ATLANTIS PRESS
Advances in Computer Science Research (ACSR), volume 76
7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)

Lossless Compression Algorithm for Multi-source Sensor Data Research

Changchi Huang[1, a] and Yuwen Chen[1,b*]
[1]Hainan University, Danzhou, Hainan, 571737, China
[a]78387729@qq.com, [b]18876950776@139.com

Keywords: Multi-source sensor; DEFLATE algorithm; Sub-search; Compression algorithm

Abstract. In recent years, multi-source sensor system has been widely used in military, agriculture, forestry and other fields, the diversity of information performance, the huge number of information, the complexity of information relations, and require information processing real-time, have been greatly Beyond the general computer integrated processing capacity. In this paper, studies have found that the DEFLATE algorithm is consistent with the "neighbor principle" of sensor data. This is a combination of LZ77 and Huffman algorithms, which in theory should deal with multi-source sensor data more effectively. This paper proposes to increase the "secondary search" function during the DEFLATE algorithm matching process. This improvement effectively increases the possibility of finding a longer matching string, thus further reducing the coding length.

Introduction

Research Background. With the development of microelectronics technology, integrated circuits and their design technology, computer technology, modern signal processing technology and sensor technology, a variety of multi-sensor systems for complex application background have emerged[1]. Multi-sensor data fusion, it is the 20th century, 80 years to form and develop an automated information processing technology, refers to the data from multiple sensors for multi-level, multi-level, multi-level processing, resulting in new and meaningful information, and this new information is not available to any single sensor. However, the technology in the development process is facing a huge problem such as the problem, which multi-sensor data transmission, save a great impact, so now based on such data compression algorithm demand is increasingly urgent[2].

Study the Significance of Data Compression. Data compression is widely used in various fields. Company data backup needs to compress a lot of data to reduce storage space. In order to take full advantage of the channel, the TV signal, streaming media and other information need to be compressed. Data compression is usually divided into lossless compression and lossy compression[3]. For those who do not pay attention to details of the data, such as images, video, most people use lossy compression technology, such as MPEG, H.263, H.264 and other popular compression technology. And for programs, electronic files and other similar important information, you must use lossless compression technology. So that data recovery will not undermine its integrity. The research on lossless compression technology has been a long time, and the most important lossless compression technology is the LZ series compression algorithm and the minimum redundancy construction algorithm Huffman algorithm. In today's multi-sensor measurement system, in order to accurately measure some of the impact signal, the sensor output signal needs high-speed, high-resolution sampling. This will inevitably produce a large number of data needs to be stored, take up a lot of storage space, but also not easy to real-time transmission. Therefore, in order to ensure the measurement accuracy, in the sampling rate and resolution can not be reduced, the sensor signal compression storage, transmission is very necessary. In this paper,

LZ77 and Huffman combine a method to deal with multi-source sensor data, and achieved good results. The research of this topic will be beneficial to the further development of multi-source sensor system, effectively improve the efficiency of transmission and reduce the storage space used[4].

Development Status at Home and Abroad. Data compression technology is developed with the rapid development and extensive application of computer, digital communication, signal processing and other information technology, and has gradually developed into an independent discipline in recent years. The early stages of development from the end of the 18th century to the early 1950s were the embryonic period of data compression. Development period is the development of long-term, from the early 1950s to the 1970s, this period is the rapid development of theoretical basis period[5]. The theory of information created by the father of information c. E. Shannon laid the theoretical foundation for all data compression algorithms. The recent development of data compression began to become a theoretical and technically more systematic period of independent discipline, the statistical code has developed to a peak, most people along this idea to try to improve the algorithm, and always reach the ideal Effect. In recent years, more and more data compression software has been developed and used. Since the beginning of the 21st century, the rapid development of science and technology, related disciplines and the emergence of marginal disciplines, broaden the people's research ideas, to the development of data compression technology has injected new vitality. Therefore, compression technology based on multi-source sensor data will be developed rapidly and increasingly popular.

Data Compression Theory Foundation

Compression Technology. For compression technology, in fact refers to two algorithms[6]. One is the compression algorithm, which takes the input, generates a representation that requires fewer bits, and the other is the reconstruction algorithm, which performs the operation on the compressed representation to produce the reconstructed result. The schematic diagram of these operations is .shown in Fig. 1 This paper will follow the convention, the compression algorithm and reconstruction algorithm together, known as the compression algorithm. According to the requirements of reconstruction, data compression method can be divided into two categories: the first category is lossless compression, reconstruction results with the same; the second category is lossy compression, compression ratio of such compression is usually higher than lossless compression, But the reconstruction results may be different.

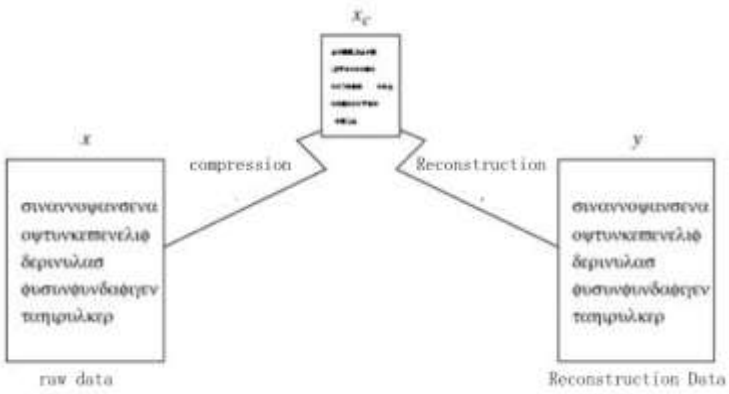

Figure 2.1. Compression and reconstruction

Introduction to Information Theory. Data compression and information theory are inseparable. In general, the establishment of the theoretical basis of information theory began in Shannon

published in 1948, known as the theory of information for the mountains of "communication theory of mathematics." He used a professional term entropy, which was originally used in thermodynamics to represent the degree of disorder in the physical system. In essence, the purpose of data compression is to eliminate the redundancy in the information, the degree of information redundancy is precisely the information entropy and related theorem with mathematical means to accurately describe. The steps of data compression are generally divided into three parts: modeling expression, quadratic quantization and entropy coding[7]. Firstly, a certain model is established, the original data is transformed into a model reference, and then the model reference is transformed into a quantization symbol by quadratic quantization. Finally, the quadratic quantized symbol is transformed into a compressed stream by entropy coding The The general steps for data compression are shown in Fig. 2.

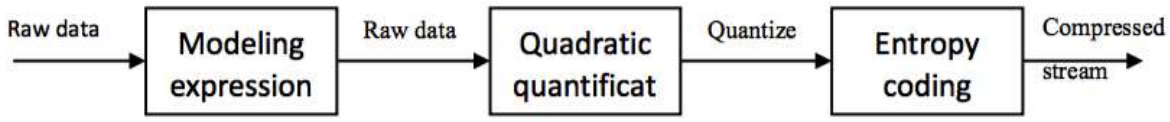

Figure 2. General steps of data compression

Let a discrete source be X, and the set $A = \{x_1, x_2, \cdots, x_n\}$ represents a set of N symbols. In the source, the probability of occurrence of each symbol is determined, that is, there is a probability distribution table $\{p_1, p_2, \cdots, p_n\}$ and is satisfied $\sum_{k=1}^{N} p_k = 1$. If the appearance of each symbol is irrelevant, or that independent, in the information theory, said the source X is no memory. Set the $p(x_i)$ probability that the X will issue the symbol x_i at a certain moment, Shannon defines the amount of information for $I(x_i)$ as Eqs. (1).

$$I(x_i) = -\log p(x_i) \tag{1}$$

And $i = 1,2,\cdots,n$。

The unit $I(x_i)$ is determined by the bottom of the logarithm. When 2 is the end, $I(x_i) = -\log_2 p(x_i)$, the amount of information in units of bits , when the natural number e for the end, the unit for the nat.

As can be seen from the above equation, the information function is a decreasing function. Specifically, the less likely the occurrence of a random event, the more information it is, the greater the likelihood of occurrence, the less the amount of information[8]. When X represents all the possible symbol sets of the source output, the information quantity of each symbol x_i of X is averaged by probability $P(x_i)$, and the average amount of information $I(x_i)$ of each symbol of X is Eqs. (2).

$$H(X) = \sum_{i=1}^{N} \log p(x_i) I(x_i) = -\sum_{i=1}^{N} p(x_i) \log p(x_i) \tag{2}$$

Because this expression is the same as the entropy in thermodynamics, it is called the entropy of the source X in information theory. The principle of information theory shows that if we know the entropy of a memoryless source X, we can always find a compression method for the information source so that the number of bits required for each symbol is as close $H(X)$ as possible. Shannon information theory that the source of the average amount of information is the theoretical limit of distortionless coding[9].

For discrete memoryless sources, when all information symbols are output with equal probability, the entropy of the source is obtained as Eqs. (3).

$$H(X) = -\sum_{i=1}^{N} \frac{1}{n} \log \frac{1}{n} = -\log \frac{1}{n} = \log n \quad (3)$$

This value is the maximum value of entropy and is defined as the maximum entropy $H_0(X)$.

Classic Lossless Compression Algorithm

Lossless compression technology, commonly known as general compression technology, also known as information retention coding, entropy coding, no distortion coding, etc., that is, according to a certain method of coding a large number of data to achieve information compression stored in the data compression process Does not allow the loss of precision, the compressed data should be able to recover by decoding to the original state before compression. Mainly used for text files, databases, program data and special application of the image data (such as fingerprint images, medical images, etc.) compression. This kind of algorithm has a low compression ratio, generally 1/2 ~ 1/5.

Usually compressed objects are text or numbers that require accurate data, lossless compression is an inevitable choice[10]. Lossless compression from the compression model can be roughly divided into statistical-based compression algorithm and dictionary-based compression algorithm. The specific classification chart is shown in Fig. 3.

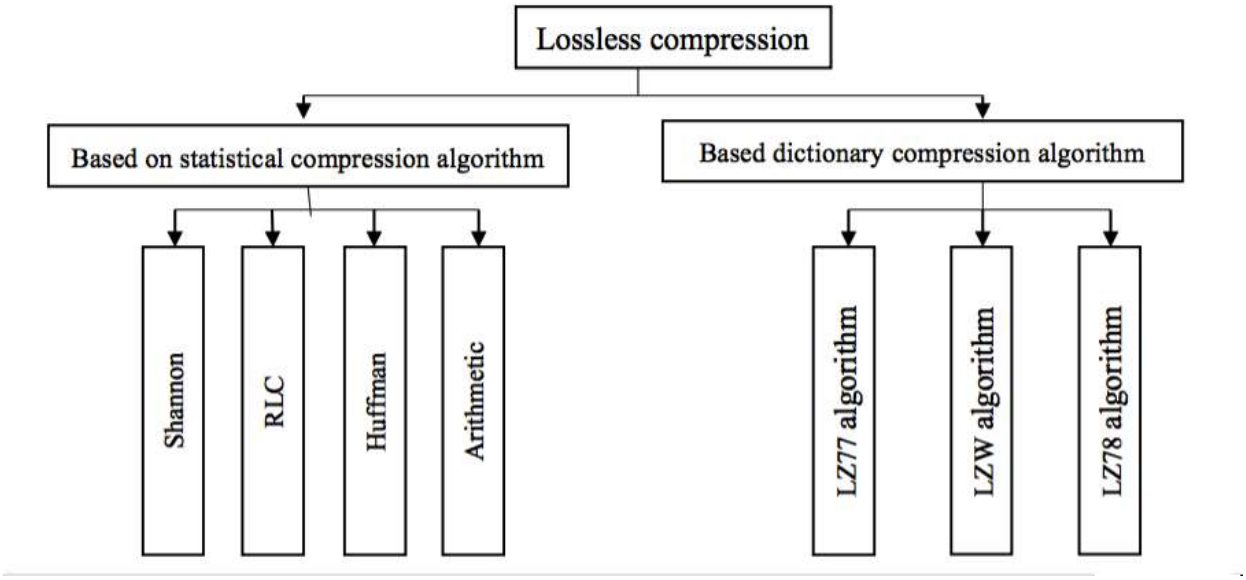

Figure 3. Classification of lossless compression algorithms

LZW Encoding. LZW algorithm is Welch proposed in 1984, it is an improved LZ78 algorithm, the application is very extensive. The main feature is the removal of the second field identified by the LZ78, that is, the LZW logo contains only a pointer to the dictionary, because the LZW method first initializes all the characters in the alphabet into the dictionary. The LZW method first initializes all the characters in the alphabet into the dictionary, most commonly the 8-bit character, and the first item of the dictionary (0 ~ 255) is already occupied. Because the dictionary has been initialized, the next input character can always be found in the dictionary.

The LZW encoder uses a very useful analysis algorithm called the greedy analysis algorithm. In the greedy analysis algorithm, each analysis must be serial to check the string from the character stream, the analysis of the longest string has been identified, that is, the dictionary has been the longest prefix. To be added to the dictionary, the string is also determined by this analysis, but it is

first expanded with its next input character into a new string, giving it a unique identifier, ie its code value.

The basic principle of LZW encoding is to first create a string table, put each first occurrence of the string into the string table, and a number that this number and the string in the string table position, and the This number is stored in the compressed file. If the string reappears, it can be replaced by the number that represents it, and the number is stored again in the file, and the string is discarded after the compression is complete. It applies to a large number of substrings repeatedly repeated the original data string, repeat the more, the better the compression effect. And vice versa is worse, and may even diminish[11].

Huffman Decoding Method. Huffman decoding is an inverse transform of the Huffman coding. When the receiving end decodes the received compressed compressed data stream, it must be decoded according to the coding tree, or according to the coding table compiled by the coding tree. The sender should therefore send the compressed data stream of the encoded tree or the encoded table to the receiver. In the above example, the coding table prepared by the Huffman coded coding tree is shown in Table 1. The receiving end can be decoded according to the code table or code tree sent by the sending end. From the root of the tree, read the first bit of the compressed data stream, if it is 0, then walk along the tree branch, if 1, then walk along the tree branch; then read into the compression Data flow second, to the leaves forward a section; so until the leaves, so get the original uncompressed codeword.

Table 1 Huffman coding table

source symbol	code	code length
a_1	0	1
a_2	10	2
a_3	110	3
a_4	1110	4
a_5	11110	5
a_6	11111	5

The Overall Design of the Algorithm. For multi-source sensor data, especially natural image data, we found that the use of traditional classical compression algorithm can not be effectively compressed, and sometimes there will be compression increases, and these compression algorithms in dealing with computer-generated images are very effective The It has been found that if you look at these images at the pixel level, you will find that there is almost no repetition pattern that can be compared to the text source, which makes the LZW algorithm, which generates the dictionary compression, not compress well, and because the dictionary needs extra Space, resulting in a larger compression result. Using LZ77 to test, found that LZ77 is not perfect to show the effect I expected, because the data is relatively low efficiency. After finding it later, this article finally takes the DEFLATE compression algorithm. DEFLATE algorithm is the basic theory of LZ77 dictionary algorithm, which is based on the output of LZ77 algorithm and the second Huffman coding. Because of its excellent performance in both compression ratio and compression speed, there is no need to make more improvements[12]. This paper only aims at the characteristics of multi-source sensor data.

DEFLATE Algorithm Matching Process Improvement. In the experiment found that finding the longest match string immediately after the replacement effect is not the best way. Because in some cases, the longest matching string of adjacent next strings may be longer. So this article has

made some improvements here, so that LZ77 with deputy search function, is a specific description. The first search finds a match of the symbol string beginning with the current symbol, and then performs a matching search with the symbol string beginning with the next symbol of the current symbol. If a longer match is found in "secondary search", then the current symbol is written in the form of next symbol, followed by sub-match, otherwise, only master match.

Under the precondition of satisfying the sub search, the sub search does not limit the number of times. That is, once the secondary search found a longer match string, will still be "vice search", if this time found a longer match string, then the last time to find the match string is not used. And the secondary search must meet two conditions, first, the next processing byte to start the string with a matching string; second, the current matching string of the matching length is less than a preset value.

Lossless Compression Algorithm to Achieve and Compare the Experimental Results

In this chapter, the Huffman algorithm, LZW algorithm and improved DEFLATE algorithm are used to compare the multi-source sensor data, and the experimental results are given.

Improving the Realization of DEFLATE Algorithm. The algorithm's internal Huffman encoding implementation, static Huffman compression, uses a fixed code at the time of encoding, and uses the same encoding at the time of decoding, so there is no need to save the Huffman tree's information in the result of the encoding output. The dynamic Huffman compression coding, in the construction Huffman tree when used in the character probability is based on the actual source to determine. Therefore, in the encoding output results will be added to build Huffman tree information. Using the probability in the actual source, the advantage is that the Huffman can be constructed more precisely, but because of the probability of transmitting the character probability information to the decoding side, this raises the advantage of probability accuracy to a certain extent. To this end, the DEFLATE algorithm uses the DEFLATE tree in order to minimize the space occupancy of the transmitted character probability information, and Huffman coding it at the time of transmission[13].

DEFLATE on the two algorithms are implemented, the actual use of a high compression rate to choose the output. In general, if the amount of data is less use static Huffman coding output, the amount of data can be used when the dynamic Huffman coding. Specifically, it can be configured according to the actual situation. The main data structures and functions are defined. The algorithm mainly implements the process.

(1) Initialize the corresponding data structure, such as the initialization of the additional length and distance encoding, the length of the word length Huffman tree and distance range Huffman tree node allocation space;

(2) The first scan of the source, the process of scanning the implementation of LZ77 algorithm, save all the success of the length of the match and the failure of the single word information. And the emergence of single-character, length range, distance range of frequency information statistics;

(3) Call generateTree based on the obtained frequency information to construct the character length tree and the distance tree;

(4) Call generateBitLength, respectively, have scanned the generated character length tree and distance tree, get each leaf node bit length information;

(5) Finally, according to the DEFLATE tree definition, according to the bit length information has been generated, call generateCode for each leaf node to generate coding. Finally, the LZ77 processing result which has been obtained is re-coded according to the coding of the leaf node.

Comparison of Algorithms and Analysis of Results. In the previous section gives a different

algorithm for binarydata1.dat compression decompression, you can see deflate algorithm and improved deflate algorithm for such data model, the compression ability is significantly stronger. This is mainly because the current compression algorithm is a common compression algorithm, and no distinction between data types, and this study is multi-source sensor data, according to the data model design algorithm is clearly more dominant. A number of multi-source sensor data are processed by the algorithm, and their results are plotted as a line graph, as shown in Fig. 4 and Fig. 5, where the size of each algorithm input data and the final compression ratio are plotted with different colors relationship.

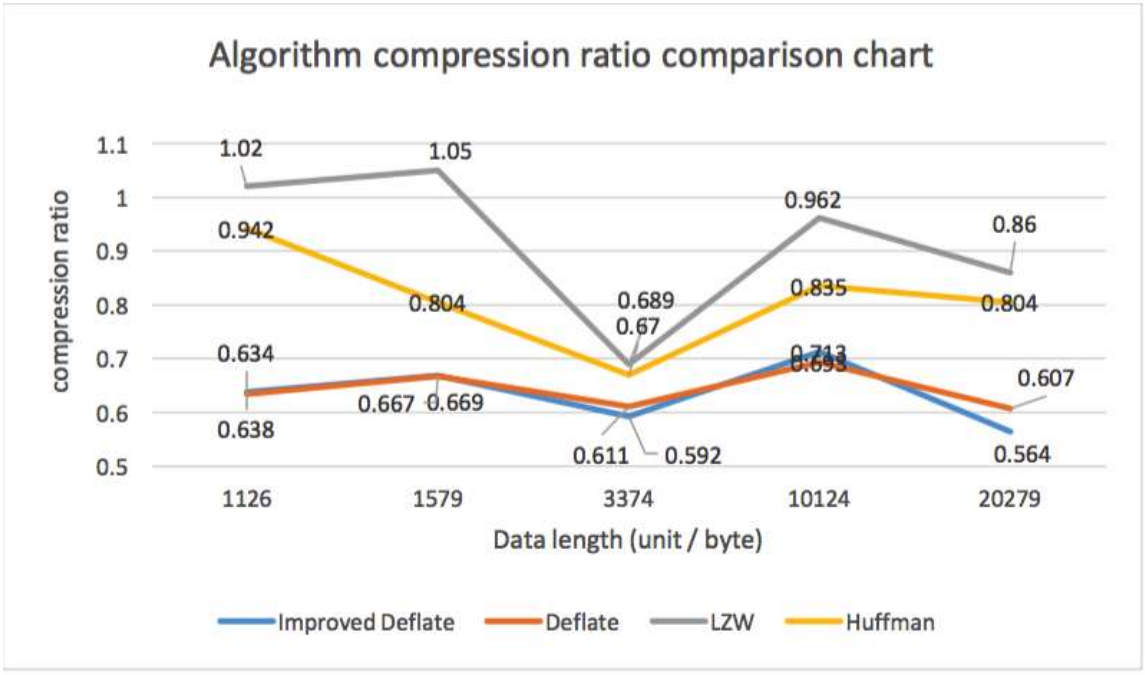

Figure. 4 The comparison of compression ratio (1)

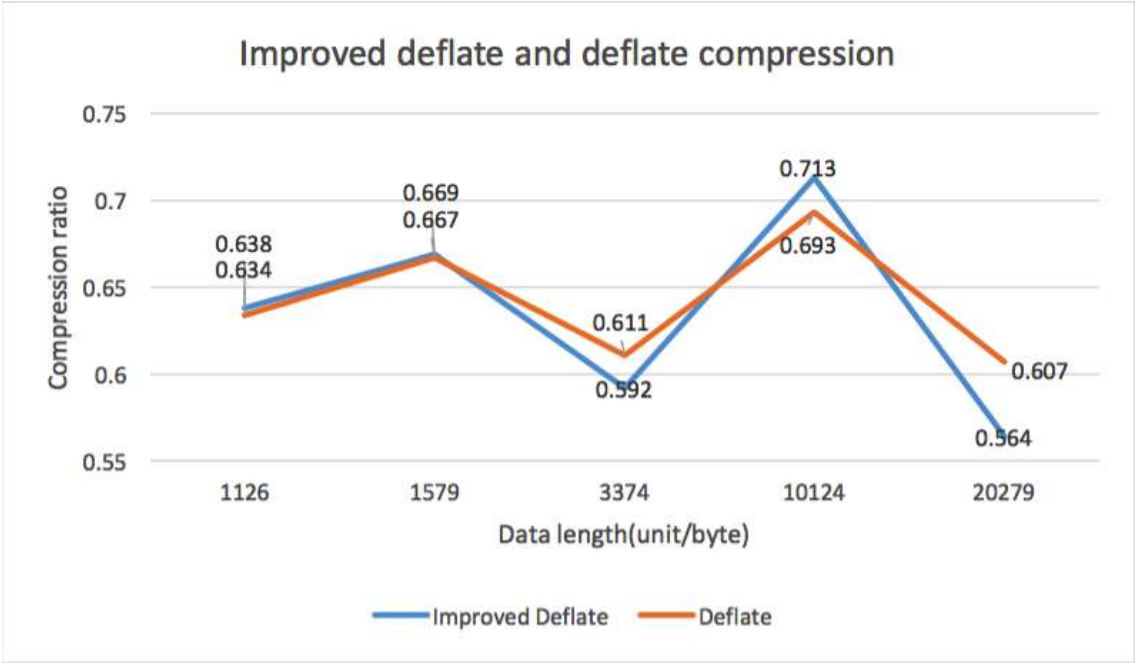

Figure. 5 The comparison of compression ratio (2)

As can be seen from the figure, LZW algorithm to deal with such data, the compression situation is very unstable, there is greater likelihood of compression than the original file is also large, this convenience is due to set the dictionary is too small, on the other hand because of such data The Huffman algorithm is similar to the LZW in the overall trend, but only in the test data, there is no compression becomes larger. Improved deflate in the overall trend with the original deflate little difference, but when the amount of data becomes larger, change complexity, improve deflate obvious advantages.

Summary

The object of this paper is based on multi-source sensor lossless data compression algorithm. The core of the paper is to use the DEFLATE algorithm for multi-source sensor compression. As the sensor data generally has the "neighbor principle" characteristics, and LZ77 algorithm is a sliding window of the compression method, you can effectively use a part of the previous string to encode the adjacent part of the string, which just meet the needs. On this basis, DEFLATE uses the second Huffman code to further compress the LZ77 encoding result, making the previously encoded string further shortened. So in the end, the DEFLATE algorithm is the main research object. In order to improve the compression performance, this paper adds the function of "vice search" to make it more likely to find a longer match string, so that the code is shorter. Increase the complexity of the algorithm. Through the final comparison experiment, DEFLATE algorithm is superior to the multi - source sensor data.

References

[1] A. Ganonlay: MULSIM Computer Program Research Laboratory, Vol. 1 (1999) No.1, p1.
[2] D. Le Gall: Comm.ACM, Vol. 1 (1999) No.3, p46.
[3] H. Afiffy: *Lossless Differential Compression Algorithm* (LAP Lambert Academic Publishing, Germany 2012).
[4] Bormin Huang: Satellite Data Compression (Springer, England 2014).
[5] M. Bassiouni, A. Mukherjee, and N. Ranganathan: Inform. Processing Manage, Vol. 25 (2015) No.3, p293.
[6] K. Sayood. Introduction to Data Compression[M]. Morgan Kaufmann Publishers. 2000.
[7] A. Weleh: IEEE Computer, Vol. 17 (1984) No.6, p8.
[8] V. Bhaskaran and K. Konstantinides：*Image and Video Compression Standards Algorithm* (Springer, England 2013).
[9] A. Storer: *Image and Text Compression* (Springer, England 1992).
[10] A. Weleh: IEEE Computer, Vol. 1 (2014) No.1, p25.
[11] A. Huffman: Proc IRE, Vol. 40 (1952) No.1, p1098.
[12]M. Sayood and P. Khalid：*Introduction to Data Compression* (Elsevier, Holand,2012).
[13] S. Ho and P. Law: Electron. Lett, Vol. 27(2013) No.2, p35.