# Layered lossless compression method of massive fault recording data

Jinhong Di<sup>\*</sup>, Pengkun Yang, Chunyan Wang, Lichao Yan, School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou, 450046 China

Received: July 3, 2021. Revised: December 21, 2021. Accepted: January 2, 2022. Published: January 3, 2022.

Abstract—In order to overcome the problems of large error and low precision in traditional power fault record data compression, a new layered lossless compression method for massive fault record data is proposed in this paper. The algorithm applies LZW (Lempel Ziv Welch) algorithm, analyzes the LZW algorithm and existing problems, and improves the LZW algorithm. Use the index value of the dictionary to replace the input string sequence, and dynamically add unknown strings to the dictionary. The parallel search method is to divide the dictionary into several small dictionaries with different bit widths to realize the parallel search of the dictionary. According to the compression and decompression of LZW, the optimal compression effect of LZW algorithm hardware is obtained. The multi tree structure of the improved LZW algorithm is used to construct the dictionary, and the multi character parallel search method is used to query the dictionary. The multi character parallel search method is used to query the dictionary globally. At the same time, the dictionary size and update strategy of LZW algorithm are analyzed, and the optimization parameters are designed to construct and update the dictionary. Through the calculation of lossless dictionary compression, the hierarchical lossless compression of large-scale fault record data is completed. Select the optimal parameters, design the dictionary size and update strategy, and complete the lossless compression of recorded data. The experimental results show that compared with the traditional compression method, under this compression method, the mean square error percentage is effectively reduced, and the compression error and compression rate are eliminated, so as to ensure the integrity of fault record data, achieve the compression effect in a short time, and achieve the expected goal.

Keywords—Fault recording data, lossless compression, LZW Algorithm, optimal parameters, query the dictionary.

### I. INTRODUCTION

The electric power recording system is a kind of device that automatically records related electric quantities in the power system. It can record the changes of system current, voltage and derived quantities caused by serious disturbances such as short circuit fault, system oscillation, frequency collapse and voltage collapse, such as the whole process changes of active power, reactive power and system frequency, which is mainly used to detect the action of relay protection and automatic safety device. In order to understand the variation law of electric parameters in the transient process of power system, so as to check and verify the correctness of calculation program and model parameters of power system [1]. By analyzing and processing the electric quantity recorded by the electric power recording system accurately and timely, and displaying the analysis and processing results, the operators and analysts can better and faster understand the cause and condition of the fault, and quickly solve the fault, reducing the loss caused by the fault. The saved data can be used for statistical analysis of operators, making reasonable configuration for future power grid design and equipment protection according to specific conditions, and also conducive to effective prediction of possible faults [2].

Under this background, a new problem also arises. The fault signal of power system is a wide-band non periodic signal, which includes power frequency, high frequency and local pulse signals. To record this non-stationary signal, it needs to have a high sampling frequency and recording accuracy, resulting in a huge capacity of sampling data that the equipment needs to store. At the same time, when a large amount of recording data is uploaded to the control center. It will increase the transmission time and reduce the reliability of data transmission, which makes the storage and transmission of power system recording signal face a huge challenge [3]. Therefore, it has become an urgent and challenging task to find a high compression ratio, low fidelity, good denoising effect, fast real-time or quasi real-time compression method for fault recording data. At present, the common method of power sampling data compression is lossy compression. It compresses

the secondary information data, reduces the amount of data at the expense of some data, and greatly improves the compression elimination rate. Representative algorithms include pulse coding, prediction coding and transform domain coding, such as Fourier transform coding, discrete cosine coding, wavelet and wavelet packet coding. However, this compression method will lose some data after all, which will damage the integrity of the data, lead to some errors in the application of the data, and affect the accuracy of the application results.

In view of the problems of the above compression methods, a layered lossless compression method for massive fault recording data is proposed. Lossless compression is the compression of the file itself. Like other data files, lossless compression optimizes the data storage mode of the file. It uses an algorithm to represent the repeated data information. The file can be completely restored without affecting the file content. For digital images, it will not cause any loss of image details. The representative algorithms are Huffman coding, LZ77 and LZ SS compression algorithm, LZ78 and LZW compression algorithm, arithmetic coding algorithm and run length coding [4]. The advantage of this compression method is that it can save the quality of data better, but the compression error of this method is relatively high. An improved LZW compression algorithm is selected for layered lossless compression of massive fault recording data. Finally, the method is tested, the integrity of the recorded data is guaranteed, the compression error is low, the compression speed is accelerated, and the expected goal of the study is achieved.

# II. LAYERED LOSSLESS COMPRESSION OF MASSIVE FAULT RECORDING DATA

Power system recorder is an important equipment for recording electrical parameters of power system. The data recorded by the recorder provides the main basis for fault analysis of power system and analysis and evaluation of various protection actions, and plays an important role in accident investigation and analysis. In this context, because the fault signal of power system is a wide-band non periodic signal, which contains many signal components, to record this non-stationary signal, it needs to have a high sampling frequency and recording accuracy, resulting in a huge capacity of sampling data that the equipment needs to store; At the same time, when a large amount of recording data is uploaded to the control center, it will increase the transmission time and reduce the data transmission Reliability. This makes the storage and transmission of power system recording signal face a huge challenge [5]. In view of this situation, it is necessary to compress the fault recording data.

Data compression is the process of representing all the signals sent by the source with the least code words and reducing the signal space of a given message set or data sampling set. Data compression methods of fault recording can be divided into lossless compression and lossy compression.

Lossy compression is irreversible compression. In the process of lossy compression, some information will be lost.

Therefore, it is impossible to reproduce the compressed data without distortion when restoring the compressed file. It achieves better compression effect at the cost of losing some information [6].

The working mechanism of lossless compression is to remove or as far as possible remove the duplicate and redundant parts of the data without losing any information in it, so as to ensure that the compressed data is completely consistent with the data before compression after restoration. Because lossless compression will not bring any loss to the information carried by data in the process of compression and decompression, it has been widely used in many occasions requiring accurate data recovery and lossless information transmission. For example, text data, program code, etc., but the compression ratio of lossless compression method is limited by the theory of data statistical redundancy. At present, there are many kinds of lossless compression methods, such as Huffman code, run code, LZW code, arithmetic code and so on [7].

In this study, LZW coding algorithm is selected for layered lossless compression of massive fault recording data. The research ideas are as follows: Firstly, the common dictionary search methods and storage structure are analyzed; Secondly, an LZW algorithm is proposed to improve the dictionary search method [8]. The improved LZW algorithm uses the multi tree structure to build the dictionary, and uses the multi character parallel search method to query the dictionary. At the same time, the influence of different dictionary size and update strategy on the compression performance of the algorithm is discussed. The optimal parameters are selected to design the dictionary size and update strategy to complete the lossless compression of the recorded data. Finally, the comparative analysis is carried out through MATLAB software simulation. The research flow is shown in Fig. 1.



Fig. 1 Research flow of layered lossless compression method for massive fault recording data

# A. LZW Algorithm and Problems

LZW algorithm is a popular variant of LZ78, which was developed by Terry Welch in 1984. LZW algorithm first initializes all the characters in the alphabet to the dictionary. It usually uses 8 characters. Before entering any data, it takes priority to occupy the first 256 items (0-255) of the dictionary [9].

In the process of compression and decompression, LZW

algorithm needs to build dictionary dynamically. The cell density of dynamic dictionary is:

$$P_f = L \sum_{j=1}^n p_j \rho S_j \tag{1}$$

In the formula, *n* represents the number of dictionary units, *L* is the length of the dictionary unit,  $\rho$  is the density of the dictionary unit,  $S_j$  is the unit area of the *j* -th dictionary unit,  $p_i$  is the density probability of the *j* -th dictionary unit.

The calculation formula of dictionary unit probability density is as follows:

$$P_{j} = k \sum_{j=1}^{n} f(B_{x}^{2} + B_{y}^{2})$$
(2)

In the formula, k represents the occurrence probability of dictionary unit,  $B_x$ ,  $B_y$  represent the length of the dictionary horizontally and vertically.

After the calculation of the unit density and probability density of the dictionary, the dictionary is compressed. In the compression process, the dictionary index value is used instead of the input string sequence, and the unknown string is added to the dictionary dynamically:

$$R_j = \frac{P_f \rho r \delta}{\mu} \tag{3}$$

In the formula,  $\mu$  is the compression rate of the dictionary, r is the compression coefficient of the dictionary,  $\delta$  is the length of the dictionary.

After the dynamic addition of strings, you need to decompress the strings. In the decompression process, search the corresponding string in the dictionary according to the compressed code word as the output, and update the dictionary constantly at the same time:

$$J = \sum_{i=1}^{n} (f_{xi} + \Delta f_{xi}) R_j / C_i$$
(4)

In the formula,  $C_i$  represents the linear factor of dictionary update constraint,  $f_{xi}$  and  $\Delta f_{xi}$  represent the update and update changes of dictionary units respectively.

From the LZW compression and decompression process, we can see that LZW algorithm has many advantages, such as simple algorithm, easy to implement in hardware, good compression effect and so on. However, the compression delay of LZW algorithm is related to the size of the dictionary and the search method of the dictionary. The conventional sequential search method needs to search the whole dictionary in each

compression process, resulting in a long search delay and a long compression delay of LZW algorithm, which can not meet the requirements of high real-time of aerospace telemetry system [10]. In order to solve the above problems and obtain better compression effect, the LZW algorithm is proposed to improve the dictionary search method, which uses the multi character parallel search method to query the dictionary globally. At the same time, the dictionary size and update strategy of LZW algorithm are analyzed, and the optimal parameters are designed to build and update the dictionary.

# B. Dictionary Search Method

At present, LZW algorithm mainly uses three dictionary search methods: sequential search, parallel search and hash table search. The traditional LZW algorithm uses the sequential search method to query the dictionary from the top to the bottom. In order to further reduce the compression delay, parallel search and hash table search are proposed one after another. Compared with the sequential search method, these two search methods can greatly reduce the search time of the dictionary and significantly speed up the search speed of the dictionary [11]. The improved LZW algorithm uses the parallel search method to query the dictionary. The search formula of the dictionary is:

$$S = \sqrt{\sum_{i=1}^{n} \frac{J^2}{n} - \lambda_i} \tag{5}$$

In the formula,  $\lambda_i$  represents the dictionary query remainder class.

Parallel search method is to divide a dictionary into several small dictionaries with different bit width to realize parallel search for dictionaries. Specifically, a single dictionary in LZW algorithm is divided into several small dictionaries, and the size and bit width of each small dictionary are different. Generally, the bit width of each small dictionary after the first small dictionary is increased by one byte [12]. For example, for a given Dictionary of size 1024, the dictionary can be divided into ten different dictionary combinations, as shown in Table I. Among them, the first dictionary with a size of 256 is generally a virtual dictionary, which contains initialization characters of 0-255, occupying only the first 256 addresses, not the actual hardware storage space. The combination of dictionaries is shown in Table I.

Table I Dictionary combination mode										
Type		Dictionary division method								
1	256	256	256	256						
2	256	256	256	128	12 8					
3	256	256	128	128	12 8	12 8				
4	256	256	128	128	12 8	64	64			

5 6 7	256 256 256	256 128 128	128 128 128	128 128 128	64 12 8 64	64 64 64	64 64 64	64 64 64	64 64	64			
8 9	256 256	128 128	128 64	64 64	64 64	64 64	64 64	64 64	64 64	64 64	64 64	64	
10	256	64	64	64	64	64	64	64	64	64	64	64	64

# C. Dictionary Storage Structure

LZW algorithm can store dictionary entries in two ways: One is direct storage structure, that is, the dictionary entries composed of character prefix and character suffix are directly stored in a certain address of the dictionary; The other is multi tree storage structure. The multi fork tree structure refers to that each node in the first level sub node tree composed of empty string as the tree root and 256 basic characters below may have many sub nodes. Each sub node includes two pieces of information, one is a pointer to the parent node, the other is a new character. When implemented in FPGA, if the dictionary size of LZW algorithm is 4096, the master node pointer includes 12bit, and the new characters include 8bit. The master node and the new characters are stored in the corresponding address as new dictionary entries, and the bit width of each dictionary entry is fixed to 20bit, which effectively improves the utilization efficiency of the storage unit [13].

Volume 16, 2022

Suppose the input string sequence to be compressed is {95, 3135, 95, 3130, 252, 3135, 231, 48} LZW algorithm with dictionary size of 4096 is used to compress the above string sequence. In the process of compression, multi tree structure is used to build the dictionary, and sequential search method is used to query the dictionary [14]. The output of the dictionary and the compressed code word based on the multi tree storage structure is shown in Table II.

Dictionary address	Parent node pointer (12bit)	New character (8bit)	Code word (12bit)
(0X000)	4095(0XFFF)	0	
(0X001)	4095 (0XFFF)	1	
(0X002)	4095 (0XFFF)	2	
(0X0FF)	4095 (0XFFF)	255	
(0X100)	95 (0X05F)	3	(0X05F)
(0X101)	3 (0X003)	135	(0X003)
(0X102)	135 (0X087)	95	(0X087)
(0X103)	256 (0X100)	130	(0X100)
(0X104)	130 (0X082)	252	(0X082)
(0X105)	252 (0X0FC)	3	(0X0FC)
(0X106)	257 (0X 101)	231	(0X101)
(0X107)	231 (0X0E7)	48	(0X0E7)
(0XFFE)	Parent node pointer	Child node character	
(0XFFF)	NULL	NULL	

Table II LZW	algorithm multi	i tree dictionary	storage structure

Among them, the address in brackets is the hexadecimal representation of dictionary address or code word, and the address or character representation of parent node pointer in decimal representation is outside brackets. In order to fill the first 256 initialization characters of the dictionary with a fixed width of 20bit, a constant value (0xfff) is added before the first 256 initialization characters, and the dictionary item with address 4096 is discarded and no longer used [15].

As can be seen from Table II, LZW algorithm based on multi tree storage structure only changes the storage mode of dictionary entries. Each dictionary entry includes 12 bit node pointer and 8-bit new characters. The width of dictionary entries is fixed to 20 bits. Compared with the conventional direct storage structure, it improves the utilization efficiency of the storage unit and does not affect the compression rate of the algorithm.

# D.Dictionary Size

In the process of compression and decompression, LZW algorithm needs to build dictionary dynamically, and the size of dictionary affects the final compression rate of LZW algorithm. The larger the dictionary is built, the easier it is to find the matching string sequence in the dictionary. The lower the compression rate is, the better the compression effect is. On the contrary, the smaller the dictionary is built, the higher the compression rate is, and the worse the compression effect is. At the same time, the size of the dictionary also affects the compression time. The larger the dictionary is established, the longer it takes to search the whole dictionary, and the longer the compression time is. On the contrary, the smaller the dictionary is established, the shorter the compression time is. Because the decompression process is to map the codeword directly to the corresponding string sequence in the dictionary, so it is not

necessary to search the whole dictionary dynamically, so the decompression time is basically not affected by the size of the dictionary [16]. In the hardware implementation of LZW algorithm, the dictionary is implemented by the RAM resource in FPGA, while the RAM resource in FPGA is limited, and the dictionary cannot grow unlimited. Therefore, the fixed size dictionary is generally used to realize LZW compression algorithm, and the selection of dictionary size needs to consider three factors: compression rate, compression time and occupied RAM resource [17].

Because the dictionary size of LZW algorithm is related to the compression rate and compression time of telemetry data, the relationship between the dictionary of LZW algorithm and compression rate and compression time can be analyzed by MATLAB software simulation, so as to select the optimal Table III Four strategies

dictionary size parameter value.

#### E. Achieve Lossless Compression

The size of the dictionary established by LZW algorithm in the process of compression and decompression is fixed. If the dictionary overflows, the dictionary needs to be updated. Different update strategies will affect the final compression rate. For example, if a 4K dictionary is full, it can't fill in new dictionary entries. In this case, the expansion of the dictionary theoretically increases the dictionary capacity, but the longer dictionary address pointer leads to poor compression effect [18]-[21]. At present, the updating methods of dictionaries mainly include stop updating, delete reconstruction, FIFO mode and LRU mode, etc. The following four dictionary updating strategies are defined and analyzed, as shown in Table III.

Table III Four strategies of dictionary updating					
Method	Definition				
Stop updating	Stop updating is to stop updating the dictionary once it is full				
Deleting and rebuilding	Deleting and rebuilding means that if the dictionary overflows, the whole dictionary will be				
	deleted and a new dictionary will be created again. However, in deletingRetain the first 256				
	initialization characters of the dictionary in addition				
FIFO	FIFO means first in, first out. When the dictionary is full, if a new dictionary entry needs to be				
	added to the dictionary, the storage address of the dictionary entry will start again from 257				
LRU	LRU (least recently used) method is to count the usage times of each dictionary entry at the				
	same time in the process of compression, and put the dictionary entry address with the least usage				
	times in the first place of the usage times statistics table through sorting method. When the				
	dictionary is full, only the dictionary entry with the least usage times in the dictionary is replaced				

During the updating of the dictionary, some losses will occur in the process of data compression due to deletion, addition and other steps. Therefore, it is necessary to determine the loss in the process of data compression in order to improve the quality of later data compression. The calculation formula of dictionary compression loss is:

$$F_{i}(x) = S(2^{x} - 1) \frac{n}{\min\{g_{ij}\}}$$
(6)

In the formula,  $F_i(x)$  represents the loss in dictionary compression,  $\min\{g_{ij}\}F_i(x)$  represents the set of dictionary compression loss knots, *S* represents the loss factor, *n* represents the actual number of dictionary compression.

According to the determination of the dictionary compression loss determined above, the layered lossless compression of large-scale fault record data is completed through the calculation of lossless dictionary compression, that is:

$$O_i(x) = \frac{F_i(x)(o_i(x)) - m_i}{\delta_i}$$
(7)

In the formula,  $o_i(x)$  is the compression loss function,  $m_i$  is the mean value,  $\delta_i$  is the standard deviation.

#### **III. SIMULATION EXPERIMENT ANALYSIS**

In order to verify the overall performance of the proposed lossless compression method, a comparative experiment is carried out. In MATLAB environment and C environment, the original radar echo data is compressed, and the compression performance is compared with three general lossless compression algorithms (Huffman coding algorithm, run length coding algorithm, arithmetic coding algorithm). The hardware and software platform used in the experiment are: WinXP system, Pentium 42.4 processor, 512M memory, MATLAB 7.0.1 simulation environment and VC + + 6.0 development environment. Specific experimental environment is as shown in Fig. 2:



Fig. 2 Experimental environment

# A. Sample Collection

The recorder in Fig. 3 is used to collect the real fault wave recording data of 110 kV line in a 220 kV substation of central China Power Grid in period B as the simulation verification sample. The sampling frequency is 1024 hz and the sampling time is 0.4 s.



Fig. 3 Schematic diagram of wave recorder

#### B. Recording Data Processing and Waveform Display

Firstly, using the M programming language of MATLAB, open the file path obtained by the human-machine interface module, and directly load the wave recording data. Read the configuration file first, then the data file. After reading the data, according to the description in COMTRADE standard, the real analog value needs to convert the original stored data. After conversion, the real analog quantity will be displayed and saved for subsequent analysis. The flow is shown in Fig. 4.



Fig. 4 Recording data processing flow The collected real fault waveform of 110 kV line in period B is processed to get the original waveform as shown in Fig. 5.



C. Evaluating Indicator

(1) Mean square error percentage R:

$$R = \frac{\sqrt{\sum_{0}^{N} \left[ F(t) - F(t)' \right]^{2}}}{\sqrt{\sum_{0}^{N-1} F(t)^{2}}}$$
(8)

In the formula, F(t) is the original signal, F(t)' represents the signal reconstructed from the compressed fault recording data. The smaller F(t)' is, the smaller the error with the original fault recording data signal is and the closer it is to the original signal.

(2) Compress to eliminate error P:

$$P = \frac{M - N}{M} \tag{9}$$

In the formula, M is the original fault recording data, N is the fault recording data after compression. The smaller the compression elimination error is, the less the data after compression is, the more the data is eliminated, and the better the compression result is.

(3) Compression rate v:

$$v = \frac{M - N}{M \cdot t} \cdot 100\% \tag{10}$$

In the formula, t is time. The higher the compression efficiency is, the higher the compression elimination rate per unit time is, the better the average compression effect is.

#### D.Result Analysis



Fig. 6 The waveform of recording data compressed and decompressed by four compression methods

Compare the waveform of the recorded wave data obtained by compressing and decompressing the above four compression methods with the Fig. 6, and the percentage of mean square error obtained is shown in Table IV.

Fig. 5 Original waveform of fault recording data

#### Table IV Mean square error percentage

Method	Improved LZW encoding	Huffman encoding	Run length encoding	arithmetic coding
Mean square error	0.004	0.122	0.236	0.255
percentage (%)				

It can be seen from Table IV that after compression and decompression of the improved LZW coding algorithm, the mean square error percentage between the reconstructed fault recording data waveform and the original fault recording data waveform is only 0.004, which is far less than the Huffman coding algorithm, run length coding algorithm and arithmetic coding algorithm. Therefore, after the proposed method is compressed, almost no information is lost in the whole compression process, and the integrity of fault recording data is guaranteed.





As can be seen from Fig. 7, after the improved LZW coding algorithm is compressed, the maximum compression elimination error is no more than 0.5, which is lower than Hoffman coding algorithm, run length coding algorithm and arithmetic coding algorithm. It can be seen that after compressing the improved LZW coding algorithm, more fault record data are eliminated, and the compression effect is good.

(3) Compression rate

Table V Compression rate							
Method	Impro ved LZW encodi ng	Huffm an encoding	Run length encoding	arithme tic coding			
Compressi on rate(%/s)	8.56	5.32	5.63	6.47			

It can be seen from Table V that the improved LZW coding algorithm is used to compress the fault recording data with a compression rate of 8.56%/s, while the Huffman coding algorithm, run length coding algorithm and arithmetic coding algorithm need 5.32%/s, 5.63%/s and 6.47%/s. The results show

that the improved LZW coding algorithm has higher compression rate and can complete the compression work faster.

All in all, under the application of the improved LZW compression method, the integrity of fault recording data is guaranteed, and the compression effect is achieved in a short time, and the expected goal is achieved.

# IV. DISCUSSION

In the experiment, starting with the error of data compression, the errors of this method / Huffman coding algorithm, run length coding algorithm and algorithm coding algorithm in data compression under the same conditions are compared. According to the waveform range of the compression error of the four methods, it can be seen that the errors of the four methods are different, and the error range of the method in this paper is smaller. In the compression error elimination experiment, it is found that the lowest error eliminated by the four methods is the method in this paper; Finally, compared with the data compression rate, LZW coding algorithm has higher compression rate and can complete the compression work faster. Three groups of experiments verify the compression effectiveness of the proposed method.

# V. CONCLUDING REMARKS

To sum up, due to the problems of large error and low precision in traditional power fault record data compression, a new layered lossless compression method for massive fault record data is proposed in this paper. In the power system, the recording equipment records the electrical parameters of the whole process from before the fault to the normal operation of the system, and the amount of data is very large. Under the technical condition of the current dispatching system communication equipment, it will take a long time to transmit a large amount of data to the dispatching center. In order to solve this problem, the compression of recorded data is theoretically analyzed and simulated. LZW coding algorithm in lossless compression is used for lossless compression of fault record data. The multi character parallel search method is used to query the dictionary globally. At the same time, the dictionary size and update strategy of LZW algorithm are analyzed, and the optimization parameters are designed to construct and update the dictionary. Through the calculation of lossless dictionary compression, the hierarchical lossless compression of large-scale fault record data is completed. Select the optimal parameters, design the dictionary size and update strategy, and complete the lossless compression of recorded data. Finally, the online test shows that the proposed compression scheme can meet the requirements of high real-time performance and ensure better data compression effect.

# ACKNOWLEDGMENT

The study was supported by "HEVC rate control algorithm based on visual perception (Key project of technology department in Henan Province" (No. 172102210080); "Research on UVA aided wireless communication system transmission technology for B5G air-to-ground integration (Key project of technology department in Henan Province" (No. 192102210246); "Research on the key techniques of image processing in low and slow small target detection (Key project of Foundation of Henan Education Committee" (No. 20A510011).

### References

- [1] X. L. Du, "Research and application of the panoramic perceptive recording system in the smart grid," Electrical Automation, vol. 41, no. 4, pp. 8-10+49, 2019.
- [2] F. Wen, N. Wu, and X. Gong, "China's carbon emissions trading and stock returns," Energy Economics, vol. 86, pp. 104627, 2020.
- [3] W. L. Chiou, "Advantages of using slower recorder chart SP EEDS in liquid - chromatographic and gas chromatographic analyses," Clinical Chemistry, vol. 25, no. 1, pp. 197, 2019.
- [4] L. Q. Chen, L. X. Yan, and H. S. Sang, "High-throughput architecture for both lossless and near-lossless compression modes of LOCO-I algorithm," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 12, pp. 3754-3764, 2018.
- [5] W. J. Warwick, and L. G. Hansen, "Measurement of chloride in sweat by use of a selective electrode and strip-chart recorder," Clinical Chemistry, vol. 24, no. 2, pp. 381-382, 2019.
- [6] V. Ravanmehr, M. Kim, and Z. Y. Wang, "ChIPWig: A random access-enabling lossless and lossy compression method for ChIP-seq data," Bioinformatics, vol. 34, no. 6, pp. 911-919, 2017.
- [7] Y. H. Han, W. W. Sun, and B. H. Zheng, "Compress: A comprehensive framework of trajectory compression in road networks," Acm Transactions on Database Systems, vol. 42, no. 2, pp. 1-49, 2017.
- [8] X. Chen, and S. Zhang, "Three dimensional range geometry and texture data compression with space-filling curves," Optics Express, vol. 25, no. 21, pp. 26103, 2017.
- [9] Y. Wang, Q. X. Chen, and C. Q. Kang, "Sparse and redundant representation-based smart meter data compression and pattern extraction," IEEE Transactions on Power Systems, vol. 32, no. 3, pp. 2142-2151, 2017.
- [10] A. O. Zaid, M. Hachani, and F. Dufaux, "Phase-shifting digital holographic data compression," Journal of Optics, vol. 48, no. 3, pp. 412-428, 2019.
- [11] M. R. Nassar, J. C. Helmers, and M. J. Frank, "Chunking as a rational strategy for lossy data compression in visual working memory," Psychological Review, vol. 125, no. 4, pp. 486-511, 2018.
- [12] M. A. Islam, D. T. Anderson, and A. J. Pinar, "Data-driven compression and efficient learning of the choquet integral,"

IEEE Transactions on Fuzzy Systems, vol. 26, no. 4, pp. 1908-1922, 2017.

- [13] S. Dai, W. Gao, and C. Wang, "Damage evolution of heterogeneous rocks under uniaxial compression based on distinct element method," Rock Mechanics and Rock Engineering, vol. 52, no. 1, pp. 2631-2647, 2019.
- [14] V. N. Manju, and A. L. Fred, "AC coefficient and K-means cuckoo optimisation algorithm-based segmentation and compression of compound images," Iet Image Processing, vol. 12, no. 2, pp. 218-225, 2018.
- [15] D. Gierczycka, and D. Cronin, "Influence of the chest compression measurement method on assessment of restraint performance in side-impact crash scenarios," Journal of Biomechanics, vol. 75, pp. 53-57, 2018.
- [16] P. M. Pieczywek, and A. Zdunek, "Compression simulations of plant tissue in 3D using a mass-spring system approach and discrete element method," Soft Matter, vol. 13, no. 40, pp. 7318-7331, 2017.
- [17] C. Jiang, W. H. Wu, and Z. Ding, "LTE multimedia broadcast multicast service provisioning based on robust header compression," IEEE Transactions on Wireless Communications, vol. 17, no. 2, pp. 1161-1172, 2017.
- [18] Q. H. Yue, "Multi-domain physical system association rules data lossless compression simulation," Computer simulation, vol. 36, no. 4, pp. 111-114+198, 2019.
- [19] H. X. Gong, J. He, and D. Z. Geng, "Low bit rate fractal video image stratified compression method simulation," Computer Simulation, vol. 35, no. 7, pp. 135-138, 2018.
- [20] I. M. Osbert, and K. B. Hammond, "Letter: Avoidance of errors in the automated determination of serum enzyme activities by use of a strip-chart recorder," Clinical Chemistry, vol. 22, no. 8, pp. 1411-1412, 2019.
- [21] F. Wen, F. Min, and Y. J. Zhang, "Crude oil price shocks, monetary policy, and China's economy," International Journal of Finance & Economics, vol. 24, no. 2, pp. 812-827, 2019.



Jinhong Di, female, was born in October,1980, her title is Doctor of Engineering, Associate Professor. She graduated from Zhengzhou University in 2001, major in automation, with a baccalaureate. She graduated from Zhengzhou University in 2004, major in Communication and information system, with a master's degree and the Ph.D. degree in Communication. degree in Communication and information systems from Beijing University of Posts and Telecommunications in 2012. Now she is working in School of Intelligent Engineering, Zhengzhou University of Aeronautics. She mainly engages in the research of Multimedia communication, video coding and transmission. So far, she has published 20 papers and participated in 10 projects.



Pengkun Yang, male, was born in September, 1997. Received the B.S. degree in Communication Engineering from Zhengzhou University of Aeronautics in 2020. He mainly engages in the research of the mobile communication.

Chunyan Wang, male, was born in April, 1979. His title is Master of Engineering, Lecturer. He graduated from Xinyang Normal University in 2000, major in Applied Electronic Technology, with a baccalaureate. He graduated from China Academy of Telecommunications Technology in 2005, major in Electromagnetic field and microwave technology, with a master's degree. Now he is working in School of Intelligent Engineering, Zhengzhou University of Aeronautics. He mainly engages in the research of Intelligent signal processing technology and circuit system design. So far, He has published 10 papers.



Lichao Yan, female, was born in March, 1979. Her title is Master of Engineering, Lecturer. She graduated from Zhengzhou University in 2002, major in Automatic Control, with a baccalaureate. She graduated from Nanjing University of Posts and Telecommunications in 2006, major in Signal and Information Processing, with a master's degree. Now she is working in School of Intelligent Engineering, Zhengzhou University of Aeronautics. She mainly engages in the research of signal and information processing, video coding and transmission. So far, she has published 12 papers

# **Author Contribution**

Jinhong Di and Pengkun Yang: In order to overcome the problems of large error and low precision in traditional power fault record data compression, a new layered lossless compression method for massive fault record data is proposed in this paper. Chunyan Wang and lichao Yan: At the same time, the dictionary size and update strategy of LZW algorithm are analyzed, and the optimization parameters are designed to construct and update the dictionary. All authors discussed the results and wrote the manuscript.

# **Creative Commons Attribution License 4.0** (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en\_US