



(12) 发明专利

(10) 授权公告号 CN 109923861 B

(45) 授权公告日 2021.10.08

(21) 申请号 201780068190.2

(22) 申请日 2017.08.31

(65) 同一申请的已公布的文献号

申请公布号 CN 109923861 A

(43) 申请公布日 2019.06.21

(30) 优先权数据

16306307.6 2016.10.05 EP

(85) PCT国际申请进入国家阶段日

2019.05.05

(86) PCT国际申请的申请数据

PCT/EP2017/071847 2017.08.31

(87) PCT国际申请的公布数据

W02018/065154 EN 2018.04.12

(73) 专利权人 交互数字麦迪逊专利控股公司

地址 法国巴黎

(72) 发明人 F.莱利内克 G.拉思 F.厄本

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 刘茵

(51) Int.Cl.

H04N 19/13 (2006.01)

(续)

(56) 对比文件

WO 2013006446 A1,2013.01.10

US 2013003824 A1,2013.01.03

CN 104081777 A,2014.10.01

CN 104205830 A,2014.12.10

WO 2008061008 A3,2008.07.10

US 2007071090 A1,2007.03.29

CN 103733627 A,2014.04.16

CN 102769744 A,2012.11.07

(续)

审查员 刁春帆

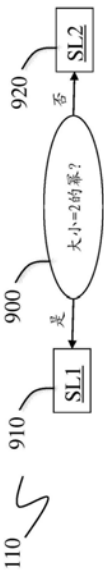
权利要求书3页 说明书17页 附图14页

(54) 发明名称

用于上下文自适应二进制算术编码的方法和  
设备

(57) 摘要

本原理涉及一种对二进制符号序列进行上下文自适应二进制算术编码/解码的方法设备，该二进制符号序列表示与属于图片块的图片数据有关的语法元素。所述方法包括，对于二进制符号序列的每个二进制符号：-根据图片块的大小和要编码的二进制符号来确定(110)上下文；以及-考虑所述确定的上下文，对所述二进制符号进行算术编码/解码(120)；其特征在于，所述确定(110)上下文包括：-当所述图片块的大小是2的幂时，根据所述图片块的大小来确定(910)上下文；以及-当所述图片块的大小不是2的幂时，根据紧接地高于或低于所述图片块的大小的块大小来确定(920)上下文，所述块大小等于2的幂。



[接上页]

(51) Int.Cl.

*H04N 19/91* (2006.01)

*H04N 19/157* (2006.01)

*H04N 19/18* (2006.01)

(56) 对比文件

CN 104205835 A, 2014.12.10

Joel Sole.Transform Coefficient

Coding in HEVC.《IEEE TRANSACTIONS ON  
CIRCUITS AND SYSTEMS FOR VIDEO  
TECHNOLOGY》.2012,

Detlev Marpe.Context-Based Adaptive  
Binary Arithmetic Coding in the H.264/AVC  
Video Compression Standard.《IEEE  
TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR  
VIDEO TECHNOLOGY》.2003,

1. 一种对二进制符号进行上下文自适应二进制算术编码或解码的方法,所述二进制符号表示与图片块的图片数据有关的语法元素,所述方法包括:

- 当所述图片块的大小是2的幂时,按照根据所述图片块的大小所确定的上下文对所述二进制符号进行算术编码或解码;以及

- 当所述图片块的大小不是2的幂时,按照根据紧接地高于或低于所述图片块的大小的块大小所确定的上下文对所述二进制符号进行算术编码或解码,所述块大小等于2的幂。

2. 如权利要求1所述的方法,其中块大小或图片块的大小是所述块的宽度或高度。

3. 如权利要求1所述的方法,其中通过以下根据所述图片块的大小或者紧接地高于或低于所述图片块的大小的块大小来确定上下文:

- 根据等于2的幂的不同图片块大小来获得至少一个上下文集合,每个上下文集合是针对等于2的特定幂的图片块大小获得的;

- 检查所述图片块的大小是否是2的幂;

- 如果所述图片块的大小是2的幂,则根据所述图片块的大小,在所述获得的至少一个上下文集合之中选择上下文集合;

- 否则,根据紧接地高于或低于所述图片块的大小的块大小,在所述获得的至少一个上下文集合之中选择上下文集合,所述块大小等于2的幂;以及

- 根据所述二进制符号序列中的二进制符号的索引,在所述选择的上下文集合之中确定上下文。

4. 如权利要求3所述的方法,其中与图片数据有关的所述语法元素表示当根据扫描顺序扫描所述图片块时属于所述图片块的最后有效系数的空间坐标之一,并且所述方法还包括用于以下的步骤:

- 在至少一个区间索引的集合之中选择识别整数值的区间的区间索引,其中所述最后有效系数的空间坐标之一属于所述区间;

- 通过二值化所述选择的区间索引来获得所述二进制符号序列;以及

- 对所述二进制符号序列的每个二进制数进行编码。

5. 如权利要求4所述的方法,其中所述方法还包括获得所述选择的区间内的偏移值,以便所述最后有效系数的空间坐标之一由所述区间索引和所述偏移值无损地表示。

6. 如权利要求1所述的方法,其中所述图片块被分裂成至少两个图片子块,与图片数据有关的所述语法元素表示与所述至少两个图片子块中的一个图片子块的系数有关的有效系数标志,并且其中通过以下根据所述图片块的大小或者紧接地高于或低于所述图片块的大小的块大小来确定上下文:

- 检查所述图片块的大小是否是2的幂;

- 如果所述图片块的大小是2的幂,则根据所述图片块的大小来确定基本上下文索引;

- 否则,根据紧接地高于或低于所述图片块的大小的块区域的大小来确定基本上下文索引,所述块区域的大小等于2的幂;以及

- 通过将所述基本上下文索引添加到从相邻系数的有效变换系数标志获得的上下文增量来确定所述上下文索引。

7. 如权利要求1-6之一所述的方法,其中所述图片块或子块的大小是所述图片块或子块的宽度和/或高度,并且所述方法还包括获得图片块或子块,所述图片块或所述子块具有

等于整数与2的幂的乘积以便所述乘积不是2的幂的高度和/或宽度。

8. 如权利要求7所述的方法, 其中获得图片块包括将父块分裂成两个子块, 所述子块具有等于所述整数与2的幂的乘积的高度和/或宽度。

9. 一种用于对二进制符号进行上下文自适应二进制算术编码或解码的设备, 所述二进制符号表示与图片块的图片数据有关的语法元素, 所述设备包括至少一个处理器, 所述至少一个处理器被配置为:

- 当所述图片块的大小是2的幂时, 按照根据所述图片块的大小所确定的上下文对所述二进制符号进行算术编码或解码; 以及

- 当所述图片块的大小不是2的幂时, 按照根据紧接地高于或低于所述图片块的大小的块大小所确定的上下文对所述二进制符号进行算术编码或解码, 所述块大小等于2的幂。

10. 如权利要求9所述的设备, 其中块大小或图片块的大小是所述块的宽度或高度。

11. 如权利要求9所述的设备, 其中通过以下根据所述图片块的大小或者紧接地高于或低于所述图片块的大小的块大小来确定上下文:

- 根据等于2的幂的不同图片块大小来获得至少一个上下文集合, 每个上下文集合是针对等于2的特定幂的图片块大小获得的;

- 检查所述图片块的大小是否是2的幂;

- 如果所述图片块的大小是2的幂, 则根据所述图片块的大小, 在所述获得的至少一个上下文集合之中选择上下文集合;

- 否则, 根据紧接地高于或低于所述图片块的大小的块大小, 在所述获得的至少一个上下文集合之中选择上下文集合, 所述块大小等于2的幂; 以及

- 根据所述二进制符号序列中的二进制符号的索引, 在所选上下文集合之中确定上下文。

12. 如权利要求11所述的设备, 其中与图片数据有关的所述语法元素表示当根据扫描顺序扫描所述图片块时属于所述图片块的最后有效系数的空间坐标之一, 并且所述至少一个处理器还被配置为:

- 在至少一个区间索引的集合之中选择识别整数值的区间的区间索引, 其中所述最后有效系数的空间坐标之一属于所述区间;

- 通过二值化所述选择的区间索引来获得所述二进制符号序列; 以及

- 对所述二进制符号序列的每个二进制数进行编码。

13. 如权利要求12所述的设备, 其中所述至少一个处理器还被配置为获得所述选择的区间内的偏移值, 以便所述最后有效系数的空间坐标之一由所述区间索引和所述偏移值无损地表示。

14. 如权利要求9所述的设备, 其中所述图片块被分裂成至少两个图片子块, 与图片数据有关的所述语法元素表示与所述至少两个图片子块中的一个图片子块的系数有关的有效系数标志, 并且其中通过以下根据所述图片块的大小或者紧接地高于或低于所述图片块的大小的块大小来确定上下文:

- 检查所述图片块的大小是否是2的幂;

- 如果所述图片块的大小是2的幂, 则根据所述图片块的大小来确定基本上下文索引;

- 否则, 根据紧接地高于或低于所述图片块的大小的块区域的大小来确定基本上下文

索引,所述块区域的大小等于2的幂;以及

-通过将所述基本上下文索引添加到从相邻系数的有效变换系数标志获得的上下文增量来确定所述上下文索引。

15.如权利要求9-14之一所述的设备,其中所述图片块或子块的大小是所述图片块或子块的宽度和/或高度,并且所述至少一个处理器还被配置为获得图片块或子块,所述图片块或所述子块具有等于整数值与2的幂的乘积以便所述乘积不是2的幂的高度和/或宽度。

16.如权利要求15所述的设备,其中获得图片块包括将父块分裂成两个子块,所述子块具有等于所述整数值与2的幂的乘积的高度和/或宽度。

17.一种非临时性计算机可读介质,包括用于使一个或多个处理器执行对二进制符号进行上下文自适应二进制算术编码或解码的方法的指令,所述二进制符号表示与图片块的图片数据有关的语法元素,所述方法包括:

-当所述图片块的大小是2的幂时,按照根据所述图片块的大小所确定的上下文对所述二进制符号进行算术编码或解码;以及

-当所述图片块的大小不是2的幂时,按照根据紧接地高于或低于所述图片块的大小的块大小所确定的上下文对所述二进制符号进行算术编码或解码,所述块大小等于2的幂。

## 用于上下文自适应二进制算术编码的方法和设备

### 技术领域

[0001] 本原理一般地涉及图片/视频编码。具体地,本原理的技术领域涉及对表示解码所述图片数据所需的语法元素的二进制符号序列进行上下文自适应二进制算术编码。

### 背景技术

[0002] 本部分旨在向读者介绍技术的各个方面,其可能与下面描述和/或要求保护的本原理的各个方面有关。相信该讨论有助于向读者提供背景信息,以便更好地理解本原理的各个方面。因此,应当理解,要在该角度来阅读这些陈述,而不是将这些陈述作为对现有技术的承认。

[0003] 在下文中,图片数据包含以特定的视频格式的一个或若干个采样(像素值)阵列,该特定的视频格式指定与图片的像素值有关的所有信息以及例如可以由显示器和/或任何其他设备使用以对图片进行可视化和/或解码的所有信息。静止图片(或视频的图片)包括以第一采样阵列的形状的至少一个分量,通常为亮度(luma)(或照度(luminance))分量,以及可能地包括以至少一个其他采样阵列的形状的至少一个其他分量,通常为色彩分量。或者,等同地,相同的信息也可以通过彩色采样阵列的集合来表示,诸如传统的三色RGB表示。

[0004] 像素值是可以由C个值的矢量来表示的图片数据,其中C是分量的数量。矢量的每个值用定义像素值的最大动态范围的位数来表示。

[0005] 一般而言,图片块表示图片数据的元素或与图片数据有关的语法元素的集合。例如,图片块是属于图片的像素的集合。

[0006] 视频编码设备可以试图通过利用空间和时间冗余来压缩图片数据。例如,视频编码器可以通过相对于相邻的(先前编码块)对图片块进行编码来利用空间冗余。同样地,视频编码器可以通过相对于先前编码图片的数据对图片块进行编码来利用时间冗余。特别地,视频编码器可以从空间相邻者的数据或者从先前编码图片的数据来预测当前图片块。然后,视频编码器可以将图片块的残差计算为图片块的实际像素值与图片块的预测像素值之间的差。因此,图片块的残差可以包括像素(或空间)域中的逐像素差值。

[0007] 然后,视频编码器可以将变换应用于残差的值,以将像素值的能量压缩成频域中的相对少量的变换系数。然后,视频编码器可以对变换系数进行量化,扫描经量化变换系数以将经量化变换系数的二维矩阵转换成包含经量化变换系数的一维矢量。然后,视频编码器可以应用熵编码处理来对扫描系数进行熵编码。示例熵编码处理可以包括例如上下文自适应可变长度编码(CAVLC)、上下文自适应二进制算术编码(CABAC)、基于语法的上下文自适应二进制算术编码(SBAC)、概率区间划分熵(PIPE)编码或其他熵编码方法。

[0008] 视频解码器可以执行通常与由视频编码器执行的编码技术相对的技术。虽然通常是相对的,但是在一些情况下,视频解码器可以执行与由视频编码器执行的技术类似的技术。

[0009] 视频编码器也可以对与编码图片数据相关联的语法元素进行熵编码,以供视频解码器在解码图片数据中使用。然后,视频解码器可以依赖于包含在接收到的比特流中的语

法元素或其他数据,该比特流包括关于视频编码器描述的数据。

[0010] 根据语法元素的一些示例,“最后有效系数坐标”是语法元素,其根据与图片块相关联的扫描顺序向解码器提供图片块中的最后非零系数的空间位置。然后,可以对“最后有效系数x坐标”语法元素进行编码,以用于向解码器提供图片块中的最后有效系数的x坐标,并且可以对“最后有效系数 y坐标”语法元素进行编码,以用于向解码器提供图片块中的最后有效系数的y坐标。“有效系数标志”是另一语法元素,其指示图片块中的经量化变换系数是否等于零。

[0011] 为了对经量化变换系数的级别进行熵编码,视频编码器可以对每个经量化变换系数的级别进行熵编码。例如,视频编码器可以将每个非零的经量化变换系数的绝对值转换成二进制形式。以该方式,每个非零的变换系数可以被“二值化”,例如,使用包括一个或多个位或“二进制数(bin)”的一元代码。也可以对用于经量化变换系数的符号的位进行编码。

[0012] 另外,可以包括多个其他语法元素以允许视频解码器对图片数据进行解码。例如,如H.264/MPEG-4AVC(“Advanced video coding for generic audiovisual Services”,系列H:视听和多媒体系统,ITU-T H.264建议书,ITU 电信标准化部门,2014年2月)和高效视频编码(HEVC)标准(ITU-T H.265 ITU电信标准化部门(10/2014),H系列:视听和多媒体系统,视听服务基础架构-移动视频编码,高效视频编码,ITU-T H.265建议书)中所描述的,语法元素可以表示运动矢量残差、变换单元编码标志、编码组编码标志、经量化变换系数幅度(大于1和大于2)标志、SA0数据等。

[0013] 视频编码器可以使用每个二进制符号的概率估计,对图片数据块的每个二进制符号(或二进制数)进行熵编码,无论其与图片块的经量化变换系数还是语法元素(信息)相对应。概率估计可以指示具有给定二进制值(例如,“0”或“1”)的二进制符号的似然性。概率估计可以包括在也称为“上下文”的概率模型内。然后,视频编码器可以确定要编码的二进制符号的上下文(包括概率模型)。

[0014] 针对表示语法元素的二进制符号序列的二进制符号,上下文可以包括先前编码的相邻语法元素的相关二进制符号的值。上下文可以属于上下文集合,并且于是从识别所述上下文集合中的所述上下文的上下文索引来获得上下文。

[0015] 作为一个示例,针对与属于图片块的经量化变换系数有关的有效系数标志,上下文包括所述图片块的类型(例如,图片块大小,亮度或色度元素的块),以及根据与图片块相关联的扫描顺序,所述经量化变换系数在图片块内的空间位置。

[0016] 作为另一示例,针对图片块的二值化残差变换系数绝对值的二进制符号的上下文包括二进制符号在表示绝对值的一元码字内的空间位置,以及图片块的先前编码系数的值。

[0017] 在其他示例中,针对表示编码块图案(“CBP”)的二进制符号序列的二进制符号的上下文包括先前编码的相邻语法元素(例如,在当前语法元素的顶部和左侧)的相关二进制符号的值。

[0018] 本原理不限于以上示例,而是扩展到任何语法元素和上下文,诸如例如在H.264/MPEG-4AVC和高效视频编码(HEVC)标准中定义的。

[0019] 在任何情况下,针对每个上下文定义不同的概率模型。

[0020] 在对二进制符号进行熵编码之后,可以根据编码二进制符号的值来更新概率模型

以反映最当前的概率估计。

[0021] 可以在一个或多个编码“通道”中对与图片块相关联的二进制符号进行编码。例如,在第一通道期间,视频编码器可以对有效系数标志和最后有效系数坐标进行熵编码。在第二通道期间,视频编码器可以对经量化变换系数级别的第一二进制数进行熵编码。视频编码器可以继续执行编码通道,直到与图片块的经量化变换系数相关联的所有信息被编码。

[0022] 在一些示例中,视频编码器可以使用上下文自适应和非上下文自适应编码的组合对图片块的二进制数进行编码。例如,对于一个或多个通道,视频编码器可以使用旁路模式来绕过或省略常规算术编码处理。在这种情况下,可以使用固定等概率模型对旁路编码二进制数进行编码。旁路编码二进制数不包括上下文或概率更新。

[0023] 当执行上下文自适应编码时,在一些示例中,可能由于多个反馈环路而存在相对高的串行依赖性。

[0024] 例如,指示用于对二进制符号进行编码的具体概率模型的上下文可能受先前编码的二进制符号(例如,先前编码的语法元素的相关二进制符号)的值的影响。

[0025] 另外,用于对二进制符号进行编码的概率模型也可能受先前编码的二进制符号的值的影响。也就是说,概率模型可以被维持为有限状态机中的状态。每个具体状态可以对应于特定概率值。与概率模型的更新相对应的下个状态可以取决于当前二进制符号的值(例如,当前正被编码的二进制数)。

[0026] 另外,如上所述,具体概率模型可以被维持为有限状态机中的状态,其中每个具体状态对应于特定概率值。与概率模型的更新相对应的下个状态可以取决于当前二进制符号的值。可以称为状态更新的这样的反馈环路可能呈现计算瓶颈。例如,可能由于反馈回路中的依赖性而难以增加吞吐量。也就是说,一个二进制符号可能不被处理,直到来自先前二进制符号的概率模型已被更新。在一些示例中,可以以高频率连续地(逐个二进制符号)调用相同的上下文。

[0027] 上下文自适应二进制算术编码(CABAC)是H.264/MPEG-4和HEVC 中使用的熵编码的形式。

[0028] CABAC是无损压缩技术,尽管其中使用的视频编码标准典型地用于有损压缩应用。

[0029] 图1示出了在CABAC中对语法元素进行编码的一般块图。

[0030] 基本上,CABAC包括三个基本步骤。

[0031] 在步骤100中,二值化器将给定的有非二进制值的语法元素映射到二进制符号序列(二进制数字符串)。当给定有二进制值的语法元素时,绕过该初始步骤,如图1中所示。

[0032] 对于二进制数字符串的每个当前二进制数或者对于每个有二进制值的语法元素,取决于编码模式,可以遵循一个或多个后续步骤。

[0033] 在所谓的常规编码模式中,上下文建模步骤110之后是常规编码步骤 120。

[0034] 在上下文建模步骤110中,上下文建模器根据要编码的二进制数(通常为识别若干个上下文模型的集合之中的上下文模型的上下文索引)选择上下文模型,并且考虑在当前二进制数正被编码时,以同步方式在编码器和解码器侧上都可用的上下文信息(诸如先前编码的语法元素或编码的二进制符号),从所选上下文模型中确定上下文。然后,使用所确定的上下文来获得当前二进制数具有特定值(例如‘0’)的条件概率。

[0035] 然后,在常规编码步骤120中,将所确定的上下文(条件概率)和当前二进制数的值  $s$  (图1中的二进制数值)传递给引擎,该引擎考虑所确定的上下文对当前二进制数进行算术编码。当前二进制数的编码值由此获得并用于更新上下文信息。这是图1上的上下文模型更新反馈环路。更新上下文信息涉及针对每个二进制数更新上下文(只要该二进制数正被编码),从而允许逐步细化针对二进制数字字符串的每个二进制数的上下文建模。因此, CABAC 编码器逐步学习每个常规编码二进制数的统计行为。

[0036] 常规编码位(常规编码步骤120的输出)被添加到输出比特流。

[0037] 备选地,选择所谓的旁路编码模式以用于对二进制数进行编码,以便允许通过简化的编码引擎来加速整个编码(和解码)处理,而不使用明确分配的概率模型,如图1中的开关的右下分支所示。旁路模式包括简单地在输出比特流中写入一位并将因此获得的位值分配给当前二进制数。该编码模式具有直接、因此快速的优点。其典型地有效,因此用于具有均匀统计分布(即等于‘1’或‘0’的相等概率)的二进制数。

[0038] 图2图示了当二进制符号序列表示与图片块有关的有效性系数标志时的 CABAC 编码的示例。

[0039] 这是非限制性示例,因为本原理适用于表示语法元素的任何二进制符号序列。

[0040] 在图2的示例中,表示图片块(这里是二进制符号的图片块),该图片块表示有效性系数标志的块(标志可以与图片块的每个经量化变换系数相关联)。如果经量化变换系数等于0,则该标志等于0,否则为1。根据特定的扫描顺序来扫描该图片块。例如,由虚线示出的扫描顺序从图片块的右下角开始到左上角。假设二进制符号  $s$  要被编码,根据所述扫描顺序,先前为二进制符号  $s$  定义若干个候选上下文  $C_i$ , 并且每个上下文  $C_i$  与识别候选上下文集合中的上下文  $C_i$  的上下文索引  $\text{ContVal}_i$  相关联。

[0041] 例如,从两个先前编码的语法元素  $x_1$  和  $x_2$  (有效性系数标志) 为二进制符号  $s$  定义四个候选上下文  $C_i$  ( $i=1, \dots, 4$ )。根据  $x_1$  和  $x_2$  的二进制值来定义每个上下文  $C_i$ , 并且将特定上下文索引  $\text{ContVal}_i$  分配给每个上下文。

[0042] 如图3中所示的示例,上下文索引包括表示二进制符号  $s$  的特定二进制值(所谓的最可能符号(MPS))的位  $\text{valMPS}$  和表示从中推导概率  $p$  的概率  $p'$  (或状态)的位(例如7位)  $\text{pStateIdx}$ 。

[0043] 根据示例,在上下文建模步骤110中,根据  $x_1$  和  $x_2$  的值,从四个候选上下文中选择上下文  $C_s$ , 并且将关联的上下文索引  $\text{ContVal}_s$  分配给二进制符号  $s$ 。

[0044] 在常规编码阶段120中,取决于二进制符号  $s$  是否等于MPS,遵循图4 中所示的过程来进行具有索引  $\text{ContVal}_s$  的所选上下文的上下文模型的更新。

[0045] 通过两个表格  $\text{transIdxMPS}$  (如果二进制符号  $s$  是MPS的话) 和  $\text{transIdxLPS}$  (如果二进制符号  $s$  不是MPS的话,即它是最小可能符号(LPS)) 进行演进。针对条目  $\text{pStateIdx}$ , 在图5中提供这些表格。

[0046] 遵循该示例,二进制符号  $s$  为MPS的概率  $p_{\text{MPS}}$  被线性地量化为8位,从0到127。它通过以下从所选上下文索引中推导

[0047]  $p_{\text{MPS}} = (p' + 64) / 127 = (\text{pStateIdx} + 64) / 127$

[0048] 并且取决于MPS的值,从  $p_{\text{MPS}}$  显然地推导符号  $s$  为1的概率  $p$ 。

[0049] 如果  $\text{MPS} = 1$ , 则  $p = p_{\text{MPS}}$ ,

[0050] 如果 $MPS=0$ ,则 $p=1-p_{MPS}$ 。

[0051] 上下文自适应编码是功能强大的工具,它允许动态地遵循与编码图片数据有关的不同语法元素的统计。

[0052] 因此,使用CABAC以用于对图片数据和/或与图片数据有关的语法元素进行熵编码已导致在HEVC/H265中广泛使用很多上下文(高达数百个),以便对其统计进行建模。

[0053] 选择上下文索引取决于很多事情,因此通常使用大量上下文以便捕获语法元素的统计,并且从而优化CABAC的性能。然而,因为上下文应存储在存储器中以用于对一个或多个二进制符号序列的所有二进制符号进行编码,所以增加上下文的数量会增加存储器和处理复杂性方面的要求。这就是为何限制上下文的数量对确保视频解码器的有效硬件可实现性至关重要,该视频解码器实现用于对语法元素进行编码的CABAC。

[0054] 因此,挑战之一是受益于由CABAC提供的增加的压缩能力而不增加太多上下文的数量。

[0055] 图6示出了描绘给定包括根据图1的方法编码的位的输入比特流,对语法元素进行解码的CABAC的一般块图。输入比特流可以符合HEVC规范。

[0056] 在解码的任何点,假设解码器知道接下来要解码哪个语法元素。这在标准化比特流语法和解码处理中完全指定。此外,还假设如何将要解码的当前语法元素二值化(即表示为二进制数字串,每个二进制数等于‘1’或‘0’),以及用于编码每个二进制数的编码模式是什么(常规编码或旁路模式)。

[0057] 这些信息在所考虑的视频压缩标准(H.264或HEVC)中完全指定。

[0058] 因此,在步骤600中,解析器通过从输入比特流解码(解析)编码位来获得一系列的位。

[0059] 对于要解码的每个当前编码二进制数,取决于用于编码所述当前编码二进制数的编码模式,可以遵循一个或多个后续步骤。

[0060] 当使用了旁路编码模式来编码当前编码二进制数时,在旁路解码步骤 610中,当前编码二进制数的值是当前编码二进制数的解码值。

[0061] 当使用了常规编码模式来编码当前编码二进制数时,上下文建模步骤 110之后是常规解码步骤620。

[0062] 图1和图6的上下文建模步骤严格相同,并且图6的常规解码步骤620 与图1的常规编码步骤110非常相似。

[0063] 在常规解码步骤620中,从由上下文建模步骤110给出的上下文获得当前编码二进制数的解码值:所述上下文(条件概率)和当前位的值被传递到引擎,该引擎的算法考虑所述上下文对当前二进制数进行解码。当前二进制数的解码值由此获得并用于更新上下文信息。这是图6上的上下文模型更新反馈环路。更新上下文信息涉及针对每个二进制数更新上下文(只要该二进制数正被解码),从而允许逐步细化针对每个二进制数的上下文建模。因此,CABAC解码器逐步获得等于当前二进制数的原始值的当前编码二进制数的解码值(无损压缩)。

[0064] 当获得当前二进制数的解码值(旁路解码步骤610或常规解码步骤620 的输出)时,当前二进制数的解码值与有二进制值的语法元素或者有非二进制值的语法元素有关。这可以由所考虑的视频压缩标准(H.264或HEVC) 指定。接下来,如果当前二进制数的解码

值与有二进制值的语法元素有关,则CABAC解码结束,并且在当前二进制数的解码值与有非二进制值的语法元素有关时,解码器知道如何从二进制数的解码值(步骤620或610的输出二进制数字字符串)构建语法元素。

[0065] 在步骤630中,然后,转换器通过应用与二值化(步骤100)相对的处理将所述输出二进制数字字符串转换成有非二进制值的语法元素。

[0066] 在视频压缩中,众所周知的是,将整个图片编码为单个图片块不是最佳的,并且在

对编码树的每个叶片进行编码之前将整个图片递归地划分成子块会增加编码效率。

[0067] 例如,在HEVC标准中,要编码的图片被划分成典型地具有正方形形状 ( $64 \times 64$ ,  $128 \times 128$ 或 $256 \times 256$ 个像素)的编码树单元(CTU)。每个CTU 由压缩域中的编码树表示。编码树描述CTU的递归分裂。如果使用四叉树划分,则编码树的每个叶片(编码单元)具有正方形形状,但是当使用四叉树加二叉树(QTBT)分解时,一些叶片也可以具有矩形形状。更准确地说,CTU的QTBT分解由两个阶段构成:首先以四叉树方式分割CTU,然后可以以二进制方式进一步划分每个四叉树叶片。这在图7的右侧示出,其中实线表示四叉树分解阶段,而虚线表示在空间上嵌入四叉树叶片中的二进制分解。通过速率失真优化过程在编码器侧判定编码单元的分裂,该过程包括确定具有最小速率失真成本的CTU的QTBT表示。

[0068] 在HEVC标准中,编码单元的大小(即宽度和高度)总是2的幂,并且典型地从4到128。

[0069] 发明人发现,在QTBT分解中允许不对称分裂模式会增加编码效率,因为编码树更好地匹配提供具有更均一内容的编码单元的图片中的空间结构。

[0070] 非对称分裂模式包括将具有等于整数值 $m$ 的倍数的大小(宽度和/或高度)的编码单元分裂成具有等于 $m \cdot 2^n$ 的大小(宽度和/或高度)的两个子编码单元,其中 $n$ 是整数值, $m$ 是为了使 $m \cdot 2^n$ 不等于2的幂而选择的整数值。

[0071] 图8示出了当 $m$ 等于3时的四个不对称分裂模式的示例。这些示例仅出于说明性目的而给出,并且可以扩展到其他值的 $m$ ,诸如5、7等等。在具有等于 $m$ 的倍数的大小(宽度 $w$ 和/或高度 $h$ )的编码单元的编码中引入不对称分裂模式,允许通过所述不对称分裂模式之一来分裂所述编码单元。例如,根据图8中的非对称分裂模式HOR\_UP(水平向上)将具有等于3的倍数的宽度 $w$ 和高度 $h$ 的编码单元分裂,将导致具有各自矩形大小 $(w, \frac{h}{4})$ 和 $(w, \frac{3h}{4})$ 的2个子编码单元。

[0072] 当使用非对称分裂模式时,一些编码单元具有等于 $m \cdot 2^n$ 的宽度和/或高度。于是,最初设计用于对大小等于2的幂的编码单元进行编码的通常HEVC 工具必须适配于这些不等于2的幂的编码单元的新的尺寸。

[0073] 本原理解决的问题之一是如何以提供良好的压缩效率(速率失真性能)以及编码设计的最小复杂性增加的方式,有效地对包含在非对称编码单元中的经量化变换系数进行熵编码/解码。换句话说,挑战之一是受益于由使用非对称分裂模式提供的增加的压缩能力,而不增加由HEVC熵编码(CABAC)使用的上下文的数量。

## 发明内容

[0074] 以下呈现本原理的简要概述,以便提供对本原理的一些方面的基本理解。本概述

不是对本原理的广泛概述。其并不旨在识别本原理的关键或重要元素。

[0075] 以下概述仅以简化形式呈现本原理的一些方面,作为下面提供的更详细描述的前言。

[0076] 本原理旨在利用对表示与属于图片块的图片数据有关的语法元素的二进制符号序列进行上下文自适应二进制算术编码/解码的方法,来补救现有技术的至少一个缺点。所述方法包括,对于二进制符号序列的每个二进制符号:

[0077] -根据图片块的大小和要编码的二进制符号来确定上下文;以及

[0078] -考虑所述确定的上下文,对所述二进制符号进行算术编码/解码;

[0079] 其特征在于,所述确定上下文包括:

[0080] -当所述图片块的大小是2的幂时,根据所述图片块的大小来确定上下文;以及

[0081] -当所述图片块的大小不是2的幂时,根据紧接地(immediately)高于或低于所述图片块的大小的块大小来确定上下文,所述块大小等于2的幂。

[0082] 根据本原理的其他方面,本原理涉及一种设备,以及一种计算机程序产品,其包括程序代码指令,以当在计算机上执行该程序时执行以上方法的步骤,以及一种非临时性储存介质,其携带程序代码的指令,以用于当在计算设备上执行所述程序时执行以上方法的步骤。

[0083] 从以下结合附图的示例的描述中,本原理的特定性质以及本原理的其他目的、优点、特征和用途将变得明显。

## 附图说明

[0084] 在附图中,图示了本原理的示例。附图示出:

[0085] -图1示出了使用CABAC对语法元素进行编码的一般块图;

[0086] -图2图示了当二进制符号序列表示与视频数据块有关的有效性图时的上下文选择阶段的示例;

[0087] -图3示出了上下文索引的示例;

[0088] -图4示出了更新上下文索引的一般块图;

[0089] -图5示出了通常用于上下文索引更新的众所周知的表格;

[0090] -图6图示了根据本原理的示例的对二进制符号序列的二进制符号s进行上下文自适应二进制算术解码的方法;

[0091] -图7图示了当二进制符号s表示例如与视频数据块有关的有效性图的有效性标志时的图6的方法的实施例的示例;

[0092] -图8示出了当m等于3时的四个不对称分裂模式的示例;

[0093] -图9示出了图示根据本原理的示例的根据图片块的大小针对二进制符号在若干个上下文的集合之中选择上下文的块图(步骤110);

[0094] -图10示出了8×8TB中的扫描顺序的示例;

[0095] -图11示出了用于“最后有效系数坐标”语法元素的二值化的表格;

[0096] -图12和图13示出了根据不同TB大小给出上下文索引集合的表格的示例;

[0097] -图14示出了根据本原理的实施例的计算上下文增量ctxIncrement的示例;

[0098] -图15示出了根据本原理的示例的设备的架构的示例。

[0099] 相似或相同的元素用相同的参考标号来表示。

### 具体实施方式

[0100] 在下文中将参考附图更全面地描述本原理,附图中示出了本原理的示例。然而,本原理可以以很多替选形式来实施,并且不应被解释为限于在此阐述的示例。因此,虽然本原理易有各种修改和替选形式,但是其特定示例通过附图中的示例的方式示出并且将在此详细描述。然而,应理解,并不旨在将本原理限于所公开的具体形式,而是相反,本公开要覆盖落入如由权利要求限定的本原理的精神和范围内的所有修改、等同和替选。

[0101] 在此使用的术语仅用于描述具体示例的目的,并不旨在限制本原理。如在此使用的,单数形式“一(a)”,“一(an)”和“该”旨在也包括复数形式,除非上下文清楚地另有指示。还将理解,当在本说明书中使用时,术语“包括(comprises)”、“包括(comprising)”,“包含(includes)”和/或“包含(including)”指定存在所述特征、整数、步骤、操作、元素和/或组件,但是不排除存在或添加一个或多个其他特征、整数、步骤、操作、元素、组件和/或其群组。此外,当元素被称为“响应”或“连接”于另一元素时,它可以直接响应或连接于其他元素,或者可以存在中间元素。相反,当元素被称为“直接响应”或“直接连接”于其他元素时,不存在中间元素。如在此使用的,术语“和/或”包括一个或多个关联所列项中的任何和所有组合,并且可以缩写为“/”。

[0102] 将理解,尽管可以在此使用术语第一、第二等来描述各种元素,但是这些元素不应受这些术语的限制。这些术语仅用于将一个元素与另一元素区分开。例如,第一元素可以被称为第二元素,并且类似地,第二元素可以被称为第一元素而不脱离本原理的教导。

[0103] 尽管一些图包括通信路径上的箭头以示出通信的主要方向,但是要理解,通信可以发生在与所示箭头相反的方向上。

[0104] 关于块图 and 操作流程图描述一些示例,其中每个块表示电路元件、模块或代码部分,该代码部分包括用于实现指定逻辑功能的一个或多个可执行指令。还应注意,在其他实现方式中,块中注释的功能可以不按所注释的顺序发生。例如,连续示出的两个块实际上可以基本同时执行,或者这些块有时可以以相反的顺序执行,这取决于所涉及的功能。

[0105] 在此对“根据示例”或“在示例中”的引用意指结合示例描述的特定特征、结构或特性可以包括在本原理的至少一个实现方式中。短语“根据示例”或“在示例中”在说明书中各处的出现不一定都指代相同的示例,单独的或替选的示例也不一定与其他示例相互排除。

[0106] 权利要求中出现的参考标号仅仅通过说明的方式,并且应对权利要求的范围没有限制作用。

[0107] 虽然没有明确描述,但是可以以任何组合或子组合来利用本示例和变型。

[0108] 本原理公开修改版本的CABAC,其中用于在上下文集合之中选择上下文的策略取决于要编码的图片块的大小。

[0109] 图9示出了图示根据本原理的示例的根据图片块的大小针对二进制符号在若干个上下文的集合之中选择上下文的块图(步骤110)。

[0110] 在步骤900中,模块检查图片块的大小(宽度和高度)是否是2的幂。

[0111] 如果是,则在步骤910中,当所述图片块的大小是2的幂时,选择器 SL1根据所述图片块的大小来确定上下文。

[0112] 如果不是,则在步骤920中,选择器SL2根据紧接地高于或者(替选地) 低于所述图片块的大小的块大小来确定上下文,所述块大小是2的幂。

[0113] 可以应用这些本原理以对任何种类的图片元素或与具有非均匀统计行为的图片数据有关的语法元素进行编码。

[0114] 在下文中,这些本原理的实施例关于HEVC标准 (ITU-T H.265ITU电信标准化部门 (10/2014),系列H:视听和多媒体系统,视听服务基础架构- 移动视频编码,高效视频编码,ITU-T H.265建议书) 被描述,但是可以扩展到需要用于编码图片数据或语法元素的熵编码的任何其他视频编码/解码方案。

[0115] 在HEVC标准和与“Algorithm Description of Joint Exploration Test Model 3”(文档JVET-C1001\_v3,ISO/IEC JTC1/SC29/WG11联合视频探索小组,第3次会议,2015年5月26日-6月1日,日内瓦,CH) 相关联的联合探索模型中,属于称为变换块(TB)的图片块的经量化变换系数通常使用CABAC 进行熵编码。TB是图片块,其可以是例如如上所述获得的编码树的编码单元(叶片)。

[0116] 首先,变换块TB被分裂成经量化变换系数的子块,称为编码组(CG)。熵编码/解码由若干个扫描通道构成,其根据在若干个可能的扫描图案之中选择的扫描图案来扫描TB。

[0117] 作为图10上所示的示例,8×8TB被分裂成四个4×4编码组(CG)。对于帧间预测的编码单元,始终使用对角线扫描图案,而对于4×4和8×8 帧内预测的编码单元,所选扫描图案取决于对该CU起作用的帧内预测模式。

[0118] 然后,TB上的扫描通道包括根据三个扫描顺序(对角线、水平、垂直) 之一按顺序处理每个CG,并且也根据所考虑的扫描顺序来扫描每个CG内的16个经量化变换系数。扫描通道在TB中的最后有效(非零) 系数开始,并处理所有经量化变换系数,直到位于CG左上角的经量化变换系数(通常携带CG的大部分能量)。

[0119] 因此,将称为“最后有效系数x坐标”和“最后有效系数y坐标”的两个语法元素添加到用于每个TB的HEVC兼容编码器的输出比特流,以根据与所述TB相关联的扫描顺序向解码器提供TB中的最后非零的经量化变换系数的空间位置(x和y坐标)。

[0120] 经量化变换系数的熵编码可以包括多达五个扫描通道,分别专用于以下语法元素中的每一个的编码:

[0121] • significant-coeff-flag:其为二进制标志,指示经量化变换系数是否等于零;

[0122] • coeff-abs-level-greater1-flag:其为二进制标志,指示经量化变换系数级别的绝对值是否大于1;

[0123] • coeff-abs-level-greater2-flag:其为二进制经量化变换系数,指示系数级别的绝对值是否大于2;

[0124] • coeff-sign-flag:其为二进制标志,指示有效(非零) 经量化变换系数的符号;

[0125] • coeff-abs-level-remaining:该语法元素表示经量化变换系数级别的绝对值的剩余值(在经量化变换系数值大于在先前通道中编码的值的值的情况下)。

[0126] 一旦通过解码以上四个第一通道的子集而知道系数的绝对值,则不关于其绝对值在该系数的剩余通道中对另外的语法元素进行编码。对于符号编码是一样的:仅针对非零系数发送coeff-sign-flag。

[0127] 对于给定的CG,在进入下个CG之前,对所有扫描通道进行编码,直到可以重建该CG

中的所有经量化变换系数。

[0128] 从比特流整体解码TB包括解析所述比特流,向解码器提供以下语法元素:

[0129] 1. “最后有效坐标”语法元素。

[0130] 2. 对于从TB中包含最后有效系数的CG到TB中的左上CG的每个连续的CG:

[0131] a. CG有效性标志,其在HEVC标准中称为coded\_sub\_block\_flag;

[0132] b. 所考虑的CG中的每个经量化变换系数的有效系数标志。其对应于 HEVC标准中的语法元素sig\_coeff\_flag。

[0133] c. 对于所考虑的CG中的非零经量化变换系数:

[0134] i. coeff\_abs\_level\_greater1\_flag:该标志指示当前系数的绝对值是否高于1。如果不是,绝对值是否等于1;

[0135] ii. coeff\_abs\_level\_greater2\_flag:该标志指示当前系数的绝对值是否高于1。如果不是,绝对值是否等于1;

[0136] iii. coeff\_sign\_flag:其指示非零系数的符号;

[0137] iv. coeff\_bas\_level\_remaining:其对绝对值中高于2的系数的绝对值进行编码。

[0138] 本原理的以下实施例仅关注“最后有效系数坐标”语法元素的编码/解码,以及“有效系数标志”语法元素的编码/解码,因为在与Algorithm Description of Joint Exploration Test Model 3”(文档JVET-C1001\_v3,ISO/IEC JTC1/SC29/WG11联合视频探索小组,第3次会议,2015年5月26日-6月 1日,日内瓦,CH)相关联的联合探索模型中,用于对它们进行编码/解码的上下文取决于TB的大小。

[0139] 根据本原理的第一实施例,将图9的方法应用于对与编码组CG有关的“最后有效系数坐标”语法元素进行基于上下文的算术编码/解码。

[0140] “最后有效系数x坐标”语法元素与属于TB的CG有关并且无损地通过以下来表示:

[0141] -区间索引(前缀),其识别(N+1)个区间的集合之中的包含要编码的x坐标的整数值的区间,其中 $N = 2 \times \log_2(T) - 1$ ,其中T是变换的大小,也就是说,在正方形块的情况下的宽度或高度,用于获得CG中的变换系数;以及

[0142] -偏移(后缀),其确定由所述区间索引以信号通知的区间内的最后有效系数的x坐标。

[0143] 该方法获得所述区间索引,以便x坐标值属于由所述区间索引识别的值的区间,并且获得所识别区间内的所述偏移,以便“最后有效系数x坐标”值无损地通过所述前缀和后缀来表示。

[0144] 后缀是有二进制值的语法元素,其通常以旁路模式进行编码,例如通过使用b个位的固定长度代码, $b = \max(0, \lfloor \text{prefix}/2 \rfloor - 1)$ ,如图11所示。将编码后缀添加到输出比特流。

[0145] 表示区间索引的前缀是有非二进制值的语法元素,其根据本原理的示例进行基于上下文的算术编码(步骤110,120)。然后,将前缀添加到 out-bit-stream。

[0146] 为了编码“最后有效系数x坐标”(“最后有效系数y坐标”)语法元素,首先,在步骤100(图1)中,二值化器将区间索引映射到截断一元代码,如图11所示(当T=32时)。对于T=4、8、16和32,一个坐标的截断一元代码的最大长度(也是基于上下文的编码二进制数的数量)分别为3、5、7 和9。实际上,例如,如果T等于4,则最后有效系数x坐标可以等于0、1、2或

3。这些值分别通过截断一元代码‘0’、‘10’、‘110’和‘1110’来识别。因此,对于T等于4,需要多达3个二进制数。此外,对于T等于8,则最大截断码字为‘111110’。由于最后有效系数x坐标最多为7,于是一旦解码了5个连续的等于‘1’的二进制数,解码器就能够推断出前缀代码为‘111110’。这就是为何在变换大小为8的情况下前缀需要最多5个二进制数。

[0147] 接下来,在步骤900(图9)中,检查TB大小是否是2的幂。根据等于2的幂的不同TB大小来获得至少一个上下文集合,每个上下文集合是针对等于2的特定幂的TB大小获得的。

[0148] 如果是,则在步骤910中,选择器SL1根据TB大小,针对截断一元代码的每个二进制数,在所述至少一个获得的上下文集合之中选择上下文集合。

[0149] 如果否,则在步骤920中,选择SL2SL1根据紧接地高于或者(替选地)低于TB大小的块大小,针对截断一元代码的每个二进制数,在所述至少一个获得的上下文集合之中选择上下文集合,并且所述块大小等于2的幂。

[0150] 最后,考虑根据截断一元代码中的二进制数的索引在所述所选上下文集合之中确定的上下文,对表示“最后有效系数x坐标”语法元素的截断一元代码的每个二进制数进行算术编码(步骤120)。

[0151] 图12示出了根据不同TB大小给出上下文索引集合的表格的示例。注意,图12上的表格的“二进制数索引”识别截断一元代码中要编码的二进制数的索引。

[0152] 根据该示例,当TB大小是2的幂、即 $T=2,4,8,16$ 或32时,根据T获得上下文集合。例如,当 $T=8$ 时,选择上下文索引集合(3,3,3,4,4,5)。最终确定的上下文索引取决于截断一元代码中当前二进制数的索引。例如,如果截断一元代码中秩为2的二进制数是当前二进制数,则考虑上下文索引4。

[0153] 当TB大小不是2的幂、即 $T=3,6,12$ 或24时,根据紧接地低于TB大小的块大小来选择上下文集合。

[0154] 例如,当TB大小等于12时,选择与等于8的TB大小有关的上下文索引集合。所确定的上下文索引取决于截断一元代码的当前二进制数的索引。例如,如果截断一元代码中秩为2的二进制数是当前二进制数,则考虑上下文索引4。

[0155] 图13示出了根据不同TB大小给出上下文索引集合的表格的示例。注意,图13上的表格的“二进制数索引”识别截断一元代码中要编码的二进制数的索引。

[0156] 根据该示例,当TB大小是2的幂、即 $T=2,4,8,16$ 或32时,根据TB大小选择上下文集合。例如,当 $T=8$ 时,选择上下文索引集合(3,3,3,4,4,5)。所确定的上下文索引取决于截断一元代码中当前二进制数的索引。例如,如果截断一元代码的秩为2的二进制数是当前二进制数,则考虑上下文索引4。

[0157] 当TB大小不是2的幂、即 $T=3,6,12$ 或24时,根据紧接地高于TB大小的块大小来选择上下文集合。

[0158] 例如,当TB大小等于12时,选择与等于16的TB大小有关的上下文索引集合。所确定的上下文索引取决于截断一元代码的当前二进制数的索引。例如,如果截断一元代码的秩为2的二进制数是当前二进制数,则考虑上下文索引7。

[0159] 为了对与“最后有效系数x坐标”有关的前缀进行编码,本原理重新使用与作为2的幂的、紧接地高于或低于等于 $m \cdot 2^n$ (不同于2的幂)的当前TB大小的TB大小相关联的上下文。

[0160] 在数学上讲,对于等于 $3 \cdot 2^n$  ( $m=3$ )的TB大小(宽度或高度),使用与TB大小 $2^{n+1}$ (图12)或 $2^{n+2}$ (图13)有关的上下文集合。

[0161] 因此,本原理重新使用已经用于大小等于2的幂的图片块的上下文(如在HEVC标准中或者在与HEVC标准相关联的联合探索模型中定义的)而不是为大小不等于2的幂的图片块定义新的上下文。这避免了要存储和处理的上下文的数量增加,从而避免了大小不等于2的幂的变换块(TB)的编码/解码的复杂性的显著增加。

[0162] 根据实施例,可以如下根据TB大小来确定(步骤910或920)对与“最后有效系数x坐标”语法元素有关的前缀进行编码的截断一元代码的二进制数的上下文。所述上下文在上下文集合之中通过上下文索引currCtxIdx来识别,该上下文集合本身在至少一个先前确定的上下文集合之中被识别。

[0163] 首先,针对“最后有效系数x坐标”计算基本上下文索引baseCtxIdx,以根据TB的宽度对前缀进行编码。

[0164] 该基本上下文索引在所述至少一个先前确定的上下文集合之中识别上下文集合。

[0165] 对于图片的色度分量,基本索引可以系统地设置为零,并且对于亮度分量,可以将基本索引计算为TB的宽度w的函数g\_uiLastPrefixCtx:

$$[0166] \quad \text{baseCtxIdx} = \text{Chroma? } 0: g\_uiLastPrefixCtx[\log_2(w \% m == 0? w * \frac{4}{m} : w)]$$

[0167] 接下来,可以如下计算上下文偏移值ctxIdx\_shiftX:

$$[0168] \quad \text{CtxIdx}_{\text{shiftX}} = \text{is chroma? Clip3}(0, 2, (>> 3)): \left( \left( \log_2\left(\frac{w}{m} == 0? w * \frac{4}{m} : w\right) + 1 \right) \gg 2 \right)$$

[0169] 备选地,

$$[0170] \quad \begin{aligned} \text{baseCtxIdx} &= \text{Chroma? } 0: g\_uiLastPrefixCtx[\log_2(w \% m == 0? w * \frac{2}{m} : w)] \\ \text{CtxIdx}_{\text{shiftX}} &= \text{is chroma? Clip3}(0, 2, (>> 3)): \left( \left( \log_2\left(\frac{w}{m} == 0? w * \frac{2}{m} : w\right) + 1 \right) \gg 2 \right) \end{aligned}$$

[0171] 接下来,随着根据截断一元代码中要编码的当前二进制数的索引 prefixBinIdx确定基本上下文索引baseCtxIdx和上下文偏移值 ctxIdx\_shiftX的函数,对于前缀的二进制数确定上下文索引currCtxIdx:

$$[0172] \quad \text{currCtxIdx} = \text{baseCtxIdx} + (\text{prefixBinIdx} \gg \text{ctxIdx\_shiftX}) \quad (1)$$

[0173] 注意,前缀的第一二进制数的上下文索引可以等于基本上下文索引 baseCtxIdx。

[0174] 根据图6的实施例,如下从输入比特流解码“最后有效系数x坐标”(“最后有效系数y坐标”)语法元素。

[0175] 通过等式(1)如下计算上下文索引currCtxIdx。一旦在步骤110获得基本上下文索引baseCtxIdx,解码器就从输入比特流生成一系列解码二进制数。针对要解码的一系列二进制数的每个二进制数,将上下文索引 currCtxIdx计算为基本上下文索引与上下文增量值((prefixBinIdx >> ctxIdx\_shiftX))的总和,该上下文增量值在编码器侧以相同方式严格获得(等式1)。如此获得的上下文索引currCtxIdx对应于由上下文建模步骤110提供的上下文模型的索引。然后,借助于具有所确定标识符的上下文,对要解码的一系列二进制数的当前二进制数值(1或0)进行算术解码(图6的步骤620)。

[0176] 然后,一旦要解码的一系列二进制数的‘0’二进制数值被解码或者‘1’二进制数的

最大数量达到所考虑的TB大小,则这意味着一系列解码二进制数对应于图11的截断一元二值化前缀。然后,解码器知道要旁路解码的后缀二进制数的数量 $b$ ,其提供suffix,该suffix确定由所述解码前缀以信号通知的区间内的最后有效系数的 $x$ 坐标。

[0177] 最后,如下基于解码的prefix和suffix值来重建最后有效系数 $x$ 坐标,在以下等式中标为last:

[0178]  $last = 2^b (2 + (prefix \% 2)) + suffix$ , 如果  $prefix > 3$

[0179]  $last = prefix$  否则

[0180] 可以以类似的方式根据TB大小在上下文集合之中选择“最后有效系数  $y$  坐标”语法元素的上下文索引。

[0181] 根据本原理的第二实施例,将图9的方法应用于对与属于TB的编码组 CG有关的“有效系数标志”语法元素进行基于上下文的算术编码/解码。

[0182] “有效系数标志”是与CG的经量化变换系数有关的有二进制值的语法元素。

[0183] 首先,在步骤110如下计算基本上下文偏移ctxOffset:

[0184]  $ctxOffset = isChroma ? 0 : indexOnTbSize$  (2)

[0185] 其中indexOnTbSize是如下计算的区间索引:

[0186]  $indexOnTbSize = uiLog2BlockSize \leq 2 ? 0 : f(uiLog2BlockSize)$  (3)

[0187] 其中uiLog2BlockSize是根据TB的大小(区域)计算的值。

[0188] 其次,从最后系数朝向最低频率系数来考虑CG中的每个经量化变换系数。对于每个经量化变换系数,通过将上下文增量ctxIncrement添加到基本上下文偏移ctxOffset来计算用于对其有效性进行编码的上下文索引 currCtxIdx,该基本上下文偏移确定由所述区间索引indexOnTbSize以信号通知的整数值的区间内的最后有效系数的 $x$ 坐标。

[0189] 最后,考虑与上下文索引相关联的上下文,对“有效系数标志”进行算术编码(步骤120)

[0190]  $currCtxIdx = ctxOffset + ctxIncrement$ 。 (4)

[0191] 根据实施例,如下计算上下文增量ctxIncrement,如图14上所示。

[0192] 上下文增量ctxIncrement取决于由局部模板覆盖的邻域中的先前编码的经量化变换系数的“有效系数标志”值的值。更具体地,基于相邻系数的有效标志的总和来计算上下文增量ctxIncrement。图14图示了这样的模板的示例。根据该示例,如下计算与当前经量化变换系数(由白圈表示)相关联的上下文增量:

[0193]  $ctxIncrement = sigCoeffFlag(x+1,y) + sigCoeffFlag(x+2,y) +$

[0194]  $sigCoeffFlag(x,y+1) + sigCoeffFlag(x,y+2) + sigCoeffFlag(x+1,y+1)$  (5)

[0195] 其中 $x$ 和 $y$ 是所考虑的TB中的当前系数的坐标,并且如果位置 $(x,y)$ 处的经量化变换系数是非零,则 $sigCoeffFlag(x,y)$ 等于1,否则等于0。注意, $x_0$ 、 $x_3$ 、 $x_1$ 、 $x_4$ 和 $x_2$ 分别是具有坐标 $(x+1,y)$ 、 $(x+2,y)$ 、 $(x,y+1)$ 、 $(x,y+2)$ 和 $(x+1,y+1)$ 的相邻系数。

[0196] 为了以不同频率捕获经量化变换系数的特性,可以将TB分裂成多达三个区,并且分裂方法是固定的而不管TB大小,如图14上所示。将上下文索引分配给每个区。

[0197] 根据本原理的第二实施例,在步骤900(图9)中,检查TB的宽度 $w$ 和/或高度 $h$ 是否是2的幂。

[0198] 如果是,则在步骤910中,选择器SL1根据基本上下文索引 indexOnTbSize来确定

上下文,该基本上下文索引本身根据TB大小来确定。

[0199] 如果否,则在步骤920中,选择器SL2根据基本上下文索引  $\text{indexOnTbSize}$  来确定上下文,该基本上下文索引本身根据紧接地高于或者 (替代地) 低于TB大小的块大小来确定,所述块大小等于2的幂。

[0200] 根据实施例,通过以下给出基本上下文索引  $\text{indexOnTbSize}$ :

$$\begin{aligned} \text{uiLog2BlkSize} &= (\log_2(w \% m == 0 ? w * \frac{2^{\lfloor \log_2(m) \rfloor}}{m} : w) + \log_2(h \% m == \\ [0201] &= 0 ? h * \frac{2^{\lfloor \log_2(m) \rfloor}}{m} : h)) >> 1 \end{aligned}$$

[0202] 其中w和h分别是TB的宽度和高度,f(.)是数学函数。

[0203] 根据变型,

$$\begin{aligned} \text{uiLog2BlkSize} &= (\log_2(w \% m == 0 ? w * \frac{2^{\lfloor \log_2(m) \rfloor + 1}}{m} : w) + \log_2(h \% m == \\ [0204] &= 0 ? h * \frac{2^{\lfloor \log_2(m) \rfloor + 1}}{m} : h)) >> 1 \end{aligned}$$

[0205] 基本上下文索引  $\text{indexOnTbSize}$  识别与具体TB区域有关的具体上下文集合。这意味着可以采用单独的上下文集合对具有不同区域的TB的“有效系数标志”进行编码。

[0206] 上下文偏移  $\text{ctxOffset}$  可以对于亮度分量等于所述基本上下文索引,并且对于色度分量等于0。

[0207] 根据图6的实施例,如下从输入流解码“最后有效标志”语法元素。

[0208] 首先,从上下文建模步骤110) 获得与至少“有效系数标志”相关联的上下文索引,其与编码器的上下文建模步骤相同。

[0209] 因此,根据等式 (4) 和等式 (5),将上下文索引计算为基本上下文偏移  $\text{ctxOffset}$  和上下文增量  $\text{ctxIndexIncrement}$  (各自与编码器侧完全相同地被计算) 的总和。

[0210] 接下来,给定具有所述计算的上下文索引的上下文模型和输入比特流的位,算术解码步骤620能够对二进制数值进行解码,该二进制数值表示与当前正被解码的CG的当前系数相关联的“有效系数标志”。

[0211] 在图1-图14上,模块是功能单元,其可以与可区分的物理单元有关或无关。例如,这些模块或它们中的一些可以在唯一的组件或电路中集合在一起,或者有助于软件的功能。相反,一些模块可以可能地由单独的物理实体组成。与本原理兼容的装置使用纯硬件实现,例如使用专用硬件,诸如ASIC 或FPGA或VLSI (分别为“专用集成电路”、“现场可编程门阵列”、“超大规模集成”),或者来自嵌入在设备中的多个集成电子组件或来自硬件和软件组件的混合。

[0212] 图15表示设备1500的示例性架构,其可以被配置为实现关于图1-图 14描述的方法。

[0213] 设备1500包括通过数据和地址总线1501链接在一起的以下元件:

[0214] -微处理器1502 (或CPU),其为例如DSP (或数字信号处理器);

[0215] -ROM (或只读存储器) 1503;

[0216] -RAM (或随机存取存储器) 1504;

[0217] -I/O接口1505,用于从应用接收要传送的数据;以及

[0218] -电池1506。

[0219] 根据示例,电池1506在设备的外部。在每个提及的存储器中,说明书中使用的词语“寄存器”可以对应于小容量的区域(一些位)或非常大的区域(例如整个程序或大量的接收或解码的数据)。ROM 1503至少包括程序和参数。ROM 1503可以存储算法和指令以执行根据本原理的技术。当接通时,CPU 1502上传RAM中的程序并执行对应的指令。

[0220] RAM 1504在寄存器中包括由CPU 1502执行并在设备1500接通之后上传的程序,在寄存器中包括输入数据,在寄存器中包括方法的不同状态下的中间数据,以及在寄存器中包括用于执行该方法的其他变量。

[0221] 在此描述的实现方式可以例如以方法或处理、装置、软件程序、数据流或信号来实现。即使仅在单一形式的实现方式的背景下进行了讨论(例如,仅作为方法或设备进行了讨论),但是所讨论的特征的实现方式也可以以其他形式(例如程序)来实现。装置可以在例如适当的硬件、软件和固件中实现。方法例如可以在诸如例如处理器(其一般涉及处理设备,包括例如计算机、微处理器、集成电路或可编程逻辑器件)之类的装置中实现。处理器还包括诸如例如计算机、蜂窝电话、便携/个人数字助理(“PDA”)之类的通信设备,以及促成在最终用户之间进行信息通信的其他设备。

[0222] 根据编码或编码器的示例,从源获得视频数据的块或语法元素或二进制符号序列。例如,源属于包含以下的集合:

[0223] -本地存储器(1503或1504),例如视频存储器或RAM(或随机存取存储器)、闪速存储器、ROM(或只读存储器)、硬盘;

[0224] -储存接口(1505),例如与大容量储存器、RAM、闪速存储器、ROM、光盘或磁性支撑的接口;

[0225] -通信接口(1505),例如有线接口(例如总线接口、广域网接口、局域网接口)或无线接口(例如IEEE 802.11接口或蓝牙®接口);以及

[0226] -图片捕获电路(例如传感器,诸如例如CCD(或电荷耦合器件)或CMOS(或互补金属氧化物半导体))。

[0227] 根据解码或解码器的示例,将解码数据发送到目的地;具体地,目的地属于包括以下的集合:

[0228] -本地存储器(1503或1504),例如视频存储器或RAM,闪速存储器、硬盘;

[0229] -储存接口(1505),例如与大容量储存器、RAM、闪速存储器、ROM、光盘或磁性支撑的接口;

[0230] -通信接口(1505),例如有线接口(例如总线接口(例如USB(或通用串行总线))、广域网接口、局域网接口、HDMI(高清多媒体接口)接口)或无线接口(例如IEEE 802.11接口、WiFi®或蓝牙®接口);以及

[0231] -显示器。

[0232] 根据编码或编码器的示例,将比特流发送到目的地。作为示例,将比特流存储在本地或远程存储器中,例如视频存储器(1504)或RAM(1504)、硬盘(1503)。在变型中,将比特流发送到储存接口(1505)(例如与大容量储存器、闪速存储器、ROM、光盘或磁性支撑的接口),和/或将比特流通过通信接口(1505)(例如到点对点链路、通信总线、点对多点链路或广播

网络的接口) 传送。

[0233] 根据解码或解码器的示例,从源获得比特流。示例性地,从本地存储器(例如视频存储器(1504)、RAM(1504)、ROM(1503)、闪速存储器(1503)或硬盘(1503))读取比特流。在变型中,从储存接口(1505)(例如与大容量储存器、RAM、ROM、闪速存储器、光盘或磁性支撑的接口)接收比特流,和/或从通信接口(1505)(例如到点对点链路、总线、点对多点链路或广播网络的接口)接收比特流。

[0234] 根据示例,被配置为实现关于图1-图14描述的方法的设备1500属于包括以下的集合:

- [0235] -移动设备;
- [0236] -通信设备;
- [0237] -游戏设备;
- [0238] -平板(或平板计算机);
- [0239] -膝上型机;
- [0240] -静止图片相机;
- [0241] -摄像机;
- [0242] -编码芯片;
- [0243] -静止图片服务器;
- [0244] -视频服务器(例如广播服务器、视频点播服务器或网络服务器);
- [0245] -机顶盒;
- [0246] -电视机;
- [0247] -显示器,以及
- [0248] -解码芯片。

[0249] 在此描述的各种处理和特征的实现方式可以以各种不同的装备或应用来实施。这样的装备的示例包括编码器、解码器、处理来自解码器的输出的后处理器、向编码器提供输入的预处理器、视频编码器、视频解码器、视频编解码器、网络服务器、机顶盒、膝上型机、个人计算机、蜂窝电话、PDA 以及用于处理图片或视频的任何其他设备或者其他通信设备。如应当清楚的那样,该装备可以是移动的,并且甚至安装在移动交通工具中。

[0250] 另外,该方法可以由处理器正执行的指令来实现,并且这样的指令(和/或由实现方式产生的数据值)可以存储在计算机可读储存介质上。计算机可读储存介质可以采取计算机可读程序产品的形式,该计算机可读程序产品包含在一个或多个计算机可读介质中并且具有包含在其上的、可由计算机执行的计算机可读程序代码。在此使用的计算机可读储存介质被视为非临时性储存介质,给定在其中存储信息的固有能力以及提供从中取回信息的固有能力。计算机可读储存介质可以是例如但不限于电子、磁、光、电磁、红外或半导体系统、装置或设备,或者前述的任何合适组合。要理解,以下虽然提供可以应用本原理的计算机可读储存介质的更特定的示例,但仅仅是本领域普通技术人员容易理解的说明性而非详尽的列表:便携式计算机磁盘;硬盘;只读存储器(ROM);可擦除可编程只读存储器(EPROM或闪速存储器);便携式压缩盘只读存储器(CD-ROM);光储存设备;磁储存设备;或者前述的任何合适组合。

[0251] 指令可以形成有形地包含在处理器可读介质上的应用程序。

[0252] 指令可以例如在硬件、固件、软件或组合中。指令可以存在于例如操作系统、单独的应用或两者的组合中。因此,处理器可以表征为例如被配置为执行处理的设备和包括具有用于执行处理的指令的处理器可读介质(诸如储存设备)的设备二者。此外,除了指令或者代替指令,处理器可读介质可以存储由实现方式产生的数据值。

[0253] 如对于本领域技术人员将明显的那样,实现方式可以产生被格式化为携带例如可以被存储或传送的信息的各种信号。信息可以包括例如用于执行方法的指令或者由所描述的实现方式之一产生的数据。例如,可以将信号格式化为携带用于写入或读取本原理的所描述示例的语法的规则作为数据,或者携带由本原理的所描述示例写入的实际语法值作为数据。例如,这样的信号可以被格式化为电磁波(例如使用频谱的射频部分)或者基带信号。格式化可以包括例如对数据流进行编码以及以经编码的数据流调制载波。例如,信号携带的信息可以是模拟或数字信息。如已知的那样,可以通过各种不同的有线或无线链路来传送信号。信号可以存储在处理器可读介质上。

[0254] 已经描述了多个实现方式。然而,将理解,可以进行各种修改。例如,可以对不同实现方式的元素进行组合、补充、修改或移除,以产生其他实现方式。另外,普通技术人员将理解,其他结构和处理可以替代所公开的那些结构和处理,并且所得到的实现方式将以与所公开的实现方式至少基本相同的方式来执行与所公开的实现方式至少基本相同的功能,以实现与所公开的实现方式至少基本相同的结果。因此,本申请想到这些以及其他实现方式。

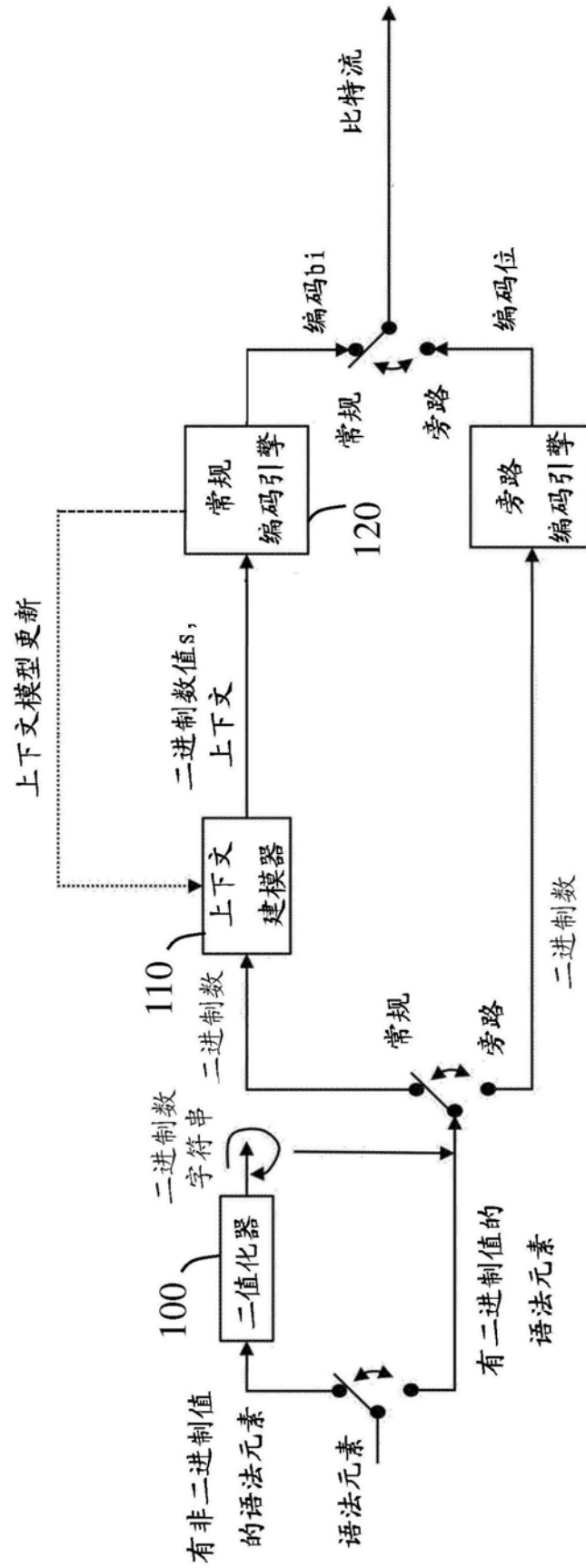


图1

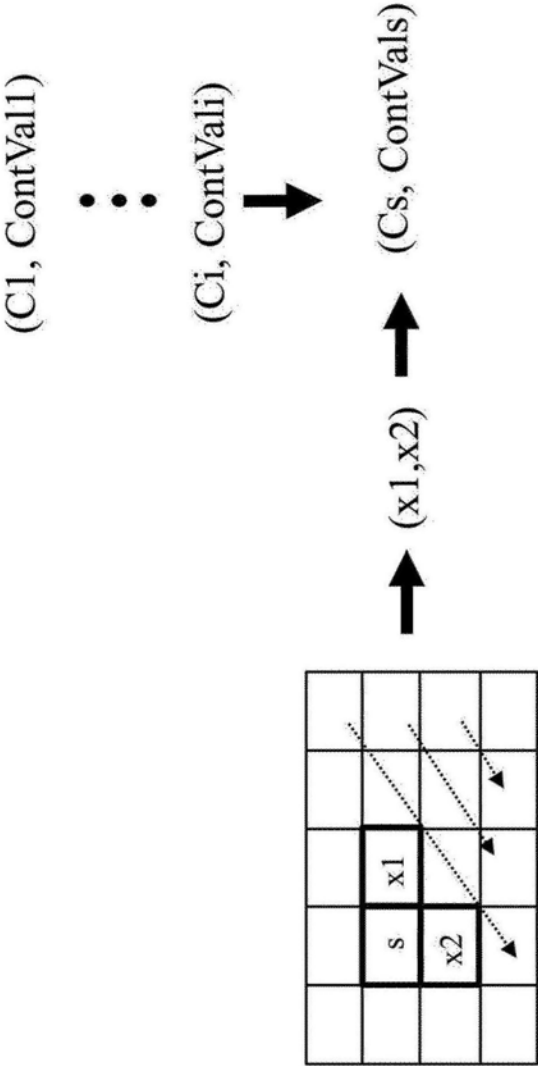


图2

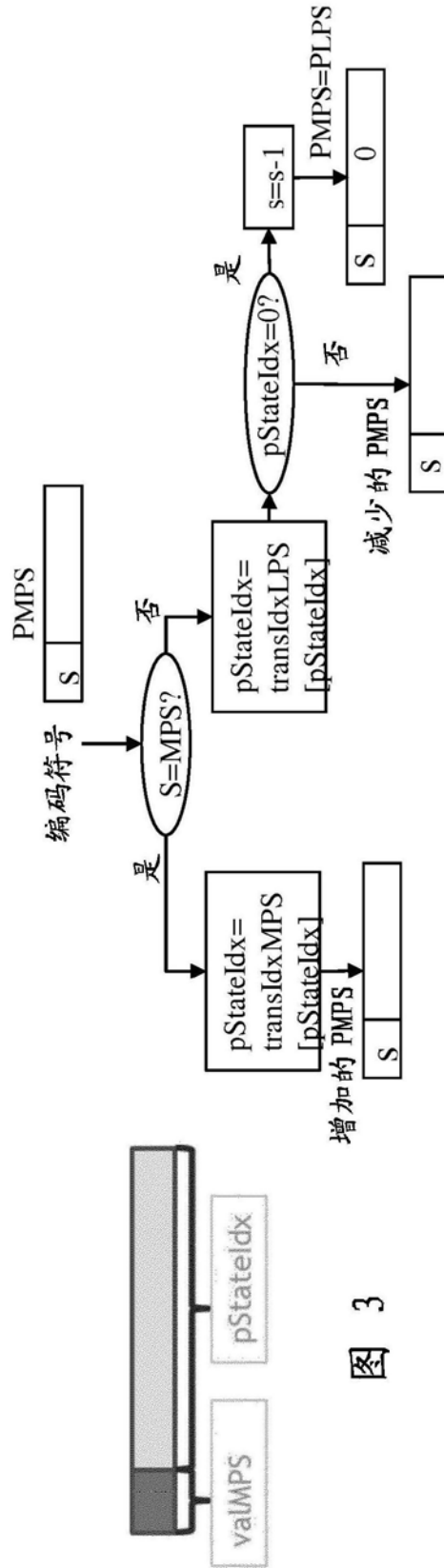


图 3

图 4

|             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| pStateIdx   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| transIdxLps | 0  | 0  | 1  | 2  | 2  | 4  | 4  | 5  | 6  | 7  | 8  | 9  | 9  | 11 | 11 | 12 |
| transIdxMps | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| pStateIdx   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| transIdxLps | 13 | 13 | 15 | 15 | 16 | 16 | 18 | 18 | 19 | 19 | 21 | 21 | 22 | 22 | 23 | 24 |
| transIdxMps | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| pStateIdx   | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| transIdxLps | 24 | 25 | 26 | 26 | 27 | 27 | 28 | 29 | 29 | 30 | 30 | 30 | 31 | 32 | 32 | 33 |
| transIdxMps | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| pStateIdx   | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| transIdxLps | 33 | 33 | 34 | 34 | 35 | 35 | 35 | 36 | 36 | 36 | 37 | 37 | 37 | 38 | 38 | 63 |
| transIdxMps | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 62 | 63 |

图5

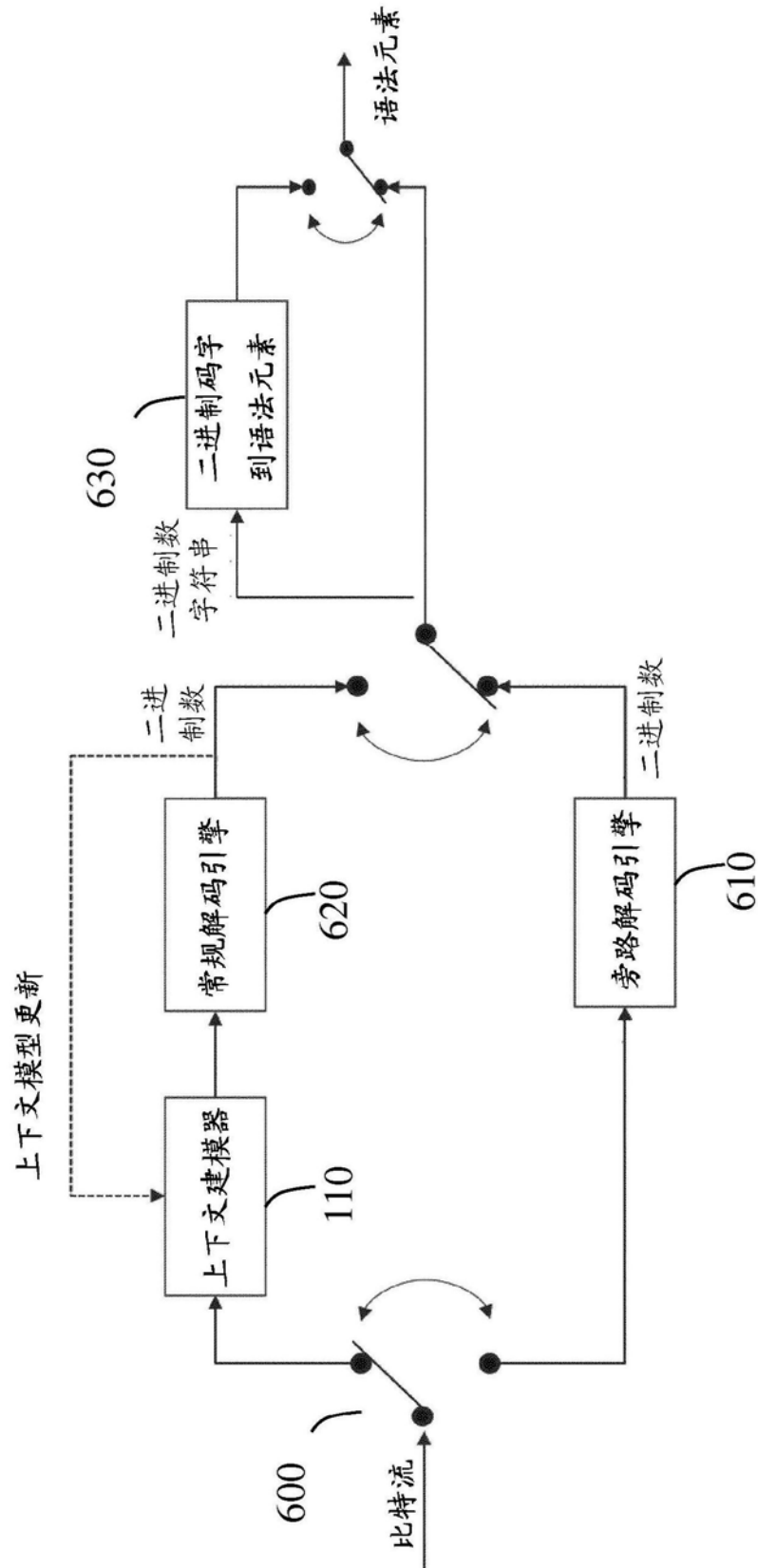


图6

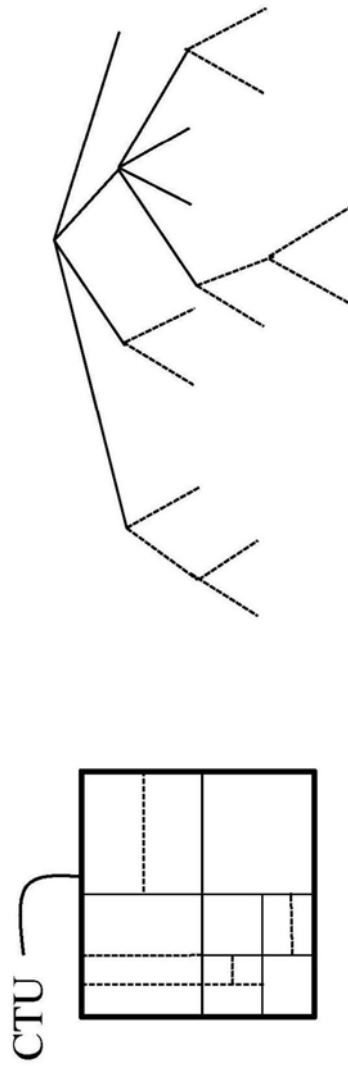


图7

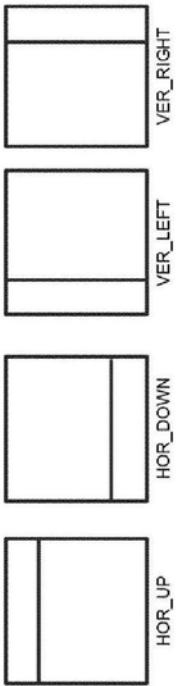


图8

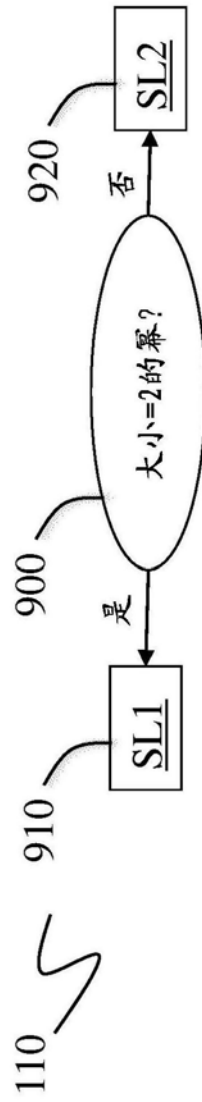


图9

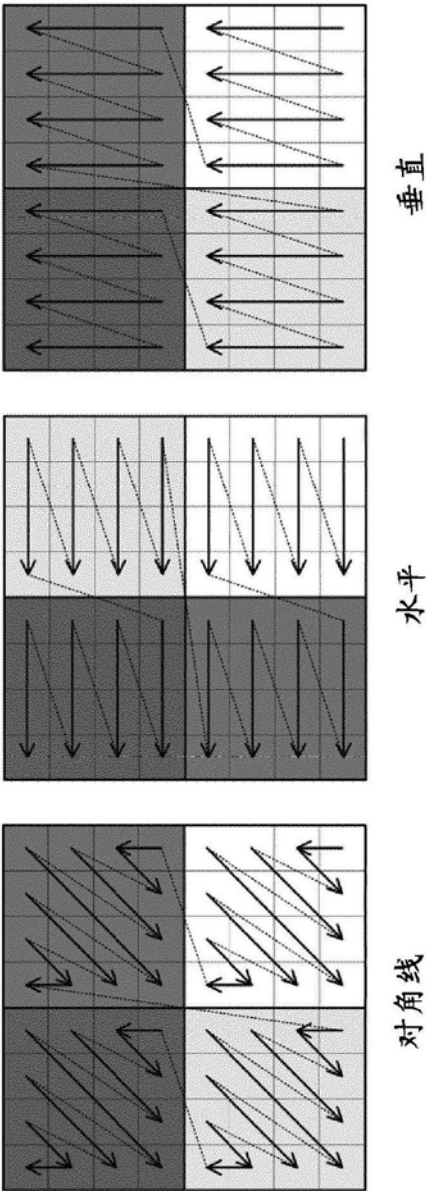


图10

| 位置    | prefix         | suffix         | 后缀范围   |
|-------|----------------|----------------|--------|
|       | 截断一元<br>(常规模式) | 固定长度<br>(旁路模式) |        |
| 0     | 0              | -              | -      |
| 1     | 10             | -              | -      |
| 2     | 110            | -              | -      |
| 3     | 1110           | -              | -      |
| 4-5   | 11110          | X              | 0 to 1 |
| 6-7   | 111110         | X              | 0 to 1 |
| 8-11  | 1111110        | XX             | 0 to 3 |
| 12-15 | 11111110       | XX             | 0 to 3 |
| 16-23 | 111111110      | XXX            | 0 to 7 |
| 24-31 | 111111111      | XXX            | 0 to 7 |

图11

| 二进制索引 | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
|-------|----|----|----|----|----|----|----|----|----|
| T     | 亮度 |    |    |    |    |    |    |    |    |
| 3     | 0  | 1  |    |    |    |    |    |    |    |
| 4     | 0  | 1  | 2  |    |    |    |    |    |    |
| 6     | 0  | 1  | 2  | 3  |    |    |    |    |    |
| 8     | 3  | 3  | 4  | 4  | 5  |    |    |    |    |
| 12    | 3  | 3  | 4  | 4  | 5  | 5  |    |    |    |
| 16    | 6  | 6  | 7  | 7  | 8  | 8  | 9  |    |    |
| 24    | 6  | 6  | 7  | 7  | 8  | 8  | 9  | 9  |    |
| 32    | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 | 14 |
| T     | 色度 |    |    |    |    |    |    |    |    |
| 3     | 15 | 16 |    |    |    |    |    |    |    |
| 4     | 15 | 16 | 17 |    |    |    |    |    |    |
| 6     | 15 | 16 | 17 | 18 |    |    |    |    |    |
| 8     | 15 | 15 | 16 | 16 | 17 |    |    |    |    |
| 12    | 15 | 15 | 16 | 16 | 17 | 17 |    |    |    |
| 16    | 15 | 15 | 15 | 15 | 16 | 16 | 16 |    |    |
| 24    | 15 | 15 | 15 | 15 | 16 | 16 | 16 | 16 |    |
| 32    | 15 | 15 | 15 | 15 | 16 | 16 | 16 | 16 | 17 |

图12

| 二进制索引 | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
|-------|----|----|----|----|----|----|----|----|----|
| T     | 亮度 |    |    |    |    |    |    |    |    |
| 3     | 0  | 1  |    |    |    |    |    |    |    |
| 4     | 0  | 1  | 2  |    |    |    |    |    |    |
| 6     | 3  | 3  | 4  | 4  |    |    |    |    |    |
| 8     | 3  | 3  | 4  | 4  | 5  |    |    |    |    |
| 12    | 6  | 6  | 7  | 7  | 8  | 8  |    |    |    |
| 16    | 6  | 6  | 7  | 7  | 8  | 8  | 9  |    |    |
| 24    | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 |    |
| 32    | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 | 14 |
| T     | 色度 |    |    |    |    |    |    |    |    |
| 3     | 15 | 16 |    |    |    |    |    |    |    |
| 4     | 15 | 16 | 17 |    |    |    |    |    |    |
| 6     | 15 | 15 | 16 | 16 |    |    |    |    |    |
| 8     | 15 | 15 | 16 | 16 | 17 |    |    |    |    |
| 12    | 15 | 15 | 15 | 15 | 16 | 16 |    |    |    |
| 16    | 15 | 15 | 15 | 15 | 16 | 16 | 16 |    |    |
| 24    | 15 | 15 | 15 | 15 | 16 | 16 | 16 | 16 |    |
| 32    | 15 | 15 | 15 | 15 | 16 | 16 | 16 | 16 | 17 |

图13

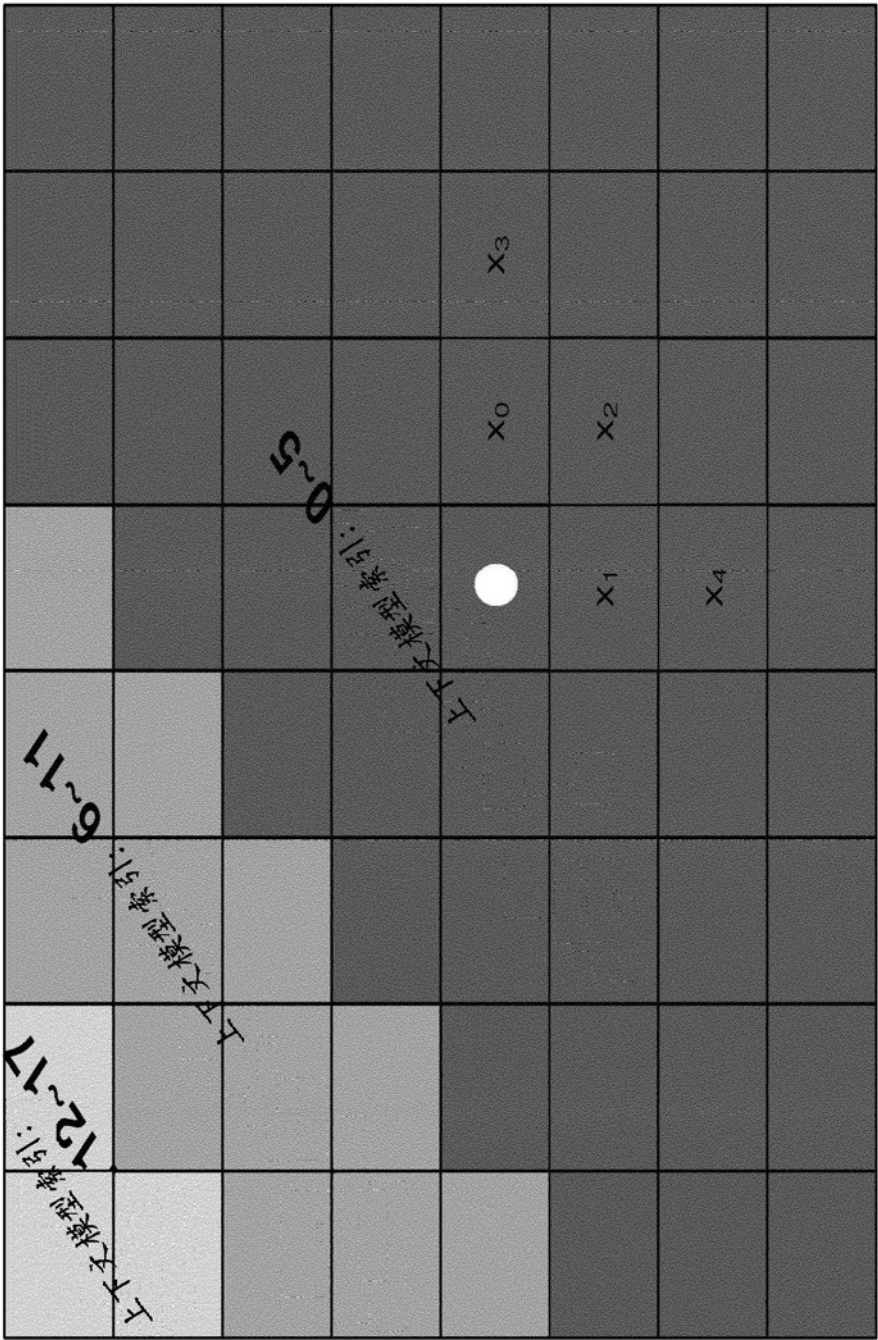


图14

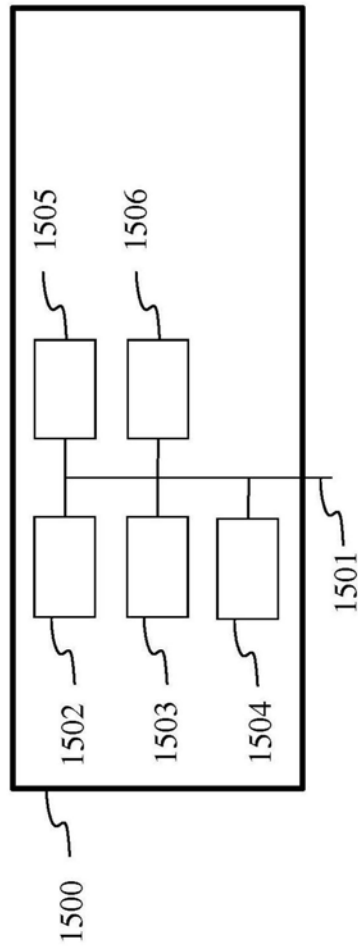


图15