车载数据压缩算法①

王思洋，许 勇

(桂林电子科技大学 电子工程与自动化学院，桂林 541004)

摘 要：提出了一种适用于车载数据存储的线性拟合算法，采用基于 ARM-Linux 操作系统的 SQL 数据库为平台，对提出的压缩算法加以实现。实验结果表明：针对大量的连续数据具有良好的压缩效果，压缩后数据量急剧减少，尤其是在数据波动和采样间隔时间较小的情况下，给出的方法可实现较高的压缩比。

关键词：数据压缩；信息存储；数据库算法；线性拟合；数据采集

Vehicular Data Compression Algorithm

WANG Si-Yang, XU Yong

(School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: A linear-fit algorithm used for on-board vehicle data storage is presented, which is implemented by using a SQL database based on ARM-Linux operating system. The results show that the compression performance is very good for larger numbers of successive data and the number of data would reduce rapidly. In particular, a high compression ratio can be gained by the proposed method for low data fluctuation and less sample interval.

Key words: data compression; information storage; database algorithm; linear fit; data gathering

1 引言

随着汽车技术的日新月异，汽车正朝着电子化、智能化和网络化发展，车内设备的运行监控和调度管理将越来越复杂，动态信息量也越来越大[1]。当汽车出现问题后，其故障诊断也会因此变得更加复杂，若能提供必要的历史数据信息势必将使汽车故障诊断和维修变得更加容易。目前市场上的汽车黑匣子已经能够完整、准确地记录汽车行驶状态下的有关情况，能将汽车行驶轨迹完整地记录下来，并通过专用软件在电脑上再现。

汽车黑匣子主要是通过记录短期的行驶数据来提供交通事故分析依据，但是，在进行汽车维护、维修和故障诊断时，更多的是需要大量历史数据作为参考，而记录长期的历史数据需要大量的存储空间，这将为存储介质的容量增加了负担，若将每次的行驶数据进行备份又十分不便。为此，提出了一种适用于车载数据长期存储的压缩算法，该方法既可以对近期数据进行无损压缩，也可以对长期数据进行适当的有损压缩，从而节省了大量的存储空间，减少了存储介质备份的次数，为汽车数据提供了长期的备份。

2 算法原理

车载动态数据可以看成是随时间变化的动态函数，并且该函数的大部分区域服从线性或者服从符合精度要求的准线性分布，因此，利用这一特性将动态函数划分成若干个服从线性或者准线性分布的段函数，对于服从符合精度要求的准线性分布函数采用线性拟合算法进行压缩，即对一些服从准线性分布的函数段进行线性化处理，最后提取每个段函数的初始值、斜率以及时间长度，并依次存储到数据库列表中[2]。当需要查看历史数据时，只需要对数据库列表依次进行解压即可。

算法原理如图 1 所示，上图为动态数据仿真，横坐标 t 是汽车运行时间，纵坐标 V 是汽车运行速度，该曲线是模拟汽车在一段时间内从启动到平稳运行的

① 收稿时间:2011-01-24;收到修改稿时间:2011-03-05

速度函数，下图为利用线性拟合算法将该函数划分成若干个段函数，可以看出，越平稳的区域拟合度越高。

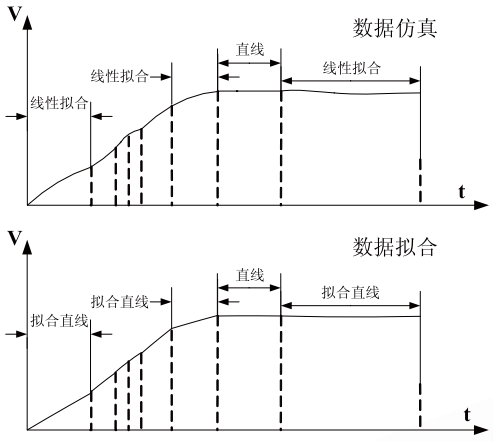

图 1 算法原理

3 压缩算法设计

压缩算法线性拟合过程依据初始斜率偏差值、波动率偏差值以及最大预期偏差值来进行拟合，线性拟合原理如图 2 所示，图 2 中线段为一段时间对汽车速度 V 的采样函数，Kb 为基准斜率，每个新采样点确定下来的斜率值与 Kb 的差值为基准斜率偏差，基准偏差值的大小决定了线性拟合精度，精度要求越低线性拟合度越高；Kc 与 Kn 的差值为波动率，即新的采样点确定下来的斜率与之前采样点确定下来斜率的差值，波动率的大小决定线性拟合的稳定性；最大偏差值是第 N 个采样值与以基准斜率为标准的第 N 个理论值的实际偏差，最大偏差值决定了整体线性拟合偏离度，偏差值越大，允许的偏离度越高。

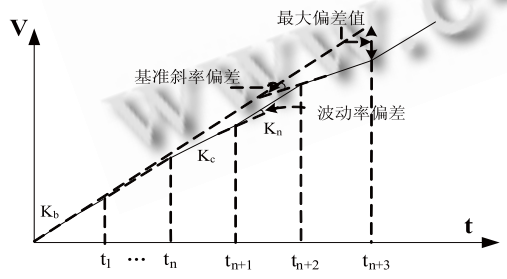

图 2 线性拟合原理

为了方便说明，假设待压缩缓存数组空间大小为 Data[10]，先提取前两个采样点数据以确定拟合直线的基本斜率 K_b(K_b=Data[1]-Data[0])，再计算采样点数据 3 与采样点数据 2 的待比较斜率 K_c(K_c=Data[2]-Data[1])，然后对 K_c 和 K_b 做差分运算求出差分值 O_b（O_b=K_c-K_b），若|O_b|>&$_b$(&$_b$ 为基准斜率精度)，说明不符合该拟合精度，那么只保存前两个采样点 Data[0]和 Data[1]，并重新确定下一个拟合直线的基本斜率，若|O_b|<=&$_b$，那么将继续提取采样点数据 4,并计算新的采用斜率 K_n(K_n=Data[3]-Data[2])，然后进行差分运算 Vc（V_c=K_n-K_c），若|O_c|>&$_c$(&$_c$ 为波动率值)，那么保存之前 3 个采样点，然后从新确定拟合直线的基本斜率；若|O_c|<=&c，那么继续与 K_b 作差分运算来判断基准斜率偏差，计算 O_b(O_b=K_n-K_b)，若|O_b|>&$_b$ 则保存之前 3 个采样点，若|O_b|<=&$_b$ 那么将 K_n 赋给 K_c，继续提取新的采样点，计算 K_n(K_n=Data[4]- Data[3])，依次判断是否符合波动率和基准斜率的范围值，若不符合则保存之前 4 个采样点，若符合范围之内，则继续提取新的采样点。

当提取完该数组的最后采一个样数据 Data[9]时，重新提取缓存区数据，然后继续提取采样值数据 Data[0]，依次循环，直到超出精度范围后，再将拟合数据记录进行保存，每次保存的记录数据包括基本斜率 K_b、初始采样点、拟合个数和采用时间间隔。

4 硬件平台搭建

在 ARM-Linux 操作平台的基础上移植 SQLite3 数据库，首先下载第三方数据软件包到 PC 虚拟机上，然后编译相关文件，再把生成的文件下载到开发板。采用的是开放的嵌入式 Linux 交叉开发环境，Linux 交叉编译工具链使用的软件包是 cross-3.4.1.tar.gz。PC 机是 RedHat9.0 环境。以下步骤都是在虚拟机上执行：

① 首先需要建立交叉编译环境，下载 cross-3.4.1.tar.gz 并拷贝到/usr/local/arm 目录，进入该目录下，执行 tar zxvf cross-3.4.1.tar.gz 解压命令，编译/etc/bashrc 文件，修改交叉编译路径为 export PATH=/usr/local/arm/3.4.1/bin：$PATH。

② 下载第三方数据库软件包 sqlite-3.6.18.tar.gz，并拷贝其到/temp 目录下，执行 tar -zxvf cross-3.4.1.tar.gz 命令解压。进入解压目录执行../configure --disable-tcl-host=arm-Linux--prefix=/root/sqlite-3.6.18/build/target 会生成相关配置文件，该文件生成目录为--prefix 后面指定的目录。

③ 执行 make 和 make install 命令对配置文件编译安装。执行完毕后会在 target 目录下生成 bin、include 和 lib 三个目标文件夹，bin 为数据库的命令文件，include 为

头文件库，lib 为执行数据库时需要调用的库文件。将这 3 个文件夹拷贝到 OK2440 开发板/usr 目录下所对应的 bin、include 和 lib 目录中。执行 chmod +x sqlite3，chmod +x bin include lib 命令修改目录权限，最后执行./sqlite3 命令，待出现 sqlite3>提示符，表明数据移植成功。

5　压缩算法的软件实现

数据压缩过程如图 3 所示。首先根据要求确定各项拟合精度值，然后将采样数据提取到待压缩缓存数组中，缓存区的数据利用压缩算法进行处理，最后将每一段压缩数据按顺序存储到指定数据库列表中。

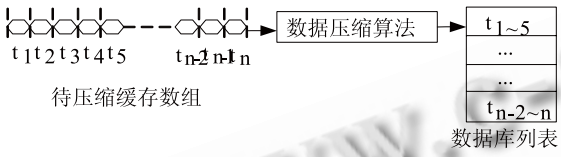

图 3　数据压缩过程

压缩算法实现是在虚拟机上编写 C 文件脚本，并调用相关数据库函数，再通过交叉编译环境编译后移植到 ARM-linux 系统平台上，主体框架思想如下：

采用双缓存数据存储方式，采样数据依次存储到数组缓存区中，待缓存区存满后，被锁定并使能待压缩标志，当缓存区的数据被提取后才能释放该缓存区，在缓存区的数据未被提取之前，采样数据将自动存储到第二缓存区中。采样数据以及相关存储变量在未压缩到数据库列表之前，会实时存储在数据库的临时存储列表中，这样有效防止因断电或故障带来的数据丢失。

压缩算法流程如图 4 所示，进入数据压缩函数入口后，首先打开数据库，恢复相关存储变量，将上次未进行压缩处理的数据到待压缩缓存区中，待扫描到压缩标志使能后，提取被锁定的压缩数组，进入线性拟合处理阶段，具体步骤如下：

① 根据基准斜率标志位是否有效，判断是否需要确定初始基准斜率 K_b；

② 确定比较斜率 K_c，得出基准斜率偏差 K_b（该步骤只在基准斜率标志位有效后，执行一次）；

③ 提取新采样数据并得出斜率 K_n，计算基准斜率偏差 K_b 和波动率偏差 K_c（该步骤在基准斜率标志位有效后，第一次不执行）；

④ 根据基准斜率计算最大偏差值；

⑤ 判断各个偏差值是否在允许范围内，若都符合精度范围那么继续提取待压缩采样数据，重复步骤③和④，若超出精度范围，那么存储初始采样点、拟合基准斜率 K_b、采样个数 C 和采样间隔时间 T 到数据库列表中，然后回到步骤①；

⑥ 若在拟合到该数组的最后一个数据时，各项偏差值仍然符合精度范围要求，那么判断缓存空间是否还有待压缩数组，若没有待压缩数据则存储当前状态到数据库中，以便下次恢复数据使用；若存在待压缩数组，则继续提取数组，并回到步骤①。

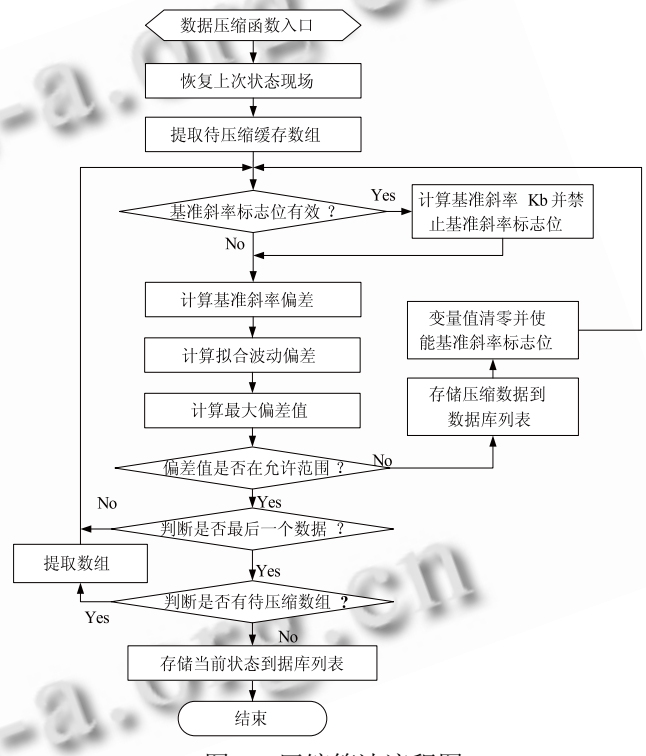

图 4　压缩算法流程图

当需要对压缩数据进行恢复时，只需要依次对数据库列表数据进行解压即可，解压方法是根据公式“Dre=Dinit +Kb*C”（Dre 为被解压数据，Dinit 为初始采用值，C 拟合个数）来实现，再根据初试采样时间和采样间隔来确定时间位置，最后就可以将解压数据读取到界面上。

系统可以根据数据的时间自动改变各项偏差值从而对长久的数据进行适当的低精度压缩，提高压缩率。

6　系统测试

该测试是以车载速度变化为例，将原采用数据与解压缩数据（拟合数据）放到同一坐标上进行对比。

缓存空间大小设为 10，采用双缓存方式，有 22 个数据采样点，采样间隔时间为 0.5s，设置两组拟合参数，参数设置如下：基准斜率偏差为 1，波动率为 1，最大偏差值分别设为 2 和 3。

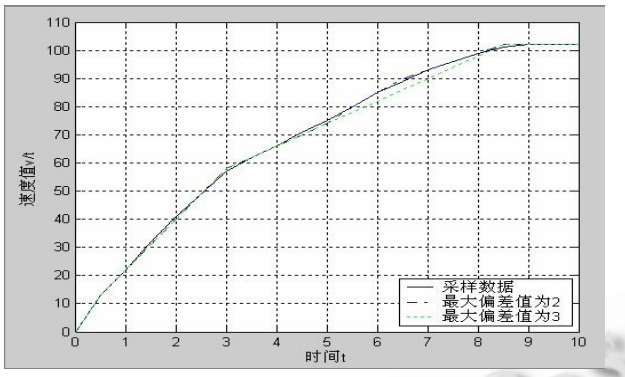

图 5 线性拟合结果

线性拟合结果如图 5 所示，由于缓存空间大小为 10，最后两个采样数据正处在待压缩状态，所以未被压缩。实线为原采样数据，虚线为拟合数据，其中与原采样数据比较贴近的拟合数据的最大偏差值为 2，另一拟合数据的最大偏差值为 3。经计算得出：最大偏差值为 2 的拟合曲线压缩率为 50%，平均误差率为 0.615%，最大误差率为 3.2%；最大偏差值为 3 的拟合曲线压缩率为 70%，平均误差率为 1.37%，最大误差率为 3.7%，可以看出，在数据变化较明显的区域，对于小于 1%精度要求，压缩率可达到 50%以上。

7 结语

本系统进行的软硬件通用性设计，采用双存储结构，通过将线性拟合存储算法与车载数据存储算法相结合的方式，实现了数据压缩以及压缩数据可移植性的功能。该数据存储系统可以为故障诊断提供有力的依据，可应用到工业网络和意外事故的黑匣子中。该设计为以后数据压缩设计提供有效的思路，并为其技术实施做了必要的铺垫。

参考文献

1 王爱勇,李荣雨,陆新建.基于关系数据库的实时数据压缩探讨.计算机应用与软件,2009,26(5):137–139.
2 潘海涛,何洁月.流程业实时数据压缩技术的研究与实现.计算机工程,2003,29(2):274–277.

（上接第 218 页）

distortion control. IEEE Trans. on Image Processing, 2002, 11(3):213–222.
3 林冬梅,马义德,张北斗.基于 PCNN 与粗集理论的细胞图像二重数字水印算法.计算机应用研究,2010,2:721–725.
4 桑军,向宏,叶春晓,胡海波.基于神经网络的实用小波域零水印技术.计算机工程,2009,4:139–141.
5 戴红亮,金文标.基于对象传播神经网络的音频水印算法.重庆邮电大学学报(自然科学版),2009,2:95–99.
6 陈海军.基于遗传优化的神经网络控制策略的研究.燕山大学,2010.
7 段明秀,基于遗传算法的模糊 RBF 神经网络设计及应用.吉首大学学报(自然科学版),2010.
8 闵松强,贺昌政,等.进化数据分组处理算法研究进展.计算机应用研究,2010,2:405–407.
9 张杨,房斌,徐传运,等.基于遗传神经网络的可信 Web 服务度量模型.计算机应用研究,2010,(1):215–217.
10 原清,贺新锋,刘湘崇.遗传算法和神经网络在导弹测试设备故障诊断中的应用研究.测试技术学报,702–706.
11 吴微.神经网络计算.北京:高等教育出版社,2003.
12 侯媛彬,杜京义,汪梅.神经网络.西安:西安电子科技大学出版社,2007.