

一种有效的 GPS 数据压缩方案

郑宇, 何大可, 张文芳, 路献辉

(西南交通大学 计算机与通信工程学院, 四川 成都 610031)

摘要: 针对静态测量中 GPS 数据的特性, 采用去除信息冗余度、局部字典编码和差分霍夫曼编码相结合的策略, 对测点采集的 GPS 数据进行信源编码。与现有 GPS 数据压缩技术相比, 该方案所需硬件资源少, 编码速度快, 压缩效率可达 25%, 并可在 8 位处理器 (如 8051) 中实施。因此, 可在不更改现有测点硬件体系的前提下实现对 GPS 数据的无损压缩, 节约测点的存储空间, 减少通信负担和传输时延。

关键词: 数据压缩; 差分霍夫曼编码; 字典编码; GPS

中图分类号: TN911.21

文献标识码: A

文章编号:

An Efficient Scheme for GPS Data Compression

Zheng Yu, He Dake, Zhang Wenfang, Lu Xianhui

(School of computer & communication engineering. Southwest Jiaotong University. Chengdu 610031. China)

Abstract: According to the property of static measurement based on GPS, the received GPS data is compressed with the help of combination of eliminating redundancy of data, dictionary coding and differential Huffman coding method. In the proposed scheme preferable compression ratio can be achieved compare with normal methods of source coding. About 25 percent GPS data can be reduced in receiving device with a 8bit microprocessor like 8051. So the storage of GPS data and the communication loads can be released in low performance receiving device without any additional hardware resource.

Keywords: Data Compression; Differential Huffman coding; Dictionary coding; GPS

当前, 基于 GPS 的测量技术在铁道上得到广泛的应用。为获得较高的测量精度, 通常利用测点和基准站相同时间段的卫星数据共同解算出测点的精确坐标 (如静态后处理技术)。由于受到地理条件和经济成本的限制, 测点多是以单片机为核心的硬件系统, 并利用通信模块将自己的 GPS 数据传送到基准站处理。所以, 庞大的数据量将会加重测点的存储负担, 导致数据传输时延的加大和通信费用的增长 (如 GPRS 以流量计费)。因此, 在测点对 GPS 数据进行无损压缩, 可大大节约测点的硬件和通信资源, 提高系统的整体性能和工作效率。

文献[1]利用差分霍夫曼编码^[2]来压缩路况信息, 文献[3]将字典编码和预测编码相结合来压缩采集数据, 但以上两种方式都要求在数据采集端有高性能处理器 (如 DSP) 作为硬件支撑, 因此现有的大部分测点必须更换硬件设备才能使用以上编码方案。文献[4,5]提出的方案只适用于 NMAE-0813 格式的 GPS 导航语句 (误差 10m 以上), 无法压缩精定位所必须的 GPS 原始数据。针对静态测点 GPS 数据的特点, 本文提出一种可在 8 位处理器 (如 8051) 上实施的数据压缩方案, 以较高的运算速度获得较大的数据压缩率。该方案性价比高, 可广泛适用于高精度测量中 GPS 数据的压缩。

1. 静态 GPS 数据的特点

当前 GPS 原始数据的存储格式较多, 但都适用于本文的压缩原理, 现以 CMC 格式^[6]的 GPS 数据为例具体说明。CMC 以二进制字节流方式输出, 其数据由若干条语句组成。如图 1 所示, 每条语句包含语句头、语句内容和校验和三部分。图 2 和图 3 是静态测量所需要

本文章由“信息安全与国家计算网格实验室”基金资助。

郑宇 (1979-), 男, 湖南衡阳人, 博士研究生。主要研究领域: 通信保密、信源编码、信息系统安全工程。

的 23 和 22 号语句的内容。

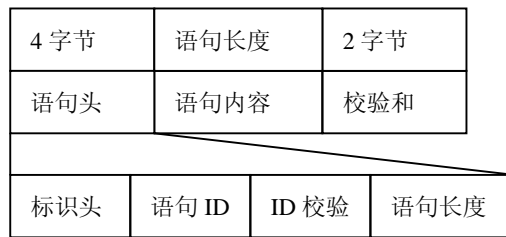


图 1 CMC 的语句格式

| | | | |
|----|------|----|---------|
| 2 | 1 | 8 | 11×n |
| 保留 | 测量块数 | 时间 | 测量块 1-n |

图 2 23 号语句的内容

| | |
|-----|----|
| 1 | 72 |
| 卫星号 | 星历 |

图 3 22 号语句的内容

| | |
|------|---------|
| 1 | 10×n |
| 测量块数 | 测量块 1-n |

图 4 精简后的 23 号语句

观察实际收到的语句可发现，静态测点输出的 GPS（CMC 格式）数据具有以下特点。

1) 23 号语句存在冗余信息，精简后的 23 号语句如图 4 所示。

— (1) 可去除 2 个没有含义的保留字节。

— (2) 由于 23 号语句是按照固定频率（如 1HZ）顺序输出，因此，测点只需存储第 1 条 23 号语句所包含的 8 字节时间，而以后同类语句的时间以第 1 条 23 号语句的时间为基准，顺序加 1 即可恢复。

— (3) 由于测量块的长度固定为 10 字节，测量块之间 1 字节的分割符可以省略。

2) 同类语句间的重复字节较多：由于静态测点的位置并未改变，不同卫星输出的定位语句虽略有偏差，但相邻的同类语句对应字段仍有很多相同字节。

3) 同一静态测点，相同跨度的不同时间段（如 7 点—8 点和 10 点—11 点）输出的 GPS 数据中，各个字节出现的概率基本稳定。图 5 反映了经度为 30 41 59.403，纬度为 104 02 47.625，海拔高度为 502.300 的一个测点在 1h（共 483KB）内各个字节出现的概率。以下编码方案均利用这个测点的数据进行分析和仿真。

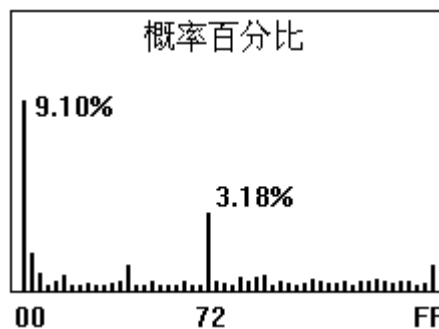


图 5 某测点 GPS 原始数据中各字节出现的概率

2.对 GPS 数据的压缩

2.1 编码和译码的总体过程

如图 6 所示，根据静态测点 GPS 数据的特性，先去除 23 号语句的冗余信息，并对两种语句做局部字典编码。随后对同类语句进行异或处理，以增加字节出现的不均匀性，提高霍夫曼编码对数据的压缩率。由于受硬件资源的限制，不管是静态还是动态霍夫曼编码^[7,8]都不可能直接在单片机中实施。因此，首先用 PC 机按照以上步骤处理待测点 1h 的数据（根据解算需要的数据量来选择时间跨度），并做静态霍夫曼编码，然后将编码结果做成一张表

格预先存放到单片机中。这样单片机在收到 GPS 数据时只需要做简单的查表运算就可完成编码工作。图 7 是基准站收到数据后的解码过程。

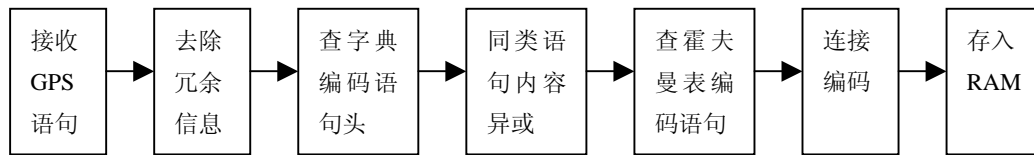


图 6 测点对 GPS 数据的编码过程

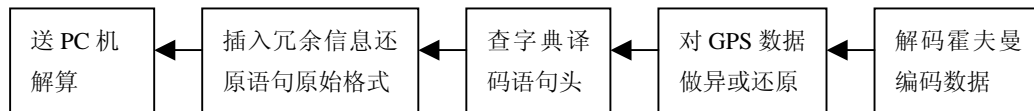


图 7 基准站对 GPS 数据的译码过程

2.2 局部字典编码

由于语句头在 GPS 数据中出现的频率很高，而其他连续字节重复出现的概率很小，因此，对仅语句头进行静态字典编码^[9]可获得较好的压缩率。实际上 22 号语句定长，语句头只有一种格式。23 号语句中测量块的数目可能是：1—12 个（GPS 接收板最多只能同时收到 12 颗卫星的数据）。因此除去冗余信息后的 23 号语句有 12 种长度可能。而第 1 条 23 号语句因包含时间信息而与后面的 23 号长度不同，也有 12 种长度可能。所以 23 号语句的前 5 字节（语句头和测量块数）只可能有 $12+12=24$ 种形式，加上 1 种 22 号语句头，共 25 种语句头出现。如表 1 所示，用 5bit 编码（ $2^5=32>25$ ）代替 23 号语句的前 5 字节和 22 号语句的头，可大大压缩 GPS 数据。该字典由 PC 机产生并预先存储在处理器的 FLASH 中。

表 1 语句头字典

| 标识头 | 语句 ID | ID 校验 | 语句长度 | 测量块数 | 字典编号 |
|-----|-------|-------|------|------|-------|
| 01 | 16 | E8 | 49 | / | 00000 |
| 01 | 17 | E9 | 0C | 01 | 00001 |
| 01 | 17 | E9 | 17 | 02 | 00010 |
| 01 | 17 | E9 | ... | | |
| 01 | 17 | E9 | 81 | 0C | 11000 |

2.3 差分霍夫曼编码

如图 8 所示，用 PC 机对解算待测点所需时间长度的数据做静态霍夫曼编码，并将生成的编码表写入单片机的 FLASH 中。同时，为提高霍夫曼编码对数据的压缩率，可先对同类语句按字节异或，以增大 0 出现的概率（相同字节异或为 0），减少数据包含的信息量。

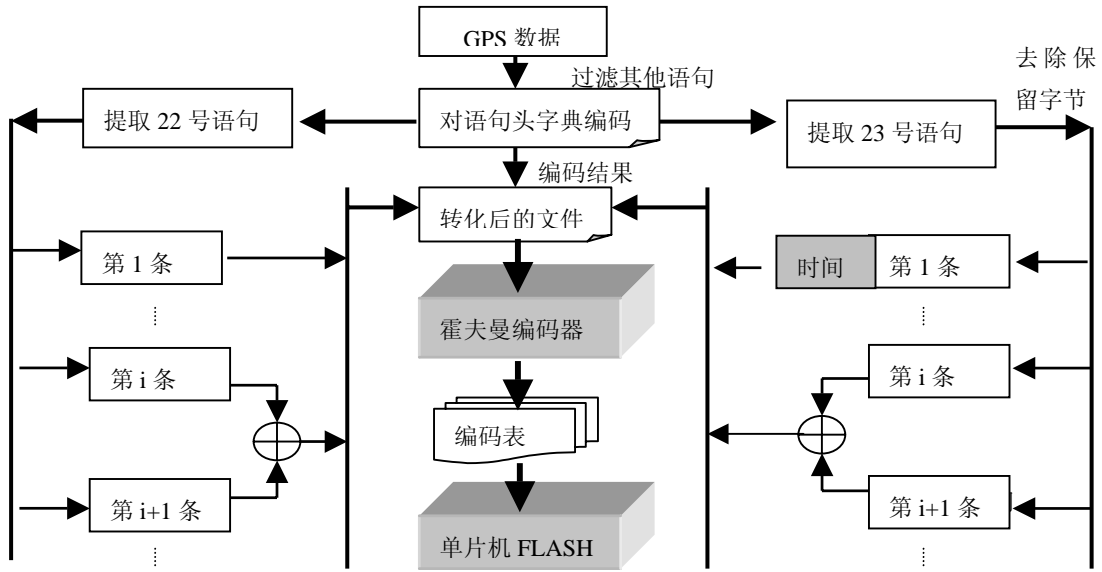


图 8 霍夫曼编码表的生成

异或过程如图 9 所示，获得的编码如表 2（按出现概率排序）所示。

```

Msg22[0], Msg23[0] → file;
for (i = 0; i < n22; i++)
{
    Msg22[i] ⊕ Msg22[i + 1] → file;
    Msg22[i] = Msg22[i + 1];
}
for (i = 0; i < n23; i++)
{
    Msg23[i] ⊕ Msg23[i + 1] → file;
    Msg23[i] = Msg23[i + 1];
}

```

表 2 霍夫曼编码表

| 码字 | 霍夫曼编码 |
|-----|----------|
| 00 | 101 |
| 72 | 011111 |
| ... | |
| 9B | 00000010 |

图 9 异或变换的过程

符号说明： n_{22}, n_{23} 为一个 GPS 文件中 22 和 23 号语句的数目； Msg_{22}, Msg_{23} 表示 22 和 23 号语句的内容； TL 数据总长度； $BlckeNum[i]$ 为第 i 条 23 号语句中测量块的数目； $len(x)$ 表示 x 的长度； p_i, n_i 表示字符出现的概率和霍夫曼编码后对应的码长。

直接进行霍夫曼编码，各个字符出现的概率如图 10 (a) 所示，可做如下计算。

平均信息熵： $H(X) = -\sum_{i=0}^{255} p_i \log p_i = 7.77$ ；平均码长： $L = \sum_{i=0}^{255} p_i n_i = 7.65$ 。

而经过异或处理后，各个字符出现的概率如图 10 (b)，根据获得编码结果可做如下计算。

平均信息熵： $H(X) = -\sum_{i=0}^{255} p_i' \log p_i' = 7.35$ ；平均码长： $L = \sum_{i=0}^{255} p_i' n_i' = 7.51$ 。

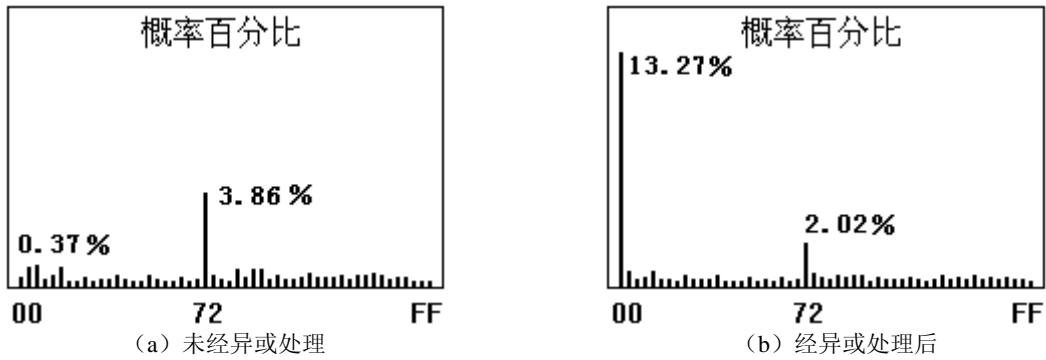


图 10 经字典编码后 GPS 数据中各字符出现的概率

3. 性能分析

3.1 压缩效率

本方案对 GPS 数据的压缩量来自三个部分：去除的冗余信息量、局部字典编码和差分霍夫曼编码对数据的压缩量。因此数据压缩率 η 应满足以下表达式：

$$\eta = \frac{Redundance + DictionaryEncode + HuffmanEncode}{TL} \times 100\%$$

$$Redundance = \sum_{i=0}^{n_{23}} (BlockNum[i] + 2) + (n_{23} - 1) \times 8;$$

$$DictionaryEncode = n_{22} \times (4 - 5/8) + n_{23} \times (5 - 5/8)$$

$$HuffmanEncode = \frac{\sum_{i=0}^{TL} (8 - \text{len}(\text{code}[i]))}{8} \approx \frac{(8 - L) \times TL}{8};$$

$$TL = n_{22} \times 73 + \sum_{i=1}^{n_{23}} \text{len}(Msg23[i]) + (n_{22} + n_{23}) \times [\text{len}(MsgHead) + \text{len}(Checksum)];$$

$$= n_{22} \times 73 + \sum_{i=1}^{n_{23}} \text{len}(Msg23[i]) + (n_{22} + n_{23}) \times 6$$

运用此方案对同一测点大小为 168KB、481KB 和 637KB 三个不同长度的 GPS 数据进行编码，在 PC 机上的仿真结果如表 3 和图 11 所示。可清楚地看到，去除的冗余信息量占 16% 左右，字典编码的压缩率在 4% 左右，而霍夫曼编码的压缩率在 5% 左右。总共压缩量维持在 25% 的水平。

表 3 压缩效率统计

| 原始数据 长度 (KB) | 22 号 语句 数量 | 23 号 语句 数量 | 冗余信息量 | | 局部字典编码 | | 差分霍夫曼编码 | | 总压 缩量 KB | 压缩 率 % |
|-----------------|------------------|------------------|----------|------------|----------|------------|----------|------------|----------------|-----------|
| | | | 大小 KB | 压 缩 比 % | 大小 KB | 压 缩 比 % | 大小 KB | 压 缩 比 % | | |
| 169 (27min) | 11 | 1614 | 28.68 | 17.07 | 6.93 | 4.13 | 7.62 | 4.54 | 43.23 | 25.73 |
| 481 (63min) | 14 | 3754 | 74.40 | 15.47 | 16.08 | 3.34 | 25.14 | 5.23 | 115.62 | 23.94 |
| 637 (105min) | 51 | 6269 | 109.34 | 17.16 | 26.95 | 4.23 | 28.39 | 4.46 | 164.68 | 25.77 |

如表 4 和图 12 所示，将此方案和另外两种方案进行比较可发现：直接对 GPS 数据做霍夫曼编码的压缩率在 5% 左右，差分霍夫曼编码压缩率可大到 10% 左右。而本文所提出方案的压缩效率基本维持在 25% 左右。

表 4 文章提出的压缩方案与其他方式的比较

| 原始数据长度 KB | 直接霍夫曼编码压缩 | | 差分霍夫曼编码 | | 文章提出的编码方案 | |
|-----------|-----------|------|---------|-------|-----------|-------|
| | 压缩量 KB | 压缩率% | 压缩量 KB | 压缩率% | 压缩量 KB | 压缩率% |
| 168 | 10.05 | 5.92 | 14.67 | 9.71 | 43.23 | 25.73 |
| 481 | 29.74 | 6.18 | 48.94 | 10.16 | 115.62 | 23.94 |
| 637 | 29.55 | 4.64 | 63.58 | 9.98 | 164.68 | 25.77 |

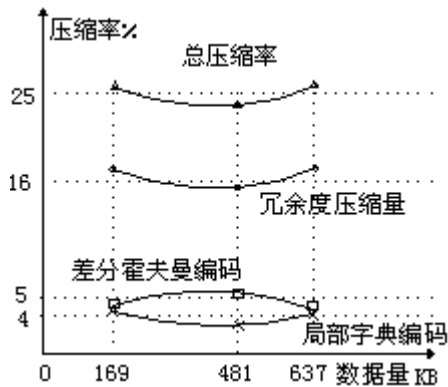


图 11 方案各个部分的压缩率

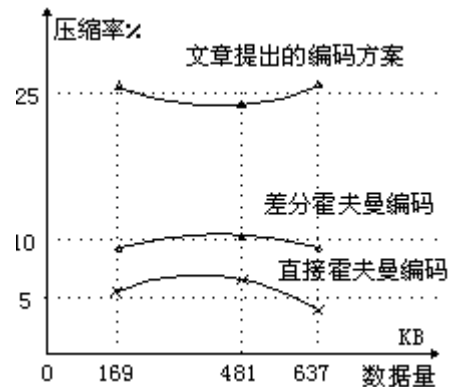


图 12 不同方案压缩率的比较

3.2 编码速度

该方案的静态字典和差分霍夫曼编码表都是由 PC 生成,并预先存储在处理器的 FLASH 中,测点处理器只需要在差分变换时对收到的 GPS 语句按字节异或,以及对其结果进行查表和移位操作,速度不低于直接在 DSP 上做动态霍夫曼编码或字典编码。而且,测点是在接收 GPS 数据(而不是将数据上传至基准站)时完成编码过程,并将编码结果缓存在扩展的 RAM 中,以备及时上传。因此,对数据编码所花的短暂时间也不在传输时间之内。

6. 结论

本文针对静态测点 GPS 数据的特性,采用去除信息冗余度、局部字典编码和差分霍夫曼编码相结合的策略,对测点采集的 GPS 数据压缩。该方案所需硬件资源少,编码速度快,压缩率基本维持在 25%左右,远远高于直接霍夫曼编码和差分霍夫曼编码的压缩率,并可在 8051 等 8 位处理器中实施。因此,可在不改动现有测点硬件系统的前提下实现对 GPS 数据的无损压缩,对提高测量系统的工作效率和整体性能有着非常积极而现实的意义。

参考文献

- [1]师延山,王珂.智能交通系统中路况信息的编码[J].北京航空航天大学学报,2000,26(3):303-306.
- [2] Gioutsos T, Whalen M. A hybrid differential encoder and non-linear filter (DEN filter)[A]. In: Computers and Communications[C], Scottsdale: IEEE Press, 1988: 470 - 473
- [3]南金瑞,王建群,孙朋春.车辆数据采集系统中数据压缩技术的研究[J].北京理工大学学报,2003,23(1):46-49
- [4]杨宏业,张跃.GPS定位数据压缩算法的设计与实现[J].电子技术应用,2002,(12):29-32
- [5]谢小娟.GPS数据采集系统[J].微机发展,1997,(2):55-56
- [6]Marconi Company. Allstar Users Manual [Z]. Canada: Supersedes Publication, 1998.11: 5.1-5.19(技术手册)

[7]Bernard Sklar. Digital Communications: Fundamentals and Applications [M]. BeiJing: Publishing House of Electronics Industry, 2002: 609-663

[8]靳蕃.信息论与编码方法在计算机和通信中的应用[M].成都：西南交通大学出版社,1990:99-113

[9]Sorer J A. Data compression: Methods and theory [M]. New York: Computer Science Press,1988: 81-121.