(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105409129 B
(45)授权公告日 2018.11.16

(21)申请号 201480024696.X
(22)申请日 2014.03.01
(65)同一申请的已公布的文献号
申请公布号 CN 105409129 A
(43)申请公布日 2016.03.16
(30)优先权数据
1303658.7 2013.03.01 GB
(85)PCT国际申请进入国家阶段日
2015.10.30
(86)PCT国际申请的申请数据
PCT/EP2014/000531 2014.03.01
(87)PCT国际申请的公布数据
WO2014/131528 EN 2014.09.04

(73)专利权人 古如罗技微系统公司
地址 芬兰土尔库市里南路34号,邮编20100
(72)发明人 奥西·卡雷沃
(74)专利代理机构 北京英赛嘉华知识产权代理有限责任公司 11204
代理人 王达佐 王艳春

(51)Int.Cl.
H03M 7/30(2006.01)
(56)对比文件
CN 1636405 A,2005.07.06, (续)
审查员 葛运滨

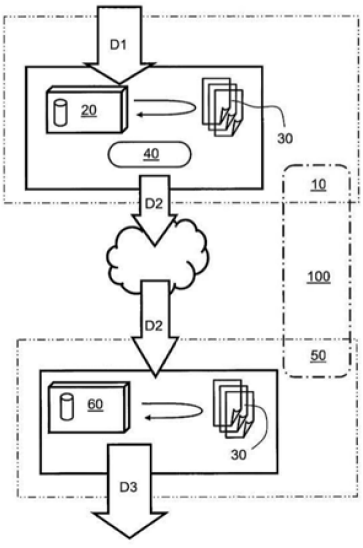
权利要求书4页 说明书18页 附图1页

(54)发明名称

编码器设备、解码器设备和方法

(57)摘要

提供了一种编码器(10),用于对提供到该编码器(10)的数据(D1)编码以产生相应编码数据(D2),其中编码器(10)包括数据处理装置(20),用于将一个或多个编码过程应用于数据(D1)以产生编码数据(D2)。数据处理装置(20)可操作用于,如果该数据(D1)不是以数值符号表示的,则至少部分地以数值符号的集合表示该数据(D1)。数据处理装置(20)可操作用于产生中间数据(40),其中以原始值表示数值符号,以修正值表示至少一个符号,该修正值具有由连续统运算器产生的一个或多个连续统符号,其中该一个或多个连续统符号修正前面的符号值以适应扩展符号范围。数据处理装置(20)可操作用于处理中间数据(40)以产生编码数据(D2)。提供了一种解码器(50),用于对提供到该解码器(50)的编码数据(D2)解码以产生相应解码数据(D3),其中解码器(50)包括数据处理装置(60),用于将一个或多个解码过程应用于编码数据(D2)以产生解码数据(D3)。数据处理装置(60)可操作用于处理编码数据(D2)以产生中间数据(40)。数据处理装置(60)可操作用于处理中间数据(40)以解码中间数据,其中在中间数据中,数值符号由输出符号及至少由一个修正输出符号表示,该修正输出符号具有一个或多个连续统符号,该一个或多个连续统符号随后由逆连续统运算器解码,其中所述一个或多个连续统符号修正该修正输出符号的值以适应扩展符号范围。数据处理装置(60)可操作用于 (续)



CN 105409129 B

[转续页]

[接上页]

(56)对比文件

CN 1875634 A,2006.12.06,

CN 1983432 A,2007.06.20,

US 5764167 A,1998.06.09,

(57)摘要

变换和/或转换处理后的中间数据以通过符号集合来表示解码数据(D3)。该编码器(10)和解码器(50)合称编解码器(100),可操作用于处理表示以下项目的数据:捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据,但编解码器(100)可处理的数据不限于此。

1.一种编码器(10),用于对提供到该编码器(10)的数据(D1)编码以产生相应编码数据(D2),其中编码器(10)包括数据处理装置(20),用于将一个或多个编码过程应用于数据(D1)以产生编码数据(D2),其特征在于:

(a)数据处理装置(20)可操作用于,如果该数据(D1)不是以数值符号表示的,则至少部分地以数值符号的集合表示该数据(D1);

(b)数据处理装置(20)可操作用于产生中间数据(40),其中以原始值表示数值符号,以修正值表示至少一个符号,该修正值具有由连续统运算器产生的一个或多个连续统符号,其中该一个或多个连续统符号修正前面的符号值以适应扩展符号范围;以及

(c)数据处理装置(20)可操作用于处理中间数据(40)以产生编码数据(D2)。

2.根据权利要求1所述的编码器(10),其特征在于,数据处理装置(20)可操作用于基于计算出的符号在多个部分或片段中的出现概率及其压缩效率,将数据(D1)分割为所述多个部分或片段。

3.根据权利要求1所述的编码器(10),其特征在于,针对数据(D1)中的符号,使用连续统运算器产生中间数据(40),所述中间数据(40)是通过一种或多种变换方法和/或通过一个或多个转换表(LUT)对数据(D1)中的符号进行变换而产生的。

4.根据权利要求3所述的编码器(10),其特征在于,通过多个数据流提供编码数据(D2)其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。

5.根据权利要求4所述的编码器(10),其特征在于,指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从编码器(10)的备选源得到指示所述一个或多个转换表(LUT)的信息。

6.根据权利要求1所述的编码器(10),其特征在于,编码器(10)可操作用于处理的数据(D1)包括以下中的至少一种:捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

7.根据权利要求1所述的编码器(10),其特征在于,数据处理装置(20)可操作用于通过采用以下中的至少一种对中间数据(40)编码以产生编码数据(D2):熵修正、熵修正(EM)编码、ODelta编码、行程长度编码(RLE)、分解行程长度编码(SRLE)、算术编码、行程编码、可变长度编码(VLC)。

8.一种编码器(10)中的方法,该编码器(10)用于对提供到该编码器(10)的数据(D1)编码以产生相应编码数据(D2)其中编码器(10)包括数据处理装置(20),用于将一个或多个编码过程应用于数据(D1)以产生编码数据(D2)其特征在于,该方法包括:

(a)使用数据处理装置(20)在该数据(D1)不是以数值符号表示的情况下,至少部分地以数值符号的集合表示该数据(D1);

(b)使用数据处理装置(20)产生中间数据(40),其中以原始值表示数值符号,以修正值表示至少一个符号,该修正值具有由连续统运算器产生的一个或多个连续统符号,其中该一个或多个连续统符号修正前面的符号值以适应扩展符号范围;以及

(c)使用数据处理装置(20)处理中间数据(40)以产生编码数据(D2)。

9.根据权利要求8所述的方法,其特征在于,该方法包括使用数据处理装置(20)基于计算出的符号在多个部分或片段中的出现概率及其压缩效率,将数据(D1)分割为所述多个部

分或片段。

10.根据权利要求8所述的方法，其特征在于，该方法包括针对数据(D1)中的符号，使用连续统运算器产生中间数据(40)，所述中间数据(40)是通过一种或多种变换方法和/或通过一个或多个转换表(LUT)对所述符号进行变换而产生的。

11.根据权利要求10所述的方法，其特征在于，该方法包括通过多个数据流提供编码数据(D2)，其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。

12.根据权利要求11所述的方法，其特征在于，指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从编码器(10)的备选源得到指示所述一个或多个转换表(LUT)的信息。

13.根据权利要求8所述的方法，其特征在于，该方法包括使用编码器(10)处理的数据(D1)包括以下中的至少一种：捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

14.根据权利要求8所述的方法，其特征在于，该方法包括使用数据处理装置(20)通过采用以下中的至少一种对中间数据(40)编码以产生编码数据(D2)：熵修正、熵修正(EM)编码、ODelta编码、行程长度编码(RLE)、分解行程长度编码(SRLE)、算术编码、行程编码、可变长度编码(VLC)。

15.根据权利要求8所述的方法，其特征在于，使用记录在非瞬态机器可读数据存储介质上的一个或多个软件产品来实现该方法，其中所述一个或多个软件产品能够在编码器(10)的数据处理装置(20)上执行，以实现对数据(D1)编码以产生编码数据(D2)的方法。

16.一种解码器(50)，用于对提供到该解码器(50)的编码数据(D2)解码以产生相应解码数据(D3)其中解码器(50)包括数据处理装置(60)，用于将一个或多个解码过程应用于编码数据(D2)以产生解码数据(D3)其特征在于：

(a)数据处理装置(60)可操作用于处理编码数据(D2)以产生中间数据(40)；

(b)数据处理装置(60)可操作用于处理中间数据(40)以解码中间数据，其中在中间数据中，数值符号由输出符号及至少由一个修正输出符号表示，该修正输出符号具有一个或多个连续统符号，该一个或多个连续统符号随后由逆连续统运算器解码，其中所述一个或多个连续统符号修正该修正输出符号的值以适应扩展符号范围；以及

(c)数据处理装置(60)可操作用于变换和/或转换处理后的中间数据以通过符号集合来表示解码数据(D3)。

17.根据权利要求16所述的解码器(50)，其特征在于，数据处理装置(60)可操作用于基于计算出的符号在多个部分或片段中的出现概率及其压缩效率，将编码数据(D2)作为所述多个部分或片段进行处理。

18.根据权利要求16所述的解码器(50)，其特征在于，通过由一个或多个转换表(LUT)和/或一种或多种变换方法限定的一次或多次变换对解码数据(D3)中的值进行变换和/或转换，来实现连续统运算器产生符号集合。

19.根据权利要求18所述的解码器(50)，其特征在于，通过多个数据流提供编码数据(D2)其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的

信息。

20.根据权利要求19所述的解码器(50),其特征在于,指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从解码器(50)的备选源得到指示所述一个或多个转换表(LUT)的信息。

21.根据权利要求16所述的解码器(50),其特征在于,解码器(50)可操作用于处理的编码数据(D2)包括以下中的至少一种的编码版本:捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

22.根据权利要求16所述的解码器(50),其特征在于,数据处理装置(60)可操作用于通过采用以下中的至少一种的逆操作对编码数据(D2)解码以产生中间数据(40):熵修正、熵修正(EM)编码、ODelta编码、行程长度编码(RLE)、分解行程长度编码(SRLE)、算术编码、行程编码、可变长度编码(VLC)。

23.一种使用解码器(50)对提供到该解码器(50)的编码数据(D2)解码以产生相应解码数据(D3)的方法,其中解码器(50)包括数据处理装置(60),用于将一个或多个解码过程应用于编码数据(D2)以产生解码数据(D3)其特征在于,该方法包括:

(a)使用数据处理装置(60)处理编码数据(D2)以产生中间数据(40);

(b)使用数据处理装置(60)处理中间数据(40)以解码中间数据,其中在中间数据中,数值符号由输出符号及至少由一个修正输出符号表示,该修正输出符号具有一个或多个连续统符号,该一个或多个连续统符号随后由逆连续统运算器解码,其中所述一个或多个连续统符号修正该修正输出符号的值以适应扩展符号范围;以及

(c)使用数据处理装置(60)来变换和/或转换处理后的中间数据以通过符号集合来表示解码数据(D3)。

24.根据权利要求23所述的方法,其特征在于,该方法包括使用数据处理装置(60)基于计算出的符号在多个部分或片段中的出现概率及其压缩效率,将编码数据(D2)作为所述多个部分或片段进行处理。

25.根据权利要求23所述的方法,其特征在于,该方法包括通过由一个或多个转换表(LUT)和/或一种或多种变换方法限定的一次或多次变换对解码数据(D3)中的值进行变换和/或转换,来使用连续统运算器产生符号集合。

26.根据权利要求25所述的方法,其特征在于,通过多个数据流提供编码数据(D2)其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。

27.根据权利要求26所述的方法,其特征在于,指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从解码器(50)的备选源得到指示所述一个或多个转换表(LUT)的信息。

28.根据权利要求23所述的方法,其特征在于,该方法包括使用解码器(50)处理包括以下中的至少一种的编码版本的编码数据(D2):捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

29.根据权利要求23所述的方法,其特征在于,该方法包括使用数据处理装置(60)通过采用以下中的至少一种的逆操作对编码数据(D2)解码以产生中间数据(40):熵修正、熵修

正(EM)编码、ODelta编码、行程长度编码(RLE)、分解行程长度编码(SRLE)、算术编码、行程编码、可变长度编码(VLC)。

30.根据权利要求23所述的方法,其特征在于,使用记录在非瞬态机器可读数据存储介质上的一个或多个软件产品来实现该方法,其中所述一个或多个软件产品能够在解码器(50)的数据处理装置(60)上执行,以实现对编码数据(D2)解码以产生解码数据(D3)的方法。

31.一种编解码器(100),包括至少一个根据权利要求1所述的编码器(10)和至少一个根据权利要求16所述的解码器(50)的组合,其中所述至少一个解码器(50)可操作用于解码由编码器(10)编码的数据(D2)。

32.一种电子设备(10、50),包括至少一个根据权利要求1所述的编码器(10)和至少一个根据权利要求16所述的解码器(50),其中该电子设备(10、50)实现为以下中至少一种的一部分:个人计算机、音频/视频设备、电视机、无线计算设备、智能电话、移动电话、交互游戏操控台、汽车电子信息系统。

编码器设备、解码器设备和方法

技术领域

[0001] 本公开涉及用于对接收到的数据进行编码以产生相应编码数据的编码器设备。此外，本公开还涉及用于对数据进行编码以产生相应编码数据的方法。此外，本公开涉及用于对接收到的编码数据进行解码以产生相应解码数据的解码器设备。此外，本公开涉及用于对编码数据进行解码以产生相应解码数据的方法。本公开还涉及包括一个或多个上述编码器设备与一个或多个上述解码器设备的组合的编解码器。此外，本公开涉及记录在非瞬态(非瞬时)机器可读数据存储介质上的软件产品，其中该软件产品可在计算硬件上执行以执行上述方法。英国专利申请GB1303658.7的公开内容通过全文引用合并于此。

背景技术

[0002] 一般而言，当前的数据通信网络和数据处理设备需要处理越来越大量的数据。这种数据处理相应地需要更大的数据通信带宽和/或更大的数据存储容量。提供这样的带宽和/或存储容量的成本是很高的。因此，在进行数据通信和/或存储时对数据进行压缩将具有可观的益处。

[0003] 当前，信息通常表示为数据的形式，例如音频、图像、视频、图表、ECG、地震数据、测量数据、数字、Excel表单、字符、文本、新闻、ASCII字符、Unicode字符、二进制数据、广告、多维数据等。此外，这些数据可以表达为不同格式，例如比特、字节、字、字符、数字、图片，等等。此外，可以采用近几十年间开发的多种不同编码方法对当前的信息进行编码。如上所述，信息的存储和/或传输常常是必要的，因此，将信息表达为尽可能小量(例如，就以比特为单位的数据大小而言)的编码数据(例如，熵编码数据和附加信息)将会是有益的。

[0004] 在考虑数据编码方法时，将每条信息视为一个元素或符号是很方便的。将多条信息这样表示为元素或符号允许例如使用Shannon熵计算方法(见参考文献[2]、[3]和[4])来计算信息的熵。可以在多种不同算法(例如熵编码算法和/或熵修正(entropy modifynig)算法)之前或之后，针对多种不同符号表示来执行这种计算。例如，可以使用多种熵编码方法来对单个符号进行熵编码。此外，还可以将符号从一种形式转换为另一种形式，例如，数字可以转换为文本，文本可以转换为字，比特可以转换为字节。

[0005] 单个符号的示例包括例如比特值(1、6、8、10、……比特)、字节值(8比特)、字值(16、32、64、128、……比特)、(ASCII、Unicode、中文、阿拉伯……)字符、二进制位置标记(以2为基)、八进制(以8为基)标记、十进制(以10为基)标记、十六进制(以16为基)标记、或罗马数字标记。可选地，数字符号可以具有小数点，即分数或实数值，或具有非小数形式(自然数或整数值)。此外，可选地，符号可以包括图片、数据或数据库元素，等等。另外，数字和字符还可以表示为基于单个数字或字符(例如ASCII)的符号，或表示为例如数值、字或句的多个ASCII符号的组合。

[0006] 如上所述，符号表示允许执行信息的熵的计算。此外，可以通过使用不同符号表示来计算同一条给定信息的熵，从而产生不同的熵结果，例如，采用不同符号集合来表示信息可能使得由集合中一个或多个符号表示的信息具有不同的熵。而且，根据需要，不同符号的

熵编码可以差异很大。上述信息的一些表示适于被熵编码为非常接近它们的理想熵(例如利用算术编码器或行程编码器得到的熵),而一些表示则需要更多的附加信息以实现成功的熵编码,例如字或数据库元素所遇到的情况。

[0007] 上面提到的附加信息需要从给定编码器以这样或那样的方式传递到对应解码器,以使得能够实现编码数据的唯一解码。另外,当一些附加信息在给定编码器和给定解码器处均已知时,这也是有益的,这样,这些信息就完全不需要传递,或例如通过使用标识一个或多个表的一个或多个索引而以很小的格式来传递。

[0008] 换言之,以相应编码数据传递上述信息的方式在编码数据所实现的数据压缩度方面的差别很大,可能的替代方案是,例如,将整个原始信息本身以原始符号的形式、以压缩形式的符号、或以可用信息替代的选择索引的形式来发送。此外,整个被传递的信息或其部分可以被重用,这也为压缩信息、相应数据或相应编码数据在更大程度上创造了多种可选项。

[0009] 特别地,当表示信息的原始数据量增加时,在传送该信息时,常常没有可供选择的合适静态表或数据库。但是,在与编码形式的信息传送相关联地传递一个或多个表之后,对于例如编码方法参考了所述表的后来传送的其他信息而言,就可能存在一些可再用的表。还可以理解,要压缩的信息可能是较大信息体的一部分,例如,一条信息可能是全部或部分数据的分析结果、一个或多个数据块的方法参数,等等,例如多级方法的层级、数据库引用、原始数据的一部分(例如视频的ROI、帧片、图像)。

[0010] 当要传送大量数据时,数据的熵占据大部分要传递的数据量。类似地,当仅要传递较少数量的数据时,附加信息通常在很大程度上成为被传递数据的主要部分,换言之,附加信息可能占据可观的数据开销。因此,需要优化以最小化熵编码数据和附加数据的总和,如下所说明的那样,根据本公开的连续统运算器(Continuum Operator)是非常出色的优化工具。

[0011] 当前存在很多种不同数据压缩方法可用于压缩数据。一些压缩方法是专用于一些特定种类的数据的,例如JPEG/PNG用于压缩图像、AAC/MP3用于压缩音频、PNG/GIF用于压缩图表、HEVC/VP9用于压缩视频,等等。一些方法更为折中,例如BZip、7Zip、RLE、SRLE、VLC、行程编码、算术编码等。此外,还有一些方法可用于修正比特数据的熵(例如:英国专利申请GB1303658.7中描述的在熵修正器(EM)中采用的方法),并且还有一些方法可以修正并非以各个比特位表示的符号数据的熵,例如在英国专利申请GB1303661.1中描述的DPCM、Delta编码和ODelta编码,以及在英国专利申请GB1303660.3中描述的RLE和SRLE。尽管参考文献[2]、[3]和[4]中描述的Shannon熵是公知的,但其并未在当前的压缩方法中被广泛适当地使用。可以使用以下公式1来计算Shannon熵:

[0012] $$Entropy = -\sum_{i=1}^{n} p(x_i) * \log p(x_i)$$ 公式1

[0013] 其中:

[0014] n是不同符号的数量;以及

[0015] $p(x_i)$是由i索引的符号的概率。

[0016] 通常将熵乘以所有符号的数量以使其值与其他计算得到的熵值更加有可比性。也可以通过将该可比熵值除以值log(2)来改变该可比熵值,以估计所使用的比特。

[0017] 与熵不同的是，在有损编码中常常使用率失真(RD)优化来选择最佳压缩方法或方法组合。在无损编码中，熵本身可用于选择方法或算法，因为在无损编码中，没有RD优化所基于的失真，因此与附加信息一起，仅通过熵即可方便地估计率本身。

[0018] 与上述信息相对应的数据的交织也是已知的现有技术方法。例如，给定图像中如参考文献[11]中描述的表示为RGB的像素颜色值可以表示为平面形式(RRRR...,GGGG...,BBBB...)或交织形式(RGB,RGB,RGB,RGB,...)。

[0019] 在英国专利申请GB2301252(见参考文献[10])中，描述了一种对数据中存在的比特进行编码的已知方法。该已知方法采用多个不同长度剩余符号，但是以严格定义的方式逐个使用该多个不同剩余符号，它们表示不同的比特动态，并且该已知方法仅适用于比特符号。但是，单独或以已知组合来使用任何已知方法不足以解决与数据压缩有关的三个主要问题中的任何一个。所有上述方法及其组合都有很多缺点。

[0020] 在对信息编码(例如压缩)时，出现三个主要问题：

[0021] 1)第一个问题涉及在压缩信息时要使用的符号的最适当形式的选择方式；

[0022] 2)第二个问题涉及最有效地减小结果中相似符号的方式；以及

[0023] 3)第三个问题涉及最有效地减小编码数据和附加信息的数据大小(例如将其最小化)但仍使得能够(例如在解码器中解压缩时)解码出唯一数据的方式。

发明内容

[0024] 本公开意在提供对表示信息的数据进行编码以产生相应编码数据的改进方法。

[0025] 此外，本公开还意在提供对表示信息的数据进行编码以产生相应编码数据的改进编码器。

[0026] 此外，本公开意在提供对例如由上述改进的数据编码方法产生的编码数据进行解码的改进方法。

[0027] 此外，本公开意在提供对例如由上述改进的编码器产生的编码数据进行解码的改进解码器。

[0028] 根据第一方面，提供了一种编码器，用于对提供到该编码器的数据(D1)编码以产生相应编码数据(D2)，其中编码器包括数据处理装置(20)，用于将一个或多个编码过程应用于数据(D1)以产生编码数据(D2)，其特征在于：

[0029] (a)数据处理装置(20)可操作用于，如果该数据(D1)不是以数值符号表示的，则至少部分地以数值符号的集合表示该数据(D1)；

[0030] (b)数据处理装置(20)可操作用于产生中间数据(40)，其中以原始值表示数值符号，以修正值表示至少一个符号，该修正值具有由连续统运算器产生的一个或多个连续统符号，其中该一个或多个连续统符号修正前面的符号值以适应扩展符号范围；以及

[0031] (c)数据处理装置(20)可操作用于处理中间数据(40)以产生编码数据(D2)。

[0032] 本发明的优势包括编码器添加的一个或多个连续统符号的组合能够提供相比于要编码的数据(D1)具有更高数据压缩度的编码数据(D2)。

[0033] 可选地，在编码器中，数据处理装置可操作用于基于计算出的符号在多个部分或片段中的出现概率及其压缩效率，将数据(D1)分割为所述多个部分或片段。

[0034] 可选地，在编码器中，使用连续统运算器产生中间数据(40)，以通过一种或多种变

换方法和/或通过一个或多个转换表(LUT)来变换数据(D1)中的符号。可选地,在编码器中,通过多个数据流提供编码数据(D2),其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。可选地,在编码器中,指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从编码器的备选源得到指示所述一个或多个转换表(LUT)的信息。

[0035] 可选地,编码器可操作用于处理的数据(D1)包括以下中的至少一种:捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

[0036] 可选地,在编码器中,数据处理装置可操作用于通过采用以下中的至少一种对中间数据(40)编码以产生编码数据(D2):熵修正、熵修正(EM)编码、ODelta编码、RLE、SRLE、算术编码、行程编码、VLC。

[0037] 根据第二方面,提供了一种编码器中的方法,该编码器用于对提供到该编码器的数据(D1)编码以产生相应编码数据(D2),其中编码器包括数据处理装置,用于将一个或多个编码过程应用于数据(D1)以产生编码数据(D2),其特征在于,该方法包括:

[0038] (a)使用数据处理装置(20)在该数据(D1)不是以数值符号表示的情况下,至少部分地以数值符号的集合表示该数据(D1);

[0039] (b)使用数据处理装置(20)产生中间数据(40),其中以原始值表示数值符号,以修正值表示至少一个符号,该修正值具有由连续统运算器产生的一个或多个连续统符号,其中该一个或多个连续统符号修正前面的符号值以适应扩展符号范围;以及

[0040] (c)使用数据处理装置(20)处理中间数据(40)以产生编码数据(D2)。

[0041] 可选地,该方法包括使用数据处理装置基于计算出的符号在多个部分或片段中的出现概率及其压缩效率,将数据(D1)分割为所述多个部分或片段。

[0042] 可选地,该方法包括针对数据(D1)中的符号,使用连续统运算器产生中间数据(40),所述中间数据是通过由一个或多个转换表(LUT)和/或一种或多种变换方法限定的一次或多次变换和或转换,对所述符号进行变换而产生的。可选地,该方法包括通过多个数据流提供编码数据(D2),其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。可选地,在该方法中,指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从编码器的备选源得到指示所述一个或多个转换表(LUT)的信息。

[0043] 可选地,该方法包括使用编码器处理的数据(D1)包括以下中的至少一种:捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

[0044] 可选地,该方法包括使用数据处理装置通过采用以下中的至少一种对中间数据编码以产生编码数据(D2):熵修正、熵修正(EM)编码、ODelta编码、RLE、SRLE、算术编码、行程编码、VLC。

[0045] 可选地,使用记录在非瞬态机器可读数据存储介质上的一个或多个软件产品来实现该方法,其中所述一个或多个软件产品能够在编码器的数据处理装置上执行,以实现对数据(D1)编码以产生编码数据(D2)的方法。

[0046] 根据第三方面,提供了一种解码器,用于对提供到该解码器的编码数据(D2)解码

以产生相应解码数据(D3)，其中解码器包括数据处理装置，用于将一个或多个解码过程应用于编码数据(D2)以产生解码数据(D3)，其特征在于：

[0047] (a)数据处理装置(60)可操作用于处理编码数据(D2)以产生中间数据(40)；

[0048] (b)数据处理装置(60)可操作用于处理中间数据(40)以解码中间数据，其中在中间数据中，数值符号由输出符号及至少由一个修正输出符号表示，该修正输出符号具有一个或多个连续统符号，该一个或多个连续统符号随后由逆连续统运算器解码，其中所述一个或多个连续统符号修正该修正输出符号的值以适应扩展符号范围；以及

[0049] (c)数据处理装置(60)可操作用于变换和/或转换处理后的中间数据以通过符号集合来表示解码数据(D3)。

[0050] 可选地，在解码器中，数据处理装置可操作用于基于计算出的符号在多个部分或片段中的出现概率及其压缩效率，将编码数据(D2)作为所述多个部分或片段进行处理。

[0051] 可选地，在解码器中，通过由一个或多个转换表(LUT)和/或一种或多种变换方法限定的一次或多次变换对解码数据(D3)中的值进行变换和/或转换，来实现连续统运算器产生符号集合。

[0052] 可选地，在解码器中，其特征在于，通过多个数据流提供编码数据(D2)，其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。可选地，在解码器中，指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从解码器(50)的备选源得到指示所述一个或多个转换表(LUT)的信息。

[0053] 可选地，解码器可操作用于处理的编码数据(D2)包括以下中的至少一种的编码版本：捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

[0054] 可选地，在解码器中，数据处理装置可操作用于通过采用以下中的至少一种的逆操作对编码数据(D2)解码以产生中间数据：熵修正、熵修正(EM)编码、ODelta编码、RLE、SRLE、算术编码、行程编码、VLC。

[0055] 根据第四方面，提供了一种使用解码器对提供到该解码器的编码数据(D2)解码以产生相应解码数据(D3)的方法，其中解码器包括数据处理装置(60)，用于将一个或多个解码过程应用于编码数据(D2)以产生解码数据(D3)，其特征在于，该方法包括：

[0056] (a)使用数据处理装置(60)处理编码数据(D2)以产生中间数据(40)；

[0057] (b)使用数据处理装置(60)处理中间数据(40)以解码中间数据，其中在中间数据中，数值符号由输出符号及至少由一个修正输出符号表示，该修正输出符号具有一个或多个连续统符号，该一个或多个连续统符号随后由逆连续统运算器解码，其中所述一个或多个连续统符号修正该修正输出符号的值以适应扩展符号范围；以及

[0058] (c)使用数据处理装置(60)来变换和/或转换处理后的中间数据以通过符号集合来表示解码数据(D3)。

[0059] 可选地，该方法包括使用数据处理装置基于计算出的符号在多个部分或片段中的出现概率及其压缩效率，将编码数据(D2)作为所述多个部分或片段进行处理。

[0060] 可选地，该方法包括使用连续统运算器产生符号集合，所述符号也通过由一个或多个转换表(LUT)和/或一种或多种变换方法限定的一次或多次变换从解码数据(D3)而被

变换和/或转换。可选地，该方法包括通过多个数据流提供编码数据(D2)，其中至少一个流传递指示所述一个或多个转换表(LUT)和/或一种或多种变换方法的信息。可选地，在该方法中，指示所述一个或多个转换表(LUT)的信息参考在传递编码数据(D2)之前传递的一个或多个转换表(LUT)和/或能够从解码器的备选源得到指示所述一个或多个转换表(LUT)的信息。

[0061] 可选地，该方法包括使用解码器处理包括以下中的至少一种的编码版本的编码数据(D2)：捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据。

[0062] 可选地，该方法包括使用数据处理装置通过采用以下中的至少一种的逆操作对编码数据(D2)解码以产生中间数据(40)：熵修正、熵修正(EM)编码、ODelta编码、RLE、SRLE、算术编码、行程编码、VLC。

[0063] 可选地，使用记录在非瞬态机器可读数据存储介质上的一个或多个软件产品来实现该方法，其中所述一个或多个软件产品能够在解码器的数据处理装置上执行，以实现对编码数据(D2)解码以产生解码数据(D3)的方法。

[0064] 根据第五方面提供了一种编解码器，包括至少一个根据第一方面的编码器和至少一个根据第三方面的解码器的组合，其中所述至少一个解码器可操作用于解码由编码器编码的数据。

[0065] 根据第六方面提供了一种电子设备，包括至少一个根据第一方面的编码器和至少一个根据第三方面的解码器，其中该电子设备实现为以下中至少一种的一部分：个人计算机、音频/视频设备、电视机、无线计算设备、智能电话、移动电话、交互游戏操控台、汽车电子信息系统。

[0066] 本发明的实施例有益地使用组合在一起的多个元件和操作。所述多个元件包括例如：

[0067] (i)用于熵计算的一个或多个元件，例如用于计算Shannon熵及使用Shannon熵进行算法选择和处理加速目的；

[0068] (ii)用于符号到符号转换的一个或多个元件，例如用于组合符号、分割符号、将比特转换为字节、将字节转换为比特、提供2比特符号到6比特符号或6比特符号到2比特符号的转换、将字符转换为字、将字转换为字符/句、将数字/字符转换为值或将值转换为数字/字符；

[0069] (iii)用于传递附加信息的一个或多个元件，所述附加信息例如所选编码方法、用于限定方法参数、用于提供码表、用于提供查找表、用于提供和/或指示数据库；以及

[0070] (iv)用于熵修正(EM)编码的一个或多个元件，用于Delta编码(参见参考文献[8])、用于ODelta编码、用于RLE、用于SRLE(见参考文献[9])、用于UK专利申请GB2301252(见参考文献[10]，其通过引用合并于此)中描述的方法，以及用于熵编码(例如算术编码、行程编码或VLC、RLE、SRLE)。

[0071] 因此，本公开的实施例采用改进的符号数据编解码方法。该改进的方法可以直接用于一比特数据符号或在变换到较高动态符号之后使用，还用于具有原始较高动态的数据符号。还公开了通过使用熵和附加信息进行计算来在不同编码方法直接有效地进行选择的方法。此外，在本公开中，描述了用于识别要采用的最佳符号格式的算法、最佳熵修正方法

和最佳编码方法的适当组合，以及最有效地使用该组合进行数据压缩和解压缩的最优形式。

[0072] 此外，本公开的实施例有益地采用连续统运算器，即数字符号值(数据)范围修正器的新形式，同时采用了表或符号比特计数缩减器，其能够大大减少熵编码数据和附加信息的和。此外，本公开提供了对比特和符号数据编码和解码的新方法。

[0073] 可以理解，在不脱离所附权利要求限定的本发明范围的情况下，本发明的特征易于组合为多种形式。

附图说明

[0074] 现在将参考以下附图，仅通过示例来描述本公开的实施例，在附图中：

[0075] 图1是根据本公开的用于编码和相应地解码数据的编码器和解码器(总称为编解码器)的示意图。

[0076] 在附图中，使用带下划线的数字来表示该带下划线的数字位于其上的项目或该带下划线的数字与之相邻的项目。不带下划线的数字与通过将该不带下划线的数字和一项目相连的线所标识的项目相关联。当一个数字不带下划线并且伴随有相关联的箭头时，使用该不带下划线的数字来标识该箭头所指向的一般项目。

具体实施方式

[0077] 整体上，本公开涉及压缩数据的改进方法以及相应地解压缩数据的改进方法，这些改进方法通过它们对数据范围(data range)修正器(即连续统运算器)的使用而加以区分。此外，该数据压缩和解压缩方法组合了各种不同的比特或符号修正器、不同熵修正器和熵编码器，以对可表示为不同种类的比特或符号数据的原始信息进行压缩和解压缩。此外，该压缩和解压缩方法将例如Split Run Length编码(SRLE)(见参考文献[1]、[9]和[10])与熵修正器(见参考文献[13])以及至少一个上述高级连续统运算器的特性相结合。该至少一个连续统运算器包含至少一个连续统值(continuum value)，具有正或负连续统，其将给定原始数字符号值修正为相应的修正符号值。针对原始或已修正符号中的至少一个，至少一次使用为该连续统值创建的修正符号。该连续统值是事先已知的，或者是从已应用或未应用熵编码的实际数据编码单独得出的。所使用的连续统值符号包含在原始或修正符号数据中，它们在应用或未应用任何熵编码器的情况下被编码，即与连续统运算器修正的符号数据中的原始或修正符号数据相似。

[0078] 以后仍可以利用已知熵编码方法对连续统运算器产生的符号流进行压缩，例如参见参考文献[2]、[3]和[4]中例如算术编码(见参考文献[5])、行程编码(见参考文献[6])、或Variable Length编码(VLC)(见参考文献[7])以及已知熵修正器例如ODelta编码(见参考文献[8]和[14])，或熵修正器(EM)(见参考文献[13])。还可以事先通过任何前述方法，还可以通过采用其他方法或变形来修改要使用本公开的压缩方法压缩的数据。可以通过使用计算的熵值和附加信息的相应估计值，来进行对这种压缩方法组合的选择，但该选择也可以基于可实现的真实编码结果来进行，例如通过对原始数据的一个或多个部分采用压缩方法的不同组合，以迭代方式进行。此外，在本公开中，还描述了可将编码数据恢复回原始信息(例如为比特或符号数据的形式)的解码器。

[0079] 在本公开的实施例中，数据压缩方法和相应的数据解压缩方法采用数据范围修正器，称为连续统运算器，其适用于所有比特或符号数据。此外，使用连续统运算器还优化了使用任何熵编码器被熵编码的数据符号所需的码表传递或传递单个数据符号所需的比特。

[0080] 上述改进的压缩数据方法和相应的解压缩数据方法组合了各种不同比特或符号修正器、不同熵修正器以及熵编码器，以将表示为不同种比特或符号数据的原始信息压缩/解压缩。该压缩/解压缩方法组合了例如Split Run Length编码(RLE/SRLE)(见参考文献[1]、[9]、和[10])、熵修正器(EM)(见参考文献[13])、ODelta编码器(见参考文献[14])、以及连续统运算器的特性，所述参考文献[1]、[9]、[10]、[13]和[14]通过引用合并于此。

[0081] 连续统运算器仅向原始数值符号数据添加连续统值符号，并同时修改连续统值中的原始数字符号值。这意味着任何原始数值符号都与一个或多个数值符号一起存在于修正数据中。当原始数字符号值与被接受为原始数值的数值不同时，它被替换为与被接受的原始值相似的修正数值以及一个或多个连续统值。通常，在实践中，这意味着当原始数字符号值小于等于符号值阈值(SVT)时，原始符号值被原样加入修正符号数据。否则，一个或多个连续统值符号被添加到符号值传递。在此情况下，最后一个符号或第一个符号是小于等于SVT的原始符号的修正版本，并且连续统值符号在它之前或之后。连续统值和修正符号值的结果必须与原始符号值相同是为了实现连续统运算器的正确解码，例如作为无损数据压缩和解压缩。

[0082] 由于前述特性，总是可以将符号值与编码数据分离开。可接受的每个值是它自己的解码值，并在计算它的解码符号值时将它之前(或之后)的所有可能连续统值与它一起使用。换言之，在修正数据中小于等于SVT的符号(如果0值也是可接受的值)数量总是与原始符号数据中的原始符号数量相同。此外，还存在一个或多个连续统值符号，这些连续统值符号与小于等于SVT的修正数据符号或原始数据符号(如果0值也是可接受的值)是不同的符号。有益地，原始和修正符号呈现为从0到SVT的符号，连续统值符号则呈现为符号SVT+1、SVT+2，等等。有益地，最小的SVT值是2，否则使用LUT就比使用连续统运算器更有效率。

[0083] 采用连续统运算器来产生可使用已知熵编码方法(见参考文献[2]、[3]和[4])以及其他编码方法来压缩的修正符号流，所述其他编码方法例如是算术编码(见参考文献[5])、行程编码(见参考文献[6])、或Variable Length编码(VLC)(见参考文献[7])。以后也可以使用已知熵修正器来修正由连续统运算器修正的符号流，所述已知熵修正器例如是ODelta编码(见参考文献[8]和[14])，或具有或不具有熵编码器的熵修正器(EM)(见参考文献[13])。要压缩的数据也可通过各种上述方法以及其他方法或变形被事先(即先验地)修正。

[0084] 如上所述，可以通过使用计算的熵值和附加信息(例如表传递、方法选择连续统值传递)的估计或计算量来进行对所采用的压缩方法组合的选择，但也可以例如以迭代的方式，基于可获得的真实编码结果来进行该选择，直到找到压缩原始数据和/或其一个或多个部分的最优组合。此外，下面的说明书中将描述可用于将编码数据恢复(即解压缩)回例如以比特或符号数据表达的原始信息的解码器。

[0085] 下面将描述本公开的实现方式的实施例。这些方法例如以编码器或相应解码器的硬件逻辑来实现。此外，要压缩的原始数据可选地从一个或多个传感器得到，所述一个或多个传感器例如是一个或多个照相机和/或一个或多个麦克风，其将真实的物理现象转换为

相应的表示数据。备选地，通过使用布置为执行记录在非瞬态(非瞬时)机器可读数据存储介质上的一个或多个软件产品的计算硬件来实现所述方法，其中软件产品实现本公开的上述方法。在下面将要描述的示例中，通过使用本公开的方法来压缩一条原始样本信息，还提供了与使用已知现有技术方法可获得的结果的一些比较。

[0086] 包含N个符号的一条示例信息可由公式2表示：

[0087] s1,s2,s3,...sN

[0088] 或s0,s1,s2,...,sN-1

[0089] 或s(1),s(2),s(3),...s(N),,公式2

[0090] 等等

[0091] 在原始信息中，符号可以在使用上述连续统运算器之前或之后被分割为分离的序列或流。类似地，如果存在很多序列或流，则它们可在连续统运算器之前或之后被组合以提供一个或多个符号。

[0092] 基于由符号表示的原始数据的一个或多个部分的熵的一个或多个计算和估计的附加信息，可以确定表示原始信息的符号的最佳备选形式来编码(即压缩)原始信息。可以基于表示原始信息所需符号的概率来计算上述表示原始信息的符号的熵。类似地，可以基于组合符号的出现概率来计算组合符号的熵。可以理解，组合符号通常明显少于原始符号，而且组合流少于原始信息的原始流。

[0093] 还可以理解，与组合符号或组合流相关联的附加信息总是多于相应原始符号或原始流的附加信息，换言之，具有相关联的数据头部来指示一个或多个符号和/或一个或多个流已经在给定过程中被组合以产生相应的压缩数据。这是因为，一般地，组合符号之间的相关性不等于1。此外，需要一些机制来指示哪些组合符号具有非零概率，否则，例如在采用熵编码时，传递VLC码表或行程编码概率需要大量附加信息。当需要传递或编码较少原始数据时，附加信息的量也是更显著的数据部分。

[0094] 类似地，可以基于分割流或符号的概率来计算分割流或符号的熵。如果基于原始数据的分割创建了多个流，则分别计算每个流的熵然后求和是有益的。此外，每个流分别需要附加信息。一般地，与该分割有关的附加信息的单独序列中的任意序列的数据大小都小于一个流所需的附加信息，但是流的所有附加信息序列之和可以小于或大于一个流所需的附加信息。当从1比特到8比特符号流分割或组合原始信息时，附加信息的数据大小一般会减小，例如减小到最小值。当已知附加信息的所有熵和序列时，可以通过将熵比特和附加信息减小到最小值来选择最佳符号格式。

[0095] 优选地，还可以基于原始符号创建SRLE符号和串流(runs stream)用于根据本公开压缩数据，参见英国专利申请GB 1303660.3，类似的美国专利申请US13/782,872，其相关内容通过引用合并于此。以下公式3示出了用于获得两个流的符号的示例：

[0096] S(1),S(x1),S(x2),...(针对符号)；以及

[0097] R(1),R(x1),R(x2),...(针对串) 公式3

[0098] 其中：

[0099] S(1)是第一符号；以及

[0100] R(1)是类似的随后符号S(1)的串。

[0101] 符号S(x1)是随后符号S(1)的串R(1)之后的下一个不同符号，R(x1)是类似的随后

符号S(x1)的串。

[0102] 符号S还可以表示如下：

[0103] S(1),S(1+R(1)),S(1+R(1)+R(2)),...(针对符号) 公式4

[0104] 现在，例如，在压缩原始信息时，在迭代到采用的最优方法的过程中，还可以将这种组合的熵与较早的备选方案相比较。

[0105] 熵修正器和熵编码可用于处理比特数据，即用于以逐个比特的方式处理数据，如英国专利申请GB 1303658和GB 2301252中所述，但是它们不适于处理高阶符号数据。用于处理比特数据的这些连续统符号、剩余符号或换码符号方法也可以被修改以用于具有本公开所描述的特征的符号数据，例如，可以通过使用在用于比特串流的EM中使用的类似连续统符号技术来修正SRLE而非RLE串流(其中在同一流中存在串和符号)。除了以上描述的比特之外，RLE/SRLE的大多数已知实现方式不采用连续统符号、剩余符号或换码符号。当存在可以利用最大串值(例如8比特)表达的更多随后类似符号时，通过将同一符号再次与新串号相加，来传递类似符号的极长串。

[0106] 在第一示例中，如果将8比特符号与8比特串一起使用，并且存在300个随后“A”字符，则相应RLE写作：A255 A45(即流中有相同数量的符号和串)。还存在第二RLE方案，当存在至少两个类似符号时仅使用且始终使用一个串值，但对于单个符号不使用串值。同一示例可以表示为AA255 AA45(即流中符号的数量是串的至少两倍)。如上所述，RLE/SRLE实际上不使用任何连续统符号、剩余符号或换码符号(即编码)，但可能存在自身具有串的随后类似符号。此外，还可以使用SRLE(见参考文献[9])将符号和串划分到不同流。现在，第一流可表示为AA和255、45，第二流可表示为AAAA和255，45。但是，可以理解，值255具有特殊的重要性，并且基于字节(8比特)动态，值255既不是连续统值、剩余值或换码符号，也不是编码。值255也可单独在不同位置使用，其并不额外需要以高于8比特(例如以16比特)来传递换码串值。当与SRLE一起使用换码符号(编码)方案时，有益地，使用至少三个流，例如A在符号流中，换码符号(例如0)在串流中，300在换码串值流中。

[0107] 接下来，将描述与SRLE方法结合使用上述连续统运算器的示例。连续统运算器修正串的数量必须大于原始串符号的数量，从而当串符号值高于所选一个或多个连续统值并至少高于符号值阈值(SVT)的值时，添加用于表示仍连续统的类似符号的数量的连续统值符号，即(连续统+/-值和修正符号值或修正符号值和+/-连续统值)。例如，当存在N(=330)个符号时，串值如下公式5所示：

[0108] 2,7,9,102,12,21,19,3,1,6,18,18,16,10,5,37,19,5,2,18 公式5

[0109] 该流，即总共20个符号值，具有串值，范围在1到102之间，可以例如通过使用Variable Length编码(VLC)、Range编码或算术编码而被压缩。此外，可以理解，所有原始串值必须大于等于1，最大串值必须小于等于N=所有符号数=330。传递这些符号直接需要例如20*9=180比特，其中值330可以用9比特或4+20*7=144比特表示，其中4比特用于传递用于符号的比特(=7，因为值102可以用7比特表示)。如果没有可用的合适固定码表，如固定的预定码表，则传递相应码表将需要多得多的比特，所以很有可能不能与公式5中的这种数据一起有效地使用熵编码方法。

[0110] 根据本公开的实施例，可以通过使用SVT值优化器经过大量计算，然后使用符号0(因为0不能在原始串流中)选择与SVT值类似的一个连续统值(例如19+)来表示修正串流中

的连续统值。现在，可以将这些串R(x)值修正为修正串R'(x)值，如公式6所示：

[0111] 2，7，9，0,0,0,0,0,7，12，0,2，19，3，1，6，18，18，16，10，5，0,18，19，5，2，18 公式6

[0112] 符号(0,x)表示解码期间的值(19+x)，类似的符号(0,0,0,x)例如表示解码期间的值(19+19+19+x)或(19+(19+(19+x)))。连续统运算器修正串R'(x)符号值之和在解码期间仍是330，因为在计算总和时，0值可被替换为值19。符号数量因此从20增加到27，即由于有3个高于SVT值的串值，添加了7个0。可以用例如6+27*5＝141个比特来传递该流，即27个符号，符号值为(1到19+0)，6个比特用于传递连续统值(cont+)，其类似于SVT值(＝19)，20(＝1到19+0)个不同符号值可通过5个比特加以区分。还可以例如通过采用VLC、行程编码或Arithmetic编码，使用用于参考文献[15]和[16]中描述的编码方法的表传递来压缩该数据，上述文献通过引用合并于此。

[0113] 可选地，将连续统运算器修正为包含更多连续统值，例如cont1+＝19+，cont2+＝50+。此外，SVT值与cont1+＝(19+)值类似，等于19。第一连续统值即cont1+仍使用符号0，第二连续统值即cont2+将被分配符号20(＝SVT+1)。连续统运算器修正串流R'(x)可以写作公式7提供的形式：

[0114] 2，7，9，20,20,2，12，0,2，19，3，1，6，18，18，16，10，5，0,18，19，5，2，18 公式7

[0115] 现在，在公式7中，通过采用符号(20,20,2)而不是符号(0,0,0,0,0,7)来对串值“102”编码，于是串符号的总数减小到24个符号。公式7中的该流，即24个符号，具有符号(0到20)，可以例如通过6+6+24*5＝132比特来传递，6比特用于传递连续统(19和50)值，21(cont1(＝0)，cont2(＝20)和从1到19的值)个不同符号值可通过使用每符号值5个比特来呈现。还可以通过使用例如上述VLC、行程编码或算术编码，使用用于参考文献[15]、[16]中描述的编码方法的表传递来压缩该数据，该参考文献通过引用合并于此。当使用多个连续统值时，这些连续统值的传递也可以为不同方式，例如19和31而不是19和50。可选地，如果使用具有负值的连续统值，则在传递连续统值时，也需要例如通过采用符号比特或不同编码方法来将指示负值的信息发送到解码器或数据存储设备。有益地，如果必须使用多于两个或三个其他连续统值，则定义新连续统值。

[0116] 有益地，在对原始符号流中的原始符号使用连续统运算器之前，对原始符号流中的原始符号执行变换，使得变换后的流包含0值作为其中的最小值。下面将描述在使用连续统运算器之前的示例串值变换，这修正了可获得的结果。下面的示例中还对连续统值做了微小修正。可以理解，串符号值必须在范围1到N之间，如上所述，这些符号需要被1变换而无需传递任何信息。现在，原始R(x)流表示为用于传输的T(x)流，如公式8所示：

[0117] 1,6,8,101,11,20,18,2,0,5,17,17,15,9,4,36,18,5,1,17 公式8

[0118] 可选地，可以将SVT值选择为例如18(＝19-1)，但在下文中，选择SVT值为19。此外，将两个连续统值选择为针对符号20是cont1+(＝SVT+＝19+)，针对符号21是cont2+(＝100+)。这使得相应的连续统运算器修正转换符号T'(x)成为公式9所示：

[0119] 1，6，8，21,1，11，20,1，18，2，0，5，17，17，15，9，4，20,17，18，4，1，17 公式9

[0120] 因此，该流中的串符号总数是23个符号。该23个符号的流具有符号(0到21)，可以

例如以(5,6或7)+7+23*5＝127到129个比特来传递,其中5-7比特用于传递cont1+＝(19+)值,7比特用于传递cont2+(＝100+)值,可以通过将5比特用于每个符号值,来表示22(＝0到19+20+21)个不同符号值。还可以例如通过采用VLC、行程编码或算术编码,使用用于参考文献[15]、[16]中描述的编码方法的表传递来压缩该数据,上述文献通过引用合并于此。

[0121] 可以理解,还可以使用任何熵编码方法来压缩针对SRLE实现方式优化的连续统运算器之后的符号流S(x)。如果S(x)包含比特值,则仅需要传递第一比特值,如参考文献[13]中EM那样,或需要传递第一比特值的信息,因为当针对串值使用连续统运算器时已知其他值逐个改变,从而使得对于连续统类似符号值仅需要传递一个串值,例如,如果S(x)＝1,0,1,0,1,0,1,0,1,则对于符号S(x)仅传递第一值1或例如利用编码方法_1_EM_CM_Range而产生的第一值1的信息,与之一起传递的还有针对串R(x)值的连续统运算器修正R'(x)或T'(x)值。当利用针对SRLE实现方式优化的连续统运算器压缩比特值以外的值时,已知不会发生S(x)流中有两个类似随后值,但仍不知道下一个值实际是什么。因此,必须还以某种熵编码方法压缩符号值S(x),并且压缩连续统运算器修正R'(x)或T'(x)值。

[0122] 如上所述,虽然连续统运算器主要在EM或SRLE之后传递串符号,但也可以在压缩任何符号时使用连续统运算器。当首先利用连续统运算器修改包含很多小值和仅一些高值的任意符号流时,可以利用或不利用熵编码来高效地编码该符号流。有时,当还存在很多高和/或中值时,连续统运算器也可能是可采用的适当方法,例如,当符号s(x)值如公式10所示时:

[0123] 3,1,252,254,2,252,1,1,252,0,254,3,6 (13个符号) 公式10

[0124] 于是,连续统运算器修正符号值s'(x)对应于SVT＝3,连续统值cont1+(＝3+)对应于符号＝4,cont2+(＝252+)对应于符号＝5,如公式11所示:

[0125] 3, 1, 5,0, 5,2, 2, 5,0, 1, 1, 5,0, 0, 5,2, 3, 4,3(13+1+5＝19个符号) 公式11

[0126] 因此,公式10中的原始流包含8比特符号值,即0到254,可以利用13*8＝104比特来传递。公式11中的连续统运算器修正流包含19个符号,可以仅通过每符号3比特(0到5)来传递和通过8+8+19*3＝16+57＝73比特来传递,该结果是可观的压缩,该计算中的16比特使得最多需要73比特来传递这两个连续统值。

[0127] 当尝试利用使用一个或多个换码符号(码)的方法来压缩相同符号流(即公式10)时,可得到的数据压缩并不如公式11呈现的成功。符号频率非常类似,一般根本不使用换码符号(码)。为了使用大量换码符号(码)来传递码表或原始符号,例如当利用换码符号(码)和具有其本身符号值的其他符号来传递6以上的值时,可以尝试减少码表传递或减少传递小符号值所需的比特,于是码表或小值本身仅包含从0到6的值和换码符号(码)(＝8个不同符号)。在这种情况下,符号的传递使用8(必须传递值6)+13*3(用于每个符号-原始或换码)+5*8(在换码符号之后的换码值)＝8+39+40＝87比特。该结果明显好于不使用一个或多个换码符号(码),但该结果仍旧劣于通过使用连续统运算器修正结果获得的结果。当流中还存在一些中值时,获得连续统运算器的更多有益结果。

[0128] 下面将描述另一示例。当原始流s(x)如公式12所示时:

[0129] 3,1,252,254,2,252,1,1,252,0,254,3,6,147,149,2,253,148,1,152(20个符号) 公式12

[0130] 于是,连续统运算器修正符号值s'(x)对应于SVT＝3,连续统值cont1+(＝3+)对应于符号＝4,cont2+(＝147+)对应于符号＝5,cont3+(＝252+)对应于符号＝6,如公式13所

示：

[0131] 3，1，6,0，6,2，2，6,0，1，1，6,0，0，6,2，3，4,3，5,0，5,2，2，6,1，5,1,1，5,4,2 公式13

[0132] 原始流包含8比特值(0到254)，可以利用20*8＝160比特来传递，连续统运算器修正流包含32(＝20+2+4+6)个符号，可以仅通过每符号3比特(0到6)来传递和通过8+8+8+32*3＝24+96＝120比特来传递，最多需要24比特来传递这三个连续统值。

[0133] 关于换码符号(码)，当利用至少一个换码符号(码)和具有其本身值的其他符号来传递6以上的值时，码表或小值仅包含从0到6的值和至少一个换码符号(码)(＝8个不同符号)。在这种情况下，符号的传递使用8(必须传递值6)+20*3(用于每个符号-原始或换码)+10*8(在换码符号之后的换码值)＝8+60+80＝148比特。该压缩结果仍好于不使用至少一个换码符号(码)，但该结果仍旧劣于通过使用连续统运算器获得的结果。

[0134] 可以理解，有益地，第一连续统值总是限定SVT。因此，取决于原始流中的符号频率，第一连续统值应当尽可能小。还可以使用其他连续统值来减少传递较大符号值所需的比特，一般地，所述较大符号值明显高于第一连续统值。所有连续统值都被分配其自身的符号，有益地，这些符号是SVT+1、SVT+2，等等。如果X表示所使用的连续统值的量，则存在0到SVT个原始或修正符号和X个连续统值＝＞即当在连续统运算器之前使用转换时，修正流中有(SVT+1)+X个不同符号。

[0135] 当然，如果不使用转换，则例如在原始流中不使用0符号，于是符号0可以用于第一连续统值。在此情况下，连续统运算器实现为其从不为连续统目的之外的目的而产生0符号，例如，如果连续统值是3(符号＝0)，并且串是3，则它不能由符号(0,0)编码，而必须以符号(3)编码。在此示例中，可以理解符号(3)往往比符号(0,0)更好编码，但是当例如cont1＝3+(符号＝0)和cont2＝30+(符号＝4)，并且需要传递串值30时，需要使用符号(0,0,0,0,0,0,0,0,0,3)，不能使用符号(4,0)。

[0136] 通常，当一个或多个第一值在原始数据流中不可用时，最好通过使用已知(例如对于串是1)偏移值或传递的偏移值来在连续统运算器之前转换符号值。然后，例如，如果存在从3到200的给定流值，则使用用于SVT符号的顶部的连续统值符号，然后，有益地，对原始符号采用偏移-3，从而获得值0到197，以提供转换符号。现在，有益地，将SVT选择为10，将cont1值选择为10+，从而对其得到符号11。可以向其他连续统值分配从12到所需任意多个符号。因此，所有的小值都用于原始或修正符号，连续统值被分配符号SVT+1、SVT+2，等等。

[0137] 还可以理解，利用连续统运算器，必须有至少一个连续统值是以它自身的符号(例如SVT+1)定义(＝SVT+)的，并且在修正符号流中使用至少一次。在修正符号流中还可以定义和使用多个连续统值(cont1,cont2,...,contN)。一个原始符号可以呈现为0、一个或多个连续统值符号和小于等于SVT的一个符号，例如，如果cont1＝10+(通过使用符号11来表示)并且cont2＝100+(通过使用符号12来表示)，并且值(0到10)采用原始符号(0到10)，则例如值7呈现为不具有连续统值符号(7)、值17和107呈现为具有一个连续统值符号(11,7和12,7)，值27、117和207呈现为具有两个连续统值符号(11,11,7和12,11,7;和12,12,7)，值37,127,217和307呈现为具有三个连续统值符号(11,11,11,7和12,11,11,7和12,12,11,7和12,12,12,7)，等等。可以从一个或多个连续统值符号构造一个原始符号，这些连续统值符号可以彼此相同或不同。对于一个原始符号，还可以有单独使用的更高连续统值符号，即

不使用较低连续统值符号。根据所选连续统值和所选解码公式，通常可以很自由地改变实际连续统符号的顺序。例如，在前一示例中，当仅使用正连续统+值时，例如，值217可以表示为12,12,11,7或12,11,12,7或11,12,12,7。

[0138] 通过引用合并于此的参考文献[14]中描述的ODelta编码还可以用于改进可获得的编码结果。一般地，当原始符号是数字或值时，将其用于原始符号，然后可以实现符号之间的减法或加法计算。

[0139] 可选地，还可以首先将可用符号转换为调色板(palette)，换言之，通过查找表(LUT)来表示可用符号的数字，所述数字也需要被表达或传递，然后，可用如上所述地直接或利用ODelta编码来使用这些符号。在符号已经是数字或值但数字或值离散分布在整个对应动态符号范围内的情况下，通过使用表来缩减符号数量可能是非常有益的。

[0140] 当要处理的信息包含多个通道时(例如RGB图像像素)，也可以将信息变换为颜色空间内的不同值，通过使用逆变换可将所述不同值逆变(有时近似逆变即可)回原始值。这种变换对的一个代表示例是RGB到YUV和YUV到RGB变换，如上所述。一般地，在YUV颜色空间中，自然图像是可压缩的，如参考文献[12]所述，其通过引用合并于此，因为信息的主体以及大部分重要信息存在于Y通道中，颜色通道U和V可被下采样例如至YUV420或YUV422以获得更好的压缩率。在R、G和B通道之间存在很明显的关联，通过执行RGB到YUV变换来减小该关联。此外，如果其他信息能够减小不同通道的信息直接的关联，则可以使用所述其他信息。

[0141] 还存在很多不是数值的符号。例如，在利用连续统运算器处理之前，字符需要先变为数值。一般地，该变换是例如A＝0，B＝1，C＝2，等等，但也可以使用其他变换。例如，一些语言中不同字符的概率往往可以是非常有用的变换方法，即这些字符可以通过它们的概率顺序来表示。

[0142] 连续统运算器总是输入具有相同动态范围(例如用于表达符号的比特宽度、字长度、比特数)的值。此外，连续统运算器通常在处理后返回具有较小动态范围的值，换言之，在处理后，使用较少比特来表达符号。至少，与原始输入数据相比，结果数据中的不同符号备选的可能性必须总是更小。

[0143] 使用连续统运算器减小数据大小的一个很好的示例是罗马数字系统。该数字系统总是很好地限定一个小符号集合，即I＝1，V＝5，X＝10，L＝50，C＝100，D＝500，以及M＝1000，该小符号集合可以用于表示例如从1到4000的正整数值，可以通过使用比特标记，以每符号12比特来表示这4000个不同符号。类似地，可以利用每符号3比特来表示这7个不同罗马数字标记符号，即I＝0、V＝1、X＝2、L＝3、C＝4、D＝5，M＝6。通常，可以用少于4个罗马数字符号来表示正整数值，因此对于这种特定符号，罗马数字标记比比特标记更有效率。在使用这种运算器时还有三个大的缺点。首先，不支持零值或在不使用正负号时不支持负值。其次，如果没有可用的附加信息，例如值之间的空格，没有办法将正整数值彼此分离开。第三，相关联的“SVT”值太小。SVT值可以被检测到，其他字符可以被检测为其他连续统值。但是，现在SVT值太小，例如当要表达数字8或3333时，所以可能产生问题。例如，如果有如下正整数：

[0144] 11,9,33,101,4,2,1,8,3,10,34,68,

[0145] 则存在12个符号->12*12比特＝144比特或12*7比特＝84比特。此外，必须存在至

少101个不同符号以呈现所有值而无需特定分隔符。于是，相应的罗马数字标记可以呈现为：

[0146] XI,IX,XXXIII,CI,IV,II,I,VIII,III,X,XXXIV,LXVIII

[0147] 其中存在36个字符+11个分隔符->36*3＝108比特。

[0148] 可以理解，由于这些数是正整数，所以不需要传递正负号信息。因此，如果使用偏移＝-1则得到如下流s(x)：

[0149] 10,8,32,100,3,1,0,7,2,9,33,67

[0150] 此外，如果选择SVT＝3,cont1+＝3+(符号＝4),cont2+＝10+(符号＝5),cont3+＝33+(符号＝6),cont4+＝100+(符号＝7)，等等，则这种标记被方便地称为罗马连续统运算器，当使用该罗马连续统运算器时不需要传递相关方法本身之外的附加信息。现在，该运算器之后的流s'(x)如下所示：

[0151] 5,0, 4,4,2, 5,5,5,2, 6,6,6,1, 3, 1, 0, 4,4,1, 2, 4,4,3, 6,0, 6,6,1

[0152] 其中存在28个符号+0个分隔符->28*3＝84个比特。

[0153] 这些经过连续统运算的符号不需要任何特定分隔符，而是可以写在一起，而仍旧可以仅通过检测修正流中从0到3的所有符号或符号值知道有多少个符号以及如何形成它们。这意味着对于经连续统运算器(即，使用8个不同符号)处理的值504425552666131044124436066l也是有效的，但对于使用5个不同符号的罗马数字标记，XIIXXXXIIICIIVIIIVIIIIIIXXXXIVLXVIII是无效的，因为不能从流解码出数字量和它们的值，即出现一个或多个歧义。还可以理解，例如当在运算之后使用熵编码方法时，不同符号的量越小，压缩表传递所需的附加信息越少。因此，经连续统运算器处理的符号优于二进制标记，尽管在不经过熵编码的情况下，前一示例中的最小必要比特量是相同的。

[0154] 如上所述，根据本公开，也可以将原始符号数据划分为多个片段或部分。通常，不同数据片段可用的符号的概率是不同的，因此独立编码每个片段或部分是有益的。因此，在如上所述地应用连续统运算器之前，可以将符号流划分为多个片段。

[0155] 可以使用不同方法来编码原始信息，即符号数据，以产生编码数据，可选地，例如，以顺序(i)到(iii)执行不同方法：

[0156] (i)ODelta编码；

[0157] (ii)连续统运算器；以及

[0158] (iii)行程编码。

[0159] 在实现为硬连线逻辑、专用集成电路(ASIC)和/或记录在非瞬态(非瞬时)机器可读记录数据存储介质上的一个或多个软件产品的根据本公开的编码器中执行这种编码方法，其中该一个或多个软件产品可以在计算硬件上执行以实现该编码方法。可选地，该计算硬件包括在个人计算机、音频/视频设备、电视机、无线计算设备、智能电话、移动电话、交互式游戏操控台、汽车电子信息系统中，但不限于上述设备。

[0160] 例如在根据本公开的解码器中，在步骤(i)到(iii)中，以逆序将编码数据解码回原始信息(符号数据)：

[0161] (i)逆行程编码；

[0162] (ii)逆连续统运算器；以及

[0163] (iii)逆ODelta编码。

[0164] 在实现为硬连线逻辑、专用集成电路(ASIC)和/或记录在非瞬态(非瞬时)机器可读记录数据存储介质上的一个或多个软件产品的根据本公开的解码器中执行这种解码方法,其中该一个或多个软件产品可以在计算硬件上执行以实现该解码方法。可选地,该计算硬件包括在个人计算机、音频/视频设备、电视机、无线计算设备、智能电话、移动电话、交互式游戏操控台、汽车电子信息系统中,但不限于上述设备。

[0165] 下面描述本公开的其他示例,其中要压缩一条原始样本信息。该原始样本信息可选地包括以下中的至少一种:捕获音频信号、捕获视频信号、捕获图像、文本数据、地震数据、传感器信号、模数(ADC)转换数据、生物信号数据、日历数据、经济数据、数学数据、二进制数据,但不限于此。在压缩原始信息期间,可选地采用已知和新数据处理方式的组合。

[0166] 如上所述,可选地,利用负连续统提供连续统值。当利用连续统运算器修正ODelta编码的符号时,这是非常有益的方法。如果存在例如7比特原始值并且使用高值80,采用ODelta运算器来修正这些原始值,则在ODelta运算器修正流中存在大量接近0的值,以及大量接近80的值。此外,可以理解,在ODelta运算器修正符号中,符号0意味着与前值相同,符号1意味着前值+1,符号80意味着前值-1,以此类推。流D(x)可以表达为公式14:

[0167] 4,0,80,0,1,80,79,0,0,2,0,0,0,80,1,0,1,0,1,80,78(21个符号) 公式14

[0168] 如果采用SVT=2,cont1=2+对应于符号3,cont2=80-对应于符号4,则在连续统运算器之后获得D'(x)如公式15所示:

[0169] 3,2,0,4,0,0,1,4,0,4,1,0,0,2,0,0,0,4,0,1,0,1,0,1,4,0,4,2 公式15

[0170] 符号(4,x)现在意味着值(80-x)。此外,原始流包含7比特值(0到80),通过采用21*7=147比特来传递,连续统运算器修正流包含28(=21+1+6)个符号,可利用每符号仅3比特(0到4)来传递,因此可以用7+8+28*3=15+84=99比特来传递,传递这两个连续统值最多需要15比特。第一值表示SVT,必须为0到80之间的正值,即需要7比特来表达它,第二值使用例如1比特来限定连续统值是正还是负,最多需要7比特来表达第二绝对连续统值。随后可以例如通过使用行程编码、算术编码或VLC编码来容易地压缩该流,因为符号频率是0到4即(13,5,3,1,5)。可以理解,ODelta运算器和连续统运算器的组合允许将与前值相同的值,即差值为0的值,或具有正delta即正差值的值作为绝对值传递,而无需任何比特用于正负号,它还允许通过使用cont2-符号作为正负号指示以及随后的绝对值来传递具有负delta即负差值的值。因此,该组合是非常有效率的压缩方法,并实现了传递原始符号值所需的比特的极大压缩,换言之,实现了高效的数据压缩。

[0171] 可选地,可以使用多个正和负连续统值,总是可以通过使用以下公式来计算解码结果,此时存在三个连续统值符号用于一个符号s(x)。公式16使用了标记,其中至少一个符号是修正符号,连续统值符号在该修正符号之前。可选地,还可以使用逆方案。

[0172] s(x)=conta(s'(y-3))+/-(contb(s'(y-2))+/-(contc(s'(y-1))+/-s'(y))) 公式16

[0173] 其中

[0174] x是原始符号的索引;

[0175] y是相应修正符号的索引;以及

[0176] conta、contb和contc是所使用的连续统值。

[0177] 可选地,可以使用正或负连续统值,因此在实际计算中使用相应+/-运算符,因此,

例如，在前一示例中出现第一符号4,3,3,1，于是解码值如下公式17所示

[0178] s(0)＝conta(s′(3-3))+/-(contb(s′(3-2))+/-(contc(s′(3-1))+/-s′(3)))＝80-(2+(2+1))＝75 公式17

[0179] 还可以使用另一公式18，其中正和负连续统值彼此跟随，即采用相邻方式，而不使用任何附加括号，如下所示：

[0180] s(x)＝conta(s′(y-3))+/-contb(s′(y-2))+/-contc(s′(y-1))+/-s′(y) 公式18

[0181] 例如，在前一示例中，当第一符号是4,3,3,1时，解码值如公式19所示：

[0182] s(0)＝conta(s′(3-3))+/-contb(s′(3-2))+/-contc(s′(3-1))+/-s′(3)＝80-2+2+1＝81 公式19

[0183] 该示例不是最优的，因为末值在原始符号范围之外。但是，如果还有第三符号5用于第三连续统值(cont3+＝10+)，则得到改进，例如，在包括第一符号4,5,3,1的前一示例中，解码值如公式20所示：

[0184] s(0)＝conta(s′(3-3))+/-contb(s′(3-2))+/-contc(s′(3-1))+/-s′(3)＝80-10+2+1＝73 公式20

[0185] 利用另一公式(公式20)将同一第一符号4,5,3,1解码为公式21所示：

[0186] s(0)＝conta(s′(3-3))+/-(contb(s′(3-2))+/-(contc(s′(3-1))+/-s′(3)))＝80-(10+(2+1))＝67 公式21

[0187] 此外，根据选择使用哪种连续统编码运算器，可以使用很多其他解码功能。公式中总是有1个修正符号和一个或多个连续统符号，可以使用或不使用括号来使用不同运算器。例如，一个很不同的功能是以下公式22，其中没有cont+/-运算器，而是使用+/-运算器，其中修正符号在连续统值之前：

[0188] s(x)＝-(s′(y)+/-conta(s′(y+1))+/-contb(s′(y+2))+/-contc(s′(y+3))) 公式22

[0189] 如果现在符号是1,4,5,3，则：

[0190] s(0)＝-(s′(0)+/-conta(s′(0+1))+/-contb(s′(0+2))+/-contc(s′(0+3)))＝-(1-80+10+2)＝67 公式23

[0191] 可以理解，当修正符号在连续统值之前时，可以仅基于下一新符号(即新原始或修正值)或基于数据末尾(对于最后一个符号)来检测修正符号的末尾。因此，使用在修正值之前的连续统值而不是修正值之后的连续统值通常是更有益的。

[0192] 参考图1，提供了根据本公开的编码器10的示意图。编码器10可操作用于接收数据D1和处理数据D1以产生相应的编码数据D2。有益地，编码数据D2对应于数据D1的至少一部分的压缩版本。编码器10包括可操作用于应用根据本公开的编码方法的硬连线逻辑和/或计算硬件20。编码器10接收数据并且如果D1不是符号形式则将其转换为相应符号。可选地，输入数据D1被划分为多个部分或片段30，其中每个部分或片段30有益地被选择为使得其符号的出现概率对于实现数据压缩是非常有效率的。然后，例如使用上述一个或多个原始或修正符号编码和/或一个或多个连续统值符号，利用上述连续统运算器来对该符号集合编码，以产生中间数据40。该中间数据40可选地被例如使用上述行程编码或类似编码方式而被进一步编码，以产生编码数据D2。

[0193] 在根据本公开的解码器50中，有益地执行在编码器10中执行的计算操作的逆操作，来将编码数据D2转换为解码数据D3，其中解码数据D3有益地至少是数据D1的实质相似

再现，当编码器10和解码器50以无损方式工作时，数据D3和数据D1可以是相同的。解码器50包括可操作用于应用根据本公开的解码方法的硬连线逻辑和/或计算硬件60。

[0194] 编码器10和解码器50的结合构成了附图标记100所示的编解码器。如上所述，数据D1有益地对应于表示实际物理变量的数据，例如通过使用一个或多个传感器捕获的物理变量。该一个或多个传感器可选地包括无线通信设备(例如智能电话、平板电脑、PDA，等等)的麦克风或照相机中的至少一个。

[0195] 可以在不脱离由所附权利要求限定的本发明的范围的情况下对上面描述的本发明的实施例进行修改。用于描述和限定本发明的例如“包括”、“包含”、“合并”、“由……构成”、“具有”、“是”等表述意在被理解为非排他形式，即允许存在未明确描述的项目、组件或元件。单数形式应理解为不排除多数。所附权利要求中的括号中的数字意在辅助对权利要求的理解，而不应被理解为以任何方式限制权利要求所请求的主题。

[0196] 附录

[0197]

参考文献	具体内容
[1]	Run-length encoding - Wikipedia, the free encyclopedia (accessed November 28, 2012). URL: http://en.wikipedia.org/wiki/Run-length_encoding
[2]	Shannon's source coding theorem - Wikipedia, the free encyclopedia (accessed November 28, 2012). URL: http://en.wikipedia.org/wiki/Source_coding_theorem
[3]	Shannon, Claude E. (1948) (accessed November 28, 2012) A Mathematical Theory of Communication. URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf
[4]	Entropy - Wikipedia, the free encyclopedia (accessed November 28, 2012). URL: http://en.wikipedia.org/wiki/Entropy
[5]	Arithmetic coding - Wikipedia, the free encyclopedia (accessed November 28, 2012). URL: http://en.wikipedia.org/wiki/Arithmetic_coding
[6]	Range encoding - Wikipedia, the free encyclopedia (accessed April 26, 2013). URL: http://en.wikipedia.org/wiki/Range_encoding
[7]	Variable-length code - Wikipedia, the free encyclopedia (accessed November 28, 2012). URL: http://en.wikipedia.org/wiki/Variable-length_code
[8]	Delta encoding - Wikipedia, the free encyclopedia (accessed November 28, 2012). URL: http://en.wikipedia.org/wiki/Delta_coding
[9]	Patent EP1685651A2 - Split runlength encoding method and apparatus - Google Patents (accessed October 1, 2013). URL: http://www.google.com/patents/EP1685651A2?cl=en
[10]	Run length data compression - United Kingdom Patent 2301252-A (accessed October 1, 2013). URL: http://patent.ipexl.com/GB/GB2301252.html
[11]	RGB color model - Wikipedia, the free encyclopedia (accessed October 1, 2013). URL: http://en.wikipedia.org/wiki/RGB_color_model
[12]	YUV - Wikipedia, the free encyclopedia (accessed October 1, 2013). URL: http://en.wikipedia.org/wiki/YUV
[13]	EM patent application (Gurulogic case 010) (patent application ?)
[14]	ODelta Coding patent application (Gurulogic case 011) (patent application ref?)
[15]	Table delivery patent application (Gurulogic case 016) (patent application ref?)
[16]	Range Coding patent application (Gurulogic 028) (patent application ref?)

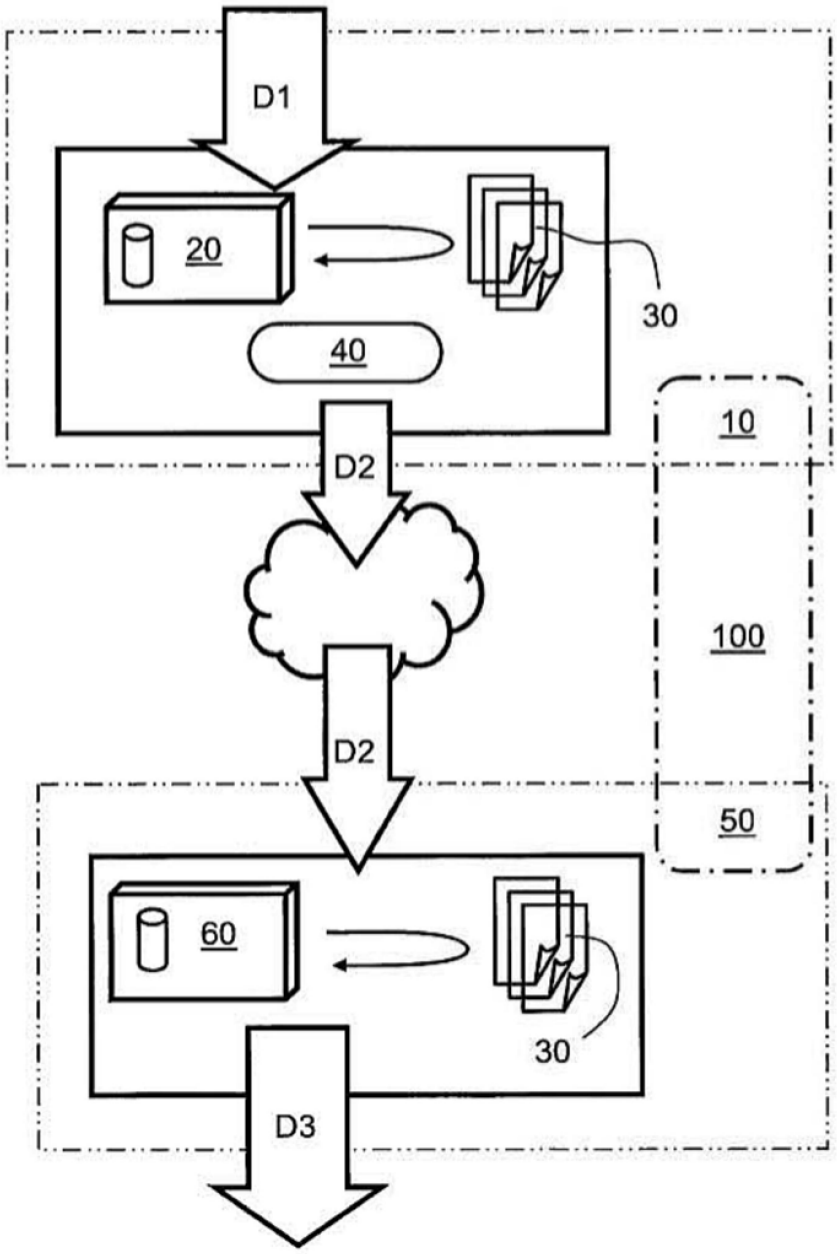


图1