



[12] 发明专利申请公开说明书

[21] 申请号 200510108887.7

[43] 公开日 2006年6月7日

[11] 公开号 CN 1783144A

[22] 申请日 2005.9.29

[21] 申请号 200510108887.7

[30] 优先权

[32] 2004.10.29 [33] US [31] 10/977,701

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 H·S·马尔瓦

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 沈昭坤

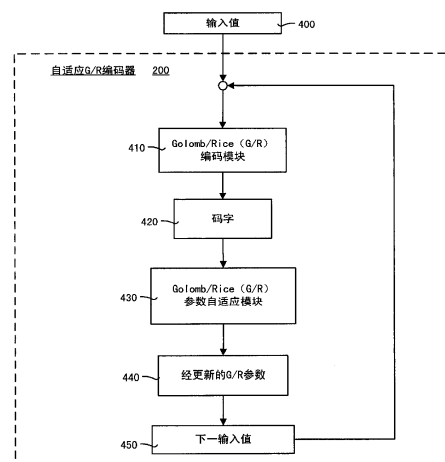
权利要求书 5 页 说明书 20 页 附图 9 页

[54] 发明名称

使用后向自适应规则进行整数数据的无损自适应 Golomb/Rice 编码和解码

[57] 摘要

一种使用具有新颖的自适应规则的新颖后向自适应技术进行整数数据的无损自适应 Golomb/Rice (G/R) 编码的方法和系统。该自适应 G/R 编码器和解码器(编解码器)及方法使用了在生成每个码字后调整 G/R 参数的自适应规则。这些自适应规则包括定义一自适应值并根据该自适应值调整 G/R 参数。如果自适应值等于零,则使 G/R 参数减少一整数常数。如果自适应值等于一,则 G/R 参数保持不变。如果自适应值大于一,则使 G/R 参数增加该自适应值。此外,该自适应 G/R 编码器和方法包括小数自适应,它根据 G/R 参数定义了经比例缩放的 G/R 参数,并更新和调整该经比例缩放的 G/R 参数来减慢自适应速率。



1. 一种用于处理数字数据的方法，其特征在于，包括：
使用 Golomb/Rice (G/R) 参数编码所述数字数据的输入值以生成所述输入值
5 的码字；
在生成所述码字后使用具有自适应规则的后向自适应技术来更新 G/R 参数；
以及
对所述数字数据的每个值重复所述编码和更新。
2. 如权利要求 1 所述的方法，其特征在于，还包括：
10 定义一自适应值；以及
当所述自适应值等于零时，减少所述 G/R 参数。
3. 如权利要求 2 所述的方法，其特征在于，还包括当所述自适应值等于零时，
使所述 G/R 参数减少一整数常数。
4. 如权利要求 2 所述的方法，其特征在于，还包括当所述自适应值等于一时，
15 保持所述 G/R 参数不变。
5. 如权利要求 4 所述的方法，其特征在于，还包括当所述自适应值大于一时，
增加所述 G/R 参数。
6. 如权利要求 5 所述的方法，其特征在于，还包括当所述自适应值大于一时，
使所述 G/R 参数增加所述自适应值。
- 20 7. 如权利要求 1 所述的方法，其特征在于，还包括：
定义一比例缩放参数；以及
定义一经比例缩放的 G/R 参数为等于所述 G/R 参数乘以所述比例缩放参数。
8. 如权利要求 7 所述的方法，其特征在于，还包括在生成所述码字后，通过
使用具有自适应规则的后向自适应技术更新所述经比例缩放的 G/R 参数而非所述
25 G/R 参数。
9. 如权利要求 8 所述的方法，其特征在于，还包括：
设置所述经比例缩放的参数等于十六；以及
基于所述数字数据的衰减参数确定所述 G/R 参数的值。
10. 如权利要求 1 所述的方法，其特征在于，所述数字数据还包括具有如下
30 值的整数向量：(a) 对于每个值为零的最可能值；以及 (b) 非零值，它具有按所
述非零值增加而减少的概率。

11. 一种具有用于执行如权利要求 1 所述的计算机可执行指令的计算机可读介质。

12. 一种具有用于编码具有整数值的数字整数数据的计算机可执行指令的计算机可读介质，其特征在于，包括：

5 使用自适应 Golomb/Rice (G/R) 编码和 G/R 参数 k 来编码所述整数值的每一个，以对所述整数值的每一个生成一码字；

使用所述 G/R 参数 k 来定义一经比例缩放的 G/R 参数 K；以及

在生成每个码字后使用后向自适应规则来更新所述经比例缩放的 G/R 参数 K。

13. 如权利要求 12 所述的计算机可读介质，其特征在于，还包括：

10 定义一比例参数 L；以及

定义所述经比例缩放的 G/R 参数 K 为 $K = k$ 乘以 L。

14. 如权利要求 13 所述的计算机可读介质，其特征在于，还包括设置所述经比例缩放的参数等于是二的幂的值。

15 15. 如权利要求 12 所述的计算机可读介质，其特征在于，所述数字整数数据还包括包含 N 个整数的向量 x，且其中，所述向量 x 的每个元素 x(n)，其中 n = 0 到 N-1，具有这样的概率分布：最可能值为零且距零更远的值有着快速下降的概率。

16. 如权利要求 15 所述的计算机可读介质，其特征在于，所述概率分布由以下公式给定：

$$P(x, \theta) = \frac{1-\theta}{1+\theta} \theta^{|x|}$$

20 并且其中，参数 θ 控制当 x 的绝对值增长时概率的衰减速率。

17. 如权利要求 12 所述的计算机可读介质，其特征在于，还包括：

定义一自适应值 p；以及

当 $p = 0$ 时用 $(K-B3)$ 代替 K，其中，B3 是一整数常数。

25 18. 如权利要求 17 所述的计算机可读介质，其特征在于，还包括当 $p = 1$ 时保持 K 不变。

19. 如权利要求 18 所述的计算机可读介质，其特征在于，还包括当 $p > 1$ 时用 $(K+p)$ 代替 K。

20. 如权利要求 17 所述的计算机可读介质，其特征在于，还包括：

定义一参数 u，当 $x > 0$ 时它为 $2x$ ；

30 定义 u，当 $x < 0$ 时它为 $-2x-1$ ；以及

定义 $p = u \gg k$ ，意指 p 等于 u 向右移 k 位，

21. 一种用于编码和解码数字整数数据的计算机实现的过程，其特征在于，包括：
- 使用自适应 Golomb/Rice (G/R) 编码和 G/R 参数 k 来编码所述数字整数数据的每个值 x ；
- 5 定义一经比例缩放的 G/R 参数为 $K=k*L$ ，其中 L 是比例缩放参数；
- 在编码了所述数字整数数据的每个值 x 后，使用具有自适应规则的后向自适应技术来更新所述经比例缩放 G/R 参数 K ；
- 将所述数字整数数据的已编码值追加到比特流中；以及
- 使用 G/R 解码器解码所述比特流，来精确恢复所述数字整数数据的每个值 x 。
- 10 22. 如权利要求 21 所述的计算机实现的过程，其特征在于，还包括：
- 当 $x>0$ 时用映射参数 $u=2x$ 代替 x ；
- 当 $x<0$ 时用 $u=-2x-1$ 代替 x ；以及
- 定义一自适应参数 p 为 u 向右移 k 位， $p=u>>k$ 。
23. 如权利要求 22 所述的计算机实现的过程，其特征在于，所述自适应规则
- 15 还包括当 $p=0$ 时用 $(K-B3)$ 代替 K ，其中 $B3$ 是一整数常数。
24. 如权利要求 23 所述的计算机实现的过程，其特征在于，所述自适应规则还包括当 $p=1$ 时保持 K 不变。
25. 如权利要求 24 所述的计算机实现的过程，其特征在于，所述自适应规则还包括当 $p>1$ 时用 $(K+P)$ 代替 K 。
- 20 26. 一个或多个其上具有计算机可读指令的计算机可读介质，当由一个或多个处理器执行所述指令时，使一个或多个处理器实现如权利要求 21 所述的计算机实现的过程。
27. 一种用于编码包含整数值的数字整数数据的自适应 Golomb/Rice (G/R) 编码器，其特征在于，包括：
- 25 一具有 G/R 参数 k 的 Golomb/Rice (G/R) 编码器，用于编码所述整数值；以及
- 一用于在编码了所述整数值的每一个后使用具有自适应规则的后向自适应技术来更新 G/R 参数 k 的装置。
28. 如权利要求 27 所述的自适应 G/R 编码器，其特征在于，所述自适应规则
- 30 还包括一用于如下更新所述 G/R 参数 k 的装置：
- 定义一自适应值 p ；

当 $p = 0$ 时使 k 减少一整数常数；

当 $p = 1$ 时保持 k 不变，以及

当 $p > 1$ 时使 k 增加 p 。

29. 如权利要求 28 所述的自适应 G/R 编码器，其特征在于，还包括一用于如下定义经比例缩放的 G/R 参数 K 的装置：

定义一比例参数 L ；

定义 $S = s * L$ ；以及

定义 $K = k * L$ 。

30. 如权利要求 29 所述的自适应 G/R 编码器，其特征在于，还包括一用于用所述自适应规则更新 K 而非 k 的装置。

31. 一种用于解码已编码比特流的方法，其特征在于，包括：

从所述已编码比特流接收一码字；

用一 Golomb/Rice (G/R) 参数解码所述码字；

在解码所述代码字后使用具有自适应规则的后向自适应技术来更新所述 G/R 参数；以及

对所述已编码比特流的每个码字重复所述解码和更新过程，以恢复重建的数字数据。

32. 如权利要求 31 所述的方法，其特征在于，还包括：

定义一自适应值；

当所述自适应值等于零时减少所述 G/R 参数；

当所述自适应值等于一时保持所述 G/R 参数不变；以及

当所述自适应值大于一时增加所述 G/R 参数。

33. 如权利要求 32 所述的方法，其特征在于，还包括当所述自适应值等于零时使所述 G/R 参数减少一整数常数。

34. 如权利要求 32 所述的方法，其特征在于，还包括当所述自适应值大于一时使所述 G/R 参数增加所述自适应值。

35. 如权利要求 31 所述的方法，其特征在于，还包括：

定义一比例缩放参数；以及

定义一经比例缩放的 G/R 参数为等于所述 G/R 参数乘以所述比例缩放参数。

36. 一种具有用于执行如权利要求 31 所述的方法的计算机可执行指令的计算机可读介质。

37. 一种用于解码已由一编码过程编码为已编码比特流的数字整数数据的过程，所述编码过程使用一 Golomb/Rice (G/R) 参数来编码所述数字整数数据的输入值以生成所述输入值的码字、在生成所述码字后使用具有自适应规则的后向自适应技术更新所述 G/R 参数、并对所述数字整数数据的每个输入值重复所述编码和更新，所述过程包括：
- 5 从所述已编码比特流接收一串码字；
使用自适应 G/R 解码和 G/R 参数 k 来解码每个码字；
用所述 G/R 参数 k 定义经比例缩放的 G/R 参数 K ；以及
在解码每个码字后使用所述具有自适应规则的后向自适应技术来更新所述经
- 10 比例缩放的 G/R 参数 K 。
38. 如权利要求 37 所述的过程，其特征在于，还包括：
定义一自适应值 p ；以及
当 $p = 0$ 时用 $(K-B3)$ 代替所述经比例缩放的 G/R 参数 K ，其中 $B3$ 是一整数常数。
- 15 39. 如权利要求 37 所述的过程，其特征在于，还包括当 $p = 1$ 时保持所述经比例缩放的 G/R 参数 K 不变。
40. 如权利要求 37 所述的过程，其特征在于，还包括当 $p > 1$ 时用 $(K+p)$ 代替所述经比例缩放的 G/R 参数 K 。

使用后向自适应规则进行整数数据的无损自适应 Golomb/Rice 编码和解码

5 技术领域

本发明一般涉及数字数据的处理，尤其是使用具有新颖的后向自适应规则的 Golomb/Rice 编码进行整数数据的无损编码与解码的改进方法和系统。

背景技术

10 随着计算机数据（诸如文本、音频、视频、图像和程序文件）的大小持续增长，数据压缩正变得越来越重要。数据压缩是一种将数字数据编码为比原始数据使用更少比特的经编码表示的方法。以较少比特表示数据意味着数据占据较少的存储空间并需要较小的传输带宽。

通常，数据压缩通过预测最常出现的数据并将之以较少的空间存储来压缩数据。具体地，数据压缩涉及至少两个不同的任务：（1）定义一数据模型来预测输入数据的概率；以及（2）用编码器从这些概率生成代码。此外，某些数据压缩技术在数学上变换并量化数据以达到甚至更大的压缩。

压缩技术可以是无损或有损的。无损压缩技术是可逆的，从而编码前的原始数据和解码后的解压缩数据的每一比特都是相同的。有损压缩利用了数据中有很多可以扔掉的重复而在质量上有更多损失这一事实。有损压缩接受某些原始数据的损失以便得到更高的压缩。

无损压缩一般用于压缩文本或二进制数据，而有损压缩一般用于音频、图像和视频数据。然而，即使有损压缩技术有时也使用无损压缩技术。例如，两种常用的压缩（或编码）技术是变换编码和预测编码。对于这些种类的压缩系统，原始数据经变换并随后被量化（舍入到最近的整数），或者根据（固定或自适应的）信号模型进行预测，且预测误差（原始和预测的数据之间的差异）随后被量化。在这两种情况下，经量化的数据是整数形式的。一旦获得这些整数，就要用无损压缩技术来编码经量化的值，以减少表示该数据所需要的比特数。

这些整数值集合通常有相关联的概率分布函数（PDF）。这些 PDF 具有这样的分布：当在预测编码中数据性质由预测者很好地模型化时，预测误差应在大多数时间接近于零。类似地，在变换编码中，大多数量化变换系数为零。图 1 示出对于

这些整数值的典型概率分布；零是最可能的值，而非零值的概率随数量增加呈接近指数级下降。数据具有图 1 所示的概率分布，因为使用无损压缩编码的数据不是原始数据。图 1 是通过量化变换系数或预测误差所得的整数数据。

从数学上讲，问题是要找出编码包含 N 个整数的向量 x 的有效解决方案。每个元素 $x(n)$, $n=0,1,\dots,N-1$ ，有一个按照类似于图 1 的概率分布的值，因而最可能值为零，而距零更远的值具有快速减少的概率。

像图 1 中这样的概率分布的简单数学模型是拉普拉斯 (Laplacian) 分布，或者双边几何 (TSG) 分布，由参数 θ 特征化：

$$P(x, \theta) = \frac{1-\theta}{1+\theta} \theta^{|x|} \quad (1)$$

注意，参数 θ 控制当 $|x|$ 增长时概率的衰减速率。 θ 值越大，衰减越快。参数 θ 可直接相关于概率，当 $x=0$ 时，即 $P(0, \theta) = (1-\theta)/(1+\theta)$ 。同样，源码元的期望绝对值是：

$$E[|x|] = \frac{2\theta}{1-\theta^2} \quad (2)$$

源的熵按比特/码元给出，如下式

$$H(x) = \log_2 \left(\frac{1+\theta}{1-\theta} \right) - \frac{2\theta}{1-\theta^2} \log_2(\theta) \quad (3)$$

因此，一个好的编码器应将 x 的 N 个值的向量映射到包含不多于 $N \cdot H(x)$ 个比特（理论最小值）的比特流。

拉普拉斯分布是媒体压缩系统中的常用模型，用于预测编码器（如大多数无损音频和图像编码器）中的预测误差，或者用于量化变换系数（如大多数有损音频、图像和视频编码器）。

对于具有拉普拉斯/TSG 分布的源有多种建议的编码器。一个简单而有效的编码器是 Golomb/Rice 编码器。首先，通过以下简单的可逆映射将 TSG 源值 x 映射到非负值 u ：

$$u = Q(x) = \begin{cases} 2x, & x \geq 0 \\ -2x-1, & x < 0 \end{cases} \quad (4)$$

即，等效于将 u 视作重新排序的字母表的索引 $\{0, -1, +1, -2, +2, \dots\}$ 。新源 u 具有接近几何源的概率分布，对它而言 Golomb 码是最优的，因为它们几何源的 Huffman 码，只要适当地选择 Golomb 参数。

Golomb/Rice (G/R) 码的一个示例示于表 1，示出参数 m 的几个值。应该注意，当 m 等于二的幂时，使用了参数 k ，它通过 $m=2^k$ 与 m 相关。G/R 码优于 Huffman 码的主要优点在于对于任意输入值，二进制码字可通过简单的规则来计算。因此，

不需要存储表。这对现代处理器特别有用，对它们而言，从存储表条目的存储位置读取可能比执行若干指令花更长时间。易于明白，参数 m 确定多少连续的码字具有相同的比特数。这也表明计算码字涉及计算 u/m 。其中 u 是输入值。对于大多数处理器，整数除法采用多次循环，因而 G/R 码对于一般的 m 并没有吸引力。当选择了对应于 Rice 码的 $m=2^k$ ，则除法 u/m 可用位移来代替，因为 $u/m=u \gg k$ （其中 \gg 表示右移运算符）。从而，对于任意输入 u 计算 G/R 码是容易的；只需计算 $p=u \gg k$ 和 $v=u-(p \ll k)$ 。随后通过将具有 p 个 1 的串与 v 的 k 位二进制表示串接来形成代码。

输入值	m=1 k=0	m=2 k=1	m=3	m=4 k=2	m=5	...	m=8 k=3
0	0	00	00	000	000		0000
1	10	01	010	001	001		0001
2	110	100	011	010	010		0010
3	1110	101	100	011	0110		0011
4	11110	1100	1010	1000	0111		0100
5	111110	1101	1011	1001	1000		0101
6	1111110	11100	1100	1010	1001		0110
7	11111110	11101	11010	1011	1010		0111
8	111111110	111100	11011	11000	10110		10000
9	1111111110	111101	11100	11001	10111		10001
10	11111111110	1111100	111010	11010	11000		10010
11	111111111110	1111101	111011	11011	11001		10011
12	1111111111110	11111100	111100	111000	11010		10100
13	11111111111110	11111101	1111010	111001	110110		10101
	:	:	:	:	:		:

表 1

10

从表 1 中显而易见，G/R 参数 k 的选择必须依赖于源的统计量。当 u 增加时，概率衰减得越慢，因此应选择更大的 k 。否则，码字长度增长过太。选择 k 的简单规则是：对于给定输入值 u 的码字长度应接近于该值出现概率以 2 为底的对数。

尽管 G/R 码对于几何分布源是最优的，但对于经由公式 4 的映射从拉普拉斯/TSG 源编码码元并非最优。这是因为对于具有 TSG 分布的输入变量 x ，从公式 4 得到的变量 u 具有接近但不完全是几何的概率分布。实际上，性能足够接近于最优（例如，一般具有比熵少 5% 的速率），因此 G/R 码相当流行。TSG 源的最优码包
5 括一组四个代码变量，在大多数情况下，实现它们且按 5% 或更少来改进压缩更为复杂。因此，在大多数情况下，G/R 编码器在性能和简便之间提供最佳折衷。

在图 1 中，概率分布由单个参数表示，即指数衰减速率。衰减速率越快，则更可能是零值。这意味着在许多情况下零是很可能的，使得几个零连串变得非常可能。换言之，如果概率分布衰减速率足够快，则编码连串是个好方法。编码零连串
10 意味着仅有几个比特用于处理输入数据中的许多项。

例如，如果数据与预测编码中预测器所使用的模型相匹配，则预测误差更可能为零。但即使有了好模型，偶尔有个大值也是可能的。这会在到达边界时发生，诸如像素值从背景值到前景值。不时地会出现大数字。当这种现象发生时，比游程编码更有用的一种类型的编码技术被称为“游程 Golomb/Rice (RLGR)”编码技术。
15 一种这样的 RLFT 编码技术在美国专利第 6,771,828 号，Malvar 的题为“System and Method for Progressively Transform Coding Digital Data (用于渐进变换编码数字数据的系统和方法)”和美国专利第 6,477,280 号，Malvar 的题为“Lossless Adaptive Encoding of Finite Alphabet Data (有限字母数据的无损自适应编码)”中公开。

事实上，随着数据源变化，概率将不保持常量而随时间变化。这对于例如图
20 像和音频都是真实的。通常，这些输入数据中的概率变化可用大量不同的方法来处理。在 JPEG 中，例如，有一熵编码器 (Huffman 编码器)，其中不同长度的码字用于要编码的不同值。Huffman 表通常是预先设计的，即，通常是获得大量图像、测试它们的概率、并构造用于所有图像的平均模型。这种方法的一个问题是对于图像的每个部分，编码效率上都有损耗，因为熵编码器使用的概率模型平均上很好，
25 但对图像的某个部分不够好。

从表 1 可看出，Golomb/Rice 码有两个主要问题：(1) 概率衰减参数 θ ，或者等价的概率 $P(x=0)$ 必须已知，因此 k 的近似值才能确定；以及 (2) 如果衰减参数太小，则熵 $H(x)$ 比 1 小，且因而 Golomb/Rice 码不是最理想的，因为其平均码字长度不能小于 1 比特/码元。

30 实际上，第一个问题 (最优 Golomb/Rice 参数的估算) 通常是通过将输入向量划分成预定长度的块来解决的。对于每个块，编码器两次通过数据。第一次通过

时，计算输入值的平均量。为此，参数 θ 可从公式 2 估算，且可确定相应的最优 k 。在第二次通过时，编码器通过首先输出二进制形式的 k 的值，随后是块内数据值的 Golomb/Rice 码的串接串，来生成该块的比特流。这是实际上所有使用 Golomb/Rice 码的无损压缩系统中都使用的方法，诸如用于无损图像压缩的 JPEG-LS、用于无损

5 音频压缩的 SHORTEN 及其它等。这被称为“按块自适应”或者“前向自适应”模型。前向自适应模型在以下意义上是前向的：编码器在编码前先考察数据、测量统计参数（通常是平均量）、随后基于该参数编码并将用于编码数据的参数值放入首部，以便由解码器使用。数据被拆成小部分，即块，而不是试图同时编码所有数据。对于每个块，测量该块的统计量，对于与缓冲区的内容相匹配的数据部分测量统计参

10 数，且将熵编码器调整到该参数。在编码文件中，插入首部，指示用于编码该数据块的参数值。

实践中的第二个问题，即，具有非常低的熵的编码源，通常是使用按块自适应或前向自适应模型来解决的，且如果块中输入码元的平均量的值小到足以使估算的熵 $H(x)$ 小于 1，则编码器使用游程编码而非 Golomb/Rice 编码。

15 尽管这些方法在实践中用得很好，但有两个主要缺点。一个缺点是编码器需要两次读取每个输入块，这样在数据上执行两次通过：第一次计算平均量以确定 Golomb/Rice 参数，而第二次执行真正的编码。这要求编码器执行附加的工作并增加了复杂性。在某些应用中，编码时间不是问题，但对于例如数码相机而言，它会减慢编码过程或增加随机存取存储器的开销。具体而言，前向自适应模型必须先考

20 察数据并测量统计量，找出模型参数，然后编码。如果编码器在具有大量处理能力的个人计算机上运行，则这不是个问题。然而，如果用手机拍照，照片要由手机自己编码，而其处理能力相当有限。

第二个但最重要的缺点涉及选择块尺寸的难度。如果块尺寸太大，则统计量在块内会剧烈变化。另一方面，如果块尺寸太小，则必须告知解码器哪个参数用于

25 编码该数据块的额外开销变得难以承担。对于每个块，编码器必须存储正用于编码该块的参数值。从某种观点看，编码小块所需要的额外开销对于所得到的压缩是不值得的。这就造成了权衡。一方面，如果用了小块，则块的统计量可匹配，但测量统计量是困难的，因为只有几个数，且编码的额外开销是巨大的。另一方面，如果使用大块，则问题是统计量在块内变动巨大。实际上，很难找到这两个矛盾因素之

30 间的折衷，因此块尺寸通常被选择在 128 和 1,048 个样值之间，取决于要编码的数据类型。

一种解决方案是在编码器中使用后向自适应技术。采用后向自适应，编码始于解码器和编码器对每个块达成初始状态的一致。换言之，每个参数被初始化为预定值，随后编码开始。每次当编码器产生一输出码元，该码元即可立即被发送至解码器，因为解码器知道用于编码它的参数值。在编码器输出码元后，它就按照预定的自适应规则计算用于编码参数的新值，这取决于所输出的码元。解码器知道该参数自适应规则，因此它也可计算用于编码参数的新值。因而，编码参数可在每个编码的码元后调整，且编码器和解码器始终是同步的，即，解码器追踪编码参数中的变化。这意味着编码器不需要向解码器发送有关什么参数用于编码该数据方面的任何额外开销信息。

因此，所需的是一种提供有效压缩并能处理和编码可能出现的任何输入整数的无损 Golomb/Rice (G/R) 编码器和解码器（编解码器）和方法。此外，还需要的是一种通过使用后向自适应技术提供对输入数据的快速追踪和有效压缩来避免上述前向自适应所存在问题的自适应 G/R 编解码器和方法。

15 发明内容

此处所公开的本发明包括用于整数数据无损编码和解码的自适应 Golomb/Rice (G/R) 编码器和解码器（编解码器）和方法。该自适应 G/R 编解码器和方法使用了具有新颖的自适应规则的新颖后向自适应技术。使用后向自适应，该自适应 G/R 编解码器和方法快速获知输入数据的统计量中的任何变化。此外，该自适应 G/R 编解码器和方法能够编码任何输入整数值。

该自适应 G/R 编解码器和方法还使用了在每个编码码元后调整编码参数的新颖的自适应规则。不需要概率表或码字表，因而该自适应 G/R 编解码器和方法可适用于小存储器的情况。该自适应 G/R 编解码器和方法因此很适于现代处理器，其存储器访问通常比指令取出和执行花费更多的周期。它也适于只有有限存储器和有限处理能力的小型设备，因为该自适应 G/R 编解码器和方法不需要缓冲块中的输入数据，也不需要两次处理每个数据值。

该自适应 G/R 和方法的主要优点之一是在生成的每个码字后调整和更新 G/R 参数(k)。这使得输入数据的统计量中的任何变化都能被很快追踪到。不需要将 G/R 参数发送到解码器的额外开销，因为它们的变化都能被解码器追踪。由于自适应规则是简单的，使用后向自适应的计算复杂度很低。因而，该自适应 G/R 编解码器和方法对许多实际应用都有吸引力。

该自适应 G/R 方法包括使用编码和自适应规则。编码规则规定下一输入值 x 是通过先经简单的 1-1 映射规则（当 $x > 0$ 时 $u = 2x$ ，当 $x < 0$ 时 $u = -2x-1$ ）将它映射到非负值 u 上、然后使用带有参数 k 的 Golomb/Rice 编码器编码来编码的，因此输出码字被表示为 $GR(u,k)$ 。

- 5 在编码了码元后，使用自适应规则。该自适应 G/R 方法使用简单而新颖的自适应规则。对于 k 的自适应规则如下。从输入值 u （记住 G/R 编码器总是在 u 值上运算），通过 $p = u \gg k$ 计算临时值 p （其中 \gg 表示右移运算符）。如果 $p = 0$ ，则 k 的经比例缩放的形式，称为 K ，减少第五整数常数 $B3$ 。如果 $p = 1$ ，则 k 保持不变。如果 $p > 1$ ，则 K 增加 p 。以这种方法，在生成每个码字后，参数 k 对于第一
10 和第二模式中的 G/R 编码器而更新。随后计算用于生成下一码字的 k 值为 $k = K/L$ ，其中 L 是固定参数（记住如果 L 选为二的幂，则除以 L 的除法只是位移运算）。

- 从以上自适应规则的描述中可以看到，该自适应 G/R 方法还可包括称为“小数自适应”的特征。小数自适应允许对自适应速度的更精细控制。首先，定义比例缩放参数 L ，且 L 的值一般设为二的幂。接着，定义经比例缩放的 G/R 参数 $K = k * L$ 。
15 当对 k 使用自适应规则时，经比例缩放的参数值 K 根据所生成的码字增加或减少整数常数。在 K 的自适应后，最终的参数值 k 通过 $k = K/L$ 来计算。用这种方式， K 的整数增量可看作 k 的小数增量，它允许对 k 值的更平滑控制，从而能更好地追踪输入统计量中的变化。如果 k 在每个编码码元后按整数增量调整，其值将波动太大。参数值中的这种噪声将导致压缩比（以直接二进制形式存储输入数据所需的比特数与存储已编码比特流所需的比特数之比）的下降。在一经测试的实施例中，比例缩放参数等于十六，且 G/R 参数的值基于数字数据的衰减参数。
20 一种自适应 G/R 编码器包括结合上述自适应 G/R 方法的模块和装置。

- 数字整数数据包括具有值的整数向量。值是这样的，对每个值，最可能的值是零，且非零值具有随非零值增加而减少的概率。该自适应 G/R 方法还包括用于
25 编码和解码数据的过程。该过程包括使用自适应 Golomb/Rice (G/R) 编码和 G/R 参数 k 来编码数字整数数据的每个值 x ，以及定义小数 G/R 参数为 $K = k * L$ ，其中 L 是比例缩放参数。该过程还包括在编码了每个数字整数数据的每个值 x 后，使用具有自适应规则的后向自适应技术来更新小数 G/R 参数，以及将数字整数数据的编码值追加至比特流中。该过程还包括使用 G/R 解码器解码比特流以精确地恢复
30 数字整数数据的每个值 x 。

自适应 G/R 解码器和方法通过使用对应于以上编码规则的解码规则并使用上

述相同的自适应规则来工作。解码器处的解码规则反转了前述的编码器处的编码规则。即，解码器根据 GR 参数 k 的当前值，按所需多少从输入比特流（或文件）中读取多个比特。以这种方法，解码器按照表 1 读取对应于有效的 Golomb/Rice 码 $GR(u,k)$ 的完整码字。由于 Golomb/Rice 码对每个参数 k 是唯一可解码的，因此解
5 码器随后可解码该码字。换言之，解码器可确定目前在解码器处的码元值 u 。从 u ，解码器可简单地通过使用 1-1 逆映射规则来确定相应的数据值 x 。具体而言，如果 u 是偶数，则 $x = u/2$ ，而当 u 是奇数时，则 $x = -(u+1)/2$ 。执行上述解码过程来将输入码字解码成完全匹配在编码器处所见到的输出值或值串。这样，解码过程是无损的。

10 在如上所述地解码了来自输入比特流或文件的码字后，解码器则如以上对编码器所述的相同自适应规则地计算。以这种方式，解码器将以与编码器完全相同的方法调整参数值 k 。从而，参数将具有用于解码下一比特流（或文件）码字的正确值。

15 附图说明

本发明可通过参考以下描述和示出本发明各方面的附图来进一步理解。通过结合附图阅读以下本发明的详细描述，其它特征和优点将是显而易见的，附图作为示例示出了本发明的原理。

现在参考附图，其中相同的参考标号通篇表示相应的部件：

20 图 1 示出适用于此处所公开的自适应游程 Golomb/Rice (RLGR) 编码器和方法的整数值的典型概率分布。

图 2A 是示出此处所公开的自适应 Golomb/Rice (G/R) 编解码器和方法的编码器部分的示例性实现的框图。

25 图 2B 是示出此处所公开的自适应 Golomb/Rice (G/R) 编解码器和方法的解码器部分的示例性实现的框图。

图 3 示出其中可实现图 2 所示的自适应 G/R 编解码器的合适的计算系统环境的示例。

图 4 是示出图 2 所示的自适应 G/R 编码器的组件的概括框图。

30 图 5 是示出图 2 和 4 所示的自适应 G/R 编码器和方法的概括操作的概括流程图。

图 6 是进一步示出图 5 所示的自适应 G/R 编码器和方法的细节的流程图。

图 7 是图 4 所示的自适应 G/R 编解码器和方法的 Golomb/Rice (G/R) 参数自适应模块的操作的详细流程图。

图 8 是图 7 所示的 Golomb/Rice (G/R) 参数自适应模块所使用的自适应值的计算的流程图。

- 5 图 9 是一工作示例，示出图 2 和 4 所示的自适应 G/R 编码器的编码细节，包括 G/R 参数 k 自适应规则。

具体实施方式

- 10 在本发明的以下描述中，对附图进行了参考，附图构成本发明的一部分，其中作为说明给出可实施本发明的特定示例。要理解，也可使用其它实施例且可作出结构性变化，而不脱离本发明的范围。

I. 介绍

- 15 此处所公开的自适应 Golomb/Rice (G/R) 编解码器和方法可用于各种各样的压缩应用中。例如，该自适应 G/R 编解码器和方法可用于数据库应用来编码索引。索引一般是具有类似于图 1 的概率分布的正整数数字，因为对于索引，小值比大值更可能。另一示例是将该自适应 G/R 编解码器和方法用于编码硬盘的磁头位置。直到硬盘满之前，数据更可能位于硬盘开始处而不是末尾处。因此，小磁头值比大磁头值更可能，因而整数数据具有类似于图 1 的概率分布。

- 20 这里所公开的 RLGR 编解码器和方法是用于整数数据的无损压缩的改进技术。包含整数值的向量由编码器映射到比特流中，它稍后可由解码器精确重建。为改进性能使用后向自适应，该自适应 G/R 编解码器和方法快速获知和自适应输入数据的统计量中的变化。

- 25 该自适应 G/R 编解码器和方法使用了一种在每个编码的码元后调整 G/R 参数的后向自适应策略。概率表或码字表是不需要的，它们使自适应 G/R 编解码器和方法适合于很小的存储器情况。因而该自适应 G/R 编解码器和方法特别适合现代处理器，其中存储器访问通常比指令取出和执行花费更多的周期。

- 30 该自适应 G/R 编解码器和方法与以前各种熵编码器相比的一个关键优点是其后向自适应策略快速获知数据的统计量中的变化。因而，实际上该自适应 G/R 编解码器和方法展示出比诸如 Huffman 编码器、块自适应 Golomb/Rice 编码器或上下文自适应算术编码器等其它各种编码器更好的性能。使用后向自适应策略用于编码

参数的另一个优点是不需要概率估算器。该自适应 G/R 编解码器和方法还有另一个优点是它在每个编码的码元之后执行自适应，只在数据上通过一次，这样比使用按块或前向自适应的编码器产生更好的压缩结果和更快的编码。

5 II. 概览

图 2A 和 B 是方框图，示出此处所公开的自适应 Golomb/Rice (G/R) 编解码器和方法的示例性实现。在图 2A 中，示出了自适应 G/R 编解码器和方法的编码器部分的框图。在图 2B 中，示出了自适应 G/R 编解码器和方法的解码器部分的框图。应该注意，图 2A 和 B 只是其中可实现和使用自适应 G/R 编解码器和方法的几种
10 方式中的两种。

参考图 2A，自适应 G/R 编码器 200 运行在第一计算设备 210 上。自适应 G/R 编码器 200 输入并处理整数数据 220。一般而言，给定整数数据 220，诸如包含整数数值的向量，自适应 G/R 编码器 200 将整数数据 220 编码或映射为已编码比特流 230。整数数据 220 一般包含整数向量，因而大多数可能值为零且任意非零值具有
15 随值增长而减少的概率。这种类型的整数数据一般具有类似于图 1 所示的概率分布函数 (PDF)。在编码了整数数据后，已编码比特流 230 可被存储或发送。

参考图 2B，G/R 解码器 240 驻留在第二计算设备 250 上。应该注意，尽管示出为单独的计算设备，但第一计算设备 210 和第二计算设备 250 可以是同一计算设备。换言之，G/R 编码器 200 和解码器 240 可驻留在同一计算设备上。一般而言，
20 G/R 解码器 240 处理编码器比特流 230 并输出经重建的整数数据 260。因为自适应 G/R 编码器 200 执行整数数据 220 的无损编码，因此 G/R 解码器 240 可读取已编码比特流 230，并精确地重建包含在整数数据 220 中的原始数据向量。

应该注意，在实际应用中，设备或仪器可包含 G/R 编码器而不包含 G/R 解码器（例如，数码相机）。同样地，设备或仪器可包含 G/R 解码器而不包含 G/R 编
25 码器（例如，数字音频播放器或数字图像查看器）。

III. 示例性操作环境

自适应 Golomb/Rice (G/R) 编解码器和方法被设计为在计算环境中在计算设备上运行，诸如图 2 所示的第一计算设备 210 和第二计算设备 250。现在将讨论自
30 适应 G/R 编解码器和方法在其中运行的计算环境。以下讨论旨在提供可在其中实现自适应 Golomb/Rice (G/R) 编解码器和方法的适当计算环境的简要概括描述。

图 3 示出可在其中实现图 2 所示的自适应 G/R 编解码器和方法的适当计算系统环境的示例。计算系统环境 300 只是适当计算环境的一个示例，而不旨在对本发明使用范围或功能提出任何限制。计算环境 300 不应被解释为对于示例性操作环境 300 中所示任意一个组件或其组合具有任何依赖或需求。

5 自适应 G/R 编解码器和方法可用大量其它通用或专用计算系统环境或配置来操作。众所周知的适于自适应 G/R 编解码器和方法使用的计算系统、环境和/或配置的示例包括，但不限于，个人计算机、服务器计算机、诸如蜂窝电话和 PDA 等手持式、膝上型或移动计算机或通信设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、网络 PC、小型机、大型计算机、可包括任意以上
10 系统或设备的分布式计算环境等等。

自适应 G/R 编解码器和方法可在诸如程序模块等由计算机执行的计算机可执行指令的通用上下文环境中描述。一般而言，程序模块包括例程、程序、对象、组件、数据结构等，它们执行特定任务或实现特定抽象数据类型。自适应 G/R 编解码器和方法还可在分布式计算环境中实施，其中任务由通过通信网络连接的远程处
15 理设备执行。在分布式计算环境中，程序模块可位于包括存储器存储设备在内的本地和远程计算机存储介质中。参考图 3，用于实现自适应 G/R 编解码器和方法的示例性系统包括计算机 310 形式的通用计算设备。计算机 310 是图 2 所示第一计算设备 210 和第二计算设备 250 的示例。

计算机 310 的组件可包括，但不限于，处理单元 320、系统存储器 330 和将包
20 括系统存储器的各种系统组件耦合到处理单元 320 的系统总线 321。系统总线 321 可以是任意几种类型的总线结构的任一种，包括存储器总线或存储器控制器、外围总线或外部总线、和/或使用任意各种各样总线体系结构的任一种的局部总线。作为示例，而非限制，这样的体系结构包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MSA) 总线、扩展 ISA (EISA) 总线、视频电子技术标准协会 (VESA)
25 局部总线以及外围部件互连 (PCI) 总线，也称为 Mezzanine 总线。

计算机 310 通常包括各种计算机可读介质。计算机可读介质可以是能够被计算机访问的任何可用介质，包括易失性和非易失性介质、可移动和不可移动介质。作为示例，而非限制，计算机可读介质可包括计算机存储介质和通信介质。计算机存储介质包括以用于诸如计算机可读指令、数据结构、程序模块或其它数据等信息
30 的存储的方法或技术实现的任何易失性和非易失性、可移动和不可移动介质。

计算机存储介质包括，但不限于，RAM、ROM、EEPROM、闪存或其它存储

器技术、CD-ROM、数字多功能盘（DVD）或者其它光盘存储、磁盒、磁带、磁盘存储或者其它磁存储设备，或者可用于存储需要的信息并且可由计算机 310 访问的任何其它介质。通信介质一般具体化为在如载波或其它传送机制等已调制数据信号中的计算机可读指令、数据结构、程序模块或其它数据，并且包括任何信息传递
5 介质。

注意，术语“已调制数据信号”意指以将信息编码到该信号中的方式设置或改变其一个或多个特征的信号。作为示例，而非限制，通信介质包括有线介质，诸如有线网络或者直接线连接，以及无线介质，诸如声学、RF、红外和其它无线介质。任何上述内容的组合也包括在计算机可读介质的范围内。

10 系统存储器 330 包括以易失性和/或非易失性存储器形式的计算机存储介质，诸如只读存储器（ROM）331 和随机存取存储器（RAM）332。基本输入/输出系统 333（BIOS）一般存储在 ROM 331 中，它包含诸如在启动期间帮助在计算机 310 内各元件间传送信息的基本例程。RAM 332 一般包含由处理单元 320 可直接访问和/或目前正运行的数据和/或程序模块。作为示例而非限制，图 3 示出操作系统 334、
15 应用程序 335、其它程序模块 336 和程序数据 337。

计算机 310 还可包括其它可移动/不可移动、易失性/非易失性计算机存储介质。仅作为示例，图 3 示出读写不可移动非易失性磁介质的硬盘驱动器 341、读写可移动非易失性磁盘 352 的磁盘驱动器 351，以及读写如 CD-ROM 或其它光介质等可移动非易失性光盘 356 的光盘驱动器 355。

20 可在示例性操作环境中使用的其它可移动/不可移动、易失性/非易失性计算机存储介质包括，但不限于，磁带盒、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等等。硬盘驱动器 341 一般通过诸如接口 340 等不可移动存储器接口连接到系统总线 321，而磁盘驱动器 351 和光盘驱动器 355 一般通过诸如接口 350 等可移动存储器接口连接到系统总线 321。

25 以上讨论并在图 3 中示出的驱动器及其相关联的计算机存储介质为计算机 310 提供了计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 3 中，硬盘驱动器 341 被示出为存储操作系统 344、应用程序 345、其它程序模块 346 和程序数据 347。注意这些组件可以相同或不同于操作系统 334、应用程序 335、其它程序模块 336 和程序数据 337。操作系统 344、应用程序 345、其它程序模块
30 346 和程序数据 347 在这里给出不同的标号是为了指出至少它们是不同的副本。用户可通过诸如键盘 362 和定点设备 361（通常指鼠标、轨迹球或触摸板）等输入设

备将命令和信息输入到计算机 310。

其它输入设备（未示出）可包括话筒、操纵杆、游戏垫、圆盘式卫星天线、扫描仪、无线电接收器或者电视或广播视频接收器等等。这些和其它输入设备通过耦合到系统总线 321 的用户输入接口 360 连接到处理单元 320，但也可通过其它接口和总线结构，如并行端口、游戏端口或通用串行总线（USB）来连接。显示器 391 或其它类型的显示设备也通过接口，如视频接口 390 连接到系统总线 321。除了显示器以外，计算机还可包括其它外设输出设备，诸如扬声器 397 和打印机 396，它们可通过输出外围接口 395 连接。

计算机 310 可使用到一或多个远程计算机，如远程计算机 380 的逻辑连接在 10 网络化环境中工作。远程计算机 380 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它普通网络节点，并且一般包括上面相对于计算机 310 所述的许多或所有元件，尽管在图 3 中所示的只有存储器存储设备 381。图 3 所示的逻辑连接包括局域网（LAN）371 和广域网（WAN）373，但也可包括其它网络。这样的网络环境在办公室、企业范围计算机网络、企业内部互联网和因特网中是很常见的。

当在 LAN 环境中使用时，计算机 310 通过网络接口或适配器 370 连接到 LAN 371。当在广域网环境中使用时，计算机 310 一般包括调制解调器 372 或通过 WAN 373，如因特网建立通信的其它装置。调制解调器 372 可以是内置或外置的，通过用户输入接口 360 或其它适当机制连接到系统总线 321。在网络化环境中，相对于 20 计算机 310 描述的模块或其部分可存储在远程存储器存储设备中。作为示例而非限制，图 3 例示了远程应用程序 385 驻留在存储器设备 381 上。要理解，所示的网络连接是示例性的，并且可使用在计算机之间建立通信链路的其它手段。

IV. 系统组件

25 图 4 是示出图 2 所示的自适应 G/R 编码器 200 组件的总框图。自适应 G/R 编码器接收输入值（或值串）400 作为输入。使用 Golomb/Rice 编码模块 410 来编码输入值（或串）400 以获得码字 420。在每个输入值（或串）400 的编码后，编码参数被自适应来追踪输入数据的统计量。

30 Golomb/Rice（G/R）参数自适应模块 430 用于使用后向自适应技术和新颖的自适应规则来更新原始 G/R 参数。这产生经更新的 G/R 参数 445。G/R 参数的自适应将在下面详细讨论。一旦参数被更新，下一个输入值 450 将由自适应 G/R 编码

器 200 用更新后的 G/R 参数 440 来处理。

V. 操作概览

5 现在将讨论如图 2 和 4 所示的自适应 G/R 编码器 200 的操作和其中所使用的方法。图 5 是总流程图，示出图 2 和 4 所示的自适应 G/R 编码器和方法的总操作。该方法始于输入要编码的数字数据（框 500）。在一个经测试的实施例中，输入数字数据是具有整数值元素的向量形式的整数数据。应该注意，每个输入数字数据值可以是任意整数值，而限于特定的范围（例如，二进制或二进制加符号，因为这

10 在其它熵编码器中很常见）。接下来，使用 G/R 参数来编码数字数据（框 510）。数字数据是使用被初始化为某一值的 G/R 参数来编码的。然而，由于输入数字数据的统计量可以变化，因此 G/R 编码器 200 是自适应的。这种自适应使得自适应 G/R 编码器 200 能够追踪输入数字数据的统计量并快速自适应到那些统计量，以提供更高的编码效率。自适应 G/R 编码器 200 和方法使用后向自适应技术来更新 G/R 参数（框 520）。G/R 参数的这种更新在编码了输入数字数据的每个值或值串后发生。此外，后向自适应技术包括新颖的自适应规则，这将在下面详细讨论。然后，输出已编码数字数据（框 530）。接下来，使用刚才描述的方法处理输入数字数据的下一值或串。G/R 参数的更新值用于编码下一输入值或串。一直重复这个过程，直到所有数字数据都被编码成已编码比特流。

20 图 6 是一流程图，示出图 5 所示的自适应 G/R 编码器和方法的进一步细节。特别地，数字数据的值或串作为输入接收（框 600）。接下来，使用 G/R 参数编码输入值或串（框 610）。在编码了输入值或串后，更新 G/R 参数。该自适应过程始于定义一经比例缩放的 G/R 参数（框 520）。该经比例缩放的 G/R 参数用来减慢 G/R 参数的自适应，这样可更紧密地追踪最优参数值。经比例缩放的 G/R 参数将在下面更详细地讨论。接下来，使用后向自适应技术和新颖的自适应规则来更新经比例缩放的 G/R 参数（框 630）。经编码的输入值或串被追加到已编码比特流后，且输入要编码的下一个数字数据值或串（框 640）。该过程重新开始，用更新后的经比例缩放的 G/R 参数编码下一值或串。

VI. 操作细节

30 现在将讨论以上所述图 4、5 和 6 的自适应 G/R 编码器 200 和方法的操作细节。

Golomb/Rice (G/R) 参数自适应模块

图 7 是图 4 所示的自适应 G/R 编码器 200 和方法的 Golomb/Rice (G/R) 参数自适应模块 430 的操作的详细流程图。一般而言, G/R 参数自适应模块 430 使用具有新颖的自适应规则的后向自适应技术来更新初始 G/R 参数。更新是在编码数字数据的每个值或串后执行的。

该操作始于接收初始 G/R 参数 (框 705) 和自适应值 (框 710) 作为输入, 稍后将描述其计算。随后做出关于自适应值是否等于零的判断 (框 715)。如果是, 则自适应规则将使经比例缩放的 G/R 参数减少一整数常数 (框 720)。

如果自适应值不等于零, 则做出关于自适应值是否等于一的判断 (框 725)。如果是, 则自适应规则保持经比例缩放的 G/R 参数不变 (框 730)。如果不是, 则自适应规则将经比例缩放的 G/R 参数增加自适应值 (框 735)。

一旦 G/R 参数已被更新, 用更新后的 G/R 参数替换当前 G/R 参数 (框 740)。这可通过将经比例缩放的 G/R 模式参数除以固定比例缩放因子并保留结果的整数部分来获得。由于自适应按照整数步长调整经比例缩放的 G/R 模式参数, 因此实际 G/R 参数就象它是由小数步长来自适应的一样。再一次, 这是“小数自适应”的示例, 它允许对自适应速度的更精细控制。当然, 如果 G/R 参数保持不变 (框 730), 则不进行更新, 且当前 G/R 参数是相同的。最后, 输出更新后的 G/R 参数 (框 745)。

图 8 是图 7 所示的 Golomb/Rice (G/R) 参数自适应模块 430 使用的自适应值的计算的详细流程图。参考图 7 和 9, 自适应值计算模块 800 产生作为图 7 的流程图的输入的自适应值 (框 710)。该操作始于接收两个输入, 即当前 G/R 参数值 (框 805) 和输入值 (框 810)。接着, 输入值右移与 G/R 参数值相同的位数 (框 820)。结果值就是自适应值, 随后输出它 (框 830)。

VII. 工作示例

为了更全面地理解这里所公开的自适应 G/R 编码器和方法, 给出示例性工作示例的操作细节。应该注意, 这个工作示例只是其中可实现自适应 G/R 编码器和方法的一种方法。

自适应 Golomb/Rice (G/R) 编解码器和方法是在以上提到的美国专利第 6,477,280 号中公开的 PTC 熵编码器的扩展。然而, 美国专利第 6,477,280 号的 PTC 熵编码器是用于编码二进制数据的 (一般是整数数据的位平面)。此处所公开的自适应 G/R 编码器和方法可使用任意输入值来编码整数数据。换言之, 此处所公开

的自适应 G/R 编码器和方法可编码任意字母的数据。

5 此处所公开的自适应 G/R 编码器和方法的一个优点是，不象 PTC 熵编码器，不需要了解输入数据的最大可能数。相反，自适应 G/R 编码器和方法可处理任意大小的输入值，不论有多大。这意味着自适应 G/R 编码器假设输入数据具有如图 1 所示的拉普拉斯分布，当突然有大数字出现在输入数据中，自适应 G/R 编码器和方法能够编码这个大数字。尽管大数字将比小数字使用更多比特，但大数字仍将被编码。然而，使用更多比特只在大数字出现时才自承其果，而不是对每个其它值。这归因于以下提出的新模式选择和自适应规则。

10 用 PTC 熵编码器，接收输入数据、分解成位平面、然后用 G/R 编码器编码每个位平面。在此处所公开的自适应 G/R 编解码器和方法中，自适应 G/R 编解码器被扩展成直接处理拉普拉斯数据。这有自适应 G/R 编解码器和方法使用单次通过编码的优点，使得它比 PTC 熵编码器要快得多。

15 PTC 熵编码器的输入数据具有拉普拉斯分布，其中小数字更可能。有时，小数字有更多可能，使得编码零连串对于比特流的特定部分更加有效。然而，PTC 熵编码器将拾取数据，在最高位平面上进行一次通过，并回去在下一位平面上进行一次通过。例如，如果数据是 16 位，通过首先在位#16 上完成并编码。当然，大多数数据将是零，因为该位只对很大的数分割，然后继续向下。当到了位#5、4、3、2 和 1，这些位具有许多零和一，这意味着它到达了编码它们已没有什么帮助的一点。通常最低位是随机的，以至于用一个比特来编码这个位，即每个输入比特都直接复制到输出。PTC 熵编码带来的问题是位平面上的编码需要几次通过数据。具体而言，PTC 熵编码器不得不编码最高位、下一位、然后再下一位等等。显然，这花费了太大量的时间，而且在某些情况下，PTC 熵编码器比此处所公开的自适应 G/R 编码器和方法慢 1.5 到 3 倍。

25 编码规则

自适应 G/R 编解码器和方法使用基于 G/R 参数 k 的新颖的编码规则。表 2 给出自适应 G/R 编解码器和方法用于将整数值 x 映射到二进制比特流的编码规则。

自适应 Golomb/Rice	$x \geq 0: u = 2 x $ $x < 0: u = 2 x - 1$	代码 = GR(u,k)
-----------------	---	--------------

表 2

在这个工作示例中，定义映射值 u 。自适应 G/R 编码器和方法的输入值 x 可正可负。输入值 x 映射到值 u ，其中 u 只有正值。因而，带符号的输入值 x 被转换成不带符号的等价表示 u 。公式 4 给出从 x 到 u 的映射。具体而言，该映射表示，

5 0 映射到 0、1 映射到 1、-1 映射到 2、2 映射到 3、-2 映射到 4 等等，因而 u 值总是正的。这完成后，就可使用 G/R 表（表 1），因为 G/R 表只用于非零值。这种映射使得自适应 G/R 编解码器和方法能处理任何输入字母。换言之，由于使用了 G/R 表（它可处理任何输入数），因此输入字母可以是无限的，且自适应 G/R 编解码器和方法可处理任意大小的数字输入。自适应 G/R 编解码器和方法只受限于

10 操作系统可处理的数字的大小。应该注意，实际上，G/R 编码表 1 不需要存储在存储器中。很容易明白，表条目具有足够的结构，使得对于任意 u 值和编码参数 k ，都能方便地计算出码字。

给定 u 数据，表 2 规定使用自适应 G/R 编码器和表 1 例示的 G/R 编码规则来编码输入值 x 的映射值 u 。因而，用于编码 x 的码字基于 u 和 k 值。G/R 参数 k 是

15 使用后向自适应技术和创新自适应规则来更新的，如以下详细讨论。表 2 中的规则精确定义了编码器如何编码，这意味着解码器可使用表 2 中的同样规则来恢复（或解码）已编码数据。

小数自适应

20 小数自适应使用经比例缩放的 G/R 参数 K 来代替 G/R 参数 k 。小数自适应是减慢自适应的一种方法。使用没有小数自适应的自适应 G/R 编解码器和方法是可能的。然而，如果没有小数自适应，则自适应通常变化太快，而无法正确追踪用于输入数据的最优参数。

在这个工作示例中， k 参数的自适应是使用经比例缩放的 K 参数来完成的。

25 因而，更新 K 而非 k 。 k 和 K 之间的关系如下： $k=K/L$ ，其中 L 是如上解释的比例缩放参数。因此，当进行自适应时，自适应 K 值，而 K 除以 L 就获得 k 值。注意，所有值都是整数，所以通过 $k=K/L$ ，意味着是结果的整数部分。还要记住，固定比例缩放参数 L 被设为等于 2 的幂的值（例如 $L=16$ ），则除以 L 的除法可通过位移运算来有效执行。

30 小数自适应是较佳的，因为自适应 G/R 编解码器和方法对生成的每个代码都进行了 G/R 参数 k 的调整。换言之，在编码了输入值或串后，执行自适应规则。

如果 k 直接通过整数值变化来自适应, 则由于它们是整数, 所有可做的是保持相同或至少增减 1。然而, 假设输入数据变得更大, 意味着参数应该增加。

5 小数自适应允许 k 的小数增加。例如, k 可以用 .5 代替 1 来增加。然而, 这是不允许的, 因为 k 是整数参数。因此小数自适应执行 K 的整数增加, 且用 L 除以 K 给出 k 的小数增加。这保证了参数 k 没有振荡。

作为使用自适应, 定义一个标志是可能的, 这样如果数据有增减, 在增减该参数前通过则某些数量的编码周期。可定义追踪编码周期数的新参数。换言之, 该参数将追踪在参数改变前条件 (输入数据增加或减少) 发生了多少次。但是应该注意, 这一技术是经试验的, 且小数自适应提供了较好的结果。

10

Golomb/Rice (k) 参数自适应

15 G/R 参数 k 在编码了每个输入值或串后被自适应。当使用小数自适应时, 实际上自适应经比例缩放的 G/R 参数 K , 而非直接自适应 k 。表 3 给出用于 G/R 参数 k 的自适应规则。在编码了值 u 后, 通过自适应值 $p=u \gg k$ 来控制自适应, 这意味着 u 向右移 k 位。在自适应后, k 的值被设为 $k=K/L$, 其中 L 是常数。在这个工作示例中, $L=16$ 。

$p=0$	通过设置 $K \leftarrow K-B3$ 来减少 K
$p=1$	不改变
$p>1$	通过设置 $K \leftarrow K+p$ 来增加 K

表 3

20 表 1 的 G/R 码取决于参数 k 。例如, 如果一个不完全连串后的值为 13, 13 的 GR 码是“1111111111110” (对于 $k=0$), 且如果 $k=1$, 则为“11111101”。 k 越大, 则表示 13 的数将越小。而 k 越小, 则表示 13 的数将越大。因而, 参数 k 必须已知。这意味着如果对 k 选择了合适的值, 自适应 G/R 编解码器和方法会完成得很好。但是, 使用 k 的大值并不总是有利的, 因为这将对于输入数据的较小值产生较长串, 如表 1 所示。换言之, k 值的合适选择取决于输入数据。如果值为 13, 则使用 k 的大值是个好主意。但是, 假设不完全连串后的值为“1”。则 k 的较小值是所希望的。因而, 对于不完全连串之后的小值, 使用小 k 更好, 而对于大值, 使用大 k 更好。因此, k 的选择与值的概率有关。在现有技术中, 有对为此目的的大量理论

25

成果：如果输入数据的概率已知（例如，如果输入数据是拉普拉斯，其中存在控制衰减的单个参数），则有众所周知的公式，从中可计算要使用的衰减参数，即参数 k 。这平均起来使映射使用尽可能少的比特。

因而， k 参数是自适应的是很重要的。以这种方法，如果在输入数据上有大值出现，则 k 应该增加，因为对于大值， k 越大越好。另一方面，如果有较小值出现，则 k 应该减少。直观上，能够明白，对于大数字， k 应该增加，而对于小数字， k 应该减少。因此，只要 k 以足够小的幅度改变（诸如当使用小数自适应时），对于输入数据的最优参数将总是被正确地追踪。

表 3 所示的 k 的自适应规则是相当新颖的。在自适应 G/R 编解码器和方法中，任何值都可出现，因此这个值必须被编码。编码是使用自适应 G/R 编码器和 G/R 参数 k 来完成的。参考表 3，输入数据是 x 。输入数据 x 可以是任何整数，小 x 更可能（可以是正的或是负的）。但是，G/R 编码只用于正数。 x 的直接映射（见公式 4）用于将 x 映射成 u 。 k 的自适应是由自适应值 p 来控制的， p 被定义为 u 向右移 k 位。因而，自适应值 p 是 u 的按比例缩小形式。或者，等价地， p 参数是近似于 $u/2^k$ 的整数。向右移 k 位等于将此数除以 2^k 。例如，如果一个数向右移 5 位，这就与将此数除以 32（即 2^5 ）相同。余数被舍去，而只使用商。

参考表 3，如果自适应值 p 等于零，则 K 被更新并由 K 减少一整数常数 $B3$ 来替换。如果自适应值 p 等于一，则 K 不变。如果自适应值 p 大于一，则 K 被更新并由 K 减少自适应值 p 来替换。

如果 p 的自适应值等于一，这意味着 u 值接近于 2^k ，且就是对它们而言参数 k 是正确的。因此，如表 3 所示，没有变化。如果自适应值 p 的值是 0，这意味着输入值小于 2^k 。即要开始减少 k （因为输入值小于 2^k ）。自适应值 p 大于 1 的情况是不太可能的，因为输入值不可能非常大。但如果数字是大的且 $p > 1$ ，则要开始增加 k 参数。

25

自适应 G/R 编码器

图 9 是示出图 2 和 4 所示自适应 G/R 编码器 200 的编码细节的工作示例，包括 G/R 参数 k 自适应规则。该过程开始（框 905）于读取输入值 u （框 910）。自适应 G/R 编码器 200 的两个主要过程是 G/R 编码（框 915）和 G/R 参数自适应（框 920）。

30

G/R 编码过程 915 始于计算自适应值 p 和 v （框 925）。向比特流追加等于一

的 p 个比特（框 930）。 v 的 k 位二进制值则被追加在该比特流后（框 935）。这些操作包括如表 1 所定义的自适应 Golomb/Rice 编码器。

- 5 G/R 参数自适应过程 920 包括确定自适应值 p 是否等于一（框 940）。如果是，则自适应值 p 保持不变（点 945）。否则，做出关于自适应值 p 是否等于零的另一判断（框 950）。如果不是，则 K 被更新并用 K 减少一整数常数 $B3$ 来替换（框 955）。否则， K 被更新并用 K 增加自适应值 p 来替换（框 960）。最后，该过程设置 k 等于 K 除以 L （框 965），并且该过程结束（框 970）。

结果

- 10 该工作示例的自适应 G/R 编解码器和方法已在图像、音频和地图数据压缩的应用中实现。在这些应用中使用自适应 G/R 编解码器和方法的结果是压缩比可与最复杂熵编码器相比，但以更简单的实现方式。

- 特别地，对于用于整数数据的现有熵编码器，自适应 G/R 编解码器和方法对于大量如图 1 中的源码元概率分布达到了接近于理论最大值（由源熵所规定的）压缩比。作为示例，众所周知的 Golomb/Rice 和 Huffman 编码器只对于每码元 1 个比特或更高些的源熵有效。

VIII. 解码

- 20 自适应 G/R 编解码器和方法还包括可基于上述编码器精确实现的解码器。参考图 2B，计算设备（框 250）可以只实现 G/R 解码器 240。自适应 G/R 解码器 240 和方法从已编码比特流接收码字（框 230）。接着，自适应 G/R 解码器 240 通过应用上文对于自适应 G/R 编码器 200 所述的逆规则来解码码字。接下来，G/R 参数使用与自适应 G/R 编码器的规则完全相同的规则来自适应。最后，输出解码后的（或重建的）整数数据（框 260）。

- 25 由于编码规则是唯一可解码的，且解码器的自适应规则与编码器的相同，因此前面描述的编码规则和自适应规则也精确地描述了解码器的操作。

本发明的以上描述已为说明和描述之途而给出。它并不旨在穷尽本发明或将本发明限于所公开的精确形式。各种修改和变体鉴于以上示教都是可能的。本发明范围不受本发明详细描述所限，而只受所附权利要求书所限。

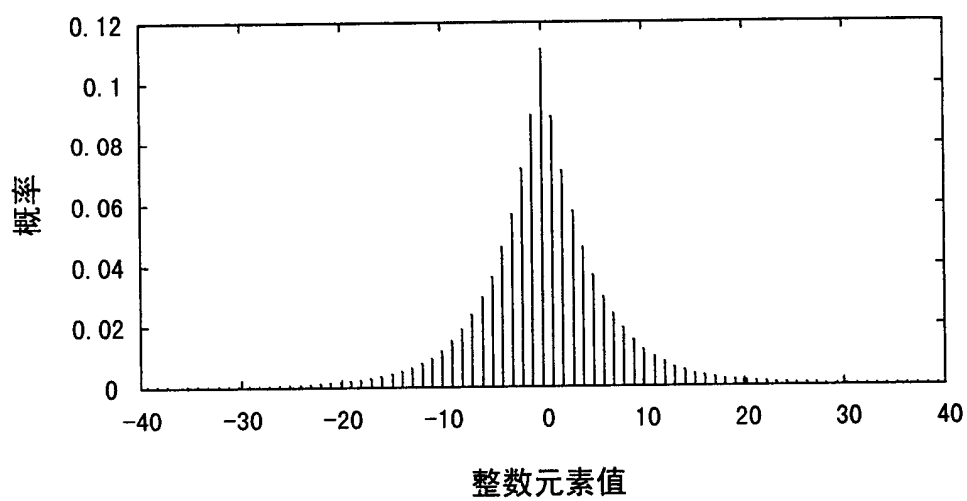


图 1

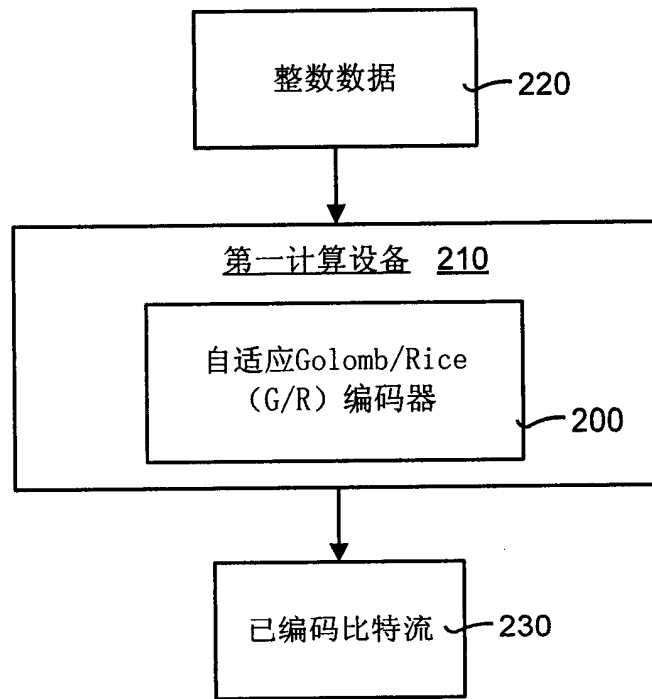


图 2A

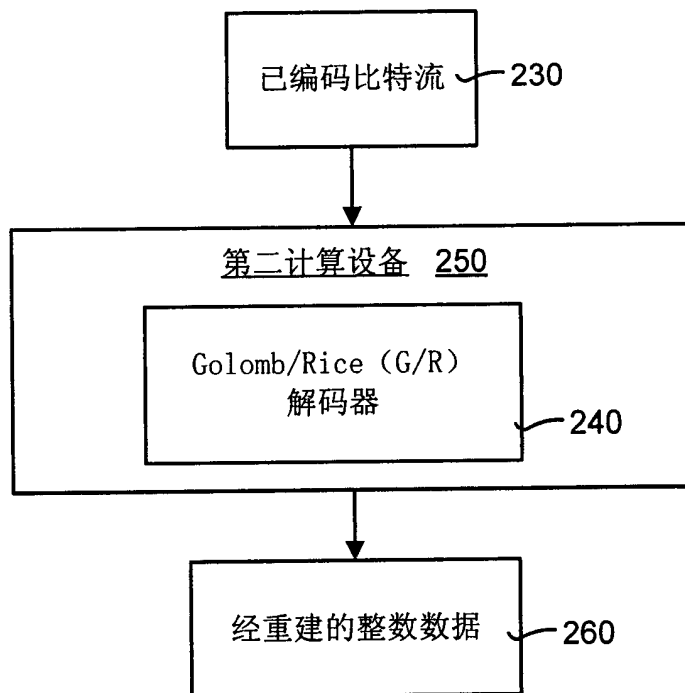


图 2B

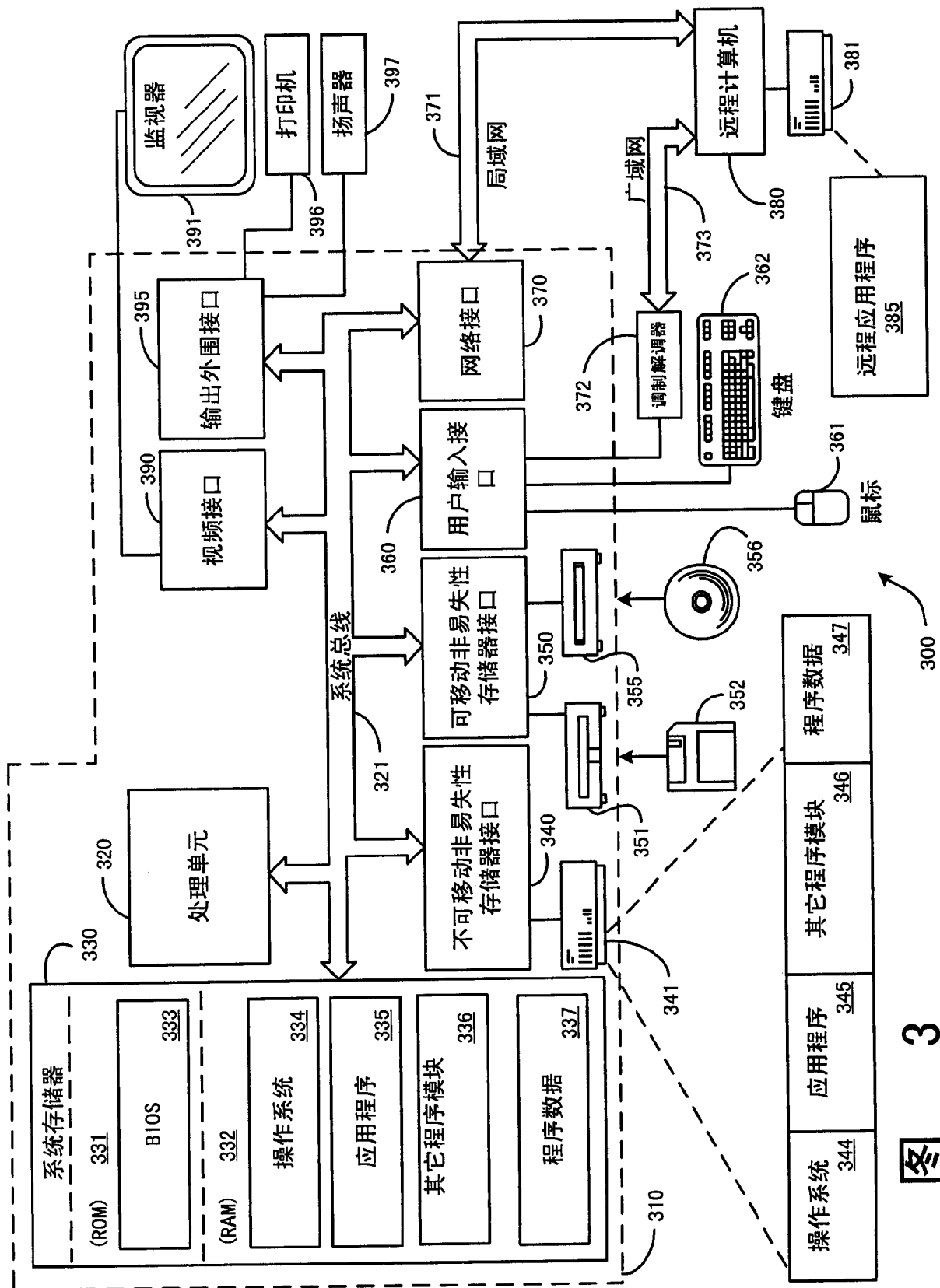


图 3

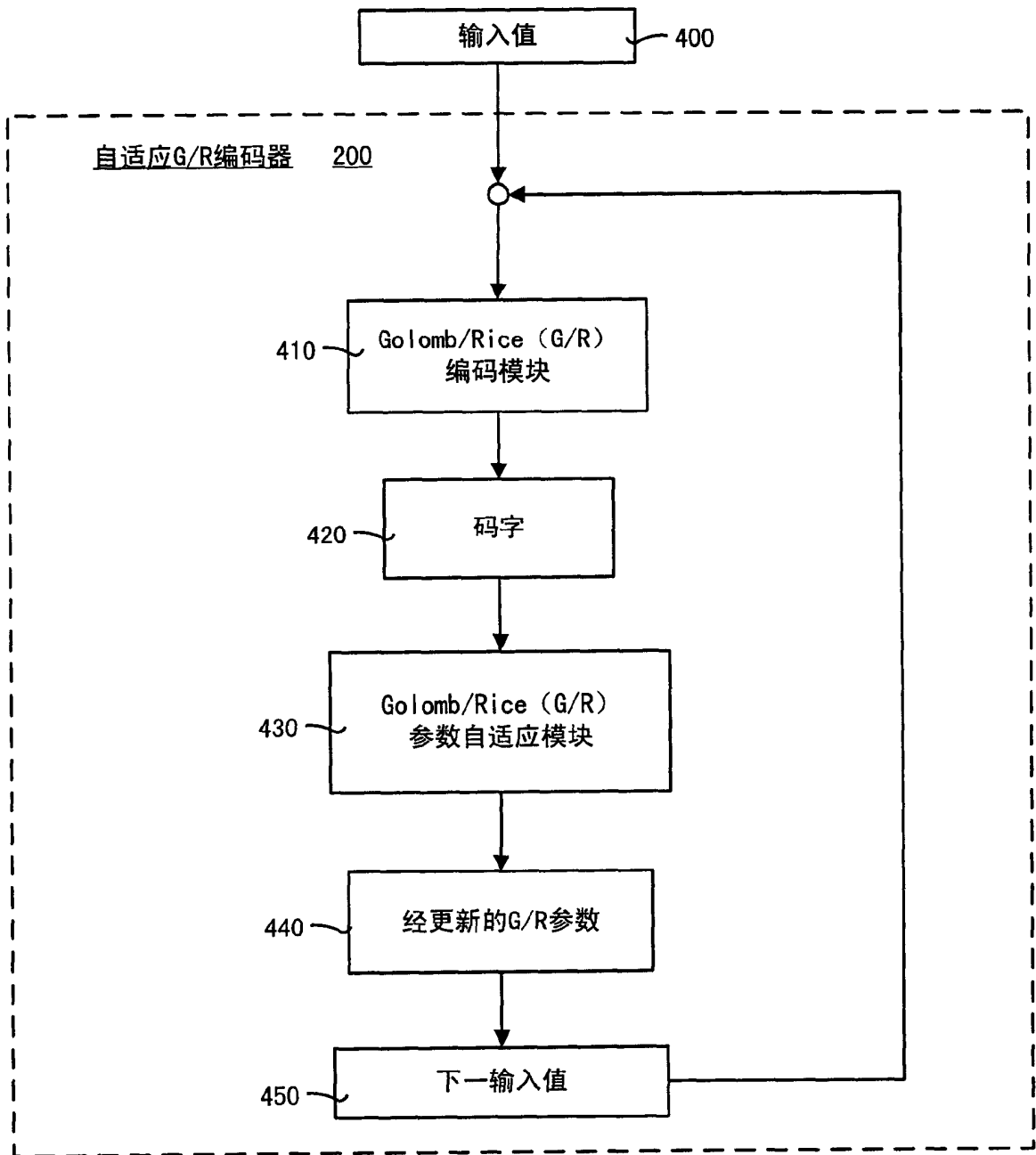


图 4

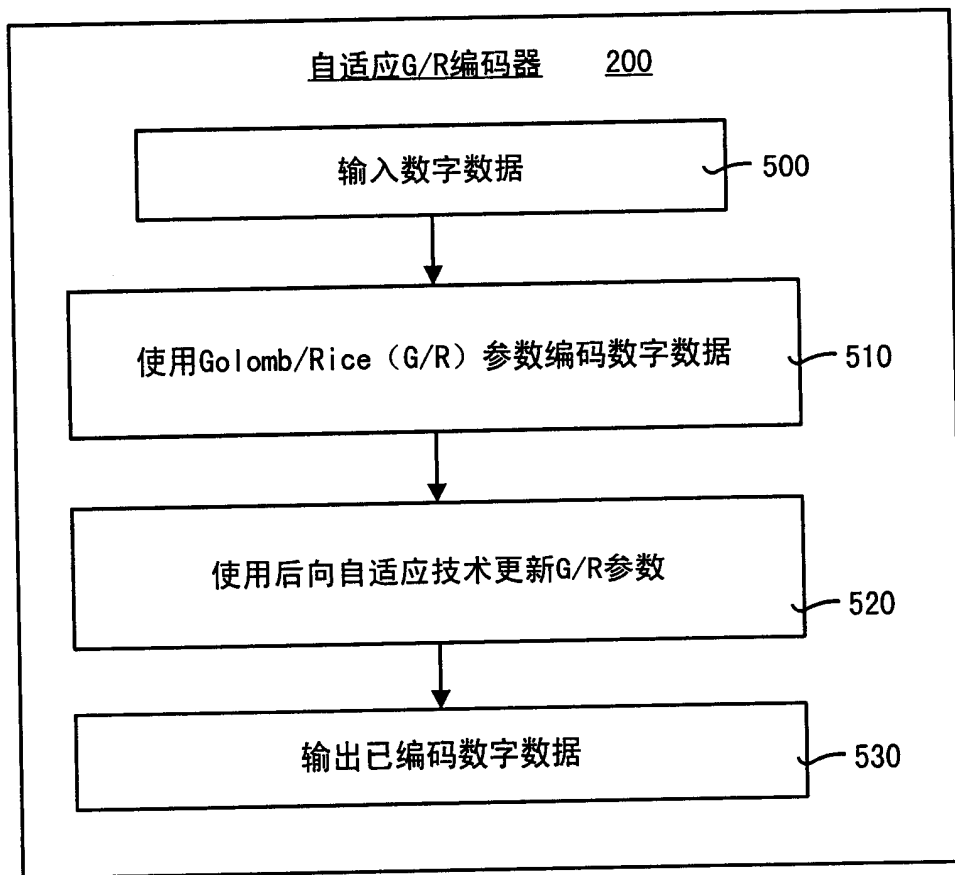


图 5

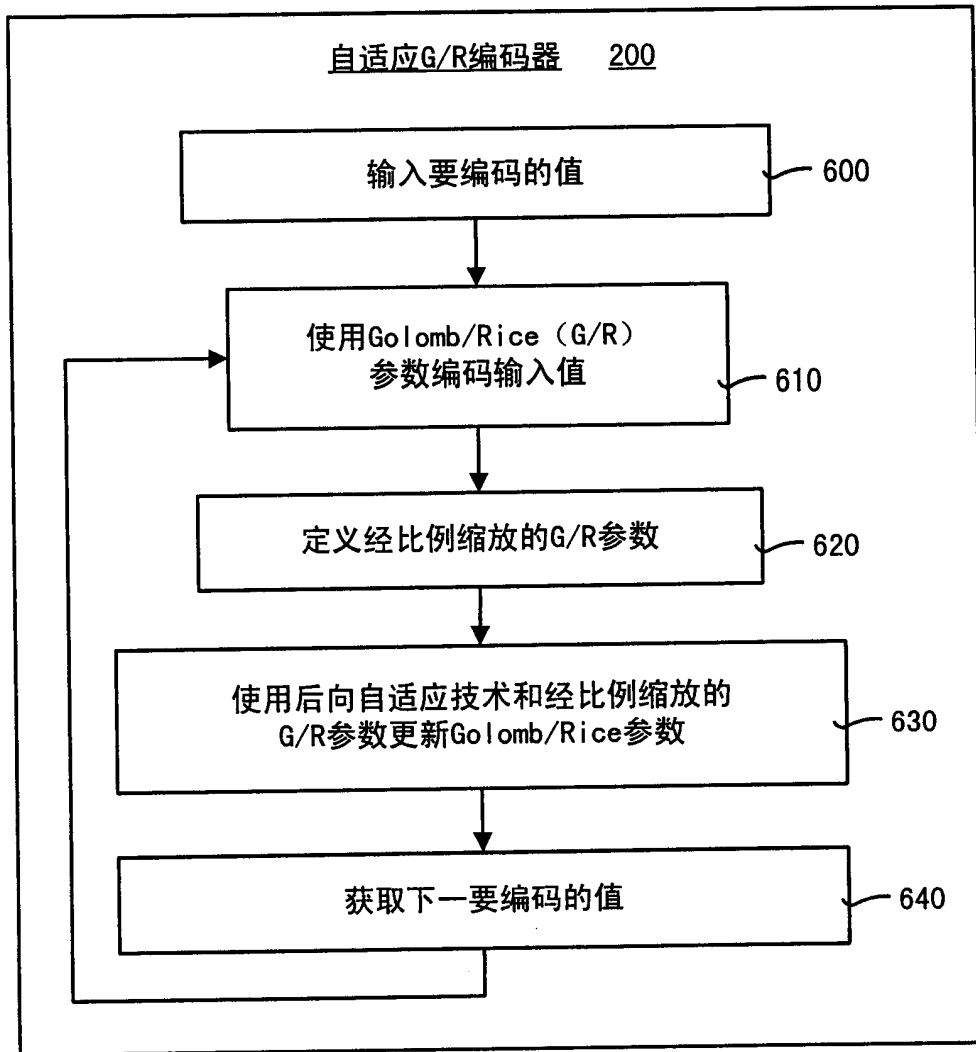


图 6

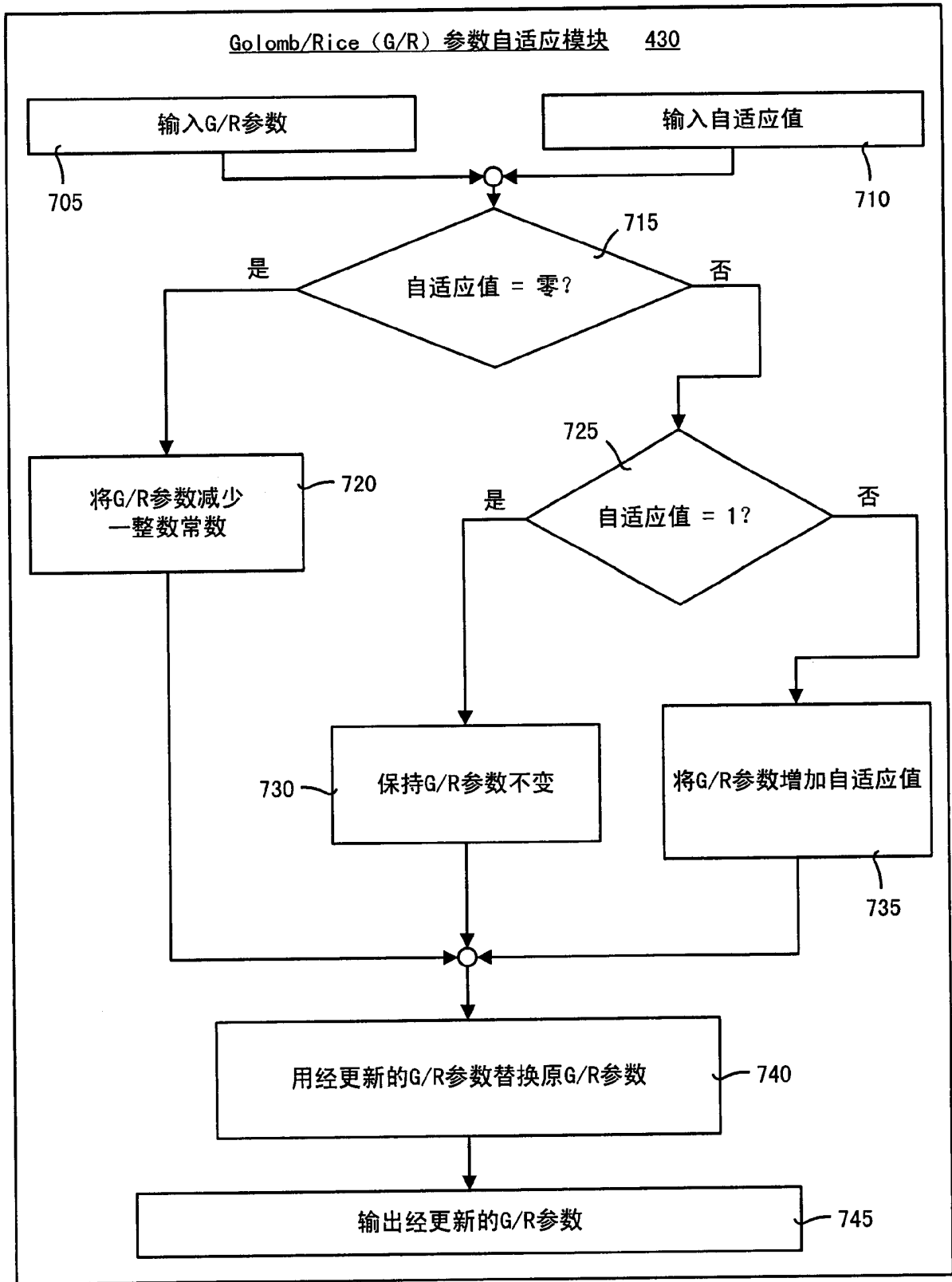


图 7

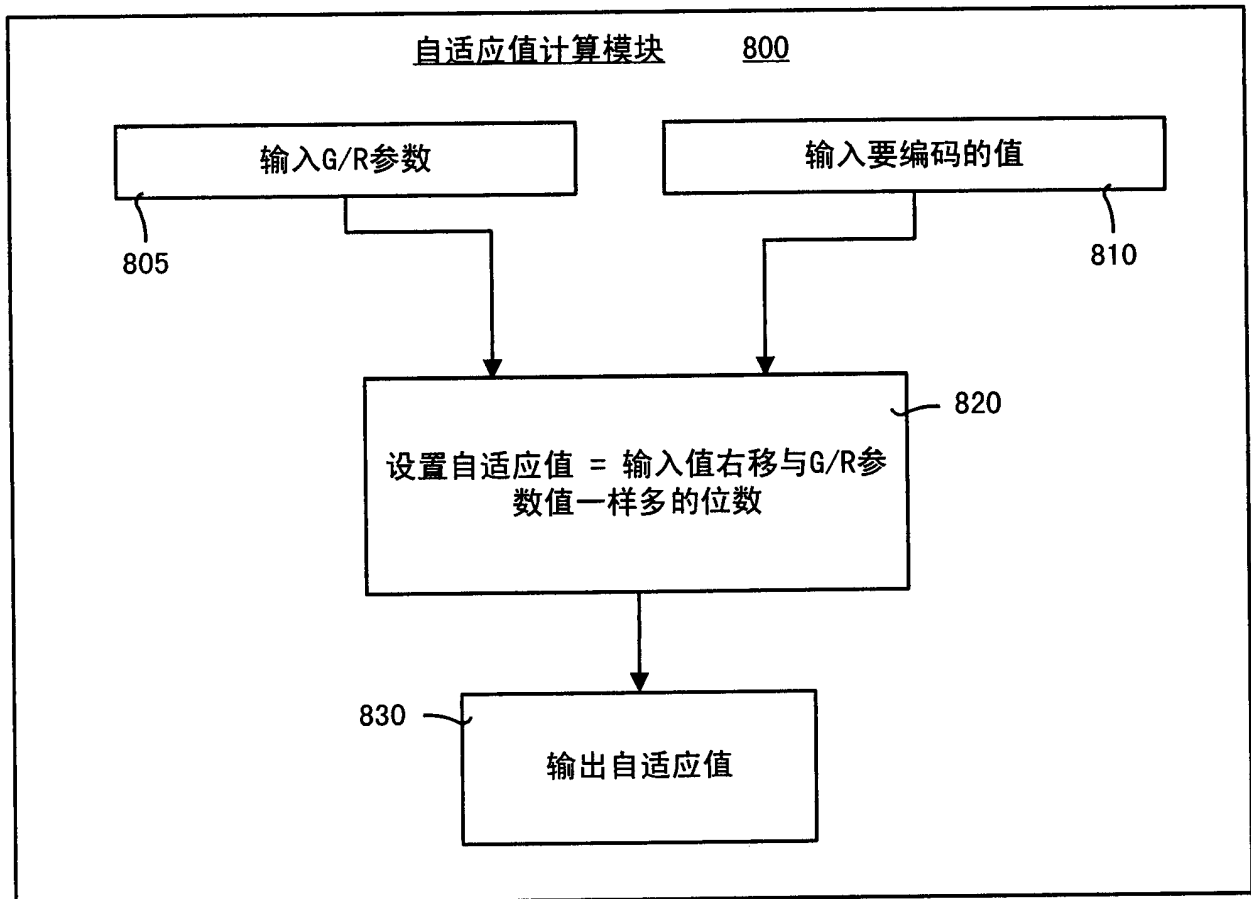


图 8

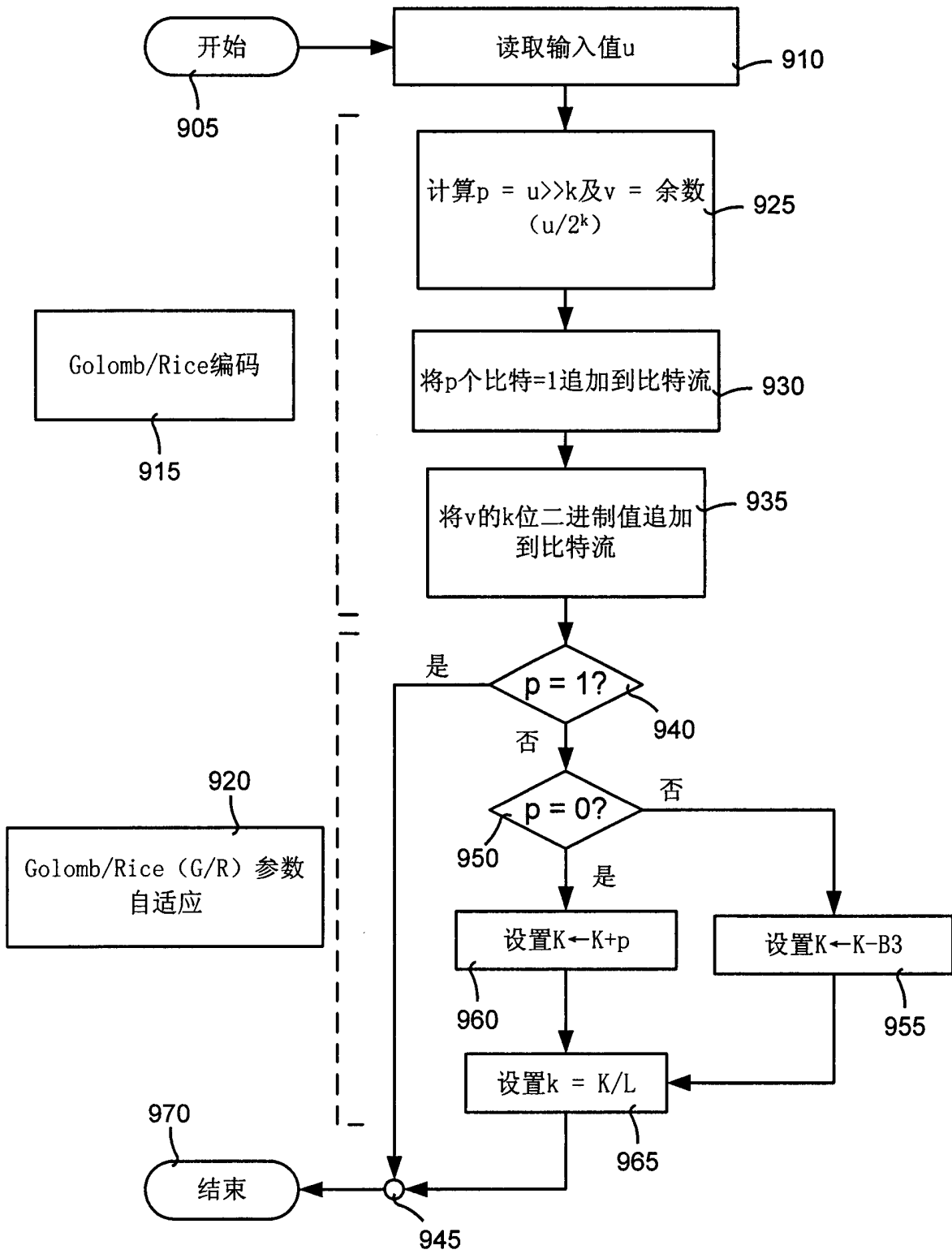


图 9