Two-dimensional data-block compression of information-measuring system algorithms

A V Levenets¹, D I Nefed'ev², R I Bazhenov³, V B Masyagin⁴ and S S Beknazanova⁵

¹Pacific National University, 136, Tihookeanskaya st., Khabarovsk, 680035, Russia
²Penza State University, 40, Krasnaya st., Penza, 440026, Russia
³Sholom-Aleichem Primursky State University, 70A, Shirokaya st., 679015, Birobidzhan, Russia
⁴Omsk State Technical University, 11, Mira ave., Omsk, 644050, Russia
⁵Tashkent University of Information Technologies named after Muhammad Al-Khwarizmi, 108, Amir Temur st., Tashkent, 100200, Uzbekistan

Abstract. The authors suggest a new approach to develop measurement data compression algorithms. According to that a data frame is considered as a bit sequence formed into a two-dimensional well-ordered linear structure. It is supposed that there is no explicit bind to the source data / specifications width within the collected structure. The researchers propose block compression algorithms. They take into account not only the relationship between one sensor readings in adjacent samples, but also the connection of the samples in one frame so that it is expected to contribute to higher compression efficiency. The studies of the proposed compression algorithms have shown that they can be effectively used for compressing data frames of information-measuring systems. The mid-compression ratio of the proposed block compression algorithms falls in the range 8.0 to 9.5, and a simple adaptive algorithm observed in the paper provides the maximum value of the average compression ratio.

1. Introduction

Measuring signals compression as an independent task first appeared in the middle of the twentieth century. Then the related terminology used in this area nowadays was common and compression methods that have become conventional now have been developed. It is important to emphasize that despite the modern high technology level, the issue of a balance between the economic and functional efficiency of information-measuring systems (IMS) is still important for engineers.

The main measures aimed at the overall IMS cost cutting they usually propose to use lower-rate communication channels, reduce the device capacity of a storage area used to store data, choose less high-speed calculators, etc. However, in some cases, in particular, for IMS collecting research data from complex technical objects, such measures are not always proper. Therefore, the development of new compression algorithms continues [1-3]. A general fairly complete review of current compression methods is observed in [4, 5]. The surveys are quite often devoted to versions of well-known algorithms for particular purposes [6, 7]. As a rule, new algorithms are focused on a specific data type, most often on multimedia data, the transmitted volumes of which are constantly increasing [8-10]. The development of measurement data compression algorithms is paid much less attention, because the current approach to compression involves dealing with each individual source of information (sensor) separately to the exclusion of a possible connection between the sources, for example, through the
measuring object. As a result of this, current compression algorithms have nearly hit overload of theoretical values of the compression coefficient and its increase became possible only for database sets / collections having certain statistical properties, i.e. for some specific practical application, for example, for sensor networks [11], space-based technology [12], computer engineering [13, 14], processing of measured data [15, 16], etc.

As a way of further developing methods in [17], it was proposed to examine the data collection from various sources as an integral information object with both spatial and temporal correlation interactions. The connection between different sources of information may be caused by physical interactions between the constituent parts of the measuring object, engineering solutions underlying IMS, for example, encoding mechanisms, etc.

IMS data is usually transmitted in frames of a fixed size and a permanent structure. According to the proposed approach, telemetry data of large-scale software systems correlate not only with the readings of one sensor in adjacent frames, but also with the readings within the frame. Therefore, it lays the groundwork for taking into account implied correlation dependencies of elements of a data frame that are not technically linked to information system, for example, by the difference method applied not to individual samples, but to the frame totally.

The application of the difference method to the data frame in the extreme case of no difference between adjacent frames leads to the maximum value of the compression ratio which is the greater the more information volume in one frame and the less volume of overhead information in compressed data. However, the decomposition of the source data frame into a two-dimensional structure and the distribution of amounts of data in it that are most correlated in time is of the authors’ main interest. The use of the difference method with variable length coding for these amounts of data is supposed to provide better compression than its application to the source data. So, in the extreme case of completely identical amounts of data in adjacent frames, a feature of the amount of data immutability and its symbolic / indirect address in the frame will be transmitted to the communicating channel.

Thus, the development of new methods and algorithms for compressing measurement data continue to be a live issue today. The main aim of the research is to assess the prospects of the proposed new approach to the development of compression algorithms for measuring data frames.

2. Method and algorithms for two-dimensional block compaction

Information field of data sources should be considered as a linear collection of bits with its subsequent transformation into a structure of increased dimension in order to provide efficient compression. Such a structure is a collection of areas filled with some fixed values. In general, the number of such areas with fixed values is equal to the number of characters of the alphabet used. Thus, the resulting information object (in the general case, an n-dimensional hypercube) will have a more ordered structure, the application of which even the well-known compression approaches will produce a greater effect than before the transformation. A natural constraint of this effect is the available overhead information necessary to describe the procedure for converting the source data in compress form. The obvious fact is the probable lack of a positive impact in case of processing uncorrelated data.

The most obvious transformation of this type can be considered the representation of all frame data in the form of a bit sequence, which, according to given rules can be converted later into more complex geometric data structures. Such structures, if described simply, can experience a compression procedure with greater effect than the original data frame. One should expect the improved compression of such structures even by the known compression methods. However, the greatest effect can be produced only if specialized compression methods and algorithms are applied.

Further, the authors assume that the measurement object is characterized by $N$ terms. The values of them form the vector $x(t) = (x_1(t), x_2(t), \ldots, x_N(t))$, at an arbitrary point of time $t$. The terms assuming their stationary, are described by the vector of auto correlation functions $R_t = (R_1(\tau), R_2(\tau), \ldots, R_N(\tau))$.

As a rule, present compression algorithms are limited by this information on compressed data assuming that the correlation matrix $R_t$ is diagonal. However, given that the object is a complicated
system, it is possible without losing generality, to assume that there is information connection between the elements of the vector \( x_i \), i.e. non-zero values in the off-diagonal elements of the correlation matrix \( R_x \). Thus, the object should be described by the matrix of correlation functions \( R_{st} \), representing both temporal and spatial parameter associations.

If the telemetry frame contains readings of \( N \) sensors, then the frame can be represented as a column bit vector \( x_d \) of dimension \( N \):

\[
x_d = (x_1, x_2, \ldots, x_j, \ldots, x_N)^T,
\]

where \( x_j \) is a readout of \( i \)-sensor.

In the belief that the data has number of bits \( n \), each element of vector \( x_j \) can be represented as a bit sequence and written as a row vector \( b \) of dimension \( n \):

\[
x_j = b_i = (b_{i1}, b_{i2}, \ldots, b_{in}).
\]  

Thus, the vector \( x_0 \) can be presented as a matrix \( X_b \), dimension \( N \times n \):

\[
X_b = \left( b_1, b_2, \ldots, b_N \right)^T = \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \ldots & b_{nn} \end{pmatrix}.
\]  

Such an approach to the representation of compressed data will make it possible to detect hidden correlations between the samples forming the frame. Indeed, considering the obtained bit field, it is possible to select and encode homogeneous areas regardless of the fact that such an area will consist of bits from the samples from different information sources.

It can be argued that for telemetric information coming from a complex technical object, the measured features of which are related to each other, including indirectly the compaction method that considers the frame as a whole will have high efficiency. Among other things, the compressed frame in the general case can be represented as a single object, the internal structure of which can be changed in some irreversible way.

So, the method is based on the representation of the original data frame as a bit field, which is later divided into homogeneous amounts of data, i.e. parts of a bit field filled with the same bits. Further, the scholars will assume that the development of the bit field from the source data has already been carried out. For example, as it is shown in [18]. An important point is that the original data frame is formed as the difference of adjacent frames. So, if there are two data frames represented by vectors \( x_1 \) and \( x_2 \), then the vector \( \Delta x = x_2 - x_1 \) should experience a conversion to a bit two-dimensional representation, according to (1) and (2). To make the operation of selecting homogeneous areas in the resulting frame easy, one should use the logical operation Exclusive OR. It seems clear that the process of specifying homogeneous areas will increase the required number of computing operations compared to a simple difference compression algorithm. For example, \( N \) of operations of arithmetic subtraction and \( N \) of operations of comparison are required for a difference algorithm, while for the proposed approach, the number of operations will depend on the method for specifying homogeneous areas as well as the number of areas identified. However, the operations used are considered to be bit operations, and as a rule, they take far less time for calculation. Since the computational capabilities of current computing aids depend much on the architecture of the microprocessors used, the command system, the features of the ram used, etc., the computational efforts for the proposed approach should be evaluated separately for each specific application.

The simplest difference algorithm uses the previous sample according to time parameter to predict. The result of the algorithm operation serves a bit string. The transmission of it will not allow the decoder to restore the original data without additional information. The obvious solution is supposed to use variable-length codes, such as Golomb coding [4]. However, it should be noted that in such case, the compression efficiency will be mostly affected by the frequency of character occurrence. In
fact, it can be admitted that such an algorithm will function most effectively for the case with uneven frequency of character occurrence.

The development of the difference algorithm is the block compression algorithm, which analyzes the dependence not only on the readings of a particular sensor in adjacent samples, but also takes into account the interconnection of samples from different sensors within the frame. Data compression efficiency can be improved by combining some sensors in blocks and applying differential compression for these blocks.

Depending on the method of selecting block features, data, the class of algorithms can be divided into two types. They are static block compression and adaptive block compression. The algorithms of the first group assume the dimensional stability of the frame size and block parameters over the system operating process. As for adaptive algorithms, these parameters are changed until the parameters providing the highest value of the compaction coefficient are determined. Selected blocks can have any configuration and can undergo affine transformations (scaling, rotation, reflection). Such a version of the block algorithm will be effective only for quite large amounts of data and if there is no hard limit on the processing time.

To process the sequence of frames, the researchers propose to use the basic idea of dictionary algorithms. So, to describe any block repeating in several frames, the number of such a block in the supporting substitution spreadsheet is used. The table is used both for blocks and for single readings. Here the authors present the results of the test algorithm, for which the size of the table is not limited and its reset does not happen until the current compression operation is completed.

The following algorithm is based on the well-known RLE algorithm. In case of repeating the readings of some sensor, a sequence of the form “0xxxxxxx” is written to the flow destination, where the least seven bits are a counter device of duplicate values. If there is no repetition, a sequence of “1xxxxxxx” is formed and the least significant bits stand for the number of non-repeating characters.

In the worst case, the message length will increase by $\frac{N}{128}$ bytes (with rounding upward / ceiling), where $N$ is the size of the original message in bytes. In case when all elements are equal, the maximum message size will be $\frac{N}{64}$ bytes. This algorithm is primarily intended to compress data that had been compressed by other algorithms.

The last algorithm observed here is based on predicting the readings of the current sensor from the average reading in all previous frames. The input algorithm receives data obtained after differential compression. It is assumed that in the general case the average value of the difference is non-zero, therefore, there is an opportunity to reduce the amount of data for the opposed values.

The average value can be found in two ways, that is, by storing the previous average and calculating the new value based on the current difference reading, as well as by storing the differences history with a view to the well-known formula. In the first case, only some small number is stored, and in the second, the division of errors is not accumulated, but a large array of numbers is required to be stored.

Thereby, the suggested compression algorithms use the laws of statistical regularity of data sequence in their work. The article does not cover algorithms that use higher order differences because such a review is redundant for most real-world applications.

3. Adaptive compression algorithms

Adaptation must be used to obtain the most efficient compression because of the changes in the statistical properties of the source data. The simplest known adaptation algorithm is that the entire set of algorithms used in the system is sequentially applied to the current frame. After that the best option is selected according to the compression ratio. Such an adaptation algorithm is expected to be computer resource intensive. However, one can reach acceptable performance if amounts of data are small and there is a detailed selection of algorithm parameters.

For testing and initial selection of adaptive compression parameters, the authors use the static block compression algorithm. A character property of it is its parameters are recorded for the whole course of the program operating and are not adapted to change the properties of compressible data. This
algorithm can be easily implemented in hardware or in low-performance computing tools. The main features that were selected using this algorithm are the frame size $W$ as well as the width $\Delta W$ and the height $\Delta H$ of the block. In the future, the configurations that showed the best values of the compression ratio were used in the adaptive algorithm as basic ones. An example of the behavior compression coefficient $k_c$ when the parameters $W = 16, \Delta W = 4, \Delta H = 5$ is shown in Figure 1.

Figure 1. Example of a program operating that implements the static block compression algorithm.

Based on the studies, it can be argued that the best compression results are achieved with an asymmetric frame, i.e. in the case when the value of one of the parameters exceeds the value of the other greatly. This is due to the fact that when the block height decreases, the correlation of the sensors in adjacent samples begins to reveal more strongly than the correlation between the readings of different sensors in one frame. Similar results were obtained for all frame width versions.

As the first version of the adaptive algorithm, an adaptive block compression (ABC) algorithm was developed. Its peculiarity deals with the individual selection of block parameters for each frame that result in fuller consideration of current dependencies between adjacent samples in the frame. It is clear from general considerations that such an algorithm provides proper compression for all data types.

The development of ABC algorithm is an algorithm with a substitution box (ABCSB). This algorithm carries out components of predictive coding, moreover, block differential compression prepares data, and an algorithm with a substitution box is used for the actual compression operation. The size of the box is not limited for testing. As a result of the studies, a low catching up of this algorithm with ABC algorithm is revealed. It can be explained by data pattern which the testing was conducted. So, an increasing number of identical sections in one frame cause an increase in the efficiency of the algorithm and vice versa.

An important feature is the amount of a compressed frame. Obviously, it is inefficient to use a large box for small frames. So, if the cell number of the table is encoded with eight bits, then a large number of identical blocks must be provided, and the sizes of the blocks must be much larger than the bit capacity of the number that determines the cell number. It is difficult to achieve for measurement data when the packets do not exceed one kilobyte. Nevertheless, the maximum and minimum values of the compression coefficient will be steady as the basic algorithm was and it allows using a substitution box /lookup table.

The third option of the adaptive algorithm is ABC algorithm using RLE compression, which is based on additional compaction of the sequence obtained by ABC algorithm by RLE algorithm. Such
additional compression leads to an increase in the compression efficiency of certain types of data, for example, in case of a large number of repeating sequences. In the most general case, one should expect that the compression coefficient of such an algorithm will be no worse than that of the already given ones. Although, in the worst case one can expect an increase in the volume of output data.

The generic adaptive compression algorithm (GACA) is a combination of all the algorithms observed above. Moreover, it should be noted that for it the advantage of adaptability is revealed on real data. If the data flow from the sensors is not homogeneous, then different algorithms work at different stages. Although, each individual frame is compressed by one algorithm only. This assumption is related to the nature of telemetry data, as dividing the blocks, overhead information will be a significant share in the resulting package. In case of a small input frame size, this fraction will affect the amount of transmitted data rather greatly.

The overhead information of an adaptive algorithm is recorded at the beginning of the output flow and includes the algorithm number in the list of implemented ones in the system and its set of parameters. In the considered implementation of the algorithm, two bits are allocated to the number of the selected algorithm. The transmitting and receiving sides agree on the possible compression methods before the transmission begins.

4. Testing compression algorithms

4.1. Block Compression Algorithms

To test the algorithms, telemetry data obtained from the data collection system from power stations was used. Tests were carried out on samples from 100 to 500 frames because the average compression coefficient with a smaller amount is strongly affected by the transient behavior of the sensor readings but a larger number of frames slightly influence the average value of the compression coefficient. Software that implement the algorithms are written in C++ in QT Creator programming environment.

A difference algorithm with variable-length codes is often used to compress measurement data in the IMS, since it is simple to implement. It also has high speed of operation and time symmetry of operation when encoding and decoding. This is important when the collection system is performed online [5]. For this reason, the authors used this algorithm as a reference for a comparative evaluation of the operational results of the proposed algorithms. The maximum possible (theoretical) value of the compression ratio for this algorithm for eight-bit data is 8.0. A histogram of the algorithm compression coefficient values is shown in Figure 2. Since this algorithm is the simplest of the submitted ones, it does not provide large compression ratios, but is characterized by the most quiescent operation. The compression ratio of the vast majority of frames (approximately 90%) is in the range from 5.5 to 7.7.

A higher compression ratio is arranged by an algorithm using standard compression methods applied to a compressed sequence. So, the modified RLE algorithm shows one of the best results in the average compression ratio and the highest maximum compression ratio among all other algorithms. It is caused by the fact that the values of the sensors in adjacent samples are strongly correlated. Therefore, applying differential compression to the same sensor readings in adjacent samples causes a large number of consecutive 0s in the consequent frame. RLE algorithm processes such sequences efficiently and, as a result, increases the final compression ratio. The studies prove that the compression ratio attains the maximum value 117 for this algorithm, the average value is 11, and the compression ratio was from 7 to 11 for most frames. This is mainly due to the properties of the compressed data. So, finding the same frames, the algorithm transfers the block size only and duplicate content.
Figure 2. Histogram of the compression coefficient values by the differential method with variable length codes for 500 frames sample.

The efficiency of the compression algorithm based on linear extrapolation depends on the behavior of the change in the sensor readings mostly. That is why the worst results were obtained on the test data for it. The effectiveness of this algorithm can be improved if it is used in combination with other algorithms, for example, with block algorithms as part of an adaptive compression algorithm. In this case, the main task of linear extrapolation is to work at the moment of full load on the compression system, i.e. when passing unsteady sections with a constant trend. Under normal conditions, when the function of changing the average value is of zero order, this algorithm becomes unproductive, which is well monitored by the histogram.

Some decrease in the compression coefficient and increase in the amount of data on the first measures of the algorithm is due to the fact that the first two tic marks are used to construct an extrapolational polynomial/ multinomial. Consequently, a rather sudden change of the input data values leads to the fact that the algorithm output becomes the original samples and overhead information, which reduces the compression ratio.

The study of the algorithm on artificial test data whose element type is that of signals with a well sounded prevalence of increasing and decreasing sections showed that the algorithm provided the highest value of the compression coefficient (about 12), and block algorithms are about 7, which determines this algorithm application in the adaptive compression algorithm.

Originally expected that the compression algorithm with the accumulation history should show the best compression efficiency if the average value of the input data flow is not equal to zero and large deviations in the samples are balanced out by the readings of preceding frames. However, the studies showed that the maximum value of the compression ratio was small, but the variation of the average value was the smallest. The bottom compression ratio is kept approximately the same as for the rest of the algorithms.

The data obtained show that the appearance rate of the compression coefficients are distributed proportionally enough and are shifted aside high values. It proves both good performance and stable algorithm functioning. The reason for this is that the first samples are used to accumulate history, while soft deviations of indicators affect the generated average value, since it becomes close to these indicators. While non-stationary flow is increasing, the average value begins to change more intensively and the values in preceding frames begin to compensate for the values in the current frame.

It should be noted that the average data value in this algorithm is added to the current sensor indicator, which was done to ensure the correct operation of the algorithm in short unstable sections. In this case, nonstationarity that has arisen will be compensated by the introduced mean value to some
extent. Thus, such an algorithm should work more efficiently on unstable data compared to other algorithms.

4.2. Adaptive Compression Algorithms
The results of the following algorithms reported here showed the best characteristics in pilot exploration:

- differential compression with variable length codes;
- ABC algorithm;
- ABCSB algorithm.

For ABC algorithm, the maximum value of the compression coefficient is 58, and similar results were obtained with different parameters of the algorithm.

The implementation of ABCSB algorithm explored determines the dependences for no more than two samples. So it provides for worse results than the extrapolation algorithm for some input data sets. The lag areas with monotonous increase and decrease in readings are especially strong. The analysis result suggests that the average performance of this algorithm is a little lower than that of the ABC.

The study results of the cumulative adaptive algorithm are shown in Fig. 3. The studies showed that the predominance of the second algorithm was revealed on the telemechanical data, and the differential algorithm was never chosen. The average compression ratio increased, while the maximum compression ratio has remained steady.

![Figure 3. A histogram of compression ratios for GACA.](image)

When testing artificially generated data, the efficiency of the ABCSB algorithm was revealed. As a result of it the other algorithms were not selected, and the compression ratio turned out to be close to its average value during compression of the exhaustive data flow.

5. Conclusion
The algorithms developed within this approach and considered in this paper are quite efficient in processing data with various statistical properties. An adaptive measurement data compression algorithm has been developed that implements the adaptive approach according to a algorithm. The maximum value of the compression coefficient when working with test data received from the telemetry data acquisition system of power stations is 117, and the value of the average compression ratio is not less than 8.0. In this case, for the generalized adaptive algorithm, the average compression ratio increases to 9.0. It should be emphasized that the average value of the compression ratio for the considered algorithms is equal to or exceeds the maximum possible value of the compression ratio for the reference algorithm. Thus, the studies showed that the approach suggested to the development of
new compression algorithms, which involves considering the correlation of different data sources, appears promising and deserves progressive development and application.

Reference
[1] Fout N and Ma K An 2012 Adaptive Prediction-Based Approach to Lossless Compression of Floating-Point Volume Data *IEEE Transactions on Visualization and Computer Graphics* **18** 2295–304
[2] Yaguchi K, Kobayashi N and Shinohara A 2014 *Proc. Data Compression Conf* 434
[3] Trofimov V K and Khramova T V 2017 Optimal output-uniform coding for a union of different sets of sources *Optoelectronics, instrumentation and data processing*. **53** 43-50
[4] Salomon D 2006 *Data compression: the complete reference* (New York: Springer-Verlag) p 900
[5] Sayood K 2006 *Introduction to data compression* (New York: Wiley–Interscience) p 680
[6] Goyal M, Tatwawadi K, Chandak S and Ochoa I 2019 *Proc. Data Compression Conf*. 575
[7] Naqvi S S, Naqvi R, Riaz R A and Siddiqui M F 2011 Optimized RTL design and implementation of LZW algorithm for high bandwidth applications *Electrical Review*. **4** 279–85
[8] Pujar J H and Kadlaskar L M A New Lossless 2010 Method of Image Compression and Decompression Using Huffman Coding Techniques *J. Theoretical and Applied Information Technology* **15** 18–23
[9] Jain A 2009 *Proc Int. Conf. on Signal Proc. Systems* 762-5
[10] Woottton C *A practical guide to video and audio compression* (New York: Springer-Verlag) p 787
[11] Duarte M, Shen G, Ortega A and Baraniuk R 2012 Signal compression in wireless sensor networks *Philos. Trans. R. Soc. A* **370** 118–35
[12] Villafranca A G, Portell J and García-Berro E 2010 *Proc. Data Compression Conf*. 554
[13] Yamagiwa S, Morita R and Marumo K Bank 2016 *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)* 1962-5
[14] Mittal S and Vetter J 2015 A Survey Of Architectural Approaches for Data Compression in Cache and Main Memory Systems *IEEE Trans. on Parallel and Distributed Systems*
[15] Huang B, Huang H, Chen H, Ahuja A, Baggett K, Schmit T J and Heymann R W 2004 Data compression studies for NOAA hyperspectral environmental suite (HES) using 3D integer wavelet transforms with 3D set partitioning in hierarchical trees *Proc. SPIE* **5238** 255–65
[16] Ringwelski M, Renner C, Reinhardt A, Weigely A and Turau V 2012 *Proc. Int. Energy Conf. (ENERGYCON)* 935–40
[17] Levenets A V 2018 The Basic principles and methods of the system approach to compression of telemetry data *Journal of Physics: Conference Series* **944** 012074
[18] Bogachev I V, Chye E U and Levenets A V 2018 *Proc. Int. Multi-Conf. on Industrial Engineering and Modern Technologies (Vladivostok, Russia)* (New York: Springer-Verlag)