



(12)发明专利申请

(10)申请公布号 CN 106537914 A

(43)申请公布日 2017.03.22

(21)申请号 201580041108.8

(22)申请日 2015.06.29

(30)优先权数据

62/018,644 2014.06.29 US

(85)PCT国际申请进入国家阶段日

2017.01.24

(86)PCT国际申请的申请数据

PCT/KR2015/006621 2015.06.29

(87)PCT国际申请的公布数据

W02016/003130 KO 2016.01.07

(71)申请人 LG 电子株式会社

地址 韩国首尔

(72)发明人 阿米尔·赛义德

阿博·塔利波·马赫福兹

(74)专利代理机构 中原信达知识产权代理有限
责任公司 11219

代理人 谢丽娜 夏凯

(51)Int.Cl.

H04N 19/13(2006.01)

H04N 19/50(2006.01)

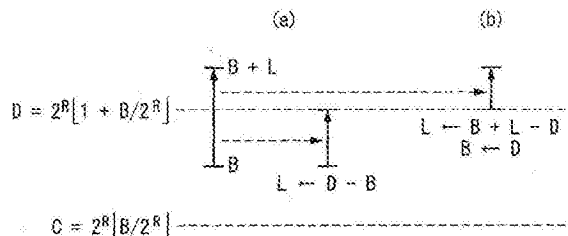
权利要求书2页 说明书11页 附图8页

(54)发明名称

通过限制的进位运算来执行算术编译的方法和设备

(57)摘要

所公开的是一种对数据符号执行算术编译的方法,该方法包括以下步骤:为各个数据符号生成区段,其中区段基于区段的起始点和长度来表示;针对各个数据符号来更新区段;检查所更新的区段是否落在预先确定的范围中;以及基于检查的结果重新归一化所更新的区段。



1. 一种针对数据符号执行算术编译的方法,所述方法包括:
为所述数据符号中的每一个创建区间,所述区间基于所述区间的起始点和长度来表示;
针对所述数据符号中的每一个更新所述区间;
检查所更新的区间是否被包括在特定范围中;以及
调整所更新的区间以减少未决比特。
2. 根据权利要求1所述的方法,还包括:
重新归一化所调整的区间。
3. 根据权利要求1所述的方法,还包括:
如果所更新的区间未被包括在特定范围中,则选择所更新的区间中的顶部部分和底部部分中的一个,
其中,基于选择的部分调整所更新的区间。
4. 根据权利要求3所述的方法,还包括:
在所更新的区间内计算阈值点;以及
将第一距离与第二距离进行比较,
其中,所述第一距离指示从所述阈值点到所述顶部部分的长度,并且所述第二距离指示从所述阈值点到所述底部部分的长度,并且
其中,基于所述比较步骤的结果调整所更新的区间。
5. 根据权利要求4所述的方法,
其中,如果所述第一距离小于所述第二距离,则所更新的区间的长度被设定为所述第二距离。
6. 根据权利要求4所述的方法,
其中,如果所述第一距离不小于所述第二距离,则所更新的区间的长度被设定为所述第一距离,并且由所述阈值点设定所更新的区间的起始点。
7. 一种针对数据符号执行算术编译的设备,所述设备包括二进制算术编译单元,所述设备包括:
所述二进制算术编译单元,所述二进制算术编译单元被配置成:为所述数据符号中的每一个创建区间,所述区间基于所述区间的起始点和长度来表示,
针对所述数据符号中的每一个来更新所述区间,
检查所更新的区间是否被包括在特定范围中,并且
基于所述检查的结果重新归一化所更新的区间。
8. 根据权利要求7所述的设备,
其中,如果所更新的区间被包括在特定范围中,则位于所述特定范围内的比特被存储在缓冲器中并且所更新的区间被重新归一化。
9. 根据权利要求7所述的设备,
其中,如果所更新的区间未被包括在特定范围中,则所述二进制算术编译单元还被配置成选择所更新的区间的顶部部分和底部部分中的一个,
其中,基于选择的部分调整所更新的区间。
10. 根据权利要求9所述的设备,

其中,所述二进制算术编译单元还被配置成对所述顶部部分的长度和所述底部部分的长度进行比较,

其中,基于所述比较的结果重新归一化所更新的区间。

11. 根据权利要求10所述的设备,

其中,如果所述顶部部分的长度小于所述底部部分的长度,则所更新的区间的长度被设定为所述底部部分的长度。

12. 根据权利要求10所述的设备,

其中,如果所述顶部部分的长度不小于所述底部部分的长度,则所更新的区间的长度被设定为所述顶部部分的长度,并且所更新的区间的起始点被设定为比特位置。

通过限制的进位运算来执行算术编译的方法和设备

技术领域

[0001] 本发明涉及用于处理视频信号的方法和设备,并且更具体地,涉及一种用于利用约束进位运算执行算术编译的技术。

背景技术

[0002] 熵编译是用最少平均数量的比特表示数据的过程,并且因此是关于所有媒体压缩方法的基本技术。算术编译是与类似前缀编译(例如,霍夫曼(Huffman)、莱斯(Rice)、exp-Golomb码)的其它编译技术相比具有更高的计算复杂性的熵编译方法,但是因为它产生更好的压缩(接近理论极限),所以它已经在最近的视频编译标准H.265/HEVC和VP9中作为唯一熵编译方法被采用,并且因此期望为将来标准的单一熵编译方法。

发明内容

[0003] 技术问题

[0004] 存在用于实践上实现算术编译的数个选项,并且其中每个可以取决于实施方式是用通用处理器和常规软件工具、还是专门硬件(ASIC)来完成的而具有优点和缺点。

[0005] 因此,本发明定义一种能够基于在通用或专门硬件中有效地实现的字节重新归一化来使得能够实现算术编译的技术。

[0006] 定制硬件的一个问题是,在数据符号被编码之后,对压缩的数据字节的改变的最大数量可能是无界的。因此,本发明提供用于以小的压缩损失有效地限制这个数量的方法。

[0007] 技术方案

[0008] 本发明的实施例定义了基于在通用或专门硬件中有效地实现的字节重新归一化来使得能够实现算术编译的技术。

[0009] 此外,本发明提供用于以小的压缩损失有效地限制对压缩的数据字节的改变的最大数量的方法。

[0010] 此外,本发明提供用于调整区间的方法。

[0011] 此外,本发明提供用于针对区间重新归一化执行延迟判定的方法。

[0012] 此外,本发明提供用于执行受缓冲器值约束的判定的方法。

[0013] 此外,本发明提供用于消除在扩展精度寄存器中溢出或者在具有未决比特(outstanding bits)的缓冲器等效地溢出的可能性。

[0014] 有益效果

[0015] 本发明能够通过基于字节重新归一化使得能够实现算术编译来用通用或专门硬件有效地实现。

[0016] 此外,本发明能够以小的压缩损失有效地限制对压缩的数据字节的改变的最大数量。

附图说明

[0017] 图1和图2图示根据本发明被应用于的实施例的处理视频信号的编码器和解码器的示意框图。

[0018] 图3是图示根据本发明被应用于的实施例的更新算术编译区间数据所需要的操作的集合的流程图。

[0019] 图4和图5图示根据本发明被应用于的实施例的基于二进制算术编译来处理视频信号的熵编码单元和熵解码单元的示意框图。

[0020] 图6是用于图示根据本发明被应用于的实施例的在有效比特的窗口处利用算术的编译过程的图。

[0021] 图7是用于图示根据本发明被应用于的实施例的针对算术编译区间的重新归一化的操作的图。

[0022] 图8是用于图示根据本发明被应用于的实施例的算术编译区间的可能的位置的图。

[0023] 图9是用于图示根据本发明被应用于的实施例的调整算术编译区间的过程的图。

[0024] 图10是图示根据本发明被应用于的实施例的利用约束进位操作执行算术编译的方法的流程图。

[0025] 图11是图示根据本发明被应用于的实施例的重新归一化算术编译区间的的方法的图。

[0026] 图12是图示根据本发明被应用于的实施例的调整算术编译区间的的方法的图。

[0027] 图13和图14是图示根据本发明被应用于的另一实施例的调整算术编译区间的的另一方法的流程图。

具体实施方式

[0028] 根据本发明的一个方面,提供了针对数据符号执行算术编译的方法,包括:为所述数据符号中的每一个创建区间,该区间基于区间的起始点和长度来表示;针对数据符号中的每一个来更新区间;检查所更新的区间是否被包括在特定范围中;以及基于检查的结果重新归一化所更新的区间。

[0029] 在本发明中,该方法还由重新归一化所调整的区间组成。

[0030] 在本发明中,如果所更新的区间未被包括在特定范围中,则该方法还由选择所更新的区间的顶部部分和底部部分中的一个组成,其中,所更新的区间基于选择的部分被调整。

[0031] 在本发明中,该方法还由以下步骤组成:在所更新的区间内计算阈值点;以及将第一距离与第二距离进行比较,其中,第一距离指示从阈值点到顶部部分的长度并且第二距离指示从阈值点到底部部分的长度,并且其中,所更新的区间基于比较步骤的结果被调整。

[0032] 在本发明中,如果第一距离小于第二距离,则所更新的区间的长度被设定为第二距离。

[0033] 在本发明中,如果第一距离不小于第二距离,则所更新的区间的长度被设定为第一距离,并且由阈值点设定所更新的区间的起始点。

[0034] 根据本发明的另一方面,提供了针对数据符号执行算术编译的设备,该设备包括二进制算术编译单元,包括:二进制算术编译单元,该二进制算术编译单元被配置成:为数

据符号中的每一个创建区间,该区间基于区间的起始点和长度来表示;针对数据符号中的每一个来更新区间;检查所更新的区间是否被包括在特定范围中;并且基于检查的结果重新归一化所更新的区间。

[0035] 发明的模式

[0036] 在下文中,参考附图描述根据本发明的实施例的示例性元素和操作。然而应当注意,参考附图所描述的本发明的元素和操作仅作为实施例被提供,并且本发明的技术精神以及核心配置和操作不限于此。

[0037] 此外,本说明书中使用的术语是现在广泛地使用的常见术语,但是在特殊情况下,使用由本申请人随机地选择的术语。在这种情况下,在对应部分的详细描述中清楚地描述对应术语的含义。因此,应当注意,本发明不应该被解释为基于仅本说明书的对应描述中使用的术语的名称并且本发明应该通过甚至检查对应术语的含义来解释。

[0038] 此外,本说明书中使用的术语是被选择来描述本发明的常见术语,但是在存在具有类似含义的这些术语的情况下,可以为了更适当的分析用其它术语替换。例如,可以在每个编译过程中适当地替换并解释信号、数据、样本、图片、帧和块。另外,可以在每个算术编译过程中适当地替换并解释范围、长度、区间(或编译区间、算术编译区间)以及区间长度。

[0039] 图1和图2图示根据本发明被应用于的实施例的处理视频信号的编码器和解码器的示意框图。

[0040] 图1的编码器100包括变换单元110、量化单元120以及熵编码单元130。图2的解码器200包括熵解码单元210、去量化单元220以及逆变换单元230。

[0041] 编码器100接收视频信号并且通过将该视频信号减去预测信号来生成预测误差。

[0042] 所生成的预测误差被发送到变换单元110。变换单元110通过对预测误差应用变换方案来生成变换系数。

[0043] 量化单元120对所生成的变换系数进行量化并且将所量化的系数发送到熵编码单元130。

[0044] 熵编码单元130对所量化的信号执行熵编译并且输出熵编译的信号。在这种情况下,熵编译是用来最优地定义进入压缩数据序列的比特的数量的过程。作为最优熵编译技术之一的算术编译是通过单个实数来表示多个符号的方法。

[0045] 在常规编译条件下,压缩的数据缓冲器中的比特的值是几乎同样概率的,并且因此未决比特的非常长的序列是非常不可能的,因为n个未决比特的序列的概率是 $1/2^n$ 。然而,必须将算术编译的实际实施方式设计成与任意输入数据和符号概率集合一起工作,包括可能偶然地(或故意地)产生未决比特的无界数的那些。

[0046] 为软件(通用处理器)实施方式而设计的编码器采用了未决比特的计数器,或者简单地假定了压缩的数据缓冲器上的进位。然而,另一方面,使用专门处理器(ASIC)实现的编码器通常对每个操作所需要的周期的数量需要强限制,并且因此不能够适应平均情况与最坏情况之间的非常大的变化。

[0047] 针对这个问题的一个解决方案(被称作“比特填充”)是为一次生产一个比特的算术编译器而开发的:当连续的“1”比特的数量超过特定限制时,可以人为地插入“0”比特以限制进位传播。在压缩方面存在一些损失,因为那些额外的比特不携带任何有用的信息,但是能够使它保持相对小。

[0048] 这个解决方案的问题是它在生成全字节(8比特字,而不是单个比特)的集合的算术编译的新实施方式中是无效的。新实施方式意指当在难以提高时钟速度的同时数据总线和并行处理便宜时,由算术编码器生成除了单个比特单位信息以外的字节单位信息的方案。

[0049] 对于那些实施方式能够将“比特填充”方法扩展到“字节填充”,即,人为地添加零字节以限制进位传播。虽然这实现所期望的结果,但是这是不实际的,因为必须添加不传达任何信息的8个比特,并且相对压缩损失变得过度。

[0050] 因此,本发明定义基于要用通用或专门硬件有效地实现的字节重新归一化使得能够实现算术编译的技术。

[0051] 此外,本发明提供用于以小的压缩损失有效地限制对压缩的数据字节的改变的最大数量的方法。

[0052] 在本发明的一个方面中,熵编码单元130可以为数据符号中的每一个创建一个区间,并且针对数据符号中的每一个来更新该区间。在这种情况下,可以基于区间的起始点和长度来表示区间。

[0053] 熵编码单元130可以检查所更新的区间是否被包括在特定范围中,并且基于检查的结果重新归一化所更新的区间。

[0054] 如果所更新的区间被包括在特定范围中,则熵编码单元130可以在缓冲器中存储该位于特定范围内的比特,并且重新归一化所更新的区间。

[0055] 图2的解码器200接收由图1的编码器100所输出的信号。

[0056] 熵解码单元210对所接收到的信号执行熵解码。例如,熵解码单元210可以接收包括码值的位置信息的信号,检查与码值的位置信息相对应的符号,并且对经检查的符号进行解码。

[0057] 在算术编译中,可以通过压缩的序列的码值来确定所解码的序列。在解码过程中,码值能够被用于对正确的序列进行解码。

[0058] 熵解码单元210可以从第k个归一化码值中搜索第k个正确值,然后可以根据第k个正确值和第k个归一化码值来计算第(k+1)个归一化码值。

[0059] 此外,去量化单元220基于关于量化步长的信息来从经熵解码的信号获得变换系数。

[0060] 逆变换单元230通过对变换系数执行逆变换来获得预测误差。重建信号是通过将所获得的预测误差加到预测信号而生成的。

[0061] 图3是图示更新算术编译区间数据所需要的操作的集合的流程图。

[0062] 本发明被应用于的算术编译器能够包括数据源单元(310)、数据建模单元(320)、第一延迟单元(330)和第二延迟单元。

[0063] 数据源单元(310)能够如以下等式1那样每个从M个符号的字母表生成N个随机符号的序列。

[0064] [等式1]

[0065] $S = \{s_1, s_2, s_3, \dots, s_N\}, s_k \in \{0, 1, 2, \dots, M-1\}$

[0066] 在这种情况下,本发明假定数据符号是完全独立且同分布的(i.i.d.),如以下等式2具有非零概率。

[0067] [等式2]

[0068] $\text{Prob}\{s_k=n\} = p(n) > 0, k=1, 2, \dots, N, n=0, 1, \dots, M-1$

[0069] 另外,本发明能够如以下等式3定义累计概率分布。

[0070] [等式3]

$$[0071] \quad c(n) = \sum_{s=0}^{n-1} p(s), \quad n = 0, 1, \dots, M$$

[0072] 在这种情况下, $c(s)$ 是严格单调的, 并且 $c(0) = 0$ 以及 $c(M) = 1$ 。

[0073] 即使那些条件可以似乎与在实际的复杂媒体信号中发现的条件大不相同, 实际上所有熵编译工具基于从那些假定得到的技术, 所以本发明也能够提供受这个更简单模型约束的实施例。

[0074] 算术编译主要由以形式 $[b_k, b_{k+1})$ 更新实数线中的半开区间, 其中 b_k 表示区间基并且 l_k 表示其长度。可以根据每个数据符号 s_k 并且从初始条件 $b_1 = 0$ 和 $l_1 = 1$ 开始更新区间, 它们使用以下等式4和等式5来针对 $k=1, 2, \dots, N$ 递归地更新。

[0075] [等式4]

$$[0076] \quad l_{k+1} = p(s_k) l_k$$

[0077] [等式5]

$$[0078] \quad b_{k+1} = b_k + c(s_k) l_k$$

[0079] 在这种情况下, 区间可以如以下等式6渐进地嵌套。

[0080] [等式6]

$$[0081] \quad [b_k, b_{k+1}) \subset [b_i, b_{i+1}), k=1, 2, \dots, i-1, i=2, 3, \dots, N+1$$

[0082] 如上所述, 参考图3, 数据建模单元 (320) 能够接收 N 个随机符号的序列 S_k , 并且输出累计概率分布 $C(S_k)$ 和符号概率 $p(S_k)$ 。

[0083] 能够通过从数据建模单元 (320) 输出的 S_k 以及从第一延迟单元 (330) 输出的 l_k 的乘法运算来获得区间长度 l_{k+1} 。

[0084] 另外, 能够通过从第二延迟单元 (340) 输出的 b_k 的加法运算以及 $C(S_k)$ 和 l_k 的乘法来获得区间基 b_{k+1} 。

[0085] 能够通过乘法和加法的算术运算来定义本发明被应用于的算术编译。在这种情况下, 能够以无限精度表示 b_k 和 l_k , 但是完成这个以首先在直观简单的版本中引入计数法。稍后本发明提供用于使用有限精度运算来近似地实现算术编译的方法。

[0086] 在已经计算最终区间 $[b_{N+1}, b_{N+1} + l_{N+1})$ 之后, 由码值 $\hat{v} \in [b_{N+1}, b_{N+1} + l_{N+1})$ 来定义算术编码的消息。可以证明存在一个能够使用至多 $1 + \log_2(l_{N+1})$ 个比特来表示的这样的值。

[0087] 为了使用码值 \hat{v} 来对序列 S 进行解码, 本发明再次从初始条件 $b_1 = 0$ 和 $l_1 = 1$ 开始, 然后使用以下等式7至等式9来渐进地获得 s_k, l_k 和 b_k 。

[0088] [等式7]

$$[0089] \quad s_k = \left\{ s : c(s) \leq \frac{\hat{v} - b_k}{l_k} < c(s+1) \right\}$$

[0090] [式8]

[0091] $l_{k+1} = p(s_k) l_k$

[0092] [等式9]

[0093] $b_{k+1} = b_k + c(s_k) l_k$

[0094] 能够从所有区间被嵌套、 $\forall \in [b_{N+1}, b_{N+1} + l_{N+1})$ 、并且假定解码器完美地再现由编码器所做的操作的特性推断出这个解码过程的正确性。

[0095] 本发明能够分别使用符号 B_k 、 L_k 和 D_k 来表示 b_k 、 l_k 和 $\forall - b_k$ 的有限精度值(通常被换算为整型(integer)值)。编码的方面能够通过以下等式10和等式11来定义。

[0096] [等式10]

[0097] $L_{k+1} = \lceil [c(s_{k+1}) L_k] \rceil - \lceil [c(s_k) L_k] \rceil$

[0098] [等式11]

[0099] $B_{k+1} = B_k + \lceil [c(s_k) L_k] \rceil$

[0100] 在这种情况下,围绕乘积的双方括号表示乘法是有限精度近似。

[0101] 等式10对应于等式4,因为 $p(s) = c(s+1) - c(s)$ ($s = 1, 2, \dots, M$)。

[0102] 因此,解码过程能够通过以下等式12至等式14来定义。

[0103] [等式12]

[0104] $S_k = \{s : \lceil [c(s) L_k] \rceil \leq D_k < \lceil [c(s+1) L_k] \rceil\}$

[0105] [等式13]

[0106] $L_{k+1} = \lceil [c(s_{k+1}) L_k] \rceil - \lceil [c(s_k) L_k] \rceil$

[0107] [等式14]

[0108] $B_{k+1} = B_k + \lceil [c(s_k) L_k] \rceil$

[0109] 图4和图5图示根据本发明被应用于的实施例的基于二进制算术编译来处理视频信号的熵编码单元和熵解码单元的示意框图。

[0110] 参考图4,熵编码单元(130)包括二值化单元(410)、上下文建模单元(420)以及二进制算术编码单元(430)。

[0111] 二值化单元(410)可以接收数据符号的序列并且通过执行二值化来输出由二值化值0或1构成的二进制串(bin string)。二值化单元(410)可以将语法元素映射到二进制符号(bin)。数个不同的二值化过程(例如,一元(U)、截断一元(TU)、k阶Exp-Golomb(EGK)和固定长度(FL))可以被用于二值化。可以基于语法元素的类型来选择二值化过程。

[0112] 所输出的二进制串被发送到上下文建模单元(420)。上下文建模单元(420)针对熵编码执行概率估计。也就是说,上下文建模单元(420)可以估计二进制(bins)的概率。

[0113] 上下文建模单元(420)可以提供实现高编译效率所需要的准确的概率估计。因此,高度自适应且不同的上下文模型能够被用于不同的二进制并且可以基于先前编译的二进制的值来更新该上下文模型的概率。

[0114] 具有类似分布的二进制可以共享同一上下文模型。能够基于语法元素的类型、二进制在语法元素中的位置(binIdx)、亮度/色度、邻近信息等中的至少一个来选择用于每个二进制的上下文模型。

[0115] 二进制算术编码单元(430)对所输出的二进制串进行熵编码并且输出压缩数据比特。

[0116] 二进制算术编码单元(430)基于递归区间划分来执行算术编译。

[0117] 具有0至1的初始值的区间(或范围)基于二进制的概率被划分成两个子区间。编码的比特当被转换为二进制小数时提供选择两个子区间中的指示所解码的二进制的值的一个的偏移。

[0118] 在每个解码的二进制之后,可以更新区间以等于所选择的子区间,并且区间划分过程重复其本身。区间和偏移值具有有限比特精度,所以每当区间落在特定值以下时可能需要重新归一化以防止下溢。可能在每个二进制被解码之后发生重新归一化。

[0119] 能够使用估计概率或者假定0.5的相等概率来完成算术编译。对于绕开编译的二进制,能够通过移位来完成范围变成子区间的划分,然而对于上下文编译的二进制可能需要查找表。

[0120] 参考图5,熵解码单元(210)包括二值制算术解码单元(510)、上下文建模单元(520)以及去二值化单元(530)。

[0121] 熵解码单元(210)能够逆执行在图4的描述中所说明的以上编码过程。

[0122] 在二进制算术解码单元(510)处,可以反馈所更新的区间以用于递归区间划分。可以将所更新的上下文从二进制算术解码单元(510)反馈给上下文建模单元(520)以用于准确的概率估计。

[0123] 上下文建模单元(520)可以基于语法元素的类型来选择上下文。在熵解码单元(210)处,可以反馈所解码的二进制以确定是否继续处理同一语法元素还是切换到另一语法元素。上下文建模单元(520)也可以基于二进制在语法元素中的位置(binIdx)来选择上下文。

[0124] 去二值化单元(530)可以将二进制符号串转换为多值符号。

[0125] 图6是用于图示根据本发明被应用于的实施例的在有效比特的窗口处利用算术的编译过程的图。

[0126] 对于算术编译的实际实施方式,本发明可能认为所有加法是按无限精度来完成的,但是乘法被以保存一些特性的方式使用有限精度来近似。本说明书将仅涵盖理解本发明所需的方面。

[0127] 算术编译的一个重要方面是它可能被认为是针对加法按无限精度完成算术但是用保证所有运算隐式地在无限精度寄存器中完成的近似来替换确切的乘法的过程。

[0128] 基于这个解释,如图6中所示,编译过程能够在有效比特的“窗口”处利用算术来完成。在图6中,符号‘a’表示任意比特值,并且B表示将换算的 b_k 的十进制值乘以 2^P 并且使它成为整型值的值。

[0129] 因为区间长度 l_k 减小,所以有必要周期性地增加P,并且在被称作重新归一化的过程中重新换算 $L=2^P L$ 的值。在重新归一化期间,可以将独立比特或完整字节从寄存器复制制到包含压缩的数据的缓冲器。

[0130] 在图6中,被称作“已决”和“未决”的比特表示已经被移动到具有压缩数据的缓冲器的比特,并且因此可以未被有效地用在算术运算中。然而,能够在图6中观察到的本发明的一个重要方面是可能存在可能由于来自进一步加法的二进制算术进位而可能改变的任何数量的未决比特。

[0131] 图7是用于图示根据本发明被应用于的实施例的针对算术编译区间的重新归一化

的操作的图。

[0132] 对于算术编译的实际实施方式,因为在有限精度寄存器中完成操作,所以当要输出的区间的上部比特被确定时,可以输出该比特并且可以执行重新归一化以便扩展区间的宽度。

[0133] 如图7中所示,粗线表示所选择的区间。在图7的第二条线中,因为要输出的区间 $[0.64, 0.8)$ 的比特值明显大于0.5,所以可以在这个时间点输出编译的比特‘1’,并且区段 $[0.5, 1.0)$ 可以与扩展区段 $[0.0, 1.0)$ 相匹配。通过这些步骤,可以使所选择的区间重新归一化。

[0134] 图8是用于图示根据本发明被应用于的实施例的算术编译区间的可能的位置的图。

[0135] 在以下说明中,本发明假定可以利用P位寄存器(例如, $P=32$ 或 $P=64$)完成算术编译操作,但是为了简化记数法,本发明能够定义 P_c 位的“进位精度”,其可以包括为进位传播而预留的额外的存储器字节。例如,使用具有 $P=32$ 位的寄存器但是具有为进位传播而预留的单独字节的处理器的实施方式为了方便被解释为使用具有 $P_c=40$ 位的寄存器的实施方式。此外,本发明假定区间基 B_k 和长度 L_k 中的索引k被省略并且B和L表示最新索引值。

[0136] 本发明提出了调整区间的方法。在这种情况下,编码器和解码器能够保持跟踪B和L,并且两者均能够识别当期期望限制进位传播时的情形。

[0137] 本发明能够执行正确的解码,只要编码器和解码器使用相同规则来改变B和L并且该规则不违反正确的算术解码的条件即可。

[0138] 在本发明的一个方面中,有限精度算术编译需要在其区间长度L小于特定值时重新归一化其区间长度L。

[0139] 图8示出当满足特定条件时的可能的情形的图。例如,当满足条件 $L < 2^R$ (例如, $R=24$ 能够被用于32位算术)时,可以像图8中所示出的表示区间的可能的情形。在这种情况下,本发明能够如以下等式15定义比特位置。

[0140] [等式15]

$$[0141] \quad C = 2^R \left\lfloor \frac{B}{2^R} \right\rfloor, \quad D = 2^R \left\lfloor 1 + \frac{B}{2^R} \right\rfloor$$

[0142] [等式16]

$$[0143] \quad [C, D) \supset [B, B+L)$$

[0144] 另外,如果如等式16嵌套区间,则在位置R及之上的比特将不因进位而改变,并且因此能够被立即保存到压缩的数据缓冲器,并且能够重新归一化区间。

[0145] 图9是用于图示根据本发明被应用于的实施例的调整算术编译区间的过程的图。

[0146] 如果区间 $[C, D)$ 不包含区间 $[B, B+L)$,则编码器不能够知道是否将发生进位。在这种情况下,本发明提出两个实施例。

[0147] 在第一实施例中,本发明不假定未决比特的任何暂时存储,即, $P_c=P$ 。在第二实施例中,本发明认为存储少量的未决字节以允许一些进位,并且 $P_c > P$ 。

[0148] 在本发明的一个方面中,要由编码器和解码器使用的以下算法完整地描述针对区间重新归一化的判定。在图9中,可以根据对顶部区间的选择隐式地确定是否发生了进位。

[0149] 熵编码单元可以如以下等式17计算比特位置。

[0150] [等式17]

$$[0151] \quad T = 2^R \lfloor (B + L - 1) / 2^R \rfloor$$

[0152] 熵编码单元可以检查T是否大于B。如果T大于B,则熵编码单元可以检查L是否大于 2^S 。如果L大于 2^S ,则返回。另外,熵编码单元可以检查顶部部分的长度(B+L-T)是否小于底部部分的长度(T-B)或者T是否等于0。

[0153] 如果顶部部分的长度(B+L-T)小于底部部分的长度(T-B)或者T等于0,则熵编码单元可以将L设定为(T-B)。否则,熵编码单元可以将L设定为(B+L-T)并且将B设定为T。

[0154] 然后,熵编码单元可以使区间归一化。

[0155] 在这种情况下,本发明假定 $L < 2R$ 并且 $S \leq R$,以及条件 $T=0$ 仅在存在无符号整型算术溢出时才发生。这与用来在一些实施方式上开始进位传播的条件有关。

[0156] 在第二实施例中,熵编码单元可以执行受缓冲器值约束的判定。本发明可以将最新数据字节存储在短缓冲器中,仅在那些字节中具有可能的进位。如上面所说明的,这可能等效于具有带附加比特的寄存器,并且本发明能够使用以下算法。

[0157] 熵编码单元可以如以上等式17计算比特位置。在这种情况下,本发明可以假定P_c个比特的精度。

[0158] 另外,熵编码单元可以检查T是否等于0。

[0159] 如果T等于0,则熵编码单元可以将L设定为(T-B)。然后,熵编码单元可以使区间归一化。

[0160] 如所说明的,本发明可以调整区间以消除在扩展精度寄存器中溢出或者在具有未决比特的缓冲器中等效地溢出的可能性。

[0161] 图10是图示根据本发明被应用于的实施例的利用约束进位操作执行算术编译的方法的流程图。

[0162] 二进制算术编码单元(440)可以执行递归区间划分。也就是说,能够基于二进制的概率将具有初始值0至1的区间划分成两个子区间。

[0163] 二进制算术编码单元(440)可以选择两个子区间中的指示编译的二进制的值的一个。能够更新区间以等于所选择的子区间。在这种情况下,编译区间的范围和基数通过有限数量的比特来表示,有必要重新归一化区间以防止精度降级。另外,在重新归一化期间可以作为编译的比特输出基数的上部比特。可以在范围小于阈值时重新归一化编译区间(基数、基数+范围)。

[0164] 对于编译的比特的进位传播和输出的处理,不输出编译的比特直到确认了进一步进位传播将不影响比特值为止。

[0165] 在本发明的一个方面中,二进制算术编码单元(440)可以为数据符号中的每一个创建一个区间(S1010)。可以基于二进制的概率来更新为数据符号中的每一个所创建的区间(S1020)。

[0166] 二进制算术编码单元(440)可以检查区间是否被嵌套(S1030)。例如,二进制算术编码单元(440)可以检查所更新的区间是否被包括在特定范围中。也就是说,参考图8,图8(a)示出了区间[B, B+L)被包括在特定范围[C, D)中,并且图8(b)示出了区间[B, B+L)未被包括在特定范围[C, D)中。因此,如图8(a)中所示,如果区间[B, B+L)被包括在特定范围[C, D)

中,则在位置R及之上的比特可能不因进位而改变。因此,可以将位置R及之上的比特保存到缓冲器。

[0167] 然后,二进制算术编码单元(440)可以基于检查的结果重新归一化所更新的区间(S1040)。

[0168] 图11是图示根据本发明被应用于的实施例的重新归一化算术编译区间的方法的图。

[0169] 在本发明的一个方面中,二进制算术编码单元(440)可以执行区间调整。

[0170] 首先,二进制算术编码单元(440)可以定义特定范围(S1110)。参考图9,可以如等式15定义特定范围[C,D)。

[0171] 二进制算术编码单元(440)可以检查区间是否被包括在特定范围中(S1120)。例如,二进制算术编码单元(440)能够检查区间是否对应于图8(a)或图8(b)。

[0172] 如果区间被包括在特定范围中,则二进制算术编码单元(440)能够将位于特定范围内的比特存储到缓冲器(S1130)。否则,二进制算术编码单元(440)可以在特定条件下选择区间的顶部部分或底部部分中的一个(S1140)。

[0173] 另外,二进制算术编码单元(440)可以重新归一化区间(S1150)。

[0174] 图12是图示根据本发明被应用于的实施例的调整算术编译区间的方法的图。

[0175] 在本发明的一个方面中,二进制算术编码单元(440)可以执行针对区间重新归一化的判定。

[0176] 熵编码单元可以如等式17计算比特位置(S1210)。

[0177] 熵编码单元(440)可以检查比特位置的值是否大于区间基的值(S1220)。例如,参考图9,熵编码单元(440)可以检查T是否大于B。

[0178] 如果比特位置的值大于区间基的值,则熵编码单元(440)也可以检查区间的顶部部分的长度是否小于底部部分的长度(S1230)。例如,参考图9,熵编码单元(440)可以检查顶部部分的长度(B+L-T)是否小于底部部分的长度(T-B)。或者,熵编码单元(440)可以检查。

[0179] 如果顶部部分的长度小于底部部分的长度,则熵编码单元(440)可以将区间的长度设定为底部部分的长度(S1240)。

[0180] 否则,熵编码单元(440)可以将区间的长度设定为顶部部分的长度,并且将区间的起始点设定为比特位置(S1250)。

[0181] 然后,熵编码单元(440)可以重新归一化区间(S1260)。

[0182] 图13和图14是图示根据本发明被应用于的另一实施例的调整算术编译区间的另一方法的流程图。

[0183] 在本发明的另一方面中,熵编码单元(440)可以执行受缓冲器值约束的判定。本发明可以将最新数据字节存储在短缓冲器中,仅在那些字节中具有可能的进位。如上面所说明的,这可能等效于具有带附加比特的寄存器,并且本发明能够使用以下算法。

[0184] 二进制算术编码单元(440)可以定义特定范围(S1310)。参考图9,可以如等式15定义特定范围[C,D)。

[0185] 二进制算术编码单元(440)可以检查区间是否被包括在特定范围中(S1320)。例如,二进制算术编码单元(440)能够检查区间是否对应于图8(a)或图8(b)。

[0186] 如果区间被包括在特定范围中,则二进制算术编码单元(440)能够将位于特定范围内的比特存储到缓冲器(S1330)。否则,二进制算术编码单元(440)可以调整区间以消除在扩展精度寄存器中溢出或者在具有未决比特的缓冲器等效地溢出的可能性(S1340)。另外,二进制算术编码单元(440)可以重新归一化区间(S1350)。

[0187] 在本发明的另一实施例中,熵编码单元(440)可以如以上等式17计算比特位置(S1410)。

[0188] 熵编码单元(440)可以检查比特位置是否等于0(S1420)。

[0189] 如果比特位置等于0,则熵编码单元(440)可以将区间长度设定为将比特位置减去区间基的值(S1430)。

[0190] 然后,熵编码单元(440)可以重新归一化区间(S1440)。

[0191] 如上所述,可以在处理器、微处理器、控制器或芯片上实现和执行本发明中所说明的实施例。例如,可以在计算机、处理器、微处理器、控制器或芯片上实现和执行图1至图5中所说明的功能单元。

[0192] 此外,本发明被应用于的解码器和编码器可以被包括在多媒体广播传输/接收设备、移动通信终端、家庭影院视频设备、数码影院视频设备、监视相机、视频聊天设备、实时通信设备(诸如视频通信)、移动流设备、存储介质、录像机、VoD服务提供设备、互联网流服务提供设备、三维(3D)视频设备、电话会议视频设备以及医疗视频设备中,并且可以被用来处理视频信号和数据信号。

[0193] 此外,可以以程序的形式产生本发明被应用于的处理方法,所述程序将由计算机执行并且可以被存储在计算机可读记录介质中。具有根据本发明的数据结构的多媒体数据也可以被存储在计算机可读记录介质中。计算机可读记录介质包括可由计算机系统读取的数据被存储在其中的所有类型的存储装置。例如,计算机可读记录介质可以包括BD、USB、ROM、RAM、CD-ROM、磁带、软盘以及光学数据存储装置。此外,计算机可读记录介质包括以载波(例如,通过互联网的传输)的形式实现的介质。此外,通过编码方法所生成的比特流可以被存储在计算机可读记录介质中或者可以通过有线/无线通信网络来发送。

[0194] 工业适用性

[0195] 已经出于说明性目的公开了本发明的示例性实施例,并且本领域的技术人员可以在所附权利要求中所公开的本发明的技术精神和范围内改进、改变、替换或者添加各种其它实施例。

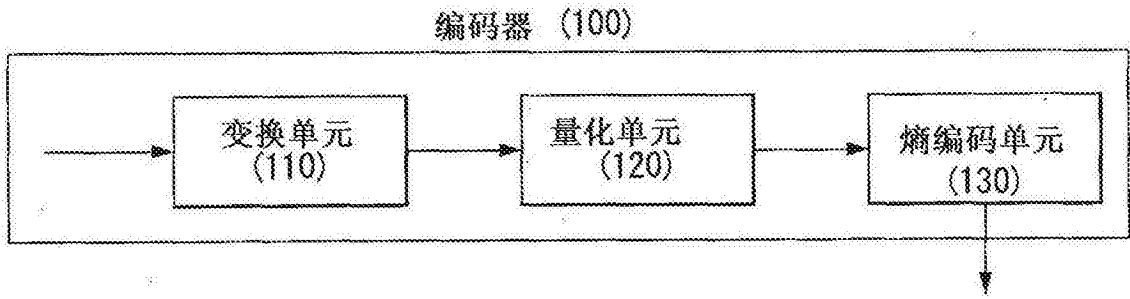


图1

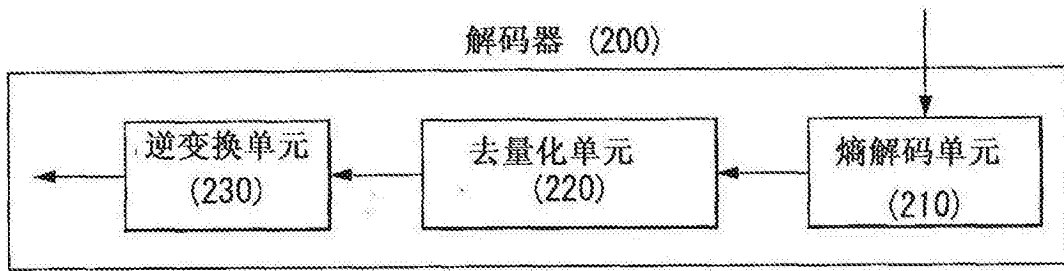


图2

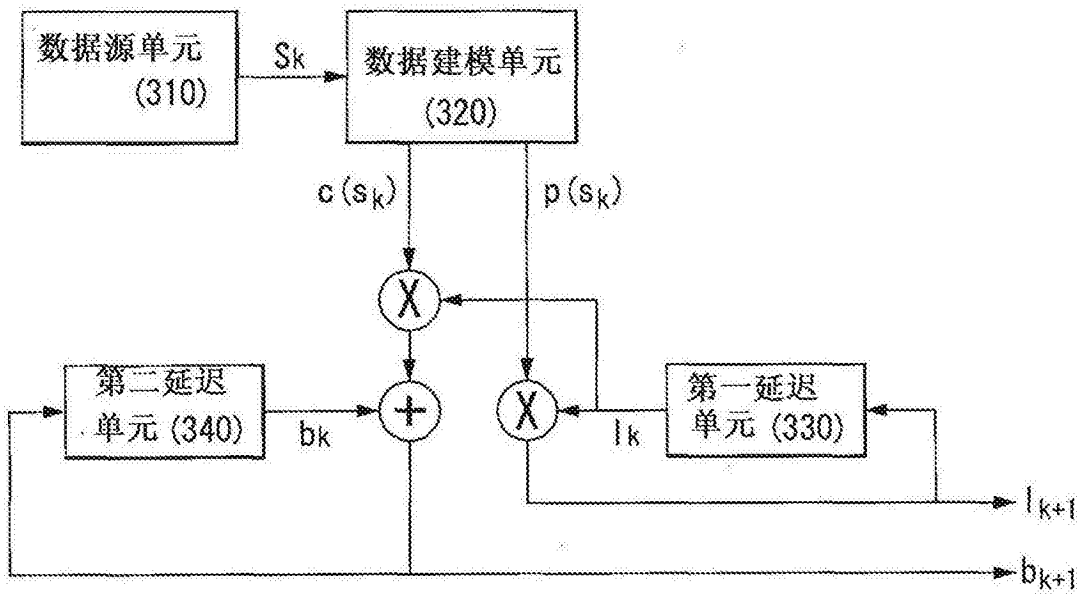


图3

熵编码单元 (130)

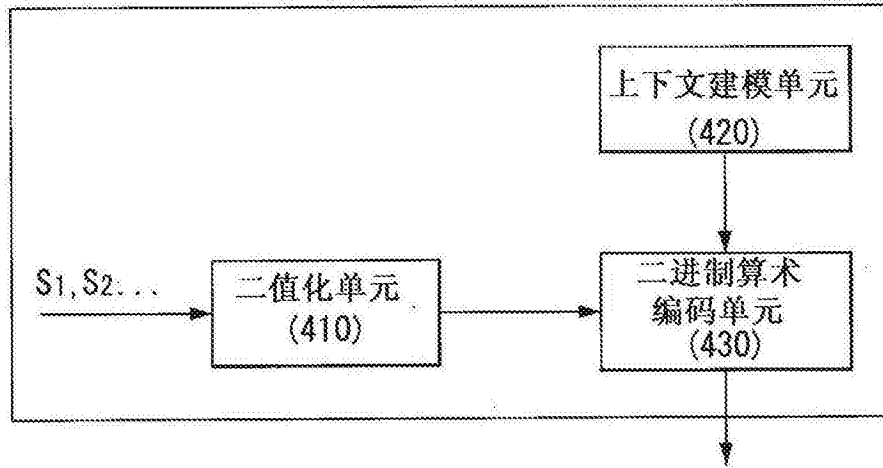


图4

熵解码单元 (210)

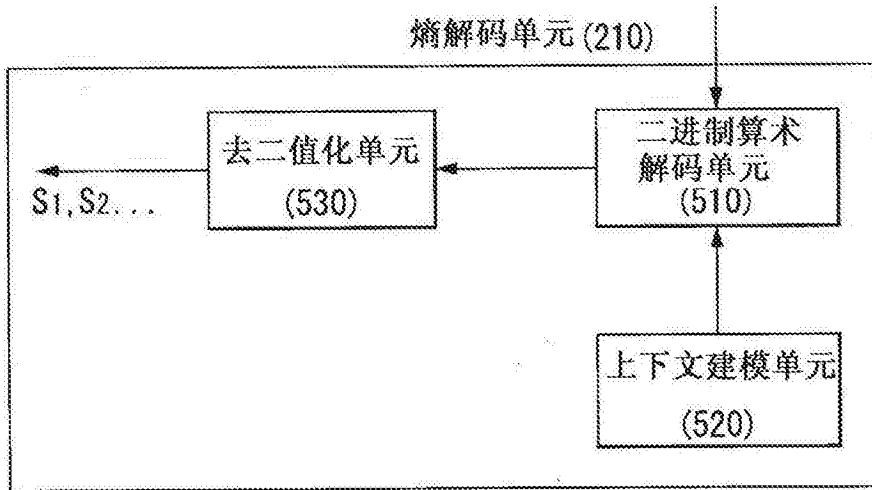


图5

$$\begin{array}{ccccccc}
 & & & & \overbrace{1aaa \dots aa}^{(L=2^P l)} & & 000000 \dots_2 \\
 l_k = & 0.0000 \dots 00 & 0000 \dots 00 & & & & \\
 \\
 b_k = & \underbrace{0.aaaa \dots aa}_{\text{已决的}} & \underbrace{0111 \dots 11}_{\text{未决的}} & & \underbrace{aaaa \dots aa}_{\substack{(B=2^P b) \\ \text{有效的}}} & & \underbrace{000000 \dots_2}_{\text{尾零}}
 \end{array}$$

图6

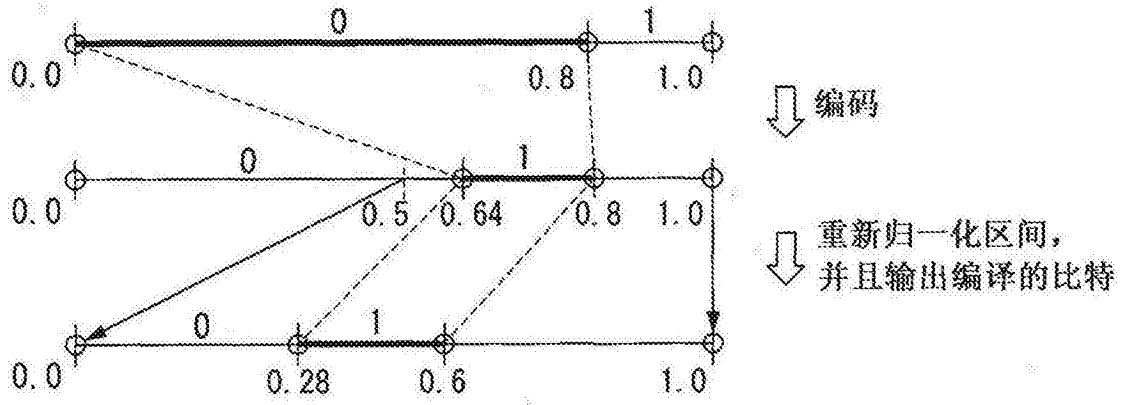


图7

(a)

(b)

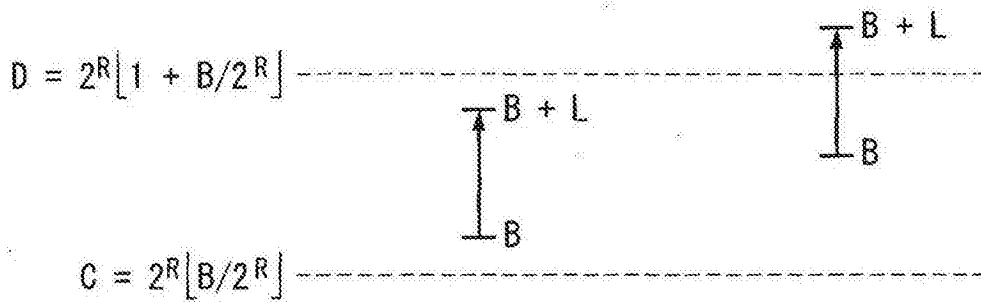


图8

(a)

(b)

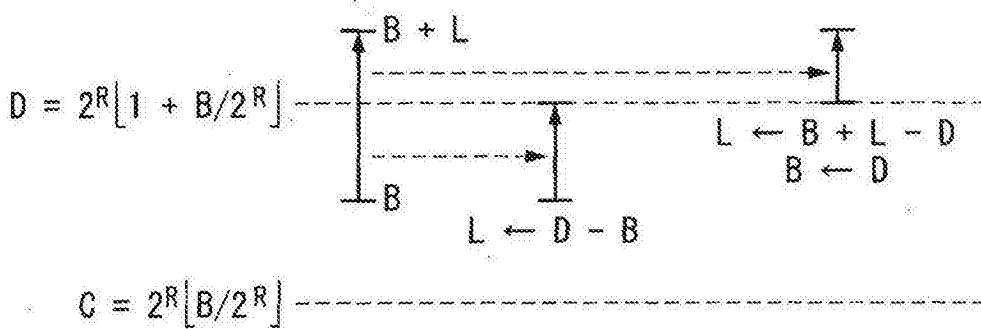


图9

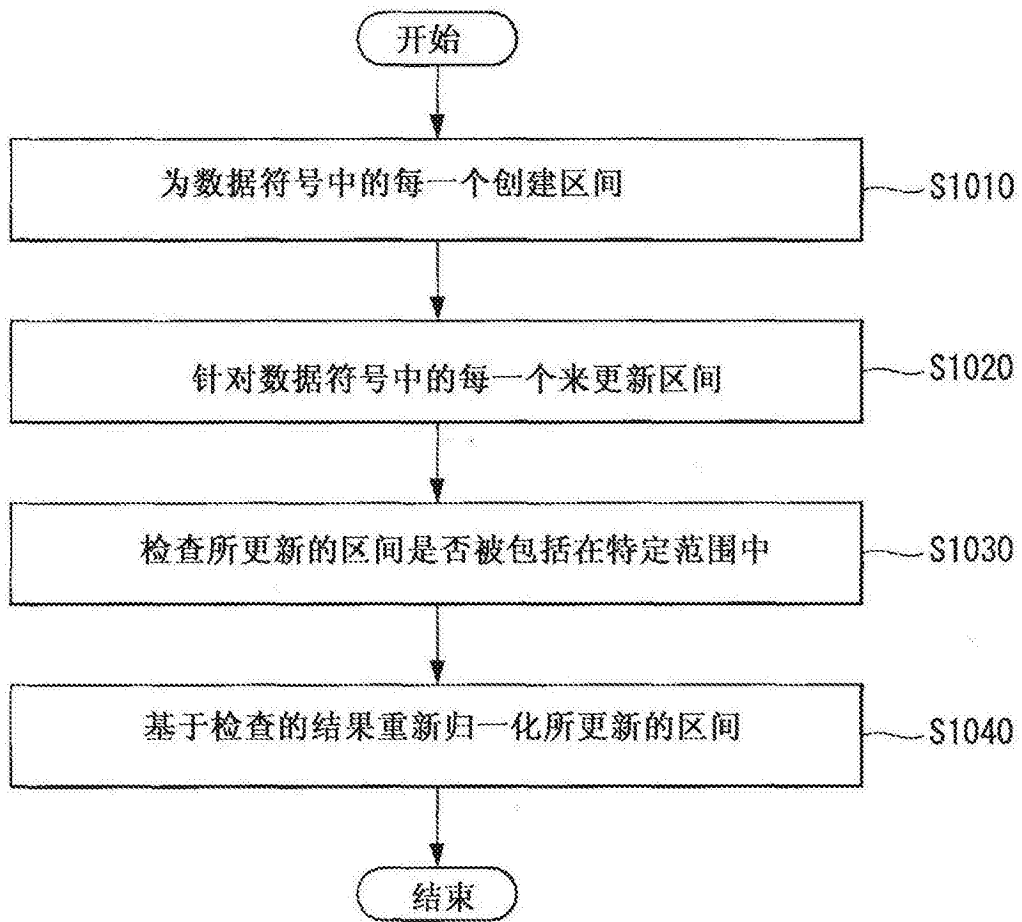


图10

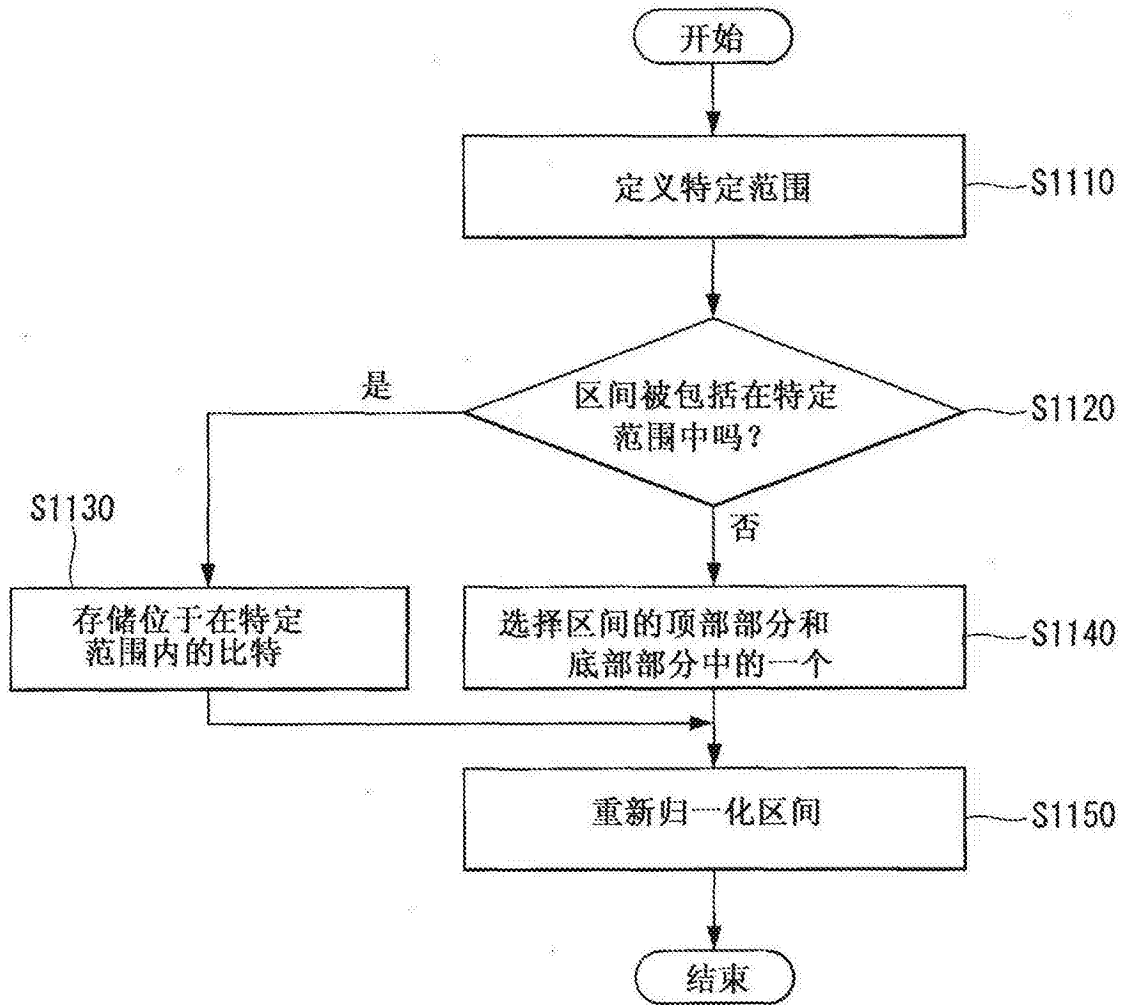


图11

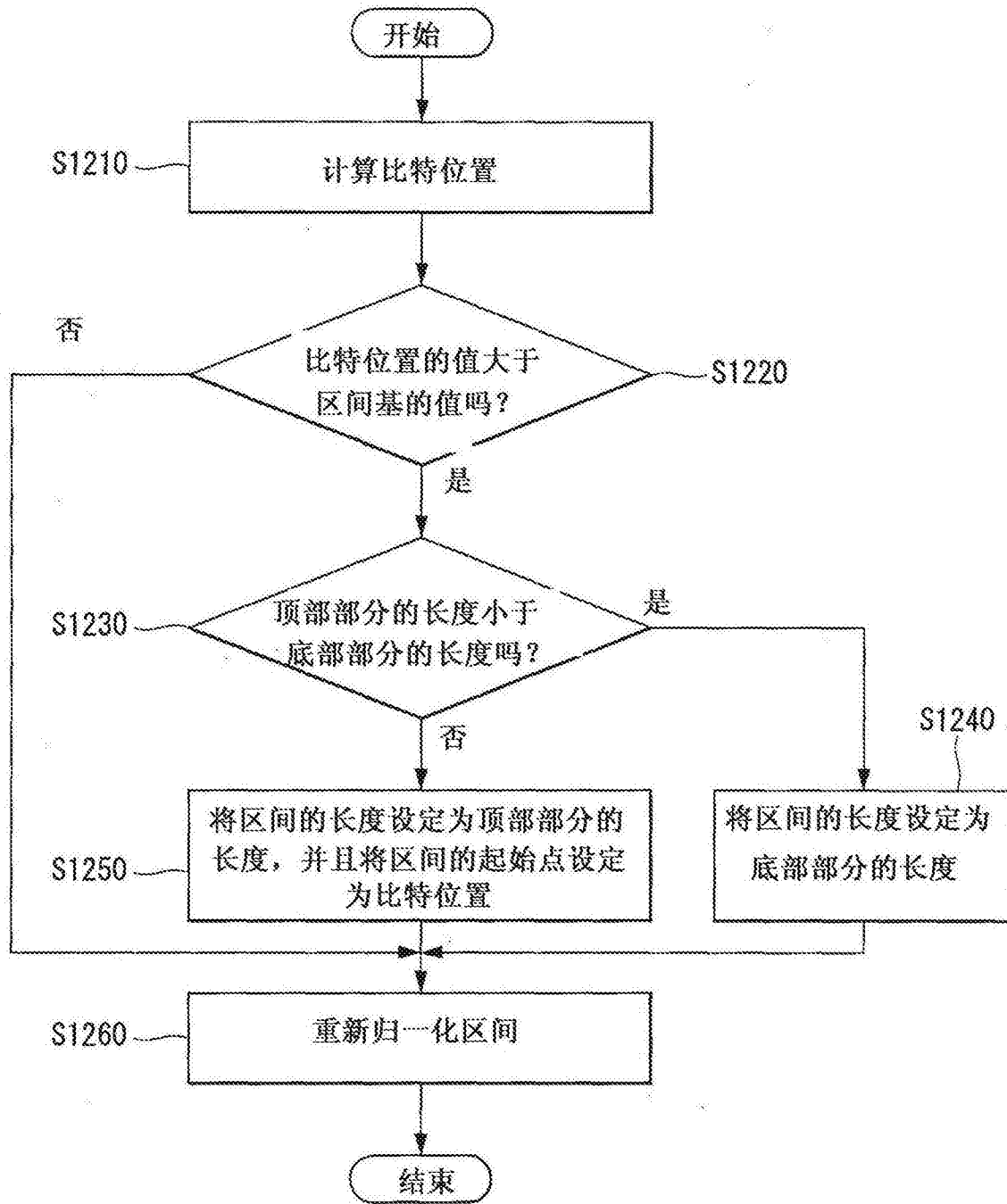


图12

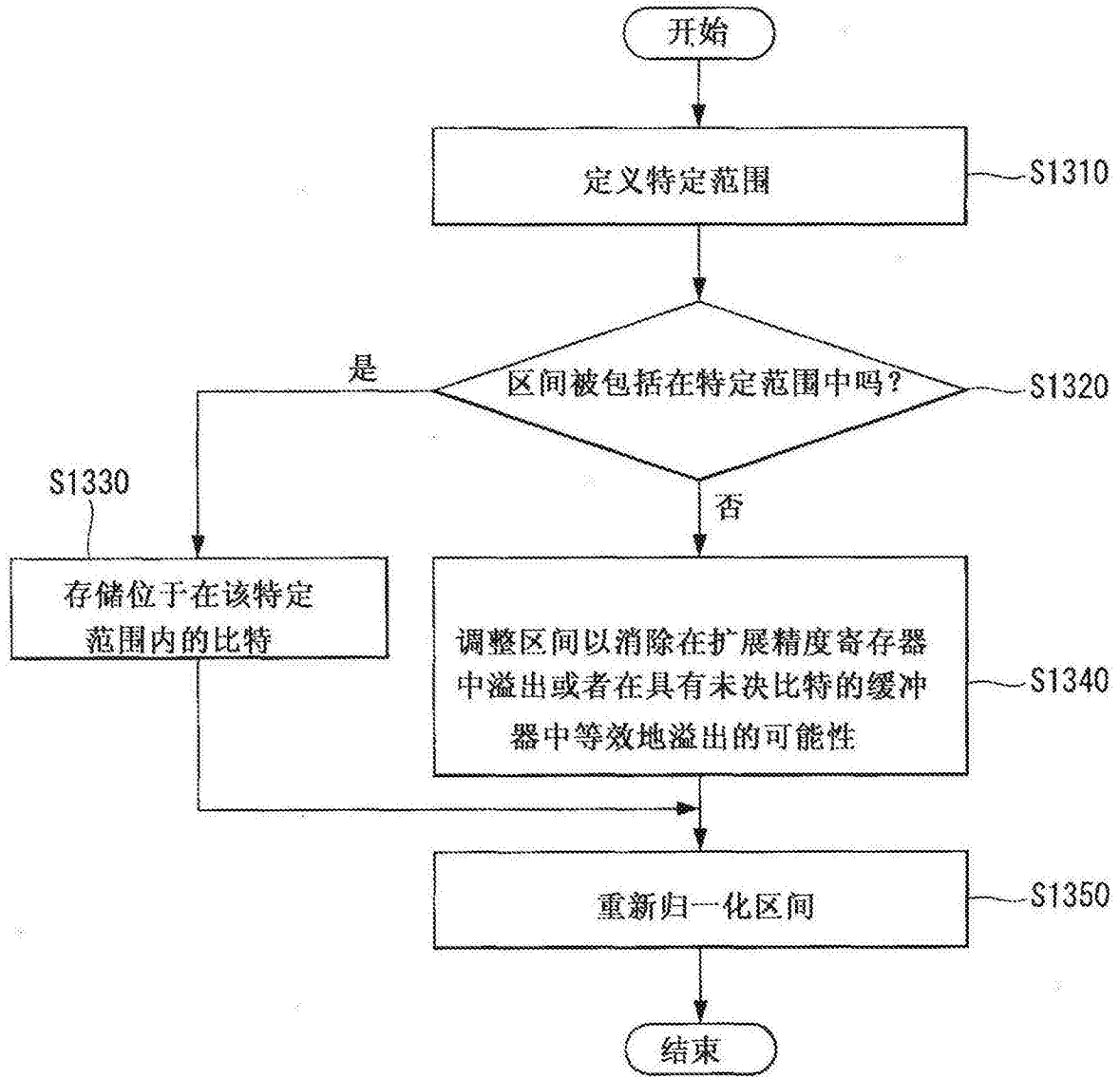


图13

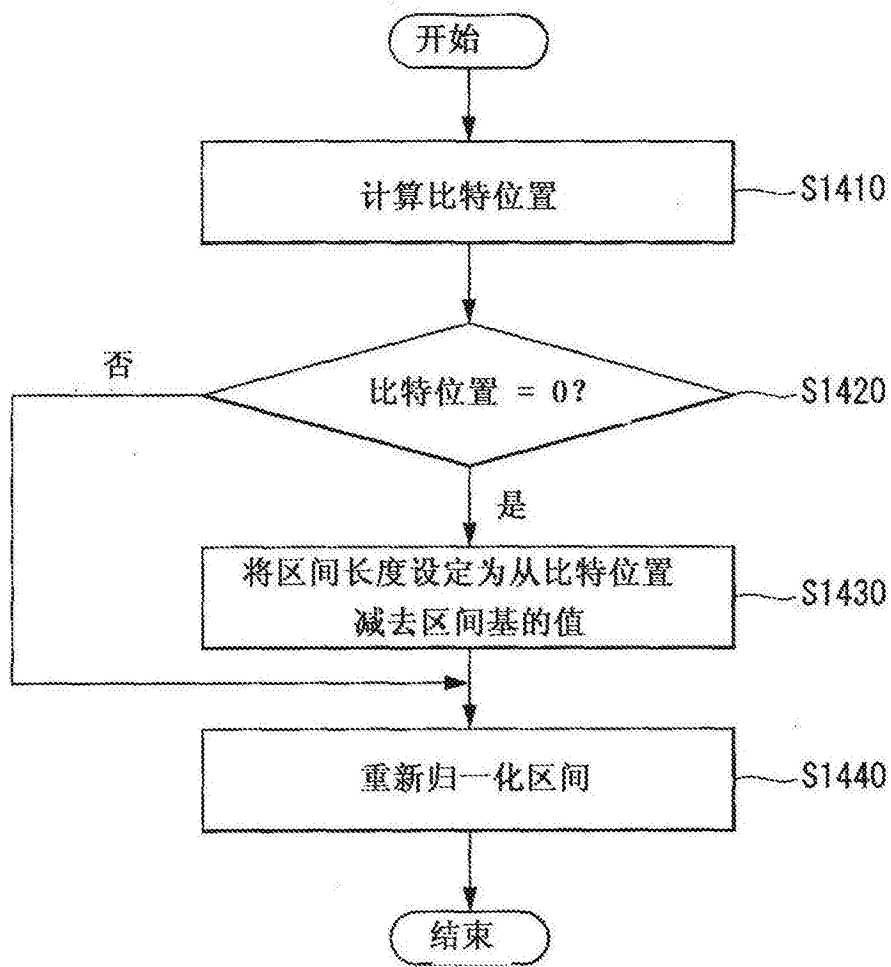


图14