

工业设备远程监控中的数据压缩研究

史桂红

(苏州健雄职业技术学院 软件与服务外包学院,江苏 太仓 215411)

摘要:针对大型工业翻车机设备远程监控系统中实时数据传输效率的问题,提出了一种对字符数据和图像采用改进的 LZW 算法压缩、对时间属性采用时间预测压缩算法进行压缩的方法。其中,改进的 LZW 算法主要是对构造的字符编码表的长度进行动态分配,并建立父亲指针索引和 HASH 索引;而时间预测编码压缩方法主要是对时间属性数值进行了压缩。最后采用检验数据压缩性能的指标,通过实例验证改进的 LZW 算法的数据压缩效果。实验结果表明,改进的 LZW 算法较 WINRAR 压缩方法具有较好的数据压缩效果。

关键词:远程监控;数据压缩;时间预测编码压缩

中图分类号:TP311.5 **文献标识码:**A **文章编号:**1671-5322(2018)04-0022-08

随着经济的快速发展,基于网络通信的工业设备的远程监控越来越多^[1]。长时间的远程监控必然产生大量的数据,从而产生数据的采集、保存与传输的问题。随着 CAN(Controller Area Network)总线技术的发展,基于 CAN 总线的采集监测数据在工程机械领域的应用越来越多;同时数据压缩技术也已广泛应用于数字通信、广播、存储和多媒体等领域^[2-6],对数据压缩算法的研究也越来越多。

张凤林等^[7]提出了改进的 LZW(Lempel-Ziv-Welch Encoding)算法,对字符字典建立 HASH 索引和父亲指针索引使得搜索字符串字典匹配效率较快,但解决地址冲突采用增量为 1 的开放定址法使得装填因子 α 较小、空间效率较差;胡平等^[8]提出改进的 HASH 索引的 LZW 算法,验证了解决地址冲突方法在空间效率和压缩比率上优于前者,但没有建立父亲指针索引使得在搜索字符串字典匹配效率上较前者略微逊色。

本文针对火电厂大型翻车机设备生产厂家提高实时远程监控系统中数据(如运行参数、设备

信息与视频图像信息)传输效率的问题,对字符数据和图像采用改进的 LZW 算法进行压缩,对时间属性采用时间预测压缩算法^[9]进行压缩。

1 数据压缩

在实际应用中,设备的监控通常采用固定周期的,随着时间的推移形成一个时间序列的数据。设采集的翻车机设备远程监控数据为 ASCII 格式的字符流,形成时间序列的数据流为 $Y = [x(1), x(2), x(3), \cdots, x(N)]$ 。式中 $x(i)$ 表示当前时刻监测的数据,包括设备的基本信息如设备编号、压力、温度等。

由于设备信息完整性的要求,需采用无损压缩的方式进行压缩。无损数据压缩通常根据建模方法的不同而有两种不同的类型,分别是统计模型(哈夫曼编码,算术编码等)和基于字典的模型(LZ78, LZW 等),它们的区别如表 1 所示。

考虑到翻车机设备监测的数据量大、实时性要求高,需要较高的数据压缩率,同时考虑数据的相关性等因素,采用改进的 LZW 算法对监测数据进行压缩。

收稿日期:2018-05-20

基金项目:苏州健雄职业技术学院“三级联动”自然科学类科研基金项目(2017SJLD21);苏州健雄职业技术学院教改立项研究课题(JG201516)

作者简介:史桂红(1971—),女,内蒙古赤峰人,讲师,硕士,主要研究方向为物联网及移动互联应用技术。

表 1 常见无损压缩比较表
Table 1 Comparative statement of common lossless compression

压缩算法	所需内存	复杂度	实时性	压缩率	应用场合
行程编码	较小	较低	较高	较高	二值图像
哈夫曼编码	较大	中等	较差	很高	任何数据
算术编码	较小	较高	不高	较高	任何数据
LW77 算法	几 K 字节	中等	较高	较高	任何数据

1.1 LZW 算法简介及改进

LZW 压缩算法基本原理详见文献[10-11]。采用 LZW 压缩算法时,在压缩过程中,LZW 压缩算法会自适应建立一个字典,这个生成的字典不需要随着压缩的数据一同传输,而是根据压缩的数据在解压时重新动态生成一模一样的字典^[7]。后面的数据如果与字典中的数据相匹配,则输出字典的索引。由于输出字典的索引所用的比特数远小于字符的比特数,从而达到数据压缩的效果。但是这种 LZW 压缩算法也存在如下不足之处:

- (1)编码存储长度固定。LZW 压缩算法在压缩过程中会建立一个编码表,将输入字符串映射成定长的码字输出,通常码长设置为 12 bit。在编码过程中,编码存储空间长度是固定不变的,有时会造成存储空间的浪费。
- (2)搜索效率不高。LZW 算法在编译表中搜索时如果能查找到当前的字符串如 CH,会把后续的字符直接连接在当前字符 CH 的后面;若下一个待处理的字符如字符 F 在编译表中搜索失败,则会把当前字符和字符 F 存储在编译表中。显然,为了添加一个个字符,需要耗费较多的时间和空间去搜索和存储,搜索效率较低。

针对 LZW 算法存在的问题,提出以下改进:

- (1)动态分配策略。给不同代码字分配不同长度的空间,如编码中第 257 个码字设置为变动开始的标志,设定一个变化标志 FLAG 数组(初始化为 0 且第一个 257 为变动的标志)。若编码的长度为 8 字节,则分配 $2^8=256$ 个码字,若为 9~12,依此类推,每当长度变化时相对应的标志数组的内容变化为 1。

- (2)建立父亲指针、当前节点指针和 HASH 索引。传统 LZW 算法在编码过程中,将新出现的字符加入字典中已知字符串之后,并将新的字符

串全部更新到字典中,导致在只增加一个新字符时却需复制增加后的整个字符串,造成空间和处理时间上的浪费。本文提出建立父亲指针、当前节点指针和 HASH 索引,构造基于指针链的节点对象数组(该节点对象包含 3 个元素),用于链接父亲字符串的地址指针元素、当前节点数值元素和节点的 HSAH 地址属性,从而在构造编译表的时候,利用 HSAH 函数生成一个地址,并检查地址是否冲突。

HASH 函数如下所示:

$$H(P,C_H) = (C_H \ll S_H) \wedge P \tag{1}$$

式中: P 为字符字典的前缀字符串; C_H 为当前输入的字符; S_H 是字符需要向左移动的位数,它保证了 C_H 和前缀字符串对齐;符号 $\wedge$ 代表异或运算。

处理 HASH 冲突的函数为

$$H_{n+1} = (n + 1) * H_n - (n - m) * T \tag{2}$$

式中: n 为冲突的次数; m 为 H 产生负数时的调整因子, $m=1,2\cdots$,并且当 H 为负数的时候, m 就加 1; T 为 HASH 地址表的长度。

在构建字符字典过程中,如果输入的字符串不在字典中,此时若没地址冲突则直接将该字符串添加到字典中,并将父亲指针链接到对应前缀字符串的地址,把当前字符数值保存到生成的 HASH 地址的对象数组中;否则用解决地址冲突的函数生成地址,然后保存节点。

如输入字符串在字典中,且它的前缀字符和当前节点值都与字典中的节点值相同,说明该字符串已存在,继续读入下一字符。

下面以一个小例子进行说明。对于字符序列 efefef,首先对对象数组进行初始化,设字符 efefef 前缀字符串指针为 Nodefather,数值为 Nodevalue, HASH 地址为 Nodeindex,字典初始化有 2 个字符 ef,Nodeindex 全部赋值为 -1,如表 2 所示。

表 2 字典初始化表
Table 2 Table of the initialization of dictionaries

节点属性	节点属性值							
Nodefather								
Nodevalue	e	f						
Nodeindex	0	1	-1	-1	-1	-1	-1	-1

输入第一个字符 e,无需搜索字典;输入第二个字符 f,查找前缀 ef 是否在字典中,并计算 HASH 地址。假设计算结果为 3 (即第四个元

素),节点类对象 Node 的属性 Node. Nodeindex 为 -1,则 Node. Nodefather = 0,Node. Nodeindex = 3,Node. Nodevalue = f,如表 3 所示。

同样方式输入第三个字符 e,没产生冲突,故结果如表 4 所示。

表 3 输入第 2 个字符结果表

Table 3 Result table of entering the second character

节点属性	节点属性值						
Nodefather	0						
Nodevalue	e	f	f				
Nodeindex	0	1	-1	3	-1	-1	-1

表 4 输入第 3 个字符结果表

Table 4 Result table of entering the third character

节点属性	节点属性值						
Nodefather	0						1
Nodevalue	e	f	f				e
Nodeindex	0	1	-1	3	-1	-1	7

接着输入第四个字符 f,由于字符 ef 在字典中已经存在,故 Node. Nodeindex = 3,Node. Nodevalue = f。当输入第五个字符 e 的时候,HASH 地址为 7,产生地址冲突,此时用式(2)处理冲突的函数生成地址为 6,结果如表 5 所示。

表 5 输入第 5 个字符结果表

Table 5 Result table of entering the fifth character

节点属性	节点属性值						
Nodefather	0		3		1		
Nodevalue	e	f	f		e	e	
Nodeindex	0	1	-1	3	-1	-1	6

1.2 时间属性压缩算法

时间属性压缩算法是指对于时间标签序列 $[t_1, t_2, t_3, \dots, t_i, \dots, t_n]$ ($i \in [1, n], n$ 为正整数),选择一个基准点时间,利用基准点时间标签、时间采样周期以及预测误差进行时间标签预测的方法。例如,对于时间标签 t_{i-1}, t_{i-2} ($i \in [1, n]$) 来说, t_i 可以通过 t_{i-1} 和 t_{i-2} 来预测。预测方法如下:

$$t_i = t_{i-1} + T + e_i \Rightarrow e_i = t_i - t_{i-1} - T$$
$$\hat{t}_{i+1} = t_i + T + e_i = t_i + t_i - t_{i-1} = 2t_i - t_{i-1}$$

(3)

式中: $\hat{t}_{i+1}$ 为 t_{i+1} 的预测值, T 为时间周期, e_i

为 i 时刻的实际值 t_i 与预测值 $\hat{t}_i$ 之间的实际误差。

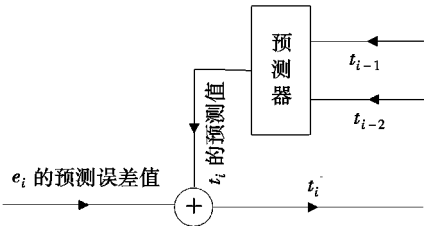
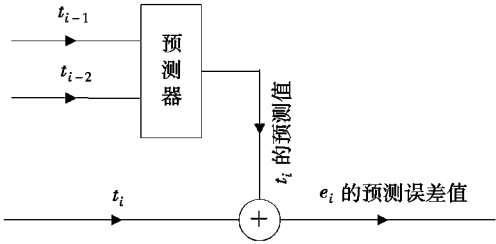
若上一时刻的实际误差为 e_i ,由于两个相邻时间标签存在的误差相等,因此可以用上一周期的误差来预测下一周期的时间标签,即 $i+1$ 时刻的值可由之前的 2 个已知时刻 $i, i-1$ 的值进行线性预测。

同理,当 $i \geq 3$ 时, i 时刻的预测值 $\hat{t}_i$ 可由 $i-1, i-2$ 时刻值进行预测(公式(3))。当 $i=0$ 或 1 时,无法预测,因此需要初始化。

综上所述, i 时刻预测值 $\hat{t}_i$ 计算公式如下:

$$\hat{t}_i = \begin{cases} 2t_{i-1} - t_{i-2}, & i \geq 3 \\ T, & i = 2 \\ 0, & i = 1 \end{cases} \quad (4)$$

由此可得出 $\hat{e} = |t_i - \hat{t}_i|$, $\hat{e}$ 是预测误差。从而解码的数值 $\hat{t} = |t_i| + \hat{e}$,其编码原理如图 1 所示。与图 1 对应的解码如图 2 所示。



本预测编码算法编程实现时,只需保持 2 个历史时刻,需要很少的数据缓冲区,且编解码运算简单,对硬件要求低,压缩效率高。

2 改进的 LZW 算法

改进的 LZW 算法流程如图 3 所示。由图 3 可知,该流程包含如下步骤:

(1)初始化,包含初始化字典,以及公式(1)~(3)的参数,并令当前字符串前缀 Prefix = NULL。

(2)读入字符流到 Char,并判断是否结束。如结束,则将代表当前前缀 Prefix 的码字输出到码字流,同时输出结束标识符;否则,转步骤 3。

(3)判断当前前缀 + 字符串,即 Prefix + Char 是否在字典中。

如是,则令 Prefix = Prefix + Char,并输出,同时将字典编码加 1;

否则,

a. 利用公式(1)的 HASH 函数生成地址。

b. 检查地址是否冲突。

若地址不冲突,直接利用 a 里面生成的地址,转 c;若地址冲突,利用公式(2)的解决地址冲突的函数生成地址,转 c。

c. 把代表当前前缀 Prefix 的码字输出到码字流,将字典编码 + 1,并检查 FLAG 标志数组,进行长度动态分配。

d. 把 Prefix + Char 添加到词典。

e. 令 Prefix = Char。

(4)返回步骤(2),继续对新的字符进行编码。

其流程图如图 3 所示。

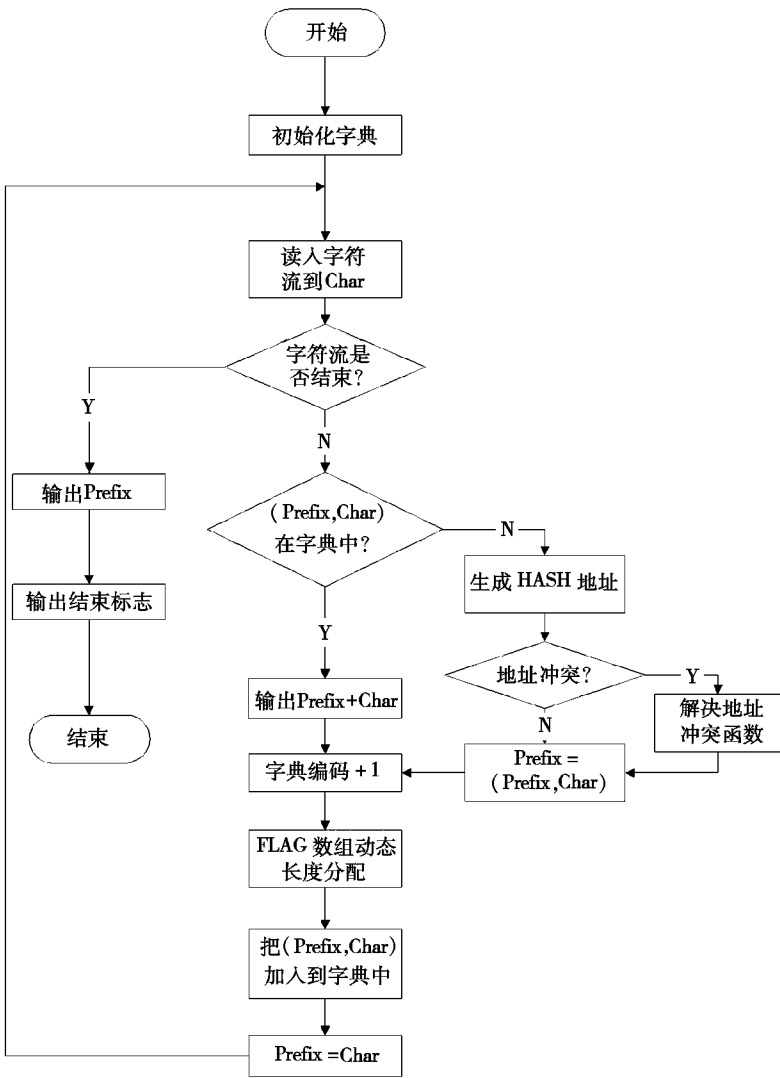


图 3 改进的 LZW 算法流程图

Fig. 3 Flow chart of improved LZW algorithm

3 实验数据分析

在数据压缩中,常用评价数据压缩好坏的指标之一为数据压缩比率,其基本含义是数据压缩之后的容量大小和压缩前容量大小的比值,计算公式如下所示。

$$R = \frac{R_{out}}{R_{in}} \tag{5}$$

式中: R_{out} 为压缩之后的文件大小,KB; R_{in} 为压缩之前的文件大小,KB。

为了检验改进的 LZW 算法在字符图像压缩方面的压缩效果,现通过实验将其与压缩软件

WINRAR 的压缩效果进行对比。实验通过 CAN 总线获取翻车机设备基本数据信息和监控图像,利用改进的 LZW 算法对设备的基本信息如设备编号、设备监控图像等进行压缩。其中设备整形的数据在压缩前要进行数据类型的转换,将其转换成字符串后再进行压缩;而时间属性则采用实际预测编码压缩方法进行压缩。

现以某一时间基准点为起始点,检测时间周期为 1 s,以某次连续监测的 60 个数据包为例分析两种方法的压缩效果。两种压缩方法压缩图像、压缩字符得到的压缩比率值的分布分别如图 4、图 5 所示。

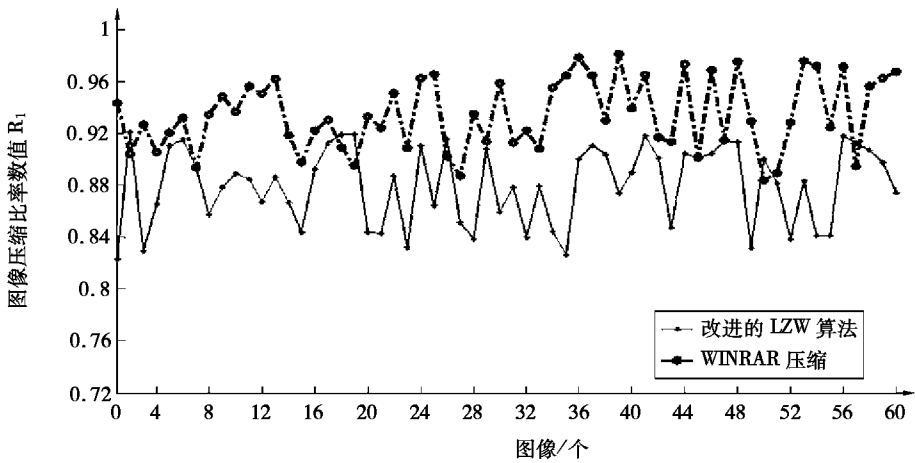


图 4 图像压缩比率值对比图
Fig. 4 Comparison chart of image compression ratio value

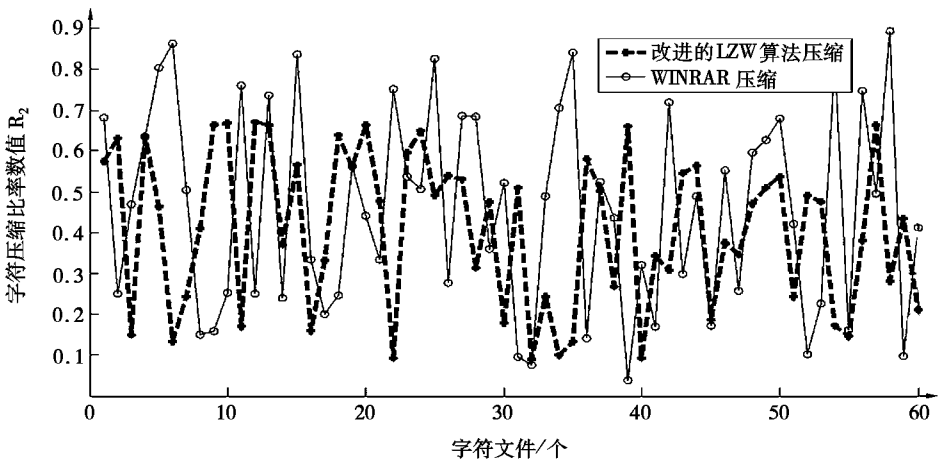


图 5 字符压缩比率值对比图
Fig. 5 Comparison chart of character compression ratio value

由图 4 可以看出,利用改进的 LZW 算法进行图像压缩得到 60 个图像的压缩比率值的分布基本上在 WINRAR 软件压缩的图像下方。利用改进的 LZW 算法压缩得到图像压缩比率值的变化

范围为[0.823 1,0.921 2],平均值为 0.880 2;利用 WINRAR 软件进行图像压缩得到压缩比率值变化范围为[0.883 6,0.981 2],平均值为 0.934 5。显然,利用改进的 LZW 算法进行图像压缩比用

WINRAR 软件进行压缩的效果更好。

图 5 中利用改进的 LZW 模型得到字符压缩比率值的变化范围为 $[0.071\ 95,0.690\ 4]$,均值为 $0.414\ 1$;采用 WINRAR 压缩软件的压缩比率值变化范围 $[0.028\ 31,0.898\ 8]$,均值为 $0.458\ 1$ 。从图 5 中可以看出利用改进的 LZW 压缩算法比 WINRAR 软件的整体压缩效果更好,但在个别点上比 WINRAR 压缩软件的压缩比率高。

时间属性值压缩采用时间预测编码压缩,其压缩比率值分布如图 6 所示。

图 6 中时间属性压缩比率变化范围为 $[0,0.189\ 2]$,均值为 $0.066\ 36$ 。从图 6 中可以看出,时间属性值的变化具有一定的周期性,相邻两个时间周期之间的变化很小。显然,时间预测方法

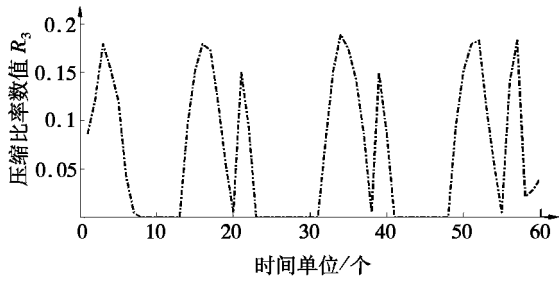


图 6 时间属性值比率值图

Fig.6 Chart of time attribute value ratio value

较好。

以上是 60 个数据包的实验仿真,下面以连续 200 个数据进行压缩率的对比试验,实验结果如图 7 ~ 图 9 所示。

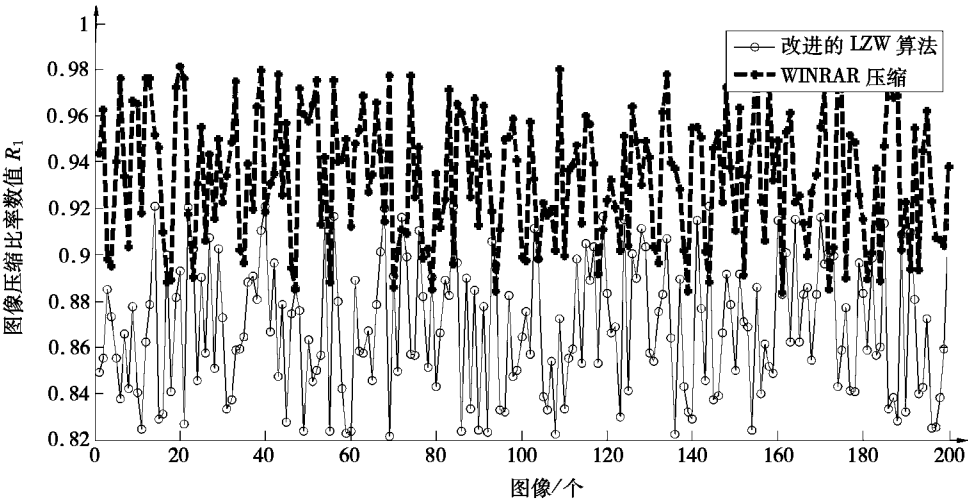


图 7 图像压缩比率值对比图

Fig.7 Comparison chart of image compression ratio value

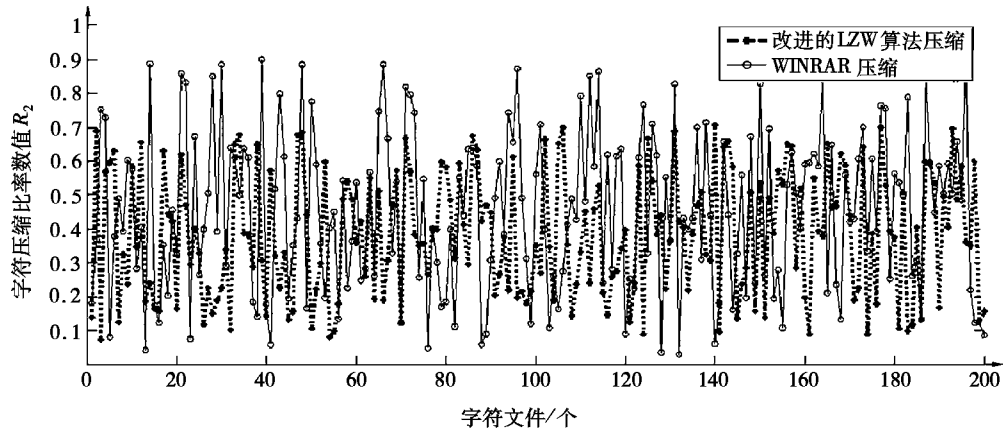


图 8 字符压缩比率值对比图

Fig.8 Comparison chart of character compression ratio value

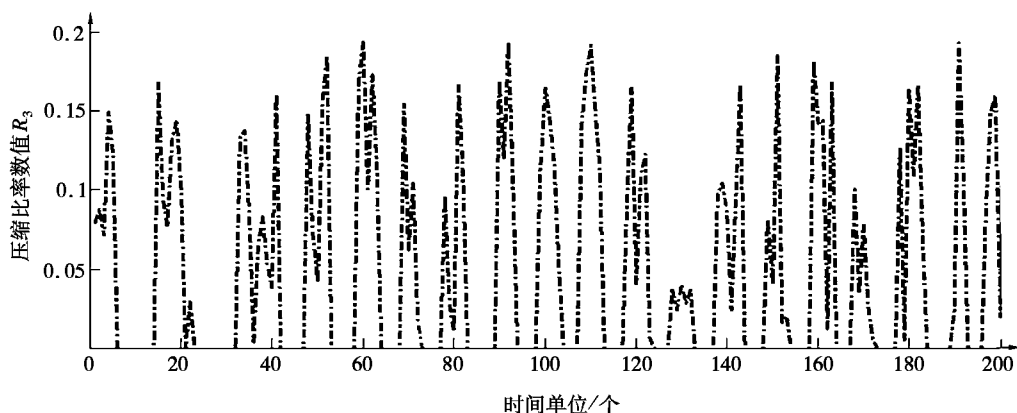


图 9 时间属性值比率值图

Fig.9 Chart of time attribute value ratio value

由图 7 可以看出,利用改进的 LZW 算法进行图像压缩得到 200 个图像的压缩比率值的分布基本上在 WINRAR 软件压缩的图像下方。利用改进的 LZW 算法压缩得到图像压缩比率值的范围为 $[0.821\ 7, 0.921\ 1]$, 均值为 0.868 9; 而 WINRAR 软件压缩相应的范围为 $[0.883\ 8, 0.983\ 3]$, 均值为 0.933 3。图 8 中改进的 LZW 算法压缩字符得到字符压缩率的范围为 $[0.073\ 6, 0.706\ 1]$, 均值为 0.382 8; 而 WINRAR 压缩字符得到字符压缩率的范围为 $[0.030\ 5, 0.900\ 4]$, 均值为 0.454 4。图 9 中时间属性压缩比率值的变化范围为 $[0, 0.193\ 4]$, 均值为 0.049 8。从图 9 可以看出, 时间属性值的变化具有一定的周期性, 且相邻两个时间周期之

间的变化很小, 说明时间预测方法较好。

以上分析表明改进的 LZW 模型较 WINRAR 软件的压缩效果更好。

4 结束语

本文针对字符串和图像的压缩采用改进的 LZW 算法, 最后用实例验证其压缩效果较 WINRAR 软件的压缩效果更好。后续的研究可继续深入, 如图像压缩可采用 XIE^[12-13] 等提出的改进 LZW 图像和神经网络结合的图片压缩方法, 也可用小波技术^[14-15] 对随时间而形成的序列信号进行分析。

参考文献:

- [1] 詹宏宏, 贾民平, 胡建中, 等. 机群状态远程监测与故障诊断系统的研制[J]. 东南大学学报(自然科学版), 2005, 35(5): 706-709.
- [2] LIANG Y, LI Y M. An Efficient and Robust Data Compression Algorithm in Wireless Sensor Networks[J]. IEEE Communications Letters, 2014, 18(3): 439-442.
- [3] ABDELAAL M, THEEL O. An efficient and adaptive data compression technique for energy conservation in wireless sensor networks[C]//2013 IEEE Conference on Wireless Sensors? (ICWISE), Dec. 2-4, 2013, Kuching, Malaysia.
- [4] 马发勇, 厉启鹏, 马志斌, 等. 电力调度 SCADA 系统中历史数据压缩及存储策略[J]. 电网技术, 2014, 38(4): 1109-1114.
- [5] MAGHANIAN H, ANSALONI G, ATIENZA D, et. al. Power-efficient joint compressed sensing of multi-lead ECG signals [C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 4-9, 2014, Florence, Italy.
- [6] 李文航, 汪国平, 陈毅松. 基于移动网络的远程视频监控系統[J]. 计算机应用研究, 2009, 26(10): 3798-3800.
- [7] 张凤林, 刘思峰. LZW ~*: 一个改进的 LZW 数据压缩算法[J]. 小型微型计算机系统, 2006, 27(10): 1897-1899.
- [8] 胡平, 张金钟. 远程故障诊断终端的数据压缩技术研究与实现[J]. 计算机工程与应用, 2012, 48(34): 130-135.
- [9] 何创新, 刘成良, 李彦明. 工程机械远程监控动态实时数据压缩[J]. 上海交通大学学报, 2009, 43(6): 914-917.

[10] 陈庆辉,陈小松,韩德良. 中文文本压缩的 LZW 算法[J]. 计算机工程与应用,2014,50(3):112-116.

[11] 邹学玉,冯振,张少华,等. 基于 LZW 算法的声波测井数据压缩研究[J]. 测井技术,2013,37(3):294-296.

[12] XIE Y H, TANG X A, SUN M Y. Image compression based on classification row by row and LZW encoding[C]//2008 Congress on Image and Signal Processing, May 27-30, 2008, Sanya, Hainan.

[13] LIN J X, LIN J H. Accelerating BP neural network-based image compression by CPU and GPU cooperation[C]//2010 International Conference on Multimedia Technology, Oct. 29 - 31, 2010, Ningbo.

[14] 张忠林,许凡. 基于小波变换的动态关联规则元规则 GM(1,1)挖掘[J]. 计算机科学,2013,40(5):209-202.

[15] 张忠林,许凡. 基于小波变换的动态关联规则元规则挖掘方法[J]. 计算机应用,2012,32(7):1983-1986.

Research on Data Compression in Remote Monitoring and Control of Industrial Equipment

SHI Guihong

(College of Software and Service Outsourcing , Suzhou Chien - shiung Institute of Technology, Taicang Jiangsu 215411, China)

Abstract: Aiming at the problem of real-time data transmission efficiency in remote monitoring and control system of large industrial dumper equipment, a method of compressing LZW algorithm for character data and image and compressing time attribute with time prediction compression algorithm is proposed. Among them, the improved LZW algorithm is mainly to dynamically allocate the length of the constructed character encoding table, and establish the father pointer index and HASH index. The time prediction coding compression method is mainly to compress the time attribute values. Finally, the data compression effect of the improved LZW algorithm is verified by an example using the indicators for checking the data compression performance.

Keywords: remote monitoring and control; data compression; time prediction coding compress

(责任编辑:李华云)

启 事

本刊已入编《中国学术期刊(光盘版)》、“中国期刊网”“万方数据——数字化期刊群”、《中国科技期刊数据库》、《美国化学文摘》,作者著作权使用费在本刊稿酬中一并给付(另有约定者除外)。对此不同意者,请在来稿时说明。