(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113170209 A
(43)申请公布日 2021.07.23

(21)申请号 201980076910.9

(22)申请日 2019.11.21

(30)优先权数据
10-2018-0144369 2018.11.21 KR
10-2019-0025071 2019.03.05 KR

(85)PCT国际申请进入国家阶段日
2021.05.21

(86)PCT国际申请的申请数据
PCT/KR2019/016059 2019.11.21

(87)PCT国际申请的公布数据
WO2020/106089 KO 2020.05.28

(71)申请人 韩国电子通信研究院
地址 韩国大田市
申请人 明达半导体股份有限公司

(72)发明人 林成昶 姜晶媛 李河贤 李镇浩 金晖容 金昊衍 郑旭帝 金贤奎

(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 于翔 曾世骁

(51)Int.Cl.
H04N 19/91(2006.01)
H04N 19/119(2006.01)
H04N 19/124(2006.01)
H04N 19/61(2006.01)
H04N 19/176(2006.01)
H04N 19/70(2006.01)
H04N 19/60(2006.01)

权利要求书2页 说明书36页 附图17页

(54)发明名称

图像编码/解码方法和设备以及存储比特流的记录介质

(57)摘要

本说明书公开了一种图像解码方法。根据本发明的图像解码方法包括以下步骤：确定是否对当前块的第一语法元素执行上下文更新，基于所述确定来更新用于对第一语法元素进行熵解码的上下文，以及基于更新的所述上下文来产生第一语法元素的二进制位，其中，基于用于当前块的已被解码的预定语法元素的数量来确定是否执行所述上下文更新。

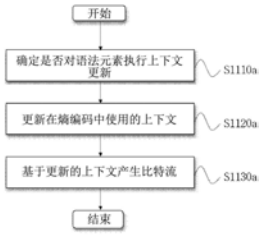

图 11a

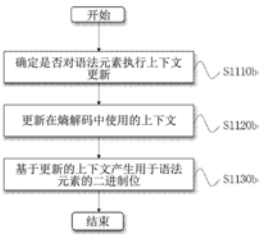

图 11b

CN 113170209 A

1.一种对图像进行解码的方法,所述方法包括:

确定是否对当前块的第一语法元素执行上下文更新;

基于所述确定来更新用于对第一语法元素进行熵解码的上下文;以及

基于更新的上下文来产生第一语法元素的二进制位,

其中,基于用于当前块的预解码的预定语法元素的数量来确定是否执行所述上下文更新。

2.如权利要求1所述的方法,其中,当预解码的所述预定语法元素的数量等于或小于预设值时,执行第一语法元素的所述上下文更新。

3.如权利要求2所述的方法,其中,所述预设值是基于当前块的尺寸被确定的。

4.如权利要求1所述的方法,其中,仅在当前块处于变换跳过模式时,执行所述上下文更新。

5.如权利要求1所述的方法,其中,第一语法元素是与当前块的残差信号相关的语法元素。

6.如权利要求5所述的方法,其中,第一语法元素是指示量化等级的符号的语法元素。

7.如权利要求6所述的方法,其中,第一语法元素是coeff_sign_flag。

8.如权利要求1所述的方法,其中,所述预定语法元素是残差信号相关语法元素中的至少一个。

9.如权利要求7所述的方法,其中,所述预定语法元素是coeff_sign_flag、sig_coeff_flag、abs_level_gtx_flag和par_level_flag。

10.一种对图像进行编码的方法,所述方法包括:

通过对第一语法元素进行二值化来产生当前块的第一语法元素的二进制位;

确定是否对第一语法元素执行上下文更新;

基于所述确定来更新用于对第一语法元素进行熵编码的上下文;以及

通过使用更新的上下文和产生的二进制位来产生当前块的比特流,

其中,基于用于当前块的预编码的预定语法元素的数量来确定是否执行所述上下文更新。

11.如权利要求10所述的方法,其中,当预编码的所述预定语法元素的数量等于或小于预设值时,执行第一语法元素的所述上下文更新。

12.如权利要求11所述的方法,其中,所述预设值是基于当前块的尺寸被确定的。

13.如权利要求10所述的方法,其中,仅在当前块处于变换跳过模式时,执行所述上下文更新。

14.如权利要求10所述的方法,其中,第一语法元素是与当前块的残差信号相关的语法元素。

15.如权利要求14所述的方法,其中,第一语法元素是指示量化等级的符号的语法元素。

16.如权利要求15所述的方法,其中,第一语法元素是coeff_sign_flag。

17.如权利要求10所述的方法,其中,所述预定语法元素是残差信号相关语法元素中的至少一个。

18.如权利要求17所述的方法,其中,所述预定语法元素是coeff_sign_flag、sig_

coeff_flag、abs_level_gtx_flag和par_level_flag。

19.一种存储由用于对图像进行解码的设备接收并且被用于重建包括在当前画面中的当前块的比特流的计算机可读记录介质，其中，所述比特流由对图像进行编码的方法产生，所述方法包括：

通过对第一语法元素进行二值化来产生当前块的第一语法元素的二进制位；

确定是否对第一语法元素执行上下文更新；

基于所述确定来更新用于对第一语法元素进行熵编码的上下文；以及

通过使用更新的上下文和产生的二进制位来产生比特流，

其中，基于用于当前块的预编码的预定语法元素的数量来确定是否执行所述上下文更新。

图像编码/解码方法和设备以及存储比特流的记录介质

技术领域

[0001] 本发明涉及一种用于对图像进行编码/解码的方法和设备，以及一种用于存储比特流的记录介质。更具体地，本发明涉及一种用于基于熵编码对图像进行编码/解码的方法和设备，以及一种用于存储比特流的记录介质。

背景技术

[0002] 近来，在各种应用领域中，对高分辨率和高质量图像(诸如高清晰度(HD)图像和超高清晰度(UHD)图像)的需求已经增加。然而，与传统图像数据相比，更高分辨率和更高质量的图像数据具有越来越多的数据量。因此，当通过使用介质(诸如传统有线和无线宽带网络)来传输图像数据时，或者当通过使用传统存储介质来存储图像数据时，传输和存储的成本增加。为了解决随着图像数据的分辨率和质量的提高而出现的这些问题，对于更高分辨率和更高质量的图像，需要高效的图像编码/解码技术。

[0003] 图像压缩技术包括各种技术，包括：帧间预测技术，从当前画面的先前画面或后续画面预测包括在当前画面中的像素值；帧内预测技术，通过使用当前画面中的像素信息来预测包括在当前画面中的像素值；变换和量化技术，用于压缩残差信号的能量；熵编码技术，将短码分配给具有高出现频率的值并将长码分配给具有低出现频率的值等。通过使用这种图像压缩技术，可以有效地压缩图像数据，并且可以传输或存储图像数据。

发明内容

[0004] 技术问题

[0005] 本发明的目的是提供一种能够提高压缩效率的图像编码/解码方法和设备，以及一种存储由该方法或设备产生的比特流的记录介质。

[0006] 本发明的另一目的是提供一种能够通过使用熵编码来提高压缩效率的图像编码/解码方法和设备，以及存储由该方法或设备产生的比特流的记录介质。

[0007] 本发明的另一目的是提供一种能够通过使用熵编码来提高吞吐量的图像编码/解码方法和设备，以及存储由该方法或设备产生的比特流的记录介质。

[0008] 技术方案

[0009] 根据本发明，一种图像解码方法包括：确定是否对当前块的第一语法元素执行上下文更新；基于所述确定来更新用于对第一语法元素进行熵解码的上下文；以及基于更新的上下文来产生第一语法元素的二进制位，其中，基于用于当前块的预解码的预定语法元素的数量来确定是否执行所述上下文更新。

[0010] 其中，当预解码的所述预定语法元素的数量等于或小于预设值时，执行第一语法元素的上下文更新。

[0011] 其中，所述预设值是基于当前块的尺寸被确定的。

[0012] 其中，仅在当前块处于变换跳过模式时，执行所述上下文更新。

[0013] 其中，第一语法元素是与当前块的残差信号相关的语法元素。

[0014] 其中,第一语法元素是指示量化等级的符号的语法元素。

[0015] 其中,第一语法元素是coeff_sign_flag。

[0016] 其中,所述预定语法元素是残差信号相关语法元素中的至少一个。

[0017] 其中,所述预定语法元素是coeff_sign_flag、sig_coeff_flag、abs_level_gtx_flag和par_level_flag。

[0018] 根据本发明,一种图像编码方法包括:通过对第一语法元素进行二值化来产生当前块的第一语法元素的二进制位;确定是否对第一语法元素执行上下文更新;基于所述确定来更新用于对第一语法元素进行熵编码的上下文;以及通过使用更新的上下文和产生的二进制位来产生当前块的比特流,其中,基于用于当前块的预编码的预定语法元素的数量来确定是否执行所述上下文更新。

[0019] 其中,当预编码的所述预定语法元素的数量等于或小于预设值时,执行第一语法元素的上下文更新。

[0020] 其中,所述预设值是基于当前块的尺寸被确定的。

[0021] 其中,仅在当前块处于变换跳过模式时,执行所述上下文更新。

[0022] 其中,第一语法元素是与当前块的残差信号相关的语法元素。

[0023] 其中,第一语法元素是指示量化等级的符号的语法元素。

[0024] 其中,第一语法元素是coeff_sign_flag。

[0025] 其中,所述预定语法元素是残差信号相关语法元素中的至少一个。

[0026] 其中,所述预定语法元素是coeff_sign_flag、sig_coeff_flag、abs_level_gtx_flag和par_level_flag。

[0027] 根据本发明,一种存储由用于对图像进行解码的设备接收并且用于重建包括在当前画面中的当前块的比特流的计算机可读记录介质,其中,所述比特流由对图像进行编码的方法产生,所述方法包括:通过对第一语法元素进行二值化来产生当前块的第一语法元素的二进制位;确定是否执行第一语法元素的上下文更新;基于所述确定来更新用于对第一语法元素进行熵编码的上下文;以及通过使用更新的上下文和产生的二进制位来产生比特流,其中,基于用于当前块的预编码的预定语法元素的数量来确定是否执行所述上下文更新。

[0028] 有益效果

[0029] 根据本发明,可以提供一种能够提高压缩效率的图像编码/解码方法和设备,并且可以提供一种存储由该方法或设备产生的比特流的记录介质。

[0030] 另外,根据本发明,可以提供一种能够通过使用熵编码来提高压缩效率的图像编码/解码方法和设备,以及一种存储由该方法或设备产生的比特流的记录介质。

[0031] 另外,根据本发明,可以提供一种能够通过使用熵编码来提高吞吐量的图像编码/解码方法和设备,以及一种存储由该方法或设备产生的比特流的记录介质。

附图说明

[0032] 图1是示出应用本发明的编码设备的配置的框图。

[0033] 图2是示出应用本发明的解码设备的配置的框图。

[0034] 图3是示意性地示出当对图像进行编码和解码时的分区结构的示图。

[0035] 图4是示出帧内预测的示例的示图。

[0036] 图5是示出帧间预测的示例的示图。

[0037] 图6是示出变换和量化的示例的示图。

[0038] 图7是示出可用于帧内预测的参考样点的示图。

[0039] 图8是示出根据本发明一个实施例的熵解码设备的示图。

[0040] 图9是示出根据本发明的一个实施例的熵编码设备的示图。

[0041] 图10是示出根据本发明实施例的熵编码方法和熵解码方法的示图。

[0042] 图11是示出根据本发明的另一实施例的熵编码方法和熵解码方法的示图。

[0043] 图12是示出根据本发明实施例的熵编码方法的另一示图。

[0044] 图13是示出根据本发明实施例的熵解码方法的另一示图。

[0045] 图14是示出根据本发明的实施例的上下文和二进制串的示图。

[0046] 图15是示出根据本发明实施例的使用查找表的上下文更新方法的示图。

[0047] 图16是示出根据本发明实施例的使用查找表的上下文更新方法的另一示图。

[0048] 图17是示出根据本发明的实施例的使用线性概率更新模型的上下文更新方法的示图。

[0049] 图18是示出根据本发明另一实施例的使用查找表的上下文更新方法的示图。

[0050] 图19是示出根据本发明的另一实施例的使用线性概率更新模型的上下文更新方法的示图。

[0051] 图20是示出根据本发明实施例的使用概率分布模型的上下文更新方法的示图。

[0052] 图21是示出根据本发明的实施例的同时执行熵编码或熵解码处理和上下文更新处理的方法的示图。

[0053] 图22是示出根据本发明实施例的上下文初始化的示图。

[0054] 图23是示出根据本发明另一实施例的上下文初始化的示图。

[0055] 图24是示出根据本发明实施例的比特流的示图。

[0056] 图25是示出根据本发明另一实施例的比特流的示图。

具体实施方式

[0057] 最佳模式

[0058] 可以对本发明进行各种修改，并且存在本发明的各种实施例，其中，现在将参照附图来提供本发明的各种实施例的示例并对其进行详细描述。然而，本发明不限于此，尽管示例性实施例可被解释为包括本发明的技术构思和技术范围内的所有修改、等同或替代。在各个方面，相似的附图标号指代相同或相似的功能。在附图中，为了清楚，可夸大元件的形状和尺寸。在本发明的以下详细描述中，参照了附图，其中，附图以图示的方式示出了可实践本发明的特定实施例。足够详细地描述了这些实施例以使本领域技术人员能够实施本公开。应当理解的是，本公开的各种实施例尽管不同，但不一定是互斥的。例如，在不脱离本公开的精神和范围的情况下，结合一个实施例在此描述的特定特征、结构和特征可在其他实施例中被实现。另外，应当理解，在不脱离本公开的精神和范围的情况下，可修改每个公开的实施例内的各个元件的位置或布置。因此，以下详细描述不应被视为具有限制意义，并且本公开的范围仅由所附权利要求(在合适的解释的情况下，还连同权利要求所要求保护的

等同物的全部范围)来限定。

[0059] 说明书中使用的术语“第一”、“第二”等可用于描述各种组件,但是组件不应解释为限于这些术语。这些术语仅用于区分一个组件与其他组件。例如,在不脱离本发明的范围的情况下,“第一”组件可被命名为“第二”组件,并且“第二”组件也可被相似地命名为“第一”组件。术语“和/或”包括多个项的组合或多个项中的任意一项。

[0060] 将理解的是,在本说明书中,当元件被简单称为“连接到”或“耦接到”另一元件而不是“直接连接到”或“直接耦接到”另一元件时,元件可“直接连接到”另一元件或“直接耦接到”另一元件,或者在元件与另一元件之间介入有其他元件的情况下连接到或耦接到另一元件。相反,应当理解,当元件被称为“直接耦接”或“直接连接”到另一元件时,不存在中间元件。

[0061] 另外,本发明的实施例中所示的构成部分被独立地示出,以表示彼此不同的特征功能。因此,这并不表示每个构成部分都以单独的硬件或软件的构成单元构成。换言之,为了方便,每个构成部分包括列举的构成部分中的每个。因此,每个构成部分的至少两个构成部分可被组合以形成一个构成部分,或者一个构成部分可被分区为多个构成部分以执行每种功能。如果没有脱离本发明的实质,则将每个构成部分被组合的实施例和一个构成部分被分区的实施例也包括在本发明的范围内。

[0062] 本说明书中使用的术语仅用于描述特定实施例,而不旨在限制本发明。除非在上下文中具有明显不同的含义,否则以单数形式使用的表述包括复数形式的表述。在本说明书中,将理解,诸如“包括”、“具有”等的术语旨在指示存在说明书中公开的特征、编号、步骤、动作、元件、部件或其组合,而并不旨在排除可存在或可添加一个或更多个其他特征、编号、步骤、动作、元件、部件或其组合的可能性。换言之,当特定元素被称为“被包括”时,并不排除除了相应元素之外的元素,而是可在本发明的实施例或本发明的范围中包括附加的元素。

[0063] 另外,某些构成部分可能不是执行本发明的基本功能的必不可少的构成部分,而是仅提高其性能的选择性构成部分。可通过仅包括用于实现本发明的本质的必不可少的构成部分而不包括用于提高性能的构成部分来实现本发明。仅包括必不可少的构成部分而不包括仅用于提高性能的选择性构成部分的结构也包括在本发明的范围内。

[0064] 在下文中,将参照附图详细描述本发明的实施例。在描述本发明的示例性实施例时,将不详细描述公知的功能或构造,因为它们可能不必要地模糊对本发明的理解。附图中相同的构成元件由相同的附图标号表示,并且对相同元件的重复描述将被省略。

[0065] 在下文中,图像可指构成视频的画面,或者可指视频本身。例如,“对图像进行编码或解码或者进行编码和解码两者”可指“对运动画面进行编码或解码或者进行编码和解码两者”,并且可指“对运动画面的图像中的一个图像进行编码或解码或者进行编码和解码两者”。

[0066] 在下文中,术语“运动画面”和“视频”可用作相同的含义并且可彼此替换。

[0067] 在下文中,目标图像可以是作为编码目标的编码目标图像和/或作为解码目标的解码目标图像。另外,目标图像可以是输入到编码设备的输入图像、以及输入到解码设备的输入图像。这里,目标图像可与当前画面具有相同的含义。

[0068] 在下文中,术语“图像”、“画面”、“帧”和“屏幕”可被用作相同的含义并且可彼此替

换。

[0069] 在下文中，目标块可以是作为编码目标的编码目标块和/或作为解码目标的解码目标块。另外，目标块可以是作为当前编码和/或解码的目标的当前块。例如，术语“目标块”和“当前块”可被用作相同的含义并且可彼此替换。

[0070] 在下文中，术语“块”和“单元”可被用作相同的含义并且可彼此替换。或者“块”可表示特定单元。

[0071] 在下文中，术语“区域”和“片段”可彼此替换。

[0072] 在下文中，特定信号可以是表示特定块的信号。例如，原始信号可以是表示目标块的信号。预测信号可以是表示预测块的信号。残差信号可以是表示残差块的信号。

[0073] 在实施例中，特定信息、数据、标志、索引、元素和属性等中的每个可具有值。信息、数据、标志、索引、元素和属性的值等于“0”可表示逻辑假或第一预定义值。换言之，值“0”、假、逻辑假和第一预定义值可彼此替换。信息、数据、标志、索引、元素和属性的值等于“1”可表示逻辑真或第二预定义值。换言之，值“1”、真、逻辑真和第二预定义值可彼此替换。

[0074] 当变量i或j用于表示列、行或索引时，i的值可以是等于或大于0的整数、或者是等于或大于1的整数。即，列、行、索引等可从0开始计数，或者可从1开始计数。

[0075] 术语描述

[0076] 编码器：表示执行编码的设备。即，表示编码设备。

[0077] 解码器：表示执行解码的设备。即，表示解码设备。

[0078] 块：是M×N的样点阵列。这里，M和N可表示正整数，并且块可表示二维形式的样点阵列。块可指单元。当前块可表示在编码时成为目标的编码目标块，或者在解码时成为目标的解码目标块。另外，当前块可以是编码块、预测块、残差块和变换块中的至少一个。

[0079] 样点：是构成块的基本单元。根据比特深度(Bd)，样点可被表示为从0到$2^{Bd}-1$的值。在本发明中，样点可被用作像素的含义。即，样点、pel、像素可具有彼此相同的含义。

[0080] 单元：可指编码和解码单元。当对图像进行编码和解码时，单元可以是通过对单个图像进行分区而产生的区域。另外，当在编码或解码期间将单个图像分区为子分区单元时，单元可表示子分区单元。即，图像可被分区为多个单元。当对图像进行编码和解码时，可以执行针对每个单元的预定处理。单个单元可被分区为尺寸小于该单元的尺寸的子单元。依据功能，单元可表示块、宏块、编码树单元、编码树块、编码单元、编码块、预测单元、预测块、残差单元、残差块、变换单元、变换块等。另外，为了将单元与块区分开，单元可包括亮度分量块、与亮度分量块相关联的色度分量块、以及每个颜色分量块的语法元素。单元可具有各种尺寸和形状，具体地，单元的形状可以是二维几何图形，诸如正方形、矩形、梯形、三角形、五边形等。另外，单元信息可包括指示编码单元、预测单元、变换单元等的单元类型以及单元尺寸、单元深度、单元的编码和解码的顺序等中的至少一个。

[0081] 编码树单元：被配置有亮度分量Y的单个编码树块以及与色度分量Cb和Cr相关的两个编码树块。另外，编码树单元可表示包括块和每个块的语法元素。可通过使用四叉树分区方法、二叉树分区方法和三叉树分区方法中的至少一个对每个编码树单元进行分区，以配置诸如编码单元、预测单元、变换单元等的更低等级的单元。编码树单元可被用作用于指定在对作为输入图像的图像进行编码/解码时成为处理单元的样点块的术语。这里，四叉树可表示四叉分树。

[0082] 当编码块的尺寸在预定范围内时，可以仅使用四叉树分区进行分区。这里，预定范围可被定义为能够仅使用四叉树分区进行分区的编码块的最大尺寸和最小尺寸中的至少一个。可通过比特流用信号发送指示允许四叉树分区的编码块的最大/最小尺寸的信息，并且可在序列、画面参数、并行块组或条带(片段)中的至少一个单元中用信号发送所述信息。可选地，编码块的最大/最小尺寸可以是编码器/解码器中预定的固定尺寸。例如，当编码块的尺寸与256×256至64×64相应时，仅使用四叉树分区来进行分区是可能的。可选地，当编码块的尺寸大于最大转换块的尺寸时，仅使用四叉树分区来进行分区是可能的。这里，将被分区的块可以是编码块和变换块中的至少一个。在这种情况下，指示编码块的分区的信息(例如，split_flag)可以是指示是否执行四叉树分区的标志。当编码块的尺寸落在预定范围内时，仅使用二叉树或三叉树分区来进行分区是可能的。在这种情况下，四叉树分区的以上描述可以相同方式被应用于二叉树分区或三叉树分区。

[0083] 编码树块：可用作用于指定Y编码树块、Cb编码树块和Cr编码树块中的任意一个的术语。

[0084] 邻近块：可表示与当前块相邻的块。与当前块相邻的块可表示与当前块的边界接触的块、或者位于距当前块预定距离内的块。邻近块可表示与当前块的顶点相邻的块。这里，与当前块的顶点相邻的块可表示与水平相邻于当前块的邻近块垂直相邻的块、或者与垂直相邻于当前块的邻近块水平相邻的块。

[0085] 重建邻近块：可表示与当前块相邻并且已经在空间/时间上被编码或解码的邻近块。这里，重建邻近块可表示重建邻近单元。重建空间邻近块可以是在当前画面内并且已经通过编码或解码或者编码和解码两者而被重建块。重建时间邻近块是在参考画面内的与当前画面的当前块相应的位置处的块或所述块的邻近块。

[0086] 单元深度：可表示单元的分区程度。在树结构中，最高节点(根节点)可与未被分区的第一单元相应。另外，最高节点可具有最小深度值。在这种情况下，最高节点的深度可以为等级0。深度为等级1的节点可表示通过对第一单元进行首次分区而产生的单元。深度为等级2的节点可表示通过对第一单元进行两次分区而产生的单元。深度为等级n的节点可表示通过对第一单元进行n次分区而产生的单元。叶节点可以是最低节点并且是不能被进一步分区的节点。叶节点的深度可以是最大等级。例如，最大等级的预定义值可以是3。根节点的深度可以是最低的，并且叶节点的深度可以是最深的。另外，当单元被表示为树结构时，单元所存在于的等级可表示单元深度。

[0087] 比特流：可表示包括编码图像信息的比特流。

[0088] 参数集：与比特流内的配置之中的头信息相应。视频参数集、序列参数集、画面参数集和自适应参数集中的至少一个可被包括在参数集中。此外，参数集可包括条带(slice)头、并行块(tile)组头和并行块头信息。术语“并行块组”表示一组并行块并且具有与条带相同的含义。

[0089] 自适应参数集是指可由不同画面、子画面、条带、并行块组、并行块或分块共享和参考的参数集。另外，画面中的子画面、条带、并行块组、并行块或分块可指代不同的自适应参数集以使用不同的自适应参数集中的信息。

[0090] 关于自适应参数集，画面中的子画面、条带、并行块组、并行块或分块可通过使用相应自适应参数集的标识符来指代不同的自适应参数集。

[0091] 关于自适应参数集，子画面中的条带、并行块组、并行块或分块可通过使用相应自适应参数集的标识符来指代不同的自适应参数集。

[0092] 关于自适应参数集，条带中的并行块或分块可以通过使用相应自适应参数集的标识符来指代不同的自适应参数集。

[0093] 关于自适应参数集，并行块中的分块可以通过使用相应自适应参数集的标识符来指代不同的自适应参数集。

[0094] 子画面的参数集或头可以包括关于自适应参数集标识符的信息。因此，可以在子画面中使用与自适应参数集标识符相应的自适应参数集。

[0095] 并行块的参数集或头可以包括自适应参数集标识符，使得可以在并行块中使用与自适应参数集标识符相应的自适应参数集。

[0096] 分块的头可以包括关于自适应参数集标识符的信息，使得可以在分块中使用与自适应参数集标识符相应的自适应参数集。

[0097] 画面可被划分为一个或更多个并行块行和一个或更多个并行块列。

[0098] 画面中的子画面可被划分为一个或更多个并行块行和一个或更多个并行块列。子画面可以是画面中的矩形区域或正方形区域，并且可包括一个或更多个CTU。子画面可包括至少一个并行块、分块和/或条带。

[0099] 并行块可以是画面中的矩形区域或正方形区域，并且可包括一个或更多个CTU。并行块可以被划分成一个或更多个分块。

[0100] 分块可指并行块中的一个或更多个CTU行。并行块可以被划分成一个或更多个分块，并且每个块可以具有至少一个CTU行。未被划分成两个或更多个分块的并行块也可以表示分块。

[0101] 条带可包括画面中的一个或更多个并行块，并且可包括并行块中的一个或更多个分块。

[0102] 解析：可表示通过执行熵解码来确定语法元素的值，或者可表示熵解码本身。

[0103] 符号：可表示编码/解码目标单元的语法元素、编码参数和变换系数值中的至少一个。此外，符号可表示熵编码目标或熵解码结果。

[0104] 预测模式：可以是指示利用帧内预测而被编码/解码的模式或利用帧间预测而被编码/解码的模式的信息。

[0105] 预测单元：可表示当执行预测(诸如帧间预测、帧内预测、帧间补偿、帧内补偿和运动补偿)时的基本单元。单个预测单元可被分区为具有更小尺寸的多个分区，或者可被分区为多个更低等级的预测单元。多个分区可以是在执行预测或补偿时的基本单元。通过分区预测单元而产生的分区也可以是预测单元。

[0106] 预测单元分区：可表示通过对预测单元进行分区而获得的形状。

[0107] 参考画面列表可指包括用于帧间预测或运动补偿的一个或更多个参考画面的列表。存在若干类型的可用的参考画面列表，包括LC(列表组合)、L0(列表0)、L1(列表1)、L2(列表2)、L3(列表3)。

[0108] 帧间预测指示符可以指当前块的帧间预测的方向(单向预测、双向预测等)。可选地，帧间预测指示符可指用于产生当前块的预测块的参考画面的数量。可选地，帧间预测指示符可指在对当前块进行帧间预测或运动补偿时使用的预测块的数量。

[0109] 预测列表利用标志指示是否使用特定参考画面列表中的至少一个参考画面来产生预测块。可使用预测列表利用标志来推导帧间预测指示符，并且相反地，可使用帧间预测指示符来推导预测列表利用标志。例如，当预测列表利用标志具有第一值零(0)时，它表示参考画面列表中的参考画面不被用于产生预测块。另一方面，当预测列表利用标志具有第二值一(1)时，它表示参考画面列表被用于产生预测块。

[0110] 参考画面索引可指的是指示参考画面列表中的特定参考画面的索引。

[0111] 参考画面可表示由特定块参考以用于特定块的帧间预测或运动补偿的目的的参考画面。可选地，参考画面可以是包括由当前块参考以用于帧间预测或运动补偿的参考块的画面。在下文中，术语“参考画面”和“参考画面”具有相同的含义并且可以互换。

[0112] 运动矢量可以是用于帧间预测或运动补偿的二维矢量。运动矢量可表示编码/解码的目标块与参考块之间的偏移。例如，(mvX，mvY)可表示运动矢量。这里，mvX可以表示水平分量，并且mvY可以表示垂直分量。

[0113] 搜索范围可以是在帧间预测期间被搜索以检索运动矢量的二维区域。例如，搜索范围的尺寸可以是M×N。这里，M和N都是整数。

[0114] 运动矢量候选可以指在对运动矢量进行预测时的预测候选块或预测候选块的运动矢量。另外，运动矢量候选可以被包括在运动矢量候选列表中。

[0115] 运动矢量候选列表可表示由一个或更多个运动矢量候选组成的列表。

[0116] 运动矢量候选索引可表示指示运动矢量候选列表中的运动矢量候选的指示符。可选地，它可以是运动矢量预测因子的索引。

[0117] 运动信息可表示包括运动矢量、参考画面索引、帧间预测指示符、预测列表利用标志、参考画面列表信息、参考画面、运动矢量候选、运动矢量候选索引、合并候选和合并索引中的至少一项的信息。

[0118] 合并候选列表可表示由一或更多个合并候选组成的列表。

[0119] 合并候选可表示空间合并候选、时间合并候选、组合合并候选、组合双预测合并候选或零合并候选。合并候选可以包括诸如帧间预测指示符、每个列表的参考画面索引、运动矢量、预测列表利用标志和帧间预测指示符的运动信息。

[0120] 合并索引可表示指示合并候选列表中的合并候选的指示符。可选地，合并索引可指示在空间上/时间上与当前块相邻的重建块中的块，其中，已从该块推导合并候选。可选地，合并索引可指示合并候选的至少一条运动信息。

[0121] 变换单元：可表示在对残差信号执行编码/解码(诸如变换、逆变换、量化、反量化、变换系数编码/解码)时的基本单元。单个变换单元可被分区为具有更小尺寸的多个更低等级的变换单元。这里，变换/逆变换可包括第一次变换/第一次逆变换和第二次变换/第二次逆变换中的至少一个。

[0122] 缩放：可表示将量化的等级乘以因子的处理。可通过对量化的等级进行缩放来产生变换系数。缩放也可被称为反量化。

[0123] 量化参数：可表示当在量化期间使用变换系数来产生量化的等级时使用的值。量化参数还可表示当在反量化期间通过对量化的等级进行缩放来产生变换系数时使用的值。量化参数可以是被映射在量化步长上的值。

[0124] 增量量化参数：可表示预测的量化参数与编码/解码目标单元的量化参数之间的

差值。

[0125] 扫描：可表示对单元、块或矩阵内的系数进行排序的方法。例如，将系数的二维矩阵改成为一维矩阵可被称为扫描，将系数的一维矩阵改成为二维矩阵可被称为扫描或逆扫描。

[0126] 变换系数：可表示在编码器中执行变换之后产生的系数值。变换系数可表示在解码器中执行熵解码和反量化中的至少一个之后产生的系数值。通过对变换系数或残差信号进行量化而获得的量化的等级或者量化的变换系数等级也可落入变换系数的含义内。

[0127] 量化的等级：可表示在编码器中通过对变换系数或残差信号进行量化而产生的值。可选地，量化的等级可表示作为在解码器中经历反量化的反量化目标的值。相似地，作为变换和量化的结果的量化的变换系数等级也可落入量化的等级的含义内。

[0128] 非零变换系数：可表示具有除零之外的值的变换系数、或者具有除零之外的值的变换系数等级或量化的等级。

[0129] 量化矩阵：可表示在为了提高主观图像质量或客观图像质量而执行的量化处理或反量化处理中使用的矩阵。量化矩阵也可被称为缩放列表。

[0130] 量化矩阵系数：可表示量化矩阵内的每个元素。量化矩阵系数也可被称为矩阵系数。

[0131] 默认矩阵：可表示在编码器或解码器中预先定义的预定量化矩阵。

[0132] 非默认矩阵：可表示在编码器或解码器中未被预先定义而是由用户用信号发送的量化矩阵。

[0133] 统计值：针对具有可计算的特定值的变量、编码参数、常量值等之中的至少一个的统计值可以是相应特定值的平均值、求和值、加权平均值、加权和值、最小值、最大值、最频繁出现的值、中值、插值之中的一个或更多个。

[0134] 图1是示出根据应用了本发明的实施例的编码设备的配置的框图。

[0135] 编码设备100可以是编码器、视频编码设备或图像编码设备。视频可包括至少一个图像。编码设备100可顺序地对至少一个图像进行编码。

[0136] 参照图1，编码设备100可包括运动预测单元111、运动补偿单元112、帧内预测单元120、切换器115、减法器125、变换单元130、量化单元140、熵编码单元150、反量化单元160、逆变换单元170、加法器175、滤波器单元180和参考画面缓冲器190。

[0137] 编码设备100可通过使用帧内模式或帧间模式或者帧内模式和帧间模式两者来执行输入图像的编码。此外，编码设备100可通过对输入图像进行编码来产生包括编码信息的比特流，并输出产生的比特流。产生的比特流可被存储在计算机可读记录介质中，或者可通过有线/无线传输介质被流传输。当帧内模式被用作预测模式时，切换器115可切换到帧内。可选地，当帧间模式被用作预测模式时，切换器115可切换到帧间模式。这里，帧内模式可表示帧内预测模式，帧间模式可表示帧间预测模式。编码设备100可产生针对输入图像的输入块的预测块。此外，编码设备100可在产生预测块之后使用输入块和预测块的残差对残差块进行编码。输入图像可被称为作为当前编码目标的当前画面。输入块可被称为作为当前编码目标的当前块，或者被称为编码目标块。

[0138] 当预测模式是帧内模式时，帧内预测单元120可使用已被编码/解码并与当前块相邻的块的样点作为参考样点。帧内预测单元120可通过使用参考样点来对当前块执行空间

预测，或者通过执行空间预测来产生输入块的预测样点。这里，帧内预测可表示帧内部的预测。

[0139] 当预测模式是帧间模式时，运动预测单元111可在执行运动预测时从参考画面检索与输入块最匹配的区域，并且通过使用检索到的区域来推导运动矢量。在这种情况下，搜索区域可被用作所述区域。参考画面可被存储在参考画面缓冲器190中。这里，当执行对参考画面的编码/解码时，参考画面可被存储在参考画面缓冲器190中。

[0140] 运动补偿单元112可通过使用运动矢量对当前块执行运动补偿来产生预测块。这里，帧间预测可表示帧之间的预测或运动补偿。

[0141] 当运动矢量的值不是整数时，运动预测单元111和运动补偿单元112可通过将插值滤波器应用于参考画面的部分区域来产生预测块。为了对编码单元执行画面间预测或运动补偿，可确定跳过模式、合并模式、高级运动矢量预测(AMVP)模式和当前画面参考模式之中的哪个模式被用于对包括在相应编码单元中的预测单元的运动预测和运动补偿。然后，依据所确定的模式，可不同地执行画面间预测或运动补偿。

[0142] 减法器125可通过使用输入块和预测块的差来产生残差块。残差块可被称为残差信号。残差信号可表示原始信号和预测信号之间的差。此外，残差信号可以是通过对原始信号与预测信号之间的差进行变换或量化或者变换和量化而产生的信号。残差块可以是块单元的残差信号。

[0143] 变换单元130可通过对残差块执行变换来产生变换系数，并输出产生的变换系数。这里，变换系数可以是通过对残差块执行变换而产生的系数值。当变换跳过模式被应用时，变换单元130可跳过对残差块的变换。

[0144] 可通过将量化应用于变换系数或应用于残差信号来产生量化的等级。在下文中，量化的等级在实施例中也可被称为变换系数。

[0145] 量化单元140可通过根据参数对变换系数或残差信号进行量化来产生量化的等级，并输出产生的量化的等级。这里，量化单元140可通过使用量化矩阵对变换系数进行量化。

[0146] 熵编码单元150可通过根据概率分布对由量化单元140计算出的值或者对在执行编码时计算出的编码参数值执行熵编码来产生比特流，并输出产生的比特流。熵编码单元150可对图像的样点信息和用于对图像进行解码的信息执行熵编码。例如，用于对图像进行解码的信息可包括语法元素。

[0147] 当熵编码被应用时，符号被表示为使得较少数量的比特被分配给具有高产生可能性的符号，并且较多数量的比特被分配给具有低产生可能性的符号，因此，可减小用于将被编码的符号的比特流的尺寸。熵编码单元150可使用诸如指数哥伦布、上下文自适应可变长度编码(CAVLC)、上下文自适应二叉算术编码(CABAC)等的用于熵编码的编码方法。例如，熵编码单元150可通过使用变长编码/码(VLC)表来执行熵编码。此外，熵编码单元150可推导目标符号的二值化方法和目标符号/二叉位的概率模型，并且通过使用推导的二值化方法和上下文模型来执行算术编码。

[0148] 为了对变换系数等级(量化的等级)进行编码，熵编码单元150可通过使用变换系数扫描方法将二维块形式的系数改成为一维矢量形式。

[0149] 编码参数可包括在编码器中被编码并且被用信号发送到解码器的诸如语法元素

的信息(标志、索引等)以及在执行编码或解码时推推导的信息。编码参数可表示在对图像进行编码或解码时所需要的信息。例如,以下项中的至少一个值或组合形式可被包括在编码参数中:单元/块尺寸、单元/块深度、单元/块分区信息、单元/块形状、单元/块分区结构、是否进行四叉树形式的分区、是否进行二叉树形式的分区、二叉树形式的分区方向(水平方向或垂直方向)、二叉树形式的分区形式(对称分区或非对称分区)、当前编码单元是否通过三叉树分区被分区、三叉树分区的方向(水平方向或垂直方向)、三叉树分区的类型(对称类型或非对称类型)、当前编码单元是否通过多类型树分区被分区、多类型树分区的方向(水平方向或垂直方向)、多类型树分区的类型(对称类型或非对称类型)、多类型树分区的树(二叉树或三叉树)结构、预测模式(帧内预测或帧间预测)、亮度帧内预测模式/方向、色度帧内预测模式/方向、帧内分区信息、帧间分区信息、编码块分区标志、预测块分区标志、变换块分区标志、参考样点滤波方法、参考样点滤波器抽头、参考样点滤波器系数、预测块滤波方法、预测块滤波器抽头、预测块滤波器系数、预测块边界滤波方法、预测块边界滤波器抽头、预测块边界滤波器系数、帧内预测模式、帧间预测模式、运动信息、运动矢量、运动矢量差、参考画面索引、帧间预测角度、帧间预测指示符、预测列表利用标志、参考画面列表、参考画面、运动矢量预测因子索引、运动矢量预测因子候选、运动矢量候选列表、是否使用合并模式、合并索引、合并候选、合并候选列表、是否使用跳过模式、插值滤波器类型、插值滤波器抽头、插值滤波器系数、运动矢量尺寸、运动矢量的表示精度、变换类型、变换尺寸、首次(第一次)变换是否被使用的信息、二次变换是否被使用的信息、首次变换索引、二次变换索引、残差信号是否存在的信息、编码块样式、编码块标志(CBF)、量化参数、量化参数残差、量化矩阵、是否应用帧内环路滤波器、帧内环路滤波器系数、帧内环路滤波器抽头、帧内环路滤波器形状/形式、是否应用去块滤波器、去块滤波器系数、去块滤波器抽头、去块滤波器强度、去块滤波器形状/形式、是否应用自适应样点偏移、自适应样点偏移值、自适应样点偏移类别、自适应样点偏移类型、是否应用自适应环路滤波器、自适应环路滤波器系数、自适应环路滤波器抽头、自适应环路滤波器形状/形式、二值化/逆二值化方法、上下文模型确定方法、上下文模型更新方法、是否执行常规模式、是否执行旁路模式、上下文二叉位、旁路二叉位、有效系数标志、最后有效系数标志、针对系数组的单元的编码标志、最后有效系数的位置、关于系数的值是否大于1的标志、关于系数的值是否大于2的标志、关于系数的值是否大于3的标志、关于其余系数值的信息、符号信息、重建亮度样点、重建色度样点、残差亮度样点、残差色度样点、亮度变换系数、色度变换系数、量化的亮度等级、量化的色度等级、变换系数等级扫描方法、在解码器侧的运动矢量搜索区域的尺寸、在解码器侧的运动矢量搜索区域的形状、在解码器侧的运动矢量搜索的次数、关于CTU尺寸的信息、关于最小块尺寸的信息、关于最大块尺寸的信息、关于最大块深度的信息、关于最小块深度的信息、图像显示/输出顺序、条带标识信息、条带类型、条带分区信息、并行块标识信息、并行块类型、并行块分区信息、并行块组标识信息、并行块组类型、并行块组分区信息、画面类型、输入样点的比特深度、重建样点的比特深度、残差样点的比特深度、变换系数的比特深度、量化的等级的比特深度、以及关于亮度信号的信息或关于色度信号的信息。

[0150] 这里,用信号发送标志或索引可表示由编码器对相应标志或索引进行熵编码并将其包括在比特流中,并且可表示由解码器从比特流对相应标志或索引进行熵解码。

[0151] 当编码设备100通过帧间预测执行编码时,编码的当前画面可被用作用于随后被

处理的另一图像的参考画面。因此，编码设备100可对编码的当前画面进行重建或解码，或者将重建或解码的图像作为参考画面存储在参考画面缓冲器190中。

[0152] 量化的等级可在反量化单元160中被反量化，或者可在逆变换单元170中被逆变换。可由加法器175将经过反量化或逆变换的系数或者经过反量化和逆变换两者的系数与预测块相加。通过将经过反量化或逆变换的系数或者经过反量化和逆变换两者的系数与预测块相加，可产生重建块。这里，经过反量化或逆变换的系数或经过反量化和逆变换两者的系数可表示执行了反量化和逆变换中的至少一个的系数，并且可表示重建残差块。

[0153] 重建块可通过滤波器单元180。滤波器单元180可将去块滤波器、样点自适应偏移(SAO)和自适应环路滤波器(ALF)中的至少一个应用于重建样点、重建块或重建图像。滤波器单元180可被称为环内滤波器。

[0154] 去块滤波器可去除在块之间的边界中产生的块失真。为了确定是否应用去块滤波器，可基于块中所包括的若干行或列中包括的样点来确定是否将去块滤波器应用于当前块。当将去块滤波器应用于块时，可根据所需的去块滤波强度来应用另一滤波器。

[0155] 为了补偿编码误差，可通过使用样点自适应偏移将合适的偏移值与样点值相加。样点自适应偏移可以以样点为单位对经过去块的图像与原始图像的偏移进行校正。可使用考虑关于每个样点的边缘信息来应用偏移的方法，或者可使用以下方法：将图像的样点分区为预定数量的区域，确定偏移被应用的区域，并对确定的区域应用偏移。

[0156] 自适应环路滤波器可基于经滤波的重建图像和原始图像的比较结果来执行滤波。可将包括在图像中的样点分区为预定组，可确定将被应用于每个组的滤波器，并且可对每个组执行差异化滤波。是否应用ALF的信息可通过编码单元(CU)被用信号发送，并且将被应用于每个块的ALF的形式和系数可变化。

[0157] 已经通过滤波器单元180的重建块或重建图像可被存储在参考画面缓冲器190中。由滤波器单元180处理的重建块可以是参考画面的一部分。即，参考画面是由滤波器单元180处理的重建块组成的重建图像。存储的参考画面可稍后在帧间预测或运动补偿中被使用。

[0158] 图2是示出根据实施例并且应用了本发明的解码设备的配置的框图。

[0159] 解码设备200可以是解码器、视频解码设备或图像解码设备。

[0160] 参照图2，解码设备200可包括熵解码单元210、反量化单元220、逆变换单元230、帧内预测单元240、运动补偿单元250、加法器255、滤波器单元260和参考画面缓冲器270。

[0161] 解码设备200可接收从编码设备100输出的比特流。解码设备200可接收存储在计算机可读记录介质中的比特流，或者可接收通过有线/无线传输介质被流传输的比特流。解码设备200可通过使用帧内模式或帧间模式对比特流进行解码。此外，解码设备200可产生通过解码而产生的重建图像或解码图像，并输出重建图像或解码图像。

[0162] 当在解码时使用的预测模式是帧内模式时，切换器可被切换到帧内。可选地，当在解码时使用的预测模式是帧间模式时，切换器可被切换到帧间模式。

[0163] 解码设备200可通过对输入比特流进行解码来获得重建残差块，并产生预测块。当重建残差块和预测块被获得时，解码设备200可通过将重建残差块与预测块相加来产产生为解码目标的重建块。解码目标块可被称为当前块。

[0164] 熵解码单元210可通过根据概率分布对比特流进行熵解码来产生符号。产生的符

号可包括量化的等级形式的符号。这里，熵解码方法可以是上述熵编码方法的逆处理。

[0165] 为了对变换系数等级(量化的等级)进行解码，熵解码单元210可通过使用变换系数扫描方法将单向矢量形式的系数改成为二维块形式。

[0166] 可在反量化单元220中对量化的等级进行反量化，或者可在逆变换单元230中对量化的等级进行逆变换。量化的等级可以是进行反量化或逆变换或者进行反量化和逆变换两者的结果，并且可被产生为重建残差块。这里，反量化单元220可将量化矩阵应用于量化的等级。

[0167] 当使用帧内模式时，帧内预测单元240可通过对当前块执行空间预测来产生预测块，其中，空间预测使用与解码目标块相邻并且已经被解码的块的样点值。

[0168] 当使用帧间模式时，运动补偿单元250可通过对当前块执行运动补偿来产生预测块，其中，运动补偿使用运动矢量以及存储在参考画面缓冲器270中的参考画面。

[0169] 加法器225可通过将重建残差块与预测块相加来产生重建块。滤波器单元260可将去块滤波器、样点自适应偏移和自适应环路滤波器中的至少一个应用于重建块或重建图像。滤波器单元260可输出重建图像。重建块或重建图像可被存储在参考画面缓冲器270中并且在执行帧间预测时被使用。由滤波器单元260处理的重建块可以是参考画面的一部分。即，参考画面是由滤波器单元260处理的重建块组成的重建图像。存储的参考画面可稍后在帧间预测或运动补偿中被使用。

[0170] 图3是示意性地示出当对图像进行编码和解码时图像的分区结构的示图。图3示意性地示出将单个单元分区为多个更低等级的单元的示例。

[0171] 为了有效地对图像进行分区，当进行编码和解码时，可使用编码单元(CU)。编码单元可被用作当对图像进行编码/解码时的基本单元。此外，编码单元可被用作用于在对图像进行编码/解码时区分帧内预测模式与帧间预测模式的单元。编码单元可以是用于预测、变换、量化、逆变换、反量化、或对变换系数的编码/解码处理的基本单元。

[0172] 参照图3，图像300按照最大编码单元(LCU)被顺序地分区，并且LCU单元被确定为分区结构。这里，LCU可以以与编码树单元(CTU)相同的含义被使用。单元分区可表示对与该单元相关联的块进行分区。在块分区信息中，可包括单元深度的信息。深度信息可表示单元被分区的次数或程度或者单元被分区的次数和程度两者。可基于树结构将单个单元分区为与深度信息分层地相关联的多个更低等级的单元。换言之，单元和通过对该单元进行分区而产生的更低等级的单元可分别与节点和该节点的子节点相应。分区出的更低等级的单元中的每个可具有深度信息。深度信息可以是表示CU的尺寸的信息，并且可被存储在每个CU中。单元深度表示与对单元进行分区相关的次数和/或程度。因此，更低等级的单元的分区信息可包括关于更低等级的单元的尺寸的信息。

[0173] 分区结构可表示LCU 310内的编码单元(CU)的分布。可根据是否将单个CU分区为多个(等于或大于2的包括2、4、8、16等的正整数)CU来确定这样的分布。通过分区产生的CU的水平尺寸和垂直尺寸可分别是分区之前的CU的水平尺寸和垂直尺寸的一半，或者可分别具有小于根据分区的次数而进行分区之前的水平尺寸和垂直尺寸的尺寸。CU可以被递归地分区为多个CU。通过递归分区，与分区之前的CU的高度和宽度之中的至少一个相比，分区之后的CU的高度和宽度之中的至少一个可减小。可递归地执行CU的分区，直到预定的深度或预定的尺寸为止。例如，LCU的深度可以是0，最小编码单元(SCU)的深度可以是预定的最大

深度。这里，如上所述，LCU可以是具有最大编码单元尺寸的编码单元，并且SCU可以是具有最小编码单元尺寸的编码单元。分区从LCU 310开始，当CU的水平尺寸或垂直尺寸或者水平尺寸和垂直尺寸两者通过分区而减小时，CU深度增加1。例如，对于每个深度，未被分区的CU的尺寸可以为2N×2N。此外，在被分区的CU的情况下，可将尺寸为2N×2N的CU分区为尺寸为N×N的4个CU。随着深度增加1，N的尺寸可减半。

[0174] 另外，可通过使用CU的分区信息来表示CU是否被分区的信息。分区信息可以是1比特信息。除SCU之外的所有CU可包括分区信息。例如，当分区信息的值为第一值时，可不对CU进行分区，当分区信息的值为第二值时，可对CU进行分区。

[0175] 参照图3，具有深度0的LCU可以是64×64的块。0可以是最小深度。具有深度3的SCU可以是8×8的块。3可以是最大深度。32×32的块和16×16的块的CU可分别被表示为深度1和深度2。

[0176] 例如，当单个编码单元被分区为4个编码单元时，分区出的4个编码单元的水平尺寸和垂直尺寸可以是CU在被分区之前的水平尺寸和垂直尺寸的一半尺寸。在一个实施例中，当尺寸为32×32的编码单元被分区为4个编码单元时，分区出的4个编码单元中的每个可具有16×16的尺寸。当单个编码单元被分区为4个编码单元时，可称编码单元可被分区为四叉树形式。

[0177] 例如，当一个编码单元被分区为两个子编码单元时，两个子编码单元中的每个子编码单元的水平尺寸或垂直尺寸(宽度或高度)可以是原始编码单元的水平尺寸或垂直尺寸的一半。例如，当具有32×32的尺寸的编码单元被垂直分区为两个子编码单元时，该两个子编码单元中的每个可具有16×32的尺寸。例如，当具有8×32的尺寸的编码单元被水平分区为两个子编码单元时，该两个子编码单元中的每个可具有8×16的尺寸。当一个编码单元被分区为两个子编码单元时，可称编码单元被二分区或者通过二叉树分区结构被分区。

[0178] 例如，当一个编码单元被分区为三个子编码单元时，可以以1:2:1的比例对编码单元的水平尺寸或垂直尺寸进行分区，从而产生水平尺寸或垂直尺寸的比例为1:2:1的三个子编码单元。例如，当尺寸为16×32的编码单元被水平分区为三个子编码单元时，该三个子编码单元以从最上方子编码单元到最下方子编码单元的顺序可分别具有16×8、16×16和16×8的尺寸。例如，当尺寸为32×32的编码单元被垂直分区为三个子编码单元时，该三个子编码单元以从左侧子编码单元到右侧子编码单元的顺序可分别具有8×32、16×32和8×32的尺寸。当一个编码单元被分区为三个子编码单元时，可称编码单元被三分区或者根据三叉树分区结构被分区。

[0179] 在图3中，编码树单元(CTU)320是四叉树分区结构、二叉树分区结构和三叉树分区结构全部应用于其的CTU的示例。

[0180] 如上所述，为了对CTU进行分区，可应用四叉树分区结构、二叉树分区结构和三叉树分区结构中的至少一个。可根据预定的优先级顺序将各种树分区结构顺序地应用于CTU。例如，可将四叉树分区结构优先应用于CTU。不能再使用四叉树分区结构进行分区的编码单元可与四叉树的叶节点相应。与四叉树的叶节点相应的编码单元可用作二叉树和/或三叉树分区结构的根节点。即，与四叉树的叶节点相应的编码单元可根据二叉树分区结构或三叉树分区结构被进一步分区，或者可不被进一步分区。因此，通过防止从与四叉树的叶节点相应的编码单元的二叉树分区或三叉树分区得到的编码块经历进一步的四叉树分区，块分

区操作和/或用信号发送分区信息的操作可被有效执行。

[0181] 可使用四分区信息用信号发送与四叉树的节点相应的编码单元被分区的事实。具有第一值(例如,“1”)的四分区信息可指示当前编码单元按照四叉树分区结构被分区。具有第二值(例如,“0”)的四分区信息可指示当前编码单元未按照四叉树分区结构被分区。四分区信息可以是具有预定长度(例如,一个比特)的标志。

[0182] 在二叉树分区与三叉树分区之间可能没有优先级。即,与四叉树的叶节点相应的编码单元可进一步经历二叉树分区和三叉树分区中的任意分区。另外,通过二叉树分区或三叉树分区产生的编码单元可经历进一步的二叉树分区或进一步的三叉树分区,或者可不被进一步分区。

[0183] 在二叉树分区和三叉树分区之间不存在优先级的树结构被称为多类型树结构。与四叉树的叶节点相应的编码单元可用作多类型树的根节点。可使用多类型树分区指示信息、分区方向信息和分区树信息中的至少一个来用信号发送是否对与多类型树的节点相应的编码单元进行分区。为了对与多类型树的节点相应的编码单元进行分区,可顺序地用信号发送多类型树分区指示信息、分区方向信息和分区树信息。

[0184] 具有第一值(例如,“1”)的多类型树分区指示信息可指示当前编码单元将经历多类型树分区。具有第二值(例如,“0”)的多类型树分区指示信息可指示当前编码单元将不经历多类型树分区。

[0185] 当与多类型树的节点相应的编码单元按照多类型树分区结构被进一步分区时,所述编码单元可包括分区方向信息。分区方向信息可指示当前编码单元将在哪个方向上针对多类型树分区被分区。具有第一值(例如,“1”)的分区方向信息可指示当前编码单元将被垂直分区。具有第二值(例如,“0”)的分区方向信息可指示当前编码单元将被水平分区。

[0186] 当与多类型树的节点相应的编码单元按照多类型树分区结构被进一步分区时,当前编码单元可包括分区树信息。分区树信息可指示将被用于对多类型树的节点进行分区的树分区结构。具有第一值(例如,“1”)的分区树信息可指示当前编码单元将按照二叉树分区结构被分区。具有第二值(例如,“0”)的分区树信息可指示当前编码单元将按照三叉树分区结构被分区。

[0187] 分区指示信息、分区树信息和分区方向信息均可以是具有预定长度(例如,一个比特)的标志。

[0188] 四叉树分区指示信息、多类型树分区指示信息、分区方向信息和分区树信息中的至少任意一个可被熵编码/熵解码。为了对那些类型的信息进行熵编码/熵解码,可使用关于与当前编码单元相邻的邻近编码单元的信息。例如,当前编码单元的左侧邻近编码单元和/或上方邻近编码单元的分区类型(被分区或不被分区、分区树和/或分区方向)与当前编码单元的分区类型相似的可能性很高。因此,可从关于邻近编码单元的信息推导用于对关于当前编码单元的信息进行熵编码/熵解码的上下文信息。关于邻近编码单元的信息可包括四分区信息、多类型树分区指示信息、分区方向信息和分区树信息中的至少任意一个。

[0189] 作为另一示例,在二叉树分区和三叉树分区中,可优先执行二叉树分区。即,当前编码单元可首先经历二叉树分区,并且随后可将与二叉树的叶节点相应的编码单元设置为用于三叉树分区的根节点。在这种情况下,对于与三叉树的节点相应的编码单元,可既不执行四叉树分区也不执行二叉树分区。

[0190] 不能按照四叉树分区结构、二叉树分区结构和/或三叉树分区结构被分区的编码单元成为用于编码、预测和/或变换的基本单元。即，所述编码单元不能被进一步分区以用于预测和/或变换。因此，在比特流中可能不存在用于将编码单元分区为预测单元和/或变换单元的分区结构信息和分区信息。

[0191] 然而，当编码单元(即，用于分区的基本单元)的尺寸大于最大变换块的尺寸时，可递归地对编码单元进行分区，直到将编码单元的尺寸减小到等于或小于最大变换块的尺寸为止。例如，当编码单元的尺寸为64×64时并且当最大变换块的尺寸为32×32时，可将编码单元分区为用于变换的4个32×32的块。例如，当编码单元的尺寸为32×64并且最大变换块的尺寸为32×32时，可将编码单元分区为用于变换的两个32×32的块。在这种情况下，不单独用信号发送编码单元的用于变换的分区，并且可通过编码单元的水平尺寸或垂直尺寸与最大变换块的水平尺寸或垂直尺寸之间的比较来确定编码单元的用于变换的分区。例如，当编码单元的水平尺寸(宽度)大于最大变换块的水平尺寸(宽度)时，可将编码单元垂直地二等分。例如，当编码单元的垂直尺寸(高度)大于最大变换块的垂直尺寸(高度)时，可将编码单元水平地二等分。

[0192] 编码单元的最大和/或最小尺寸的信息以及变换块的最大和/或最小尺寸的信息可以在编码单元的更高等级被用信号发送或确定。更高等级可以是例如序列级、画面级、条带级、并行块组级、并行块级等。例如，编码单元的最小尺寸可以被确定为4×4。例如，变换块的最大尺寸可以被确定为64×64。例如，变换块的最小尺寸可以被确定为4×4。

[0193] 与四叉树的叶节点相应的编码单元的最小尺寸(四叉树最小尺寸)的信息和/或从多类型树的根节点到叶节点的最大深度(多类型树的最大树深度)的信息可在编码单元的更高等级被用信号发送或确定。例如，更高等级可以是序列级、画面级、条带级、并行块组级、并行块级等。四叉树的最小尺寸的信息和/或多类型树的最大深度的信息可针对画面内条带和画面间条带中的每个被用信号发送或确定。

[0194] 可在编码单元的更高等级用信号发送或确定CTU的尺寸与变换块的最大尺寸之间的差信息。例如，所述更高等级可以是序列级、画面级、条带级、并行块组级、并行块级等。可基于编码树单元的尺寸和所述差信息来确定与二叉树的各个节点相应的编码单元的最大尺寸(在下文中，称为二叉树的最大尺寸)的信息。与三叉树的各个节点相应的编码单元的最大尺寸(在下文中，称为三叉树的最大尺寸)可依据条带的类型而变化。例如，针对画面内条带，三叉树的最大尺寸可以是32×32。例如，针对画面间条带，三叉树的最大尺寸可以是128×128。例如，与二叉树的各个节点相应的编码单元的最小尺寸(在下文中，称为二叉树的最小尺寸)和/或与三叉树的各个节点相应的编码单元的最小尺寸(在下文中，称为三叉树的最小尺寸)可被设置为编码块的最小尺寸。

[0195] 作为另一示例，可在条带级用信号发送或确定二叉树的最大尺寸和/或三叉树的最大尺寸。可选地，可在条带级用信号发送或确定二叉树的最小尺寸和/或三叉树的最小尺寸。

[0196] 依据上述各种块的尺寸和深度信息，四分区信息、多类型树分区指示信息、分区树信息和/或分区方向信息可被包括在比特流中或可不被包括在比特流中。

[0197] 例如，当编码单元的尺寸不大于四叉树的最小尺寸时，编码单元不包含四分区信息。因此，可从第二值推断四分区信息。

[0198] 例如,当与多类型树的节点相应的编码单元的尺寸(水平尺寸和垂直尺寸)大于二叉树的最大尺寸(水平尺寸和垂直尺寸)和/或三叉树的最大尺寸(水平尺寸和垂直尺寸)时,编码单元可不被二叉树分区或三叉树分区。因此,可不用信号发送多类型树分区指示信息,但可从第二值推断多类型树分区指示信息。

[0199] 可选地,当与多类型树的节点相应的编码单元的尺寸(水平尺寸和垂直尺寸)与二叉树的最大尺寸(水平尺寸和垂直尺寸)相同和/或是三叉树的最大尺寸(水平尺寸和垂直尺寸)的两倍大时,编码单元可不被进一步二分区或三分区。因此,可不用信号发送多类型树分区指示信息,而是可从第二值推导多类型树分区指示信息。这是因为当通过二叉树分区结构和/或三叉树分区结构分区编码单元时,产生小于二叉树的最小尺寸和/或三叉树的最小尺寸的编码单元。

[0200] 可选地,可以基于虚拟流水线数据单元的尺寸(在下文中,流水线缓冲器尺寸)来限制二叉树分区或三叉树分区。例如,当通过二叉树分区或三叉树分区将编码单元分区为不适合流水线缓冲器尺寸的子编码单元时,相应的二叉树分区或三叉树分区可能受到限制。流水线缓冲器尺寸可以是最大变换块的尺寸(例如,64×64)。例如,当流水线缓冲器尺寸是64×64时,可以限制下面的分区。

[0201] -用于编码单元的N×M(N和/或M是128)三叉树分区

[0202] -用于编码单元的水平方向的128×N(N<=64)二叉树分区

[0203] -用于编码单元的垂直方向的N×128(N<=64)二叉树分区

[0204] 可选地,当与多类型树的节点相应的编码单元的深度等于多类型树的最大深度时,编码单元可不被进一步二分区和/或三分区。因此,可不用信号发送多类型树分区指示信息,但可从第二值推断多类型树分区指示信息。

[0205] 可选地,仅当垂直方向二叉树分区、水平方向二叉树分区、垂直方向三叉树分区和水平方向三叉树分区中的至少一个对于与多类型树的节点相应的编码单元是可能的时,可用信号发送多类型树分区指示信息。否则,编码单元可以不被二分区和/或三分区。因此,可不用信号发送多类型树分区指示信息,但可从第二值推断多类型树分区指示信息。

[0206] 可选地,仅当垂直方向二叉树分区和水平方向二叉树分区两者或垂直方向三叉树分区和水平方向三叉树分区两者对于与多类型树的节点相应的编码单元是可能的时,才可用信号发送分区方向信息。否则,可不用信号发送分区方向信息,但是可从指示可能的分区方向的值推导分区方向信息。

[0207] 可选地,仅当垂直方向二叉树分区和垂直方向三叉树分区两者或水平方向二叉树分区和水平方向三叉树分区两者对于与多类型树的节点相应的编码树是可能的时,才可用信号发送分区树信息。否则,可以不用信号发送分区树信息,而是从指示可能的分区树结构的值推导分区树信息。

[0208] 图4是示出帧内预测处理的示图。

[0209] 图4中从中心到外部的箭头可表示帧内预测模式的预测方向。

[0210] 可通过使用当前块的邻近块的参考样点来执行帧内编码和/或解码。邻近块可以是重建邻近块。例如,可通过使用包括在重建邻近块中的参考样点的编码参数或值来执行帧内编码和/或解码。

[0211] 预测块可表示通过执行帧内预测而产生的块。预测块可与CU、PU和TU中的至少一

个相应。预测块的单元可具有CU、PU和TU中的一个的尺寸。预测块可以是尺寸为2×2、4×4、16×16、32×32或64×64等的正方形块，或者可以是尺寸为2×8、4×8、2×16、4×16和8×16等的矩形块。

[0212] 可根据针对当前块的帧内预测模式来执行帧内预测。当前块可具有的帧内预测模式的数量可以是固定值，并且可以是根据预测块的属性不同地确定的值。例如，预测块的属性可包括预测块的尺寸和预测块的形状等。

[0213] 不管块尺寸为多少，可将帧内预测模式的数量固定为N。或者，帧内预测模式的数量可以是3、5、9、17、34、35、36、65或67等。可选地，帧内预测模式的数量可根据块尺寸或颜色分量类型或者块尺寸和颜色分量类型两者而变化。例如，帧内预测模式的数量可根据颜色分量是亮度信号还是色度信号而变化。例如，随着块尺寸变大，帧内预测模式的数量可增加。可选地，亮度分量块的帧内预测模式的数量可大于色度分量块的帧内预测模式的数量。

[0214] 帧内预测模式可以是非角度模式或角度模式。非角度模式可以是DC模式或平面模式，并且角度模式可以是具有特定方向或角度的预测模式。帧内预测模式可由模式号、模式值、模式编号、模式角度和模式方向中的至少一个来表示。帧内预测模式的数量可以是大于1的M，包括非角度模式和角度模式。为了对当前块进行帧内预测，可执行确定是否可将包括在重建邻近块中的样点用作当前块的参考样点的步骤。当存在不能用作当前块的参考样点的样点时，通过对包括在重建邻近块中的样点中的至少一个样点值进行复制或执行插值或者执行复制和插值两者而获得的值可被用于替换样点的不可用样点值，因此替换后的样点值被用作当前块的参考样点。

[0215] 图7是示出能够用于帧内预测的参考样点的示图。

[0216] 如图7所示，参考样点线0至参考样点线3中的至少一个可以用于当前块的帧内预测。在图7中，片段A和片段F的样点可以分别利用最接近片段B和片段E的样点被填充，而不是从重建邻近块进行检索。可以用信号发送指示将被用于当前块的帧内预测的参考样点线的索引信息。在当前块的上方边界是CTU的边界时，仅参考样点线0可以是可用的。因此，在这种情况下，可以不用信号发送索引信息。当使用除了参考样点线0之外的参考样点线时，可以不执行稍后将描述的针对预测块的滤波。

[0217] 当进行帧内预测时，可基于帧内预测模式和当前块尺寸/形状将滤波器应用于参考样点和预测样点中的至少一个。

[0218] 在平面模式的情况下，当产生当前块的预测块时，根据预测目标样点在预测块内的位置，可通过使用当前样点的上方参考样点与左侧参考样点以及当前块的右上方参考样点与左下方参考样点的加权和来产生预测目标样点的样点值。另外，在DC模式的情况下，当产生当前块的预测块时，可使用当前块的上方参考样点与左侧参考样点的平均值。另外，在角度模式的情况下，可通过使用当前块的上方参考样点、左侧参考样点、右上方参考样点和/或左下方参考样点来产生预测块。为了产生预测样点值，可执行实数单元的插值。

[0219] 在颜色分量之间的帧内预测的情况下，可以基于第一颜色分量的相应重建块来产生第二颜色分量的当前块的预测块。例如，第一颜色分量可以是亮度分量，并且第二颜色分量可以是色度分量。对于颜色分量之间的帧内预测，可基于模板推导第一颜色分量与第二颜色分量之间的线性模型的参数。模板可包括当前块的上方和/或左侧邻近样点以及与其相应的第一颜色分量的重建块的上方和/或左侧邻近样点。例如，可使用模板中的样点中具

有最大值的第一颜色分量的样点值及与其相应的第二颜色分量的样点值，以及模板中的样点中具有最小值的第一颜色分量的样点值及与其相应的第二颜色分量的样点值推导线性模型的参数。当推导线性模型的参数时，可将相应重建块应用于线性模型以产生当前块的预测块。根据视频格式，可对第一颜色分量的重建块和相应重建块的邻近样点执行二次采样。例如，当第二颜色分量的一个样点与第一颜色分量的4个样点相应时，可对第一颜色分量的4个样点进行二次采样以计算一个相应样点。在这种情况下，可基于相应二次采样的样点执行线性模型的参数推导和颜色分量之间的帧内预测。是否执行颜色分量之间的帧内预测和/或模板的范围可作为帧内预测模式被用信号发送。

[0220] 当前块可在水平方向或垂直方向上被分区为两个子块或4个子块。可顺序地重建分区的子块。即，可以对子块执行帧内预测以产生子预测块。另外，可以对子块执行反量化和/或逆变换以产生子残差块。可通过将子预测块添加到子残差块来产生重建子块。重建子块可以用作子子块的帧内预测的参考样点。子块可以是包括预定数量(例如，16)或更多个样点的块。因此，例如，在当前块是8×4块或4×8块时，当前块可被分区为两个子块。此外，在当前块是4×4块时，当前块可不被分区为子块。在当前块具有其它尺寸时，当前块可被分区为4个子块。可以用信号发送关于是否基于子块和/或分区方向(水平或垂直)执行帧内预测的信息。可以限于仅在使用参考样点线0时执行基于子块的帧内预测。当执行基于子块的帧内预测时，可以不执行稍后将描述的针对预测块的滤波。

[0221] 可以通过对被帧内预测的预测块执行滤波来产生最终预测块。可以通过将预定权重应用于滤波目标样点、左侧参考样点、上方参考样点和/或左上方参考样点来执行滤波。可以基于块尺寸、帧内预测模式和预测块中的滤波目标样点的位置中的至少一个来确定用于滤波的权重和/或参考样点(范围、位置等)。可以仅在预定的帧内预测模式(例如，DC、平面、垂直、水平、对角线和/或相邻对角线模式)的情况下执行滤波。相邻对角线模式可以是给对角线模式加上k或从对角线模式减去k的模式。例如，k可以是8或更小的正整数。

[0222] 可通过预测与当前块相邻存在的块的帧内预测模式来对当前块的帧内预测模式进行熵编码/熵解码。在当前块与邻近块的帧内预测模式相另外，可通过使用预定标志信息来用信号发送当前块与邻近块的帧内预测模式相同的信息。另外，可用信号发送多个邻近块的帧内预测模式之中的与当前块的帧内预测模式相同的帧内预测模式的指示符信息。在当前块与邻近块的帧内预测模式不另外，可通过基于邻近块的帧内预测模式执行熵编码/熵解码来对当前块的帧内预测模式信息进行熵编码/熵解码。

[0223] 图5是示出画面间预测处理的实施例的示图。

[0224] 在图5中，矩形可以表示画面。在图5中，箭头表示预测方向。根据画面的编码类型，可将画面分类为帧内画面(I画面)、预测画面(P画面)和双预测画面(B画面)。

[0225] 可在不需要画面间预测的情况下通过帧内预测对I画面进行编码。可通过使用在针对当前块的一个方向(即，前向或后向)上存在的参考画面，通过画面间预测来对P画面进行编码。可通过使用在针对当前块的两个方向(即，前向和后向)上存在的参考画面，通过画面间预测来对B画面进行编码。当使用画面间预测时，编码器可执行画面间预测或运动补偿，并且解码器可执行相应运动补偿。

[0226] 在下文中，将详细描述帧间预测的实施例。

[0227] 可使用参考画面和运动信息来执行画面间预测或运动补偿。

[0228] 可通过编码设备100和解码设备200中的每个在画面间预测期间推导当前块的运动信息。可通过使用重建的邻近块的运动信息、同位置块(也称为col块或同位块)的运动信息和/或与同位块相邻的块的运动信息来推导当前块的运动信息。同位块可表示先前重建的同位置画面(也称为col画面或同位画面)内的在空间上与当前块位于相同位置的块。同位画面可以是包括在参考画面列表中的一个或更多个参考画面中的一个画面。

[0229] 运动信息的推导方法可依据当前块的预测模式而不同。例如,应用于帧间预测的预测模式包括AMVP模式、合并模式、跳过模式、具有运动矢量差的合并模式、子块合并模式、三角形分区模式、帧间-帧内组合预测模式、仿射模式等。这里,合并模式可以被称为运动合并模式。

[0230] 例如,当AMVP被用作预测模式时,可将重建的邻近块的运动矢量、同位块的运动矢量、与同位块相邻的块的运动矢量和(0,0)运动矢量中的至少一个确定为针对当前块的运动矢量候选,并且通过使用运动矢量候选产生运动矢量候选列表。可通过使用产生的运动矢量候选列表来推导当前块的运动矢量候选。可基于推导的运动矢量候选来确定当前块的运动信息。同位块的运动矢量或与同位块相邻的块的运动矢量可被称为时间运动矢量候选,并且重建的邻近块的运动矢量可被称为空间运动矢量候选。

[0231] 编码设备100可计算当前块的运动矢量与运动矢量候选之间的运动矢量差(MVD),并且可对运动矢量差(MVD)执行熵编码。另外,编码设备100可对运动矢量候选索引执行熵编码并产生比特流。运动矢量候选索引可指示包括在运动矢量候选列表中的运动矢量候选之中的最佳运动矢量候选。解码设备可对包括在比特流中的运动矢量候选索引执行熵解码,并且可通过使用经过熵解码的运动矢量候选索引从包括在运动矢量候选列表中的运动矢量候选中选择解码目标块的运动矢量候选。另外,解码设备200可将经过熵解码的MVD与通过熵解码而提取的运动矢量候选相加,从而推导解码目标块的运动矢量。

[0232] 另外,编码设备100可对计算出的MVD的分辨率信息执行熵编码。解码设备200可使用MVD分辨率信息来调整被熵解码的MVD的分辨率。

[0233] 另外,编码设备100基于仿射模型计算当前块中的运动矢量和运动矢量候选之间的运动矢量差(MVD),并对MVD执行熵编码。解码设备200通过被熵解码的MVD和仿射控制运动矢量候选的总和推导解码的目标块的仿射控制运动矢量来基于每个子块推导运动矢量。

[0234] 比特流可包括指示参考画面的参考画面索引。参考画面索引可通过编码设备100被熵编码,并且随后作为比特流被用信号发送到解码设备200。解码设备200可基于推导的运动矢量和参考画面索引信息来产生解码目标块的预测块。

[0235] 推导当前块的运动信息的方法的另一示例可以是合并模式。合并模式可表示合并多个块的运动的方法。合并模式可表示从邻近块的运动信息推导当前块的运动信息的模式。当应用合并模式时,可使用重建的邻近块的运动信息和/或同位块的运动信息来产生合并候选列表。运动信息可包括运动矢量、参考画面索引和画面间预测指示符中的至少一个。预测指示符可指示单向预测(L0预测或L1预测)或双向预测(L0预测和L1预测)。

[0236] 合并候选列表可以是存储的运动信息的列表。包括在合并候选列表中的运动信息可以是以下至少一个:与当前块相邻的邻近块的运动信息(空间合并候选)、参考画面中的当前块的同位块的运动信息(时间合并候选)、通过合并候选列表中存在的运动信息的组合产生的新运动信息、在当前块之前被编码/解码的块的运动信息(基于历史的合并候选)和

零合并候选。

[0237] 编码设备100可通过对合并标志和合并索引中的至少一个执行熵编码来产生比特流，并且可将比特流用信号发送到解码设备200。合并标志可以是指示是否针对每个块执行合并模式的信息，并且合并索引可以是指示当前块的邻近块中的哪个邻近块是合并目标块的信息。例如，当前块的邻近块可包括位于当前块的左侧的左侧邻近块、被布置在当前块上方的上方邻近块和在时间上与当前块相邻的时间邻近块。

[0238] 另外，编码设备100对合并候选的运动信息中的用于校正运动矢量的校正信息执行熵编码，并将其用信号发送到解码设备200。解码设备200可以基于校正信息校正通过合并索引选择的合并候选的运动矢量。这里，校正信息可以包括关于是否执行校正的信息、校正方向信息和校正尺寸信息中的至少一个。如上所述，基于用信号发送的校正信息对合并候选的运动矢量进行校正的预测模式可以被称为具有运动矢量差的合并模式。

[0239] 跳过模式可以是将邻近块的运动信息照原样应用于当前块的模式。当应用跳过模式时，编码设备100可对哪个块的运动信息将被用作当前块的运动信息的事实的信息执行熵编码，以产生比特流，并且可将比特流用信号发送到解码设备200。编码设备100可不将关于运动矢量差信息、编码块标志和变换系数等级中的至少任意一个的语法元素用信号发送到解码设备200。

[0240] 子块合并模式可以表示以编码块(CU)的子块为单位推导运动信息的模式。当应用子块合并模式时，可使用与参考画面中的当前子块同位的子块的运动信息(基于子块的时间合并候选)和/或仿射控制点运动矢量合并候选来产生子块合并候选列表。

[0241] 三角形分区模式可以表示通过将当前块分区为对角线方向来推导运动信息，使用推导的运动信息中的每个来推导每个预测样点，并且通过对推导的预测样点中的每个进行加权来推导当前块的预测样点的模式。

[0242] 帧间-帧内组合预测模式可以表示通过对由帧间预测产生的预测样点和由帧内预测产生的预测样点进行加权来推导当前块的预测样点的模式。

[0243] 解码设备200可自行校正推导的运动信息。解码设备200可基于由推导的运动信息指示的参考块搜索预定区域，并推导具有最小SAD的运动信息作为校正的运动信息。

[0244] 解码设备200可使用光流对经由帧间预测推导的预测样点进行补偿。

[0245] 图6是示出变换和量化处理的示图。

[0246] 如图6中所示，对残差信号执行变换处理和/或量化处理，以产生量化的等级信号。残差信号是原始块与预测块(即，帧内预测块或帧间预测块)之间的差。预测块是通过帧内预测或帧间预测产生的块。所述变换可以是首次变换、二次变换或者首次变换和二次变换两者。对残差信号的首次变换产生变换系数，并且对变换系数的二次变换产生二次变换系数。

[0247] 从预先定义的各种变换方案中选择的至少一种方案被用于执行首次变换。例如，所述预定的变换方案的示例包括离散余弦变换(DCT)、离散正弦变换(DST)和Karhunen-Loève变换(KLT)。通过首次变换产生的变换系数可经历二次变换。可根据当前块和/或当前块的邻近块的编码参数来确定用于首次变换和/或二次变换的变换方案。可选地，可用信号发送指示变换方案的变换信息。基于DCT的变换可以包括例如DCT-2、DCT-8等。基于DST的变换可包括例如DST-7。

[0248] 可以通过对残差信号或执行首次变换和/或二次变换的结果执行量化来产生量化的等级信号(量化系数)。依据块的帧内预测模式或块尺寸/形状,可以根据对角线右上方扫描、垂直扫描和水平扫描中的至少一个来扫描量化的等级信号。例如,当在对角线右上方扫描中扫描系数时,块形式的系数改变为一维矢量形式。除了对角线右上方扫描之外,依据帧内预测模式和/或变换块的尺寸,可以使用水平地扫描二维块形式的系数的水平扫描或垂直地扫描二维块形式的系数的垂直扫描。扫描的量化的等级系数可以被熵编码以插入比特流中。

[0249] 解码器对比特流进行熵解码以获得量化的等级系数。量化的等级系数可以通过反向扫描以二维块形式被布置。对于反向扫描,可以使用对角线右上方扫描、垂直扫描和水平扫描中的至少一个。

[0250] 然后可以对量化的等级系数进行反量化,然后根据需要进行二次逆变换,最后根据需要进行首次逆变换,以产生重建残差信号。

[0251] 可在环内滤波之前针对通过帧内预测或帧间预测重建的亮度分量执行动态范围中的逆映射。动态范围可以被分区为16个相等的片段,并且可以用信号发送每个片段的映射函数。可在条带级或并行块组级用信号发送映射函数。可以基于映射函数推导用于执行逆映射的逆映射函数。在逆映射区域中执行环内滤波、参考画面存储和运动补偿,并且经由使用映射函数的映射将通过帧间预测产生的预测块被转换到映射区域,然后被用于产生重建块。然而,由于在映射区域中执行帧内预测,因此经由帧内预测产生的预测块可以被用于产生重建块而无需映射/逆映射。

[0252] 在当前块是色度分量的残差块时,可以通过对映射区域的色度分量执行缩放来将残差块转换到逆映射区域。可在条带级或并行块组级用信号发送缩放的可用性。只有当亮度分量的映射可用并且亮度分量的分区和色度分量的分区遵循相同的树结构时,才可以应用缩放。可基于与色差块相应的亮度预测块的样点值的平均值来执行缩放。在这种情况下,在当前块使用帧间预测时,亮度预测块可表示映射的亮度预测块。可通过使用亮度预测块的样点值的平均值所属的片段的索引参考查找表来推导缩放所需的值。最后,通过使用推导的值对残差块进行缩放,可以将残差块转换到逆映射区域。然后,可以在逆映射区域中执行色度分量块恢复、帧内预测、帧间预测、环内滤波和参考画面存储。

[0253] 可以通过序列参数集用信号发送指示亮度分量和色度分量的映射/逆映射是否可用的信息。

[0254] 可以基于指示当前画面中的当前块与参考块之间的位移的块矢量来产生当前块的预测块。以这种方式,用于参考当前画面产生预测块的预测模式被称为帧内块复制(IBC)模式。IBC模式可被应用于M×N(M<=64,N<=64)编码单元。IBC模式可包括跳过模式、合并模式、AMVP模式等。在跳过模式或合并模式的情况下,构建合并候选列表,并且用信号发送合并索引,使得可以指定一个合并候选。指定的合并候选的块矢量可以用作当前块的块矢量。合并候选列表可包括空间候选、基于历史的候选、基于两个候选的平均值的候选和零合并候选中的至少一个。在AMVP模式的情况下,可用信号发送差块矢量。另外,可从当前块的左侧邻近块和上方邻近块推导预测块矢量。可以用信号发送将使用的邻近块的索引。IBC模式中的预测块被包括在当前CTU或左侧CTU中并且被限于已经重建的区域中的块。例如,可以限制块矢量的值,使得当前块的预测块按照编码/解码顺序位于当前块所属的64×64块

之前的三个64×64块的区域中。通过以这种方式限制块矢量的值，可减少根据IBC模式实施方案的存储器消耗和设备复杂度。

[0255] 在一般熵编码器或解码器中，在二进制符号(二进制位)的编码或解码中使用的概率值可以依据先前被编码或解码的二进制值和概率值。因此，为了执行当前二进制位的编码或解码，存在需要完成先前二进制位的熵编码或熵解码和上下文更新的问题。由于该问题，在一般熵编码中吞吐量可能增加。在本说明书中，吞吐量可以指每单位时间将被编码或解码的二进制位的数量。

[0256] 在本发明中，为了解决该问题，可以去除基于按照每个任意块对先前二进制位和当前二进制位的熵编码或熵解码中使用的概率值的依赖性，或者可以对相同的语法元素进行组群。作为去除依赖性和组群的结果，可以提高编码器或解码器的吞吐量。此外，根据本发明，可以使用多条先前二进制信息和概率分布模型同时更新一个或更多个概率值，因此可以提高编码器或解码器的吞吐量。

[0257] 在下文中，根据本发明的实施例，将详细描述通过使用熵编码对图像进行编码或解码的方法。这里，熵编码可指编码器方面的熵编码，并且可指解码器方面的熵解码。

[0258] 图8是示出根据本发明实施例的熵解码器的示图。

[0259] 根据图8，比特流可以被输入到熵解码模块和上下文更新模块。这里，熵解码模块可以指算术熵解码模块或二进制算术熵解码模块。此外，在本发明的实施例中，熵编码/熵解码可以指算术熵编码/熵解码。

[0260] 熵解码模块和上下文更新模块可使用与当前将被解码的语法元素相应的上下文来执行熵解码和上下文更新。在本说明书中，上下文可指已被编码或被解码的语法元素的每个二进制位或二进制值的出现概率信息。上下文更新模块可以执行上下文更新以将当前被解码的概率信息应用于紧接着的后续二进制位的熵解码，并且可以将更新的上下文存储在上下文存储器中。这里，与当前将被解码的语法元素相应的上下文可由上下文建模器被推导。此外，与当前将被解码的语法元素内的二进制位相应的上下文可由上下文建模器被推导。例如，上下文建模器可以包括上下文选择模块和上下文存储器。可以从上下文存储器加载与由上下文选择模块选择的当前二进制位相应的上下文以用于熵解码。在执行熵解码之后，去二值化模块可执行去二值化处理，其中，二进制值中的至少一个被变换为语法元素形式。去二值化模块可对输入二进制值中的至少一个执行去二值化并且可输出语法元素信息。这里，在去二值化处理的同时，当前语法元素和二进制信息可被发送到上下文建模器，并且可用于选择将被解码的后续二进制位的上下文。

[0261] 在根据图8的解码器中，在当前二进制位的解码中使用的概率信息依据先前被解码的二进制值。因此，只有在完成先前二进制位的熵解码和概率更新之后，才开始当前二进制位的熵解码。这可以导致吞吐量增加的问题。

[0262] 图9是示出根据本发明另一实施例的熵解码器的示图。

[0263] 根据本发明的解码器可以不在任意预定义单元(在下文中称为熵编码单元(ECU))内执行上下文更新，以便通过去除对先前二进制位和当前二进制位的概率依赖性来提高吞吐量。即，熵解码器可以基于每个ECU执行语法元素的上下文更新。

[0264] 例如，在本说明书中，ECU可以指具有N×M尺寸的任意块。这里，N和M可以是正整数。

[0265] 作为另一示例,ECU可以是画面、CTU、子画面、并行块、分块、条带、CU、PU、TU、CTB、CB、PB和TB中的至少一个。作为又一示例,ECU可指构成画面、CTU、子画面、并行块、分块、条带、CU、PU、TU、CTB、CB、PB和TB中的至少一个的子块。这里,子块可以指在熵编码或熵解码处理中执行独立变换系数扫描的基本单元。

[0266] 作为又一示例,ECU可以指包括预定义数量的二进制位的单元。这里,二进制位的预定义数量可以是除0之外的正数N。即,可以基于N个二进制位来改变ECU。这里,二进制位的数量N可指除了不执行上下文更新的旁路二进制位的执行上下文更新的上下文二进制位的数量。例如,N可以是通过序列、画面、并行块、条带(片段)头和参数集中的至少一个发送的固定值,或者可以是在编码器和解码器中使用的固定值。作为又一示例,可以基于从编码器发送到解码器的编码参数中的至少一个来确定N。作为又一示例,可基于当前块的编码参数中的至少一个来确定N。例如,可以基于包括当前二进制位的当前块的尺寸、深度、形状、预测模式和变换模式中的至少一个来确定N。这里,块的尺寸可以包括块的水平尺寸和块的垂直尺寸中的至少一个。

[0267] 例如,图9中的解码器可以是图8中的解码器的改进的示例。与图8中的熵解码器相比,图9中的熵解码器可以不包括上下文更新模块。

[0268] 根据图9,熵解码模块可以从上下文存储器加载当前二进制位的上下文,并且可以在没有上下文更新的情况下执行熵解码。这里,可以不在ECU内执行对上下文的更新。根据本发明,不执行对先前二进制位的上下文更新的等待或用于加载上下文的存储器访问操作,从而可以提高熵解码的吞吐量。

[0269] 图10是示出根据本发明实施例的熵编码方法和熵解码方法的示图。

[0270] 根据本发明的实施例的熵编码方法可包括:在步骤S1010a,通过对语法元素进行二值化来产生用于当前块的语法元素的二进制位;在步骤S1020a,更新在当前块的语法元素的熵编码中使用的上下文;以及在步骤S1030a,通过使用产生的二进制位和更新的上下文来产生用于当前块的语法元素的比特流。这里,可以基于每个ECU执行上下文更新。此外,在当前块的语法元素已经被二值化时,省略产生用于语法元素的二进制位的步骤,并且对二值化的语法元素中的二进制位进行熵编码,然后将二值化的语法元素中的二进制位产生为比特流。此外,可仅使用二进制位产生当前块的语法元素的比特流。例如,根据当前块的语法元素的类型或值,可确定是否执行二值化。

[0271] 根据本发明的实施例的熵解码方法可包括:在步骤S1010b,更新在当前块的语法元素的熵解码中使用的上下文;在步骤S1020b,通过基于更新的上下文对比特流进行熵解码来产生用于当前块的语法元素的二进制位;以及在步骤S1030b,通过对产生的二进制位进行去二值化来获得当前块的语法元素。这里,可以基于每个ECU执行上下文更新。此外,在当前块的语法元素已经被二值化时,去二值化的处理被省略,并且使用产生的二进制位中的至少一个来获得当前块的语法元素。例如,根据当前块的语法元素的类型或值,可确定是否执行去二值化。

[0272] 语法元素可包括至少一个二进制位。

[0273] 图11是示出根据本发明的另一实施例的熵编码方法和熵解码方法的示图。

[0274] 根据本发明的实施例的熵编码方法可包括:在步骤S1110a,确定是否对当前块的第一语法元素执行上下文更新;在步骤S1120a,基于所述确定来更新用于对第一语法元素

进行熵编码的上下文;以及在步骤S1130a,通过使用更新的上下文和产生的二进制位来产生当前块的比特流。此外,可仅使用二进制位来产生当前块的语法元素的比特流。

[0275] 这里,可以基于当前块的预编码的预定语法元素或二进制位的数量来确定是否执行上下文更新。

[0276] 根据本发明的实施例的熵解码方法可包括:在步骤S1110b,确定是否对当前块的第一语法元素执行上下文更新;在步骤S1120b,基于所述确定来更新用于对第一语法元素进行熵解码的上下文;以及在步骤S1130b,基于更新的上下文来产生第一语法元素的二进制位。此外,可在不使用更新的上下文的情况下产生用于第一语法元素的二进制位。

[0277] 这里,可以基于当前块的预解码的预定语法元素或二进制位的数量来确定是否执行上下文更新。

[0278] 图12是示出根据本发明实施例的熵编码方法的另一示图。

[0279] 将参照图12详细描述根据本发明的熵编码方法。根据本发明的编码器可以基于每个ECU执行上下文更新。对于画面或并行块的第一ECU,编码器可以在用预定义概率值对上下文进行初始化之后执行熵编码。

[0280] 图12中的实施例可以是假设ECU和CTU的尺寸相同的情况的实施例。因此,编码器可以基于每个CTU执行上下文更新。编码器可通过使用用于识别CTU边界的语法元素(诸如语法元素end_of_tile_flag等)来确定当前CTU的熵编码完成。即,当对语法元素end_of_tile_flag等进行编码时,编码器确定后续语法元素是后续ECU中包括的语法元素,对所有上下文执行更新,并执行熵编码。这里,在一个ECU的熵编码期间,可以不另外更新上下文。

[0281] 例如,当ECU的尺寸等于或小于CTU的尺寸时,编码器计算编码CU的面积并确定语法元素的ECU边界。

[0282] 例如,在ECU的尺寸为32×32的情况下,当完成与32×32区域相应的CU的编码时,编码器确定后续语法元素被包括在后续ECU中。

[0283] 作为另一示例,当ECU的尺寸被定义为最大变换尺寸时,编码器计算编码TU的面积并确定语法元素的ECU边界。

[0284] 例如,在ECU被定义为具有最大变换尺寸64×64的情况下,当与64×64区域相应的TU的编码完成时,编码器确定后续语法元素被包括在后续ECU中。

[0285] 上述面积是示例,并且ECU的面积可以被定义为N×M。这里,N和M可以是正整数。

[0286] 作为又一示例,在ECU指在熵编码处理中经历独立变换系数扫描的子块的情况下,当编码器确定在子块中包括的残差信号相关语法元素的熵编码处理期间子块的索引被改变时,编码器确定ECU被改变。

[0287] 作为又一示例,在ECU指在熵编码处理中经历独立变换系数扫描的子块的情况下,编码器可以使用编码的变换系数的数量来确定ECU边界。

[0288] 作为又一示例,在ECU是包括预定义数量的二进制位的单元的情况下,编码器可以对预设的N个二进制位进行编码,然后可以确定ECU被改变。

[0289] 图13是示出根据本发明实施例的熵解码方法的另一示图。

[0290] 将参照图13详细描述根据本发明的熵解码方法。根据本发明的解码器可以基于每个ECU执行上下文更新。对于并行块的画面的第一ECU,解码器可以在用预定义概率值初始化上下文之后执行熵解码。

[0291] 图13中的实施例可以是假设ECU和CTU的尺寸相同的情况的实施例。因此，解码器可以基于每个CTU执行上下文更新。解码器可通过使用用于识别CTU边界的语法元素(诸如语法元素end_of_tile_flag等)来确定当前CTU的熵解码完成。即，当语法元素end_of_tile_flag等被解码时，解码器确定后续语法元素是后续ECU中包括的语法元素，对所有上下文执行更新，并执行熵解码。这里，在一个ECU的熵解码期间，可以不另外更新上下文。

[0292] 例如，当ECU的尺寸等于或小于CTU的尺寸时，解码器计算解码的CU的面积并确定语法元素的ECU边界。

[0293] 例如，在ECU的尺寸为32×32的情况下，当与32×32区域相应的CU的解码完成时，解码器确定后续语法元素包括在后续ECU中。

[0294] 作为另一示例，当ECU的尺寸被定义为最大变换尺寸时，解码器计算解码TU的面积并确定语法元素的ECU边界。

[0295] 例如，在ECU被定义为具有最大变换尺寸64×64的情况下，当与64×64区域相应的TU的解码完成时，解码器确定后续语法元素包括在后续ECU中。

[0296] 上述面积是示例，并且ECU的面积可以被定义为N×M。这里，N和M可以是正整数。

[0297] 作为又一示例，在ECU指在熵解码处理中经历独立变换系数扫描的子块的情况下，当解码器确定在子块中包括的残差信号相关语法元素的熵解码处理期间子块的索引被改变时，解码器确定ECU被改变。

[0298] 作为又一示例，在ECU指在熵解码处理中经历独立变换系数扫描的子块的情况下，解码器可使用解码的变换系数的数量来确定ECU边界。

[0299] 例如，在子块的尺寸被预定义为4×4、2×8、8×2等的情况下，当以扫描顺序完成16个变换系数相关语法元素的解码时，解码器确定后续残差信号相关语法元素被包括在后续子块(即，ECU)中。这里，可从显式传输的残差信号相关语法元素以及可从语法元素last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix、coded_sub_block_flag、sig_coeff_flag、par_level_flag、abs_level_gtx_flag、abs_remaind、dec_abs_level、coeff_sign_flag等中的至少一个语法元素隐式推导的残差信号的语法元素来计算变换系数的数量。

[0300] 这里，last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix、coded_sub_block_flag、sig_coeff_flag、par_level_flag、abs_level_gtx_flag、abs_remaind、dec_abs_level和coeff_sign_flag中的至少一个可以指用于对当前块的残差信号进行熵编码/解码的语法元素。

[0301] 作为又一示例，在ECU是包括预定义数量的二进制位的单元的情况下，解码器可以对预设的N个二进制位进行解码，然后可以确定ECU被改变。

[0302] 根据本发明的另一实施例，编码器或解码器可仅对包括在ECU中的二进制位执行上下文更新，并且可不对不包括在ECU中的二进制位执行上下文更新。

[0303] 例如，在ECU是包括预定义数量的语法元素或二进制位的单元的情况下，编码器或解码器可以在对当前块进行语法元素编码或解码期间仅对预定义数量的二进制位执行上下文更新，并且可以不对超过预定义数量的二进制位执行上下文更新。

[0304] 例如，当编码器或解码器对当前块的任意语法元素中的至少一个进行熵编码或熵解码时，编码器或解码器仅对预定义数量的语法元素中的至少一个执行上下文更新，并且

不对剩余语法元素中的至少一个执行上下文更新。

[0305] 这里,经历上下文更新的二进制位可以被称为上下文二进制位,并且对上下文二进制位进行编码/解码的处理可以被称为上下文编码。此外,未经历上下文更新的二进制位可被称为旁路二进制位,并且对旁路二进制位进行编码/解码的处理可被称为旁路编码。

[0306] 作为另一示例,编码器或解码器可以计算当前块中的预定语法元素中的至少一个被编码或解码(解析)的次数,以便确定是否对当前块的第一语法元素执行上下文更新。例如,在当前块中的预定语法元素中的至少一个被编码或解码的次数超过预定义值时,编码器或解码器不对第一语法元素执行上下文更新。然而,在当前块中的预定语法元素中的至少一个被解析的次数等于或小于预定义值时,编码器或解码器对第一语法元素执行上下文更新。

[0307] 为了计算第一语法元素被解析的次数,编码器或解码器可以设置用于计算预定语法元素中的至少一个被编码或解码的次数的变量。这里,可以由ECU预定义的二进制位的数量N来确定用于计算对预定语法元素中的至少一个进行编码或解码的次数的变量的初始值。作为另一示例,可以基于当前块的编码参数来确定用于计算对预定语法元素中的至少一个预定语法元素进行编码或解码的次数的变量的初始值。例如,可以基于当前块的尺寸、深度和形状中的至少一个来确定用于计算语法元素被解析的次数的变量的初始值。这里,块的尺寸可以包括块的水平尺寸和块的垂直尺寸中的至少一个。

[0308] 每次对预定语法元素中的至少一个进行编码或解码时,编码器或解码器减小用于计算语法元素被编码或解码的次数的变量的值,从而确定被编码或解码的语法元素的数量。即,当变量变得等于或小于特定值时,编码器或解码器确定预定义数量的语法元素被编码或解码。当针对预定语法元素中的至少一个执行预定义数量的上下文更新时,编码器或解码器对稍后将被编码或解码的第一语法元素执行熵编码或熵解码而无需上下文更新。

[0309] 例如,预定语法元素中的至少一个可以是伴随上下文更新的语法元素。即,即使对任意语法元素执行编码或解码,当语法元素是不伴随上下文更新的语法元素时,在上述执行编码或解码的次数中也不包括语法元素的编码或解码。即,当任意语法元素是不伴随上下文更新的语法元素时,编码器或解码器不减少用于计算预定语法元素被编码或解码的次数的变量。

[0310] 例如,不伴随上下文更新的语法元素可以是coded_sub_block_flag、abs_remainder等。

[0311] 例如,在本发明中,第一语法元素可以是coeff_sign_flag。此外,预定语法元素可以是coeff_sign_flag、sig_coeff_flag、abs_level_gtx_flag和par_level_flag中的至少一个。

[0312] 这里,语法元素coeff_sign_flag可指示变换系数的等级的符号。语法元素sig_coeff_flag可指示特定位置处的量化等级是否为0。语法元素abs_level_gtx_flag可与特定位置处的量化等级的绝对值相关。语法元素par_level_flag可指示特定位置处的量化等级的奇偶性。

[0313] 此外,语法元素coded_sub_block_flag可指示是否存在基于每个子块被编码的量化等级。语法元素abs_remaining可与量化等级的绝对值的残差尺寸值相关。

[0314] 作为又一示例,可仅在以变换跳过模式对当前块进行编码或解码时执行确定是否

对将被编码或解码的语法元素执行上下文更新的上述实施例。即，在当前块不处于变换跳过模式时，编码器或解码器不对将被编码或解码的语法元素执行上下文更新。

[0315] 在下文中，将详细描述上述二值化或去二值化步骤。

[0316] 编码器可在执行熵编码之前执行将将被编码的语法元素变换成二进制串的二值化处理。二值化的二进制串可以从变换的第一二进制值开始按顺序用于上下文更新和熵编码。

[0317] 关于每个语法元素的二值化，可以使用考虑每个语法元素可以具有的值的范围内的每个值的出现概率的最佳方法和由编码器预定义的固定方法中的至少一种。例如，可根据每个语法元素选择性地使用以下方法中的至少一个：截断莱斯二值化、k阶指数哥伦布二值化、有限k阶指数哥伦布二值化、固定长度二值化、一元二值化和截断一元去二值化或截断二元二值化方法。

[0318] 例如，当表示运动矢量的差的绝对值的MVD语法元素的值的概率分布与拉普拉斯概率分布类似时，编码器使用一元二值化执行二值化。

[0319] 作为另一示例，编码器可使用先前针对每个语法元素编码的语法元素的统计信息来改变二值化方法。

[0320] 例如，从条带、并行块或画面的第一块到当前块，使用预定义0阶指数哥伦布二值化方法执行二值化。然而，当编码MVD语法元素值的概率分布指示适合于1阶指数哥伦布的概率分布时，编码器通过使用1阶指数哥伦布二值化来执行MVD语法元素的二值化。

[0321] 同时，解码器可执行使用通过熵解码输出的二进制串来输出语法元素的值的去二值化处理。输出的二进制串可用于上下文更新和解码处理。

[0322] 关于每个语法元素的去二值化，可以使用考虑每个语法元素可以具有的值范围内的每个值的出现概率的最佳方法和由解码器预定义的固定方法中的至少一种。例如，可以根据每个语法元素选择性地使用以下方法中的至少一种：截断莱斯去二值化、k阶指数哥伦布去二值化、有限k阶指数哥伦布去二值化、固定长度二值化、一元去二值化以及截断一元去二值化或截断二元去二值化方法。

[0323] 例如，当表示运动矢量的差的绝对值的MVD语法元素的值的概率分布与拉普拉斯概率分布类似时，解码器使用一元去二值化来执行去二值化。

[0324] 作为另一示例，解码器可使用先前针对每个语法元素被解码的语法元素的统计信息来改变去二值化方法。

[0325] 例如，从条带、并行块或画面的第一块到当前块，使用预定义0阶指数哥伦布二值化方法执行去二值化。然而，当解码的MVD语法元素值的概率分布指示适合于1阶指数哥伦布的概率分布时，解码器通过使用1阶指数哥伦布去二值化来执行MVD语法元素的去二值化。

[0326] 在下文中，将详细描述编码器或解码器的上述上下文更新步骤。

[0327] 根据本发明的实施例，编码器或解码器可以基于每个ECU更新上下文。这里，ECU的尺寸可以基于通过序列、画面、并行块、条带(片段)头和参数集中的至少一个发送的值被确定，或者可以使用CTU的尺寸、最大变换尺寸、CU、PU、TU、CB、PB和TB中的至少一个的尺寸等被推导。

[0328] 此外，ECU的编码/解码顺序可以与具有与ECU相同尺寸的编码单元(CU)或CU内的

变换单元(TU)的编码或解码顺序相同。当基于每个子块定义ECU时,上下文的更新顺序与子块的熵编码或熵解码顺序(扫描顺序)相同。

[0329] 作为另一示例,ECU的编码或解码顺序可以在编码器和解码器之间被预定义,或者可以根据通过序列、画面、并行块、条带(片段)头和参数集中的至少一个发送的值被确定。

[0330] 例如,上下文可以由表示一个二进制位的0或1的出现概率的变量组成,并且可以包括最可能符号(MPS)或最小可能符号(LPS)二进制值及其概率值(或表示概率的索引)。在用于相应语法元素的二进制串中,至少一个上下文可用于特定二进制位,并且因此可提高熵编码效率。在本说明书中,使用基于上下文确定的概率的二进制位可以被定义为常规二进制位,并且使用固定概率(例如,0.5)的二进制位可以被定义为旁路二进制位。此外,在编码或解码处理期间,当通过应用出现的二进制值的统计特征来更新上下文时,编码器或解码器具有更高的编码效率。这里,常规二进制位和上下文二进制位可以具有相同的含义。

[0331] 图14是示出根据本发明的实施例的上下文和二进制串的示图。

[0332] 图14中的表示出运动矢量差以及被分配的上下文的x或y绝对值(abs_mvd)语法元素的二进制串的示例。如图14的下方所示,可以仅为一些二进制索引分配上下文。可不为剩余二进制位分配上下文,并且可在假设0或1的概率为0.5的情况下执行熵编码或熵解码。当为二进制串分配上下文时,每个上下文包括基于每个ECU可更新的概率信息。概率信息可被表示为概率1(或概率0)、MPS和LPS值的概率(或MPS的概率)等。例如,可以以由编码器或解码器预定义的整数值形式的索引值被映射到每个二进制位的形式来表示概率值。

[0333] 例如,可使用关于在对当前ECU的先前ECU进行编码或解码时出现的二进制位的信息来更新上下文,并且可在对当前ECU进行熵编码或熵解码时使用更新的上下文。

[0334] 这里,可以对分配给在对至少一个先前ECU进行编码或解码时出现的所有语法元素的上下文进行更新。作为另一示例,可仅对分配到包括在残差信号相关语法元素中的二进制位的上下文进行更新。

[0335] 例如,针对在先前ECU中的编码或解码期间出现的N个(这里,N>=1)TU或子块的残差信号相关语法元素tu_cbf_luma、tu_cbf_cb、tu_cbf_cr、transform_skip_flag、tu_mts_flag、mts_idx、last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix、coded_sub_block_flag、sig_coeff_flag、par_level_flag、abs_level_gtx_flag、abs_remainder、dec_abs_level和coeff_sign_flag中的至少一个,可以在对当前ECU执行熵编码或熵解码之前更新相应的上下文。

[0336] 例如,当指示是否存在任意TU的将被编码或解码的残差信号的标志(tu_cbf_luma、tu_cbf_cb或tu_cbf_cr)为0时,更新标志,并且省略剩余的残差信号相关语法元素(transform_skip_flag、tu_mts_flag、mts_idx、last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix、coded_sub_block_flag、sig_coeff_flag、par_level_flag、abs_level_gtx_flag、abs_remainder、dec_abs_level、coeff_sign_flag等)的上下文更新。

[0337] 作为另一示例,可以仅对分配给除了比ECU更高概念的语法元素之外的语法元素的二进制位的上下文进行更新。例如,分区标志语法元素可以是指示可以是比ECU更高概念的块(当ECU基于CU或TU时为CTU)是否被分区的语法元素。在这种情况下,在分区标志语法元素中使用上下文的二进制值的上下文可以不基于每个ECU被更新,而是可以每当编码或

解码出现时被更新。为此，编码器和解码器可以对始终更新哪个上下文进行预定义。

[0338] 作为又一示例，在ECU基于子块的情况下，当子块中包括的残差信号相关语法元素的上下文更新被省略并且子块的索引改变时，在子块中包括的残差信号相关语法元素中，在对第一语法元素执行编码或解码之前，对子块中的残差信号相关语法元素执行上下文更新。其它语法元素可基于每个二进制位经历上下文更新。

[0339] 例如，关于包括在子块中的残差信号相关语法元素中的语法元素，可省略使用常规二进制位的语法元素transform_skip_flag、tu_mts_flag、mts_idx、last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix、coded_sub_block_flag、sig_coeff_flag、par_level_flag、abs_level_gtx_flag、abs_remainder、dec_abs_level和coeff_sign_flag中的至少一个的上下文更新，并且可在后续子块的编码或解码开始时执行语法元素的上下文更新。

[0340] 作为又一示例，在ECU是包括预定义数量的二进制位的单元的情况下，可以基于N个常规二进制位来执行上下文更新。

[0341] 例如，可以在没有上下文更新的情况下对当前ECU中的N个二进制位执行熵编码或熵解码。在对后续ECU执行熵编码或熵解码之前，可对N个二进制位的语法元素执行上下文更新。此外，例如，可以在没有上下文更新的情况下对当前ECU中的N个常规二进制位执行熵编码或熵解码。在对后续ECU执行熵编码或熵解码之前，可对N个常规二进制位的语法元素执行上下文更新。

[0342] 作为又一示例，在对当前ECU中的N个常规二进制位执行熵编码或熵解码时执行上下文更新。可不对超过N个常规二进制位的常规二进制位执行上下文更新。这里，常规二进制位可以指上下文二进制位。

[0343] 根据本发明的实施例，基于每个ECU执行上下文更新，使得可以去除对一个ECU内的上下文更新处理和熵编码或熵解码处理的依赖性。因此，在完成所有先前ECU的熵解码之后，解码器可使用解码的语法元素信息和二进制值来更新将用于当前ECU的熵解码的上下文。

[0344] 此外，通过去除对上下文更新处理和熵编码或熵解码处理的依赖性，编码器或解码器可以同时执行当前ECU的熵编码或熵解码和后续ECU的上下文更新。

[0345] 作为又一示例，根据本发明的另一实施例，针对包括在当前块中的ECU的二进制位，对每个二进制位执行上下文更新。可以在没有上下文更新的情况下对不包括在ECU中的二进制位执行熵编码或熵解码。

[0346] 在下文中，将详细描述上下文更新步骤的详细实施例。

[0347] 例如，对于任意语法元素，通过使用在对至少一个先前ECU和更新之前的上下文进行熵编码或熵解码时出现的二进制值，可以根据出现顺序执行上下文更新。更新的上下文可用于执行后续ECU的熵编码或熵解码。

[0348] 图15是示出根据本发明实施例的使用查找表的上下文更新方法的示图。图16是示出根据本发明实施例的使用查找表的上下文更新方法的另一示图。

[0349] 图15示出在基于出现顺序执行上下文更新的方法中的利用。可以基于查找表来执行根据实施例的上下文更新。例如，在上下文的总数量是numCtx、针对每个上下文在先前ECU中出现的二进制位的数量是numDecBin、每个二进制位的值是binVal、并且MPS的值是

valMps的情况下，编码器或解码器可根据二进制值是LPS还是MPS按照二进制值出现的顺序通过参考概率更新表transIdxLps和transIdxMps来连续地更新概率值。通过使用更新最终完成的概率值pStateIdx，可执行当前ECU的熵编码或熵解码处理。

[0350] 图16是示出当MPS和LPS已出现时存储针对每个情况改变的概率索引值的表transIdxMps和transIdxLps的总共64个概率索引的实际概率值的示图。

[0351] 例如，编码器或解码器可以存储在先前ECU中出现的每个上下文的所有二进制值，然后可以同时对所有二进制位执行上下文更新处理。作为另一示例，编码器或解码器可以通过使用在先前ECU的熵编码或熵解码处理中出现的二进制值和相应的上下文信息，在每个二进制位出现之后立即执行上下文更新。

[0352] 图17是示出根据本发明的实施例的使用线性概率更新模型的上下文更新方法的示图。

[0353] 图17示出可以在使用线性概率更新模型的基于出现顺序的上下文更新方法中使用的伪代码的示例。在上下文的总数为numCtx、针对每个上下文在先前ECU中出现的解码的二进制位的数量为numDecBin、并且每个二进制位的值为binVal的情况下，编码器或解码器可根据二进制值是LPS还是MPS按照二进制值出现的顺序通过使用线性概率模型来连续地更新概率。通过使用更新最终完成的概率值pStateIdx，可执行当前ECU的熵编码或熵解码处理。

[0354] 根据图17中的伪代码，编码器或解码器可以通过使用具有不同更新速率的两个线性概率更新模型a和b来独立地执行针对每个概率模型在先前ECU中出现的所有二进制位的上下文更新，其中，这两个线性概率更新模型a和b能够推导概率为1的概率索引。编码器或解码器可将两个最终更新的概率p0StateIdx和p1StateIdx的平均值pStateIdx用于当前ECU的熵编码或熵解码。

[0355] 此外，在小于使用每个上下文的数量numDecBin预定义的阈值的二进制位出现的情况下，编码器或解码器可以将上下文更新速率设置为快。另一方面，在等于或大于阈值的二进制位出现的情况下，编码器或解码器可以将上下文更新速率设置为慢。基于此，编码器或解码器可以基于每个ECU自适应地调整上下文的更新速率，从而提高压缩效率。

[0356] 例如，编码器或解码器可以存储在先前ECU中出现的每个上下文的所有二进制值，然后可以同时对所有二进制位执行上下文更新处理。作为另一示例，编码器或解码器可以通过使用在先前ECU的熵编码或熵解码处理中出现的二进制值和相应的上下文信息，在每个二进制位出现之后立即执行上下文更新。

[0357] 作为另一示例，对于任意语法元素，可以通过使用在对至少一个先前ECU和更新之前的上下文进行熵编码或熵解码时出现的二进制值，根据每个上下文的0和1的出现频率来执行上下文更新。通过使用更新的上下文，可以执行后续ECU的熵编码或熵解码。即，可以基于上下文的0和1的出现频率的差异来更新上下文。

[0358] 图18是示出根据本发明另一实施例的使用查找表的上下文更新方法的示图。

[0359] 图18示出可以用于基于出现频率执行上下文更新的方法的伪代码。可以基于查找表来执行根据实施例的上下文更新。例如，在上下文的总数为numCtx、并且每个上下文的先前ECU中的被编码或被解码的二进制位0和1的数量分别为numDecBin0和numDecBin1的情况下，编码器或解码器可以推导出现频率的差difCount和具有0和1的较大出现频率的二进制

值binVal。编码器或解码器可以使用推导的频率差和查找表来执行上下文更新，并且可以推导最终概率值pStateIdx。最终推导的概率值可以用于当前ECU的熵编码或熵解码。

[0360] 例如，在先前ECU的编码或解码中，使用任意上下文的0出现10次并且1出现6次的情况下，考虑到0出现比1出现多四次，编码器或解码器可以针对上下文考虑0出现四次的状态来执行上下文更新。编码器或解码器可以使用更新的上下文来对当前ECU进行熵编码或熵解码。即，编码器或解码器可以基于出现频率的差来执行上下文更新。编码器或解码器可以基于更新的上下文推导最终概率值，以用于当前ECU的熵解码处理。

[0361] 图19是示出根据本发明的另一实施例的使用线性概率更新模型的上下文更新方法的示图。

[0362] 图19示出可以用于使用线性概率更新模型的基于出现频率的上下文更新方法的伪代码。在上下文的总数是numCtx、并且在每个上下文的先前ECU中出现的被编码或被解码的二进制位中0和1的数量分别是numDecBin0和numDecBin1的情况下，编码器或解码器可以推导出现频率的差difCount和具有大的出现频率的二进制值binVal，并且可以将结果放入线性概率模型以通过单次计算推导概率值pStateIdx。推导的概率值可以用于当前ECU的熵编码或熵解码处理。

[0363] 在图19中，编码器或解码器可以使用具有不同更新速率的两个线性概率更新模型a和b来更新上下文，其中，这两个线性概率更新模型a和b能够推导概率为1的概率索引。这里，假设仅在先前ECU中具有大的频率的二进制位出现，编码器或解码器可以更新上下文。两个最终推导的概率p0StateIdx和p1StateIdx的平均值pStateIdx可用于当前ECU的熵编码或熵解码。

[0364] 作为另一示例，在小于使用每个上下文的出现频率预定义的阈值的二进制位出现的情况下，编码器或解码器可以将上下文更新速率设置为快。另一方面，在等于或大于阈值的二进制位出现的情况下，编码器或解码器可以将上下文更新速率设置为慢。这里，可以使用每个上下文的出现频率之和(numDecBin0和numDecBin1之和)来确定预定义阈值。基于此，编码器或解码器可以基于每个ECU自适应地调整上下文的更新速率，从而提高压缩效率。

[0365] 例如，编码器或解码器可以计算在先前ECU中出现的二进制值的频率差，以同时对所有二进制位执行上下文更新处理。作为另一示例，编码器或解码器可以计算在先前ECU的熵编码或熵解码处理中出现的二进制值的频率差，并且可以在每个二进制位出现之后立即执行上下文更新。

[0366] 作为又一示例，编码器或解码器可推导与至少一个先前ECU的二进制位的概率分布类似的概率分布模型，并且可使用推导的概率分布模型来更新用于当前ECU的熵编码或熵解码的至少一个上下文。

[0367] 图20是示出根据本发明实施例的使用概率分布模型的上下文更新方法的示图。

[0368] 如图20中所示，当先前ECU的abs_mvd的概率分布与0阶指数哥伦布(EG0)类似时，编码器或解码器通过使用对于0阶指数哥伦布二值化最优的概率分布P(x)来改变abs_mvd的两个或更多个二进制位的上下文的概率值。这里，先前ECU的abs_mvd概率分布与0阶指数哥伦布(EG0)类似的情况可以指确定0和1中的每个的出现概率与0阶指数哥伦布二值化最优的概率分布P(x)＝1/(2(x+1)^2)类似的情况。此外，还可以使用P(x)来改变0和1的上下

文的概率值。编码器或解码器可以通过使用概率值改变的上下文来对当前ECU的abs_mvd执行熵编码或熵解码。即,编码器或解码器可定义在任意语法元素中使用的至少一个概率分布模型,可推导与先前ECU中的语法元素的概率分布类似的概率分布模型,并且可在此基础上执行上下文更新。

[0369] 即,编码器或解码器可以使用多条预编码或预解码的二进制信息的概率分布和预定的概率分布模型同时更新概率值,从而提高吞吐量。

[0370] 在下文中,将详细描述上述熵编码或熵解码步骤。

[0371] 根据本发明的实施例,可以在没有上下文更新的情况下对当前ECU内的所有块执行熵编码或熵解码。

[0372] 此外,根据本发明的另一实施例,对于当前块中包括的ECU的语法元素,可对每个语法元素执行上下文更新,并且可执行熵编码或熵解码。对于不包括在ECU中的语法元素,可以在不执行上下文更新的情况下执行熵编码或熵解码。

[0373] 例如,对于除残差信号相关语法元素之外的语法元素,可以在当前ECU内执行上下文更新。

[0374] 作为另一示例,对于与除了比ECU更高概念的语法元素之外的语法元素相应的上下文,可以在当前ECU内执行上下文更新。

[0375] 编码器或解码器可以使用在当前ECU的熵编码或熵解码处理中出现的二进制值和上下文信息来更新将被用于后续ECU的编码或解码的上下文。即,当前ECU的算术熵编码或熵解码处理可以与后续ECU的上下文更新处理同时执行。

[0376] 图21是示出根据本发明的实施例的同时执行熵编码或熵解码处理和上下文更新处理的方法的示图。

[0377] 图21示出以管线形式实现基于每个ECU更新上下文的处理和算术熵解码处理的示例。如图21所示,解码器可以针对画面的第一ECU以预定义值执行上下文初始化,从而执行熵解码。解码器可以通过使用熵解码的语法元素对ECU内的像素执行解码。例如,可以基于ECU内的CU来执行算术熵解码和像素解码。在执行针对第一ECU的熵解码时,可以执行针对第二ECU的上下文更新。类似地,在执行针对第二ECU的算术熵解码时,可以执行针对第三ECU的上下文更新。尽管仅针对解码处理描述了图21中的示例,但是本实施例可以等同地应用于编码处理。即,各个ECU的编码步骤可以由编码器同时执行。

[0378] 当逐步同时执行当前ECU和后续ECU的编码或解码时,可以增加整个熵解码处理的吞吐量。

[0379] 在下文中,将详细描述上述上下文初始化。

[0380] 在当前ECU是画面或条带的第一ECU时,可以执行上下文初始化。在当前ECU是画面、条带或并行块的第一ECU时,使用一个或更多个预定义参数来初始化上下文。

[0381] 例如,通过使用可以接收用于任意上下文的量化参数QP、根据量化参数的概率的变量m、当量化参数为0时的概率n作为参数的函数f,可以推导用于任意上下文的二进制位1的初始概率P1。此外,可使用二进制位1的概率推导0(P0)、MPS和LPS的概率。例如,可根据下面的等式1推导二进制位0和1、MPS以及LPS的概率。

[0382] [等式1]

[0383] $P1=f(QP,m,n),P0=1-p1$

[0384] MPS＝(P1>＝0.5)?1:0,LPS＝!MPS

[0385] 可以根据条带类型、将被编码或解码的图像的类型、应用字段和配置文件中的至少一个将用于推导初始上下文概率的参数定义为具有不同的值。根据该初始设置，可以使用适合于特征(诸如图像类型、应用字段、配置文件等)的初始上下文值，因此可以提高压缩效率。

[0386] 例如，在屏幕内容图像的情况下，跳过模式的出现概率高于自然图像的出现概率。因此，可以将跳过模式的初始概率信息设置为高于自然图像中使用的初始概率的值。

[0387] 当任意语法元素的概率分布与预定义的概率分布模型类似时，编码器或解码器通过考虑语法元素中预定义的二值化方法来推导分配给任意语法元素的至少一个上下文的初始概率值。

[0388] 图22是示出根据本发明实施例的上下文初始化的示图。

[0389] 例如，如图22中所示，当任意语法元素abs_mvd的概率密度分布与一元二值化方法的概率分布P(x)＝2^(-(x-1))(x>＝0)类似时，一元二值化方法的概率密度分布函数P(x)用于执行N(N>0)个二进制值的上下文初始化。

[0390] 下面的等式2是当如上面图14所示定义abs_mvd的二值化方法时，使用概率分布模型P(x)推导二进制索引(i)为0至2的概率C0(i)为0的示例。

[0391] [等式2]

[0392] C0(0)＝P(0)＝0.5

[0393] C0(1)＝P(1)/P(0)＝0.25/0.5＝0.5

[0394] C0(2)＝P(2)/P(1)＝0.125/0.25＝0.5

[0395] 作为另一示例，当针对任意语法元素预定义的二值化方法是一元二值化时，编码器或解码器通过使用一元二值化方法的概率分布P(x)＝2^(-(x-1))(x>＝0)来推导在相应语法元素中使用的所有上下文的初始值。

[0396] 同时，在当前画面支持使用波前并行处理(WPP)的基于CTU列的并行编码和解码时，画面的CTU列0中的ECU 0使用预定义的初始上下文值来初始化所有上下文。同时，CTU列k(这里，k>0)的CTUx内的ECU 0可以使用当CTU列k-1的CTU y(这里，y>x)内的第一ECU被熵编码或熵解码时使用的上下文来经历熵编码或熵解码。

[0397] 图23是示出根据本发明另一实施例的上下文初始化的示图。

[0398] 图23示出使用WPP的基于CTU列的并行熵编码或熵解码方法，其中，WPP应用了根据本发明的上下文初始化。如图所示，可以使用用于作为更高CTU列的CTU列0的ECU 8的熵编码或熵解码的上下文来对CTU列1的ECU 0进行熵编码或熵解码。即，编码器或解码器可以使用在对CTU列0的ECU 7进行熵编码或熵解码时出现的语法元素内的二进制位的出现次数和指示0的出现次数与1的出现次数之间的差的变量中的至少一个来更新上下文。更新的上下文可以用于CTU列0的ECU 8和CTU列1的ECU 0的熵编码或熵解码。

[0399] 在下文中，将描述语法元素组群方法。

[0400] 图24是示出根据本发明实施例的比特流的示图。图25是示出根据本发明另一实施例的比特流的示图。

[0401] 图24中的比特流可以是比特流内的一般语法元素的顺序的示例。如图24所示，为了基于每个CU执行熵解码和重建，每个CU中包括的所有语法元素需要根据每个CU按编码或

解码顺序被发送。这里，当CU被分区成一或更多个TU时，CU中的若干条TU信息按顺序被编码或解码。由于CU内可存在各种语法元素，因此上下文的改变可在熵编码或熵解码期间频繁出现。因此，需要频繁地从外部存储器加载上下文，并且因此可以降低编码器和解码器在熵编码或熵解码中的吞吐量。

[0402] 图25中的比特流可以是改进图24中的比特流的示例。根据本发明的实施例，可以基于每个ECU对语法元素进行编码或解码。ECU可以由至少一个CU组成，使得可以对ECU中包括的CU的相同语法元素进行组群。

[0403] 图25示出通过执行基于ECU的语法元素组群而产生的比特流。例如，当一个ECU内存在两个CU时，编码器针对每个CU对两个pred_mode语法元素进行组群，然后针对每个CU对两个MVD进行组群。之后，编码器可对用于残差信号信息的语法元素进行组群和发送。为了对由基于ECU的语法元素组组成的比特流进行解码，解码器可以将组群的语法元素分离成基于CU的语法元素。

[0404] 例如，编码器或解码器可仅针对残差信号信息相关语法元素执行基于ECU的组群。例如，对于可由至少一个语法元素组成的残差信号信息相关语法元素，编码器或解码器可对相同语法元素执行基于ECU的组群并且可执行编码或解码。例如，当一个ECU内存在四个TU时，编码器或解码器针对四个TU构建组sig_coeff_flag、gt1_flag、gt2_flag和remain_abs_level，并执行编码或解码。

[0405] 根据本发明，可以降低上下文的改变频率，从而可以提高熵编码或熵解码的吞吐量。

[0406] 可以在编码器和解码器中以相同的方法执行以上实施例。

[0407] 以上实施例中的至少一个或组合可以用于对视频进行编码/解码。

[0408] 应用于以上实施例的顺序在编码器和解码器之间可以是不同的，或者应用于上述实施例的顺序在编码器和解码器中可以是相同的。

[0409] 可以对每个亮度信号和色度信号执行以上实施例，或者可以对亮度和色度信号相同地执行以上实施例。

[0410] 应用本发明的以上实施例的块形式可以具有正方形形式或非正方形形式。

[0411] 可以依据编码块、预测块、变换块、块、当前块、编码单元、预测单元、变换单元、单元和当前单元中的至少一个的尺寸来应用本发明的以上实施例。这里，尺寸可以被定义为最小尺寸或最大尺寸或最小尺寸和最大尺寸两者，使得应用以上实施例，或者可以被定义为应用以上实施例的固定尺寸。另外，在以上实施例中，第一实施例可以应用于第一尺寸，并且第二实施例可以应用于第二尺寸。换句话说，可以依据尺寸组合应用以上实施例。另外，当尺寸等于或大于最小尺寸并且等于或小于最大尺寸时，可以应用以上实施例。换句话说，当块尺寸包括在特定范围内时，可以应用以上实施例。

[0412] 例如，在当前块的尺寸为8×8或更大时，可以应用以上实施例。例如，在当前块的尺寸仅为4×4时，可以应用以上实施例。例如，在当前块的尺寸为16×16或更小时，可以应用以上实施例。例如，在当前块的尺寸等于或大于16×16并且等于或小于64×64时，可以应用以上实施例。

[0413] 可以依据时间层来应用本发明的以上实施例。为了识别可应用以上实施例的时间层，可用信号发送相应标识符，并且可将以上实施例应用于由相应标识符识别的指定时间

层。这里，标识符可以被定义为可以应用以上实施例的最低层或最高层或最低层和最高层两者，或者可以被定义为指示应用实施例的特定层。另外，可以定义应用实施例的固定时间层。

[0414] 例如，在当前图像的时间层是最低层时，可以应用以上实施例。例如，在当前图像的时间层标识符是1时，可以应用以上实施例。例如，在当前图像的时间层是最高层时，可以应用以上实施例。

[0415] 可以定义应用本发明的以上实施例的条带类型或并行块组类型，并且可以依据相应的条带类型或并行块组类型来应用以上实施例。

[0416] 在上述实施例中，基于具有一系列步骤或单元的流程图来对方法进行描述，但是本发明不限于步骤的顺序，而是某些步骤可以与其他步骤同时执行或以不同的顺序执行。另外，本领域普通技术人员应当理解，流程图中的步骤并不相互排斥，在不影响本发明的范围的情况下，可以向流程图中添加其他步骤，或者可以从流程图中删除某些步骤。

[0417] 实施例包括示例的各个方面。可以不对用于各个方面的所有可能的组合进行描述，但是本领域技术人员将能够认识到不同的组合。因此，本发明可以包括在权利要求的范围内的所有替换、修改和改变。

[0418] 可以以可由各种计算机组件执行的并且被记录在计算机可读记录介质中的程序指令的形式实现本发明的实施例。计算机可读记录介质可以包括独立的程序指令、数据文件、数据结构等或者程序指令、数据文件、数据结构等的组合。记录在计算机可读记录介质中的程序指令可以是为本发明特别设计和构造的，或者是计算机软件技术领域的普通技术人员公知的。计算机可读记录介质的示例包括：磁性记录介质(诸如硬盘、软盘和磁带)；光学数据存储介质(诸如CD-ROM或DVD-ROM)；磁优化介质(诸如光软盘)；以及被特别地构造成存储和实现程序指令的硬件装置(诸如只读存储器(ROM)、随机存取存储器(RAM)、闪存等)。程序指令的示例不仅包括被编译器格式化的机械语言代码，而且包括可以由计算机使用解释器实现的高级语言代码。硬件装置可以被配置为由一个或更多个软件模块操作以执行根据本发明的处理，或者反之亦然。

[0419] 尽管已经根据诸如详细元件的特定项目以及有限的实施例和附图描述了本发明，但是它们仅被提供以帮助更综合地理解本发明，并且本发明不限于上述实施例。本发明所属领域的技术人员应当理解，可以根据上面的描述进行各种修改和改变。

[0420] 因此，本发明的精神不应被限于上述实施例，并且所附权利要求及其等同物的整个范围将落入本发明的范围和精神内。

[0421] 工业适用性

[0422] 本发明可用于对图像进行编码或解码。

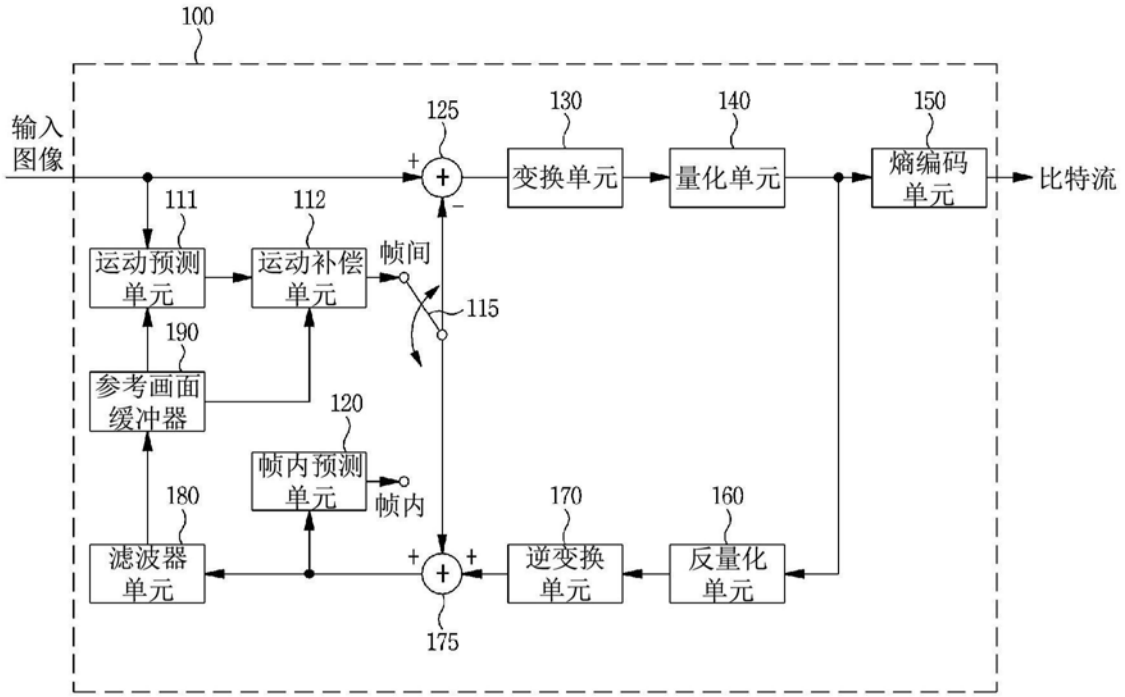
100
输入图像
125
130
140
150
变换单元
量化单元
熵编码单元
比特流
111
112
运动预测单元
运动补偿单元
帧间
115
190
参考画面缓冲器
120
帧内预测单元
帧内
180
170
160
滤波器单元
逆变换单元
反量化单元
175

图1

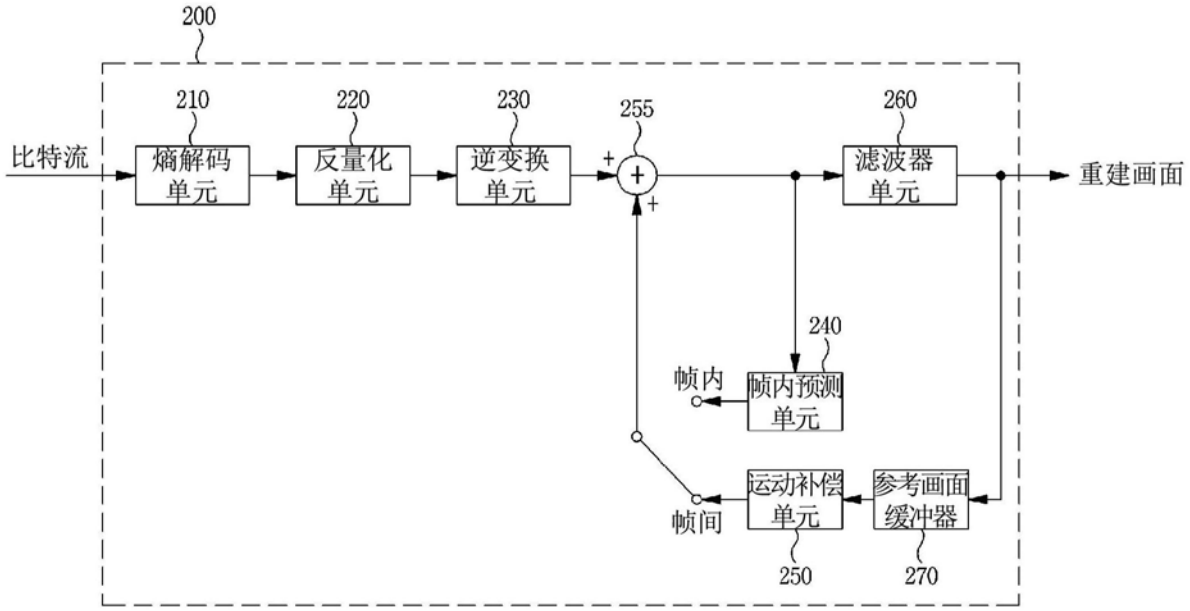
200
210
220
230
255
260
比特流
熵解码单元
反量化单元
逆变换单元
滤波器单元
重建画面
240
帧内
帧内预测单元
运动补偿单元
参考画面缓冲器
帧间
250
270

图2

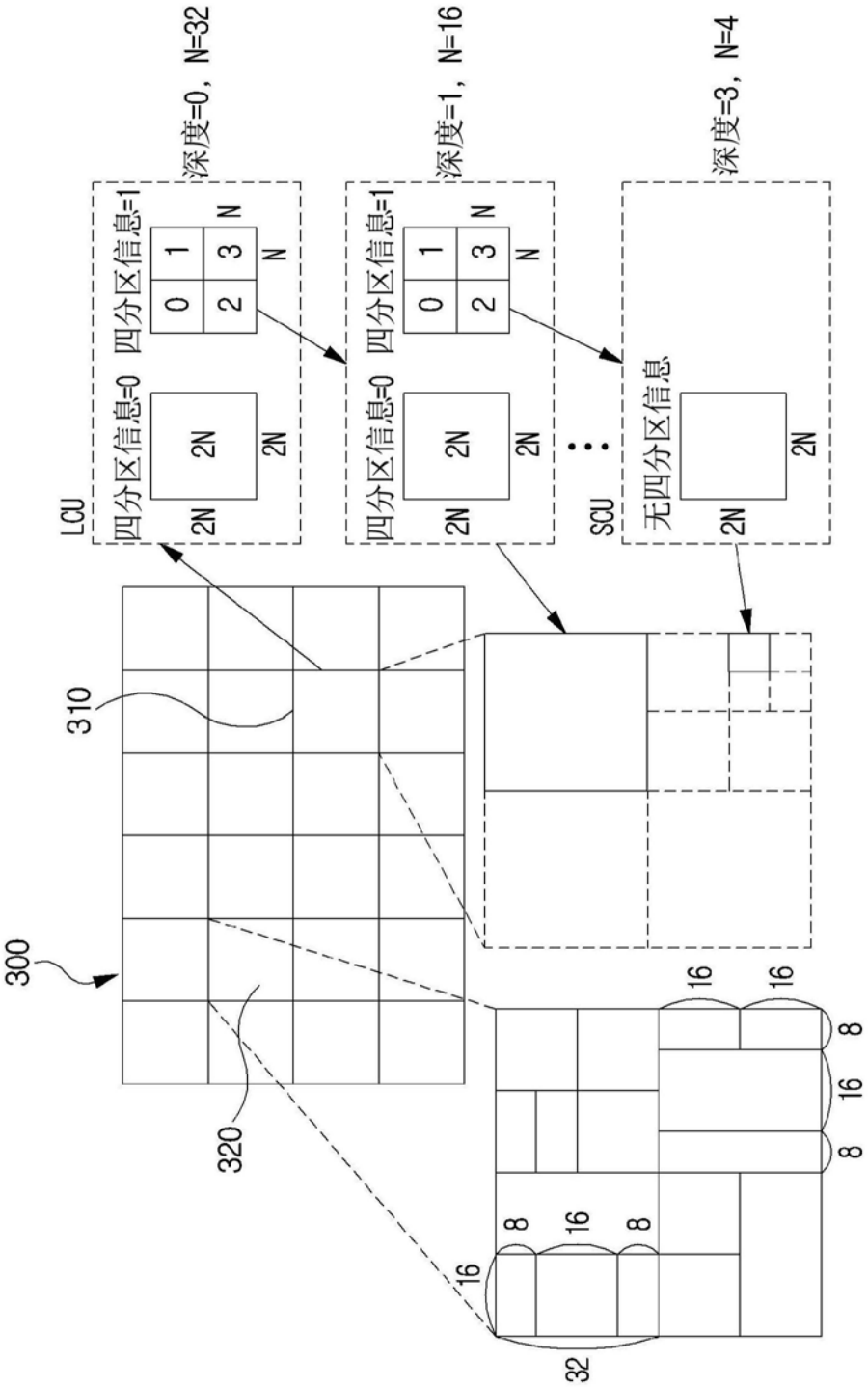
深度=0，N=32
深度=1，N=16
深度=3，N=4
LCU
四分区信息=0
四分区信息=1
0
1
2
3
N
2N
SCU
无四分区信息
310
300
320
16
8
32

图3

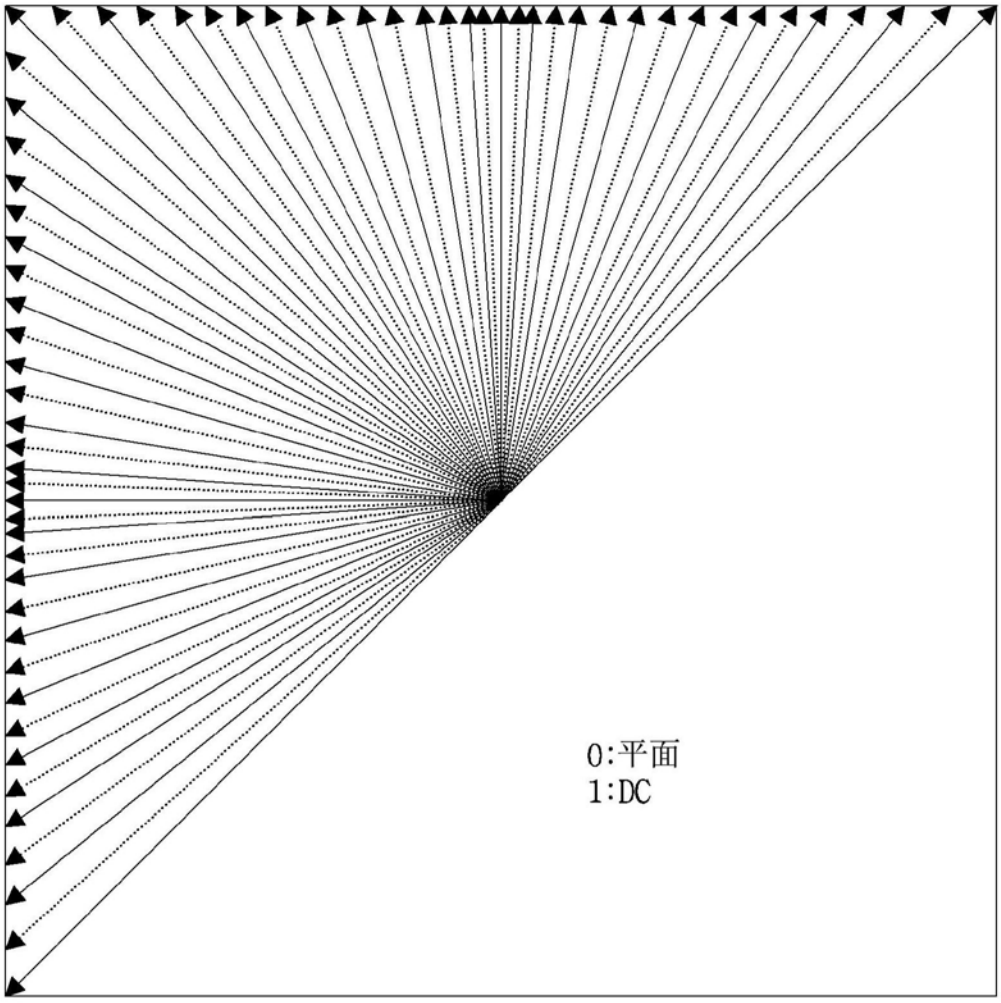

图4

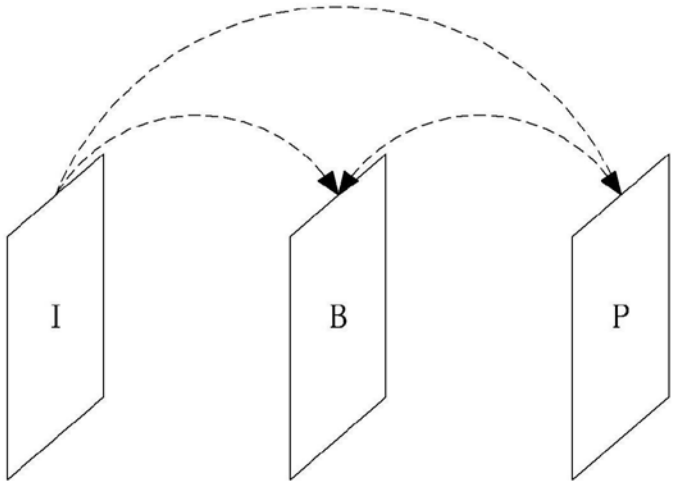

图5

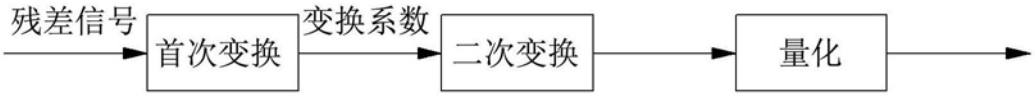
残差信号
首次变换
变换系数
二次变换
量化

图6

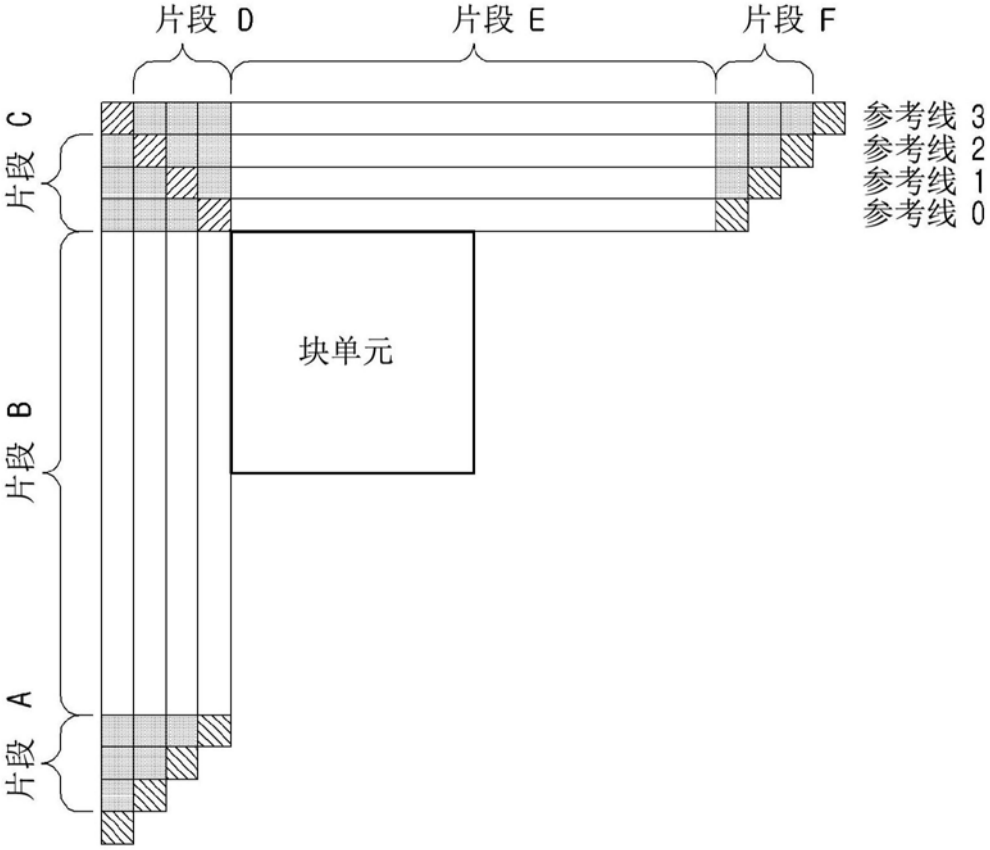
片段 D
片段 E
片段 F
参考线 3
参考线 2
参考线 1
参考线 0
片段 C
片段 B
片段 A
块单元

图7

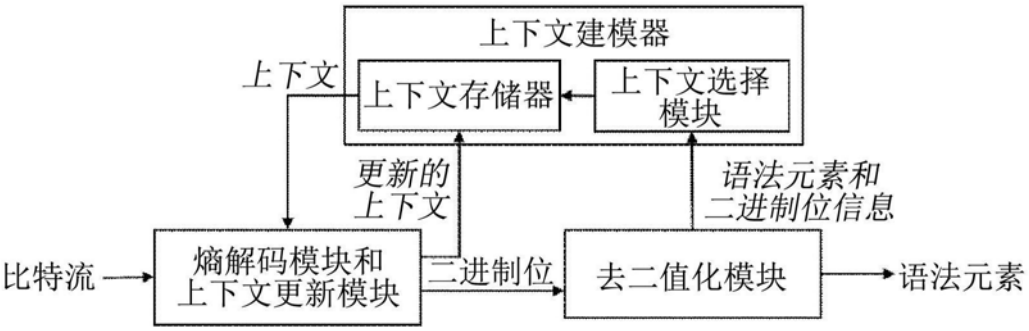
上下文建模器
上下文
上下文存储器
上下文选择模块
更新的上下文
语法元素和二进制位信息
比特流
熵解码模块和上下文更新模块
二进制位
去二值化模块
语法元素

图8

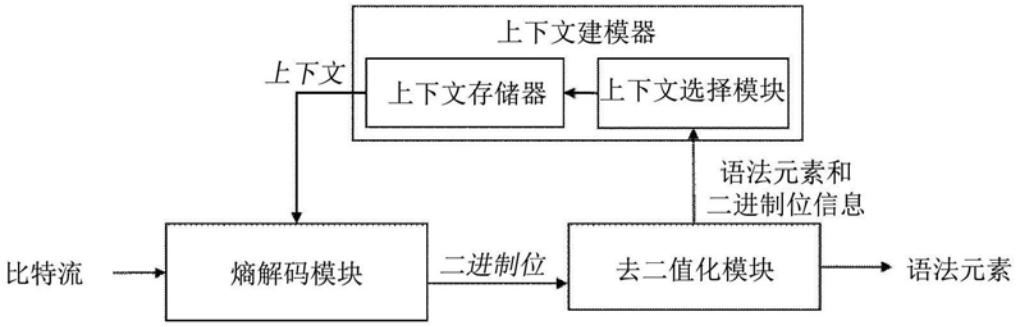
上下文建模器
上下文
上下文存储器
上下文选择模块
语法元素和
二进制位信息
比特流
熵解码模块
二进制位
去二值化模块
语法元素

图9

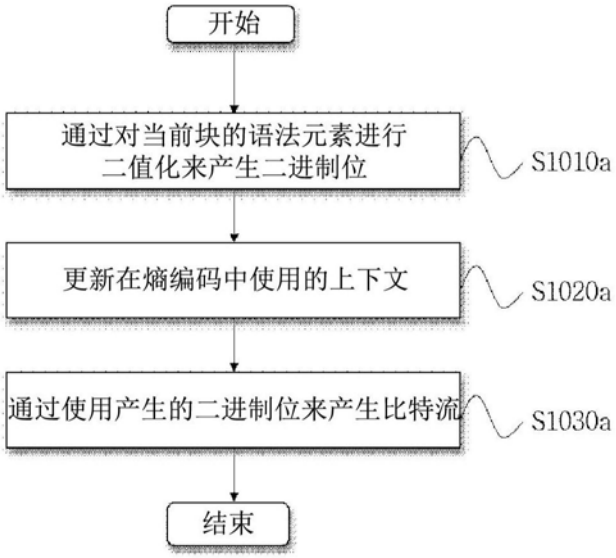
开始
通过对当前块的语法元素进行
二值化来产生二进制位
S1010a
更新在熵编码中使用的上下文
S1020a
通过使用产生的二进制位来产生比特流
S1030a
结束

图10a

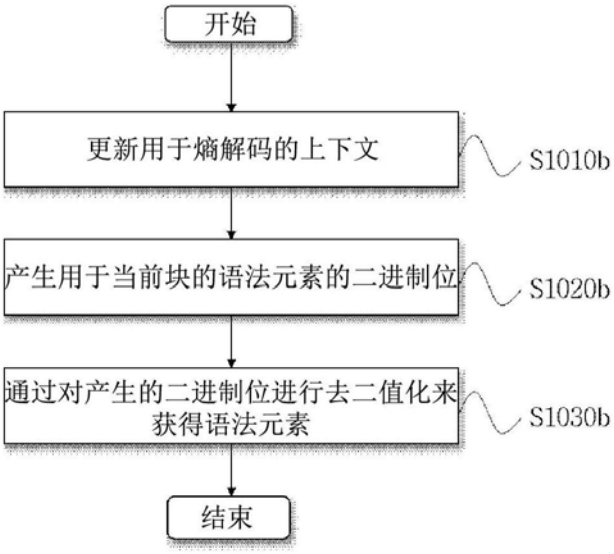
开始
更新用于熵解码的上下文
S1010b
产生用于当前块的语法元素的二进制位
S1020b
通过对产生的二进制位进行去二值化来获得语法元素
S1030b
结束

图10b

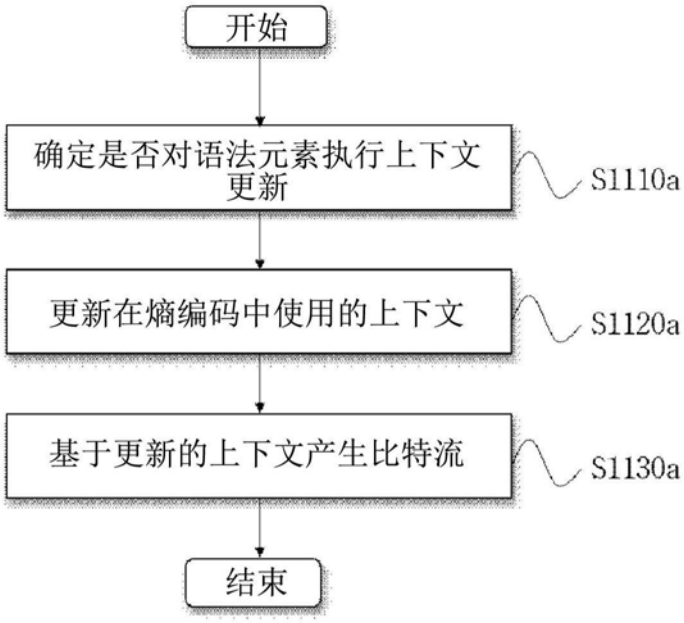
开始
确定是否对语法元素执行上下文更新
S1110a
更新在熵编码中使用的上下文
S1120a
基于更新的上下文产生比特流
S1130a
结束

图11a

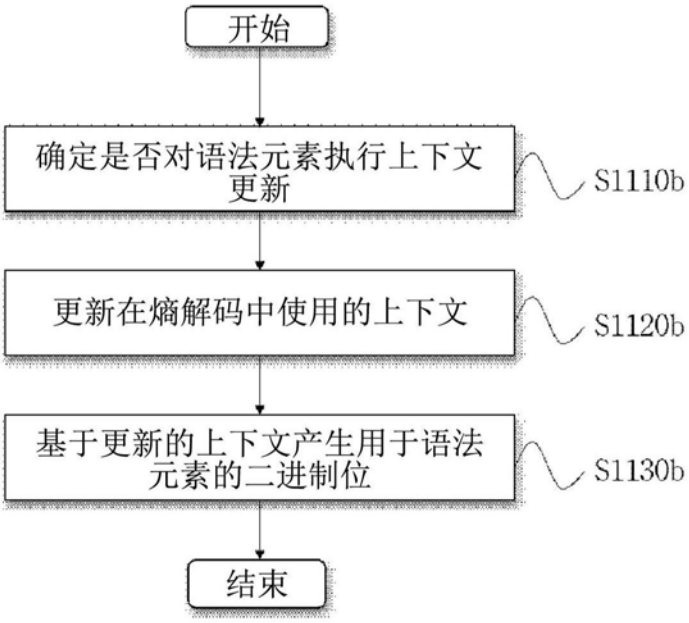
开始
确定是否对语法元素执行上下文更新
S1110b
更新在熵解码中使用的上下文
S1120b
基于更新的上下文产生用于语法元素的二进制位
S1130b
结束

图11b

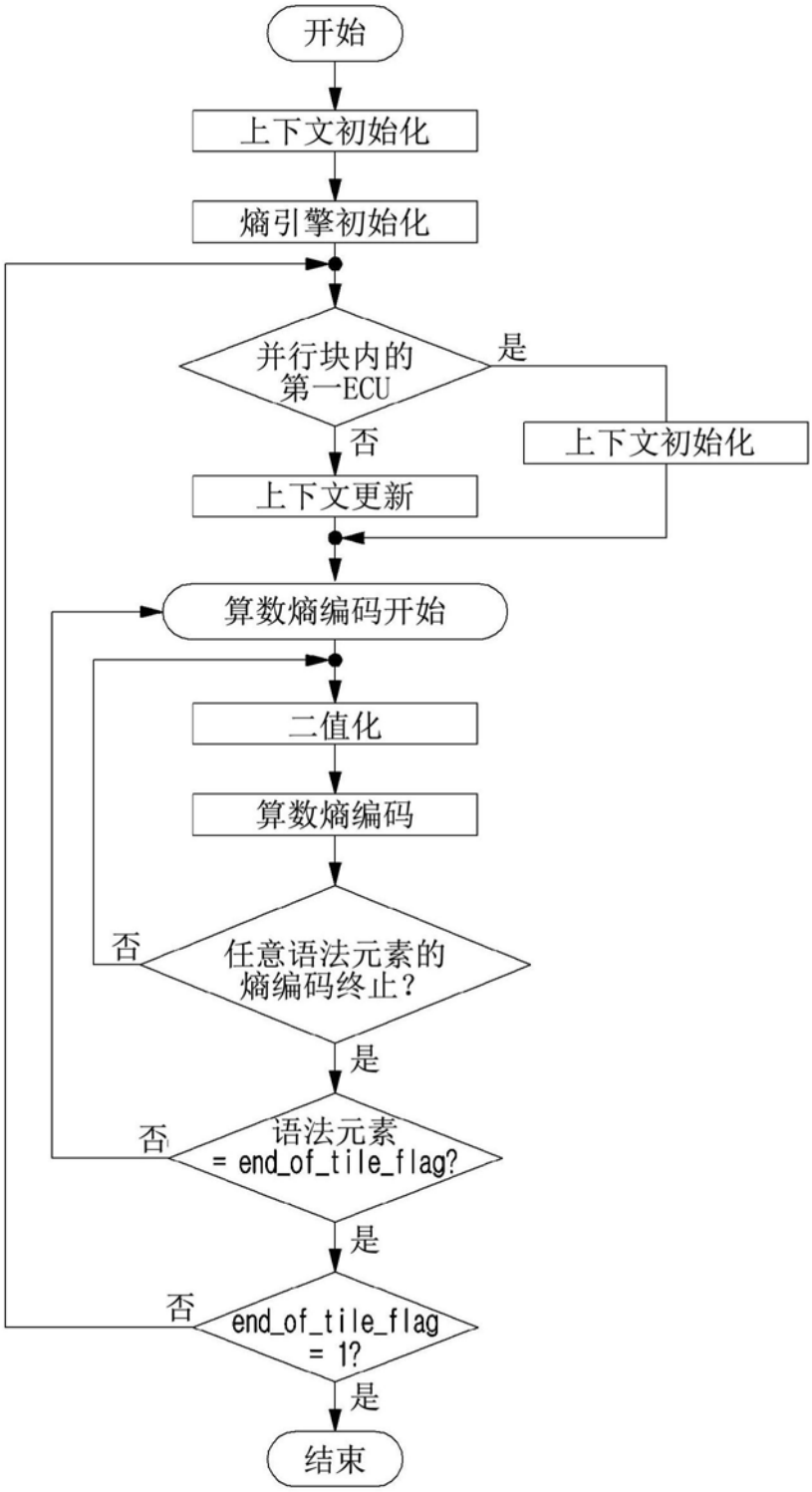
开始
上下文初始化
熵引擎初始化
并行块内的
第一ECU
是
否
上下文初始化
上下文更新
算数熵编码开始
二值化
算数熵编码
否
任意语法元素的
熵编码终止？
是
否
语法元素
= end_of_tile_flag?
是
否
end_of_tile_flag
= 1?
是
结束

图12

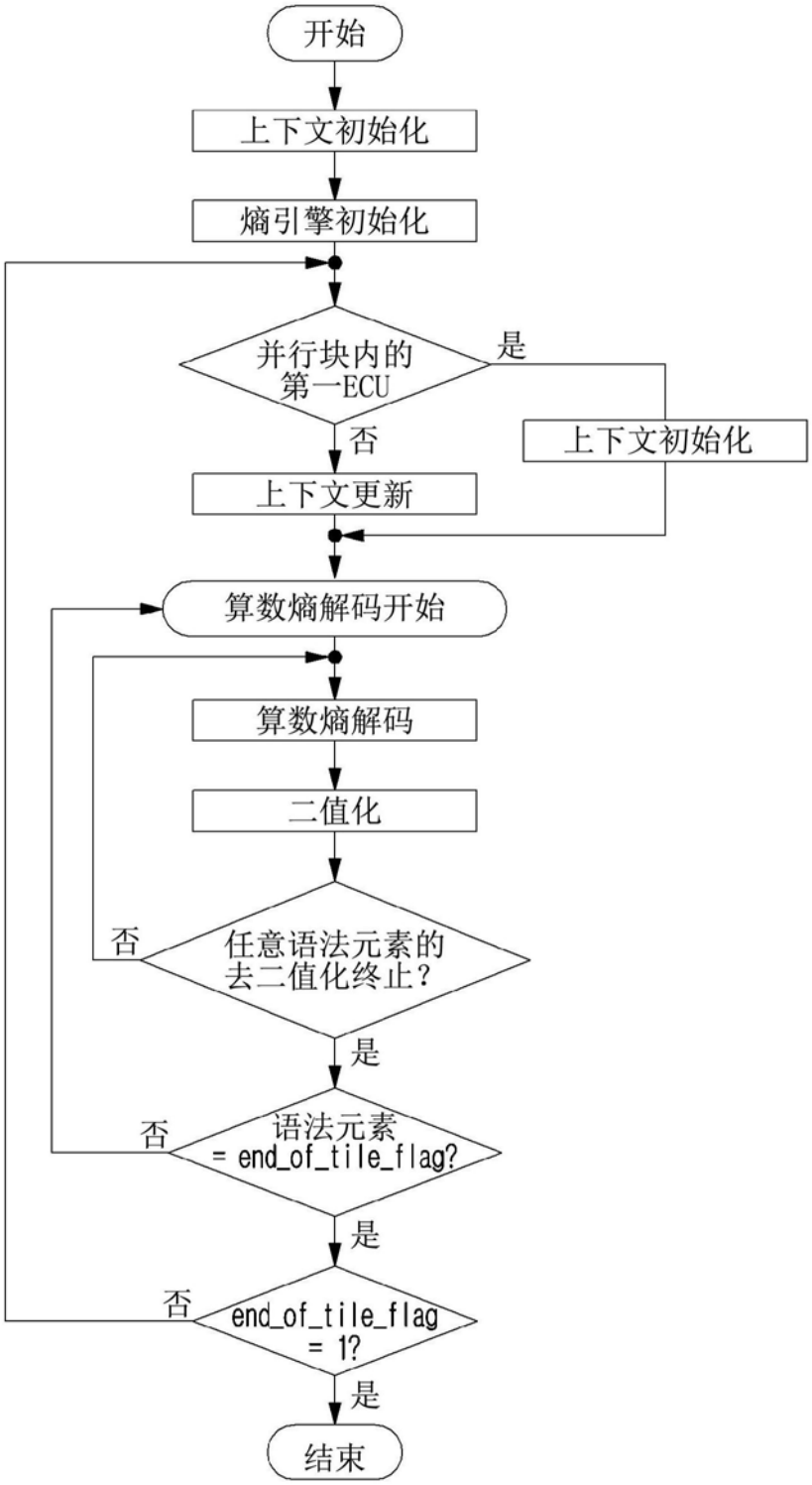
开始
上下文初始化
熵引擎初始化
并行块内的
第一ECU
是
上下文初始化
否
上下文更新
算数熵解码开始
算数熵解码
二值化
否
任意语法元素的
去二值化终止？
是
否
语法元素
= end_of_tile_flag?
是
否
end_of_tile_flag
= 1?
是
结束

图13

	二进制串									
二进制索引 / abs_mvd	0	1	2	3	4	5	6	7	8	9
0	0									
1	1	0								
2	1	1	0							
3	1	1	1	0						
4	1	1	1	1	0					
5	1	1	1	1	1	0				
6	1	1	1	1	1	1	0			
7	1	1	1	1	1	1	1	0		
8	1	1	1	1	1	1	1	1	0	
9	1	1	1	1	1	1	1	1	1	0
...										
上下文索引	0	1	2	3	-	-	-	-	-	-
1的概率	0.6	0.3	0.2	0.1	-	-	-	-	-	-
MPS	1	0	0	0	-	-	-	-	-	-
LPS的概率	0.4	0.3	0.2	0.1	-	-	-	-	-	-

图14

```
for(ctxIdx=0; ctxIdx < numCtx; ctxIdx++)
{
   for(i=0; i < numDecBin[ctxIdx]; i++)
   {
      if( binVal[ctxIdx][i] == valMps[ctxIdx])
         pStateIdx[ctxIdx] = transIdxMps[ pStateIdx[ctxIdx] ]
      else
      {
         if(pStateIdx[ctxIdx] == 0)
            valMps[ctxIdx] = 1 – valMps[ctxIdx]
         pStateIdx[ctxIdx] = transIdxLps[ pStateIdx[ctxIdx] ]
      }
   }
}
```

图15

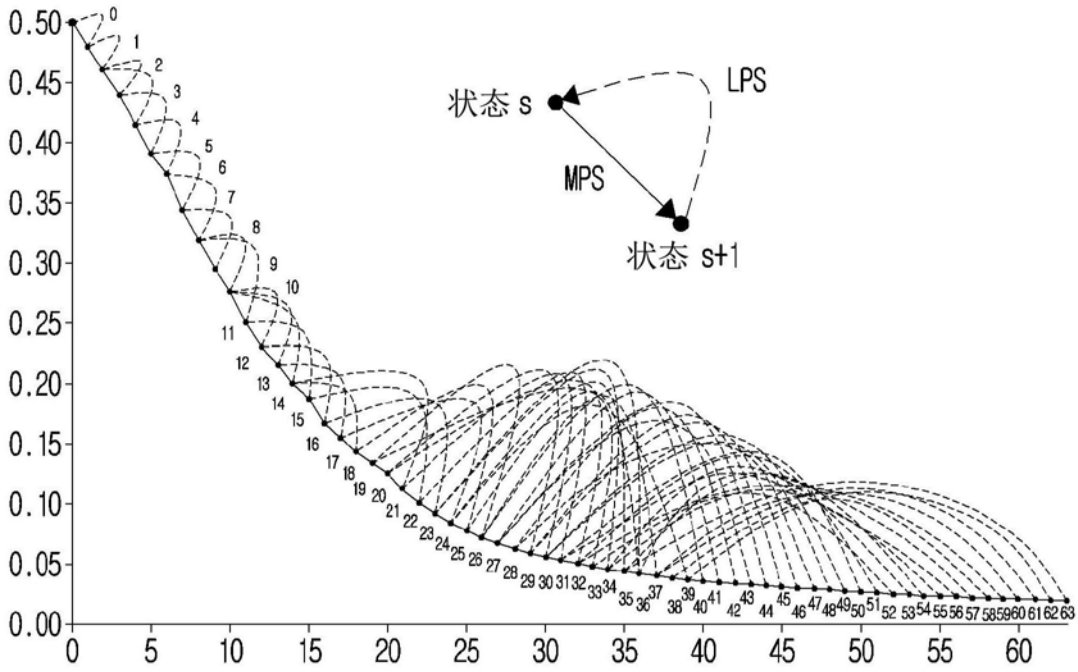

图16

```
for(ctxIdx=0; ctxIdx < numCtx; ctxIdx++)
{
   for(i=0; i < numDecBin[ctxIdx]; i++)
   {
      if( binVal[ctxIdx][i] == 0)
      {
         p0StateIdx[ctxIdx] = p0StateIdx[ctxIdx] – (p0StateIdx[ctxIdx] >> a[ctxIdx] )
         p1StateIdx[ctxIdx] = p1StateIdx[ctxIdx] – (p1StateIdx[ctxIdx] >> b[ctxIdx] )
      }
      else
      {
         p0StateIdx[ctxIdx] = p0StateIdx[ctxIdx] + ((MaxProbIdx – p0StateIdx[ctxIdx]) >> a[ctxIdx] )
         p1StateIdx[ctxIdx] = p1StateIdx[ctxIdx] + ((MaxProbIdx – p1StateIdx[ctxIdx]) >> b[ctxIdx] )
      }
   }
   pStateIdx[ctxIdx] = (p0StateIdx[ctxIdx] + p1StateIdx[ctxIdx] +1)>>1
}
```

图17

```
for(ctxIdx=0; ctxIdx < numCtx; ctxIdx++)
{
   diffCount = abs(numDecBin0[ctxIdx] – numDecBin1[ctxIdx])
   if(numDecBin0[ctxIdx] > numDecBin1[ctxIdx])
      binVal[ctxIdx] = 0
   else
      binVal[ctxIdx] = 1

   for(i=0; i < diffCount; i++)
   {
      if( binVal[ctxIdx] == valMps[ctxIdx])
       pStateIdx[ctxIdx] = transIdxMps[ pStateIdx[ctxIdx] ]
      else
      {
         if(pStateIdx[ctxIdx] == 0)
            valMps[ctxIdx] = 1 – valMps[ctxIdx]
         pStateIdx[ctxIdx] = transIdxLps[ pStateIdx[ctxIdx] ]
      }
   }
}
```

图18

```
for(ctxIdx=0; ctxIdx < numCtx; ctxIdx++)
{
   diffCount = abs(numDecBin0[ctxIdx] – numDecBin1[ctxIdx])
   if(numDecBin0[ctxIdx] > numDecBin1[ctxIdx])
      binVal[ctxIdx] = 0
   else
      binVal[ctxIdx] = 1

   if( binVal[ctxIdx] == 0)
   {
      p0StateIdx[ctxIdx] = p0StateIdx[ctxIdx] – diffCount x (p0StateIdx[ctxIdx] >> a[ctxIdx] )
      p1StateIdx[ctxIdx] = p1StateIdx[ctxIdx] – diffCount x (p1StateIdx[ctxIdx] >> b[ctxIdx] )
   }
   else
   {
      p0StateIdx[ctxIdx] = p0StateIdx[ctxIdx] + diffCount x ((MaxProbIdx – p0StateIdx[ctxIdx]) >> a[ctxIdx] )
      p1StateIdx[ctxIdx] = p1StateIdx[ctxIdx] + diffCount x ((MaxProbIdx – p1StateIdx[ctxIdx]) >> b[ctxIdx] )
   }

   pStateIdx[ctxIdx] = (p0StateIdx[ctxIdx] + p1StateIdx[ctxIdx] +1)>>1
}
```

图19

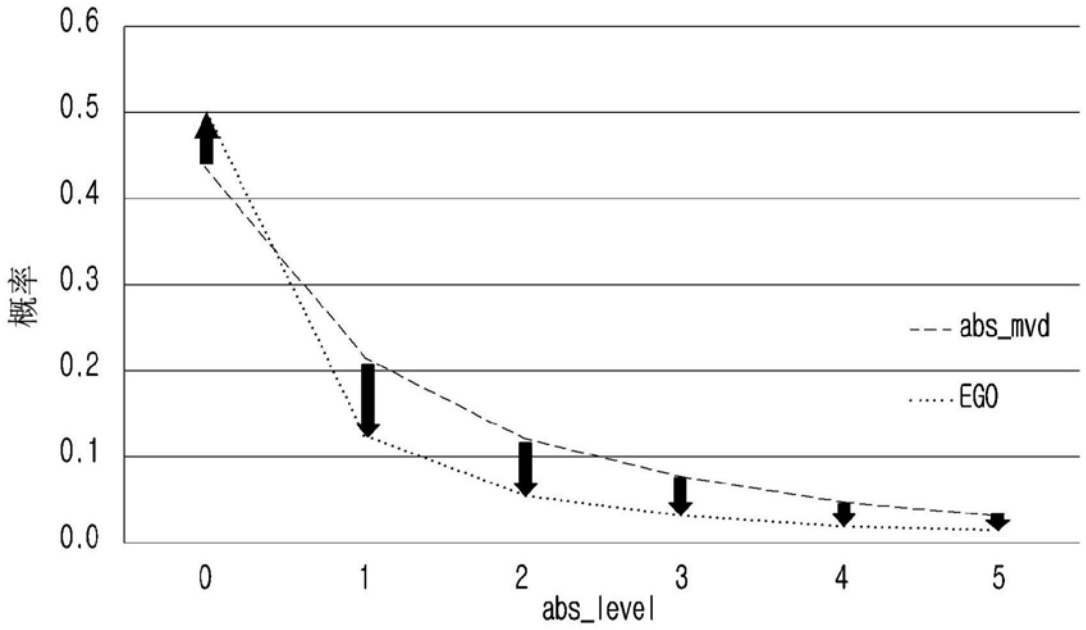

图20

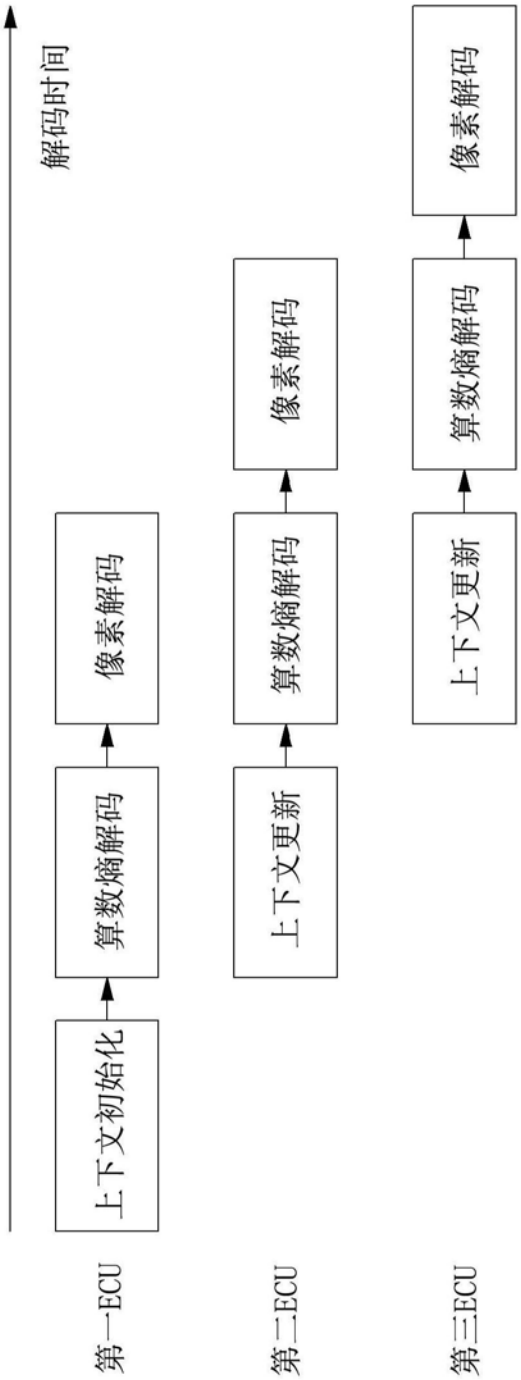
解码时间
第一ECU
上下文初始化
算数熵解码
像素解码
第二ECU
上下文更新
算数熵解码
像素解码
第三ECU
上下文更新
算数熵解码
像素解码

图21

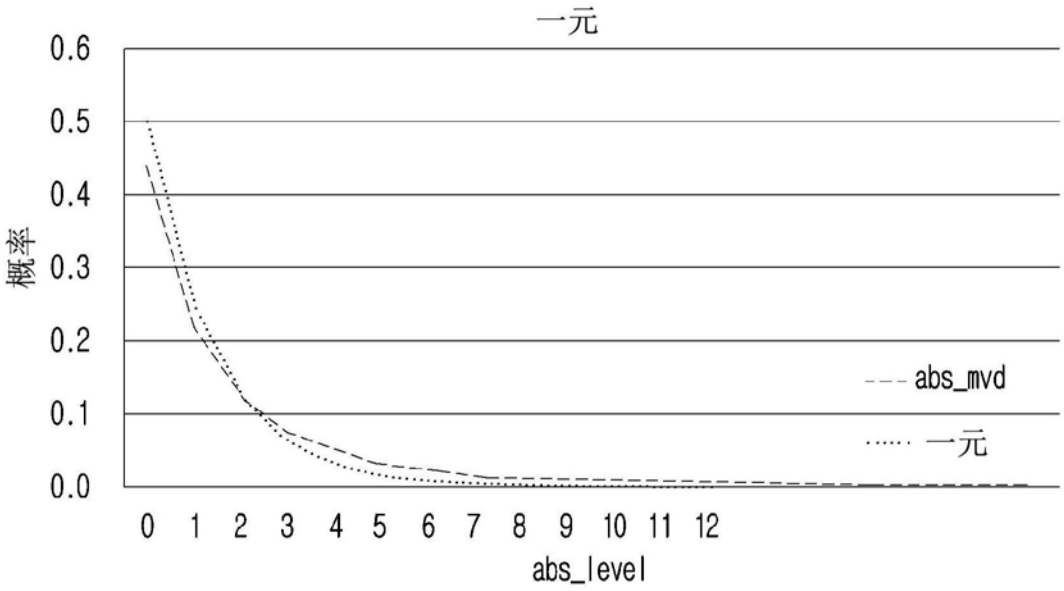
一元
0.6
0.5
0.4
0.3
0.2
0.1
0.0
概率
abs_mvd
一元
0 1 2 3 4 5 6 7 8 9 10 11 12
abs_level

图22

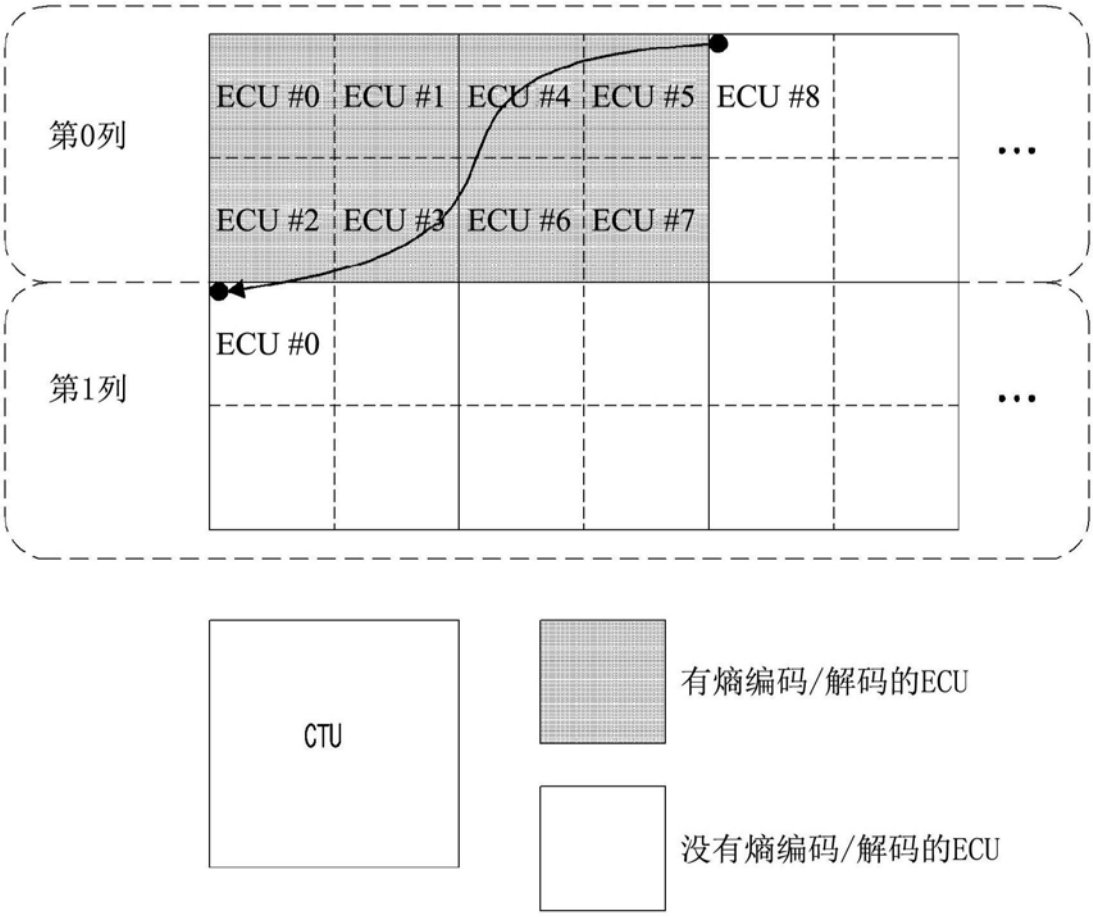
第0列
ECU #0
ECU #1
ECU #4
ECU #5
ECU #8
ECU #2
ECU #3
ECU #6
ECU #7
第1列
ECU #0
CTU
有熵编码/解码的ECU
没有熵编码/解码的ECU

图23

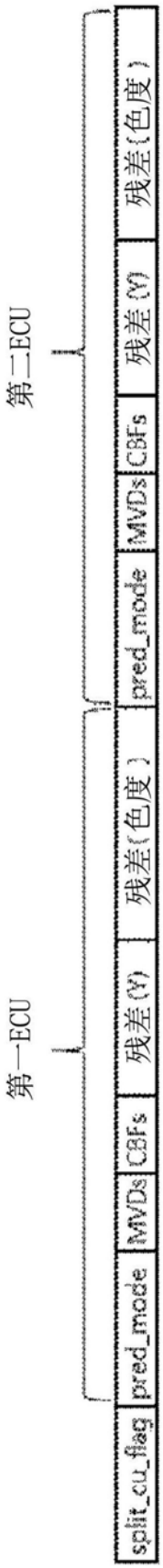
split_cu_flag
pred_mode
MVDs
CBFs
残差(Y)
残差(色度)
pred_mode
MVDs
CBFs
残差(Y)
残差(色度)
第一ECU
第二ECU

图24

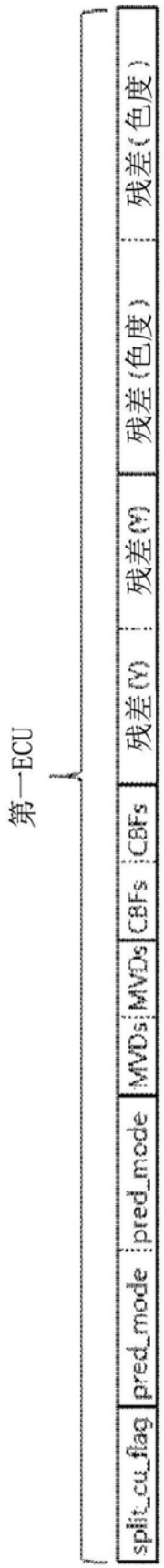
第一ECU
split_cu_flag
pred_mode
pred_mode
MVDs
MVDs
CBFs
CBFs
残差(Y)
残差(Y)
残差(色度)
残差(色度)

图25