

基于方位信息的改进 LZW 前缀编码方案

韩 宾 张红红 江 虹 丁 一

(西南科技大学信息工程学院 四川 绵阳 621010)

摘 要 LZW 压缩算法在实时采集与无线传输中具有重要应用价值,一般采用采集-压缩-传输的工作模式,该模式下的较高压缩比可极大降低对无线传输的压力。但在采集速度较快、数据传输带宽较低、硬件资源受限的情况下,在对采样点概率分布较均匀的数字信号进行压缩时,易出现压缩率不高或采集速度与压缩速度不匹配的问题。对此,文中提出了基于方位信息的改进 LZW 前缀编码方案。该改进压缩算法基于压缩比因子对采样点进行映射操作,使其能够标识后面相邻采样点的压缩情况,然后通过采样点间的方位信息,缩短采样点的码长,实现对采样点数据的压缩。实验表明,与原 LZW 压缩算法相比,该改进算法在不增加算法复杂度和硬件存储空间的前提下,压缩比可提高 26.25%,证明了该算法在采集系统中的有效性。

关键词 LZW 算法,方位信息,数字信号,映射,编码

中图分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.08.026

Improved LZW Prefix Coding Scheme Based on Azimuth Information

HAN Bin ZHANG Hong-hong JIANG Hong DING Yi

(College of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China)

Abstract LZW compression algorithm has important application value in real-time acquisition and wireless transmission. Generally, it adopts the acquisition-compression-transmission working mode, and high compression ratio in this mode can greatly reduce the pressure on wireless transmission. But in the case of fast acquisition speed, low data transmission bandwidth and limited hardware resources, it can easily lead to problems that the compression rate is not high or the speed of acquisition is mismatched when compressing the digital signal which has a sampling point of uniform probability distribution. To this end, this paper proposed an improved LZW prefix encoding scheme based on azimuth information. Firstly, the improved compression algorithm maps the sampling points based on compression ratio factor, so that it can identify the compression condition of adjacent sampling points. Secondly, through the azimuth information between the sampling points, the code length of the sampling points is shortened, so the data of the sampling points is compressed. Experiments show that, compared with the original LZW compression algorithm, the improved algorithm can increase the compression ratio by 26.25% without increasing the complexity and hardware storage space. Therefore, the effectiveness of the algorithm in the acquisition system was proved.

Keywords LZW algorithm, Orientation information, Digital signal, Mapping, Coding

1 概述

目前,数据采集与传输技术在广播、导航、雷达、遥测、计算机以及其他科学技术领域都发挥着重要作用。在某些特殊环境下,系统对传输速率的要求较高,为了满足数据无线传输速率的要求,国内外各大公司通过采用各种数字调制技术,进一步提高数据的无线传输速率。在采集数据的无线传输过程中,保证数据在特殊环境下及时传输的关键是传输速率的提高,但是网络瓶颈问题是数据传输系统的最大缺陷。当前数字调制技术虽然可有效提高大数据的传输速率,但在某些场

景下仍不能满足当前系统对数据传输的需求。例如:1)在高速旋转机械实验中,受测试环境的影响,硬件能力受到严格限制,这种限制在采集频率较高的实验中,增加了硬件及时处理采集数据的难度;2)在高速铁路基设备采集实验中,现场采集设备受测试环境影响,对能耗、体积都有非常严格的要求。在以上特殊环境下,不可能用复杂的调制技术来完成无线传输。此时,提高数据传输效率的一种有效途径就是应用数据无损压缩技术^[1-2]。

早期的无损压缩算法主要是基于统计的 Huffman 压缩算法,但该算法在对数据进行压缩时,需要对原始数据进行两

收稿日期:2018-08-24 返修日期:2018-12-07 本文受国家自然科学基金项目(51475453),西南科技大学研究生创新基金(18ycx118)资助。
韩 宾(1974-),男,博士生,副研究员,硕士生导师,主要研究方向为认知无线电、信号处理等,E-mail:hanbin@swust.edu.cn(通信作者);
张红红(1991-),女,硕士生,主要研究方向为通信信号处理与特殊环境下大数据传输;江 虹(1969-),男,博士,教授,博士生导师,主要研究方向为认知无线电、信号处理等;丁 一(1990-),男,硕士生,主要研究方向为网络传输控制。

次扫描,占用大量的内存空间,且对于数据概率分布较均匀的情况,压缩效果不理想^[3-4]。Ziv^[5-6]等提出了基于字典的LZ77和LZ78压缩算法,使得数据的压缩性能得到了明显提升。Terry等以字典集合压缩思想为基础,提出了LZW压缩算法,该算法在压缩时间及压缩性能上具有明显优势。随着LZW压缩算法的广泛应用,对该算法的改进也成为一项重要的研究内容。虽然各种改进型的LZW压缩算法在特定情况下提高了压缩比,但对于采样频率较高的数字信号压缩效果不理想。

本文通过分析LZW压缩算法的原理,针对 $(2^{(n/2)-k}+1) < R$ 的数字信号(数据编码长度(简称码长)为 n bit, R 为信号最小值到最大值之间采集点的个数, k 为常数, $k > (n/2) - 4$),提出了一种基于方位信息的改进LZW前缀编码方案,并通过数字信号的采样点进行编码,验证了其压缩的有效性。

2 LZW算法的原理及研究现状

LZW算法的基本思想是提取原始文本文件数据中的不同字符,基于这些字符创建一个编译表,然后用编译表中的字符索引替代原始文本文件中的相应字符,以减少原始数据^[7-8]。

由以上分析知,LZW算法的实现原理是用 l bit字典索引来编码一个或多个字符; l bit编码的字符越多则压缩效果越好,但该编码在每次索引时效率较低。因此,学术界基于该思想提出了许多改进方法^[9-19]。

文献^[9-10]针对图像源具有许多冗余的问题,提出利用冗余减少数据编码总长度的方法,此方法对于存在大量冗余的图像压缩有一定效果,但是对于冗余量较少的数字信号,压缩效果不理想。文献^[11-12]只是将两种算法简单地结合,压缩效果有一定提高,但增加了算法的复杂度。文献^[13]提出的算法从去除根字符所占用的空间角度考虑,提高了压缩比,但降低了压缩率。文献^[14-15]提出的算法分别针对符合路段空间性、布尔矩阵特性的数据进行压缩,其针对码字0和1出现的概率相差较大的情况,压缩效果较好,但统计出0和1出现的概率的过程较费时。文献^[16]的算法采用映射方法来改变部分数据的编码长度,提高了压缩比,但是达到映射条件的数据太少,压缩效果不明显。文献^[17]通过降低压缩率,获得较高的匹配加速度,其整体的压缩效率并没有提高。文献^[18-19]的算法分别以函数作用范围为基础切分代码段,增加数据集的角度提高压缩速度的做法有些武断,容易造成内存利用不合理的现象。

上述改进算法均不同程度地提高了压缩比,但都有不足之处,或增加了算法的复杂度,或增加了存储空间,或压缩比提高的效果不明显。在硬件体积受限、压缩算法复杂度受到严格限制、处理速度要求较高、能耗要求较高的特殊环境下,这些改进型算法的压缩效果不理想。

3 改进编码方案及分析

针对以上改进算法在特殊环境下存在的各种不足,本文针对周期信号提出一种基于矩阵方位特性来优化LZW前缀编码的改进型压缩算法。该算法利用函数映射特性与矩阵方位特性来改进LZW前缀编码方式,从而缩短了编码长度,提

高了压缩效率。本文主要解决由于硬件采集数据速度较快、受带宽的限制使得数据传输速度较慢、硬件体积受限、内存较小等导致硬件内存溢出的问题,因此只针对压缩算法进行讨论。

3.1 影响压缩比因子

设LZW压缩数据的大小为 D bit,数据的平均压缩速度为 V ,数据压缩时间为 T 。将数据分为 k 块,使用改进LZW算法进行压缩,总的压缩时间为 t ,总的压缩速度为 V_L 。每块数据大小为 D_i ,每块数据的压缩速度为 V_i ,数据 D_i 与数据 D 的比值为 P_i ,块压缩率为 σ_i ,压缩后的数据大小为:

$$\sum_{i=1}^k D_i \cdot \sigma_i = D \cdot \sum_{i=1}^k P_i \cdot \sigma_i \quad (1)$$

由式(1)可知,较低的块压缩率 σ_i 对应的块数据,在 D bit数据中所占比例 P_i 越大越好。

改进LZW算法的压缩时间与LZW算法的压缩时间的比值为 η ,计算如下:

$$\eta = \frac{t}{T} = \frac{D \cdot \sum_{i=1}^k P_i \cdot \sigma_i}{\frac{\sum_{i=1}^k P_i \cdot \sigma_i}{V_L}} = \frac{\sum_{i=1}^k P_i \cdot \sigma_i}{\frac{D}{V}} = \frac{\sum_{i=1}^k P_i \cdot \sigma_i}{\sum_{i=1}^k P_i \cdot \frac{V_i}{V}} = \sum_{i=1}^k P_i \cdot \frac{\sigma_i}{(\frac{V_i}{V})} \quad (2)$$

由式(2)可知,相对于LZW算法,若要减少改进LZW压缩算法的压缩时间,即 $\eta < 1$,则 $\sum_{i=1}^k P_i \cdot \frac{\sigma_i}{(\frac{V_i}{V})} < 1$,即 $\frac{\sigma_i}{(\frac{V_i}{V})} < 1$ 。

故在提高压缩比的同时,也要提高压缩速度,并且 $\sigma_i < (\frac{V_i}{V})$ 。

由此可得,在改进LZW压缩算法时,不能增加算法的复杂度,因此本文通过方位信息来缩短前缀的码长,进一步缩短字典集合的长度,减少压缩时间。

3.2 字典长度处理

LZW算法对于前缀编码一般采用变长编码方式,但是在实际工程中数字信号应用较多的编码长度(用 n bit表示)是12 bit,16 bit,18 bit,24 bit,其中码长与字典长度相等的情况较多。当数字信号字典长度正好是2的整数次幂即字典长度与码长相等时,LZW及其改进型压缩算法的压缩效果不理想。因此,本文采用函数相关特性改变数字信号的编码范围,对此类数字信号进行压缩。数字信号主要分为两类:1)信号中包含正负两种符号类型;2)信号中只包含一种符号类型。

针对第一类数字信号,采用抛物线的相关性质来缩短字典长度。抛物线图形如图1所示,其公式为 $y^2 = 2ax, x \geq 0$ 且 $a \geq 0$ 。

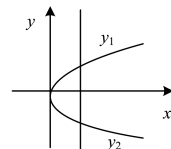


图1 抛物线图形

Fig. 1 Parabolic graph

如图1所示,该函数的每个 x 值(0点除外)在其抛物线上,都有两个对应点,即 y_1 和 y_2 。这两个点到 x 轴的垂直距

离相等。将该性质应用到数据压缩中,即将数字信号中的负数通过该性质转化为正数,负数用 1 表示,正数用 0 表示,每个数据的符号位占用 1 bit 存储空间。在实际工程中,将每帧数据中各个点的符号位作为该帧数据的帧尾,通过该方法使数据符合压缩条件。

针对第二类数字信号,利用一元一次函数的相关性压缩短字典长度。以正锯齿函数为例,其函数图形如图 2 所示。

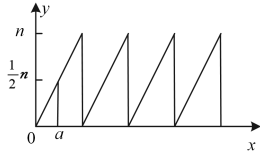


图 2 正锯齿图形

Fig. 2 Positive sawtooth pattern

根据函数性质,将函数 $f(x+a)$ 映射到 $f(x)$ 上。发生映射的数据用 1 表示,未发生映射的数据用 0 表示,每个数据的映射情况占用 1 bit 存储空间。在实际工程中,将整帧数据中各个点的映射情况作为帧尾,通过该性质的转化,使数据符合压缩条件。

以码长 $n=16$ bit 的数字信号为例,在未用抛物线性质的处理之前:码长 $n=16$ bit,字典长度 $L=2^{16}-1$ 。LZW 算法对该信号的压缩效果不理想。应用函数相关性处理后:码长 $n=16$ bit,字典长度 $L=2^{15}-1$,其中区间 $[2^{n-1}, 2^n-1]$ 永远不会被编码,借助该“空白区间”,改进 LZW 压缩算法提高了压缩比(本文称码长中未编码部分即码长与字典长度的差为“空白区间”)。由于在实际工程应用中每帧数据由帧头、帧尾、数据组成,在编码过程中,利用该特点用帧尾记录每个数据符号的变化情况。

3.3 方位特性与前缀编码

如 3.2 节所述,对于数字信号,采用 LZW 及其改进型压缩算法的效果不理想。因此,本文采用矩阵相关特性,改变数字信号的前缀编码方式,使数据压缩比符合工程中系统对数据传输的实时性要求。

将码长为 n bit 的数字信号转化成 $2^{\frac{n}{2}} \times 2^{\frac{n}{2}}$ 的矩阵形式(即将 n bit 中的高($\frac{n}{2}$)位作为矩阵的行,将 n bit 中的低($\frac{n}{2}$)位作为矩阵的列),如图 3 所示。

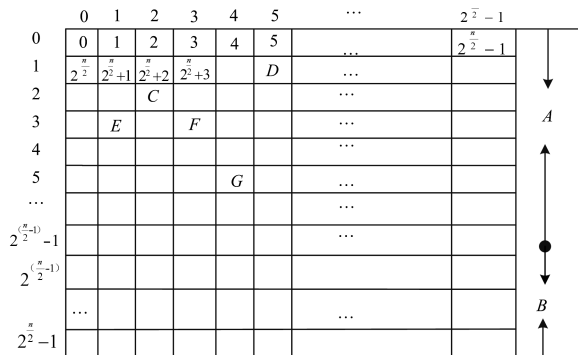


图 3 位置信息图

Fig. 3 Location map

左到右、从上到下的顺序均匀分布到 $2^{\frac{n}{2}} \times 2^{\frac{n}{2}}$ 的矩阵中,图 3 中 A 代表 $0 \sim (2^{(n-1)}-1)$ 的所有数据, B 代表 $2^{n-1} \sim (2^n-1)$ 的所有数据。根据 3.2 节的相关性质,将 B 中所有数据映射到 A 中,并根据矩阵坐标信息进行编码。编码集合由坐标信息和方位信息共同组成,规定如下:

坐标信息规定:行信息的范围为 $0 \sim (2^{\frac{n}{2}-1}-1)$,选择用 $\frac{n}{2}$ bit 表示行信息;列信息的范围为 $0 \sim (2^{\frac{n}{2}-1}-1)$,选择用 $\frac{n}{2}$ bit 表示列信息。方位信息规定:若前后两个相邻数据在对应矩阵中位于同一行(如图 3 中 E 与 F 的关系),则位置信息用 00 表示,其占用 2 bit 的存储空间;若前后两个相邻数据在对应矩阵中位于不同行,则位置信息用 01, 10, 11 表示,其占用 2 bit 的存储空间,其中,01 表示两数据中后者在前者的上一行(如图 3 中 C 与 D 的关系), 10 表示两数据中后者在前者的下一行(如图 3 中 C 与 E 的关系), 11 表示两数据既不同行又不在相邻行的情况(如图 3 中 F 与 G 的关系)。

在上述处理中,进行行信息编码时,只用到 $\frac{n}{2}$ bit 中的一半空间进行编码,剩余的另一半为“空白区间”,即每个数据都可以映射到该区间中,映射关系如下:

$$p_c = f(p_r, L) = p_r + L$$

其中, p_r 和 p_c 分别为映射前及映射后的值; L 为字典长度。解码时,一旦发现编码比当前字典长,则将其减去字典长度即可。根据上述方法将每个数据映射到“空白区间”后,采用位置信息实现数据的压缩。映射过程中的规定如下:

假设存在 A, B, C, D 4 个相邻数据,该数据中 A, B 分别用 n bit 编码,当 A 不发生映射时,说明 C 与 B 或 D 与 C 不存在方位关系,故 C 或 D 用 n bit 编码。当 A 发生映射时,说明 C 与 B 或者 D 与 C 存在方位关系,此时,只需将 C 或者 D 的列坐标与下一相邻数据的方位关系进行编码即可,故用 $(2 + \frac{n}{2})$ bit 编码。本文规定:若 A 和 B 同时发生映射,则证明 B 与 C 在同一行,故用 $(2 + \frac{n}{2})$ bit 对 C 进行编码;若 A 发生映射, B 与 C 都没有发生映射,则证明 D 在 C 的下一行;若 A, C 同时发生映射并且 B 没有发生映射时,则证明 D 在 C 的上一行。

数据码长为 16 bit 的,压缩编码情况如下:

11 7C(映射,用 16 bit 编码为: $11 * 2^8 + 7C + L$); 11 A5(映射,用 16 bit 编码为: $11 * 2^8 + A5 + L$); 11 BD(编码为 00 1011 1101), 11 CD(编码为 00 1100 1101), 11 D3(编码为 00 1101 0011), 11 CF(编码为 00 1100 1111), 11 C2(编码为 XX 1100 0010)。没有采用压缩编码时,每个数据均使用 16 bit 编码,共需要码长 $16 * 7 + 16 + 16 = 144$ bit;若采用本文压缩方法编码,共需要码长 $16 + 16 + 10 * 4 + 16 + 16 = 114$ bit,节约 $(144 - 114) / 144 = 20.8333\%$,即压缩比提高 0.263 158 倍。

3.4 改进算法的实现

本文改进型前缀编码过程如下。

步骤 1 根据本文算法得到“空白区间”。

转化过程:将码长为 n bit 的 2^n 个数据按照从小到大、从

步骤2 设置 $p_{c1}, p_{c2}, p_{c3}, p_{c4}, p_r$, 令 $p_r=0, p_{c1}=0, p_{c2}=0, p_{c3}=0, p_{c4}=0$, 其中 $p_{c1}, p_{c2}, p_{c3}, p_{c4}$ 是4个连续的待编码前缀, p_r 记录每个数据的压缩情况。

步骤3 按照原LZW算法编码字符得到待编码前缀 p_{c1} 。

步骤4 当 $p_r=0$, 或 $p_r=3$ 时, 获取 p_{c1} 后面连续的两个数据 p_{c2} 与 p_{c3} , 并转换到步骤5; 否则转到步骤6。

步骤5 当 p_{c2} 与 p_{c3} 在矩阵中的同一行时, 将 p_{c1}, p_{c2} 分别加上字典集合长度 L, p_{c3} 低 $\frac{n}{2}$ 位数据不变, 用下一个数据位置信息代替 p_{c3} 高 $\frac{n}{2}$ 位, 并用 p_r 记录该位置信息, 即 p_{c3} 用 $(2 + \frac{n}{2})$ 位取代原来的 n 位编码。当 p_{c2} 与 p_{c3} 不在同一行时, 获取 p_{c2} 与 p_{c3} 后面的 p_{c4} , 当 p_{c3} 与 p_{c4} 在相邻行时, 进行如下处理: 当 p_{c3} 在 p_{c4} 的上一行时, p_{c1} 加上字典集合长度 L, p_{c2} 与 p_{c3} 不变, p_{c4} 低 $\frac{n}{2}$ 位数据不变, 用下一个数据的位置信息代替 p_{c4} 高 $\frac{n}{2}$ 位, 并用 p_r 记录该位置信息, 即 p_{c4} 用 $(2 + \frac{n}{2})$ 位取代原来的 n 位编码; 当 p_{c3} 在 p_{c4} 的下一行时, p_{c1} 与 p_{c3} 分别加上字典集合长度 L, p_{c2} 不变, p_{c4} 低 $\frac{n}{2}$ 位数据不变, 用下一个数据的位置信息代替 p_{c4} 高 $\frac{n}{2}$ 位, 并用 p_r 记录该位置信息, 即 p_{c4} 用 $(2 + \frac{n}{2})$ 位取代原来的 n 位编码。如果数据不存在位置信息, 则不做处理, 转到步骤7。

步骤6 根据 p_r 记录的位置信息编码当前数据。即当 p_r 为0或1或2时, 用 $(2 + \frac{n}{2})$ bit 编码 p_{c1} , 并用 p_r 记录该位置信息, 转到步骤7。

步骤7 若未结束, 转步骤3, 否则结束。

在解码时, 因字典长度已知, 根据本文算法进行解压即可。

综上所述, 改进压缩算法基于压缩比因子对采样点进行映射操作, 使其能够标识后面相邻采样点的压缩情况; 然后该算法通过采样点间的方位信息, 缩短编码长度, 从而实现对采样点数据的压缩。

3.5 算法性能比较

4种压缩算法的性能比较如表1所列。

表1 算法性能比较

Table 1 Performance comparison of algorithms

性能	平均时间复杂度	是否稳定
Huffman 算法	$O(N^2)$	稳定
LZW 压缩算法	$O(N)$	稳定
文献[15]算法	$O(N)$	稳定
本文压缩算法	$O(N)$	稳定

注: N 为集合长度

Huffman 压缩算法编码涉及重排序, 常用的排序方式有两种, 即冒泡排序和插入排序, 前者的关键操作是比较和交换, 后者的关键操作是定位和移动。两种排序方式中, 后者时间复杂度较低, 故本文采用插入排序实现 Huffman 编码, 时

间复杂度分析如下:

由文献[20]可知, 插入排序的时间复杂度为 $O[\log_2(N-2)+1+(N-2)]$, 其中3项分别代表定位、求和和移动, 因此其综合时间复杂度为 $O(y) = O[\log_2(N-2)+1+(N-2)] + O[\log_2(N-3)+1+(N-3)] + \dots + O[1+1+1]$, 共 $N-2$ 项, 可以计算出以下表达式^[20]:

$$O(y) = O[(\log_2(N-2) + \dots + 1) + (N+1) * (N-2)/2]$$

即 $O(y) = O(N^2)$ 。

LZW 压缩算法: 该算法依赖快速的字典查找, 由文献[21]可知, 该算法的时间复杂度为 $O(y) = O(NM + M^2N / \log_a(N))$, 其中, M 为字典集合的个数(一般 $M=1$), $a(N)$ 为字母表的长度, 即 $O(y) = O(N)$ ^[21]。

文献[15]的算法: 该算法在LZW压缩算法的基础上采用映射方法改变集合编码顺序, 实现了变长编码, 减少了数据传输的总长度, 但并没有改变原算法的时间复杂度, 故其时间复杂度为 $O(y) = O(N)$ 。

本文压缩算法: 在LZW压缩算法的基础上, 本文算法通过改进编码方式和字典集合, 使每个采集点的编码长度缩短为原来的一半, 虽然时间复杂度依旧为 $O(y) = O(N)$, 但搜索范围缩小为原来的一半, 提高了字典集合的搜索效率。

综上所述: 本文压缩算法的性能优于其他算法。

4 实验与分析

为了验证本文改进算法的有效性, 采用风洞实验中得到的两种数字信号对该算法进行测试: 一类为码长为16 bit、采样频率为100 kHz、信号频率为5 kHz、信噪比较低的周期信号, 其标号为信号1; 另一类为码长为16 bit、采样频率为100 kHz、信号频率为5 kHz、信噪比较高的周期信号, 标号为信号2。其中数据点个数为: 100 000, 600 000, 800 000, 1 000 000, 3 000 000, 5 000 000, 7 000 000, 10 000 000。

算法初期通过 MATLAB 仿真验证算法的有效性, 后期是将算法应用到风洞实验中。

硬件实验环境: FPGA 中固态存储使用的是 Micron 公司的 MLC 型 Nand Flash, 型号为 MT29F16G08CBACAWP, 内存为 2 GB。下位机同时对多通道的信号进行同步采集, 并将数据暂存于 NAND Flash(即固态存储器)中。使用标准 C 语言实现本文压缩算法, 并生成 .ELF 文件, 供 FPGA 调用, 以实现数据的压缩与传输。

针对数字信号, 图4—图7反映了4种算法性能的差别。

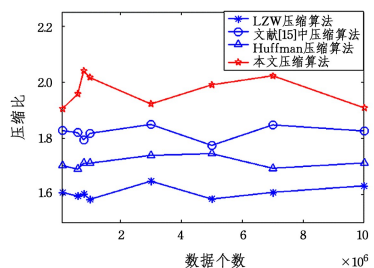


图4 信号1的压缩比对比图

Fig. 4 Compression ratio comparison chart of signal 1

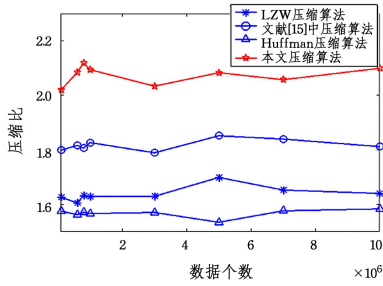


图 5 信号 2 的压缩比对比图

Fig. 5 Compression ratio comparison chart of signal 2

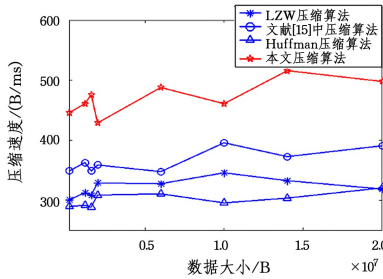


图 6 信号 1 的压缩速度对比图

Fig. 6 Compression speed comparison chart of signal 1

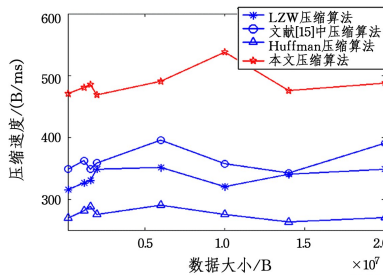


图 7 信号 2 的压缩速度对比图

Fig. 7 Compression speed comparison chart of signal 2

压缩比方面:信号 1 和信号 2 的信噪比不同,导致每个数据出现的概率不同。数据概率分布情况对 Huffman 压缩算法影响较大,因此 Huffman 压缩算法对信号 1 和信号 2 的压缩效果均不理想,并且信噪比越高,压缩效果越不理想;文献 [15] 中改进的 LZW 压缩算法的压缩比有所提高,但信号 2 比信号 1 的压缩比低;本文压缩算法对于信号 1 和信号 2 两类信号的压缩效果均较理想,压缩比可提高约 26%,并且信噪比越高,压缩效果越好。

压缩速度方面:Huffman 压缩算法需要对所有数据进行两次扫描,根据数据概率进行编码。当信号采样频率为 100 kHz,即速度为 200 B/ms 时,如果数据传输过程中出现重传现象,Huffman 易出现采集速度与压缩速度不匹配的情况;文献 [15] 的算法的压缩速度虽然有所提高,但是效果并不是最优的;本文压缩算法通过对数据进行相关处理,将 16 bit 的数据码长缩短为 10 bit,在不增加算法复杂度的前提下,很大程度上缩短了字典集合的长度,进一步提高了压缩速度。其对信号 1 和信号 2 的压缩速度效果均较理想,并且信噪比越高,压缩速度越快,对于某些特殊环境的实验例如风洞实验更加实用。

结束语 本文改进压缩算法基于压缩比因子对采样点进行映射操作,使其能够标识后面相邻采样点的压缩情况,然后通过采样点间的方位信息,缩短采样点的码长,再根据 LZW 压缩算法对数字信号进行压缩。与原 LZW 算法相比,该算法存在两方面的优势:1)在几乎不增加计算量和存储空间的前提下,可有效缩短压缩时间,提高压缩比;2)其通用性好,即只要满足条件 $(2^{(n/2)-k} + 1) < R$ 的数字信号都适用。信号 1 与信号 2 中的部分数据压缩实验验证了本文改进型算法的有效性。本文压缩算法对于满足条件 $(2^{(n/2)-k} + 1) < R$ 的数字信号,压缩效果较明显,对于重复率较低的文本文件,压缩效果一般,需做进一步研究。

参考文献

- [1] XIE R Y, HAI B Z. Lossless Compression Algorithm Based on Hybrid Coding of Adaptive Huffman and Golomb-Rice for WSN [J]. Computer Engineering, 2016, 42(7): 86-93. (in Chinese) 解瑞云,海本斋.基于自适应霍夫曼和 Golomb-Rice 混合编码的 WSN 无损压缩算法[J].计算机工程,2016,42(7):86-93.
- [2] SAYOOD K. 数据压缩导论(4版)[M].贾洪峰,译.北京:人民邮电出版社,2014.
- [3] HUFFMAN D A. A for the Construction of Minimum-redundancy Codes [J]. Proceedings of the IRE, 1952, 40(9): 1098-1101.
- [4] AL-OKAILY, ALMARRI A, YAMI B A, et al. Toward a Better compression for DNA Sequences Using huffman Encoding [J]. Journal of Computational Biology, 2017, 24(4): 280-288.
- [5] ZIV J, LEMPEL A. A Universal Algorithm for Sequential Data Compression [J]. IEEE Transactions on Information Theory, 1977, 23(3): 337-443.
- [6] ZIV J, LEMPEL A. Compression of Individual Sequences via Variable-rate Coding [J]. IEEE Transactions on Information Theory, 1978, 24(5): 530-536.
- [7] CHEN Q H, CHEN X S, HAN D L. Compression on algorithm LZW on Chinese text [J]. Computer Engineering and Application, 2014, 50(3): 112-116. (in Chinese) 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.
- [8] WANG H G, ZHANG P L, WU D H, et al. Lossless compression of slowly varying signals based on dynamic LZW and arithmetic coding [J]. Application Research of Computers, 2015, 32(9): 2742-2756. (in Chinese) 王怀光,张培林,吴定海,等.基于动态 LZW 与算术编码的缓变信号压缩[J].计算机应用研究,2015,32(9):2742-2756.
- [9] TRABUCO M H, COSTA M V C, et al. S-EMG Signal Compression in One-Dimensional and Two-Dimensional Approaches [J]. IEEE Journal of Biomedical and Health Informatics, 2018, 22(4): 1104-1113.
- [10] SANGEETHA M, BETTY P, KUMAR G S N. A Biometric Iris Image Compression using LZW and Hybrid LZW Coding Algorithm [C]//2017 International Conference on Innovations in In-

- formation, Embedded and Communication Systems (ICIIECS), Coimbatore, India; IEEE Press, 2017: 75-81.
- [11] LI T J, ZHAO T D, NHO M, et al. AN ovel RLE & LZW for Bit-Stream Compression [C] // 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT). Hangzhou, China; IEEE Conferences, 2016: 1600-1602.
- [12] GAO F, SUN C J, SHAO Q L, et al. Lossless compression of hyperspectral images based on k-mean clustering and conventional recursive least-squares [J]. Journal of Electronics & Information Technology, 2016, 38(11): 2709-2714. (in Chinese)
高放, 孙长建, 邵庆龙, 等. 基于 K-均值聚类 and 传统递归最小二乘法的高光谱图像无损压缩 [J]. 电子与信息学报, 2016, 38(11): 2709-2714.
- [13] NANDI U, MANDAL J K. A Compression Technique Based on optimality of LZW Code (OLZW) [C] // Proceedings of the 3rd International Conference on Computer and Communication Technology. Washington D. C., USA; IEEE Press, 2012: 166-170.
- [14] XU D W, WANG Y D, JIA L M, et al. Compression Algorithm of Road Traffic Spatial Data Based on LZW Encoding [J]. Journal of Advanced Transportation, 2017, 16(2): 1-13.
- [15] MARTIN T, MARKET T. Data reduction for Boolean matrix factorization algorithms based on formal concept analysis [J]. Knowledge-based Systems, 2018, 158(15): 75-80.
- [16] YAN H Z, XU B G, SHI D J, et al. Prefix Encoding Optimization and Application of Lossless Compression Algorithm LZW [J]. Computer Engineering, 2017, 43(3): 299-303. (in Chinese)
鄢海舟, 胥布工, 石东江, 等. 无损压缩算法 LZW 前缀编码优化及应用 [J]. 计算机工程, 2017, 43(3): 299-303.
- [17] XU J P, LIU L, LI W, et al. A New Lossless Compression Algorithm Based on Array Configuration Speedup Model [J]. Journal of Electronics & Information Technology, 2018, 40(6): 1492-1498. (in Chinese)
徐金甫, 刘露, 李伟, 等. 一种基于阵列匹配加速比模型的无损压缩算法 [J]. 电子与信息学报, 2018, 40(6): 1492-1498.
- [18] LI C Z, YU J B, ZHANG M, et al. Obfuscation tool for mobile apps based on Huffman and LZW encoding [J]. Journal of Software, 2017, 28(9): 2264-2280. (in Chinese)
李承泽, 於剑波, 张森, 等. 一种基于 Huffman 和 LZW 编码的移动应用混淆方法 [J]. 软件学报, 2017, 28(9): 2264-2280.
- [19] TUAMA A Y, MOHAMED M A, MUHAMMED A, et al. A new compression algorithm for small data communication in wireless sensor network [J]. International Journal of Sensor Networks, 2017, 25(3): 163-175.
- [20] LIU Y Q, GONG S R. A Sort-once and Dynamic Encoding (sode) Based Huffman Coding Algorithm [J]. Computer Applications and Software, 2009, 26(12): 86-89. (in Chinese)
刘燕清, 龚声蓉. 基于一次排序动态编码的 Huffman 编码算法 [J]. 计算机应用与软件, 2009, 26(12): 86-89.
- [21] FILATOV G, BAUWENS B, KERTÉSZ-FARKAS A. lzw-Kernel: fast kernel utilizing variable length code blocks from lzw-compressors for protein sequence classification [J]. Bioinformatics, 2018, 34(19): 3281-3288.