A FPGA-based parallel method of bridge data compression

Jianlong Zhang1*

1 Henan Provincial Expressway Network Management Center, Zhengzhou 450016, China

*Corresponding author’s e-mail: zhengxdhw@gmail.com

Abstract. With the rapid development of computer technology, sensor technology, and communication technology, people have gradually designed and continuously improved the health monitoring system of bridge structure. Due to the uninterrupted long-term work of the bridge monitoring system, the long-term operation of a large number of sensors will generate massive amounts of data, and a lot of data to be transmitted from the data collector at the bridge site to the data monitoring center requires the use of a wireless network with limited speed, which will consume a lot of time, thereby bring such negative effects as reducing transmission efficiency, reducing throughput, increasing transmission costs, and affecting the entire bridge inspection system. This paper proposes a parallel compression method of LZSS data based on FPGA, which can compress the incoming data from the sensor system in parallel in the data acquisition system to improve the transmission efficiency, increase the throughput, and reduce the transmission cost.

1. Introduction
In China, many bridges built in the 1950s and 1960s are still in operation. Due to the impact of the environment and the aging of the bridges, various safety hazards in these “old bridges” that have been in operation for more than half a century are gradually exposed [1]. In order to ensure the normal operation of roads and railways in our country, it is very necessary to grasp the health status of bridges in time.

In recent years, with the rapid development of computer technology, sensor technology and communication technology, people have gradually designed and continuously improved the health monitoring system of bridge structure [2]. Figure 1 is a schematic diagram of the current mainstream health monitoring system of bridge structure. The system deploys various sensors that monitor different locations, types, and monitoring items of the bridge to the corresponding locations of the bridge, collects various real-time field data of the bridge through the data collector located at the bridge site, and then passes the collected data transmitted to the back-end data monitoring center through the public network of the data transmission subsystem. Finally, through the analysis of these data, the health, reliability, bearing capacity and durability of the bridge are evaluated. The current advanced detection system can also be based on these data to predict the situation that will happen, and promptly report potential safety hazards to prevent them before they happen.

However, due to the uninterrupted and long-term work of the bridge monitoring system of 7*24 hours, the long-term operation of a large number of sensors will generate massive amounts of data, because the data acquisition subsystem is usually far away from the monitoring center, and many bridges are located in complex terrain and cannot be reached by wired networks. Therefore, a large amount of data to be transmitted from the data collector at the bridge site to the data monitoring center
requires the use of a wireless network with limited speed, which will consume a lot of time, thereby reducing transmission efficiency, reducing throughput, and increasing transmission costs. The entire bridge inspection system will be effected negatively.

Figure 1. Schematic diagram of bridge structure health monitoring system

Through observing the data collected by the data collection center, it can be shown in Figure 2, which is a part of the temperature data of Sutong Bridge on January 12, 2018. It is not difficult to find that these data structures are monotonous and have a very high repetition rate. Thus, Compression necessitates most, while good and effective compression method can reduce or even eliminate the above-mentioned negative effects and greatly reduce the amount of data. However, due to the large amount of raw data, it takes time to compress these data. The time cost of data compression is included in the total transmission cost from the sensor system to the monitoring center, so the realization of data compression has also become very important.

Figure 2. Bridge temperature data
This paper proposes a parallel compression method of LZSS bridge data based on FPGA [3,4], and formulates a computing architecture using this method for parallel computing. By burning the LZSS algorithm [5] into the FPGA method, N pieces of data are compressed in N channels in parallel, which speeds up the compression, thereby improving the transmission efficiency of the entire system, increasing the throughput, and reducing the time cost of transmission.

2. METHODS

2.1. LZSS Algorithm

The LZSS algorithm is a dictionary-based compression algorithm, which is an enhancement and improvement to the LZ77 algorithm. The LZ77 algorithm maintains a forward buffer and a sliding window. For example, if the forward buffer contains character sequences (A, B, C), then phrase in the buffer has {(A),(A,B),(A,B,C)}, and if the sliding window contains the phrase (A,B,C), then the phrase in the dictionary has {(A), (A,B),(A,B,C),(B,C),(B),(C)}. The data in the forward buffer is compared with the phrases in the dictionary, and if there is a match, the offset B and the matching length L are output. The LZSS algorithm improves the phenomenon that LZ77 may output null pointers and characters that may match in the next iteration although they do not match in this iteration. The LZSS algorithm sets the minimum matching length M. If there are matches greater than or equal to M characters in the iteration process, then (B, L) is output, otherwise the first character of the forward buffer is output and moved forward by one character.

| STEP | POS | MATCH | OUTPUT |
|------|-----|-------|--------|
| 1    | 0   | --    | B      |
| 2    | 0   | --    | B      |
| 3    | 2   | BAA   | (2,3)  |
| 4    | 1   | AA    | (1,2)  |

Taking the second column of the bridge temperature data shown in Figure 2 as an example (this column is the most difficult column of the temperature data shown in the figure, because this column contains different data), the specific process of the LZSS algorithm can be expressed as Table 1, where A represents the data 0.587800, and B represents the data 0.551769. In the first step, a character string that matches the dictionary is not found in the forward buffer, so B is output, the sliding window becomes AAAB, and the forward buffer becomes BBAAAA. In the second step, because only B that matches the dictionary is found in the buffer, its length is less than the minimum matching length, so B is output directly, the sliding window becomes AABB, and the forward buffer becomes BAAAA. In the third step, the character string BAA that matches the dictionary is found in the buffer, so (2,3) is output, which means that two characters are reverted forward, three characters are copied, the sliding window becomes BBAA, and the forward buffer becomes AA. In the fourth step, the character string AA that matches the dictionary is found in the buffer, so (1,2) is output, which means that one
character is reverted forward, and two characters are copied. At this point, the buffer becomes empty, and the final compression result is AAAABB (2,3)(1,2).

2.2. The advantages of FPGA compression
Programmable gate array FPGA has been widely used in various fields where technology has spread due to its advantages such as flexible design, short cycle, low cost, rich logic resources, high parallelism, and strong reconstruction. Traditional bridge data acquisition subsystems mostly use CPU as computing power to compress sensor data. Its basic structure is shown in Figure 3 (Left). FPGA contains a large number of configurable logic units (CLB), and its basic structure is shown in Figure 3 (Right). It shows that because the CPU processor adopts the Von Neumann structure, the operation process is serial, and the parallelism is poor.

![Figure 3. Basic structure of CPU and FPGA](image)

The characteristics of the bridge data acquisition subsystem receiving data from the sensor subsystem are analyzed, as shown in Figure 4, which is a part of the temperature data generated by a certain bridge in a certain day, where WD stands for temperature, the number after WD denotes sensor number, and the following underlined number is referred to the time when the temperature data was generated. It can be seen that there will be a lot of data from the sensor subsystem at the same time. The computational efficiency of using CPU for compression will be reduced, and FPGA contains a large number of logic operation units, which can process a large amount of data at the same time, analyze the LZSS data compression process, and perform a large number of logic and operations to compare data whether it matches during dictionary matching. FPGA is adept at logic operations, therefore, the use of FPGA for data compression is very feasible and efficient. The parallel compression method of LZSS data based on FPGA can improve the computational efficiency of the data acquisition subsystem and reduce the bandwidth used for data transmission.
2.3. A parallel compression method of LZSS based on FPGA

Firstly, the computing architecture for the parallel compression method of FPGA-based LZSS bridge data is defined as shown in Figure 5. The above LZSS algorithm is designed using Vivado HLS, and the FPGA chip is burned to the FPGA chip according to the FPGA chip model. Secondly, it is assumed that N pieces of data are received from the bridge at the same time and input them to the off-chip DRAM of the FPGA, then the controller transmits the input data from the off-chip DRAM to the on-chip buffer of the FPGA. Finally, after the transmission is completed, the on-chip processing unit (Processing unit) is divided into N small on-chip processing units according to the number N of input data to realize N-way parallel execution.

The specific steps for using the parallel compression method are as follows:

1) HLSC/C++ realizes LZSS compression program.
2) Vivado HLS is used to burn the program in step 1 into the data acquisition subsystem and monitoring center.
3) At a certain moment, N pieces of sensor data passed in from the sensor subsystem are read into the off-chip DRAM of the FPGA.
4) The controller controls the on-chip cache to read the data in the DRAM.
5) At this time, DRAM can read N pieces of sensor data from the next moment.
6) N pieces of data in the cache are handed over to N on-chip processing units for compression processing.
7) The compressed data first passes through the on-chip buffer area, and finally is output to the off-chip DRAM by the controller.
8) Data is transmitted from the data acquisition subsystem to the monitoring center through the public network of the data transmission subsystem.
9) The data is read into the off-chip DRAM of the FPGA.
10) Repeat Steps 4) ~ 5)
11) N pieces of data in the cache are handed over to N on-chip processing units for decompression processing.
12) The decompressed data first passes through the on-chip buffer area, and finally is output to the off-chip DRAM by the controller.
13) These data are finally analyzed and processed by other modules of the monitoring center.

Among them, Step 1 burns the LZSS compression algorithm into the FPGA, and Steps 2-12 complete the data reading, data compression, and data writing operations in the data acquisition subsystem. Due to the parallelism of the FPGA, the compression process can be executed in parallel to shorten the compression time. Step 13 transmits the compressed data from the data acquisition subsystem to the monitoring center. Steps 14-24 complete the data read-in, data decompression, and

Figure 4. Partial temperature data of the bridge on a certain day
data write-out operations of the monitoring center. The same decompression process can be executed in parallel. Step 25 passes the data to other subsystems for processing.

The pseudo code for the above steps is as follows:

1. FPGA.burn(LZSS);
2. FOR all $ni \in N$
3. DRAM.read($ni$);
4. END FOR
5. FOR all $ni \in N$
6. Buffer.read($ni$);
7. LZSS.encode($ni$, $ni_{encode}$);
8. Buffer.write($ni_{encode}$);
9. END FOR

Figure 5. FPGA-based LZSS bridge data compression parallel computing architecture
FOR all ni_encode ∈ N_encode
DRAM.write(ni_encode);
END FOR

Do transform(N_encode) FORM Acquisition subsystem TO Monitoring center
FOR all ni_encode ∈ N_encode
DRAM.read(ni_encode);
END FOR
FOR all ni_encode ∈ N_encode
Buffer.read(ni_encode);
LZSS.dencode(ni_encode, ni);
Buffer.write(ni);
END FOR
FOR all ni ∈ N
DRAM.write(ni);
END FOR
Processing(N) By other subsystems;

3. Results
The DEV-C++ tool is used in the Windows system to implement the LZSS algorithm, and the data of the Sutong Bridge at 9 am every day from January 12 to January 31, 2018 is compressed and decompressed. The data is transmitted from the sensor system at the same time. There are 40 temperature data files, that is, 40 temperature sensors transmit temperature data files to the data acquisition subsystem at the same time. Compare the LZSS algorithm using the traditional system with the LZSS algorithm implemented using ten-way parallel FPGA (ie N=10). The time comparison diagram of compression at the data acquisition system side is shown in Figure 6, and the abscissa axis represents the date. The vertical axis represents the time of compression at nine o'clock in the morning of the day. It can be seen that the compression time of FPGA is about 8 times shorter than that of traditional CPU under the same number and amount of data files. Similarly, the time comparison chart for decompression in the monitoring center is shown in Figure 6.

Based on the specific analysis of the data at 9:00 am on January 12, 2018, Figure 7 shows the relationship between the number of input files and the time when the traditional system is used for compression and decompression and the system embedded with FPGA. It can be seen that the compression using FPGA The time is about 8 times shorter than that of a traditional CPU.
Figure 7. The relationship between the number of input files and the time of compression and decompression on January 12

4. Conclusion
To deal with the problems of low transmission efficiency, low throughput and high transmission cost of the large amount of data to be transmitted from the sensor subsystem of the bridge health monitoring system to the data monitoring center, this paper proposes a parallel compression method LZSS bridge data based on FPGA, formulates a computing architecture using this method for parallel computing, and implements a compression parallel method based on this architecture, which solves the problems of long data compression and low transmission efficiency caused by high time costs, thereby improving bridge health monitoring. Thus, the system transmission efficiency is increased, the throughput is boosted, and the transmission cost is saved.

Acknowledgments
This paper is supported by Henan Provincial Department of Transportation Science and Technology Planning Project (2019J-2-5).

References
[1] Booth, N. and Smith, A. S., [Infrared Detectors], Goodwin House Publishers, New York & Boston, 241-248 (1997).
[2] Davis, A. R., Bush, C., Harvey, J. C. and Foley, M. F., "Fresnel lenses in rear projection displays," SID Int. Symp. Digest Tech. Papers 32(1), 934-937 (2001).
[3] Van Derlofske, J. F., "Computer modeling of LED light pipe systems for uniform display illumination," Proc. SPIE 4445, 119-129 (2001).
[4] Myhrvold, N., “Confessions of a cybershaman,” Slate, 12 June 1997, <http://www.slate.com/CriticalMass/97-06-12/CriticalMass.asp> (19 October 1997). www.optics4yurresearch.com/7752.html
[5] Jones, C. J., Director, Miscellaneous Optics Corporation, interview, Sept. 23 2011
[6] FamilyName, GivenName Initial., "Title," Source, pg# (year).