. Як показали експерименти, використання розрахованих різницевих кольорових моделей дає змогу зменшити коефіцієнти стиснення зображень у середньому на понад 4,5 %.

Ключові слова: безвтратне стиснення, різницеві кольорові моделі, предиктори.

In the paper several methods of determination of optimal differential color model for RGB-images lossless compression task based on predictors use are considered. Such optimal differential model allows compression parameters to be improved. An algorithm for determination and using of differential color model with integer coefficients is shown. Experiments have shown that using of the proposed differential color models allows compression rate of images to be increased approximately for 4,5 %.

Key words: lossless compression, differential color model, predictors.

Introduction

A large part of all information transmitted by channels or saved at different memory devices is multimedia information. Images are an important part of it once. Correspondingly, a problem of effective compression of images stays urgent today and will be urgent in the nearest future. All image compression algorithms can be divided into two base classes: lossy and lossless. For most lossy algorithms needed compression coefficient (rate of compressed image size to original image size – CR) can be achieved at the expense of controlled quality degradation. But CR of lossless algorithms depends only on information at images and compression algorithm itself. The CR in such situation can not be adjusted and is equal usual only 30-70 %. Any compression is possible only if input data have information redundancy. This is a reason why improvement of lossless compression is achieved not only by development and improvement of compression algorithms, but by using preprocessing algorithms also (prepressing). Such algorithms regroup and modify input data that increases their redundancy level. In this paper a method of optimal selection of differential color model of input image and method of its parameters determination are presented. Using of such model allows input data redundancy to be increased and, correspondingly, CR of lossless compression algorithm is improved.

The problem state

When images are stored in a lossless file format, RGB color model with 8 bit quantization is used [1] more often. Such model corresponds to computer displays specific. In the model color of each pixel is a combination of red, green and blue components intensities. Each of the components is an integer [0, 255].

During lossless compression image pixels are usually processed by rows from top to bottom, and from left to right in each row. As a rule, during the compression, image data after prepressing is processed by context-dependent algorithm. The algorithm goal is to decrease redundancy among different parts of the image. After this context-independent algorithm is used that takes into account data elements appearance frequencies. Different compression algorithms use different context-dependent methods oriented, as a rule, only at specific images classes [2]. Taking this into account we have considered coding concept of context-independent algorithms and predictors use. These techniques are very often used in image preprocessing stages of modern lossless compression algorithms [3].

The idea of context-independent algorithms work is based on changing of more frequent elements by shorter codes. According to Shannon theorem about data source coding [2] it is the best to code element s_i with appearance frequency $p(s_i)$ by $-\log_2 p(s_i)$ bits. Then, after using context-independent algorithm, average length of element code will be close to *source entropy* [2, 4]

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

Source entropy will decrease when probability distribution of elements is less uniform [3]. For images s_i are quantized values of intensities of pixels components. Modern image lossless compressing formats use Huffman of arithmetic codes for context-independent compressing very often [2].

Let consider a data sequence where each element s_i appears N_i times. Length of the sequence is equal $N = \sum_i N_i$. Then $p(s_i) = N_i / N$ and general length of entropy code, taking into account (1), will be equal

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

Very often, during lossless image compression, entropy is decreased by using *predictors* [3]. Predictors use values of known nearest elements for prediction of current element value. For three-component color models a predictor of each component predict current value basing on corresponded components of nearest pixels. Then differences between predicted and real values are calculated and coded. The predictor use procedure can be described by expression

$$\Delta_{ij} = C_{ij} - predict_{ij} , \quad (3)$$

where (i, j) – coordinates of current pixel, C_{ij} – component value before predictor use (real value), Δ_{ij} – component value after predictor use, $predict_{ij}$ – predicted value. Because most of nearest pixels colors are close, after using (3) most of Δ_{ij} will be close to 0. Such redistribution of values frequency (and probability) increases irregularity of the distribution and decreases source entropy (1) and entropy code length (2). In such way, predictors can improve CR of context-independent algorithm [3].

In order to get good speed during decoding, in graphic files, nonadaptive predictors are used, as a rule. These predictors determine predicted value of pixel component using corresponded components of adjacent already coded pixels [1, 2]. Very often as predictors are used nonlinear predictors: *PaethPredict* – predicts value in direction of smallest component intensity change and *MedPredict*, which tries to adapt to local horizontal and vertical edges; and linear predictors: *LeftPredict* – predicted pixel component value is equal to corresponded component value of adjacent left pixel, *RightPredict* – predicted value is selected from adjacent top pixel, *AveragePredict* – prediction is equal to averaging of these values [3]. Linear predictors calculates changes of intensities in particular direction.

When images have to be stored in lossy format (e.g. in digital video [4] and formats JPEG, as a rule, color model YCrCb is used. The main part of image energy is concentrated into first component of the model, which allows less information for storing another two components to be used [1]. As a result CR is improved. Small energy components of color model YCrCb are created by linear combinations of differences of R, G, B components [4]. In our investigations we try to use linear combinations of differences of R, G, B components for decreasing of entropy during lossless images compression.

Modern lossless compression formats of RGB-images works with pixels in fixed color model. For example, BMP and PNG formats use RGB-model, format of WinRAR packer uses model R-G, G, B-G. But they do not use a possibility of selection of effective color model that can decrease entropy as much as possible taking into account correlation among different components of an image. Such decorrelation is possible because different components of an image with several components are similar by geometric-area structures. This is a reason why a **goal of this paper** is to describe and to prove an algorithm for building and using of alternative differential color models with nonadaptive predictors in lossless compression formats of RGB-images.

The investigation results

It is reasonable to change color model component by linear combination with another component only then, when length of an entropy code of the combination will be smaller then length of an entropy code of the component [5]. In order to decoding have to be unambiguous it is possible to use maximum two changes of different components by differences with other components at all pixels. As a result, a **task of selection among six possible changes** (R_{ij} by R_{ij}-kG_{ij} or R_{ij}-kB_{ij}, G_{ij} by G_{ij}-kR_{ij} or G_{ij}-kB_{ij} and B_{ij} by B_{ij}-kR_{ij} or B_{ij}-kG_{ij}) **by two of them which allow length of entropy code of results of selected predictor work to be maximal decreased**. Let consider approaches to composition of differential color models with real and with integer coefficients *k*.

Composition of differential color models with real coefficients. All explanations will be made using a R_{ij} component and its replacement by a linear combination R_{ij}-kG_{ij}. But all descriptions can be used to any component and its replacement by linear components combinations. So, changing of R_{ij} by linear combination R_{ij}-kG_{ij} for all image pixels is reasonable only when

$$W(k) = L(\Delta(R - [kG])) < L(\Delta R), \tag{4}$$

where $\Delta R = \{\Delta R_{ij}\}$, $\Delta(R - [kG]) = \{\Delta(R_{ij} - [kG_{ij}])\}$, $i = \overline{0, height - 1}$, $j = \overline{0, width - 1}$, *height* – image rows number, *width* – number of pixels in a row, a [.] – rounding to nearest integer, because of necessity to work with integer values of intensity. From (4) we can see that, for achieving maximum compression, value of *k* have to be selected in such way that length of entropy code of predicted values of linear combination of image pixels components *W*(*k*) have to be minimal. Let we get a minimum at $k = k_0$:

$$k_0 : L(\Delta(R - [k_0 G])) = \min_k W(k),\ k \in \Re . \tag{5}$$

Because linear combination of colors practically corresponds to another color, then (5) shows a process of searching a direction in *RG* space for which length of entropy code of predicted values is minimal. According to this expression k_0 can be determined using method of **sequential values examination**. In order to calculate length of entropy codes of components and their differences in (4) and (5) after using predictor we have to determine frequencies of individual elements and to use expression (2).

In order to avoid using of enumerative techniques or cumbersome interpolation algorithms and algorithms of searching a direction of k_0 values convergence we have proposed a method of rough determination of k_0 using variance of linear predictors' values. As it is shown in [4], "experiments with large numbers shows that distribution of values of Δ tends to Laplace distribution" which has zero mean value. Probability distribution of Laplace law is described by $f(x) = \frac{\lambda}{2} e^{-\lambda |x|}$ and irregularity of the distribution is increased with λ increasing. From other side λ can be expressed by variance of the distribution: $\lambda = 2 \big/ \sqrt{D(X)}$. If we approximate image's data after predictors use by Laplace distribution then with decreasing of variance irregularity of a distribution of components' values near zero is increased. This situation corresponds to decreasing of entropy code length. Then rough value of k_0 can be determined from condition

$$k_0 \approx \bar{k}_0 : D(\Delta(R - \bar{k}_0 G)) = \min_k V(k),\ k \in \Re , \tag{6}$$

where $V(k) = D(\Delta(R - kG))$. The variance under zero mean assumption can be determined from

$$D(\xi) = \left(\sum_{i=0}^{height-1} \sum_{j=0}^{width-1} \xi_{ij}^2 \right) \Big/ (height \times width). \tag{7}$$

Then, from (6) and (7) we get

$$V(k)=\frac{\left(\sum_{i=0}^{height-1}\sum_{j=0}^{width-1}\left(\Delta\left(R_{ij}-kG_{ij}\right)\right)^2\right)}{\left(height\times width\right)}\rightarrow \min. \quad (8)$$

In the numerator of (7) is an expression of *data energy* [4]. Expression (8) describe a process of searching of direction in *RG* space where energy of intensity changes is minimal. Let determines $\bar{k}_0$ using least-square method. Taking into account that for linear predictors $\Delta\left(R_{ij}-kG_{ij}\right)=\Delta R_{ij}-k\,\Delta G_{ij}$, we can equate a derivative of $V(k)$ from (8) to zero: $\frac{dV(k)}{dk}=-2\frac{\left(\sum_{i=0}^{height-1}\sum_{j=0}^{width-1}\left(\Delta R_{ij}-k\,\Delta G_{ij}\right)\Delta G_{ij}\right)}{\left(height\times width\right)}=0$, then

$$k_0\approx\bar{k}_0=\frac{\left(\sum_{i=0}^{height-1}\sum_{j=0}^{width-1}\Delta R_{ij}\,\Delta G_{ij}\right)}{\left(\sum_{i=0}^{height-1}\sum_{j=0}^{width-1}\left(\Delta G_{ij}\right)^2\right)}. \quad (9)$$

For continuous-tone images increasing/decreasing of intensity in adjacent pixels of one component as a rule follow to increasing/decreasing of intensities of other components. In other words, for such images almost all ΔR_{ij}, ΔG_{ij} and ΔB_{ij} have the same sign (their pair products are positive) and proportional values. So, it is worth to decrease a variance, first of all, for these pixels. Taking this into account, for continuous-tone images we can determine a rough value of k_0 by

$$k_0\approx\bar{\bar{k}}_0=\frac{\left(\sum_{i=0}^{height-1}\sum_{j=0}^{width-1}\Delta R_{ij}\,\Delta G_{ij}\mid\Delta R_{ij}\,\Delta G_{ij}>0\right)}{\left(\sum_{i=0}^{height-1}\sum_{j=0}^{width-1}\left(\Delta G_{ij}\right)^2\mid\Delta R_{ij}\,\Delta G_{ij}>0\right)}. \quad (10)$$

The expressions (9) and (10) calculate a relation of cross-correlation of intensities differences to variance of component differences accordingly for all pixels and pixels, which changes have the same type for analyzed components. After rough estimation of k_0 using (5), (9) or (10) it is important to check effectively of determined linear combination using expression (4).

Building differential color models with integer coefficients. Let consider an algorithm for building differential color models for k_0=1 and linear predictors, that increase irregularity of distribution mainly for account of adjacent pixels with close intensity differences for different components (such intensity changes in images are occurred very often). Parameters of the models with integer coefficients can be determined quickly, because there are no operations with real numbers and rounding. The same reasons allow decoding to be carried out faster. In the described situation it is necessary to estimate reasonability of replacement of R_{ij} component by differences R_{ij}-G_{ij} or R_{ij}-B_{ij}, G_{ij} - by G_{ij}-R_{ij} or G_{ij}-B_{ij} and B_{ij} - by B_{ij}-R_{ij} or B_{ij}-G_{ij}.

Let express investigated entropy lengths in a form of matrix:

$$A=\begin{pmatrix}a_{00} & a_{01} & a_{02}\\ a_{10} & a_{11} & a_{12}\\ a_{20} & a_{21} & a_{22}\end{pmatrix}==\begin{pmatrix}L(\Delta(R)) & L(\Delta(R)-\Delta(G)) & L(\Delta(R)-\Delta(B))\\ L(\Delta(G)-\Delta(R)) & L(\Delta(G)) & L(\Delta(G)-\Delta(B))\\ L(\Delta(B)-\Delta(R)) & L(\Delta(B)-\Delta(G)) & L(\Delta(B))\end{pmatrix}. \quad (11)$$

Taking into account (2) we can check that a_{mn}=a_{nm}, so for building the matrix we need to determine values of six components only. Let mark *R* component by 0, *G* by 1 and *B* by 2. We can see that differential color model is determined by maximum two nondiagonal elements of different rows of the matrix *A*. These elements, among elements smaller their diagonal elements of corresponded rows, in sum have to have a larger deviation (they provide maximal decreasing of length of entropy code). When such elements exist than their row index correspond to a decreased component and column index to a subtracted component. For example, selection of a_{12} element shows that in an alternative color model for each image pixel values of *G* component have tot be decreased by values of a *B* component.

An algorithm for building and using a differential color model with integer coefficients during image compression with predictors use can be written as followed:

1. For each component of all image pixels results of selected linear predictor use are determined;
2. Length of entropy codes of components and components differences processed by selected predictor are determined and stored into matrix *A* according to (11);
3. Variables *index11*, *index12*, *index21* and *index22*, that correspond to possible differences of alternative color model, are set to zero;
4. Two elements of matrix *A* that do not lie onto main matrix diagonal, do not symmetric relative to it, are smaller than diagonal elements of their rows and in sum have a larger deviation from them are determined. If such elements present in the matrix, than variables *index11* and *index12* are assigned by corresponded row and column numbers of the first element and variables *index21* and *index22* are assigned by row and column numbers of the second element. Else nondiagonal element of matrix *A* that is smaller than a corresponded diagonal element of its row is determined. If such element is absent than we go to step 8 of the algorithm, else variable *index11* is assigned row number of the element and variable *index12* is assigned its column number;
5. If *index11* and *index22* are equal than values of *index11* is interchanged with *index21* and values of *index12* is interchanged with *index22*. In this way we change a sequence of subtraction of color model's components;
6. For each image pixel value of a component with number *index11* is decreased by value of a component with number *index12*;
7. If *index21* and *index22* are different (second difference of the color model is determined) than for each image pixel a value of a component with number *index21* is decreased by a value of a component with number *index22*;
8. This is the end of the algorithm where differential color model is built and used and transformed image data are ready to following compression.

For our experiments we have selected a linear predictor *LeftPredict* because it uses only value of a component of previous pixel of current row and is determined faster than other. But other predictors (linear or nonlinear) can be used also. It is worth to note that, firstly, during decoding components values are added in a reverse sequence: values of *index22* component are added to values of *index22* component and, only after this, values of *index12* component are added to values of *index11* component, and secondly, if during coding predictors use is refused than alternative color model has to be also refused, because it is oriented on predictors use.

Experiments results

Let analyze results of using proposed methods for compression of eight files of 24-bits images from standard ACT set. Main characteristics of these images are shown in table 1 (% of unique colors - it is a ratio of number of unique or different colors of pixels to number of all image pixels). The images in TIFF format can be downloaded for example from http://compression.ca/act/act-files.html. The set contains artificial (№№ 1, 2, 7) and natural images. Testing was carried out using own program created basing on program from CD [1]. The program compresses images in PNG format by following algorithm: building and using differential color model; breaking image into blocks of uniform rows; processing of each row pixels by selected predictor; compressing image by context-dependent dictionary algorithm LZ77 [6] that codes replaces; compression of data by context-independent Huffman prefix codes [2, 4]. Determined by different methods differential color models are shown in table 2, compression rate coefficients - in table 3, coding time - in table 4 and decoding time - in table 5.

Real coefficients k_0 for differential color models are determined in three ways (see tables 2-4): using an enumerative technique (columns 2, 3) according to (5); using a variances of intensities differences of all pixels (column 4) according to (9) and using variances of intensities differences of pixels with same sign (column 5) according to (10). Interval of values for enumerative technique is selected (0; 1.5] with step 0.05, because there are too little meaningful images with intensities differences of different components that mainly have different signs or mainly differ more than 1.5 times. For building differential color models we have used results of using nonlinear predictor *PaethPredict* (column 3) and linear predictor *LeftPredict* (columns 2, 4, 5). Average decoding time with using these methods is shown in table 5, column 2.

Table 1

Characteristics of selected images of ACT set

№ file	File name	Size, Kb	Number of unique colors	% of unique colors	Features
1	Clegg.bmp	2101	127696	17.83	Continuous-tone, artificial, noisy, several big objects
2	Frymire.bmp	3622	3622	0.29	Discrete-tone, artificial, one big object
3	Lena.bmp	769	148279	56.56	Continuous-tone, natural, several big objects
4	Monarch.bmp	1153	78617	19.99	Continuous-tone, natural, one big and several small objects
5	Peppers.bmp	769	111344	42.47	Continuous-tone, natural, several big objects
6	Sail.bmp	1153	75748	19.26	Continuous-tone, natural, many medium-sized objects
7	Serrano.bmp	1464	1313	0.26	Discrete-tone, artificial, one big fragmented object
8	Tulips.bmp	1153	118233	30.07	Continuous-tone, natural, several big objects

Table 2

Determined differential color models for ACT set images

№ file	A method of building differential color model				
	Using LeftPredict, enumerative technique	Using PaethPredict, enumerative technique	Using LeftPredict according (9)	Using LeftPredict according (10)	Using LeftPredict for k_0=1
1	R,G-R,B	R,G-R,B	R,G-0.31R,B-0.16R	R,G-0.79R,B	R,G-R,B
2	R-B,G-B,B	R,G-B,B-R	R,G,B	R,G-0.88B,B-0.93R	RGB
3	R,G-0.85R,B-0.75G	R,G-0.85R,B-0.75G	R,G-0.85R,B-0.7G	R,G-1.12R,B-0.83G	R,G-R,B-G
4	R-0.85B,G,B-0.95G	R,G-R,B-G	R-0.84B,G,B-0.83G	R-0.98B,G-1.01B,B	R,G-B,B-R
5	R,G-R,B-0.9G	R-0.75G,G,B-0.95G	R-0.68G,G,B-0.79G	R,G-R,B-0.91G	R,G-R,B-G
6	R-G,G,B-G	R-G,G,B-G	R-0.98G,G,B-0.93G	R-0.99G,G,B-0.96G	R-G,G,B-G
7	R-G,G-B,B	R-G,G-B,B	R,G,B	R,G-0.89R,B	RGB
8	R,G-R,B-G	R,G-R,B-G	R,G-0.93R,B-0.96G	R,G-0.99R,B-G	R,G-R,B-G

Table 3

Compression rate of PNG images of ACT set for different color models, %

№ file	Differential color models					RGB
	Using *LeftPredict*, enumerative technique	Using *PaethPredict*, enumerative technique	Using *LeftPredict* according (9)	Using *LeftPredict* according (10)	Using *LeftPredict* for k_0=1	
1	20.85	20.85	20.89	20.80	20.85	20.89
2	6.90	6.76	6.74	6.82	6.74	6.74
3	58.26	58.26	58.00	58.52	58.78	60.21
4	44.67	44.41	45.10	44.67	44.41	52.04
5	47.72	47.59	47.85	47.72	47.72	53.84
6	52.73	52.82	53.43	52.99	52.73	65.65
7	7.10	7.10	7.10	7.17	7.10	7.10
8	48.57	48.57	48.92	48.57	48.57	57.50
Average	**35.85**	**35.79**	**36.00**	**35.89**	**35.86**	**40.50**

Table 4

Coding time (include transition to model) of PNG images of ACT set for different color models using computer with 300MHz central processor frequency, s

№ file	Differential color models					RGB
	Using LeftPredict, enumerative technique	Using PaethPredict, enumerative technique	Using LeftPredict according (9)	Using LeftPredict according (10)	Using LeftPredict for k_0=1	
1	461.76 (413.26)	845.30 (796.14)	60.47 (11.04)	58.67 (9.77)	51.02 (1.26)	49.48
2	779.77 (711.28)	1434.22 (1367.81)	81.68 (14.77)	84.64 (18.64)	71.52 (2.47)	68.27
3	171.64 (151.32)	310.72 (291.21)	24.66 (4.12)	24.44 (4.23)	20.71 (0.61)	20.26
4	275.73 (227.89)	483.18 (436.71)	54.38 (6.09)	54.59 (6.26)	49.77 (0.82)	42.40
5	180.16 (151.37)	319.83 (290.99)	33.45 (4.12)	33.06 (4.34)	29.06 (0.55)	26.14
6	265.01 (226.78)	476.36 (438.52)	43.94 (6.15)	44.71 (6.34)	39.88 (0.83)	29.60
7	315.99 (287.70)	581.60 (553.59)	34.11 (5.99)	34.06 (6.65)	29.66 (0.99)	28.45
8	266.88 (226.79)	477.75 (438.42)	45.64 (6.04)	46.02 (6.27)	41.42 (0.88)	37.46
Average	**339.62 (299.55)**	**616.12 (576.67)**	**47.29 (7.29)**	**47.52 (7.81)**	**41.63 (1.05)**	**37.76**

Table 5

Decoding time (include transition from model) of PNG images of ACT set for different color models using computer with 300MHz central processor frequency, s

№ file	Differential color models		RGB
	All models with real coefficients	Using *LeftPredict* for k_0=1	
1	6.54 (2.53)	4.01 (0.33)	3.84
2	4.23 (0.00)	4.01 (0.00)	4.01
3	3.46 (0.88)	2.36 (0.17)	2.31
4	4.61 (1.43)	3.29 (0.16)	3.13
5	3.19 (0.93)	2.31 (0.16)	2.25
6	4.89 (1.43)	3.57 (0.17)	3.57
7	1.65 (0.00)	1.59 (0.00)	1.59
8	4.94 (1.43)	3.46 (0.17)	3.46
Average	**3.57 (1.07)**	**3.08 (0.15)**	**3.02**

Determined differential color models with integer coefficients (k_0=1) according to described algorithm and results of their using are shown in column 6 of tables 2-4 and corresponded decoding time - in column 3 of table 5. For comparison results of compression images from ACT set without using differential color models are shown in column 7 of table 3-4 and column 4 of table 5.

As can be seen from columns 2, 3 of tables 3 and 4 it is worth to use linear predictors for building differential color models. In spite of worth (0.06 % in average) compression rate in comparison to using nonlinear predictors they allow approximately two times faster coding to be achieved. Enumerative techniques are not effective, because they are more than nine times slower than other methods, in spite of better (a part of percent) compression rate.

Among all considered methods of building differential color models **the most effective is the method with integer coefficients (k_0=1) and linear predictors**. It shows the best in average compression rate, the fastest coding and decoding. Let consider the results of this method more precisely. As it can be shown from table 2 for compressing different images we need to use different color models. During compression of images № 2 and № 7 predictors use has been unnecessary and RGB color model has been used. The most often in created differential color models values of B and G components are decreased, because they have, as a rule, the largest length of entropy code of predictors use results. Average compression rate at images of ACT set has been decreased at 4.64 % mainly for account of natural images when differential color models have been used. Similar compression rate decreasing has been obtained for other images and other formats (JPEG-LS, BMF) that do not take into account correlation among images' components. Analyzing tables 1 and 3 we can show that effectiveness of using differential color models for natural images increases when % of unique colors is decreased. Using differential color models with integer coefficients can enlarge coding time (up 35 %) because algorithms needs time for determination of color models parameters and because using context-independent algorithm which works with independent elements. But decoding time of algorithms based on differential color models use is increased insignificantly. Additionally taking into account improvement of compression rate we can affirm that proposed algorithms are effective and can be used in practice.

Conclusions

1. Compression rate of images based on three-components color models can be improved not only taking into account correlation of data of independent components, but taking into account inter-components correlation.

2. When nonadaptive predictors are used for image lossless compression using of differential color models can improve compression rate at above 4.5 % in average. For natural images compression rate can be improved at above 12 % (e.g. image № 6 at table 3). Effectiveness of using differential color models for natural images increases with decreasing % of unique image colors. Building differential color models correspond to searching colors that have minimal energy of color differences. In practice the best is using differential color models with integer coefficients. Although the models do not have the best compression rate but thet provide the best coding and decoding speed.

3. Differential color models give a possibility to improve effectiveness of lossless compression of three-component natural images in formats that use nonadaptive predictors. Therefore the models can be included into next versions of the formats up to standards.

In a future, for obtaining better compression rate, it is worth to develop an algorithm of image partitioning to areas of the same differential color models and an algorithm of compact storing of the areas.

1. Miano J. Image Compressed File Formats: JPEG, PNG, GIF, XBM, BMP / J. Miano. – M: Triumph, 2003. – 336 p. 2. Methods Compression Data. Device of Archivings, Compression of Images and Video / D. Vatolin, A. Ratuschnyak, M. Smirnov, V. Yukin. – M: DIALOG-MIFI, 2003. – 384 p. 3. Bredichin D. Y. Compression of Graphic Arts without the Losses of Quality [Electronic resource] / D. Y. Bredichin . – 2004. – http: //www.compression.ru/ download/articles/i_lless/bredikhin_2004_-lossless_image_compression_doc.rar. 4. Salomon D. Data Compression Methods / D. Salomon. – M: Technosphera, 2006. – 368 p. 5. Shportko A. V. Using of difference coloured models is for lossless compression of RGB-images / A. V. Shportko // Information Extraction and Proces. – 2009. – Vol. 31 (107). – P. 90–97. 6. 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.