基于 Zlib 和 DSP 的传感器数据压缩方法的研究

谢小勇，伍瑞卿，陈 伟，顾庆水，樊 丰
（电子科技大学 电子工程学院，四川 成都 611731）

摘 要：针对传感器数据在分组传输环境下的特点和要求，提出改进型的 Zlib 算法，实现对数据的分包无损压缩并能够独立解压，同时将优化压缩程序移植到 DSP F2812。以测井传感器数据进行验证，实验结果表明，该方法保持了良好的压缩效果，而且具有抗误码和丢包容错的能力。

关键词：传感器数据；Zlib；无损压缩；DSP；容错

中图分类号：TN919.3　　**文献标识码**：B　　**文章编号**：0258-7998(2012)11-0041-03

Research of sensor data compression method based on Zlib and DSP

Xie Xiaoyong, Wu Ruiqing, Chen Wei, Gu Qingshui, Fan Feng
(School of Electronic Engineering, University of Electronic Science and Technology, Chengdu 611731, China)

Abstract: Aiming at characteristics and requirements of the logging sensor data packet transmission environment, this paper presents improved Zlib algorithm, which is able to achieve lossless compression and sub-logging data to be independent decompression. And then optimize the compression program ported to DSP F2812. With logging sensor data to verify, the test results show that the compression is good, and be able to solve the problem of fault-tolerant network errors and packet loss.

Key words: sensor data; Zlib; lossless compression; DSP; fault-tolerant

在多传感器测量系统中，为了精确测量，需要对传感器输出信号进行高速、高分辨率采样[1]。这样势必会产生大量需要存储的数据，占用很大的存储空间，同时也不便于实时传输。但在一些嵌入式系统中，由于受到工艺、成本等各种因素的限制，存储器空间非常有限，以至于经常采取降低采样率的办法来达到节省存储空间的目的。为此，在采样率和分辨率均不能降低的情况下，为了保证测量精度，对传感器信号进行压缩存储、传输是十分必要的。

目前，有关传感器数据压缩传输的研究有：(1)参考文献[2]提出一种存储有效的渐进小波数据压缩算法，使得渐进传送的数据单元能产生大的编码增益，在存储有效的同时又能够节省网络传输耗能；(2)参考文献[3]提出了一种基于改进的二叉树算法对原始采样数据进行实时压缩，对采样数据在二维方向上进行去冗余处理，从而达到节省数据存储空间、提高数据传输效率的目的；(3)参考文献[4]针对测井传感器数据特征将其分为两类，第一类采用差分编码（预处理），再用 Huffman 编码压缩输出，第二类采用基于统计的预测模型和分段线性预测模型(PLOT)进行处理输出。

但以上几种方法都是有损压缩，无法完全恢复出原始数据，并且压缩数据前后都有相关性，没有提出一旦出现误码或丢包情况将如何正确解压的问题。本文以无损压缩方法中性能较优的 Zlib 为基础，并采取改进措施，在保证压缩率和实时性的前提下，解决网络传输中误码和丢包容错问题，同时对测井传感器数据进行了测试实验。

1 Zlib 无损压缩方法

无损压缩就是利用数据的统计冗余进行压缩，并可完全回复原始数据而不引起任何失真。典型的无损压缩算法如 Shanno-Fano 编码、Huffman(哈夫曼)编码、算术编码、游程编码、LZW 编码、LZ 编码以及应用最广也是最好的结合 LZ 和 Huffman 编码的 DEFLATE 编码，其主要应用在 RAR、LZMA、Zlib、Gzip、Bzip 等压缩算法中。

虽然 LZMA 具有好的压缩率，但是消耗资源过大（最小资源消耗在 1 MB 左右）且压缩速率很慢，不适合实际嵌入式应用。所以本文选用 Zlib 压缩算法，其中 SYNC_FLUSH 模式下的流程图如 1 所示。

首先初始化程序，然后将待压缩数据存到处理窗口中，用 LZ 算法对待压缩数据进行预处理。一方面通过

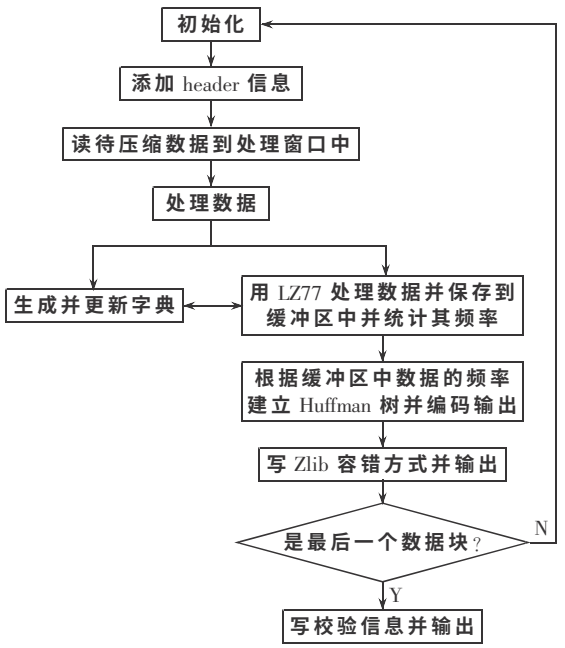

图 1 Zlib 压缩程序中 SYNC_FLUSH 模式流程图

Hash 算法生成并不断更新当前块的压缩字典；另一方面通过查找字典将数据匹配情况存到缓冲区中并统计其频率。这样通过未匹配字节、匹配长度和匹配距离描述了原始数据的匹配关系。当处理窗口数据被处理完之后，再利用动态和静态 Huffman 算法对其匹配情况进行编码，并比较选择压缩效果好的 Huffman 算法进行压缩编码输出。但是如果选择的是动态 Huffman 算法，还需要在压缩数据之前用 deflate_tree 编码压缩动态 Huffman 字典编码并输出。当每个压缩块完成之后再加上 Zlib 容错方式，如果是最后一个压缩块，则还需要最后输出校验信息，否则需要重新初始化压缩下一个数据块。

2 改进的 Zlib 算法

2.1 基本思想

源 Zlib 压缩算法中，SYNC_FLUSH 模式虽然通过添加 header 信息和 Zlib 容错方式字段实现分包压缩，同时每个压缩块都重新生成字典实现相互独立，但是仍然不能解决容错问题，原因如下。

(1)如果压缩数据在网络传输过程中出现误码的情况，并且误码之后的数据又能够通过当前块的压缩字典解压，但是解压错误，而最后输出错误的解压结果不报错；

(2)如果出现丢包的情况，则解压过程中会报异常且直接终止跳出，导致后面正确的压缩块不能正常解压。

针对以上两个问题，本文提出改进并实现独立分包压缩并且能够独立解压的方法，以解决因网络传输导致的误码和丢包的容错问题。此外，根据实际程序应用修改滑动窗口大小和内存消耗等级，大大减小了内存资源消耗。

2.2 独立分包压缩的算法流程

每次读 4 KB 样本数据，然后独立压缩成一个压缩块放到输出缓冲区，当输出缓冲区内满 2 KB 数据时就写到输出文件中。每次把 4 B 检错信息 adler 写到压缩块最后(Zlib 容错方式 0x0000ffff 之前)，便于解压检错实现容错机制。这样就得到如图 2 所示的压缩数据格式。

独立分包压缩要求每次都要初始化字典并重新建立字典，这样会导致压缩率(压缩率=压缩后数据大小/压缩前数据大小)提高(即压缩效果变差)。但是测试表明，压缩效果较压缩效果最好的 NO_FLUSH 模式(其生成的压缩包之间相互关联，如果出现误码或丢包情况，就会使出错后的压缩块不能解压)并没有下降很多，压缩率只提高了 2%；而与源程序的 SYNC_FLUSH 模式压缩率几乎相等，因为虽然每个压缩包都加了 4 B 的检错信息，但是只有第一个压缩块加了 2 B 的 header 信息，而源程序中每个压缩块都要加 header 信息且最后一个加了检错信息，最终生成的压缩数据比源程序多$(2\times N-2)$B，其中 N 表示压缩块的个数。由于都是独立分包压缩，解压可以容错，并且可以极大地减小其程序消耗的资源。因为这时并不需要分配大的滑动字典窗口和字典空间，优化之后总共动态分配空间为 96 KB 左右，而 NO_FLUSH 模式下资源消耗在 260 KB 左右。

2.3 独立解压的算法流程

本文是模拟传感器数据独立分包压缩再整体解压，因为在实际工程中解压程序是源源不断地从输入缓冲区中读数据解压。但是如果读入的待解压数据出现误码和丢包情况(而解压程序并不知道)，解压程序必须实现独立解压功能，即丢弃出错的压缩块而又不影响后面的正确的压缩块解压，这就要求每个压缩包必须是独立压缩。采用在每个压缩块后面写入检错信息 adler 和 Zlib 容错方式就能很好地解决这个问题。具体解压流程图如图3 所示。

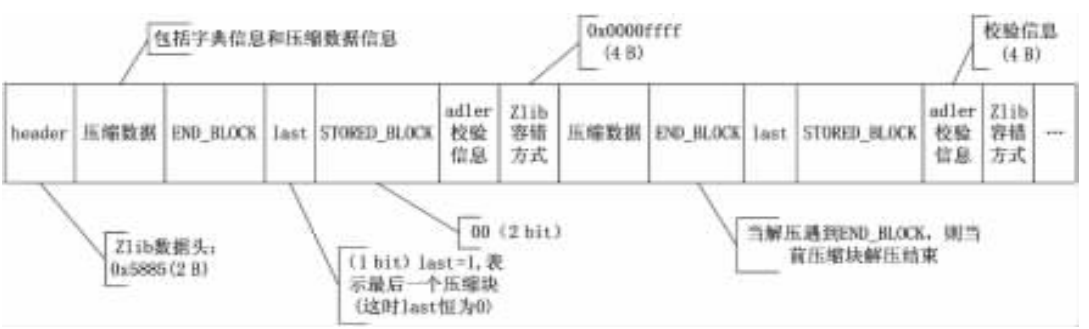

图 2 改进后 Zlib 压缩块格式

为了便于误码或丢包的测试，将压缩数据存放在一个缓冲区中。这就要求预判压缩数据大小再分配缓冲区空间保证有足够大，再整体解压；然后就是模拟误码和丢包情况：因为在分包压缩过程中可以统计每个压缩块的大小，再根据压缩块的格式就可很容易修改 Zlib 头信息、压缩数据、检错信息和 Zlib 容错方式来模拟误码，并且通过移除一定长度的压缩数据来模拟丢包情况；最后对测井传感器数据测试验证了容错能力。

2.4 压缩算法的 DSP 实现

首先在 CCS3.3 编译环境下，选择 F2812 的 Device Simulator 仿真。其 DSP 扩展一个 256 KB 的片外空间，因为程序在 F2812 中测试程序总空耗（程序执行消耗总资源，包括动态分配 heap 空间 96 KB 左右）为 140 KB 左右，并选择大寄存器模式；设置 heap 大小；在调试程序通过之后选择 XDS560 emulator 将程序下载到 F2812 上运行。在运行之前设置一个定时中断，定时从样本数据文件中读数据到输入缓冲区，而在压缩程序中每次直接从输入缓冲区中读数据进行压缩。

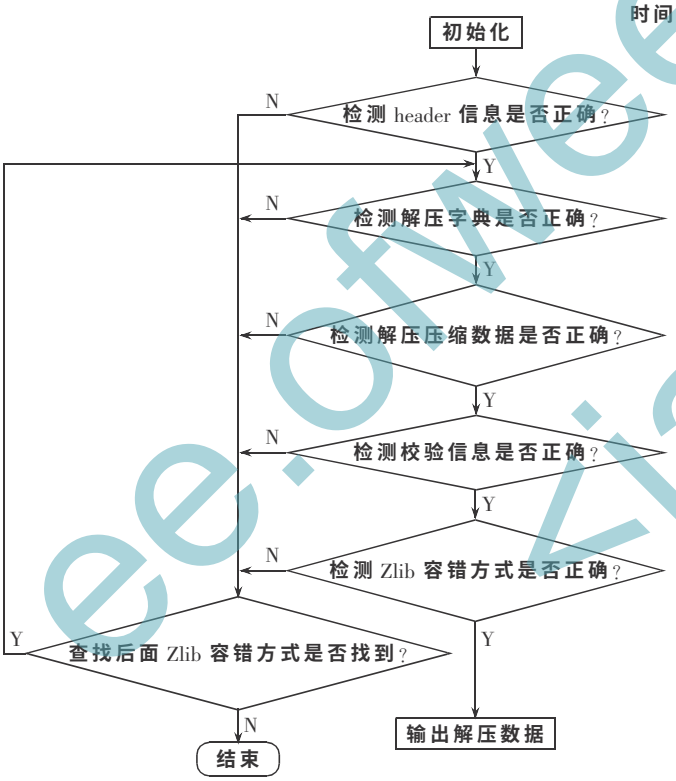

图 3 改进后 Zlib 解压流程图

3 实验与分析

在现代传感器遥测遥传网络中，石油测井网络是极具代表性的。随着测井技术的发展，尤其是组合测井、成像测井的出现，造成了测井数据的快速膨胀。但是数据传输带宽的限制和实时的要求，不能够把这些数据实时而可靠地传到地面，成为阻碍测井技术发展的一大问题，迫切需要解决。而应用无损数据压缩技术来减小数据传输量是最有效途径之一[5]。

所以本文以石油测井网络为例，利用 DSP F2812 嵌入式平台对测井数据进行实时压缩，并把压缩后的数据实时传到地面 PC 机后再由网络传输到控制中心进行实时解压缩。测试结果如表 1 所示。其中样本数据主要由自然伽玛能谱测井仪、方位测井仪、电缆遥测传输仪、总线适配器等仪器采集生成。

表 1 测试结果

数据样本	压缩率/%	平均时延/ms	最大时延/ms	压缩速率/(Kb/s)
DST61X_main	12.63	14.52	16.41	2 203.7
FAP21X_main	1.05	5.71	5.77	5 016.1
FAP21X_E211	7.7	10.82	11.14	2 957.1
ORT61X_main	33.01	26.14	27.86	1 224.1
ORT71X_E211	57.12	44.77	45.67	714.8
RTT61X_main	64.77	47.27	47.97	677
RTT61X_E211	74.61	51.55	52.53	620.7
RTT61X_E201	63.74	45.16	46.03	708.6
SAT61X_main	48.11	34.53	35.97	926.7

注：压缩率 =（压缩后数据大小）/（压缩前数据大小）*100%；时延是一个 4 KB 数据输入到压缩成一个压缩块所需要的时间；压缩速率等于单位时间内压缩处理多少 kb 数据。

测试结果表明，不同类型测井数据样本压缩率、时延和压缩速率差异还是很明显的，主要由于不同样本文件的数据结构差异很大。而且在满足独立分包压缩前提下，每个包都要重新建立字典，包越小压缩效果越差，而本方法的包只有 4 KB，压缩率基本上都在 60%以下，但实时性也能达标。因为压缩速率大于测井仪器的采集数据速率（这取决于不同的仪器速率传输能力和采集间隔的控制，但最大不超过 500 kb/s）。而在没改进之前，farheap 空间占 192 KB，heap 空间占 40 KB，程序总消耗约为 260 KB，所以改进后的 Zlib 算法运行所占资源很少且压缩效果很好，适合于嵌入式平台应用。由于压缩程序的移植优化只是在算法和程序结构上，并没有进一步在汇编级别上的优化，因此压缩速率提升空间还很大。

本文针对传感器数据在分组传输环境下的特点和要求，提出基于 Zlib 的传感器数据压缩方法，并在 DSP 嵌入式平台上得到应用验证。本压缩方法具有良好的压缩效果和压缩速率，并且能够实现分包压缩独立解压，解决了在网络传输过程中误码和丢包的容错问题，

参考文献

[1] LEMING S K, STALFORD H L. Bridge weigh-in-motion system development using superposition of dynamic truck/static bridge Interaction[C]. IEEE American Control Conference, 2003.

[2] 周四望，林亚平，叶松涛，等. 传感器网络中一种存储有效的小波渐进数据压缩算法[J]. 计算机研究与发展，2009，46(12)：2085-2092.

[3] 刘贞，王祁. 多传感器信号数据采集实时压缩算法[J]. 传

感技术学报,2006,19(6):2712-2715.
[4] 汉泽西,郭枫,吕飞.测井数据的无损压缩方法研究[J].现代电子技术,2004(22):94-96.
[5] 郭枫,汉泽西.测井数据的无损压缩方法研究[D].西安:西安石油大学,2005.

(收稿日期:2012-05-25)

作者简介:

谢小勇,男,1987年生,在读研究生,主要研究方向:测井数据压缩及网络传输。

伍瑞卿,男,1977年生,副教授,硕士生导师,主要研究方向:网络通信,遥测遥控通信。

陈伟,男,1978年生,博士,讲师,主要研究方向:OFDM宽带无线通信技术,数字视频技术,DSP技术等。