UDC 004.04
DOI: 10.15587/2312-8372.2018.134978

**Kotsuibivska K.,
Chaikovska O.,
Tolmach M.,
Khrushch S.**

IMAGES COMPRESSION BY USING CUBIC SPLINE-FUNCTIONS METHODS

Об'єктом дослідження є алгоритми стиснення зображень на основі математичних методів. Основною проблемою при стисненні файлів зображень є втрата якості при відновленні. В роботі запропоновано підхід, при якому користувач сам може обирати якість відновленого зображення. Це досягається за рахунок використання методу сплайнової інтерполяції, який дозволяє задавати коефіцієнт стиснення, таким чином, керуючи якістю відновленого зображення.

Використання методу сплайн-фукнцій для стиснення зображень дозволяє значно скоротити час на обробку файлів за рахунок простоти математичної моделі алгоритму. За заданою якістю відновленого зображення алгоритмом визначається розмір стисненого файлу, залежно від кольорової гами.

В результаті аналізу запропонованої розробки представлені коефіцієнти стиснення зображень, які показують, що розмір стисненого зображення може бути менший 50–70 % від вихідного файлу. Декодування проводиться за відомими коефіцієнтами сплайн-функції. Отриманий результат порівнюють з вихідним файлом. Різниця між інтенсивністю точок вихідного і декодованого зображення визначає якість відновлення.

Отримано алгоритм, який дозволяє задавати точність відновленого зображення. Такий результат залежить від вагових коефіцієнтів сплайн-фукнції, які впливають на точність побудови поліному апроксимації. Особливістю запропонованого підходу є те, що користувач сам може вирішувати наскільки точним і якісними повинне бути зображення після декодування. Це досягається за рахунок того, що точки, інтенсивності яких близькі за значеннями, відновлюються з невеликою похибкою.

В роботі запропоновано підхід, який передбачає послідовне виділення блоків точок однакової інтенсивності. Для виділених блоків будується апроксимуючий поліном на основі сплайн-функції, а коефіцієнти поліному передаються в файл, який містить інформацію для відновлення зображення. Так можна досягти більших коефіцієнтів стиснення за рахунок побудови поліному для блоків, які містять точки близькі за інтенсивністю.

Ключові слова: *кодування зображень, стиснення зображень, растрові зображення, коефіцієнти сплайн-функції, апроксимуючий поліном.*

1. Introduction

In a situation where information becomes more important, and the ways of disseminating data become more diverse, the issue of data compression is one of the most important places. Especially this problem concerns image files.

However, in the process of converting an analog signal into a digital form, the frequency band occupied by these signals significantly expands, requires high-bandwidth transmission channels and large amounts of memory for storing them. This is especially true for images, so it is urgent to perform compression of images before they are stored and transmitted via communication channels.

The peculiarity of image compression is that it is possible to achieve high compression rates only due to loss of information, and these losses should not be visually visible.

When choosing algorithms, it is important to understand their positive and negative sides. If an algorithm with loss of data is chosen, one should understand its nature and the conditions under which the image will deteriorate. Using the new optimal algorithms will save the quality of images, tens and hundreds of megabytes of disk space.

Among the known compression methods (encoding) of images, the most optimal relationship between the loss and the compression ratio can be achieved by applying interpolation methods, namely, by using cubic spline-functions.

2. The object of research and its technological audit

The object of research is image compression algorithms based on mathematical methods. The main task of the new development is increasing the compression rates with small loss of image accuracy. The main assessment of image accuracy is visual quality, the performance of the method is speed.

The most common commercial technology is the jpeg technique, in which compression is achieved by the cosine Fourier expansion. Another approach can be implemented based on the singular use of the matrix. Modification of existing technologies consists in developing new mathematical approaches and algorithms, as well as creating new software technologies and solutions [1].

The first to archive images began to apply methods that were used and used in backup systems, when creating distributions, etc. These algorithms archived the information unchanged. However, the main trend in recent years has been the use of new classes of images. The old algorithms have ceased to meet the requirements for archiving. Many images are almost not compressed. This led to the creation of a new type of algorithms that compress with loss of information. As a rule, the coefficient of archiving, and therefore the degree of quality loss in them, can be set. At the same time, a compromise is achieved between the size and quality of the images.

It is in the optimization of compression parameters and the subsequent restoration of images that the basic requirement is for mathematical models and the creation of algorithms for image coding.

Using the developed algorithm will make it possible to obtain good compression rates, and without performing complex calculations, increases the speed.

3. The aim and objectives of research

The aim of research is creation and studying algorithms and image compression software based on cubic spline-functions.

To achieve this aim, it is necessary to perform the following tasks:

1. To justify the benefits of compression based on cubic spline-functions.
2. To analyze the quality of image compression using the spline interpolation method.
3. To compare the results with existing standards.

4. Research of existing solutions of the problem

All compression algorithms operate with an input stream of information, the minimum unit of which is a bit, and the maximum is several bits, bytes or several bytes. The goal of the compression process, as a rule, is obtaining a more compact output stream of information units from some initial noncompact input stream with the help of some of their transformations.

Recently, images and illustrations have been used everywhere. The problem associated with their large size appeared when working on both workstations and personal computers. Despite the fact that several illustrations are usually used, and the fact that they are often much larger (for example, with color printing), keeping them unpackaged becomes expensive. In recent years, serious attention has been paid to solving this problem [1–5]. A large number of different algorithms for archiving graphics have been developed: both modified universal and completely new algorithms oriented only on images were used. Moreover, algorithms were developed that were oriented only on a specific class of images.

Let's compare the known coding methods, indicating their advantages and disadvantages.

Group coding is the most known simple approach and the algorithm for compressing information in a reversible way is the Run Length Encoding (RLE). The problem of all similar methods is determination of the way in which the decoding algorithm could distinguish the encoded series in the resulting byte stream from other – unencoded byte sequences – requires additional costs. These methods, as a rule, are quite effective for compressing raster graphic images (BMP, PCX, TIF, GIF), since the latter contain quite a lot of long series of byte sequences that are repeated. The disadvantage of the RLE method is a rather low compression ratio of files with a small number of series and, worse, with a small number of repetitive bytes in the series. When using the adaptive Huffman coding [1], the complication of the algorithm consists in the need for constant updating of the tree and character codes of the main alphabet according to the statistics of the incoming stream are changing. The main disadvantage of this method is that when encoding a stream with a two-letter compression alphabet is always missing, because despite the different probability of the appearance of symbols in the input stream, the algorithm actually reduces them to 1/2 [1]. When developing the method of arithmetic coding, two problems arise: first, floating point arithmetic, theoretically, of unlimited accuracy, and, secondly, the result of coding becomes known only at the end of the input stream. The Lempel-Ziv-Welch algorithm (LZW) is distinguished by the high speed of work both for packing and unpacking, rather modest memory requirements and simple hardware implementation, but the disadvantage is the low compression ratio in comparison with the two-stage coding scheme [2]. A feature of fractal compression is the need for colossal computing power when archiving. The most famous program-archivers to date are the Microsoft Image File Converter, based on the use of the JPEG compression method. A not very pleasant feature of JPEG is that often the horizontal and vertical bars on the display are completely invisible, and can only appear when printing as a moire pattern. In all the above algorithms by today's standards, the archiving factor is too small.

The main technical characteristics of compression processes and the results of their work are:

- compress rating or the ratio of the volumes of the source and the resulting flows;
- compression speed – the time it takes to compress a certain amount of information of the input stream before obtaining an equivalent output stream from it;
- quality of compression – a value indicating how strongly packed the output stream is determined by the ratio of the size of the packed file to the file size obtained after applying to it the repeated compression on the same or another algorithm.

In order to correctly evaluate the direction in which algorithms change, it is not enough to define only image classes. It is necessary to set and certain criteria, one of which is worse, the average and best compression ratios. That is, the proportion by which the size of the image will increase if the original data is bad; a certain average statistical coefficient for the class of images to which the algorithm is oriented; and, finally, the best ratio. The latter is only needed theoretically, since it shows the compression ratio of the best (usually black) image, often of fixed size.

Next, let's consider static raster images, representing a two-dimensional array of numbers – pixels. All images can be divided into two groups: with and without a palette. In images with a palette in a pixel a number is stored – an index in a one-dimensional color vector, called a palette. Most often there are palettes of 16 and 256 colors.

The first to archive images began to apply lossless compression algorithms. Those that were used and used in backup systems, when creating distributions, etc. However, time does not stand still, and the main trend today is the use of new classes of images. The old algorithms have ceased to meet the requirements for archiving. This led to the creation of a new type of algorithms – compression with loss of information. As a rule, they can set the archiving factor and, consequently, the degree of quality loss. At the same time, a compromise is achieved between the size and quality of the images.

The basic method of digital image coding is pulse-code modulation (PCM). It is characterized by the fact that each encoded to the word corresponds to time-quantized and amplitude-quantized video information [3]. The requirements of the discretization theorem must be satisfied:

$$f_g = 2\omega, \quad (1)$$

where ω – the maximum frequency of the signal.

To prevent the appearance of fake circuits for one component of the color, at least 50 quantization levels, corresponding to a 6–8 bit word for each element of the color component of the image, are necessary. For large volumes of information, PCM is used only for intra-studio transmission of television signals in parallel code. PCM is the basic, canonical representation of an image in digital form [3].

Among the statistical methods, the most widely used block methods of image coding. These methods work as follows. Blocks of size $M\times N$ elements are coded according to the probability of their appearance. For the most probabilistic blocks, short codewords are used, and for less probabilistic blocks, long codewords (the Huffman algorithm) are used, which results in data compression [4, 5]. Compression factors using these methods can reach 4–5.

A review of literature sources shows that the main ways to solve the problem of image compression is the use of wavelet-coding [1–3] and using the jpeg format to reduce the amount of image files [4, 5]. A technique for reducing the size of image files based on scaling methods has also been proposed [6]. A large number of works are devoted to the use of combined methods of image compression [7–10].

In particular, the paper [1] is devoted to a review and comparison of known methods of image compression. The authors of [2, 3] when comparing different coding methods also note that when the quality of the restored image is improved, the compression ratio decreases. Especially such problem occurs for images with a saturated color gamma.

When working with raster images, the primary task, before any processing, is determination of the intensities of points. The authors of [4] also propose a method for constructing a polynomial recovery function, but the original file already contains a vector image. This method gives a fairly good relationship between the compression ratio and the quality of recovery.

Considering the method [5], the author notes that the resulting raster is less redundant. Transformation can be used for images with large areas of equal coloration. This coding is the reverse, that is, the image is restored without loss. This transformation reduces entropy, hence increases the compression ratio.

Also of interest is the ICA compression model [6], which is based on the subdivision of the image into small areas for further observations using the combined Delta function and the Gaussian distribution density function.

The features of wavelet-coding, lossless coding methods are that they became the basis for further studies [7–10], but so far they are used for two-dimensional images.

5. Methods of research

Interpolation coding systems are based on numerous approximation methods, according to which a sequence or a two-dimensional array of brightness readings are approximately represented through continuous functions. The interpolation procedure can be applied at the step of converting the image into an encoded signal (interpolation on the transmitting side) or it can be part of the process of image reconstruction from an encoded signal (interpolation at the receiving end).

In coding systems with interpolation on the transmitting side, the brightness values are approximated by continuous functions with a previously established accuracy. Interpolation can be carried out along a scanning tape or covered by some part of the image plane.

The zero-order interpolator works as follows: for all elements of the image, the same interval of permissible distortions is established, within which a set of horizontal straight-length segments is constructed without additional restrictions on the placement of their initial and end points. Each element of the image is overlapped by any of these segments. On the communication channel, the vertical coordinate of each segment and its length, expressed by the number of elements, are transmitted. When restoring the image on the receiving side, all elements within the segment acquire a level corresponding to its vertical coordinate. This variant of interpolation gives greater freedom in choosing possible combinations of horizontal line segments and, in this connection, allows to obtain the most effective representation of the output data with the help of a minimum number of segments. However, the amount of computational operations necessary to construct such optimal approximation often turns out to be very large. A simplified version of zero-order interpolation is the encoding of the lengths of the series, indicating the brightness of the first element of the series.

The action of different first-order interpolators is as follows: every element of the image is overlapped by any of the straight-line segments, the placement of which within the permissible error range is not related to additional restrictions on the placement of the start and end points. The computational procedure of approximation can be somewhat simplified by joining the beginning of the next segment with the end of the previous segment. A further simplification is use of the brightness values of the elements as the start and end points, an approximation of this type is often called fan interpolation.

Polynomial functions of higher order, for example, cubic splines, can also be used for coding with interpolation, but an increase in the order of polynomials is accompanied by a rapid increase in the amount of computation. It is also possible to set the problem of two-dimensional interpolation of zero and first orders, but the corresponding interpolator is difficult to implement in practice.

Various methods of image compression are considered above. Each of them is aimed at reducing the volume of the graphic image in order to save disk space. Next, algorithms for processing images using spline interpolation will be considered.

Essence or core of the software product, is being developed – the implementation of the module for recovering missing samples using cubic spline functions. Therefore, in this section let's describe the mathematical side of spline interpolation.

Before to consider algorithms for image compression using anti-aliasing, we let's clarify the essence of the smoothing problem itself.

The measuring device responds to the input signal x_n with an output signal, with the numerical parameter n varying from 1 to N. Through the total number of N pairs of values, it is necessary to perform the function $f(x)$ so that an analytic connection between the output and input signals, characteristic.

When interpolating, the desired curve $f(x)$ passes through the support points $f(x_n)=y_n$. The function found makes it possible to calculate the output signal $f(x)$ also between the reference points $x_n \le x \le x_{n+1}$. When approximating the required curve, it is necessary to carry out a certain law between pairs of quantities (scattered or noisy), not making sure that the reference points functions coincide with the measured values of y_n.

Algorithms for interpolating functions from exact data defined on discrete sets of points are usually based on the use of Lagrange interpolation polynomials. Moreover, with respect to the function $\varphi(x)$, which is interpolated, an a priori assumption is introduced that it has derivatives up to a certain order.

Another problem, close to the problem of interpolation, arises when the value of the given function $\varphi(x)$ is known at the node points not exactly, but with some error, the maximum value of which for each point is given as a priori information. In this case, the problem is construction of a curve that best approximates the function given with random errors at the node points. Such problem is usually solved on the basis of the method of least squares.

Recently, the theory of interpolation has been enriched by new methods, called spline interpolations. It should be noted that spline is usually defined in the domain D as a piecewise polynomial function, that is, a function for which there exists a partition D on a subdomain such that inside each element of the partition the function is a polynomial of some power m. In addition, this function, as a rule, is continuous in the domain D together with the derivatives $(m-1)$. The most common in the technique of steel splines are polynomials of the third order.

Let a grid $a=x_0<x_1<...<x_n=b$ be given on the interval $[a, b]$ of the x-axis, at the nodes of which the values $\{f_k\}_{k=0}^{n}$ of the function $f(x)$ defined on $[a, b]$ are given. Then the problem of piecewise cubic interpolation is posed as follows. On the segment $[a, b]$ it is necessary to find a function $g(x)$ that satisfies the following conditions:

1) $g(x)$ belongs to the class $C^{(2)}(a, b)$, that is, continuous together with its derivatives up to the second order inclusive;

2) on each of the intervals $[x_{k-1}, x_k]$ $g(x)$ is a cubic polynomial of the form:

$$g(x)=g_k(x)=\sum_{l=0}^{3} a_l^k (x_k x)^l,\ k=1,\dots,n; \quad (2)$$

3) in the nodes of the grid $\{x_k\}_{k=0}^{n}$, there are equalities:

$$g(x_k)=f_k,\ \ (k=0,1,...,n); \quad (3)$$

4) $g^{//}(x)$ satisfies the boundary conditions:

$$g^{//}(a)=g^{//}(b)=0. \quad (4)$$

The advantages of the chosen interpolation will become clear later, when establish the extremal property of the so-defined function $g(x)$.

Let's show that the problem posed for finding an interpolating piecewise-cubic function $g(x)$ has a unique solution. To do this, let's use the conditions formulated above.

Since the second derivative of the function $g(x)$ is contiguous and linear on each interval of the net $[x_{i-1}, x_i]$ $(i=1,...,n)$, it is possible to write for $x_{i-1}\le x\le x_i$:

$$g^{//}(x)=m_{i-1}\frac{x_i-x}{h_i}+m_i\frac{x-x_{i-1}}{h_i}, \quad (5)$$

where $h_i=x_i-x_{i-1}$, $m_k=g^{//}(x_k)$. Let's integrate both sides of equality (5) twice and obtain:

$$g(x)=m_{i-1}\frac{(x_i-x)^3}{6h_i}+m_i\frac{(x-x_{i-1})^3}{6h_i}+ + A_i\frac{x_i-x}{h_i}+B_i\frac{x-x_{i-1}}{h_i}, \quad (6)$$

where A_i and B_i – some integration constants, which are calculated from the condition $g(x_{i-1})=f_{i-1}$, $g(x_i)=f_i$. Substituting $x=x_i$ та $x=x_{i-1}$ in (6) obtain:

$$g(x)=m_{i-1}\frac{(x_i-x)^3}{6h_i}+m_i\frac{(x-x_{i-1})^3}{6h_i}+ +\left(f_{i-1}-\frac{m_{i-1}h_i^2}{6}\right)\frac{x_i-x}{h_i}+\left(f_i-\frac{m_ih_i^2}{6}\right)\frac{x-x_{i-1}}{h_i}, \quad (7)$$

$$g^{/}(x)=-m_{i-1}\frac{(x_i-x)^2}{2h_i}+m_i\frac{(x-x_{i-1})^2}{2h_i}+ +\frac{f_i-f_{i-1}}{h_i}-\frac{m_i-m_{i-1}}{6}h_i. \quad (8)$$

From (8) let's find the one-sided bounds of the derivative at the points $x_1, x_2, ..., x_{n-1}$, and in accordance with the continuity condition for the function $g^{//}(x)$ and $g^{/}(x)$ on $[a, b]$ and $g^{/}(x)$ continuity at the points $x_1, x_2, ..., x_{n-1}$, let's obtain the $n-1$ equation:

$$\frac{h_i}{6}m_{i-1}+\frac{h_i+h_{i+1}}{3}m_i+\frac{h_{i+1}}{6}m_{i+1}=\frac{f_{i+1}-f_i}{h_{i+1}}-\frac{f_i-f_{i-1}}{h_i}. \quad (9)$$

Supplementing these equations from conditions (4) by the equalities $m_0=m_n=0$, let's obtain a linear algebraic system for finding the unknowns $m_1, m_2, ..., m_{n-1}$:

$$Am=Hf. \quad (10)$$

The square matrix A has the form:

$$A=\left\|\begin{matrix} \frac{h_1+h_2}{3} & \frac{h_2}{6} & 0 & ... & 0 & 0 \\ \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & ... & 0 & 0 \\ 0 & \frac{h_3}{6} & \frac{h_3+h_4}{3} & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & \frac{h_{n-1}}{6} & \frac{h_n+h_{n-1}}{3} \end{matrix}\right\|, \quad (11)$$

 ISSN 2226-3780

the vectors m and f, and the rectangular matrix H are as follows:

$$m=\begin{Vmatrix} m_1 \\ m_2 \\ \ldots \\ m_{n-1} \end{Vmatrix},\ f=\begin{Vmatrix} f_1 \\ f_2 \\ \ldots \\ f_n \end{Vmatrix},$$

$$H=\begin{Vmatrix} \frac{1}{h_1} & \left(-\frac{1}{h_1}-\frac{1}{h_2}\right) & \frac{1}{h_2} & \ldots & 0 & 0 \\ 0 & \frac{1}{h_2} & \left(-\frac{1}{h_2}-\frac{1}{h_3}\right) & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \left(-\frac{1}{h_{n-1}}-\frac{1}{h_n}\right) & \frac{1}{h_n} \end{Vmatrix}.$$

The matrix A is symmetric with a strict diagonal advantage. The coefficients m_1, m_2, ..., m_{n-1} are uniquely determined from the system (10). Thus, the spline function $g(x)$ is also uniquely reconstructed by the formulas (7) and the problem of finding a piecewise-cubic function $g(x)$ has a unique solution.

Cubic splines have a very important property, which determines the high efficiency of spline interpolation. Namely, consider on the interval $[a, b]$ a class $W_2^2[a, b]$ consisting of functions that have second-order squares added. Let's pose the problem of finding the interpolation function:

$$u\in W_2^2[a,\ b],\ u(x_k)=f_k,\ k=0,1,\ldots,n, \tag{12}$$

which minimizes the functional:

$$\Phi(u)=\int_a^b [u^{//}(x)]^2 dx, \tag{13}$$

on the class $W_2^2[a, b]$. It is stated that the minimum of such a functional is achieved on the piecewise cubic spline function $g(x)$, which was just constructed. In fact, consider the value:

$$\Phi(u-g)=\int_a^b [u^{//}-g^{//}]^2 dx. \tag{14}$$

Integrating by parts and using the properties of the functions g and $u\in W_2^2$, let's obtain:

$$\Phi(u-g)=\Phi(u)-\Phi(g)-2\sum_{k=1}^{n}\int_{x_{k-1}}^{x_k}[u^{/}-g^{/}]g^{///}dx, \tag{15}$$

but $g^{///}=c_k=\text{const}$ on the interval $[x_{k-1},\ x_k]$, therefore:

$$\Phi(u-g)=\Phi(u)-\Phi(g)-2\sum_{k=1}^{n}c_k(u-g)\Big|_{x=x_{k-1}}^{x=x_k}=$$
$$=\Phi(u)-\Phi(g).$$

From this and (14) it follows that:

$$\Phi(g)=\Phi(u)-\Phi(u-g)\leq\Phi(u). \tag{16}$$

Thus, the minimum of the functional (13) is realized on the piecewise-cubic function $g(x)$. It is not difficult to show that there aren't other points of the minimum in the functional [7].

Based on (12), (13), it is possible to give another equivalent definition of a piecewise cubic spline function: it is a function from a class $W_2^2[a,\ b]$ that takes a given value at the grid nodes and minimizes the functional (13).

Cubic spline functions have good properties. If the function f that is interpolated belongs to the class C $C^{(k)}[a,\ b]$ $(k=0,1,2,3,4)$, then for the error function $\varphi(x)=$ $=f(x)-g(x)$ inequalities:

$$\max_{a J x J b}\left|\varphi^{(p)}(x)\right| J C h^{k-p},\ k\geq p, \tag{17}$$

where C – an integral constant not depending on the grid, $h=\max_{1\leq i\leq n}|x_i-x_{i-1}|$.

The essence of the compression method is the following: first take the first two points from the smoothing area. The coefficients of the polynomial $P(x)$ are calculated, and a smoothing curve is constructed for the selected points. The values of the coefficients are entered in some other file. Using this same polynomial, a smoothed curve is constructed for the next two points, until the difference between the initial values of the function at these points and the values obtained as a result of interpolation will not exceed some given value Δ:

$$\Delta=(P(x_n)-y_n), \tag{18}$$

where $P(x_n)$ – interpolation values; y_n – the initial values.

For points where the value Δ exceeds the specified value, the coefficients of the polynomial $P(x)$ are again found, and the algorithm is repeated first.

This algorithm can be effectively applied to images with a small number of colors (black and white images). Since a large number of colors, and, correspondingly, a change in the intensities of the points, there is a need for a large number of calculations to calculate the coefficients. This reduces the processing speed of the image, and in some cases even increases the amount of file in which the coefficients are stored. For a greater degree of compression, the coefficient file can be processed using some encoding method, or by using any of the known archivers.

Typically, a bitmap image is an array of points whose value corresponds to the intensity. Another is to break the matrix into small squares, for example, 4×4, for which the average intensity value is found, which is transferred to another file. Thus, instead of 16 points, let's obtain 1. To restore the image between the obtained points, an approximating or interpolating polynomial is performed.

6. Research results

The main stages of the image encoding process are:

1. Convert the image to the optimal color space (only for color images).

2. Sub-sampling of the chrominance components by averaging the groups of pixels (only for color images), which makes it possible to reduce the amount of information without loss by half.

3. A difference coefficient Δ is specified that shows how much the polynomial value at a given point differs from the initial value of the intensity.

4. For two image points, a third degree polynomial is constructed, which is then smoothed by the next two points, until the difference between the value of the polynomial at the point and the initial value in it is different by a given value Δ.

5. Recovering the missing samples by an interpolation polynomial (samples of the low-frequency component of the image are formed – $\overline{X(m_1,m_2)}$).

6. Formation of samples of the difference (high-frequency) component of the image and their quantization:

$$\delta(m_1,m_2)=x(m_1,m_2)-\overline{X(m_1,m_2)},$$
$$\delta_{KV}(m_1,m_2)=\sum_{1}^{n}A_j 1\left\{\delta(m_1,m_2)-\Delta_j\right\}, \quad (19)$$

where $x(m_1,m_2)$, $\overline{X(m_1,m_2)}$ – respectively, the values of the samples of the original image and the low-frequency component; A_j – the steps of quantizing the original signal; Δ_j – thresholds of quantization of the input signal $1\{\bullet\}$ – a unit function.

7. Coding samples of high-frequency and low-frequency components (only reference samples) using statistical coding methods.

Recovery is the reverse process to coding, with the difference that there is no need to calculate the coefficients of the spline function and there is no quantization. It is clear that the recovery scheme is simpler and, accordingly, the recovery process is faster from the encoding process. Experimental studies have shown that the rate of acceleration of the recovery operation is in the range of 8–15 % of the data encoding time.

6.1. Image conversion to the optimal color space. This algorithm is designed to convert the RGB signal into a signal *YUV*, where *Y* – the bright signal, *U* and *V* – color difference signals. This conversion reduces image redundancy.

Conversion is performed from point to point using the following formulas:

$$Y=0.3\cdot\mathrm{R}+0.59\cdot\mathrm{G}+0.11\cdot\mathrm{B};$$
$$U=-0.15\cdot\mathrm{R}-0.29\cdot\mathrm{G}+0.44\cdot\mathrm{B};$$
$$V=0.62\cdot\mathrm{R}-0.52\cdot\mathrm{G}-0.1\cdot\mathrm{B}, \quad (20)$$

where R, G, B – color image signals (red, green and blue, respectively).

6.2. Development of an algorithm for coding images by the method of cubic spline functions. The first step in the implementation of the coding algorithm is the calculation of the coefficients of the cubic spline function. For this, two points are taken from the data area and for them a log is built.

In the second stage, the next two points are smoothed by the already constructed polynomial, which is repeated until the difference between the value of the polynomial at a given point and its initial value is different by a given value Δ. If at some point n the value of the difference $(P(x_n)-f(x_n))$ exceeds the value Δ, then a new log is constructed for this point, and the next neighboring one. The value of the first point and the $(n-1)$-th point are recorded in the auxiliary file. The construction of the polynomial is carried out first by the terms, and then by the column.

The final stage of image coding by the method of spline interpolation is quantization, which is performed on a uniform scale.

6.3. Development of an algorithm for decoding images. The input data stream for the decoding algorithm will be the values of the points, stored in the file, and the values of the coefficients of the cubic spline function on the corresponding segments. Thus, the decoding algorithm is simplified, since there is no need to calculate the spline coefficients.

The first stage of the decoding algorithm is creating a restored image, by smoothing out the specified points.

The second and final stage of the decoding algorithm is the integration of the reconstructed image with the quantized samples and the normalization of the output stream (check for overflow).

Compared to analogs, the development of an image compression algorithm based on cubical spline-function methods provides the possibility of efficient image compression without the complex use of the proposed methods, since it is universal and allows you to perform compression for images of any type. In the proposed method of smoothing is carried out on two points, it makes it possible to more accurately transmit the intensities at points, and the decoded image accuracy factor is also defined, which is defined as the difference between the point intensity value in the source file and the decoded one.

For comparison, the JPEG standard is not chosen by chance. This is one of the most common compression standards for graphic images. Therefore, this standard is an analog, with which we can compare the results obtained.

In the case of JPEG encoding, the ACDSee 32 program was used. The results for six different files are summarized in the table (Table 1).

Table 1

Summary table of the results of experimental studies

File name	File size, byte		
	BMP	JPEG	ICP
128018.bmp	1 087 074	115 140	57 794
Cat.bmp	958 878	75 759	68 470
Cindy04.bmp	387 654	42 764	35 017
Girl.bmp	192 054	15 408	14 387
Lake.bmp	921 654	284 755	101 996
Bull.bmp	663 542	101 281	60 511

By the time the converter worked in JPEG format, it worked 2–2.5 times faster than this development. However, there is an advantage in the compression ratio. Moreover, in images that do not have a large number of circuits, the encoder ICP encoded images 1.8-2.1 times more efficiently than JPEG.

7. SWOT analysis of research results

Strengths. The developed method of image compression allows not only to reduce the time for processing raster image files, but also to obtain the best ratio of the compression ratio and the quality of the restored image. In

ISSN 2226-3780

addition, development does not require special economic costs, enough personal computer. An important feature of the developed algorithm is the user's ability to choose the quality of the restored image.

Weaknesses. The use of the proposed compression method will be most effective for images in which the intensities of neighboring points differ not very strongly, or sequences of points of the same intensity are large enough. Otherwise, the compression ratio will not be large. Also, this method can't be applied to 3D images.

Opportunities. Further development of the proposed method is possible for studies of compression of 3D images and video files.

Threats. To apply the proposed method for 3D-images, an approximation in space should be used.

8. Conclusions

1. Analysis of known methods of image coding shows that using several approaches simultaneously gives better results when compressing image files. Advantages of many methods are in obtaining a high compression ratio, but there is a loss of quality of the restored image. The combined use of methods not only maintains a high compression ratio, but also significantly improves the quality of the restored image. The proposed approach of image compression by cubic spline functions not only does not require the use of additional approaches to improve the algorithm, but is also quite simple from a mathematical point of view, and from the point of view of software implementation.

The development of a mathematical and algorithmic model of the decoupling of the set image compression problem on the basis of the method of cubic spline functions yielded, in the end, not a complicated, from the point of view of computations, methods. This approach will not only improve the image quality after compression, but also will not take much time to process. An important advantage of the obtained algorithm is that the user himself can specify the accuracy of the restored image, depending on its further use.

2. Analysis of the research results shows that the ratio of the volumes of the source and compressed files for the proposed methodology is 0.25–0.5, while the existing methods show a result of 0.5–0.7.

References

1. Lezhnev V. G. Matematicheskie algoritmy szhatiya izobrazheniy. Krasnodar: Kuban. gos. un-t, 2009. 55 p.
2. Obzor algoritmov szhatiya s poteryami. URL: http://mf.grsu.by/UchProc/livak/po/comprsite/theory_fractal.html (Last accessed: 02.12.2017).
3. Metody szhatiya dannykh: Szhatie izobrazheniy. URL: http://www.compression.ru/book/part2/part2__3.htm (Last accessed: 04.12.2017).
4. Jiao L. C., Tan S., Liu F. Ridgelet theory: from ridgelet transform to curvelet // Chinese Journal of Engineering Mathematics. 2005. Vol. 22, No. 5. P. 761–773.
5. Chiang T.-H., Dung L.-R. A VLSI Progressive Coding for Wavelet-based Image Compression // IEEE Transactions on Consumer Electronics. 2007. Vol. 53, No. 2. P. 569–577. doi: http://doi.org/10.1109/tce.2007.381731
6. Velisavljevic V., Beferull-Lozano B., Vetterli M. Space-Frequency Quantization for Image Compression With Directionlets // IEEE Transactions on Image Processing. 2007. Vol. 16, No. 7. P. 1761–1773. doi: http://doi.org/10.1109/tip.2007.899183
7. Iano Y., da Silva F. S., Cruz A. L. M. A fast and efficient hybrid fractal-wavelet image coder // IEEE Transactions on Image Processing. 2006. Vol. 15, No. 1. P. 98–105. doi: http://doi.org/10.1109/tip.2005.860317
8. Utsugi A. Independent components of natural images under variable compression rate // Neurocomputing. 2002. Vol. 49, No. 1–4. P. 175–185. doi: http://doi.org/10.1016/s0925-2312(02)00530-1
9. Remya S., Dilshad Rasheed V. A. Resolution Progressive Compression of Encrypted Images // International Journal of Signal Processing Systems. 2013. Vol. 1, No. 1. P. 7–10. doi: http://doi.org/10.12720/ijsps.1.1.7-10
10. A Hybrid Compression Method for Integral Images Using Discrete Wavelet Transform and Discrete Cosine Transform / Elharar E. et al. // Journal of Display Technology. 2007. Vol. 3, No. 3. P. 321–325. doi: http://doi.org/10.1109/jdt.2007.900915

Kotsuibivska Kateryna, *PhD, Associate Professor, Department of Computer Sciences, Kyiv University of Culture and Arts, Ukraine, e-mail: katysivak@gmail.com, ORCID: https://orcid.org/0000-0002-3987-9871*

Chaikovska Olena, *PhD, Associate Professor, Head of the Department of Computer Sciences, Kyiv University of Culture and Arts, Ukraine, e-mail: lena@knukim.edu.ua, ORCID: https://orcid.org/0000-0001-7769-1004*

Tolmach Maryna, *Lecturer, Department of Computer Sciences, Kyiv University of Culture and Arts, Ukraine, e-mail: margo.tolmach@gmail.com, ORCID: https://orcid.org/0000-0002-7020-1348*

Khrushch Svitlana, *Department of Computer Sciences, Kyiv University of Culture and Arts, Ukraine, e-mail: miksa@ukr.net, ORCID: https://orcid.org/0000-0001-9349-7762*