



(12) 发明专利申请

(10) 申请公布号 CN 104365099 A

(43) 申请公布日 2015. 02. 18

(21) 申请号 201380031658. 2

(51) Int. Cl.

(22) 申请日 2013. 04. 15

H04N 19/196(2014. 01)

(30) 优先权数据

H04N 19/13(2014. 01)

61/624, 358 2012. 04. 15 US

H04N 19/96(2014. 01)

H04N 19/463(2014. 01)

(85) PCT国际申请进入国家阶段日

H04N 19/157(2014. 01)

2014. 12. 15

H04N 19/18(2014. 01)

(86) PCT国际申请的申请数据

PCT/KR2013/003151 2013. 04. 15

(87) PCT国际申请的公布数据

W02013/157794 K0 2013. 10. 24

(71) 申请人 三星电子株式会社

地址 韩国京畿道水原市

(72) 发明人 金赞烈 金宰贤 朴正辉

(74) 专利代理机构 北京铭硕知识产权代理有限公司

11286

代理人 苏银虹 曾世骁

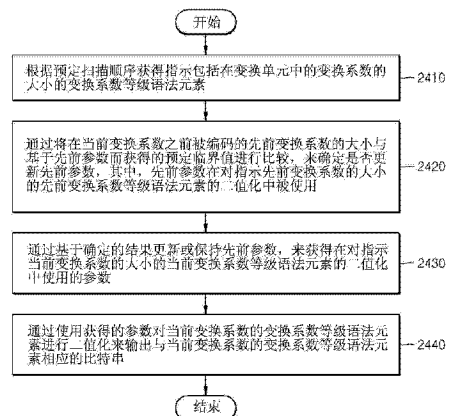
权利要求书3页 说明书26页 附图17页

(54) 发明名称

用于变换系数等级的熵编码和熵解码的参数更新方法以及使用该方法的变换系数等级的熵编码装置和熵解码装置

(57) 摘要

提供了一种更新在对变换系数等级的熵编码和熵解码中使用的参数的方法。通过使用参数对指示变换系数等级的语法元素进行二值化。基于将先前变换系数的大小与基于先前参数而获得的预定临界值进行比较的结果，更新或保持所述参数，其中，先前参数在对先前变换系数等级语法元素的反二值化中被使用。预定临界值被设置为具有与先前参数成比例的值，当先前参数被更新时，更新后的参数与先前参数相比具有逐渐增加的值。



1. 一种更新用于对变换系数等级进行熵解码的参数的方法,所述方法包括:
  - 从比特流解析指示包括在变换单元中的变换系数的大小的变换系数等级语法元素;
  - 通过将当前变换系数之前被恢复的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,确定是否更新先前参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的反二值化中被使用;
  - 通过基于确定的结果更新或保持先前参数,获得在对指示当前变换系数的大小的当前变换系数等级语法元素的反二值化中使用的参数;
  - 通过使用获得的参数对当前变换系数等级语法元素进行反二值化来获得当前变换系数的大小,
  - 其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新后的参数与先前参数相比具有逐渐增加的值。
2. 如权利要求 1 所述的方法,其中,当变换系数的大小是  $absCoeff$ , 指示变换系数是否具有大于 1 的值的语法元素是  $coeff\_abs\_level\_greater1\_flag$ , 并且指示变换系数是否具有大于 2 的值的语法元素是  $coeff\_abs\_level\_greater2\_flag$  时, 并且当根据等式  $baseLevel = 1 + coeff\_abs\_level\_greater1\_flag + coeff\_abs\_level\_greater2\_flag$  确定的基本等级值是  $baseLevel$  时, 变换系数等级语法元素指示  $(abs - baseLevel)$  的大小, 其中,  $(abs - baseLevel)$  是变换系数的大小  $absCoeff$  与确定的基本等级值  $baseLevel$  之间的差。
3. 如权利要求 1 所述的方法, 其中, 在确定是否更新先前参数的步骤中, 当先前变换系数的大小是  $cLastAbsCoeff$  并且先前参数是  $cLastRiceParam$  时, 将基于等式  $th = 3 * (1 \ll cLastRiceParam)$  获得的临界值  $th$  与先前参数  $cLastRiceParam$  进行比较; 当  $cLastAbsCoeff$  大于  $th$  时, 确定更新先前参数; 当  $cLastAbsCoeff$  不大于  $th$  时, 确定保持先前参数。
4. 如权利要求 3 所述的方法, 其中, 在获得参数的步骤中, 当确定更新先前参数时, 通过在预定范围内将先前参数增加 1 来更新先前参数。
5. 如权利要求 4 所述的方法, 其中, 更新后的参数不超过值 4。
6. 如权利要求 1 所述的方法, 其中, 获得当前变换系数的大小的步骤包括:
  - 基于通过使用获得的参数确定的预定参数, 将具有当前变换系数的等级语法元素的比特串分类为前缀比特串和后缀比特串;
  - 通过针对前缀比特串和后缀比特串中的每个比特串应用先前设置的反二值化方法来恢复当前变换系数的变换系数等级语法元素。
7. 如权利要求 6 所述的方法, 其中, 在当前变换系数的大小是  $coeff\_abs\_level\_remaining$ , 获得的参数是  $cRiceParam$  并且所述预定参数是  $cTrMax$  时, 根据等式  $cTrMax = 4 \ll cRiceParam$  确定参数  $cTrMax$ , 并且前缀比特串相应于通过使用截断一元二值化方法对与  $\min(cTrMax, coeff\_abs\_level\_remaining)$  相应的值进行二值化而获得的比特串, 后缀比特串相应于通过使用  $k$  阶指数哥伦布方法对与  $(coeff\_abs\_level\_remaining - cTrMax)$  相应的值进行二值化而获得的比特串, 其中,  $k$  等于  $(cRiceParam + 1)$ 。
8. 一种用于对变换系数等级进行熵解码的设备, 所述设备包括:
  - 解析单元, 从比特流解析指示包括在变换单元中的变换系数的大小的变换系数等级语

法元素；

参数确定单元,通过将在当前变换系数之前被恢复的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,确定是否更新先前参数,并通过基于确定的结果更新或保持先前参数,获得在对指示当前变换系数的大小的当前变换系数等级语法元素的反二值化中使用的参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的反二值化中被使用；

语法元素恢复单元,通过使用获得的参数对当前变换系数等级语法元素进行反二值化来获得当前变换系数的大小,

其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新后的参数与先前参数相比具有逐渐增加的值。

9. 一种更新用于对变换系数等级进行熵编码的参数的方法,所述方法包括：

以预定扫描顺序获得指示包括在变换单元中的变换系数的大小的变换系数等级语法元素；

通过将在当前变换系数之前被编码的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,确定是否更新先前参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的二值化中被使用；

通过基于确定的结果更新或保持先前参数,获得在对指示当前变换系数的大小的当前变换系数等级语法元素的二值化中使用的参数；

通过使用获得的参数对当前变换系数的变换系数等级语法元素进行二值化来输出与当前变换系数的变换系数等级语法元素相应的比特串,

其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新后的参数与先前参数相比具有逐渐增加的值。

10. 如权利要求 9 所述的方法,其中,当变换系数的大小是  $absCoeff$ , 指示变换系数是否具有大于 1 的值的语法元素是  $coeff\_abs\_level\_greater1\_flag$ , 并且指示变换系数是否具有大于 2 的值的语法元素是  $coeff\_abs\_level\_greater2\_flag$  时, 并且当根据等式  $baseLevel = 1 + coeff\_abs\_level\_greater1\_flag + coeff\_abs\_level\_greater2\_flag$  确定的基本等级值是  $baseLevel$  时, 变换系数等级语法元素指示  $(abs - baseLevel)$  的大小, 其中,  $(abs - baseLevel)$  是变换系数的大小  $absCoeff$  与确定的基本等级值  $baseLevel$  之间的差。

11. 如权利要求 9 所述的方法,其中,在确定是否更新先前参数的步骤中,当先前变换系数的大小是  $cLastAbsCoeff$  并且先前参数是  $cLastRiceParam$  时,将基于等式  $th = 3 * (1 << cLastRiceParam)$  获得的临界值  $th$  与先前参数  $cLastRiceParam$  进行比较;当  $cLastAbsCoeff$  大于  $th$  时,确定更新先前参数;当  $cLastAbsCoeff$  不大于  $th$  时,确定保持先前参数。

12. 如权利要求 11 所述的方法,其中,在获得参数的步骤中,当确定更新先前参数时,通过在预定范围内将先前参数增加 1 来更新先前参数。

13. 如权利要求 9 所述的方法,其中,输出比特串的步骤包括：

基于通过使用获得的参数确定的预定参数将当前变换系数等级语法元素分类为前缀和后缀；

通过针对前缀比特串和后缀比特串中的每个比特串应用先前设置的二值化方法来产生前缀比特串和后缀比特串,并输出与当前变换系数的变换系数等级语法元素相应的比特串。

14. 如权利要求 13 所述的方法,其中,在当前变换系数的大小是 `coeff_abs_level_remaining`,获得的参数是 `cRiceParam` 并且所述预定参数是 `cTrMax` 时,根据等式  $cTrMax = 4 \ll cRiceParam$  确定参数 `cTrMax`,并且前缀比特串相应于通过使用截断一元二值化方法对与  $\text{Min}(cTrMax, \text{coeff\_abs\_level\_remaining})$  相应的值进行二值化而获得的比特串,后缀比特串相应于通过使用  $k$  阶指数哥伦布方法对与  $(\text{coeff\_abs\_level\_remaining} - cTrMax)$  相应的值进行二值化而获得的比特串,其中,  $k$  等于  $(cRiceParam + 1)$ 。

15. 一种用于对变换系数等级进行熵编码的设备,所述设备包括:

参数确定单元,以预定扫描顺序获得指示包括在变换单元中的变换系数的大小的变换系数等级语法元素,通过将在当前变换系数之前被编码的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,确定是否更新先前参数,通过基于确定的结果更新或保持先前参数,获得在对指示当前变换系数的大小的当前变换系数等级语法元素的二值化中使用的参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的二值化中被使用;

比特串产生单元,通过使用获得的参数对当前变换系数的变换系数等级语法元素进行二值化来输出与当前变换系数的变换系数等级语法元素相应的比特串,

其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新的参数与先前参数相比具有逐渐增加的值。

## 用于变换系数等级的熵编码和熵解码的参数更新方法以及使用该方法的变换系数等级的熵编码装置和熵解码装置

### 技术领域

[0001] 本申请涉及视频编码和解码,更具体地讲,涉及用于对在变换系数的尺寸信息的熵编码和熵解码中使用的参数进行更新的方法和设备。

### 背景技术

[0002] 根据图像压缩方法(诸如,MPEG-1、MPEG-2 或 MPEG-4 H. 264/MPEG-4 高级视频编码(AVC)),将图像划分为具有预定尺寸的多个块,随后通过帧间预测或帧内预测获得多个块的残差数据。通过变换、量化、扫描、游程编码和熵编码来对残差数据进行压缩。在熵编码中,对诸如变换系数或运动矢量的语法元素进行熵编码以输出比特流。在解码器端,从比特流提取语法元素并基于提取出的语法元素执行解码。

### 发明内容

[0003] 技术问题

[0004] 本发明提供一种更新参数的方法,通过该方法,使在对变换系数的熵编码和熵解码中使用的参数逐渐被改变,同时防止所述参数的骤变。

[0005] 本发明还提供一种对在通过使用二值化方法(诸如Golomb-rice 方法或级联码方法)来对诸如变换系数等级的语法元素的二值化中使用的参数进行更新的方法。

[0006] 解决方案

[0007] 根据本发明的实施例,提供了一种逐渐更新在对变换系数等级的二值化中使用的参数的参数更新方法。

[0008] 有益效果

[0009] 根据本发明的实施例,通过逐渐改变在变换系数的等级信息的熵编码中使用的参数,在编码期间产生的比特量可被减少,图像的增益可被增加。

### 附图说明

[0010] 图 1 是根据本发明的实施例的用于对视频进行编码的设备的框图;

[0011] 图 2 是根据本发明的实施例的用于对视频进行解码的设备的框图;

[0012] 图 3 是用于描述根据本发明的实施例的编码单元的概念的示图;

[0013] 图 4 是根据本发明的实施例的基于具有分层结构的编码单元的视频编码器的框图;

[0014] 图 5 是根据本发明的实施例的基于具有分层结构的编码单元的视频解码器的框图;

[0015] 图 6 是示出根据本发明的实施例的根据深度的较深层编码单元和分区(partition)的示图;

[0016] 图 7 是用于描述根据本发明的实施例的编码单元和变换单元之间的关系关系的示图;

[0017] 图 8 是用于描述根据本发明的实施例的与编码深度相应的编码单元的编码信息的示意图；

[0018] 图 9 是根据本发明的实施例的根据深度的较深层编码单元的示意图；

[0019] 图 10 至图 12 是用于描述根据本发明的实施例的编码单元、预测单元和频率变换单元之间的关系示意图；

[0020] 图 13 是用于描述根据表 1 的编码模式信息的编码单元、预测单元和变换单元之间的关系示意图；

[0021] 图 14 是示出根据本发明的实施例的对变换单元中包括的变换系数信息进行熵编码和熵解码的操作的流程图；

[0022] 图 15 示出根据本发明的实施例的被熵编码的变换单元；

[0023] 图 16 示出根据本发明的实施例的与图 15 的变换单元相应的有效图；

[0024] 图 17 示出与图 15 的  $4 \times 4$  变换单元相应的 `coeff_abs_level_greater1_flag`；

[0025] 图 18 示出与图 15 的  $4 \times 4$  变换单元相应的 `coeff_abs_level_greater2_flag`；

[0026] 图 19 示出与图 15 的  $4 \times 4$  变换单元相应的 `coeff_abs_level_remaining`；

[0027] 图 20 示出显示与图 15 至图 19 中示出的变换单元相关的语法元素的表；

[0028] 图 21 示出根据本发明的实施例的被二值化的 `coeff_abs_level_remaining` 的另一示例；

[0029] 图 22 是示出根据本发明的实施例的熵编码设备的结构的框图；

[0030] 图 23 是示出根据本发明的实施例的二值化设备的结构的框图；

[0031] 图 24 是示出根据本发明的实施例的对指示变换系数等级的语法元素进行熵编码的方法的流程图；

[0032] 图 25 是示出根据本发明的实施例的熵解码设备的框图；

[0033] 图 26 是示出根据本发明的实施例的反二值化设备的结构的框图；

[0034] 图 27 是示出根据本发明的实施例的对变换系数等级进行熵解码方法的流程图。

[0035] 最佳模式

[0036] 根据本发明的一方面，提供了一种更新用于对变换系数等级进行熵解码的参数的方法，所述方法包括：从比特流解析指示包括在变换单元中的变换系数的大小的变换系数等级语法元素；通过将在当前变换系数之前被恢复的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较，来确定是否更新先前参数，其中，先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的反二值化中被使用；通过基于确定的结果更新或保持先前参数，来获得在对指示当前变换系数的大小的当前变换系数等级语法元素的反二值化中使用的参数；通过使用获得的参数对当前变换系数等级语法元素进行反二值化来获得当前变换系数的大小，其中，所述预定临界值被设置为具有与先前参数成比例的值，当先前参数被更新时，更新后的参数与先前参数相比具有逐渐增加的值。

[0037] 根据本发明的另一方面，提供了一种用于对变换系数等级进行熵解码的设备，所述设备包括：解析单元，从比特流解析指示包括在变换单元中的变换系数的大小的变换系数等级语法元素；参数确定单元，通过将在当前变换系数之前被恢复的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较，来确定是否更新先前参数，并通过基于确定的结果更新或保持先前参数，来获得在对指示当前变换系数的大小的当前变换系数等

级语法元素的反二值化中使用的参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的反二值化中被使用;语法元素恢复单元,通过使用获得的参数对当前变换系数等级语法元素进行反二值化来获得当前变换系数的大小,其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新后的参数与先前参数相比具有逐渐增加的值。

[0038] 根据本发明的另一方面,提供了一种更新用于对变换系数等级进行熵编码的参数的方法,所述方法包括:以预定扫描顺序获得指示包括在变换单元中的变换系数的大小的变换系数等级语法元素;通过将在当前变换系数之前被编码的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,来确定是否更新先前参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的二值化中被使用;通过基于确定的结果更新或保持先前参数,来获得在对指示当前变换系数的大小的当前变换系数等级语法元素的二值化中使用的参数;通过使用获得的参数对当前变换系数的变换系数等级语法元素进行二值化来输出与当前变换系数的变换系数等级语法元素相应的比特串,其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新后的参数与先前参数相比具有逐渐增加的值。

[0039] 根据本发明的另一方面,一种用于对变换系数等级进行熵编码的设备,所述设备包括:参数确定单元,以预定扫描顺序获得指示包括在变换单元中的变换系数的大小的变换系数等级语法元素,通过将在当前变换系数之前被编码的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,来确定是否更新先前参数,通过基于确定的结果更新或保持先前参数,来获得在对指示当前变换系数的大小的当前变换系数等级语法元素的二值化中使用的参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的二值化中被使用;比特串产生单元,通过使用获得的参数对当前变换系数的变换系数等级语法元素进行二值化来输出与当前变换系数的变换系数等级语法元素相应的比特串,其中,所述预定临界值被设置为具有与先前参数成比例的值,当先前参数被更新时,更新后的参数与先前参数相比具有逐渐增加的值。

## 具体实施方式

[0040] 以下,将参照附图详细描述本发明的实施例。

[0041] 以下,将参照图 1 至图 13 描述根据本发明的实施例的用于更新在对变换单元的尺寸信息进行熵编码和解码时使用的参数的方法和设备。另外,将参照图 14 至图 29 详细地描述对语法元素进行熵编码和熵解码的方法,其中,通过使用参照图 1 至图 13 描述的对视频进行熵编码和解码的方法来获得所述语法元素。当诸如“…中的至少一个”的表达在一列元件之后时,其修饰整列元件而不修饰列出的单个元件。

[0042] 图 1 是根据本发明的实施例的视频编码设备 100 的框图。

[0043] 视频编码设备 100 包括分层编码器 110 和熵编码器 120。

[0044] 分层编码器 110 以预定数据单元为单位划分被编码的当前画面,以对每个数据单元执行编码。详细地,分层编码器 110 可基于最大编码单元划分当前画面,其中,最大编码单元是最大尺寸的编码单元。根据本发明的实施例的最大编码单元可以是尺寸为  $32 \times 32$ 、 $64 \times 64$ 、 $128 \times 128$ 、 $256 \times 256$  等的的数据单元,其中,数据单元的形状是宽度和长度是 2 的若

于次方并大于 8 的正方形。

[0045] 根据本发明的实施例的编码单元可由最大尺寸和深度来表征。深度表示编码单元在空间上从最大编码单元被划分的次数,并且随着深度加深,根据深度的较深层编码单元可从最大编码单元被划分到最小编码单元。最大编码单元的深度是最高深度,最小编码单元的深度是最低深度。由于与每个深度相应的编码单元的尺寸随着最大编码单元的深度加深而减小,因此与较高深度相应的编码单元可包括与较低深度相应的多个编码单元。

[0046] 如上所述,根据编码单元的最大尺寸将当前画面的图像数据划分为最大编码单元,每个最大编码单元可包括根据深度划分的较深层编码单元。由于根据本发明的实施例的最大编码单元根据深度被划分,因此包括在最大编码单元中的空间域的图像数据可根据深度被分层地分类。

[0047] 可预先确定编码单元的最大深度和最大尺寸,其中,所述最大深度和最大尺寸限制最大编码单元的高度和宽度被分层划分的总次数。

[0048] 分层编码器 110 对通过根据深度划分最大编码单元的区域而获得的至少一个划分的区域进行编码,并根据所述至少一个划分的区域确定用于输出最终编码的图像数据的深度。换句话说,分层编码器 110 通过根据当前画面的最大编码单元按照根据深度的较深层编码单元对图像数据进行编码并选择具有最小编码误差的深度,来确定编码深度。确定的编码深度和根据最大编码单元的编码的图像数据被输出到熵编码器 120。

[0049] 基于与等于或低于最大深度的至少一个深度相应的较深层编码单元来对最大编码单元中的图像数据进行编码,并基于每个较深层编码单元来比较对图像数据进行编码的结果。可在比较较深层编码单元的编码误差之后选择具有最小编码误差的深度。可针对每个最大编码单元选择至少一个编码深度。

[0050] 随着编码单元根据深度被分层划分并随着编码单元的数量增加,最大编码单元的尺寸被划分。另外,即使在一个最大编码单元中编码单元相应于同一深度,仍通过单独测量每个编码单元的图像数据的编码误差来确定是否将与同一深度相应的每个编码单元划分到更低深度。因此,即使在图像数据被包括在一个最大编码单元中时,图像数据仍根据深度被划分成区域,并且在所述一个最大编码单元中编码误差也会根据区域而不同,因此编码深度可根据图像数据中的区域而不同。因此,可在一个最大编码单元中确定一个或更多个编码深度,并可根据至少一个编码深度的编码单元来划分最大编码单元的图像数据。

[0051] 因此,分层编码器 110 可确定包括在最大编码单元中的具有树结构的编码单元。根据本发明的实施例的“具有树结构的编码单元”包括最大编码单元中所包括的所有较深层编码单元中的与被确定为编码深度的深度相应的编码单元。可在最大编码单元的同一区域中根据深度分层地确定具有编码深度的编码单元,并可在不同区域中独立地确定具有编码深度的编码单元。类似地,当前区域中的编码深度可与另一区域中的编码深度独立地被确定。

[0052] 根据本发明的实施例的最大深度是与从最大编码单元到最小编码单元的执行的划分次数相关的索引。根据本发明的实施例的第一最大深度可表示从最大编码单元到最小编码单元的总划分次数。根据本发明的实施例的第二最大深度可表示从最大编码单元到最小编码单元的深度级别的总数。例如,当最大编码单元的深度是 0 时,最大编码单元被划分一次的编码单元的深度可被设置为 1,最大编码单元被划分两次的编码单元的深度可被设



置为 2。这里,如果最小编码单元是最大编码单元被划分四次的编码单元,则存在深度 0、深度 1、深度 2、深度 3、深度 4 这 5 个深度级别,因此,第一最大深度可被设置为 4,第二最大深度可被设置为 5。

[0053] 可根据最大编码单元执行预测编码和变换。还根据最大编码单元,基于根据等于最大深度的深度或小于最大深度的深度的较深层编码单元来执行预测编码和变换。

[0054] 由于每当最大编码单元根据深度被划分时较深层编码单元的数量增加,因此对随着深度加深而产生的所有较深层编码单元执行包括预测编码和变换的编码。为了便于描述,现在将在最大编码单元中基于当前深度的编码单元来描述预测编码和变换。

[0055] 视频编码设备 100 可不同地选择用于对图像数据进行编码的数据单元的尺寸或形状。为了对图像数据进行编码,执行诸如预测编码、变换和熵编码的操作,此时,可针对所有操作使用相同的数据单元,或者可针对每个操作使用不同的数据单元。

[0056] 例如,视频编码设备 100 可不仅选择用于对图像数据进行编码的编码单元,还可选择与该编码单元不同的数据单元以对编码单元中的图像数据执行预测编码。

[0057] 为了在最大编码单元中执行预测编码,可基于与编码深度相应的编码单元(即,基于不再被划分为与更低深度相应的编码单元的编码单元)执行预测编码。在下文中,不再被划分并且成为用于预测编码的基本单元的编码单元现在将被称为“预测单元”。通过划分预测单元而获得的分区可包括预测单元或通过划分预测单元的高度和宽度中的至少一个而获得的数据单元。

[0058] 例如,当  $2N \times 2N$  (其中,  $N$  是正整数) 的编码单元不再被划分并且成为  $2N \times 2N$  的预测单元时,分区的尺寸可以是  $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$  或  $N \times N$ 。分区类型的示例包括通过对称地划分预测单元的高度或宽度而获得的对称分区、通过非对称地划分预测单元的高度或宽度(诸如,  $1:n$  或  $n:1$ ) 而获得的分区、通过几何地划分预测单元而获得的分区以及具有任意形状的分区。

[0059] 预测单元的预测模式可以是帧内模式、帧间模式和跳过模式中的至少一个。例如,可对  $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$  或  $N \times N$  的分区执行帧内模式或帧间模式。另外,可仅对  $2N \times 2N$  的分区执行跳过模式。可独立地对编码单元中的一个预测单元执行编码,从而选择具有最小编码误差的预测模式。

[0060] 视频编码设备 100 还可不仅基于用于对图像数据进行编码的编码单元对编码单元中的图像数据执行变换,还可基于与编码单元不同的数据单元对编码单元中的图像数据执行变换。

[0061] 为了在编码单元中执行变换,可基于具有小于或等于编码单元的尺寸的数据单元执行变换。例如,用于变换的数据单元可包括用于帧内模式的数据单元以及用于帧间模式的数据单元。

[0062] 用作变换的基础的数据单元现将被称为“变换单元”。与编码单元类似,编码单元中的变换单元可被递归地划分为更小尺寸的区域,从而变换单元可以以区域为单位被独立确定。因此,可根据具有根据变换深度的树结构的变换单元来划分编码单元中的残差数据。

[0063] 也可在变换单元中设置变换深度,变换深度指示通过划分编码单元的高度和宽度来达到变换单元而执行的划分次数。例如,在  $2N \times 2N$  的当前编码单元中,当变换单元的尺寸是  $2N \times 2N$  时,变换深度可以是 0,当变换单元的尺寸是  $N \times N$  时,变换深度可以是 1,当变

换单元的尺寸是  $N/2 \times N/2$  时,变换深度可以是 2。也就是说,还可根据变换深度设置具有树结构的变换单元。

[0064] 根据与编码深度相应的编码单元的编码信息不仅需要关于编码深度的信息,还需要与预测编码和变换相关的信息。因此,分层编码器 110 不仅确定具有最小编码误差的编码深度,还可确定预测单元中的分区类型、根据预测单元的预测模式以及用于变换的变换单元的尺寸。

[0065] 稍后将参照图 3 至图 12 来对根据本发明的实施例的最大编码单元中的根据树结构的编码单元和确定分区的方法进行详细描述。

[0066] 分层编码器 110 可通过使用基于拉格朗日乘子 (Lagrangian multiplier) 的率失真优化,测量根据深度的较深层编码单元的编码误差。

[0067] 熵编码器 120 在比特流中输出最大编码单元的图像数据以及关于根据编码深度的编码模式的信息,其中,最大编码单元的图像数据基于由分层编码器 110 确定的至少一个编码深度被编码。编码图像数据可以是图像的残差数据的编码结果。关于根据编码深度的编码模式的信息可包括关于编码深度的信息、关于预测单元中的分区类型的信息、预测模式信息和变换单元的尺寸信息。具体地,如稍后将描述,熵编码器 120 在对指示变换单元的尺寸的语法元素进行熵编码时,通过使用被逐渐更新的参数,根据比特串对指示变换系数的大小的语法元素进行二值化。稍后将详细描述通过使用熵编码器 120 对变换单元进行熵编码的操作。

[0068] 可通过使用根据深度的划分信息来定义关于编码深度的信息,其中,所述根据深度的划分信息指示是否对更低深度而非当前深度的编码单元执行编码。如果当前编码单元的当前深度是编码深度,则当前编码单元中的图像数据被编码和输出,因此划分信息可被定义为不将当前编码单元划分至更低深度。可选择地,如果当前编码单元的当前深度不是编码深度,则对更低深度的编码单元执行编码,因此,划分信息可被定义为对当前编码单元进行划分以获得更低深度的编码单元。

[0069] 如果当前深度不是编码深度,则对被划分为更低深度的编码单元的编码单元执行编码。由于在当前深度的一个编码单元中存在更低深度的至少一个编码单元,因此对更低深度的每个编码单元重复执行编码,因此,可针对具有同一深度的编码单元递归地执行编码。

[0070] 由于针对一个最大编码单元确定具有树结构的编码单元,并且针对编码深度的编码单元确定关于至少一个编码模式的信息,因此可针对一个最大编码单元确定关于至少一个编码模式的信息。另外,由于根据深度分层地划分图像数据,因此最大编码单元的图像数据的编码深度可根据位置而不同,因此,可针对图像数据设置关于编码深度和编码模式的信息。

[0071] 因此,熵编码器 120 可将关于相应的编码深度和编码模式的编码信息分配给包括在最大编码单元中的编码单元、预测单元和最小单元中的至少一个。

[0072] 根据本发明的实施例的最小单元是通过将构成最低深度的最小编码单元划分为 4 份而获得的正方形形状的数据单元。可选择地,最小单元可以是可被包括在最大编码单元中包括的所有编码单元、预测单元、分区单元和变换单元中的最大正方形形状的数据单元。

[0073] 例如,通过熵编码器 120 输出的编码信息可被分类为根据编码单元的编码信息和

根据预测单元的编码信息。根据编码单元的编码信息可包括关于预测模式的信息以及关于分区的尺寸的信息。根据预测单元的编码信息可包括关于帧间模式的估计方向的信息、关于帧间模式的参考图像索引的信息、关于运动矢量的信息、关于帧内模式的色度分量的信息以及关于帧内模式的插值方法的信息。此外,关于根据画面、条带或 GOP 定义的编码单元的最大尺寸的信息以及关于最大深度的信息可被插入到比特流的头中。

[0074] 在视频编码设备 100 中,较深层编码单元可以通过将作为上一层的更高深度的编码单元的高度或宽度划分为 2 份而获得的编码单元。换句话说,在当前深度的编码单元的尺寸是  $2N \times 2N$  时,更低深度的编码单元的尺寸是  $N \times N$ 。另外,尺寸为  $2N \times 2N$  的当前深度的编码单元可包括最多 4 个更低深度的编码单元。

[0075] 因此,基于考虑当前画面的特性而确定的最大编码单元的尺寸和最大深度,视频编码设备 100 可通过针对每个最大编码单元确定具有最佳形状和最佳尺寸的编码单元,来形成具有树结构的编码单元。另外,由于可通过使用各种预测模式和变换中的任意一种来对每个最大编码单元执行编码,因此可考虑各种图像尺寸的编码单元的特性来确定最佳编码模式。

[0076] 因此,如果按照传统的宏块对具有高分辨率或大数据量的图像进行编码,则每个画面的宏块的数量急剧增加。因此,针对每个宏块产生的压缩信息的条数增加,因而难以发送压缩信息并且数据压缩效率下降。然而,通过使用视频编码设备 100,因为在考虑图像的特征的同时调整了编码单元,同时,在考虑图像的尺寸的同时增大了编码单元的最大尺寸,所以图像压缩效率可增加。

[0077] 图 2 是根据本发明的实施例的视频解码设备 200 的框图。

[0078] 视频解码设备 200 包括解析器 210、熵解码器 220 和分层解码器 230。用于视频解码设备 200 的各种操作的各种术语(诸如编码单元、深度、预测单元、变换单元以及关于各种编码模式的信息)的定义与参照图 1 和视频编码设备 100 所描述的那些术语的定义相同。

[0079] 解析器 210 接收编码的视频的比特流并对语法元素进行解析。熵解码器 220 通过对解析后的语法元素执行熵解码来提取指示基于具有树结构的编码单元的编码图像数据的语法元素,并将提取的语法元素输出到分层解码器 230。也就是说,熵解码器 220 对以 0 和 1 的比特串的形式接收的语法元素执行熵解码,从而恢复该语法元素。

[0080] 另外,熵解码器 220 从解析的比特流提取根据每个最大编码单元的具有树结构的编码单元的关于编码深度的信息、编码模式、颜色分量信息和预测模式信息等。提取的关于编码深度和编码模式的信息被输出到分层解码器 230。比特流中的图像数据被划分为最大编码单元,从而分层解码器 230 可针对每个最大编码单元对图像数据进行解码。

[0081] 可针对关于与编码深度相应的至少一个编码单元的信息设置关于根据最大编码单元的编码深度和编码模式的信息,关于编码模式的信息可包括关于与编码深度相应的相应编码单元的分区类型的信息、关于预测模式的信息以及变换单元的尺寸的信息。另外,根据深度的划分信息可被提取作为关于编码深度的信息。

[0082] 由熵解码器 220 提取的关于根据每个最大编码单元的编码深度和编码模式的信息是关于这样的编码深度和编码模式的信息,即:所述编码深度和编码模式被确定为当编码器(诸如视频编码设备 100)根据每个最大编码单元针对根据深度的每个较深层编码单

元重复执行编码时产生最小编码误差。因此,视频解码设备 200 可通过根据产生最小编码误差的编码深度和编码模式对图像数据进行解码来恢复图像。

[0083] 由于关于编码深度和编码模式的编码信息可被分配给相应的编码单元、预测单元和最小单元中的预定数据单元,因此熵解码器 220 可提取关于根据预定数据单元的编码深度和编码模式的信息。当关于相应最大编码单元的编码深度和编码模式的信息被分配给每个预定数据单元时,被分配有相同的关于编码深度和编码模式的信息的预定数据单元可被推断为包括在同一最大编码单元中的数据单元。

[0084] 此外,如稍后将描述,熵解码器 220 通过使用被逐渐更新的参数来对指示变换系数的大小的语法元素进行反二值化。稍后将详细描述通过使用熵解码器 220 对与指示变换单元的尺寸的语法元素相应的比特串进行反二值化,来获得变换系数的尺寸信息的操作。

[0085] 分层解码器 230 可通过基于关于根据最大编码单元的编码深度和编码模式的信息,对每个最大编码单元中的图像数据进行解码来恢复当前画面。换句话说,分层解码器 230 可针对包括在每个最大编码单元中的具有树结构的编码单元中的每个编码单元,基于提取的关于分区类型、预测模式和变换单元的信息对编码的图像数据进行解码。解码处理可包括预测和逆变换,其中,所述预测包括帧内预测和运动补偿。

[0086] 分层解码器 230 可基于关于根据编码深度的编码单元的预测单元的分区类型和预测模式的信息,根据每个编码单元的分区和预测模式执行帧内预测或运动补偿。

[0087] 此外,分层解码器 230 可基于关于根据编码深度的编码单元的变换单元的尺寸的信息,根据编码单元中的每个变换单元执行逆变换,以便执行根据最大编码单元的逆变换。

[0088] 分层解码器 230 可通过使用根据深度的划分信息确定当前最大编码单元的至少一个编码深度。如果划分信息指示在当前深度中图像数据不再被划分,则当前深度是编码深度。因此,分层解码器 230 可通过使用关于预测单元的分区类型、预测模式和变换单元的尺寸的信息,针对当前最大编码单元的图像数据对当前深度的编码单元进行解码。

[0089] 换句话说,可通过观察为编码单元、预测单元和最小单元中的预定数据单元分配的编码信息集,收集包含编码信息(所述编码信息包括相同的划分信息)的数据单元,收集的数据单元可被认为是将由分层解码器 230 以相同编码模式解码的一个数据单元。

[0090] 视频解码设备 200 可获得与在对每个最大编码单元递归地执行编码时产生最小编码误差的至少一个编码单元有关的信息,并且视频解码设备 200 可使用所述信息来对当前画面进行解码。换句话说,可以对每个最大编码单元中被确定为最佳编码单元的具有树结构的编码单元的编码图像数据进行解码。

[0091] 因此,即使图像数据具有高分辨率和大数据量,也可通过使用编码单元的尺寸和编码模式有效地对所述图像数据进行解码和恢复,其中,通过使用从编码器接收的关于最佳编码模式的信息,根据图像数据的特性来自适应地确定编码单元的尺寸和编码模式。

[0092] 现将参照图 3 至图 13 描述根据本发明的实施例的确定具有树结构的编码单元、预测单元和变换单元的方法。

[0093] 图 3 是用于描述根据本发明的实施例的编码单元的概念的示图。

[0094] 编码单元的尺寸可按照宽度 × 高度来表示,并可以是  $64 \times 64$ 、 $32 \times 32$ 、 $16 \times 16$  和  $8 \times 8$ 。 $64 \times 64$  的编码单元可被划分为  $64 \times 64$ 、 $64 \times 32$ 、 $32 \times 64$  或  $32 \times 32$  的分区, $32 \times 32$  的编码单元可被划分为  $32 \times 32$ 、 $32 \times 16$ 、 $16 \times 32$  或  $16 \times 16$  的分区, $16 \times 16$  的编码单元可被划

分为  $16 \times 16$ 、 $16 \times 8$ 、 $8 \times 16$  或  $8 \times 8$  的分区， $8 \times 8$  的编码单元可被划分为  $8 \times 8$ 、 $8 \times 4$ 、 $4 \times 8$  或  $4 \times 4$  的分区。

[0095] 关于视频数据 310，设置了分辨率是  $1920 \times 1080$ ，编码单元的最大尺寸是 64，并且最大深度是 2。关于视频数据 320，设置了分辨率是  $1920 \times 1080$ ，编码单元的最大尺寸是 64，并且最大深度是 3。关于视频数据 330，设置了分辨率是  $352 \times 288$ ，编码单元的最大尺寸是 16，并且最大深度是 1。图 3 中示出的最大深度表示从最大编码单元到最小解码单元的划分总数。

[0096] 如果分辨率高或者数据量大，则编码单元的最大尺寸可以较大以便不仅提高编码效率还精确地地反映图像的特性。因此，分辨率比视频数据 330 更高的视频数据 310 和视频数据 320 的编码单元的最大尺寸可以是 64。

[0097] 由于视频数据 310 的最大深度是 2，因此由于通过划分最大编码单元两次深度被加深到两层，视频数据 310 的编码单元 315 可包括长轴尺寸为 64 的最大编码单元以及长轴尺寸为 32 和 16 的编码单元。同时，由于视频数据 330 的最大深度是 1，因此由于通过划分最大编码单元一次深度被加深到一层，视频数据 330 的编码单元 335 可包括长轴尺寸为 16 的最大编码单元以及长轴尺寸为 8 的编码单元。

[0098] 由于视频数据 320 的最大深度是 3，因此由于通过划分最大编码单元三次深度被加深到 3 层，视频数据 320 的编码单元 325 可包括长轴尺寸为 64 的最大编码单元以及长轴尺寸为 32、16 和 8 的编码单元。随着深度加深，可精确地表示详细的信息。

[0099] 图 4 是根据本发明的实施例的基于具有分层结构的编码单元的视频编码器 400 的框图。

[0100] 帧内预测器 410 针对当前帧 405，对帧内模式下的编码单元执行帧内预测，运动估计器 420 和运动补偿器 425 通过使用当前帧 405 和参考帧 495，对帧间模式下的编码单元执行帧间估计和运动补偿。

[0101] 从帧内预测器 410、运动估计器 420 和运动补偿器 425 输出的数据通过变换器 430 和量化器 440 被输出为量化后的变换系数。量化后的变换系数通过反量化器 460 和逆变换器 470 被恢复为空间域中的数据，并且恢复的空间域中的数据在通过去块单元 480 和环路滤波单元 490 被后处理之后被输出为参考帧 495。量化的变换系数可通过熵编码器 450 被输出为比特流 455。

[0102] 熵编码单元 450 对如下的与变换单元的语法元素进行算术编码：诸如，指示非 0 的变换单元的位置的有效图、指示变换单元是否具有大于 1 的值的的第一临界值标记 (coeff\_abs\_level\_greater1\_flag)、指示变换单元是否具有大于 2 的值的第二临界值 (coeff\_abs\_level\_greater2\_flag)、变换系数的大小信息 (coeff\_abs\_level\_remaining)，其中，所述大小信息与基于第一临界值和第二临界值确定的基本等级 (baseLevel) 和实际的变换系数 (abscoeff) 之间的差相应。

[0103] 为了使视频编码器 400 被应用到视频编码设备 100 中，视频编码器 400 的所有元件（即，帧内预测器 410、运动估计器 420、运动补偿器 425、变换器 430、量化器 440、熵编码器 450、反量化器 460、逆变换器 470、去块单元 480 和环路滤波单元 490）必需在考虑每个最大编码单元的最大深度的同时，基于具有树结构的多个编码单元中的每个编码单元执行操作。

[0104] 具体地讲,帧内预测器 410、运动估计器 420 和运动补偿器 425 在考虑当前最大编码单元的最大尺寸和最大深度的同时,确定具有树结构的多个编码单元中的每个编码单元的分区和预测模式,并且变换器 430 确定具有树结构的多个编码单元中的每个编码单元中的变换单元的尺寸。

[0105] 图 5 是根据本发明的实施例的基于编码单元的视频解码器 500 的框图。

[0106] 解析器 510 从比特流 505 解析将被解码的编码图像数据以及解码所需的关于编码的信息。编码图像数据通过解码器 520 和反量化器 530 被输出为反量化的数据。熵解码器 520 从比特流获得如下的与变换单元相关的元素并对获得的语法元素进行算术解码以便恢复语法元素:即,指示非 0 的变换单元的位置的有效图、指示变换单元是否具有大于 1 的值的第二临界值标记 (coeff\_abs\_level\_greater1\_flag)、指示变换单元是否具有大于 2 的值的第二临界值 (coeff\_abs\_level\_greater2\_flag)、变换系数的大小信息 (coeff\_abs\_level\_remaining),其中,所述大小信息与基于第一临界值和第二临界值确定的基本等级 (baseLevel) 和实际的变换系数 (abscoeff) 之间的差相应。

[0107] 逆变换器 540 将反量化的数据恢复为空间域中的图像数据。帧内预测器 550 针对空间域中的图像数据对帧内模式下的编码单元执行帧内预测,运动补偿器 560 通过使用参考帧 585 对帧间模式下的编码单元执行运动补偿。

[0108] 已经过帧内预测器 550 和运动补偿器 560 的空间域中的图像数据在通过去块单元 570 和环路滤波单元 580 被后处理之后,可被输出为恢复的帧 595。另外,经过去块单元 570 和环路滤波单元 580 被后处理的图像数据可被输出为参考帧 585。

[0109] 为了使图像解码器 500 被应用到视频解码设备 200 中,图像解码器 500 的所有元件(即,解析器 510、熵解码器 520、反量化器 530、逆变换器 540、帧内预测器 550、运动补偿器 560、去块单元 570 和环路滤波单元 580)基于每个最大编码单元的具有树结构的编码单元执行操作。

[0110] 帧内预测器 550 和运动补偿器 560 针对具有树结构的每个编码单元确定分区和预测模式,并且逆变换器 540 必需针对每个编码单元确定变换单元的尺寸。

[0111] 图 6 是示出根据本发明的实施例的根据深度的较深层编码单元和分区的示图。

[0112] 视频编码设备 100 和视频解码设备 200 使用分层的编码单元以考虑图像的特性。编码单元的最大高度、最大宽度和最大深度可根据图像的特性被自适应地确定,或者可由用户不同地设置。根据深度的较深层编码单元的尺寸可根据编码单元的预定最大尺寸被确定。

[0113] 在根据本发明的实施例的编码单元的分层结构 600 中,编码单元的最大高度和最大宽度都是 64,并且最大深度是 4。由于深度沿着分层结构 600 的纵轴加深,因此较深层编码单元的高度和宽度都被划分。另外,作为用于每个较深层编码单元的预测编码的基础的预测单元和分区沿分层结构 600 的横轴被示出。

[0114] 换句话说,编码单元 610 是分层结构 600 中的最大编码单元,其中,深度是 0,尺寸(即,高度乘宽度)是  $64 \times 64$ 。深度沿纵轴加深,并且存在尺寸为  $32 \times 32$  且深度为 1 的编码单元 620、尺寸为  $16 \times 16$  且深度为 2 的编码单元 630、尺寸为  $8 \times 8$  且深度为 3 的编码单元 640 以及尺寸为  $4 \times 4$  且深度为 4 的编码单元 650。尺寸为  $4 \times 4$  且深度为 4 的编码单元 650 是最小编码单元。

[0115] 编码单元的预测单元和分区根据每个深度沿横轴排列。换句话说,如果尺寸为 $64 \times 64$ 且深度为0的编码单元610是预测单元,则该预测单元可被划分为包括在编码单元610中的分区(即,尺寸为 $64 \times 64$ 的分区610、尺寸为 $64 \times 32$ 的分区612、尺寸为 $32 \times 64$ 的分区614、或者尺寸为 $32 \times 32$ 的分区616)。

[0116] 类似地,尺寸为 $32 \times 32$ 且深度为1的编码单元620的预测单元可被划分为包括在编码单元620中的分区(即,尺寸为 $32 \times 32$ 的分区620、尺寸为 $32 \times 16$ 的分区622、尺寸为 $16 \times 32$ 的分区624以及尺寸为 $16 \times 16$ 的分区626)。

[0117] 类似地,尺寸为 $16 \times 16$ 且深度为2的编码单元630的预测单元可被划分为包括在编码单元630中的分区(即,包括在编码单元630中的尺寸为 $16 \times 16$ 的分区、尺寸为 $16 \times 8$ 的分区632、尺寸为 $8 \times 16$ 的分区634以及尺寸为 $8 \times 8$ 的分区636)。

[0118] 类似地,尺寸为 $8 \times 8$ 且深度为3的编码单元640的预测单元可被划分为包括在编码单元640中的分区(即,包括在编码单元640中的尺寸为 $8 \times 8$ 的分区、尺寸为 $8 \times 4$ 的分区642、尺寸为 $4 \times 8$ 的分区644以及尺寸为 $4 \times 4$ 的分区646)。

[0119] 尺寸为 $4 \times 4$ 且深度为4的编码单元650是最小编码单元以及最低深度的编码单元。编码单元650的预测单元仅被分配给尺寸为 $4 \times 4$ 的分区。

[0120] 为了确定构成最大编码单元610的编码单元的至少一个编码深度,视频编码设备100的分层编码器110对包括在最大编码单元610中的与每个深度相应的编码单元执行编码。

[0121] 随着深度加深,包括相同范围和相同尺寸的数据的根据深度的较深层编码单元的数量增加。例如,需要四个与深度2相应的编码单元以覆盖包括在一个与深度1相应的编码单元中的数据。因此,为了根据深度对相同数据的多个编码结果进行比较,与深度1相应的编码单元以及与深度2相应的四个编码单元都被编码。

[0122] 为了针对多个深度中的当前深度执行编码,通过沿分层结构600的横轴,对与当前深度相应的多个编码单元中的每个预测单元执行编码来针对当前深度选择最小编码误差。可选择地,随着深度沿分层结构600的纵轴加深,可通过针对每个深度执行编码,通过根据深度比较最小编码误差,从而搜索最小编码误差。在最大编码单元610中的具有最小编码误差的深度和分区可被选为最大编码单元610的编码深度和分区类型。

[0123] 图7是用于描述根据本发明的实施例的编码单元710和变换单元720之间的关系的示意图。

[0124] 视频编码设备100或视频解码设备200针对每个最大编码单元根据尺寸小于或等于最大编码单元的编码单元来对图像进行编码或解码。可基于不大于相应编码单元的数据单元来选择在编码期间用于变换的变换单元的尺寸。

[0125] 例如,在视频编码设备100或视频解码设备200中,如果编码单元710的尺寸是 $64 \times 64$ ,则可通过使用尺寸为 $32 \times 32$ 的变换单元720来执行变换。

[0126] 另外,可通过对小于 $64 \times 64$ 的尺寸为 $32 \times 32$ 、 $16 \times 16$ 、 $8 \times 8$ 和 $4 \times 4$ 的每个变换单元执行变换,来对尺寸为 $64 \times 64$ 的编码单元710的数据进行编码,并且随后可选择具有最小编码误差的变换单元。

[0127] 图8是用于描述根据本发明的实施例的与编码深度相应的编码单元的编码信息的示意图。

[0128] 视频编码设备 100 的输出单元 130 可对关于分区类型的信息 800、关于预测模式的信息 810 以及关于与编码深度相应的每个编码单元的变换单元的的尺寸的信息 820 进行编码,并将信息 800、信息 810 和信息 820 作为关于编码模式的信息来发送。

[0129] 信息 800 指示关于通过划分当前编码单元的预测单元而获得的分区的形状的信息,其中,所述分区是用于对当前编码单元进行预测编码的数据单元。例如,尺寸为  $2N \times 2N$  的当前编码单元  $CU_0$  可被划分为如下分区中的任何一个:尺寸为  $2N \times 2N$  的分区 802、尺寸为  $2N \times N$  的分区 804、尺寸为  $N \times 2N$  的分区 806 以及尺寸为  $N \times N$  的分区 808。这里,关于分区类型的信息 800 被设置为指示尺寸为  $2N \times 2N$  的分区 802、尺寸为  $2N \times N$  的分区 804、尺寸为  $N \times 2N$  的分区 806 以及尺寸为  $N \times N$  的分区 808 中的一个。

[0130] 信息 810 指示每个分区的预测模式。例如,信息 810 可指示对由信息 800 指示的分区执行的预测编码的模式(即,帧内模式 812、帧间模式 814 或跳过模式 816)。

[0131] 信息 820 指示当对当前编码单元执行变换时所基于的变换单元。例如,变换单元可以是第一帧内变换单元 822、第二帧内变换单元 824、第一帧间变换单元 826 或第二帧内变换单元 828。

[0132] 视频解码设备 200 的图像数据和编码信息提取器 210 可根据每个较深层编码单元来提取并使用关于编码单元的信息 800、关于预测模式的信息 810、关于变换单元的的尺寸的信息 820,以用于解码。

[0133] 图 9 是根据本发明的实施例的根据深度的较深层编码单元的示图。

[0134] 划分信息可用于指示深度的改变。划分信息指示当前深度的编码单元是否被划分为更低深度的编码单元。

[0135] 用于对深度为 0 且尺寸为  $2N_0 \times 2N_0$  的编码单元 900 进行预测编码的预测单元 910 可包括如下分区类型的分区:尺寸为  $2N_0 \times 2N_0$  的分区类型 912、尺寸为  $2N_0 \times N_0$  的分区类型 914、尺寸为  $N_0 \times 2N_0$  的分区类型 916 以及尺寸为  $N_0 \times N_0$  的分区类型 918。图 9 仅示出通过对预测单元 910 进行对称划分而获得的分区类型 912 至 918,但是分区类型不限于此,并且预测单元 910 的分区可包括非对称分区、具有预定形状的分区以及具有几何形状的分区的。

[0136] 根据每个分区类型对如下分区重复地执行预测编码:尺寸为  $2N_0 \times 2N_0$  的一个分区、尺寸为  $2N_0 \times N_0$  的两个分区、尺寸为  $N_0 \times 2N_0$  的两个分区以及尺寸为  $N_0 \times N_0$  的四个分区。可对尺寸为  $2N_0 \times 2N_0$ 、 $N_0 \times 2N_0$ 、 $2N_0 \times N_0$  以及  $N_0 \times N_0$  的分区执行帧内模式和帧间模式下的预测编码。仅对尺寸为  $2N_0 \times 2N_0$  的分区执行跳过模式下的预测编码。

[0137] 如果在尺寸为  $2N_0 \times 2N_0$ 、 $2N_0 \times N_0$  和  $N_0 \times 2N_0$  的分区类型 912 至分区类型 916 之一中的编码误差最小,则预测单元 910 可不被划分为更低深度。

[0138] 如果在尺寸为  $N_0 \times N_0$  的分区类型 918 中编码误差最小,则在操作 920,深度从 0 改变为 1 以划分分区类型 918,并且对深度为 2 且尺寸为  $N_0 \times N_0$  的分区类型编码单元重复地执行编码以搜索最小编码误差。

[0139] 用于对深度为 1 且尺寸为  $2N_1 \times 2N_1 (= N_0 \times N_0)$  的(分区类型)编码单元 930 进行预测编码的预测单元 940 可包括如下分区类型的分区:尺寸为  $2N_1 \times 2N_1$  的分区类型 942、尺寸为  $2N_1 \times N_1$  的分区类型 944、尺寸为  $N_1 \times 2N_1$  的分区类型 946 以及尺寸



为  $N_1 \times N_1$  的分区类型 948。

[0140] 如果在尺寸为  $N_1 \times N_1$  的分区类型 948 中编码误差最小,则在操作 950,深度从 1 改变为 2 以划分分区类型 948,并且对深度为 2 且尺寸为  $N_2 \times N_2$  的编码单元 960 重复地执行编码以搜索最小编码误差。

[0141] 当最大深度是  $d$  时,根据每个深度的划分操作可被执行直到深度变为  $d-1$  时,并且划分信息可被编码直到深度是 0 至  $d-2$  之一。换句话说,当执行编码直到在操作 970 与深度  $d-2$  相应的编码单元被划分之后深度是  $d-1$  时,用于对深度为  $d-1$  且尺寸为  $2N_{(d-1)} \times 2N_{(d-1)}$  的编码单元 980 进行预测编码的预测单元 990 可包括如下分区类型的分区:尺寸为  $2N_{(d-1)} \times 2N_{(d-1)}$  的分区类型 992、尺寸为  $2N_{(d-1)} \times N_{(d-1)}$  的分区类型 994、尺寸为  $N_{(d-1)} \times 2N_{(d-1)}$  的分区类型 996 以及尺寸为  $N_{(d-1)} \times N_{(d-1)}$  的分区类型 998。

[0142] 可对分区类型 992 至 998 中的如下分区重复地执行预测编码以搜索具有最小编码误差的分区类型:尺寸为  $2N_{(d-1)} \times 2N_{(d-1)}$  的一个分区、尺寸为  $2N_{(d-1)} \times N_{(d-1)}$  的两个分区、尺寸为  $N_{(d-1)} \times 2N_{(d-1)}$  的两个分区、尺寸为  $N_{(d-1)} \times N_{(d-1)}$  的四个分区。

[0143] 即使在尺寸为  $N_{(d-1)} \times N_{(d-1)}$  的分区类型 998 具有最小编码误差时,由于最大深度是  $d$ ,因此深度为  $d-1$  的编码单元  $CU_{(d-1)}$  不再被划分到更低深度,并且构成当前最大编码单元 900 的编码单元的编码深度被确定为  $d-1$  并且当前最大编码单元 900 的分区类型可被确定为  $N_{(d-1)} \times N_{(d-1)}$ 。另外,由于最大深度是  $d$ ,因此最小编码单元 952 的划分信息不被设置。

[0144] 数据单元 999 可以是当前最大编码单元的“最小单元”。根据本发明的实施例的最小单元可以通过将最小编码单元 980 划分为 4 份而获得的矩形数据单元。通过重复地执行编码,视频编码设备 100 可通过比较根据编码单元 900 的多个深度的编码误差来选择具有最小编码误差的深度以确定编码深度,并且将相应分区类型以及预测模式设置为编码深度的编码模式。

[0145] 这样,根据深度的最小编码误差在所有的深度 1 至  $d$  中被比较,并且具有最小编码误差的深度可被确定为编码深度。可对编码深度、预测单元的分区类型以及预测模式进行编码并将其作为关于编码模式的信息发送。另外,由于编码单元从深度 0 被划分到编码深度,因此仅该编码深度的划分信息被设置为 0,除了编码深度之外的深度的划分信息被设置为 1。

[0146] 视频解码设备 200 的熵解码器 220 可提取并使用关于编码单元 900 的编码深度和预测单元的信息以对编码单元 912 进行解码。视频解码设备 200 可通过使用根据深度的划分信息来将划分信息是 0 的深度确定为编码深度,并且使用关于相应深度的编码模式的信息用于解码。

[0147] 图 10 至图 12 是用于描述根据本发明的实施例的编码单元 1010、预测单元 1060 和变换单元 1070 之间的关系的示图。

[0148] 编码单元 1010 是最大编码单元中的与由视频编码设备 100 确定的编码深度相应的具有树结构的编码单元。预测单元 1060 是每个编码单元 1010 的预测单元的分区,变换单元 1070 是每个编码单元 1010 的变换单元。

[0149] 当在编码单元 1010 中最大编码单元的深度是 0 时,编码单元 1012 和 1054 的深度是 1,编码单元 1014、1016、1018、1028、1050 和 1052 的深度是 2,编码单元 1020、1022、1024、

1026、1030、1032 和 1048 的深度是 3, 编码单元 1040、1042、1044 和 1046 的深度是 4。

[0150] 在预测单元 1060 中, 一些编码单元 1014、1016、1022、1032、1048、1050、1052 和 1054 通过划分编码单元而获得。换句话说, 编码单元 1014、1022、1050 和 1054 中的分区类型的尺寸为  $2N \times N$ , 编码单元 1016、1048 和 1052 中的分区类型的尺寸为  $N \times 2N$ , 编码单元 1032 的分区类型的尺寸为  $N \times N$ 。编码单元 1010 的预测单元和分区小于或等于每个编码单元。

[0151] 以小于编码单元 1052 的数据单元对变换单元 1070 中的编码单元 1052 的图像数据执行变换或逆变换。另外, 变换单元 1070 中的编码单元 1014、1016、1022、1032、1048、1050、1052 和 1054 在尺寸和形状上与预测单元 1060 中的编码单元 1014、1016、1022、1032、1048、1050、1052 和 1054 不同。换句话说, 视频编码设备 100 和视频解码设备 200 可对相同编码单元中的数据单元单独地执行帧内预测、运动估计、运动补偿、变换和逆变换。

[0152] 因此, 对最大编码单元的每个区域中的具有分层结构的每个编码单元递归地执行编码, 以确定最佳编码单元, 因此可获得具有递归树结构的编码单元。编码信息可包括关于编码单元的划分信息、关于分区类型的信息、关于预测模式的信息以及关于变换单元的尺寸的信息。

[0153] 表 1 显示可由视频编码设备 100 和视频解码设备 200 设置的编码信息。

[0154] 表 1

[0155]

划分信息 0 (对尺寸为 $2N \times 2N$ 和当前深度为 $d$ 的编码单元进行编码)					划分信息 1
预测模式	分区类型		变换单元的尺寸		对具有更低深度 $d+1$ 的编码单元重复地进行编码
帧内 帧间 跳过(仅 $2N \times 2N$ )	对称分区类型	非对称分区类型	变换单元的划分信息 0	变换单元的划分信息 1	
	$2N \times 2N$	$2N \times nU$	$2N \times 2N$	$N \times N$ (对称类型)	
	$2N \times N$	$2N \times nD$		$N/2 \times N/2$ (非对称类型)	
	$N \times 2N$	$nL \times 2N$			
$N \times N$	$nR \times 2N$				

[0156] 视频编码设备 100 的熵编码器 120 可输出关于具有树结构的编码单元的编码信息, 并且视频解码设备 200 的熵解码器 220 可从接收的比特流提取关于具有树结构的编码单元的编码信息。

[0157] 划分信息指示当前编码单元是否被划分为更低深度的编码单元。如果当前深度  $d$  的划分信息是 0, 则当前编码单元不再被划分为更低深度的深度是编码深度, 因此可针对编码深度定义关于分区类型、预测模式以及变换单元的尺寸的信息。如果当前编码单元根据划分信息被进一步划分, 则对更低深度的四个划分编码单元独立地执行编码。

[0158] 预测模式可以是帧内模式、帧间模式和跳过模式中的一个。可在所有分区类型中定义帧内模式和帧间模式, 并且可仅在尺寸为  $2N \times 2N$  的分区类型中定义跳过模式。

[0159] 关于分区类型的信息可指示通过对称地划分预测单元的高度或宽度而获得的尺寸为  $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$  和  $N \times N$  的对称分区类型、以及通过非对称地划分预测单元的高度或宽度而获得的尺寸为  $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$  和  $nR \times 2N$  的非对称分区类型。可通过以  $1:n$  和  $n:1$  划分预测单元的高度来分别地获得尺寸为  $2N \times nU$  和  $2N \times nD$  的非对称分区类型，可通过以  $1:n$  和  $n:1$  划分预测单元的宽度来分别地获得尺寸为  $nL \times 2N$  和  $nR \times 2N$  的非对称分区类型（其中， $n$  是大于 1 的整数）。

[0160] 变换单元的尺寸可被设置为帧内模式下的两种类型以及帧间模式下的两种类型。换句话说，如果变换单元的划分信息是 0，则变换单元的尺寸可以是作为当前编码单元的尺寸的  $2N \times 2N$ 。如果变换单元的划分信息是 1，则可通过划分当前编码单元来获得变换单元。另外，如果尺寸为  $2N \times 2N$  的当前编码单元的分区类型是对称分区类型，则变换单元的尺寸可以是  $N \times N$ ，如果当前编码单元的分区类型是非对称分区类型，则变换单元的尺寸可以是  $N/2 \times N/2$ 。

[0161] 关于具有树结构的编码单元的编码信息可包括与编码深度相应的编码单元、预测单元以及最小单元中的至少一个。与编码深度相应的编码单元可包括：包含相同编码信息的预测单元和最小单元中的至少一个。

[0162] 因此，通过对邻近数据单元的编码信息进行比较，来确定邻近数据单元是否被包括在与编码深度相应的相同编码单元中。另外，通过使用数据单元的编码信息来确定与编码深度相应的相应编码单元，因此最大编码单元中的编码深度的分布可被确定。

[0163] 因此，如果基于邻近数据单元的编码信息预测当前编码单元，则与当前编码单元邻近的较深层编码单元中的数据单元的编码信息可被直接参考和使用。

[0164] 可选择地，如果基于邻近数据单元的编码信息来预测当前编码单元，则使用数据单元的编码信息来搜索与当前编码单元邻近的数据单元，并且搜索到的邻近编码单元可被参考以预测当前编码单元。

[0165] 图 13 是用于描述根据表 1 的编码模式信息的编码单元、预测单元和变换单元之间的关系的关系的示意图。

[0166] 最大编码单元 1300 包括多个编码深度的编码单元 1302、1304、1306、1312、1314、1316 和 1318。这里，由于编码单元 1318 是编码深度的编码单元，因此划分信息可被设置为 0。关于尺寸为  $2N \times 2N$  的编码单元 1318 的分区类型的信息可被设置为以下分区类型之一：尺寸为  $2N \times 2N$  的分区类型 1322、尺寸为  $2N \times N$  的分区类型 1324、尺寸为  $N \times 2N$  的分区类型 1326、尺寸为  $N \times N$  的分区类型 1328、尺寸为  $2N \times nU$  的分区类型 1332、尺寸为  $2N \times nD$  的分区类型 1334、尺寸为  $nL \times 2N$  的分区类型 1336 和尺寸为  $nR \times 2N$  的分区类型 1338。

[0167] 当分区类型被设置为对称（即，分区类型 1322、1324、1326 或 1328）时，如果变换单元的划分信息（TU 尺寸标记）为 0，则设置尺寸为  $2N \times 2N$  的变换单元 1342，如果 TU 尺寸标记为 1，则设置尺寸为  $N \times N$  的变换单元 1344。

[0168] 当分区类型被设置为非对称（即，分区类型 1332、1334、1336 或 1338）时，如果 TU 尺寸标记为 0，则设置尺寸为  $2N \times 2N$  的变换单元 1352，如果 TU 尺寸标记为 1，则设置尺寸为  $N/2 \times N/2$  的变换单元 1354。

[0169] TU 尺寸标记可以是一种变换索引的类型；与变换索引相应的变换单元的尺寸可根据编码单元的预测单元类型或编码单元的分区类型而修改。

[0170] 当分区类型被设置为对称（即，分区类型 1322、1324、1326 或 1328）时，如果变换单元的 TU 尺寸标记为 0，则设置尺寸为  $2N \times 2N$  的变换单元 1342，如果 TU 尺寸标记为 1，则设置尺寸为  $N \times N$  的变换单元 1344。

[0171] 当分区类型被设置为非对称（即，分区类型 1332 ( $2N \times nU$ )、1334 ( $2N \times nD$ )、1336 ( $nL \times 2N$ ) 或 1338 ( $nR \times 2N$ )) 时，如果 TU 尺寸标记为 0，则设置尺寸为  $2N \times 2N$  的变换单元 1352，如果 TU 尺寸标记为 1，则设置尺寸为  $N/2 \times N/2$  的变换单元 1354。

[0172] 参照图 9，以上描述的 TU 尺寸标记是具有值 0 或 1 的标记，但 TU 尺寸标记不限于 1 比特，并且在 TU 尺寸标记从 0 增加的同时，变换单元可被分层划分。变换单元划分信息（TU 尺寸标记）可被用作变换索引的示例。

[0173] 在这种情况下，当根据实施例的 TU 尺寸标记与变换单元的最大尺寸和最小尺寸一起被使用时，实际上已使用的变换单元的尺寸可被表示。视频编码设备 100 可对最大变换单元尺寸信息、最小变换单元尺寸信息和最大变换单元划分信息进行编码。编码的最大变换单元尺寸信息、最小变换单元尺寸信息和最大变换单元划分信息可被插入到序列参数集（SPS）中。视频解码设备 200 可使用最大变换单元尺寸信息、最小变换单元尺寸信息和最大变换单元划分信息以用于对视频进行解码。

[0174] 例如，(a) 如果当前编码单元的尺寸是  $64 \times 64$  并且最大变换单元是  $32 \times 32$ ，则 (a-1) 如果 TU 尺寸标记为 0，则变换单元的尺寸是  $32 \times 32$ ；(a-2) 如果 TU 尺寸标记为 1，则变换单元的尺寸是  $16 \times 16$ ，(a-3) 如果 TU 尺寸标记为 2，则变换单元的尺寸是  $8 \times 8$ 。

[0175] 可选择地，(b) 如果当前编码单元的尺寸是  $32 \times 32$  并且最小变换单元是  $32 \times 32$ ，则 (b-1) 如果 TU 尺寸标记为 0，则变换单元的尺寸是  $32 \times 32$ ，由于变换单元的尺寸不能够小于  $32 \times 32$ ，因此 TU 尺寸标记不再被设置。

[0176] 可选择地，(c) 如果当前编码单元的尺寸是  $64 \times 64$  并且最大 TU 尺寸标记为 1，则 TU 尺寸标记可以是 0 或 1，并且不能设置其它 TU 尺寸标记。

[0177] 因此，当将最大 TU 尺寸标记定义为“MaxTransformSizeIndex”，将最小 TU 尺寸定义为“MinTransformSize”，并且将在 TU 尺寸标记为 0 的情况下变换单元（即，基本变换单元 RootTu）定义为“RootTuSize”时，可通过下面的等式 (1) 来定义可在当前编码单元中可用的最小变换单元的尺寸“CurrMinTuSize”。

[0178] 
$$\text{CurrMinTuSize} = \max(\text{MinTransformSize}, \text{RootTuSize} / (2 \wedge \text{MaxTransformSizeIndex})) \dots \dots (1)$$

[0179] 与在当前编码单元中可用的最小变换单元的尺寸“CurrMinTuSize”相比，作为在 TU 尺寸标记为 0 时的变换单元的尺寸的基本变换单元尺寸“RootTuSize”可指示可在系统中选择的最大变换单元尺寸。也就是说，根据等式 (1)，“RootTuSize /  $(2 \wedge \text{MaxTransformSizeIndex})$ ”是通过按照与最大变换单元划分信息相应的划分次数对作为在变换单元划分信息为 0 时的变换单元的尺寸的“RootTuSize”进行划分而获得的变换单元的尺寸，并且“MinTransformSize”是最小变换单元的尺寸，因此，“RootTuSize /  $(2 \wedge \text{MaxTransformSizeIndex})$ ”和“MinTransformSize”中较小的值可以是作为在当前编码单元中可用的最小变换单元的尺寸的“CurrMinTuSize”。

[0180] 根据本发明的实施例的基本变换单元的尺寸“RootTuSize”可根据预测模式而变化。

[0181] 例如,如果当前预测模式是帧间模式,则可根据以下的等式(2)来确定RootTuSize。在等式(2)中,“MaxTransformSize”表示最大变换单元尺寸,“PUSize”指示当前预测单元尺寸。

$$[0182] \quad \text{RootTuSize} = \min(\text{MaxTransformSize}, \text{PUSize}) \cdots \cdots (2)$$

[0183] 换句话说,如果当前预测模式是帧间模式,则作为TU尺寸标记为0时的变换单元的基本变换单元的尺寸“RootTuSize”可被设置为最大变换单元尺寸和当前预测单元尺寸中较小的值。

[0184] 如果当前分区单元的预测模式是帧内模式,则可通过使用以下的等式(3)来确定“RootTuSize”。“PartitionSize”表示当前分区单元的尺寸。

$$[0185] \quad \text{RootTuSize} = \min(\text{MaxTransformSize}, \text{PartitionSize}) \cdots \cdots (3)$$

[0186] 换句话说,如果当前预测模式是帧内模式,则基本变换单元尺寸“RootTuSize”可被设置为最大变换单元尺寸和当前分区单元尺寸中较小的值。

[0187] 然而,应注意基本变换单元的尺寸“RootTuSize”是示例,并且用于确定当前最大变换单元的因素不限于此,其中,基本变换单元的尺寸“RootTuSize”是根据本发明的实施例的当前最大变换单元尺寸并根据分区单元中的预测模式而变化。

[0188] 以下,将详细描述在图1的视频编码设备100的熵编码器120中的执行的语法元素的熵编码操作以及在图2的视频解码设备200的熵解码器220中执行的语法元素的熵解码操作。

[0189] 如上所述,视频编码设备100和视频解码设备200通过将最大编码单元划分为小于或等于最大编码单元的编码单元,来执行编码和解码。可独立于其它数据单元基于代价确定在预测和变换中使用的预测单元和变换单元。由于可通过对包括在最大编码单元中的具有分层树结构的每个编码单元递归地进行编码,来确定最佳编码单元,因此,可配置具有树结构的数据单元。换句话说,对于每个最大编码单元,可配置具有树结构的编码单元以及均具有树结构的预测单元和变换单元。为了进行解码,需要发送分层信息以及除分层信息以外的用于解码的非分层信息,其中,分层信息是指示具有分层结构的数据单元的结构信息的信息。

[0190] 与分层结构相关的信息是如以上参照图10至图12描述的确具有树结构的编码单元、具有树结构的预测单元和具有树结构的变换单元所需的信息,并包括最大编码单元的尺寸、编码深度、预测单元的分区信息、指示编码单元是否被划分的划分标记、变换单元的尺寸信息、以及指示变换单元是否被划分的变换单元划分标记(TU尺寸标记)。除分层结构信息以外的编码信息的示例包括应用于每个预测单元的帧内/帧间预测的预测模式信息、运动矢量信息、预测方向信息、在多个颜色分量被使用的情况下应用于每个数据单元的颜色分量信息、以及变换系数信息。以下,分层信息和额外的分层信息可被称为将被熵编码或熵解码的语法元素。

[0191] 具体地讲,根据本发明的实施例,提供了一种确定用于有效地对变换系数的等级(即,语法元素的尺寸信息)进行熵编码和熵解码的上下文模型的方法。以下,将详细描述确定用于对变换系数的等级进行熵编码和熵解码的上下文模型的方法。

[0192] 图14是示出根据本发明的实施例的对包括在变换单元中的变换系数信息进行熵编码和熵解码的操作的流程图。

[0193] 参照图 14, 在操作 1410, 首先对 coded\_block\_flag 进行熵编码或熵解码, 其中, coded\_block\_flag 指示在包括在当前变换单元中的变换系数之中是否存在非 0 的变换系数 (以下, 被称为“有效系数”)。

[0194] 如果 coded\_block\_flag 是 0, 则在当前变换单元中仅存在为 0 的变换系数, 因此, 仅值 0 被熵编码或熵解码为 coded\_block\_flag, 变换系数等级信息不被熵编码或熵解码。

[0195] 在操作 1420, 如果在当前变换单元中存在有效系数, 则对指示有效系数的位置的有效图 SigMap 进行熵编码或熵解码。

[0196] 有效图 SigMap 可由有效位以及指示最后一个有效系数的位置的预定信息形成。有效位指示根据每个扫描索引的变换系数是有效系数还是 0, 并可通过 significant\_coeff\_flag[i] 来表示。如稍后将描述, 以通过划分变换单元而获得的具有预定尺寸的子集为单位设置有效图。因此, significant\_coeff\_flag[i] 指示包括在变换单元中的子集中所包括的变换系数之中的第 i 扫描索引的变换系数是否为 0。

[0197] 根据传统的 H. 264, 额外地对指示每个有效系数是否是最后一个有效系数的标记 (End-Of-Block) 进行熵编码或熵解码。然而, 根据本发明的实施例, 对最后一个有效系数本身的位置信息进行熵编码或熵解码。如以上参照图 1 至图 13 所述, 根据本发明的实施例的变换单元的尺寸不限于  $4 \times 4$ , 并还可以是更大的尺寸 (诸如,  $8 \times 8$ 、 $16 \times 16$  或  $32 \times 32$ )。由于标记 (End-Of-Block) 的大小增加, 因此额外地对指示每个有效系数是否是最后一个有效系数的标记 (End-Of-Block) 进行熵编码或熵解码是没有效率的。因此, 根据本发明的实施例, 可对最后一个有效系数的位置信息本身进行熵编码或熵解码。例如, 如果最后一个有效系数的位置是 (x, y) (其中, x 和 y 是整数), 则可对作为指示坐标值 (x, y) 的语法元素的 last\_significant\_coeff\_x 和 last\_significant\_coeff\_y 进行熵编码或熵解码。

[0198] 在操作 1430, 对指示变换系数的大小的变换系数等级进行熵编码或熵解码。根据传统的 H. 264/AVC, 变换系数的等级信息由作为语法元素的 coeff\_abs\_level\_minus1 来表示。根据本发明的实施例, 变换系数的等级信息可通过 coeff\_abs\_level\_greater1\_flag、coeff\_abs\_level\_greater2\_flag 以及 coeff\_abs\_level\_remaining 来表示, 其中, coeff\_abs\_level\_greater1\_flag 是关于变换系数的绝对值是否大于 1 的语法元素, coeff\_abs\_level\_greater2\_flag 是关于变换系数的绝对值是否大于 2 的语法元素, coeff\_abs\_level\_remaining 指示剩余变换系数的大小信息。

[0199] 指示剩余的变换系数的大小信息的语法元素 coeff\_abs\_level\_remaining 具有在变换系数的大小 (absCoeff) 与基本等级值 baseLevel 之间的差, 其中, 通过使用 coeff\_abs\_level\_greater1\_flag 和 coeff\_abs\_level\_greater2\_flag 确定基本等级值 baseLevel。根据等式 ( $baseLevel = 1 + coeff\_abs\_level\_greather1\_flag + coeff\_abs\_level\_greather2\_flag$ ) 确定基本等级值, 并且根据等式 ( $coeff\_abs\_level\_remaining = absCoeff - baseLevel$ ) 确定 coeff\_abs\_level\_remaining。当 coeff\_abs\_level\_greater1\_flag 和 coeff\_abs\_level\_greater2\_flag 具有值 0 或 1 时, 基本等级值 baseLevel 可具有从 1 到 3 的值。因此, coeff\_abs\_level\_remaining 可从 (absCoeff-1) 改变到 (absCoeff-3)。如上所述, (absCoeff-baseLevel) 作为变换系数的大小信息被发送以便减小发送的数据的大小, 其中, (absCoeff-baseLevel) 是原始变换系数的大小 absCoeff 与基本等级值 baseLevel 之间的差。

[0200] 图 22 是示出根据本发明的实施例的熵编码设备 2200 的结构框图。图 22 的熵编码设备 2200 与图 1 的视频编码设备 100 的熵编码器 120 相应。

[0201] 参照图 22, 熵编码设备 2200 包括二值化器 2210、上下文建模器 2220 和二进制算术编码器 2230。此外, 二进制算术编码器 2230 包括常规编码引擎 2232 和旁路编码引擎 2234。

[0202] 当输入到熵编码设备 2100 的语法元素不是二进制值时, 二值化器 2210 对语法元素进行二值化从而输出有二进制值 0 或 1 构成的二进制位 (Bin) 串。二进制位表示由 0 或 1 构成的流的每个比特, 并通过上下文自适应二进制算术编码 (CABAC) 被编码。如果语法元素是包括相同频率的 0 和 1 的数据, 则该语法元素被输出到不使用概率的旁路编码引擎 2234 以被编码。

[0203] 具体地, 二值化器 2210 通过使用参数 (cRiceParam) 将 coeff\_abs\_level\_remaining 二值化到前缀比特串和后缀比特串, 其中, coeff\_abs\_level\_remaining 是指示变换系数的大小信息的语法元素。稍后将描述通过使用二值化器 2210 对作为指示变换系数的大小信息的语法元素的 coeff\_abs\_level\_remaining 进行二值化的操作。

[0204] 上下文建模器 2220 将用于对与语法元素相应的比特串进行编码的概率模型提供给常规编码引擎 2232。详细地, 上下文建模器 2220 将用于对当前语法元素的比特串的每个二进制值进行编码的二进制值的概率输出到二进制算术编码器 2230。

[0205] 上下文模型是二进制位的概率模型, 并包括关于 0 和 1 中的哪一个与最大可能的符号 (MPS) 和最小可能的符号 (LPS) 相应的信息以及关于 MPS 概率或 LPS 概率的信息。

[0206] 常规编码引擎 2232 基于由上下文建模器 2220 提供的关于 MPS 和 LPS 的信息以及 MPS 或 LPS 的概率信息执行关于与语法元素相应的比特串的二进制算术编码。

[0207] 可根据变换系数的二进制位索引预先设置在对作为指示变换系数的大小信息的语法元素的 coeff\_abs\_level\_remaining 进行编码时使用的上下文模型。

[0208] 图 23 是示出根据本发明的实施例的二值化设备 2300 的结构框图。图 23 的二值化设备 2300 与图 22 的二值化器 2210 相应。

[0209] 参照图 23, 二值化设备 2300 包括参数确定单元 2310 和比特串产生单元 2320。

[0210] 参数确定单元 2310 将在当前变换系数之前被编码的先前变换系数的大小与基于先前参数获得的预定临界值进行比较, 从而确定是否更新 (renew) 先前参数, 其中, 该先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的二值化中被使用。此外, 参数确定单元 2310 通过根据确定的结果更新或保持先前参数来获得将在对指示当前变换系数的大小的变换系数等级语法元素的二值化中使用的参数。

[0211] 详细地, 当先前变换系数的大小是 cLastAbsCoeff 并且先前参数是 cLastRiceParam 时, 参数确定单元 2310 基于以下算法确定参数 cLastRiceParam 和 coeff\_abs\_level\_remaining, 其中, cLastRiceParam 将在对变换系数等级语法元素的二值化中被使用, coeff\_abs\_level\_remaining 指示当前变换系数的大小。

[0212] 
$$cRiceParam = \text{Min}(cLastRiceParam + (cLastAbsLevel > (3 * (1 \ll cLastRiceParam))) ? 1 : 0), 4)$$

[0213] 该算法可通过下面的伪码来实现。

[0214] {

[0215] If  $cLastAbsCoeff > 3 * (1 \ll cLastRiceParam)$

[0216]  $cRiceParam = \text{Min}(cLastRiceParam + 1, 4)$

[0217] }

[0218] 如在以上算法中所述,参数确定单元 2310 将基于下面的等式获得的临界值  $th$  与  $cLastRiceParam$  进行比较: $th = 3 * (1 \ll cLastRiceParam)$ 。参数确定单元 2310 通过在  $cLastAbsCoeff$  大于临界值  $th$  时将先前参数 ( $cLastRiceParam$ ) 增加 1 来更新该先前参数,在  $cLastAbsCoeff$  不大于临界值  $th$  时保持该先前参数而不进行更新。

[0219] 初始参数被设置为 0。当参数 ( $cRiceParam$ ) 被更新时,与先前参数 ( $cLastRiceParam$ ) 相比,该参数被逐渐增加 +1。此外,在确定用于更新参数的条件时使用的临界值  $th$  根据参数 ( $cRiceParam$ ) 被确定,因此,随着参数 ( $cRiceParam$ ) 被更新,临界值  $th$  也被逐渐增加。也就是说,临界值  $th$  被设置为具有与先前参数 ( $cLastRiceParam$ ) 成比例的值,并且当先前参数 ( $cLastRiceParam$ ) 被更新时,与先前参数 ( $cLastRiceParam$ ) 相比,参数 ( $cRiceParam$ ) 具有逐渐增加 +1 的值。随着参数 ( $cRiceParam$ ) 在 0 到 4 的范围内被更新,临界值  $th$  逐渐增加到 3、6、12 和 24。

[0220] 比特串产生单元 2320 通过使用参数对变换系数的变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 进行二值化,并输出与变换系数的变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 相应的比特串。

[0221] 详细地,比特串产生单元 2320 通过使用获得的参数 ( $cRiceParam$ ) 根据下面的等式获得参数  $cTrMax$ :  $cTrMax = 4 \ll cRiceParam$ 。参数  $cTrMax$  用作用于将变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 划分为前缀和后缀的标准。

[0222] 比特串产生单元 2320 基于参数  $cTrMax$  对变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 的值进行划分,以便获得具有不超过参数  $cTrMax$  的值的前缀和指示超过参数  $cTrMax$  的部分的后缀。比特串产生单元 2320 根据下面的等式确定在不超过  $cTrMax$  的范围内的前缀:  $Prefix = \text{Min}(cTrMax, coeff\_abs\_level\_remaining)$ 。仅在变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 具有大于  $cTrMax$  的值时存在后缀。后缀是与 ( $coeff\_abs\_level\_remaining - cTrMax$ ) 相应的值。当变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 不超过  $cTrMax$  时,仅存在前缀。例如,当变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 是 10 并且参数  $cTrMax$  是 7 时,变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 被分类为值为 7 的前缀和值为 3 的后缀。可选择地,当变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 是 6 并且参数  $cTrMax$  是 7 时,变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 被分类为值为 6 的前缀并不具有后缀。

[0223] 当通过基于参数  $cTrMax$  对变换系数等级语法元素 ( $coeff\_abs\_level\_remaining$ ) 的值进行划分而确定了前缀和后缀时,比特串产生单元 2320 通过使用预先设置的预定二值化方法来对前缀和后缀进行二值化,以输出与前缀和后缀相应的比特串。例如,比特串产生单元 2320 可通过使用截断一元二值化方法来对具有与  $\text{Min}(cTrMax, coeff\_abs\_level\_remaining)$  相应的值的前缀进行二值化来输出比特串,并可通过使用  $k$  阶指数哥伦布 (exponential Golomb) 方法对具有与 ( $coeff\_abs\_level\_remaining - cTrMax$ ) 相应的值的后缀进行二值化来输出比特串。可通过使用通过使用参数确定单元 2310 确定的参数 ( $cRiceParam$ ) 来确定值  $k$ 。例如,值  $k$  可具有值 ( $cRiceParam + 1$ )。



[0224] 根据截断一元二值化方法,如下面的表 2 所示,可对值为  $\text{Min}(\text{cTrMax}, \text{coeff\_abs\_level\_remaining})$  的前缀进行二值化。

[0225] [表 2]

[0226]

$\text{Min}(\text{cTrMax}, \text{coeff\_abs\_level\_remaining})$	输出比特串
0	0
1	10
2	110
3	1110
.....	.....

[0227]

[0228] 比特串产生单元 2320 可根据参数 (cRiceParam) 并通过参考预先设置的表来产生与前缀和后缀相应的比特串。根据查找表方法,预先设置的表可以以如下方式被设置:随着参数 (cRiceParam) 的值增加,与相对大的值相应的比特串的长度被减小。

[0229] 将参照图 15 至图 21 详细描述根据本发明的实施例的对与变换单元相关的语法元素进行熵编码的操作。

[0230] 图 15 示出根据本发明的实施例的熵编码后的变换单元 1500。虽然在图 15 中示出了尺寸为  $16 \times 16$  的变换单元 1500,但变换单元 1500 的尺寸不限于示出的尺寸  $16 \times 16$ ,并还可以是范围在  $4 \times 4$  到  $32 \times 32$  内的各种尺寸。

[0231] 参照图 15,为了对包括在变换单元 1500 中的变换系数进行熵编码和熵解码,变换单元 1500 可被划分为更小的变换单元。以下,将描述对与包括在变换单元 1500 中的  $4 \times 4$  变换单元 1510 相关的语法元素进行熵编码的操作。对与  $4 \times 4$  变换单元 1510 相关的语法元素进行熵编码的该操作还可应用于不同尺寸的变换单元。

[0232] 包括在  $4 \times 4$  变换单元 1510 中的变换系数均具有如图 15 所示的变换系数 (absCoeff)。包括在  $4 \times 4$  变换单元 1510 中的变换系数可根据如图 15 所示的预定扫描顺序被串行化 (serialize) 并被顺序地处理。然而,扫描顺序不限于所示,并还可被修改。

[0233] 如上所述,与  $4 \times 4$  变换单元 1510 相关的语法元素的示例是 significant\_coeff\_flag、coeff\_abs\_level\_greater1\_flag、coeff\_abs\_level\_greater2\_flag、coeff\_abs\_level\_remaining,其中,significant\_coeff\_flag 是指示包括在变换单元中的每个变换系数是否是具有非 0 值的有效变换系数的语法,coeff\_abs\_level\_greater1\_flag 是指示变换系数的绝对值是否大于 1 的语法元素,coeff\_abs\_level\_greater2\_flag 是指示所述绝对值是否大于 2 的语法元素,coeff\_abs\_level\_remaining 是指示剩余变换系数的大小信息的语法元素。

[0234] 图 16 示出根据本发明的实施例的与图 15 的变换单元相应的有效图 SigMap 1600。

[0235] 参照图 15 和图 16,设置了有效图 SigMap 1600,其中,有效图 SigMap 1600 对于包

括在图 15 的  $4 \times 4$  变换单元 1510 中的变换系数之中的值不为 0 的每个有效变换系数,具有值 1。通过使用先前设置的上下文模型对有效图 SigMap1600 进行熵编码或熵解码。

[0236] 图 17 示出与图 15 的  $4 \times 4$  变换单元 1510 相应的 coeff\_abs\_level\_greater1\_flag 1700。

[0237] 参照图 15 至图 17, 设置了 coeff\_abs\_level\_greater1\_flag 1700, 其中, coeff\_abs\_level\_greater1\_flag 1700 是关于有效图 SigMap 1600 具有值 1 的有效变换系数, 指示相应有效变换系数是否具有大于 1 的值的标记。当 coeff\_abs\_level\_greater1\_flag 1700 为 1 时, 这指示相应变换系数是值大于 1 的变换系数, 当 coeff\_abs\_level\_greater1\_flag 1700 为 0 时, 这指示相应变换系数是值为 1 的变换系数。在图 17 中, 当 coeff\_abs\_level\_greater1\_flag 1710 在值为 1 的变换系数的位置处时, coeff\_abs\_level\_greater1\_flag 1710 的值为 0。

[0238] 图 18 示出与图 15 的  $4 \times 4$  变换单元 1510 相应的 coeff\_abs\_level\_greater2\_flag 1800。

[0239] 参照图 15 至图 18, 设置了 coeff\_abs\_level\_greater2\_flag 1800, 其中, coeff\_abs\_level\_greater2\_flag 1800 指示关于 coeff\_abs\_level\_greater1\_flag 1700 被设置为 1 的变换系数, 相应的变换系数是否具有大于 2 的值。当 coeff\_abs\_level\_greater2\_flag 1800 为 1 时, 这指示相应的变换系数是值大于 2 的变换系数, 当 coeff\_abs\_level\_greater2\_flag 1800 为 0 时, 这指示相应的变换系数是值为 2 的变换系数。在图 18 中, 当 coeff\_abs\_level\_greater2\_flag 1810 在值为 2 的变换系数的位置处时, coeff\_abs\_level\_greater2\_flag 1810 的值为 0。

[0240] 图 19 示出图 15 的与  $4 \times 4$  变换单元 1510 相应的 coeff\_abs\_level\_remaining1910。

[0241] 参照图 15 至图 19, 可通过计算每个变换系数的 (absCoeff-baseLevel) 来获得 coeff\_abs\_level\_remaining 1900, 其中, coeff\_abs\_level\_remaining 1900 是指示剩余变换系数的大小信息的语法元素。

[0242] 如上所述, 作为指示剩余变换系数的大小信息的语法元素的 coeff\_abs\_level\_remaining 1900 具有变换系数的大小 (absCoeff) 与基本等级值 baseLevel 之间的差, 其中, 基本等级值 baseLevel 是通过使用 coeff\_abs\_level\_greater1\_flag 和 coeff\_abs\_level\_greater2\_flag 来确定的。根据等式 ( $\text{baseLevel} = 1 + \text{coeff\_abs\_level\_greater1\_flag} + \text{coeff\_abs\_level\_greater2\_flag}$ ) 确定基本等级值 baseLevel, 并且根据等式 ( $\text{coeff\_abs\_level\_remaining} = \text{absCoeff} - \text{baseLevel}$ ) 确定 coeff\_abs\_level\_remaining。

[0243] 参数确定单元 2310 根据示出的扫描顺序读取 coeff\_abs\_level\_remaining1900, 以获得变换系数的大小, 诸如, “0312333445588”。

[0244] 参数确定单元 2310 根据扫描顺序来顺序地确定在对每个变换系数的大小信息的二值化中使用的参数 (cRiceParam)。首先, 初始参数 (cRiceParam) 被设置为 0。根据上述算法, 仅在满足条件  $\text{cLastAbsCoeff} > 3 * (1 << \text{cLastRiceParam})$  时, 所述参数增加。初始设置的参数 (cRiceParam) 为 0, 并且初始设置的参数 (cRiceParam) 保持该值直到先前变换系数 (cLastAbsCoeff) 的大小的值为  $3 * (1 << 0)$  (即, 大于 3 的值)。参照图 19, 变换系数的大小

“12”(1920) 大于 3, 因此, 当在变换系数“12”(1920) 之后的变换系数的大小被二值化时, 具有从 0 更新到 1 的值的参数 (cRiceParam) 被使用。当参数 (cRiceParam) 被更新为 1 时, 仅在满足  $cLastAbsCoeff > 3 * (1 << 1)$  (即,  $cLastAbsCoeff > 6$ ) 的条件时参数 (cRiceParam) 再次被更新。参照图 19, 作为倒数第二个变换系数的大小的“8”(1930) 大于 6, 因此, 参数 (cRiceParam) 从 1 被更新到 2。

[0245] 图 20 示出显示与图 15 至图 19 中示出的变换单元 1510、1600、1700、1800 和 1900 相关的语法元素的表。在图 20 中, GTR1 表示 `coeff_abs_level_greater1_flag`, GTR2 表示 `coeff_abs_level_greater2_flag` 并且 Remaining 表示 `coeff_abs_level_remaining`。参照图 20, 指示变换系数等级、`coeff_abs_level_remaining` 的语法元素不是二进制值, 因此通过使用参数进行二值化。

[0246] 图 21 示出根据本发明的实施例被二值化的 `coeff_abs_level_remaining` 的另一示例。

[0247] 如上所述, 初始参数 (cRiceParam) 被设置为 0, 并且初始参数 (cRiceParam) 仅在满足  $cLastAbsCoeff > 3 * (1 << cLastRiceParam)$  的条件时增加 +1。初始参数 (cRiceParam) 的值为  $\text{Min}(cLastRiceParam + 1, 4)$ , 因此更新后的参数不会具有超过 4 的值。根据在对先前变换系数的大小进行二值化时使用的先前参数 (cRiceParam), 在确定是否更新参数时使用的临界值  $3 * (1 << cLastRiceParam)$  的值为  $3 * (1 << 0)$ 、 $3 * (1 << 1)$ 、 $3 * (1 << 2)$  或  $3 * (1 << 3)$ 。因此, 在对值大于 3 的变换系数进行处理之后, 参数 (cRiceParam) 增加 +1, 随后, 在对大于 6 的变换系数进行处理之后, 参数 (cRiceParam) 增加 +1, 随后, 在对大于 12 的变换系数进行处理之后, 参数 (cRiceParam) 增加 +1, 最后, 在对大于 24 的变换系数进行处理之后, 参数 (cRiceParam) 增加 +1。也就是说, 在变换系数之后存在值的骤变时, 参数 (cRiceParam) 也逐渐增加 +1。

[0248] 参照图 21, 在对值为 12 并第一次大于 3 的变换系数 2110 进行处理之后, 初始设置的值为 0 的参数 (cRiceParam) 增加 +1。在值为 12 的变换系数 2110 之后, 更新后的参数 (cRiceParam) 被保持, 直到大于作为下一临界值的 6 的变换系数被处理。在值为 8 (大于作为下一临界值的 6) 的变换系数 2120 被处理之后, 参数 (cRiceParam) 增加 +1 从而值为 2。在值为 8 的变换系数 2120 之后, 更新后的参数 (cRiceParam) 被保持, 直到大于作为下一临界值的 12 的变换系数被处理。在值为 8 的变换系数 2120 之后, 在值为 13 (大于作为下一临界值的 12) 的变换系数 2130 被处理之后, 参数 (cRiceParam) 增加 +1 从而值为 3。在值为 13 的变换系数 2130 之后, 更新后的参数 (cRiceParam) 被保持, 直到大于作为下一临界值的 24 的变换系数被处理。在值为 13 的变换系数 2130 之后, 在值为 25 (大于作为下一临界值的 24) 的变换系数 2140 被处理之后, 参数 (cRiceParam) 增加 +1 从而值为 4。在值为 25 的变换系数 2140 之后的变换系数的二值化操作中, 由于参数 (cRiceParam) 已达到最大值 4, 因此参数 (cRiceParam) 4 被使用, 并且更新操作不再被执行。

[0249] 如上所述, 当在指示当前变换系数的大小的变换系数等级语法元素 `coeff_abs_level_remaining` 进行二值化时使用的参数 (cRiceParam) 通过使用参数确定单元 2310 被确定时, 比特串产生单元 2320 基于通过使用参数 (cRiceParam) 确定的参数 (cTrMax) 来将变换系数等级语法元素 `coeff_abs_level_remaining` 分类为前缀和后缀, 并通过针对前缀和后缀应用预先设置的二值化方法来对前缀和后缀进行二值化, 进而输出与 `coeff_abs_`

level\_remaining 相应的比特串。

[0250] 图 24 是示出根据本发明的实施例的指示变换系数等级的语法元素的熵编码方法的流程图。

[0251] 参照图 24, 在操作 2410, 参数确定单元 2310 根据预定扫描顺序获得指示包括在变换单元中的变换系数的大小的变换系数等级语法元素 (coeff\_abs\_level\_remaining)。

[0252] 在操作 2420, 参数确定单元 2310 将在当前变换系数之前被编码的先前变换系数 (cLastAbsCoeff) 的大小与基于先前参数 (cLastRiceParam) 获得的预定临界值进行比较, 进而确定是否更新先前参数 (cLastRiceParam), 其中, 先前参数 (cLastRiceParam) 在对指示先前变换系数 (cLastAbsCoeff) 的大小的先前变换系数等级语法元素的二值化中被使用。

[0253] 在操作 2430, 参数确定单元 2310 基于操作 2420 的确定的结果更新或保持先前参数, 进而获得在对指示当前变换系数的大小的变换系数等级语法元素的二值化中使用的参数。如上所述, 参数确定单元 2310 将基于  $th = 3 * (1 << cLastRiceParam)$  获得的临界值  $th$  与先前参数  $cLastRiceParam$  进行比较; 当  $cLastAbsCoeff$  大于  $th$  时, 参数确定单元 2310 通过将先前参数增加 1 来更新先前参数; 当  $cLastAbsCoeff$  不大于  $th$  时, 参数确定单元 2310 不更新而保持先前参数。当先前参数被更新时, 更新后的参数逐渐增加 +1。

[0254] 在操作 2440, 比特串产生单元 2320 通过使用获得的参数 (cRiceParam) 对变换系数等级语法元素 (coeff\_abs\_level\_remaining) 进行二值化, 进而输出与当前变换系数的变换系数等级语法元素 (coeff\_abs\_level\_remaining) 相应的比特串。

[0255] 根据上述的根据本发明的实施例的对变换系数等级语法元素进行熵编码的操作, 即使在根据预定的扫描顺序处理的变换系数之中存在值突然变大的变换系数, 参数也不必突然地修改而可逐渐地增加 +1。

[0256] 同时, 还可将上述的根据本发明的实施例的更新参数以便对变换系数等级语法元素进行熵编码的操作应用到对除变换系数等级语法元素以外的语法元素的二值化中。

[0257] 根据本发明的实施例的更新参数的操作可应用于在通过使用 Golomb-Rice 码对其它语法元素进行二值化时使用的参数。另外, 根据本发明的实施例的更新参数的方法可应用于更新在通过应用二值化方法 (诸如, 级联码二值化方法) 来对语法元素进行二值化时使用的参数。当级联码被使用时, 语法元素被分类为前缀和后缀, 根据本发明的实施例的更新参数的方法可应用于更新预定参数, 以用于确定前缀和后缀的目的。类似地, 根据本发明的实施例的更新参数的方法可应用于更新在通过如在低复杂度熵编码 (LCEC) 方法中使用定长码和变长码 (VLC) 表来对语法元素进行编码时使用的参数。

[0258] 图 25 是示出根据本发明的实施例的熵解码设备 2500 的框图。熵解码设备 2500 与图 2 的视频解码设备 200 的熵解码器 220 相应。熵解码设备 2500 执行上述的由熵编码设备 2000 执行的熵编码操作的相反操作。

[0259] 参照图 25, 熵解码设备 2500 包括上下文建模器 2510、常规解码引擎 2520、旁路解码引擎 2530 和反二值化器 2540。

[0260] 将通过使用旁路编码而被编码的语法元素输出到旁路解码器 2530 以解码, 并通过使用常规解码器 2520 对通过常规编码而被编码的语法元素进行解码。常规解码器 2520 基于通过使用上下文建模器 2510 而提供的上下文模型对当前语法元素的二进制值进行算

术解码,进而输出比特串。根据变换系数的二进制位索引可预先设置在对指示变换系数的大小信息的语法元素 `coeff_abs_level_remaining` 进行算术解码时使用的上下文模型。

[0261] 反二值化器 2540 再次将通过使用常规解码引擎 2520 或旁路解码引擎 2530 算术解码的比特串恢复为语法元素。

[0262] 熵解码设备 2500 除了对 `coeff_abs_level_remaining` 进行算术解码以外,还对与变换单元相关的语法元素(诸如, `SigMap`、`coeff_abs_level_greater1_flag` 或 `coeff_abs_level_greater2_flag`) 进行算术解码,并进行输出。当恢复了与变换单元相关的语法元素时,基于恢复的语法元素通过使用反量化、逆变换和预测解码来对包括在变换单元中的数据进行解码。

[0263] 图 26 是示出根据本发明的实施例的反二值化设备 2600 的结构的框图。图 26 的反二值化设备 2600 与图 25 的反二值化器 2540 相应。

[0264] 参照图 26,反二值化设备 2600 包括参数确定单元 2610 和语法元素恢复单元 2620。

[0265] 参数确定单元 2610 将在当前变换系数之前被解码的先前变换系数的大小与基于先前参数而获得的预定临界值进行比较,进而确定是否更新先前参数,其中,先前参数在对指示先前变换系数的大小的先前变换系数等级语法元素的反二值化中被使用。参数确定单元 2610 基于确定的结果更新或保持先前参数,从而获得在对指示当前变换系数的大小的变换系数等级语法元素进行反二值化时使用的参数。以与上述的图 23 的参数确定单元 2310 相同的方式,参数确定单元 2610 将基于等式 ( $th = 3 * (1 \ll cLastRiceParam)$ ) 获得的临界值 `th` 与先前参数 `cLastRiceParam`。当 `cLastAbsCoeff` 大于 `th` 时,参数确定单元 2610 通过将先前参数 (`cLastRiceParam`) 增加 1 来更新先前参数 (`cLastRiceParam`);当 `cLastAbsCoeff` 不大于 `th` 时,参数确定单元 2610 不更新而保持先前参数 (`cLastRiceParam`)。

[0266] 语法元素恢复单元 2620 通过使用获得的参数对与当前变换系数等级语法元素相应的比特串进行反二值化,进而恢复指示当前变换系数的大小的语法元素 (`coeff_abs_level_remaining`)。详细地,语法元素恢复单元 2620 将比特串分类为前缀比特串和后缀比特串,并通过使用截断一元二值化方法对前缀比特串进行反二值化并通过使用  $k$  阶 ( $k$  是 `cRiceParam+1`) 指数哥伦布方法对后缀比特串进行反二值化来恢复语法元素 (`coeff_abs_level_remaining`),其中,前缀比特串相应于通过使用截断一元二值化方法对与  $\text{Min}(cTrMax, \text{coeff\_abs\_level\_remaining})$  相应的值进行二值化而获得的比特串,后缀比特串相应于通过使用  $k$  阶指数哥伦布方法对与  $(\text{coeff\_abs\_level\_remaining} - cTrMax)$  相应的值进行二值化而获得的比特串。

[0267] 图 27 是示出根据本发明的实施例的对变换系数等级进行熵解码方法的流程图。

[0268] 参照图 27,在操作 2710,从比特流解析指示包括在变换单元中的变换系数的大小的变换系数等级语法元素。解析出的变换系数等级语法元素是均由 0 和 1 构成的比特串。

[0269] 在操作 2720,参数确定单元 2610 将在当前变换系数之前被恢复的先前变换系数 (`cLastAbsCoeff`) 的大小与基于先前参数 (`cLastRiceParam`) 获得的预定临界值进行比较,进而确定是否更新先前参数 (`cLastRiceParam`),其中,先前参数 (`cLastRiceParam`) 在对指示先前变换系数 (`cLastAbsCoeff`) 的大小的先前变换系数等级语法元素的二值化中被使用。

[0270] 在操作 2730, 参数确定单元 2610 基于确定的结果更新或保持先前参数 (cLastRiceParam), 进而获得在对指示当前变换系数的大小的变换系数等级语法元素 (coeff\_abs\_level\_remaining) 的二值化中使用的参数 (cRiceParam)。如上所述, 参数确定单元 2610 将基于等式 ( $th = 3 * (1 \ll cLastRiceParam)$ ) 获得的临界值 th 与先前参数 cLastRiceParam 进行比较。当 cLastAbsCoeff 大于 th 时, 参数确定单元 2610 通过将先前参数 (cLastRiceParam) 增加 1 来更新先前参数 (cLastRiceParam); 当 cLastAbsCoeff 不大于 th 时, 参数确定单元 2610 不更新而保持先前参数。当参数被更新时, 更新的参数逐渐增加 +1。

[0271] 在操作 2740, 语法元素恢复单元 2620 通过使用获得的参数对当前变换系数等级语法元素进行反二值化, 以获得当前变换系数的大小信息。如上所述, 由于  $coeff\_abs\_level\_remaining = absCoeff - baseLevel$ , 除  $coeff\_abs\_level\_remaining$  之外,  $coeff\_abs\_level\_greater1\_flag$  和  $coeff\_abs\_level\_greater2\_flag$  也被恢复, 当基本等级值 baseLevel 根据等式 ( $baseLevel = 1 + coeff\_abs\_level\_greater1\_flag + coeff\_abs\_level\_greater2\_flag$ ) 被确定时, 当前变换系数的大小可根据等式 ( $absCoeff = coeff\_abs\_level\_remaining + baseLevel$ ) 被确定。

[0272] 本发明还可被实施为计算机可读记录介质上的计算机可读代码。计算机可读记录介质是可存储可随后由计算机系统读取的数据的任何数据存储装置。计算机可读记录介质的示例包括只读存储器 (ROM)、随机存取存储器 (RAM)、CD-ROM、磁带、软盘、光学数据存储装置等。计算机可读记录介质还可分布于联网的计算机系统, 从而计算机可读代码以分布的方式被存储和执行。

[0273] 尽管已参照本发明的示例性实施例具体示出和描述了本发明, 但本领域的普通技术人员将理解, 在不脱离由权利要求限定的本发明的精神和范围的情况下可在形式和细节上进行各种改变。因此, 本发明的范围不由本发明的详细描述而由权利要求来限定, 在所述范围内的所有不同将被解释为包括在本发明中。

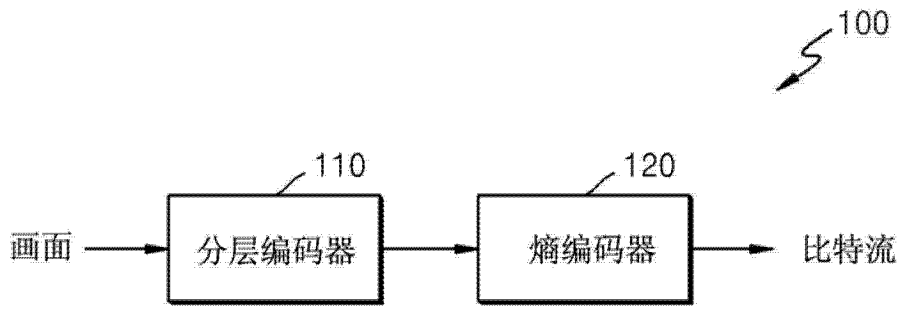


图 1

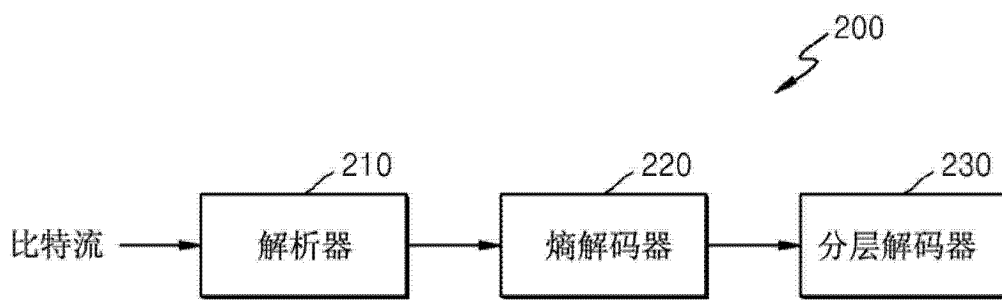


图 2

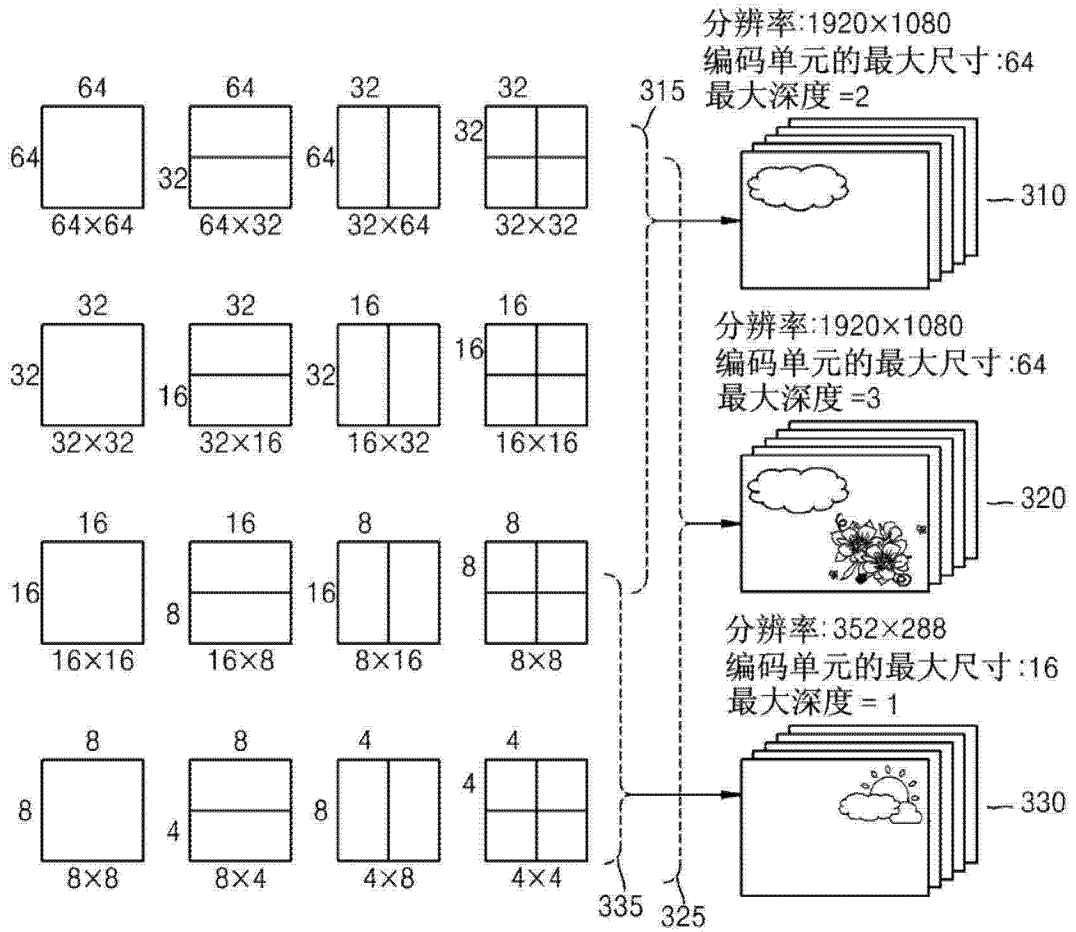


图 3



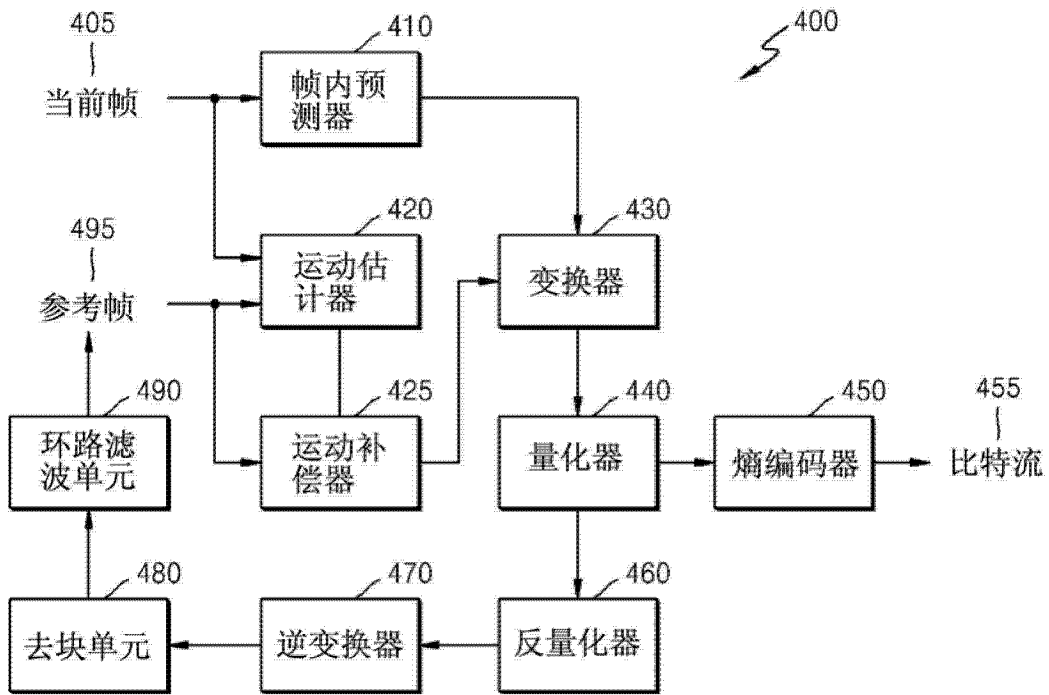


图 4

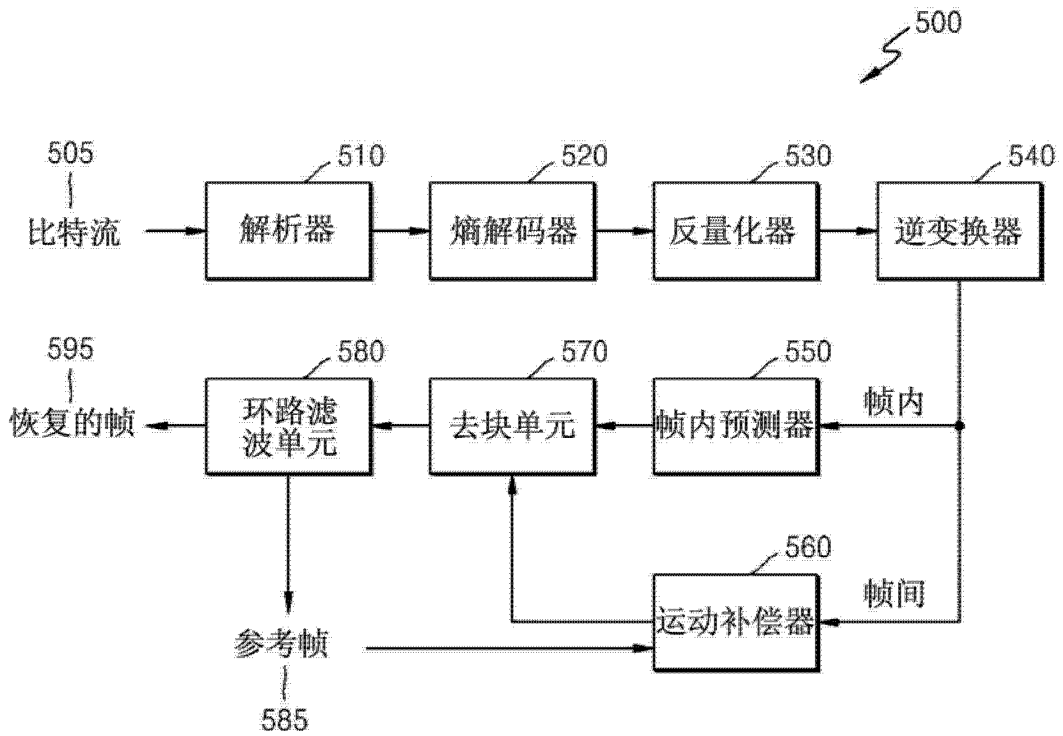


图 5

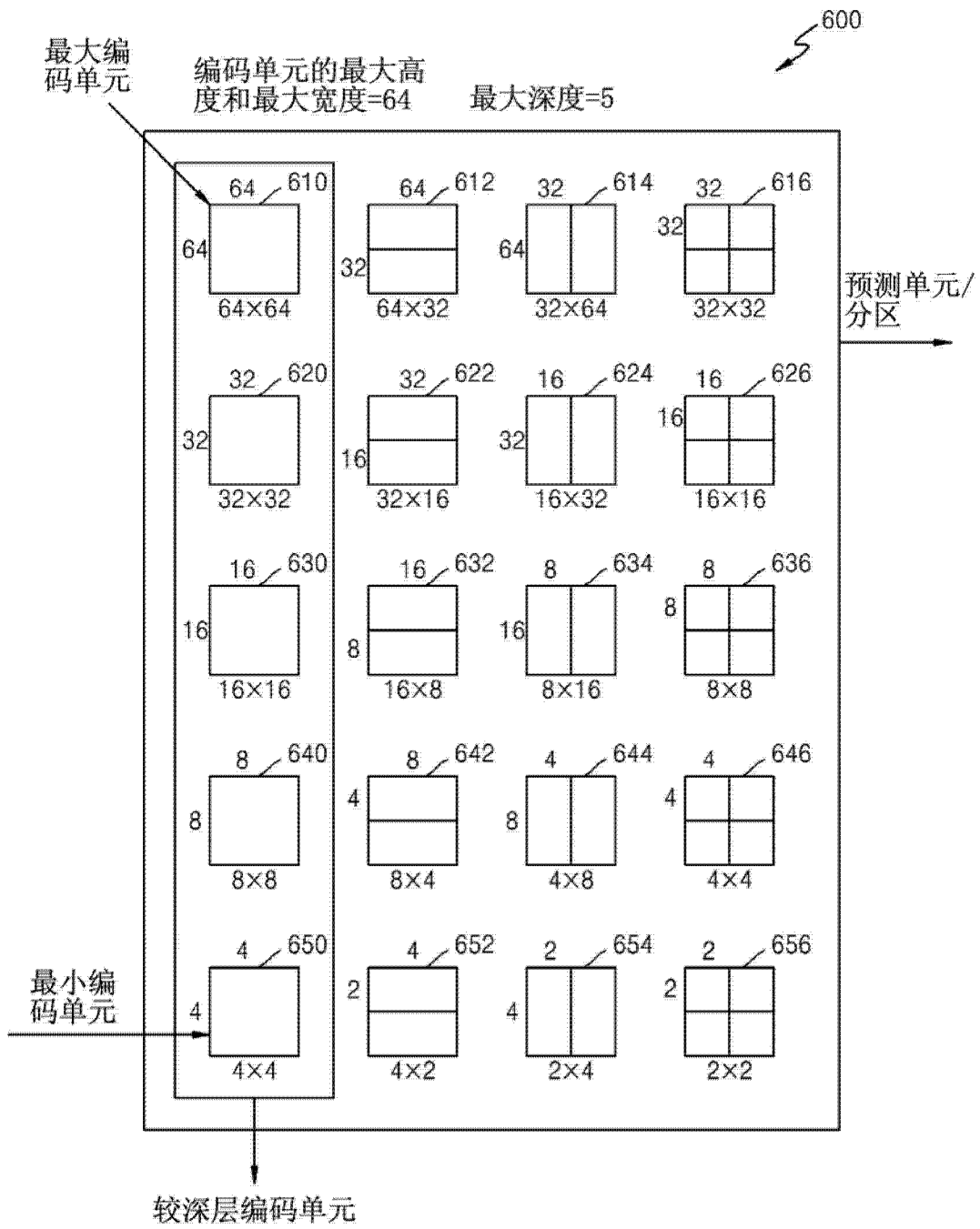


图 6

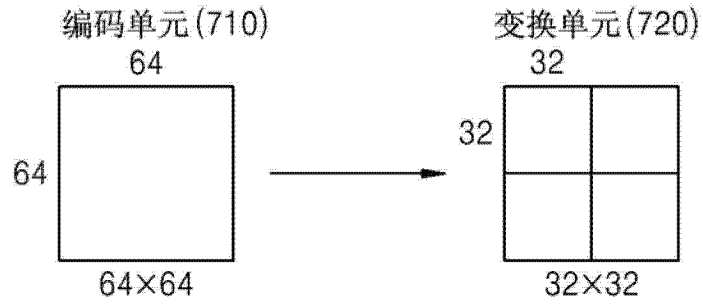
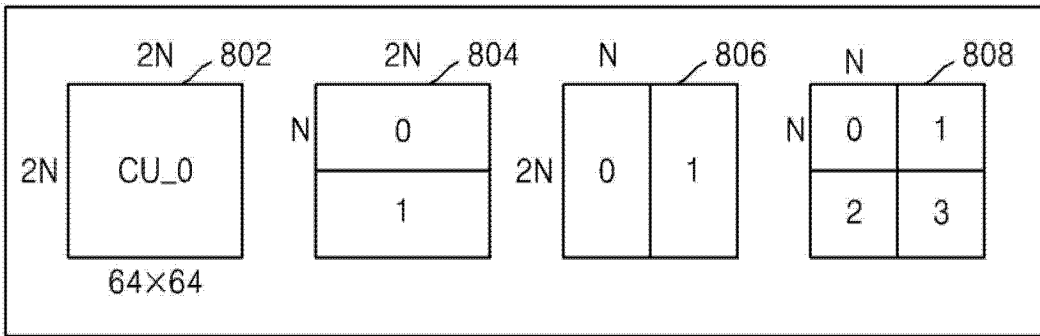
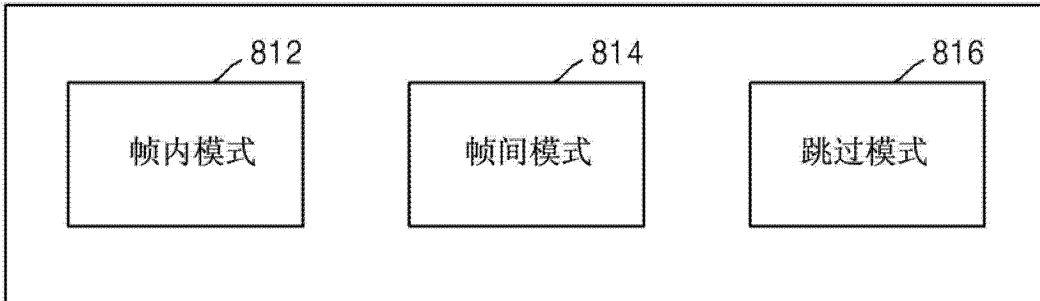


图 7

分区类型 (800)



预测模式 (810)



变换单元的尺寸 (820)

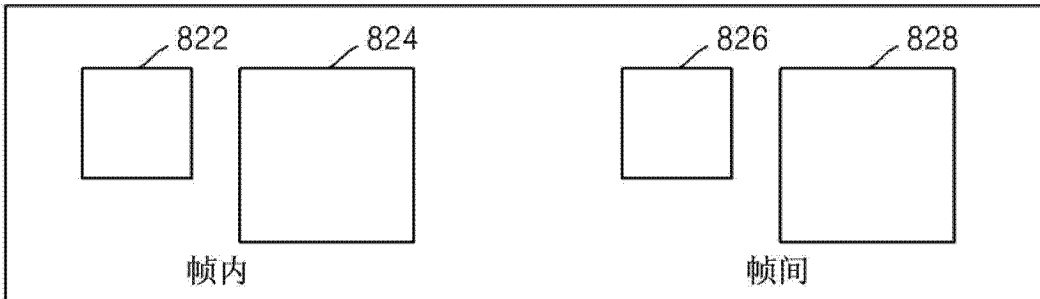


图 8

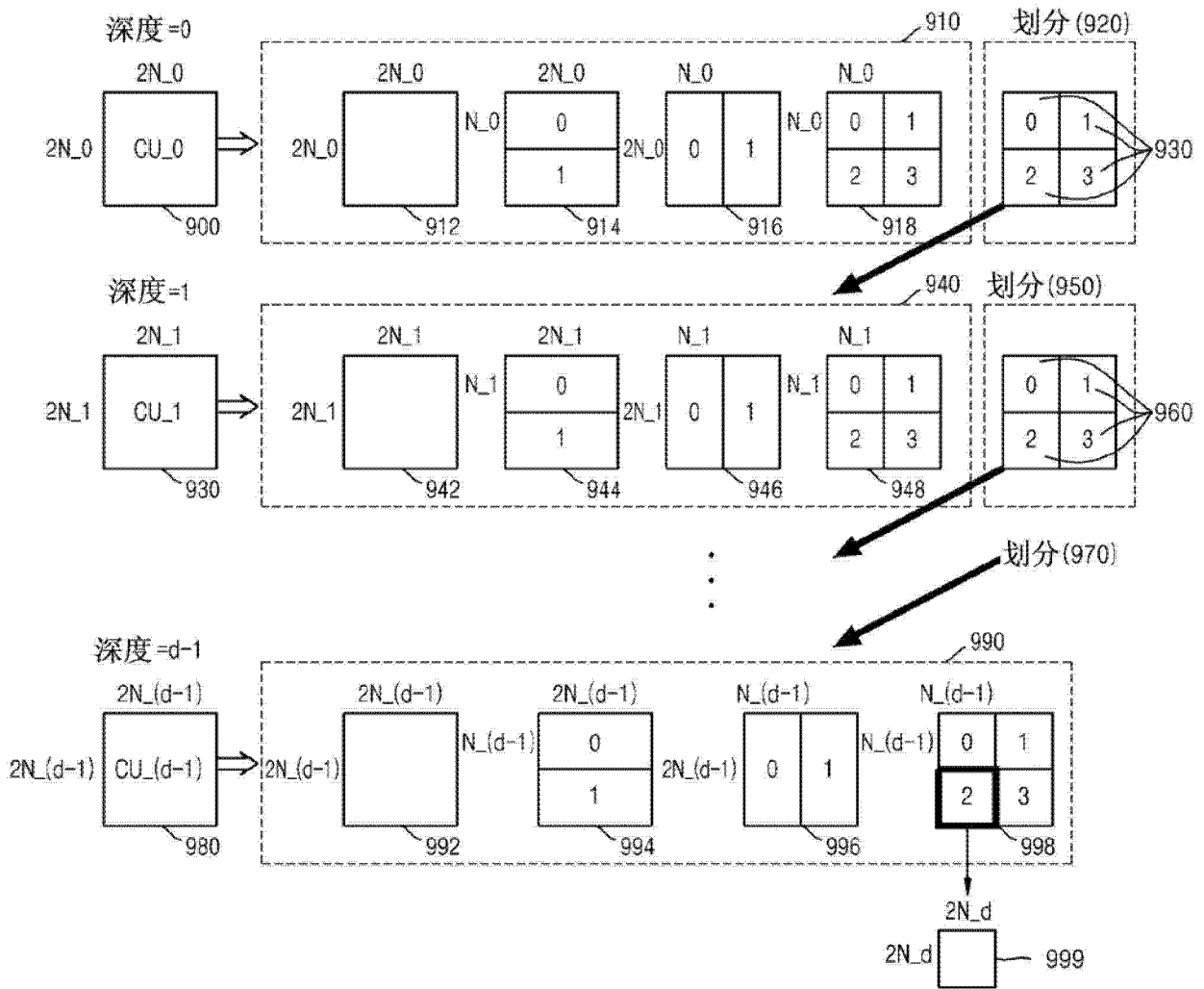
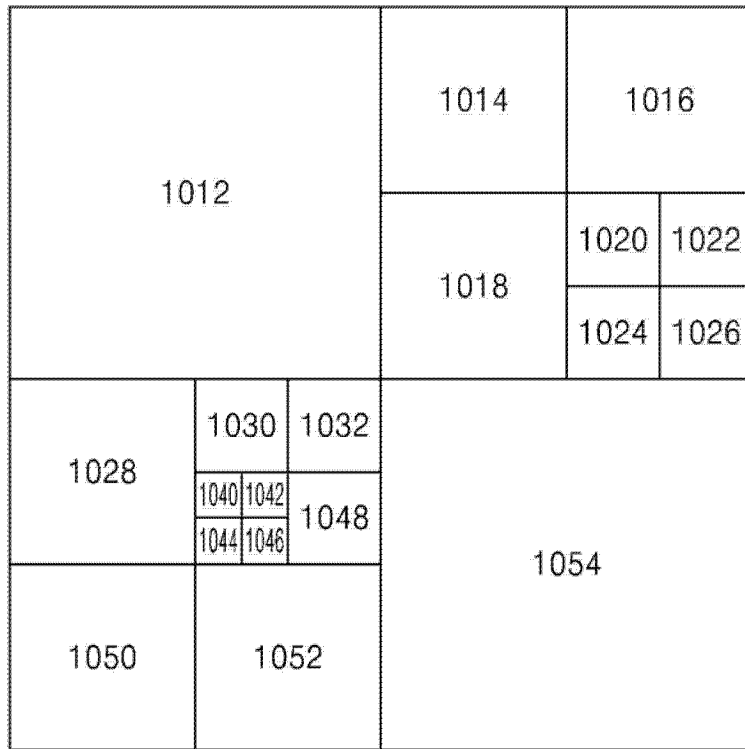


图 9



编码单元(1010)

图 10

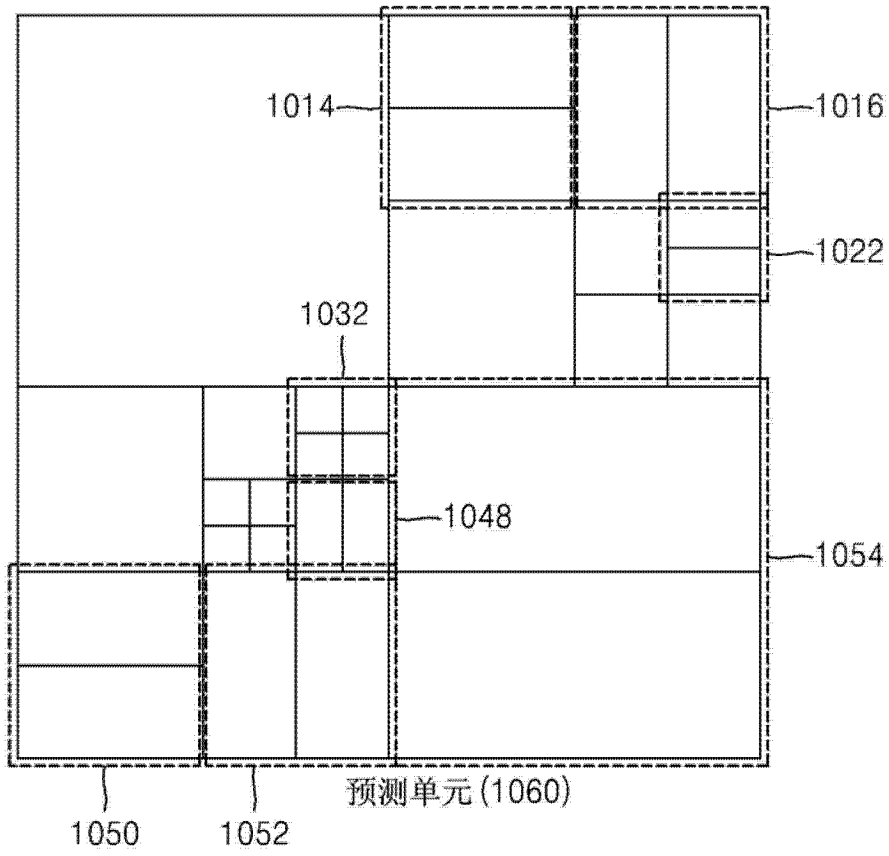


图 11

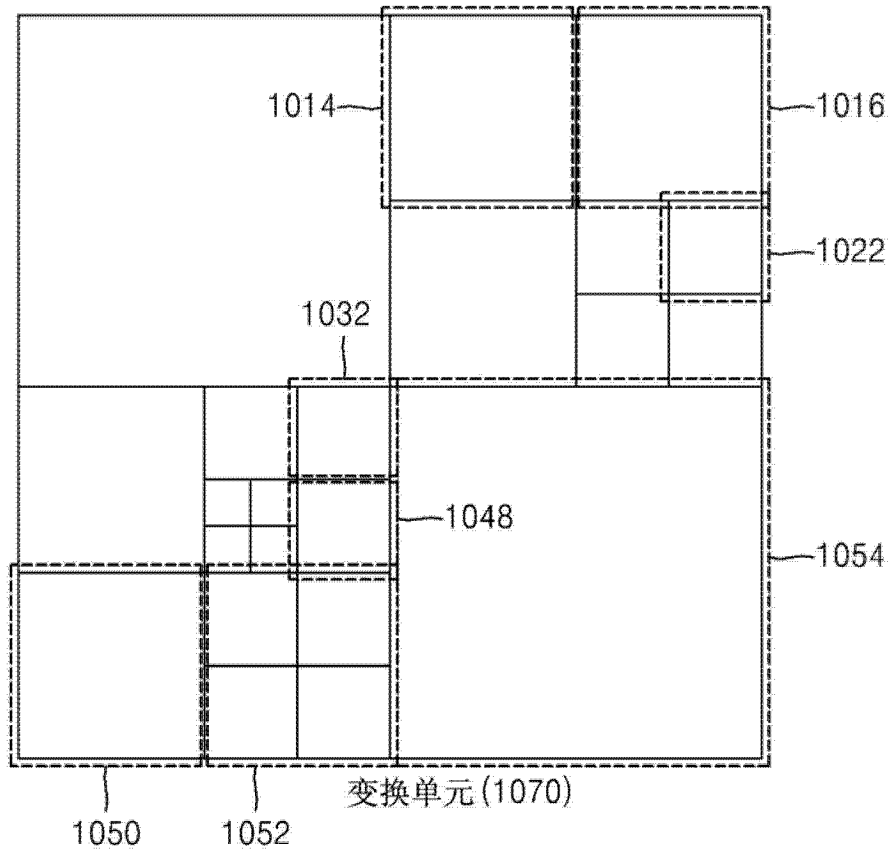


图 12

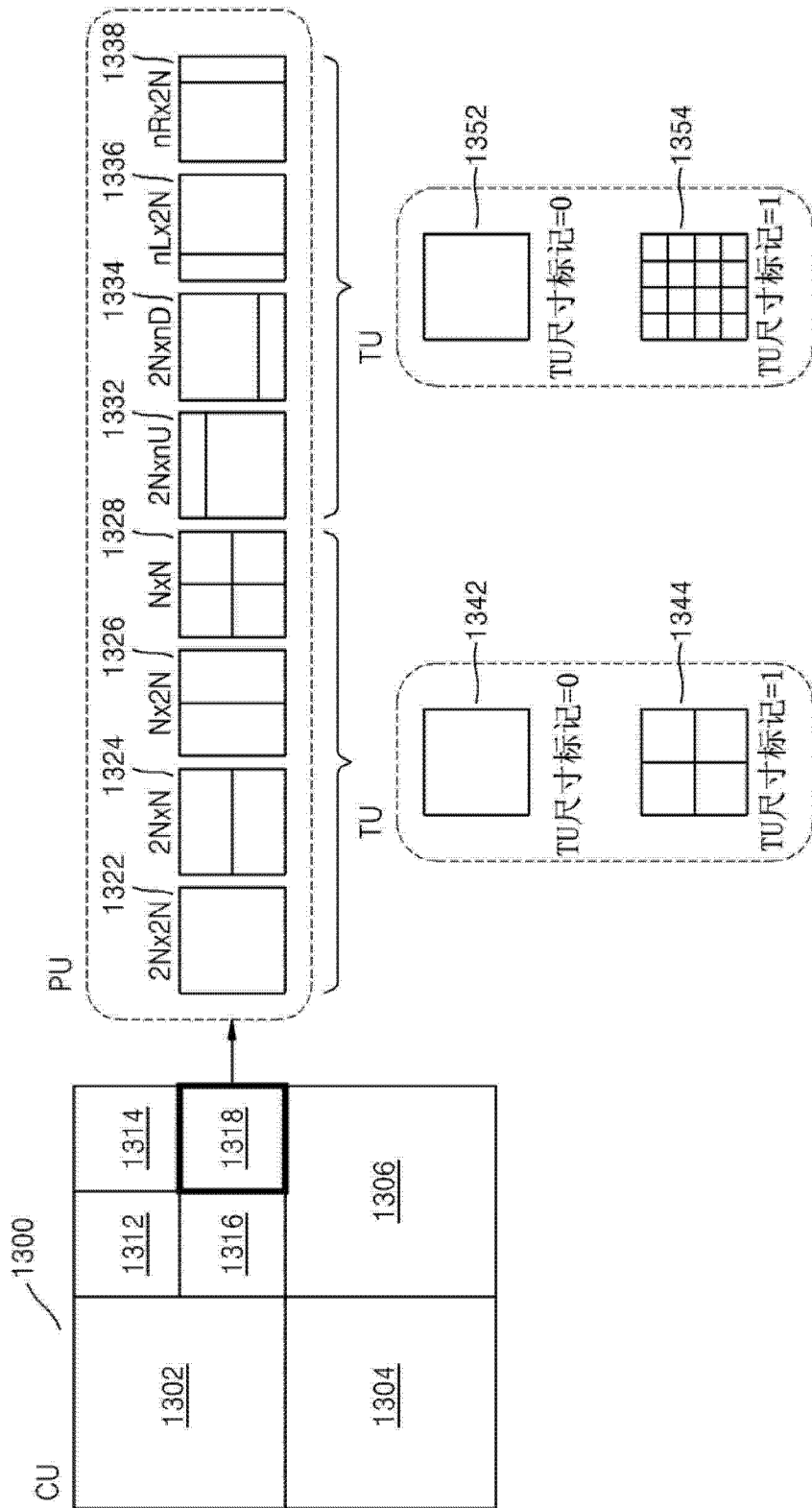


图 13



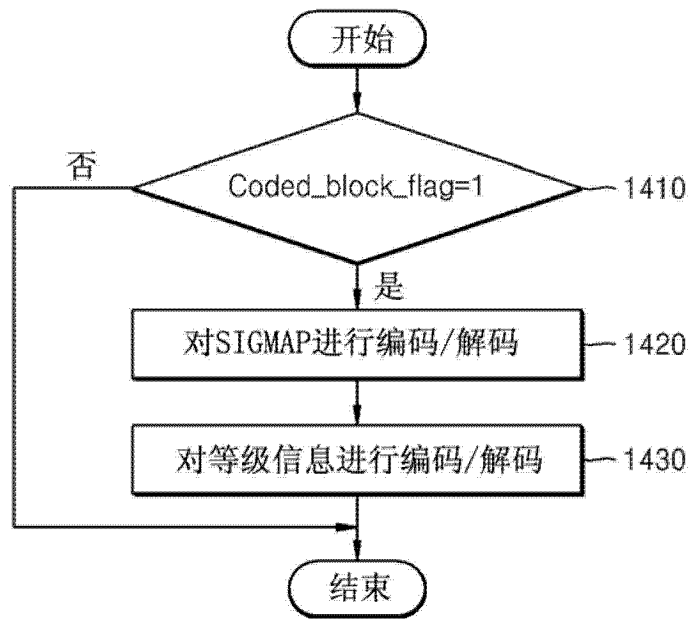


图 14

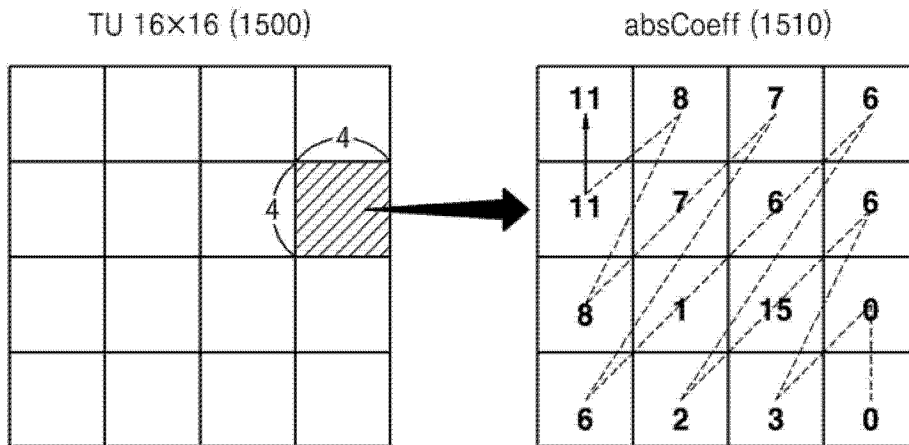


图 15

SigMap (1600)

1	1	1	1
1	1	1	1
1	1	1	0
1	1	1	0

图 16

coeff\_abs\_level\_greater1\_flag (1700)

1	1	1	1
1	1	1	1
1	0	1	
1	1	1	

1710

图 17

coeff\_abs\_level\_greater2\_flag (1800)

1	1	1	1
1	1	1	1
1		1	
1	0	1	

1810

图 18

coeff\_abs\_level\_remaining\_flag (1900)

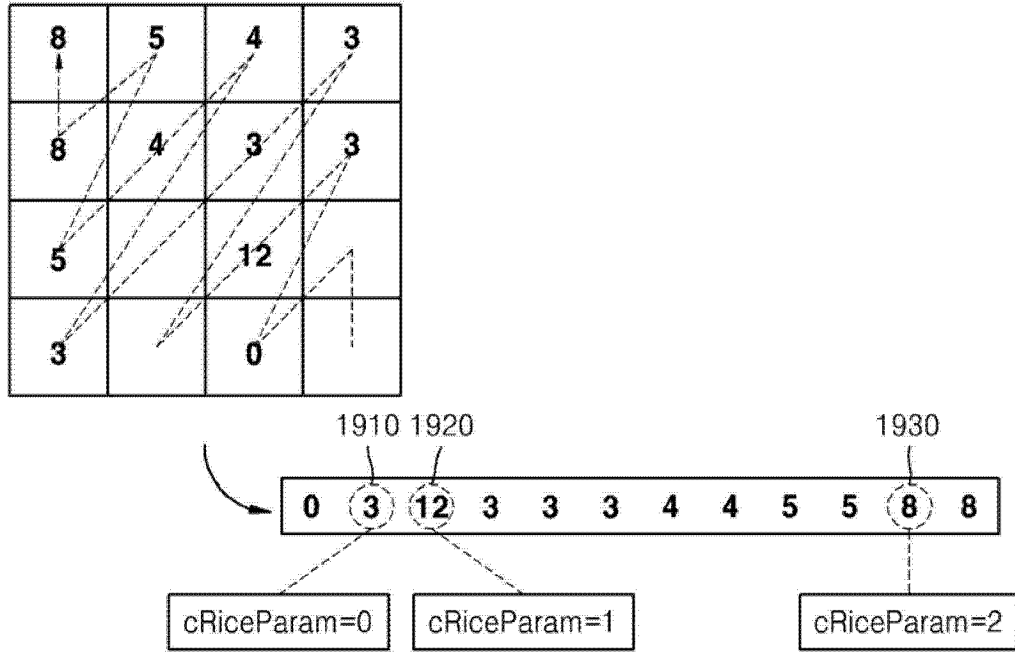


图 19

处理顺序 (扫描顺序)

absCoeff	0	0	3	6	15	2	6	6	1	6	7	7	8	8	11	11
SigMap	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
GTR1			1	1	1	1	1	1	0	1	1	1	1	1	1	1
GTR2			1	1	1	0	1	1		1	1	1	1	1	1	1
Remaining			0	3	12		3	3		3	4	4	5	5	8	8

图 20

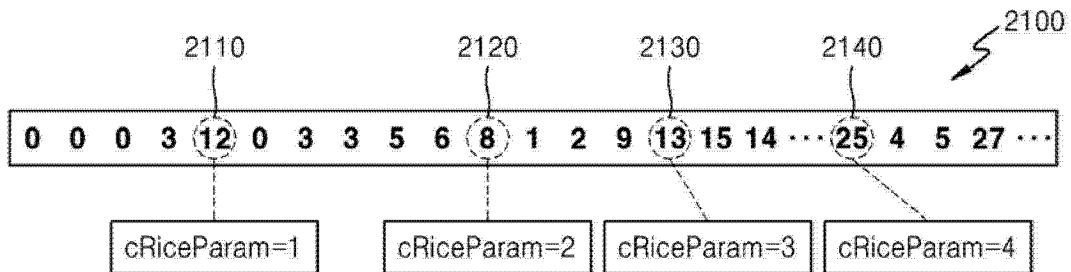


图 21

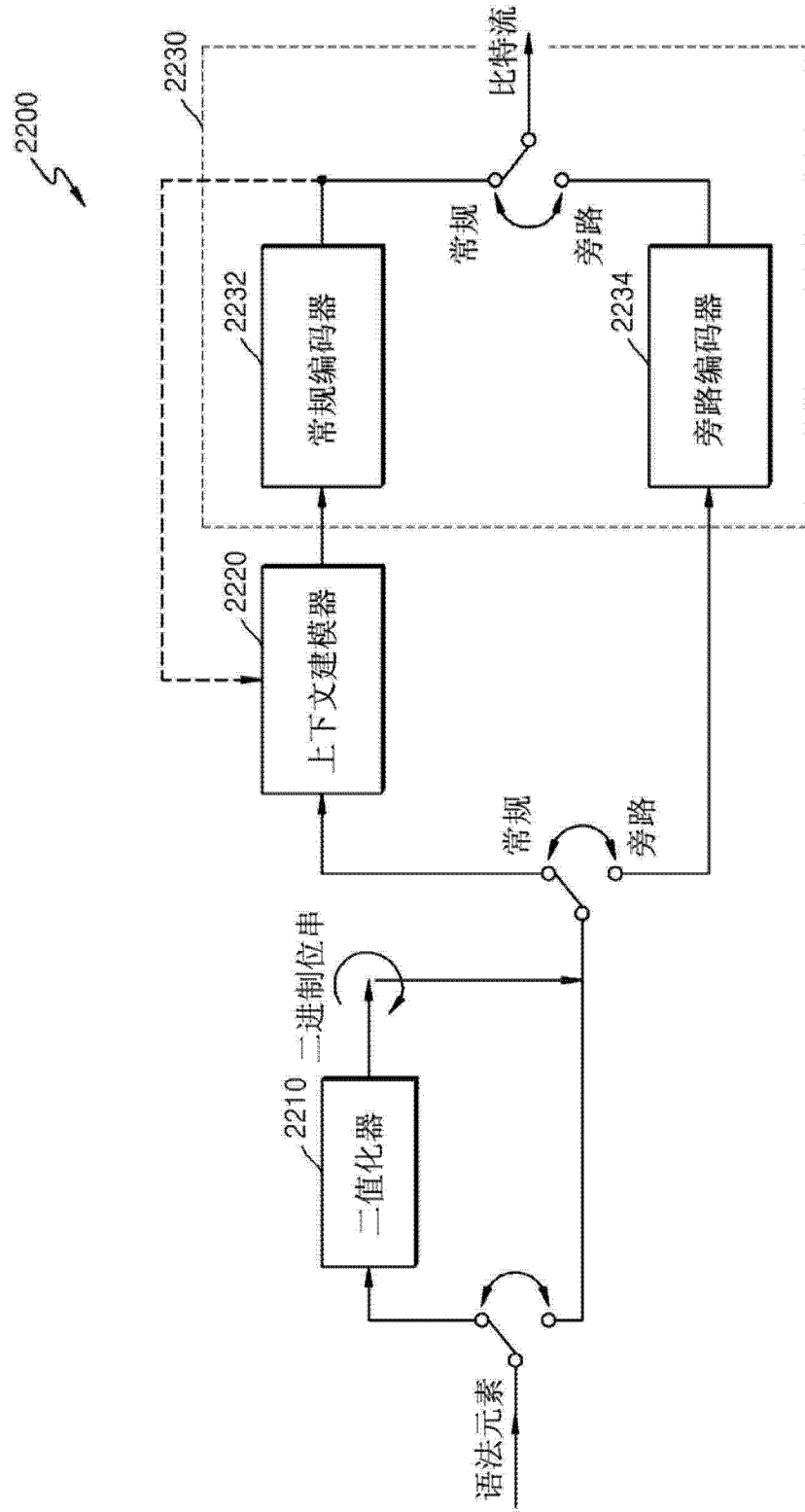


图 22

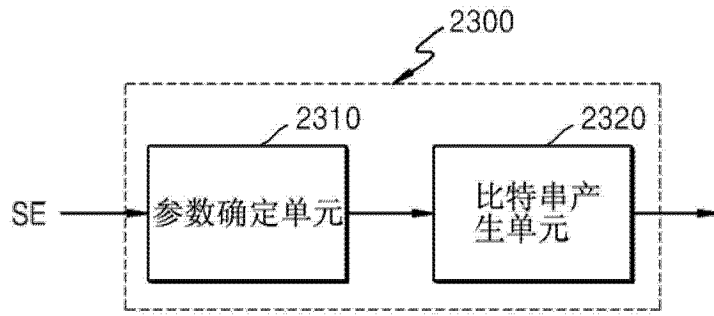


图 23

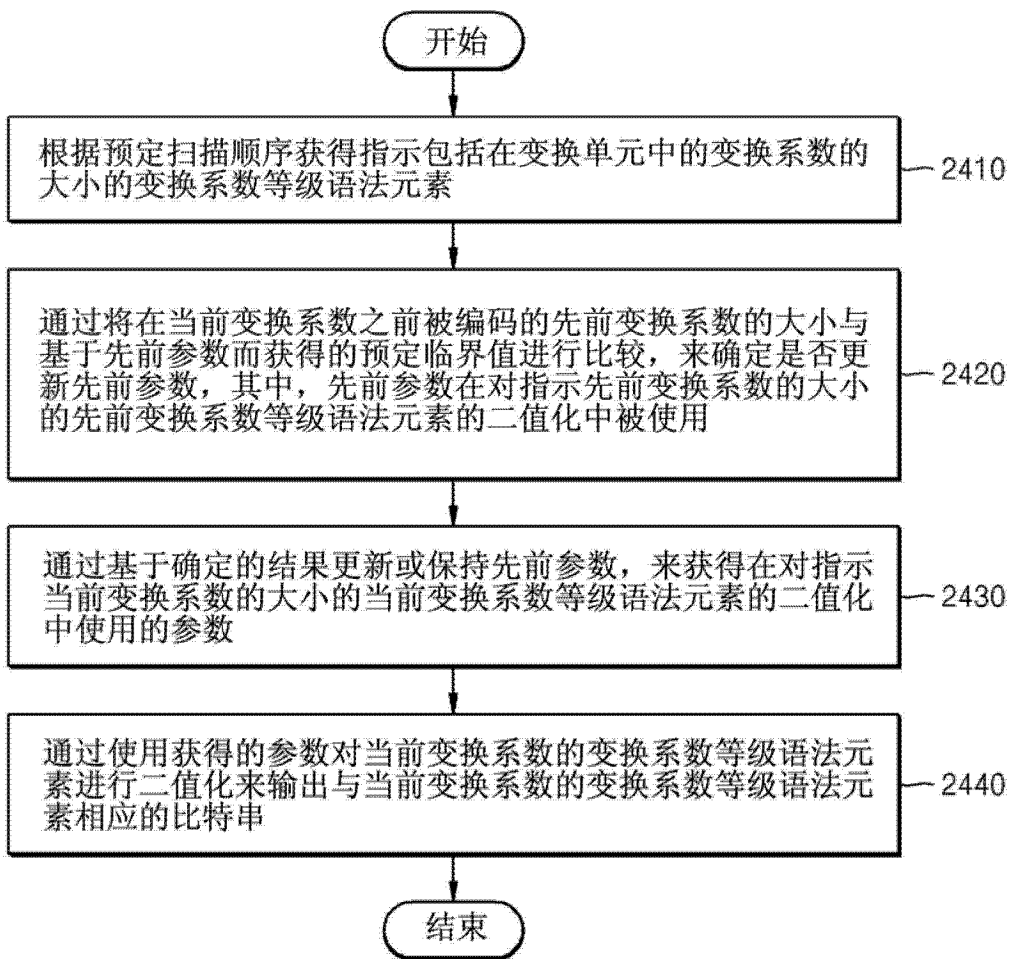


图 24

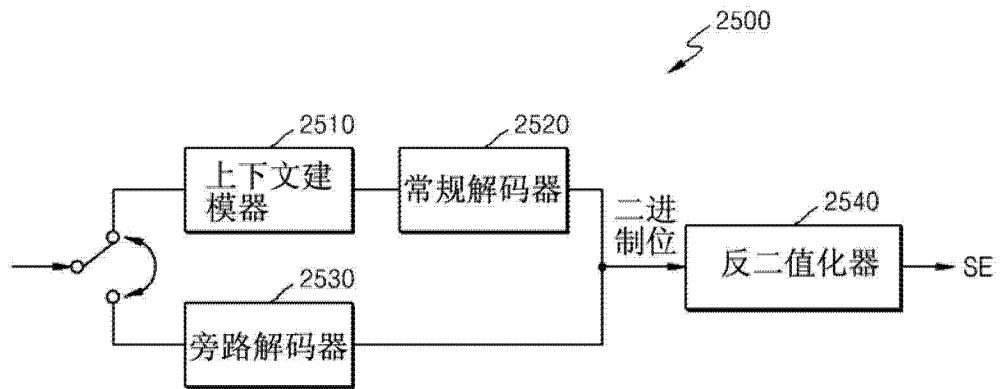


图 25

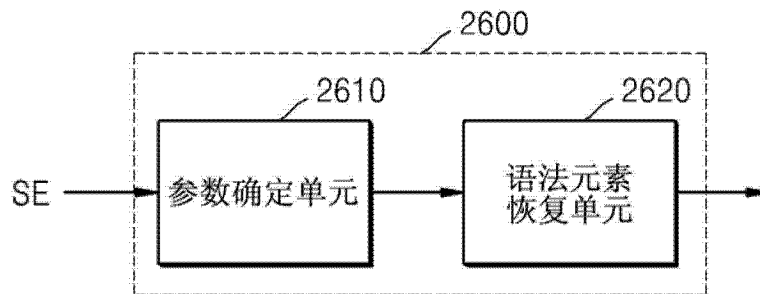


图 26

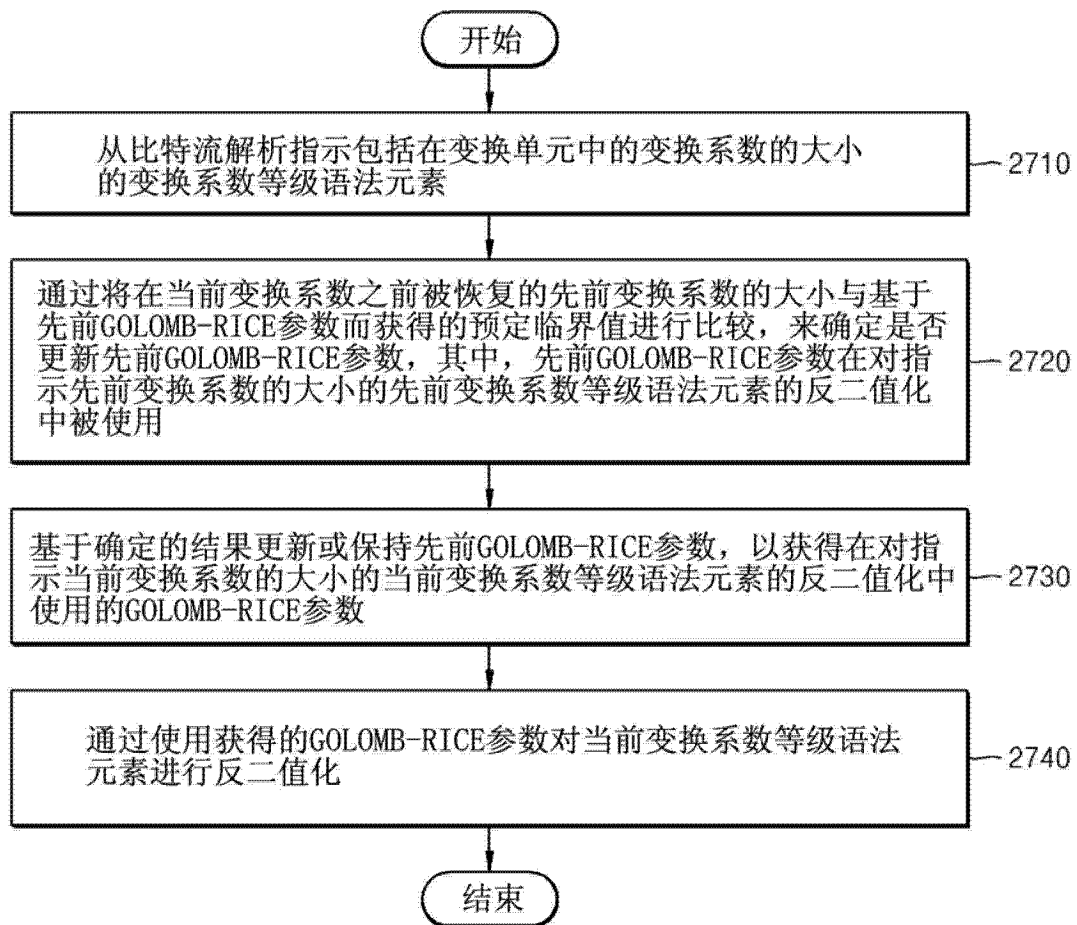


图 27