



[12] 发明专利申请公开说明书

[21] 申请号 01808587.3

[43] 公开日 2003 年 6 月 25 日

[11] 公开号 CN 1426629A

[22] 申请日 2001.2.22 [21] 申请号 01808587.3

[30] 优先权

[32] 2000. 2. 25 [33] US [31] 09/513,309

[86] 国际申请 PCT/US01/05722 2001. 2. 22

[87] 国际公布 WO01/63772 英 2001. 8. 30

[85] 进入国家阶段日期 2002. 10. 25

[71] 申请人 物理光学公司

地址 美国加利福尼亚州

[72] 发明人 伊戈尔·V·捷尔诺夫斯基

亚历山大·A·德维沃耶

约瑟夫·罗腾贝格 弗雷迪·林

[74] 专利代理机构 中科专利商标代理有限责任公
司

代理人 王 玮

权利要求书 3 页 说明书 10 页 附图 3 页

[54] 发明名称 使用多个编码器的优化无损压缩的方法和装置

[57] 摘要

一种数据流的无损压缩方法，首先包括使用多个无损编码器压缩数据流的一个测试部分(30)。一旦压缩了测试部分，该方法确定与每个无损编码器(32)相关的性能特性。然后，该方法根据性能特性选择一个无损编码器，并且利用选择的编码器编码数据流的第一部分。接下来，对于数据流的另一个测试部分和第二部分，重复使用、确定、选择和编码步骤。应当注意，重复步骤可以包括选择一个不同的无损编码器。

	算术	LEMPLE-ZIV	HUFFMAN	...	n
8	无损编码器 (Ar,8)	无损编码器 (LZ,8)			无损编码器 (n,8)
10	无损编码器 (Ar,10)				
...					
m	无损编码器 (Ar,M)				无损编码器 (n,m)

- 5 1.一种数据流的无损压缩方法，该方法包括步骤：
提供多个不同类型的无损编码器；
选择一个无损编码器压缩数据流；和
用选择的无损编码器编码数据流。
- 2.根据权利要求 1 所述的方法，进一步包括步骤：在所述选择步骤
10 之前，利用每个无损编码器独立地压缩数据流的至少一个部分；和
其中所述选择步骤是根据所述压缩步骤执行的。
- 3.根据权利要求 1 所述的方法，其中所述选择步骤根据与所述压缩
步骤相关的性能特性进行。
- 4.根据权利要求 3 所述的方法，其中性能特性包括一个对应的无损
15 编码器的压缩率和所述压缩步骤的持续时间中的至少一个。
- 5.根据权利要求 1 所述的方法，其中至少一个无损编码器使用统计
建模。
- 6.根据权利要求 5 所述的方法，其中至少另一个无损编码器使用基
于字典的建模。
- 20 7.根据权利要求 2 所述的方法，其中无损编码器并行地执行所述压
缩步骤。
- 8.根据权利要求 2 所述的方法，其中无损编码器顺序地执行所述压
缩步骤。
- 9.根据权利要求 1 所述的方法，其中无损编码器是部分地通过所述
25 编码步骤中使用的每字比特的数量定义的。
- 10.一种数据流的无损压缩方法，该方法包括步骤：
使用多个不同类型的无损编码器压缩数据流的一个测试部分；
响应所述使用步骤确定与每个无损编码器相关的性能特性；
根据所述确定步骤选择一个无损编码器；
30 利用选择的编码器编码数据流的第一部分；和
对数据流的另一个测试部分和第二部分重复所述使用、确定、选择

和编码步骤。

11.根据权利要求 10 所述的方法，其中所述重复步骤包括选择一个不同的无损编码器。

5 12.根据权利要求 10 所述的方法，其中无损编码器并行地执行所述使用步骤。

13.根据权利要求 10 所述的方法，其中无损编码器顺序地执行所述使用步骤。

14.根据权利要求 10 所述的方法，其中在所述编码步骤中，每个无损编码器使用（1）一种压缩技术，和（2）每字的比特数，比特数是通过所述选择步骤确定的。

15 15.根据权利要求 14 所述的方法，其中压缩技术是算术编码、Huffman 编码和 LZ 编码中的一种。

16.根据权利要求 10 所述的方法，其中数据流包括来自多个不同信源的数据。

17.根据权利要求 10 所述的方法，其中性能特性包括一个对应无损编码器的压缩率和所述使用步骤的持续时间中的至少一个。

18.一种无损数据压缩装置，该装置包括：

一个接收数据流的接口；

多个不同类型的无损编码器；

20 一个处理器；和

其中每个所述无损编码器独立地压缩数据流的一个测试部分，和对此响应，所述处理器（1）确定与每个所述无损编码器相关的性能特性，和（2）根据所述性能特性，选择一个所述无损编码器以编码数据流的至少第一部分。

25 19.根据权利要求 18 所述的方法，其中性能特性包括压缩率和测试部分的压缩持续时间中的至少一个。

20.根据权利要求 18 所述的装置，其中所述编码器包括多个处理器，并且每个所述无损编码器对应于一个处理器，并且其中所述无损编码器并行地压缩同一测试部分。

30 21.根据权利要求 18 所述的方法，其中数据流包括来自多个不同信

源的数据。

22.一种无损数据压缩装置，该装置包括：

包括一个接收数据流的接口、多个不同类型的无损编码器和一个处理器的编码器，其中每个所述无损编码器独立地压缩数据流的一个测试部分，和对此响应，所述处理器（1）确定与每个所述无损编码器相关的性能特性，和（2）根据所述性能特性选择一个所述无损编码器，以编码数据流的至少第一部分；和

一个接收和解压缩数据流的所述编码的第一部分的解码器。

23.一种数据流的无损压缩方法，该方法包括步骤：

使用多个不同类型的无损编码器压缩数据流的一个测试部分；
响应所述使用步骤，确定与每个无损编码器相关的性能特性；
根据所述确定步骤，选择一个无损编码器；

利用选择的编码器编码数据流的第一部分；和

对于数据流的另一个测试部分和第二部分重复所述使用、确定、选择和编码步骤，其中所述重复步骤包括选择一个不同的无损编码器；
将编码的第一部分经过通信媒体发送到解码器；和
对编码的第一部分解压缩。

24.根据权利要求 23 所述的装置，其中媒体是互联网。

25.根据权利要求 23 所述的装置，其中性能特性包括对应无损编码器的压缩率和所述使用步骤的持续时间中的至少一个。

26.根据权利要求 23 所述的装置，其中数据流包括来自多个不同信源的数据。

使用多个编码器的优化无损压缩的方法和装置

5

技术领域

本发明涉及数据压缩技术，更具体地讲，涉及一种在不同类型的无损压缩编码器中进行选择以优化系统性能的方法和装置。

10 背景技术

进行数据压缩是为了使用于存储或发送信息的比特数最少，并且包含着一系列的软件和硬件压缩技术。尤其是，根据要压缩的数据类型和任何数量的其它因素，特定的压缩技术在压缩率和编码速度方面可以提供显著优越的性能。

15

数据压缩一般包括提取一个码元或短语流，并且将它们转换成比原始数据小的（在比特长度上）代码。已知的压缩技术和算法可以分为包括有损和无损的两个主要系列。有损数据压缩可以用于大大提高数据压缩率；但是，加大压缩是以精确性上的一定损失为代价的。结果，有损压缩一般是在一些数据损失是可以接受的情况下进行的。例如，在应用到数字化话音信号和图示图像时可以有效地使用有损压缩。另一方面，无损压缩是一种利用设计在压缩/解压缩循环之后产生输入数据流的准确复制品的技术的数据压缩系列。在存储数据库记录、字处理文件等其中信息损失是绝对不可接受的情况下需要这种类型的压缩。本发明致力于无损数据压缩。

20

25

30

一些无损压缩算法使用了信息理论，以便在给出了一个给定码元集的概率表时，产生可变长度代码。输出一个特定码元或码元集（即，消息）的某种代码的决定是根据一个模型作出的。这个模型是一个用于处理输入的消息，并且响应处理结果，确定输出哪些代码的规则集。算法或程序使用模型分析码元（例如，确定与码元相关的概率），然后根据该处理输出适当的代码。有许多种方式建立数据模型，所有这些方式可以

使用相同的编码技术产生它们的输出。一般情况下，为了有效地压缩数据，应当选择一个预测具有高概率的码元或短语的模型，因为具有高概率的码元或消息具有低的信息内容，因而需要较少的比特进行编码。下一个步骤是利用一个特定无损编码器对码元编码。

5 按照惯例，可以根据无损压缩编码器是执行统计建模还是基于字典的建模对它们进行分组。统计建模利用字符出现的概率一次读出和编码一个单一的码元，而基于字典的建模使用一个单一的代码替代码元串。尤其是，在基于字典的建模中，由于显著地减少了与编码每个码元相关的问题，模型要比在基于统计的建模更为重要。

10 统计数据压缩的一种形式被称为 Shannon-Fano (S-F) 编码。开发 S-F 编码是为了提供可变长度比特编码，以便能够用包含消息或码元的信息的准确（或十分近似于）的比特数编码码元。S-F 编码依赖于知道一个消息中每个码元的出现概率。在确定了概率之后，构造一个代码表，表中的每个代码具有不同的比特数（最好是，低概率的码元具有更多的比
15 特数）。这种编码技术的一个问题是，它建立了具有整数比特数的可变长度代码，即便是要编码的信息可能需要非整数的比特数。

 另一种类型的编码，Huffman 编码，与 S-F 编码的相同之处在于它建立了具有整数比特数的可变长度代码，但是它使用了一种完全不同的算法。S-F 和 Huffman 编码在性能上总体上接近，但是，已经证明，Huffman
20 编码总是至少等于 S-F 编码的效率，因而它更好一些，特别是由于两种算法采用了相同的处理能力。尽管 Huffman 相对比较容易实现，并且对于编码和解码都比较经济，但是，由于它像 S-F 编码一样对每个代码使用了整数的比特数，因而它的效率不高。如果确定了一个特定码元具有 1.5 比特的信息内容（即，熵），Huffman 编码器将产生一个具有比特数为一
25 或二比特的代码。如果一种统计方法可以将 90% 的概率赋予一个给定码元，那么最优的代码长度应当是 0.15 比特；但是，Huffman 或 S-F 编码可能将一个一比特的代码赋予这个码元，这比需要的大六倍。

 考虑到这种与使用整数比特数有关的问题，开发了算术编码。算术编码用一个单一的浮点输出数替代一个输入码元流，并且绕过用一个专
30 用代码替代一个输入码元的步骤。由于算术代码并不限于仅在码元的概

率是二分之一的整数幂时（大多数情况下不是这样）是最优的，它达到了要编码的码元的理论熵，因而使任何已知源的压缩效率最高。也就是说，如果一个给定字符的熵是 1.5 比特，那么算术编码使用 1.5 比特编码该码元，这对于 Huffman 和 Shannon-Fano 编码是不可能的。尽管算术编码非常有效，但是就 CPU 能力和存储器而言，它消耗相当大量的计算资源。这是由于必须建立要求大量存储容量的精密复杂的模型，并且算法本身需要大量的计算操作的事实。

在一种称为替换或基于字典的编码的上述无损编码类型的替代编码中，基于字典的压缩算法参照这些短语的以前出现来替代一个数据流中特定短语的出现（即，字节组）。与上述通过将码元编码成使用比原始码元少的比特的比特串完成压缩的系统不同，基于字典的算法不编码单一的码元。基于字典的压缩技术将可变长度码元串编码成单一的“标记（token）”。正是这些标记形成了对一个短语字典的索引。由于标记比它们替代的短语小，因而产生了压缩。两类主要的基于字典的压缩方案被称为 Lempel-Ziv 系列压缩编码器的 LZ77 和 LZ78 压缩算法。尤其是，基于字典的编码被广泛地用于桌面通用压缩，并且已经被 Compuserve 信息服务用于编码位映像图示图像。例如，GIF 格式使用了一种 LZW 变量压缩重复的序列和屏幕图象。尽管基于字典的压缩技术是非常流行的压缩形式，但是这种算法的缺点在于，需要更为复杂的数据结构来处理字典。

总之，随着象互联网之类的通信媒体的发展，通过在特定领域中提供特定优点的不同压缩算法，数据压缩将继续对有效的数据通信具有非常重要的意义。本领域中存在着包括上述以及其它正在使用的多种类型的数据压缩方法。此外，不断地开发出许多有关每种已知类型的压缩算法的变化和许多改进。此外，根据与系统和要压缩的数据类型有关的任何数量的因素，可以用每一种方法来提供最优的数据编码。

由于根据包括要编码的数据在内的各种操作因素，不同的已知编码技术提供了独特的利益，因而希望有一种能够有选择地用不同类型的编码器编码数据的无损压缩系统。特别是电信工业需要一种能够实现不同类型编码器的系统，尤其是在从多个提供不同类型的未知数据的信源接

收输入数据时，即，在数据流的不同部分要优选地用不同编码技术压缩时。

发明内容

5 本发明的目的涉及一种确定多个嵌入的编码方案中哪些将最佳地压缩输入数据流的不同部分的方法和装置。设计了优选实施例的方法，以适应以具有不同的信息分组（例如，从未知的信源到编码器），每个信息分组都具有不同的相关统计表为特征的数据流。

10 根据优选实施例的第一方面，一种数据流的无损压缩方法包括：提供多个无损编码器。然后，该方法包括选择一个无损编码器压缩数据流，然后用选择的无损编码器编码数据流。

15 根据优选实施例的另一方面，一种数据流的无损压缩方法包括：使用多个无损编码器压缩数据流的一个测试部分。一旦压缩了测试部分，该方法确定与每个无损编码器相关的性能特性。然后，该方法包括根据确定步骤选择一个无损编码器，并且利用选择的编码器编码数据流的第一部分。接下来，该方法包括对数据流的另一个测试部分和第二部分重复进行使用、确定、选择和编码步骤。应当注意，重复步骤可以包括选择一个不同的无损编码器。

20 根据优选实施例的再一个方面，每个无损编码器在编码步骤中，使用（1）一种压缩技术，和（2）选择步骤确定的每个字的多个比特。并且，压缩技术是算术编码、Huffman 编码和 LZ 编码中的一种。

25 根据优选实施例的再一个方面，一种无损数据压缩的装置包括：一个接收数据流的接口。此外，该装置包括多个无损编码器和一个处理器。在操作中，每个无损编码器独立地压缩数据流中的一个测试部分，并且，作为响应，处理器确定与每个无损编码器相关的性能特性，然后，根据性能特性，选择一个无损编码器，以编码数据流的至少第一部分。

30 根据优选实施例的又一个方面，性能特性包括一个对应无损编码器的压缩率和测试部分的压缩持续时间中的至少一个。此外，编码器包括多个处理器，并且每个无损编码器对应于一个处理器，和其中无损编码器并行地压缩同一个测试部分。

通过以下的详细说明和附图，熟悉本领域的技术人员将会了解本发明的这些和其它目的、有点和特征。但是，应当知道，尽管详细说明和附图指出了本发明的优选实施例，但它们是以示例的方式而不是限制的方式给出的。在本发明的范围内可以进行多种改变和修改，而不脱离本
5 发明的精神，并且本发明包括所有这些修改。

附图说明

在附图中示出了本发明的优选示例实施例，在所有附图中相同的参考号代表相同的部件，其中：

- 10 图 1 是表示优选实施例的一种方法的一般操作的流程图；
图 1A 是表示图 1 所示方法中使用的一个无损编码器阵列的图表；
图 2 是表示优选实施例的一个编码 / 解码系统的总体方框图；和
图 3 是表示当图 2 中所示的系统编码 / 解码数据流时的数据流的示意图。

15

具体实施方式

- 参考图 1，方法 10 包括：在步骤 12 初始化和启动之后，在步骤 14 将数据输入到系统。在步骤 14 输入的数据可以是同步数据或异步数据。应当注意，数据流可以是从象监视一个对象的温度，压力等（例如，在
20 军事应用中收集的数据），并且连续地将读数发送到优选实施例的系统编码器（下面将说明）的传感器之类的未指明的信源接收的。未指明的数据必须意味着与数据相关的统计是随机的，因而与根据有关数据的统计的知识利用一个单一类型的编码器进行压缩的已知系统不同，优选实施例能够有效地编码一个由不同类型数据构成的数据流。其中这种类型的
25 随机数据可以来源于多个信源的其他类型的应用包括医院监视应用、化工厂、核电站、以及其它应用。

- 随着数据连续地输入到系统，它被发送到一个有关通信数据块的一个分割部分，在这里，方法 10 在步骤 16 通过分割数据或将数据分成帧而处理数据，以利于它的进一步通信。有关通信数据块的分割是由方法
30 10 以惯用方式实现的。接下来，在步骤 18，对数据进行预处理，预处理

可以包括产生一个指示与在步骤 16 中分成帧的数据相关的统计的直方图。

一旦在步骤 18 中预处理了数据，方法 10 在步骤 20 加上进一步处理的标识数据流中的数据比特所需的同步和首部代码。当步骤 20 完成时，
5 将数据发送到提供无损压缩的多个编码器。更具体地讲，在步骤 22，方法 10 用多个无损编码器编码数据流的一个测试部分，并且确定与每个编码器相关的系统性能标准。在图 1A 的图表 30 的 32 示出了步骤 22 中用于编码数据的这个部分的编码器。图表的各列指出了各种不同类型的无损编码技术 / 算法，它们可以包括 Huffman 编码、算术编码、Lempel-Ziv
10 编码、以及这些和其它编码技术的变化。应当注意，方法也将这些编码技术的输出与没有编码/压缩的数据流比较，因为在某些情况下未压缩的数据可能是最佳的。

一般而言，列包括无损编码技术。行包括可以用于编码数据的每个字的比特数的规定， $\text{bpw } 1\text{-}m$ 。例如，可以将与接口相关的每字的比特设置
15 置为 8 比特，10 比特，等等。结果，在步骤 22，方法 10 用 $n \times m$ 个无损编码器编码数据的一部分。最好是将步骤 22 执行一个测试时间周期，或一定的数据量，以确定哪些无损编码器最先达到了最佳系统性能。然后，进行数据编码（在以后说明）。

最好是并行地进行步骤 22 中编码器 32 执行的测试压缩，以快速地
20 编译对应于每个无损编码器的数据。由于计算能力已经如此地廉价，以致于利益（就编码速度而言）大大超过了价格的事实，测试数据的并行编码是可能的。否则，在一个替代实施例中，图 1A 中所示的每个编码器 32 可以在一个指定时间周期中顺序地编码测试数据，产生对应的性能数据。尽管这不是优选的，但在计算能力很宝贵时可以执行顺序测试。

25 以下说明表 1，表 1 示出了在步骤 22 中产生的九种不同无损编码器（三个不同字长 \times 三种不同编码技术）的性能标准。首先注意，将输入比特率设置在一个预定值，同时可以根据有关使用的无损编码器的反馈信息连续地更新输出比特率，尽管最好是根据使用的传输媒体设置输出比特率。最好使输出比特率尽可能地小。如表 1 中所示，在压缩了一定
30 量的测试数据之后，确定以字节表示的规定的输入速度（千比特/秒）和

输出速度（千比特/秒）的输出文件长度、压缩率、和编码时间。例如，对于具有 304,180,992 字节的输入文件和当使用 8 比特每字时，Huffman 编码达到了 1.8272 的压缩率，Lempel-Ziv 编码达到了 2.505 的压缩率，算术编码达到 2.7724 的压缩率。此外，这些算法的每个的编码测试数据的时间分别是 128，522 秒，和 1,582 秒。一旦产生了每个无损编码器的性能标准，方法 10 执行步骤 24，选择一个编码器进行编码、压缩数据一段预定的时间，或一个特定的数据量。

应当注意，在步骤 24 进行的选择一般不仅仅根据实现的压缩率，而是根据总的处理时间和压缩率性能特性的组合进行选择。例如，在表 1 中，对于每字 8 比特，算术编码达到了 2.7724 的压缩率，这大于 Lempel-Ziv 编码达到的 2.505 的压缩率。但是，算术编码使用的编码时间比 Lempel-Ziv 无损编码器的长 15 分钟。在这种情况下，方法 10 在步骤 24 中很可能选择 Lempel-Ziv 编码器。但是，如果所有 $n \times m$ 个无损编码器达到的性能都不满足一个最低水平，那么方法 10 可以决定发送不压缩的数据。除了其它因素之外，这种决定取决于用户的要求。

表 1 中指出的输入时钟速率依赖于数据通过其发送的媒体（例如，互联网），和执行的编码算法的类型。时间性能标准是根据下面的公式产生的：

$$t_{overall} = t_c + t_{processing} \quad (\text{公式 1})$$

在公式 1 中， $t_{processing}$ 包括与压缩数据、系统延迟等相关的时间周期。此外， t_c 是发送数据的时间，和等于文件长度除以压缩率和输出速度，即，比特率，并且反映了压缩数据取得的时间节省。压缩率（CR）等于输入文件长度除以输出文件长度。

表 1

		Huffman	Lempel-Ziv	算术
8 比特每字	输出文件长度 (字节)	166,470,896	193,662,252	184,718,156
	压缩率	1.8272	1.5707	1.6467
	时间 (秒)	128	265	2,742
	输入速度 (千比特/秒)	19,014	9,180	887
	输出速度 (千比特/秒)	10,406	5,845	538
10 比特每字	输出文件长度 (字节)	121,428,144	177,869,924	174,151,624
	压缩率	2.505	1.7101	1.7466
	时间 (秒)	522	382	330
	输入速度 (千比特/秒)	4,658	6,368	7,371
	输出速度 (千比特/秒)	1,859	3,724	4,220
12 比特每字	输出文件长度 (字节)	109,716,096	142,998,504	125,961,032
	压缩率	2.7724	2.1272	2.4149
	时间 (秒)	1,582	1,438	1,505
	输入速度 (千比特/秒)	1,537	1,693	1,617
	输出速度 (千比特/秒)	555	796	670

一旦在步骤 24 中选择了编码器 (n, m) 32 (图 1A), 方法 10 用选择的编码器编码数据, 优选是, 一个预定的时间。然后, 程序转到步骤 22, 编码数据流的一个新的测试部分, 并且为数据流的下一个部分选择一个最佳编码器。这个操作可能需要提供图表 30 中所示的一个不同的无损编码器。

转到图 2, 一个用于执行方法 10 的系统 40 包括一个具有一个输入接口 42 的编码器 41, 输入接口 42 包括一个时钟输入端 C1 和一个用于接收同步或异步数据流 43 的数据输入端 D1。接口 42 经过输入和输出数据-控制-同步输入/输出线 44 耦合到一个数字信号处理 (DSP) 芯片 46。应当注意, DSP 46 优选执行图 1 中所示的方法 10 中的步骤 16 和 18, 以将数据分成帧, 并且将它准备用于压缩。DSP 46 的输出端经过一个将分成帧的数据通信到计算机的 PCI 总线 48 耦合到计算机 50。计算机 50 优选将适当的首部代码加到数据流, 以指示不同的数据分组, 并且操作以利用图表 30 中所示的每个无损编码器编码/压缩测试数据。如上所述, 计算机 50 可以包括多个处理器, 每个处理器能够为从图 1A 中的表格 30 实现的一个对应无损编码器编码/压缩数据。作为替代, 可以使用一个单一的计算机 50 在一个预定时间周期中以顺序的方式实现每个无损编码器 32 的测试压缩。

也可以用计算机 50 将首部代码添加到数据, 以保证文件正确地解压缩。然后将压缩数据经过 PCI 总线 48 发送到 DSP 芯片 46, 以根据提供的特定通信系统分割数据。这种处理可以包括通过插入空数据块和/或删除现有数据块来缓存数据。然后, 可以加入特殊的同步代码, 并且沿输入/输出线将数据流发送回到接口 42。特殊接口代码设置包括指定比特数/字, 每帧的字数, 同步代码, 检核和, 等等。然后, 接口 43 在线路 D2 上输出数据流, 从而使它能够通过互联网之类的媒体 52 发送。最好是由操作人员并且根据提供的媒体 52 的类型来设置输出时钟速率 C2。

接下来, 系统 40 的解码器 53 包括一个具有用于以对应于从接口 42 输出的时钟速率 C2 的时钟速率 C3 接收来自媒体 52 的压缩数据的数据输入端 D3 的接口 54。应当注意, 时钟 C2 和 C3 是任选的。接口 54 经过数据-控制-同步线路 56 发送压缩数据流, 同时删除编码器 41 加入的

POC 同步代码。然后，DSP 芯片 58 检测首部代码，并且从数据流删除空数据块。然后将 DSP 芯片 58 处理的数据经过 PCI 总线 60 发送到计算机 62。计算机 62 解压缩数据，并且最好执行惯用的控制和数检验（CSC）比较技术。也可以通过计算机 62 实现附加的错误检测或纠错编码器。里德-索洛蒙纠错编码器是通信网的标准，并且最好包括它。应当注意，除
5 已说明的优选实现之外，上述处理操作也可以由计算机 50，62，或 DSP 芯片 46，48 执行。然后，将压缩数据以时钟速率 $C4=C1$ ，发送回到接口 54，并且发送到数据线路 D4 上。

图 3 中示意地示出了对于一个遥测数据流的图 1 中说明的并且由图
10 2 中所示装置执行的方法的各个步骤的代表。在图 3 右侧，标注 A 的箭头指示编码过程，而沿图 3 中所示的数据的左侧的箭头 B 指出了解码过程。更具体地讲，数据流 43 被输入到接口 42（图 2），然后，最好是通过 DSP 46，以预定的方式分帧成为数据包部分 64，66（优选是数千字节，例如，两个 8k 部分）。接下来，将部分 64，66 压缩成，例如，一个 4.5k
15 数据包 68，和一个 4.3k 数据包 70。然后，将首部 73，75 加到数据包的数据块（带有上述的接口信息），以分别建立数据包 72，74。然后，将数据包缓存以构造缓存和压缩数据包 76，如果需要建立数据流 78，可以再分割数据包 76。然后，通过 DSP 芯片 46 加入 POC 同步代码，并且可以把新数据流 80 经过，例如，互联网 52（图 2），发送到解码器 53，在
20 这里如上所述进行解码。

可以在本发明的范围内进行许多改变和修改，而不脱离本发明的精神。所附权利要求将使落入本发明范围内的其它改变和修改变得显而易见。

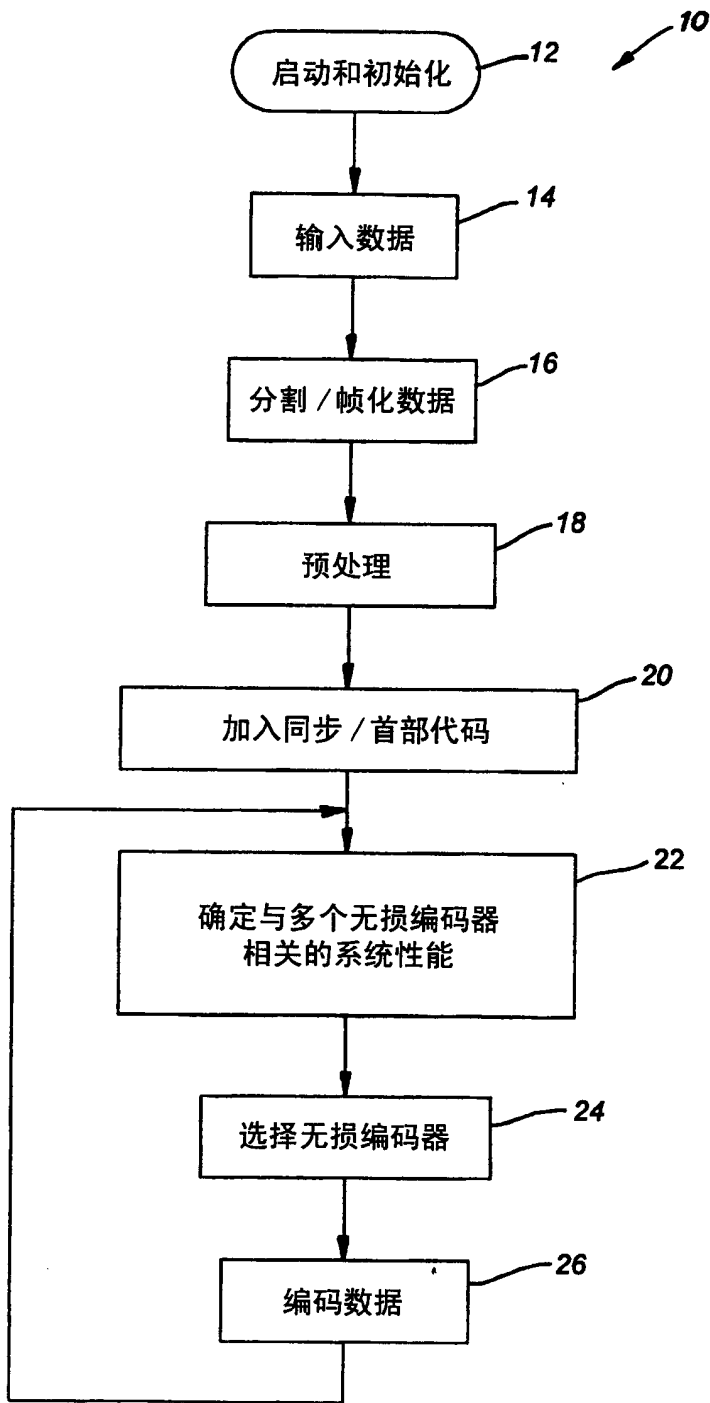


图 1

	算术	LEMPEL-ZIV	HUFFMAN	...	n
8	无损编码器 (Ar,8)	无损编码器 (LZ,8)			无损编码器 (n,8)
10	无损编码器 (Ar,10)				
...					
m	无损编码器 (Ar,M)				无损编码器 (n,m)

图 1A

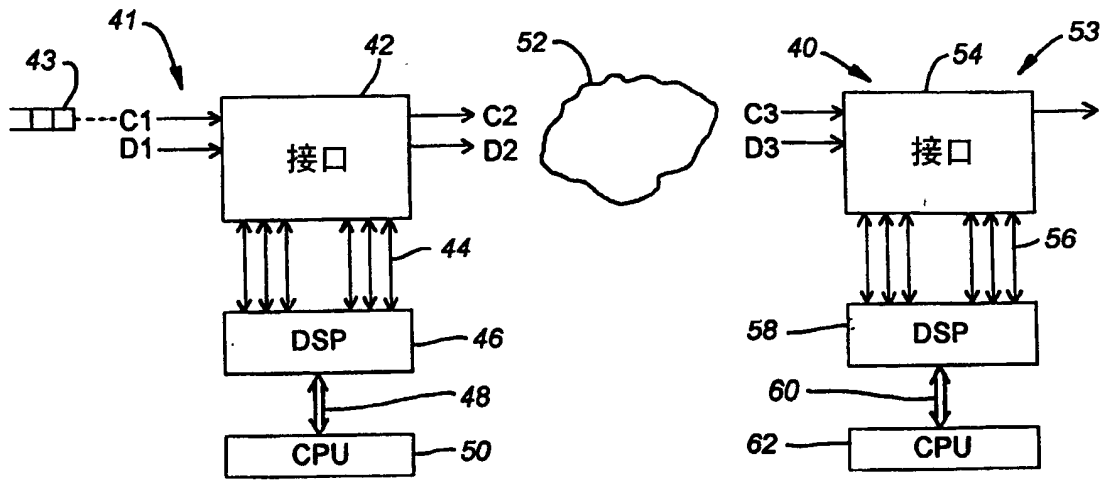


图 2

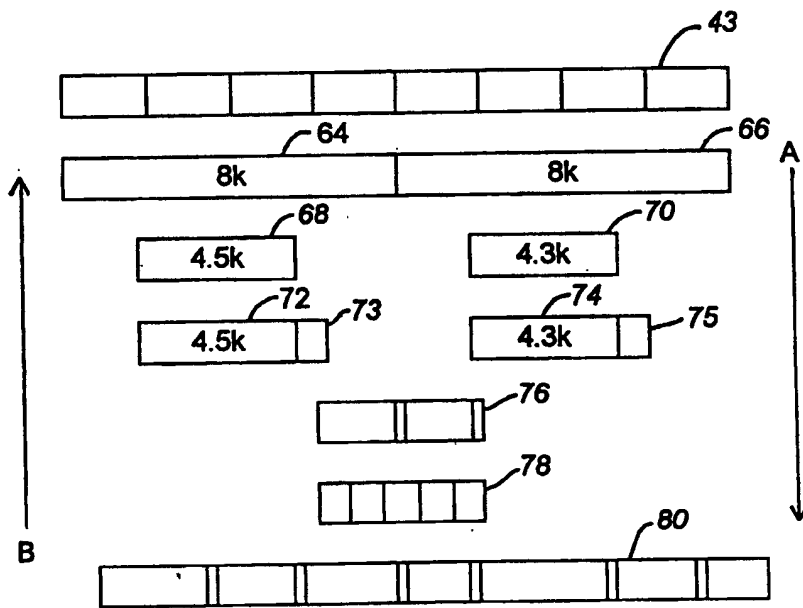


图 3