BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXII No. 2/2010	1 - 12	Seria Matematică - Informatică - Fizică

Binary Image Compression Based on Binomial Numbers

Olexiy Borysenko, Igor Kulyk, Sergiy Kostel, Olena Skordina

Sumy State University, Electronics and Computer Techniques Department, Sumy, Ukraine
e-mail: electron@sumdu.edu.ua

Abstract

In this paper we present a method of information compression that is based on the transition from a binary sequence to its serial number in the binomial number system. At the first stage of the binomial compression the transition from an even-weight code sequence to a corresponding binomial number is carried out, then at the second stage the binomial number is transformed to its serial number accordingly the numerical function of the binomial number system with the given parameters. Before the binomial compression ordinary binary sequences are accounted even-weight code combinations since computing their number of binary units. In the paper the initial binary sequences are strings or substrings, columns or subcolumns of binary images. The paper expounds the method of binary images compression based on the binomial numbers. The considered method has simple image compression and decompression algorithms, which are presented in detail in the paper. An estimation of the compression method is implemented. The presented example shows how to compress a binary image with the help of the method of compression based on binomial numbers, generated by binomial number systems.

Key words: *binomial number systems, binary images, binomial compression technique*

Introduction

Digital images' processing plays an important role in various information systems such as telecommunication systems, machine vision systems, systems of science research automation and so on. It is important to note that images are a special kind of data type that is characterized by significant information redundancy. Without special processing they occupy a lot of memory space and need a large time when transmitting over communication channels. That is why different compression techniques are widespread. At present time, there are a lot of well-known methods of image compression, and each of them has its own virtues and shortcomings, as well application areas [7, 8].

The need for processing binary (bi-level) images results from the necessity of simplifying and optimizing problem solving that involves real ternary or gray-scale images. Such approaches are typical for machine vision systems, radar stations, optical character and drawing recognition systems and others. Every binary raster image has only two levels of brightness, usually denoted by 0 and 1 [6]. Thereby development of methods for compressing bi-level images is of an important enough and actual problem. An example of such a binary raster image is represented in Figure 1. This is a graphical representation of the character string "BMIF" and its equivalent representation in the binary sequences form.

In terms of a compression ratio it is effective enough to compress images by such well-known techniques as fractal compression, DjVu, JBIG2, method of principal components and so on. But their application reduces to information loss, that is unacceptable for a lot of information systems. For example, these are systems of automatic control or machine vision, where psychophysiological factors of information perception are minimized, or systems of science research automation, where value of information is difficult to determine at stages of data acquisition and preprocessing.

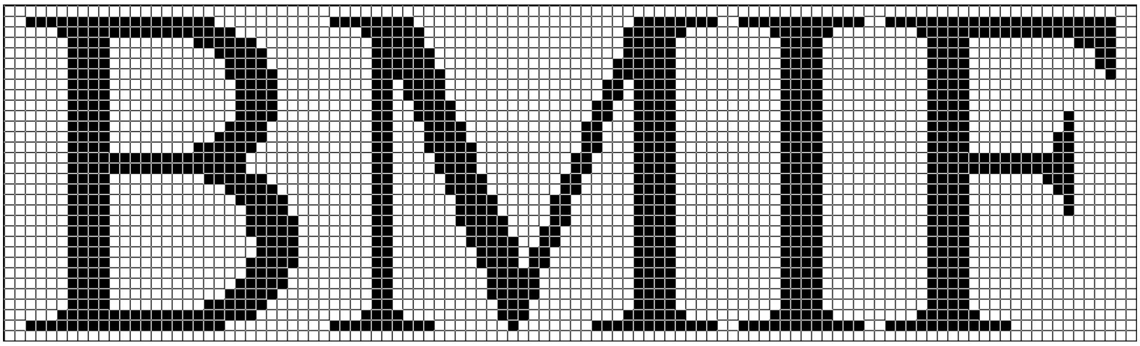

Figure 1: Example of graphical representation of the image of "BMIF"

On the other part application of lossless compression techniques such as Shannon-Fanos' and Huffman's methods demands exact knowledge on statistic properties of an information source. But it is complicated to realize in practice. For example, to receive a set of exact statistical data is of a difficult problem for such information systems as switching systems and systems of multimedia data processing, where there are some information sources with variable parameters. As a result efficiency of the statistical methods of compression is decreased and in some cases the decrease is significant.

As a rule the mentioned shortcomings aren't inherent for the compression techniques based on enumerative coding. Generally, enumeration methods use non-statistical models of information sources. One of the perspective compression technique is the method of binomial compression, using or binary, or multiple-valued binomial number systems [1, 2]. As it has be noted, we consider compression of binary raster images, therefore the binary binomial number systems and their numbers are used.

The method of binomial compression, based on binary binomial numbers, is described in [3], but as compared with that paper, we present: 1) more widened connection between theory of binary binomial number systems and the proposed compression technique; 2) compression and decompression algorithms in a more detailed form; 3) an example of an raster image, for which the binomial compression technique is very effective in comparison with the other lossless compression methods for images.

The paper proposes a method of lossless compression of binary images based on binomial numbers. The method uses binomial numbers for enumeration of binary sequences. The proposed compression method has simple hardware implementation and it is effective enough in terms of compression ratio for majority of binary images. Simultaneously with compression of an image it is possible to protect it by removing keys when restoring the compressed image.

First, the paper overviews some aspects of theory of binomial number systems. Numeric function and systems of code-forming contingencies for a binomial number system, as well some properties of binary binomial numbers, are presented in the first section. The second part of the paper describes compression and decompression algorithms for binary images. Flowcharts of the detailed algorithms are shown on Figure 2 and Figure 3. The third part of the paper contains numeric features for calculation of the size and compression ratio of a binary image. Next part of the paper illustrates the application of our binary image compression for the image, which is presented in Figure 1. The last section summarizes the aspects that are presented in this paper with regard to the compression techniques based on binomial numbers.

Basis of Binary Binomial Number Systems

Binomial number systems are called positional number systems with binomial coefficients in the capacity of basis [1, 2]. The linear binomial number system with a binary alphabet and binomial numbers are used in the paper. Numeric function and systems of code-forming contingencies have the following form:

$$F = x_{r-1}C_{n-1}^{k-q_{r-1}} + \ldots + x_i C_{n-r+i}^{k-q_i} + \ldots + x_1 C_{n-r-1}^{k-q_1} + x_0 C_{n-r}^{k-q_0} \ , \quad (1)$$

$$\begin{cases} k \le r \le n-1 \\ q = k \\ x_0 = 1 \end{cases} , \quad (2)$$

$$\begin{cases} n-k = r-q \\ 0 \le q \le k-1 \\ x_0 = 0 \end{cases} , \quad (3)$$

where r is a number of digits in a binomial number (length), $r \in 1,2,\ldots$;

k is the maximum number $q_{\max}$ of 1's in a binomial number;

i is a position index, $i = 0,1,\ldots,r-1$;

x_i is a value of a binary digit in a binomial number – 0 or 1;

n is a integer parameter of a binomial number system;

q is a number of 1's in a binomial number;

q_i is a sum of 1's digits x_i from the $(r-1)$ th digit to the $(i+1)$ th one inclusive:

$$q_i = \sum_{j=i+1}^{r} x_j \ ; \quad (4)$$

$i = 0,1,\ldots,r-1$; $x_r = 0$.

Binomial coefficient $C_{n-r+i}^{k-q_i}$ is a weighting coefficient of the i position in the numeric function (1). It depends on both the index position $i = 0,1,\ldots,r-1$ and the sum q_i of 1's digits x_i. The latter dependence is typical for the structural number systems and gives them noise immunity, as well as structure-forming properties.

From the contingencies (2) and (3) the following numerical properties are derived:

- the maximum length of a binomial code combination $r_{\max} = n-1$;
- the minimum length of a binomial code combination $r_{\min} = n-k$;
- the maximum number of 1's $q_{\max} = k$;
- the number of code combinations $P = C_n^k$.

Binary binomial numbers are of variable-length prefix code. One of the important properties of binary binomial numbers is their average code length L_{av}, which is obtained as it follows [4]:

$$L_{av} = \frac{k(n-k)(n+2)}{(k+1)(n-k+1)} \quad . \tag{5}$$

In the capacity of an example let us consider binary binomial numbers and their properties at the parameters of $n=6$ and $k=4$. In this case the numeric function (1) and systems (2, 3) of code-forming contingencies have the following form:

$$F(6,4) = \sum_{i=0}^{r-1} x_i C_{6-r+i}^{4-q_i} \quad ,$$

$$\begin{cases} 4 \le r \le 5 \\ q = 4 \\ x_0 = 1 \end{cases} \quad \text{and} \quad \begin{cases} r - q = 2 \\ 0 \le q \le 3 \\ x_0 = 0 \end{cases} .$$

At the parameters of $n=6$ and $k=4$ for binary binomial numbers the following numerical properties are obtained from the received contingencies:

- the maximum length of a binomial code combination $r_{\max} = 5$;
- the minimum length of a binomial code combination $r_{\min} = 2$;
- the maximum number of 1's $q_{\max} = 4$;
- the number of code combinations $P = C_6^4 = 15$.

According to the expression (5) the average code length L_{av} are as follows:

$$L_{av}(6,4) = \frac{4(6-4)(6+2)}{(4+1)(6-4+1)} \approx 4{,}27 \ .$$

Table 1 demonstrates the set of binary binomial variable-length numbers, which is ordered by way of increase of its elements at $n=6$ and $k=4$. On the left of binomial numbers their indexes are allocated in the table, which are calculated with the numeric function $F(6,4)$.

Table 1. Binary binomial numbers with their indexes, when $n=6$ and $k=4$

Index	Binomial number	Index	Binomial number
0	00	8	10111
1	010	9	1100
2	0110	10	11010
3	01110	11	11011
4	01111	12	11100
5	100	13	11101
6	1010	14	1111
7	10110		

The numeric function (1) of the binomial number system with a binary alphabet and properties of the binary binomial numbers found the proposed compression techniques for binary images.

Compression and Decompression Algorithms for Binary Images

The method of binary images compression is based on the transition from the binary code combination to its sequence number (1) in the binomial number system with the parameters of n and k. The process of this transformation is shown by using the algorithm of binary image compression on the basis of the binomial numbers, which is shown in Figure 2.

At the initial stage, the number k of units is counting in the binary sequence, which is represented as a string or substring of the image. Thus, the binary sequence to be compressed is expressed as an equilibrium code combination of length n with weight (number of 1's) k.

The next step is the transformation of the equilibrium code combination to the corresponding binomial one. The transformation is that all binary units from the right in equilibrium code sequence are thrown off till the first 0 appears or all binary 0's from the right are thrown off till the first 1 appears. The result of such an operation is the binomial number with parameters of n and k of the binomial number system.

The last step of the algorithm is the transition from the binomial code to its sequence number. To realize such a transition to the serial number it is necessary to use the numeric function (1), computing and adding the values of binomial coefficients with their parameters depending on values of n, k and q_i.

When the set of binary sequences is compressing, the parameter n (quantity of processed bits per algorithm cycle) is set as a constant, included in the compressing and decompressing algorithm, or selected according to the image to be compressed. As a rule it is stored at the beginning of the compressed file.

For unequivocal reconstruction of the compressed file when the value of n is constant it is necessary to know two values. The first value is the number k of 1's, which is calculated at the beginning of each cycle of the algorithm for the n-digits binary sequence. The second one is the serial number of the binomial code with the parameters n and k.

Decompression process consists of simple enough operations too. It is shown in Figure 3.

The number k of 1's and the serial number N are read from the compressed file. Then the binomial coefficients are selected with their consequent subtraction from the sequence number. The result of these steps is a binomial combination. After supplementing 0's or 1's to the combination it becomes the required n-digit binary sequence from the initial binary images.

Estimation of Compression Method

The number γ_i of digits needed for storing i-th sequence number is determined by the total number of code combinations of the binomial number system with parameters n and k_i:

$$\gamma_i = \lceil \log_2 P \rceil = \left\lceil \log_2 C_n^{k_i} \right\rceil, \tag{6}$$

where k_i is a number of 1's for the i-th binary sequence.

In addition to γ_i digits it is need to keep the number k of 1's. The number q of digits, allocated to store this parameter, has a constant value:

$$q = \lceil \log_2 (n+1) \rceil. \tag{7}$$

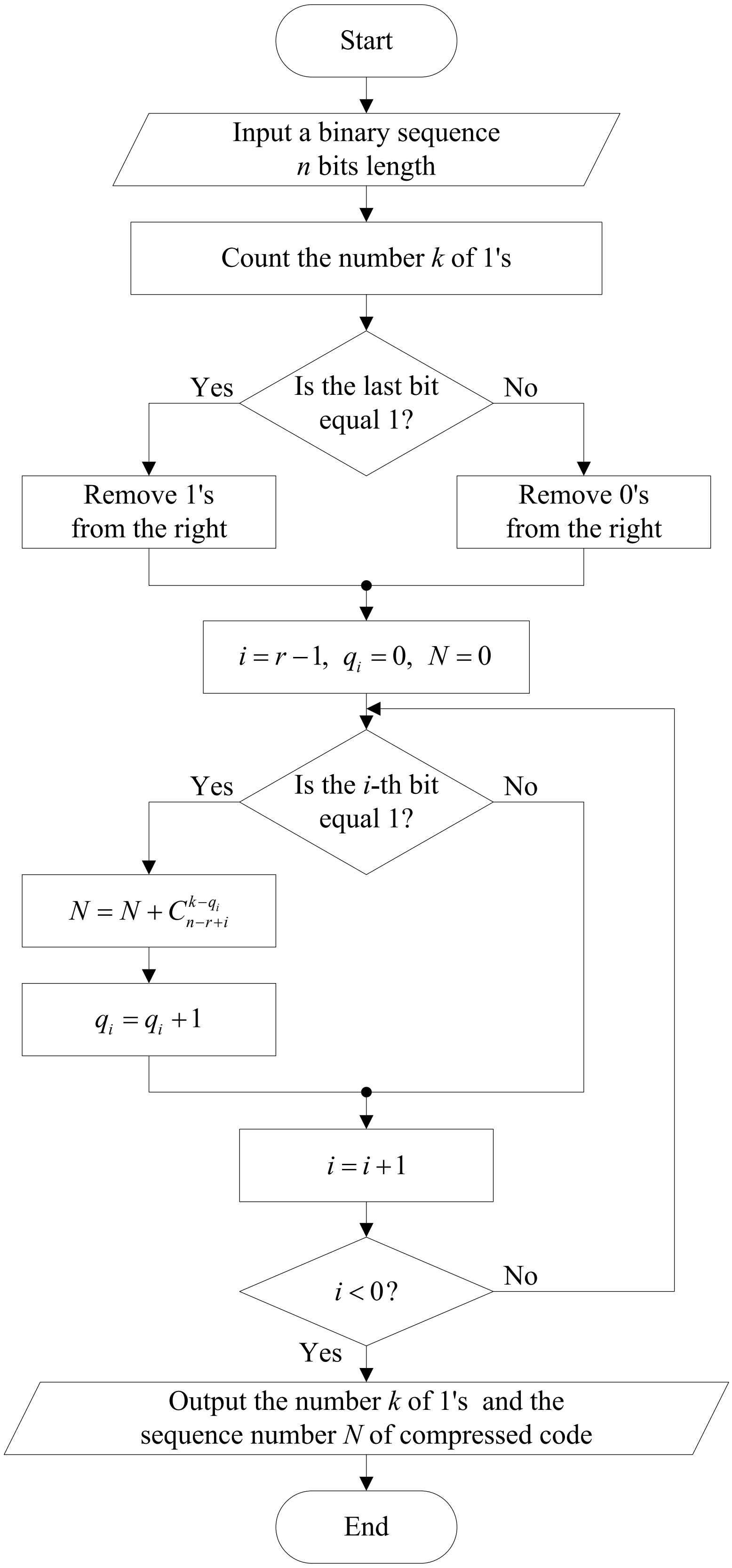

Figure 2: Algorithm of binary image compression based on the binomial numbers

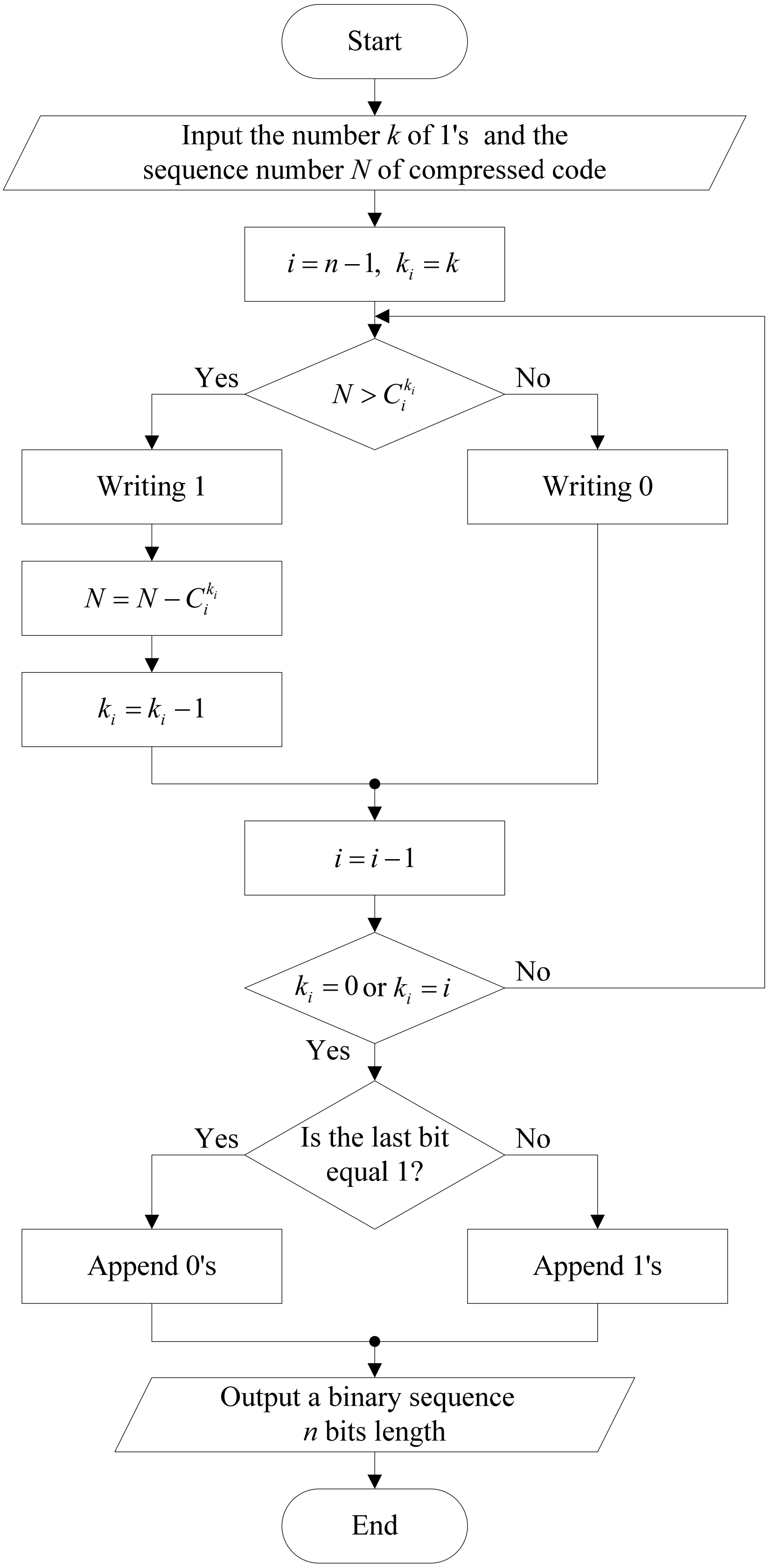

Figure 3: Algorithm of binary image decompression based on the binomial numbers

The total length l_i of the compressed binary sequence is

$$l_i = q + \gamma_i = \lceil \log_2(n+1) \rceil + \lceil \log_2 C_n^{k_i} \rceil . \quad (8)$$

As stated in the formula (8), the more difference there is between the value of k_i and the middle length $n/2$, the less the value of $\lceil \log_2 C_n^{k_i} \rceil$ and length l_i of the compressed binary sequence are. The binary sequence of length n, which contains the black or white pixels only, is compressed into the binary one of length $\lceil \log_2(n+1) \rceil$.

Compression ratio K_i of a binary sequence is calculated by the formula below:

$$K_i = \frac{n}{l_i} = \frac{n}{\lceil \log_2(n+1) \rceil + \lceil \log_2 C_n^{k_i} \rceil}, \quad (9)$$

A binary image is divided into n-digits sequences in number S shown beneath:

$$S = \frac{H \cdot W}{n}, \quad (10)$$

where H is a height of a binary image in pixels (bits), and W is a width of a binary image in pixels (bits).

The size of a compressed image can be calculated by the formula for the total length L of all the compressed binary sequences

$$L = S \cdot \lceil \log_2(n+1) \rceil + \sum_{i=1}^{S} \lceil \log_2 C_n^{k_i} \rceil . \quad (11)$$

The total compression ratio of a binary image using the considered method based on the binomial numbers is as follows:

$$K = \frac{H \cdot W}{S \cdot \lceil \log_2(n+1) \rceil + \sum_{i=1}^{S} \lceil \log_2 C_n^{k_i} \rceil} . \quad (12)$$

For images which consists of only black or white pixels the maximum compression ratio is derived as it is shown in the expression below:

$$K_{\max} = \frac{n}{\lceil \log_2(n+1) \rceil} . \quad (13)$$

The minimum compression ratio is obtained as it follows:

$$K_{\min} = \frac{n}{\lceil \log_2(n+1) \rceil + \lceil \log_2 C_n^{n/2} \rceil} . \quad (14)$$

For example, for $n = 512$ digits the best (13) and the worst (14) image compression ratios are, respectively, as following:

$$K_{\max} = \frac{512}{10} = 51{,}2 \; ; \qquad K_{\min} = \frac{512}{10+508} = \frac{512}{517} = 0{,}988 .$$

Example

Consider an example of applying the method of binary image compression introduced in this paper for the image, which is presented in Figure 1. The size of the image is 108×32 pixels. Consequently, the parameters are $W = 108$, $H = 32$. After analyzing the image of the symbol string "BMIF", it can be concluded that the largest compression ratio turns out when the columns are selected as the sequences for compression, but not strings.

Choose the length of the compressed blocks $n = 32$ bits. This parameter is the same for compression and decompression algorithms.

Let us have a look, how the algorithm of binary image compression based on the binomial numbers compress the sixth column of the image on Figure 1. The main steps of compression are as follows:

1. Read the 32-bits binary sequence: 01100000000000000000000000000110.
2. Count the number of 1's: $k = 4$.
3. The last bit of the sequence is 0, so we remove all right 0's up to the first 1. The result of the step is $r = 31$ bits binomial number: 0110000000000000000000000000011.
4. Set the parameters: $i = r - 1 = 30$, $q_i = 0$, $N = 0$.
5. Calculate the parameters of the binomial coefficients (see Table 2).

Table 2. The parameters of the binomial coefficients for the 32-bits binary sequence

Binomial number	0	1	1	0	0	…	0	0	1	1
i	30	29	28	27	26	…	3	2	1	0
q_i	0	0	1	2	2	…	2	2	2	3

6. Calculate the sequence number (index), using (1):

$$N = C_{32-31+29}^{4-0} + C_{32-31+28}^{4-1} + C_{32-31+1}^{4-2} + C_{32-31+0}^{4-3} = 27405 + 3654 + 1 + 1 = 31061_{<10>}.$$

7. The number of bits needed to store the number of 1's k and the sequence number N according to the expressions (6) and (7):

$$q = \lceil \log_2 (32+1) \rceil = 6, \ \gamma_i = \lceil \log_2 C_{32}^4 \rceil = \lceil \log_2 (35960) \rceil = 16.$$

8. The result of the compression for the initial 32-bits binary sequence is 0001000111100101010101 (the subsequence, beginning with the 7th bit inclusive to the last on the right, is the sequence number 31061 in the form of the corresponding binary number).

The results of the compression algorithm for all image blocks are presented in Table 3.

The main steps of decompression based on binomial numbers are as follows:

1. Read $q = 6$ bits of the compressed file. It is the number of 1's: $k = 000100_{<2>} = 4_{<10>}$.
2. Calculate the number of bits needed to store the sequence number N:

$$\gamma_i = \lceil \log_2 C_{32}^4 \rceil = \lceil \log_2 35960 \rceil = 16.$$

Table 3. The results of the compression algorithm based on the binomial numbers

Number of column	k_i	Binomial code	N_i	γ_i
1, 2, 29-31, 69, 70, 83, 84, 107, 108	0		0	0
3-5, 32-34, 66-68, 71-73, 80-82, 85-87	2	0100000000000000000000000000001	436	9
6, 35, 65, 74, 79, 88	4	0110000000000000000000000000011	31061	16
7-10, 36, 37, 61-64, 75-78, 89-92	30	0111111111111111111111111111111	30	9
11-18, 93	6	0110000000000000110000000000011	714261	20
19	8	0110000000000000111000000000111	7421430	24
20	10	0111000000000000111000000000111	43187290	26
21	11	011100000000000011111000000111	81588309	27
22	15	001110000000000111111100011111	137751782	30
23	20	00111100000000111111111111111	24734248	28
24	23	00111111001111110011111111111	1712025	25
25	22	0001111111111111100011111111	1550054	26
26	16	000011111111111100000011111	30416028	30
27	9	00000111111111	4686824	25
28	5	000000011111	53129	18
38	8	0110000000000000000000000111111	7413711	24
39	9	0100000000000000000000011111111	14307158	25
40	10	010000000000000000000111111111	30045069	26
41	11	01000000000000000001111111111	54628300	27
42	10	0000000000000000011111111111	8007	26
43	9	0000000000000000111111111	48619	25
44	10	000000000001111111111	352715	26
45	10	000000000111111111	1144065	26
46	9	0000000111111111	2042974	25
47	10	00001111111111	13123109	26
48	10	001111111111	30045014	26
49	9	0111111111	20160074	25
50	5	0011111	142505	18
51	5	000011111	98279	18
52	4	00000001111	12649	16
53	5	00000000011111	33648	18
54	5	0000000000011111	20348	18
55	4	000000000000001111	3059	16
56	5	0000000000000000011111	4367	18
57	6	010000000000000000011111	595776	20
58	5	01000000000000000000001111	142835	18
59	5	010000000000000000000001111	142631	18
60	7	01100000000000000000000011111	2510840	22
94-96	5	0100000000000000110000000000011	144237	18
97-99	4	0000000000000000110000000000011	1731	16
100	5	0000000000000001110000000000011	6099	18
101	8	0000000000000011111100000000011	43694	24
102	12	0000000000001111111111000000011	125927	28
103,104	3	0000000000000000000000000000111	3	13
105	5	0000000000000000000000000011111	5	18
106	6	0000000000000000000000000111111	6	20

3. Read the next $\gamma_i = 16$ bits of the compressed file. It is sequence number:

$$N = 0111100101010101_{<2>} = 31061_{<10>} .$$

4. Set the parameters: $i = n - 1 = 31$, $k_i = k = 4$.

5. Generate a binomial number from the sequence number N (see Table 4).

Table 4. Formation of the binomial number from 31061

i	31	30	29	28	27	…	3	2	1
k_i	4	4	3	2	2	…	2	2	1
$C_i^{k_i}$	31465	27405	3654	378	351	…	3	1	1
N	31061	31061	3656	2	2	…	2	2	1
Binomial number	0	1	1	0	0	…	0	1	1

The result of this operation is binomial number: 0110000000000000000000000000011.

6. The last bit of the binomial number is 1, so we append right 0's up to $n = 32$ bits length.
7. The result of the decompression is the binary sequence:

01100000000000000000000000000110.

The size of the compressed image and the compression ratio calculated by the formula (10) and (11) are as following:

$$L = 108 \cdot \lceil \log_2 33 \rceil + \sum_{i=1}^{108} \gamma_i = 2250 \text{ bits}, \quad K = \frac{H \cdot W}{L} = \frac{3456}{2250} = 1{,}536 \ .$$

Summary

In conclusion the compression method based on binomial numbers that has been introduced in this paper has the following benefits:

1. The considered method does not require the advance gathering of statistical data on compressed messages as opposed to Shannon-Fanos' and Huffman's compression techniques. In the wake of the virtue the expenditures for the needed hardware and software is cut down.

2. The considered method is a method of lossless compression as opposed to fractal compression, JBIG2, method of principal components etc. In spite of more low compression ratios as compared with lossy compression techniques this virtue allow us to apply successfully the binomial compression in such progressive information systems as systems of automatic control, systems of science research automation and others.

3. According to the compression method based on binomial numbers the compressed data are more resistant to random changes of their bits, which result to long chains of errors in case of applying Shannon-Fanos', Huffman's compression techniques and other compression methods, based on statistical properties of information.

4. It should be noted separately the proposed method operates with binomial numbers and, consequently, gives possibility to solve different computational tasks, connected with searching, ordering and filtering data (for example, strings of the compressed image) by the numerical features of the sequence numbers in the compressed file.

The compression method based on binomial numbers is very effective for binary images with a predominance of one colour (black or white). Examples of such images are scanned texts and drawings, charts, line graphs etc., which are used for recognition, transfer and storage.

The compression ratio depends on the size of the compressed blocks and on the distributions of black and white colors in the blocks. The compression ratio is greater when deviation of the pixels distribution from the middle value is greater. The compression ratio increases with increasing of the block size n. At the same time, the expenditures for hardware and software that are necessary for calculation of the binomial coefficients with large values of n and k, increase too. However, there are methods of calculating the binomial coefficients [5] that can greatly accelerate the process of compression.

References

[1] Borysenko, O.A., *Introduction in the theory of the binomial account*, University book, Sumy, 88 p., 2004 (in Russian)
[2] Borysenko, O.A., Kulyk, I.A., *Binomial coding*, University book, Sumy, 206 p., 2010 (in Russian)
[3] Kulyk, I.A., Kostel, S.V., Skordina, O.M., Application of binomial numbers for binary images compression, *Vestnik SumDU*, *Sumy Publishing House of Sumy State University*, 2, pp. 29-36, 2009 (in Russian)
[4] Kulyk, I.A., On average length of binary binomial numbers, *Vestnik SumDU*, *Sumy Publishing House of Sumy State University*, 12(71), pp. 106-112, 2004 (in Russian)
[5] Kulyk, I.A., Skordina, O.M., Method for calculation of binomial coefficients on basis of canonical decomposition of numbers, *Vestnik SumDU*, *Sumy Publishing House of Sumy State University*, 1, pp. 158-165, 2008 (in Russian)
[6] Marchand-Maillet, S., Sharaiha, Y.M., *Binary Digital Image Processing: A Discrete Approach*, Academic press, 368 p., 1999
[7] Salomon, D., *A Guide to Data Compression Methods*, Springer-Verlag, 368 p., 2002
[8] Vatolin, D., Ratushnyak, A., Smirnov, M., Yukin, V., *Data Compression Methods. Organization of Archivers, Image and Video Compression*, Dialog-MIFI, Moscow, 381 p., 2002 (in Russian)

Compresia imaginilor binare bazată pe numere binomiale

Rezumat

În acest articol prezentăm o metodă de compresie a informației care este bazată pe tranziția de la o secvență binară la numărul său serial în sistemul numerelor binomiale. În primul stadi al compresiei binomiale, este efectuată tranziția de la o secevență de cod cu pondere pară la un număr binomial corespunzător, apoi, în stadiul al doilea, numărul binomial este transformat în numărul său serial conform cu funcția numerică a sistemului de numere binomiale, cu parametrii dați. Înaintea compresiei binomiale, secvențele binare ordinare sînt luate în considerare combinațiile de cod cu pondere pară de la calcularea numărului lor de unități binare. În acest articol, secvențele binare inițiale sînt șiruri și sub-șiruri, coloane și sub-coloane ale unor imagini binare. Articolul explică metoda compresiei imaginilor binare bazată pe numere binomiale. Metoda considerată are algoritmi de compresie și decompresie simpli, care sînt prezetați în detaliu în articol. Este implementată și o estimare a metodei de compresie. Exemplul ales arată cum se comprimă o imagine binară cu ajutorul metodei de compresie bazată pe numere binomiale, generate în sistemul numerelor binomiale.