



(12) 发明专利申请

(10) 申请公布号 CN 119835418 A

(43) 申请公布日 2025. 04. 15

(21) 申请号 202411987721.6

(22) 申请日 2020.09.10

(30) 优先权数据

10-2019-0112465 2019.09.10 KR

10-2019-0175282 2019.12.26 KR

10-2019-0179609 2019.12.31 KR

(62) 分案原申请数据

202080028560.1 2020.09.10

(71) 申请人 皇家飞利浦有限公司

地址 荷兰艾恩德霍芬

(72) 发明人 李培根 全东山

(74) 专利代理机构 永新专利商标代理有限公司

72002

专利代理师 李光颖

(51) Int.Cl.

H04N 19/119 (2014.01)

H04N 19/174 (2014.01)

H04N 19/105 (2014.01)

H04N 19/122 (2014.01)

H04N 19/60 (2014.01)

H04N 19/70 (2014.01)

H04N 19/176 (2014.01)

H04N 19/44 (2014.01)

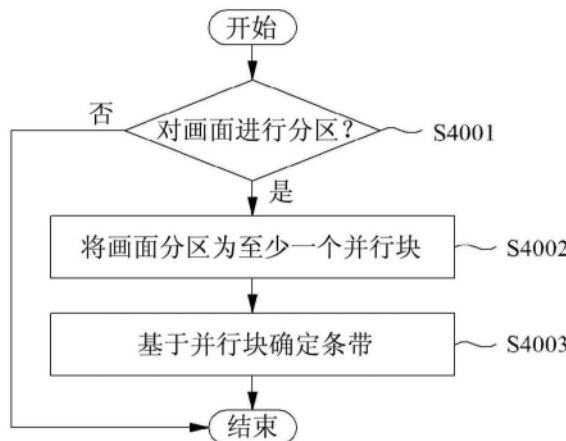
权利要求书3页 说明书91页 附图34页

(54) 发明名称

图像信号编码/解码方法及其装置

(57) 摘要

根据本公开的一种图像信号编码/解码方法及其装置。图像解码方法包括以下步骤：对指示从包括在第*i*条带中的并行块列的数量减去1得到的值的并行块列数量信息进行解析；对指示从包括在第*i*条带中的并行块行的数量减去1得到的值的并行块行数量信息进行解析；并且如果并行块列数量信息和并行块行数量信息都为0，则进一步对与包括第*i*条带的并行块中的高度信息被显式地用信号发送的条带的数量相关的数量信息进行解析。



1. 一种对视频进行解码的方法,所述方法包括:

将当前画面分区为多个并行块,其中,当前画面被分区为多个并行块列和多个并行块行;

确定矩形条带模式是否被应用于当前画面;

当矩形条带模式被应用于当前画面时,从比特流解码条带数量信息;

从比特流解码第一条带的宽度信息,所述宽度信息指示从第一条带所包括的并行块列的数量减去1得到的值;

从比特流解码第一条带的第一高度信息,所述第一高度信息指示从第一条带所包括的并行块行的数量减去1得到的值,

其中,当所述宽度信息和所述第一高度信息都等于0时,第一条带与包括第一条带的并行块相同,或者是所述并行块中包括的多个条带中的一个,

其中,当所述宽度信息和所述第一高度信息都等于0时,从比特流额外地解码显式数量信息,

其中,所述显式数量信息表示第二高度信息被显式地用信号传送的条带的数量,

其中,当所述显式数量信息为0时,第一条带与所述并行块相同,并且不针对第一条带解析所述第二高度信息,

其中,当所述显式数量信息大于0时,基于针对第一条带解析的所述第二高度信息确定第一条带的高度等于或小于所述并行块的高度,针对第一条带解析的所述第二高度信息指示从第一条带所包括的编码树单元CTU行的数量减去1得到的值。

2. 如权利要求1所述的方法,其中,通过对所述并行块中的所述多个条带中的第二条带的索引与所述显式数量信息进行比较来确定是否针对第二条带解析所述第二高度信息。

3. 如权利要求2所述的方法,其中,当确定针对第二条带解析所述第二高度信息时,基于针对第二条带解析的所述第二高度信息来推导第二条带的高度,

其中,当确定针对第二条带不解析所述第二高度信息时,基于针对所述并行块中的另一条带最后解析的所述第二高度信息来推导第二条带的高度。

4. 如权利要求2所述的方法,其中,当第二条带的索引小于所述显式数量信息时,针对第二条带解析所述第二高度信息,

其中,当第二条带的索引等于或大于所述显式数量信息时,针对第二条带不解析所述第二高度信息。

5. 如权利要求1所述的方法,其中,在针对具有第一索引的第一条带解析所述宽度信息和所述第一高度信息以后,进一步针对具有第二索引的第二条带解析所述宽度信息和所述第一高度信息,

其中,当第一条带是所述并行块中的所述多个条带中的一个时,第二索引比第一索引大所述并行块中的所述多个条带的数量,

其中,当包括第一条带的所述并行块未被分区为所述多个条带时,第二索引比第一索引大1。

6. 一种对视频进行编码的方法,所述方法包括:

将当前画面分区为多个并行块,其中,当前画面被分区为多个并行块列和多个并行块行;

对矩形条带模式是否被应用于当前画面的信息进行编码;

当矩形条带模式被应用于当前画面时,将条带数量信息编码到比特流中;

将第一条带的宽度信息编码到比特流中,所述宽度信息指示从第一条带所包括的并行块列的数量减去1得到的值;

将第一条带的第一高度信息编码到比特流中,所述第一高度信息指示从第一条带所包括的并行块行的数量减去1得到的值,其中,响应第一条带的高度等于或小于第一并行块,所述第一高度信息被编码为值0;并且

确定是否额外将显式数量信息编码到比特流中,

其中,当第一条带与包括第一条带的并行块相同,或者是所述并行块中包括的多个条带中的一个时,所述宽度信息和所述第一高度信息被编码为0,

其中,当所述宽度信息和所述第一高度信息都为0时,所述显式数量信息被额外编码到比特流中,所述显式数量信息表示第二高度信息被显式地用信号传送的条带的数量,

其中,当所述显式数量信息被编码为值0时,第一条带与所述并行块相同,

其中,当第一条带是所述并行块中的所述多个条带中的一个时,所述显式数量信息被编码为比0大的值,

其中,针对第一条带的所述第二高度信息被编码为通过从第一条带所包括的编码树单元CTU行的数量减去1得到的值。

7.如权利要求6所述的方法,其中,当所述并行块中的第二条带的索引小于所述显式数量信息时,针对第二条带编码所述第二高度信息,

其中,当所述并行块中的第二条带的索引等于或大于所述显式数量信息时,针对第二条带不编码所述第二高度信息。

8.如权利要求7所述的方法,其中,当针对第二条带不编码所述第二高度信息时,第二条带的高度是以下高度之中的最小值:所述第二高度信息最后被编码的另一条带的高度以及所述并行块中的先前条带占据的区域之外的剩余高度。

9.一种用于发送压缩视频数据的装置,所述装置包括:

处理器,被配置为获得所述压缩视频数据,以及

发送单元,被配置为发送所述压缩视频数据,

其中,获得所述压缩视频数据的操作包括:

将当前画面分区为多个并行块,其中,当前画面被分区为多个并行块列和多个并行块行;

对矩形条带模式是否被应用于当前画面的信息进行编码;

当矩形条带模式被应用于当前画面时,将条带数量信息编码到比特流中;

将第一条带的宽度信息编码到比特流中,所述宽度信息指示从第一条带所包括的并行块列的数量减去1得到的值;

将第一条带的第一高度信息编码到比特流中,所述第一高度信息指示从第一条带所包括的并行块行的数量减去1得到的值,其中,响应第一条带的高度等于或小于第一并行块,所述第一高度信息被编码为值0;并且

确定是否额外将显式数量信息编码到比特流中,

其中,当第一条带与包括第一条带的并行块相同,或者是所述并行块中包括的多个条

带中的一个时,所述宽度信息和所述第一高度信息被编码为0,

其中,当所述宽度信息和所述第一高度信息都为0时,所述显式数量信息被额外编码到比特流中,所述显式数量信息表示第二高度信息被显式地用信号传送的条带的数量,

其中,当所述显式数量信息被编码为值0时,第一条带与所述并行块相同,

其中,当第一条带是所述并行块中的所述多个条带中的一个时,所述显式数量信息被编码为比0大的值,

其中,针对第一条带的所述第二高度信息被编码为通过从第一条带所包括的编码树单元CTU行的数量减去1得到的值。

图像信号编码/解码方法及其装置

[0001] 本申请是申请日为2020年09月10日,申请号为202080028560.1,发明名称为“图像信号编码/解码方法及其装置”的发明专利申请的分案申请。

技术领域

[0002] 本公开涉及一种视频信号编码/解码方法及其装置。

背景技术

[0003] 随着显示面板变得更大,需要更高质量的视频服务。高清视频服务的最大问题是数据量大大地增加。为了解决以上问题,正在积极进行提高视频压缩率的研究。作为代表性示例,视频编码联合协作组(JCT-VC)在2009年由国际电信联盟-电信(ITU-T)下的运动图像专家组(MPEG)和视频编码专家组(VCEG)组建。JCT-VC提出高效视频编码(HEVC),一种压缩性能是H.264/AVC的约两倍且在2013年1月25日被批准为标准的视频压缩标准。然而,随着高清视频服务的快速发展,HEVC的性能逐渐显示出它的局限性。

发明内容

[0004] 技术目的

[0005] 本公开的目的是提供一种在对视频信号进行编码/解码时将画面分区为多个并行块的方法以及用于执行所述方法的装置。

[0006] 本公开的目的是提供一种在对视频信号进行编码/解码时基于并行块分区结构确定条带分区结构的方法以及用于执行所述方法的装置。

[0007] 本公开的目的是提供一种在对视频信号进行编码/解码时将画面分区为多个子画面的方法以及用于执行所述方法的装置。

[0008] 本公开的技术效果可不受上述效果的限制,并且本公开所属技术领域的普通技术人员可从以下描述清楚地理解其他未提及的技术效果。

[0009] 技术方案

[0010] 根据本公开的一种视频信号解码方法包括:对表示从第*i*条带所包括的并行块列的数量减去1得到的值的并行块列的数量信息进行解析,对表示从第*i*条带所包括的并行块行的数量减去1得到的值的并行块行的数量信息进行解析,并且当所述并行块列的数量信息和所述并行块行的数量信息都为0时,另外对与包括第*i*条带的并行块中的高度信息被显式地用信号发送的条带的数量相关的数量信息进行解析。

[0011] 在根据本公开的视频信号解码方法中,当第一条带在并行块中的序号小于所述条带的数量时,与第一条带的高度相关的第一高度信息可另外被解析。

[0012] 在根据本公开的视频信号解码方法中,当第二条带在并行块中的序号等于或大于所述条带的数量时,针对第二条带的高度信息可不被解析。

[0013] 在根据本公开的视频信号解码方法中,第二条带的高度可被设置为基于最后被用信号发送的高度信息推导出的第一高度值和除了由所述并行块中的先前条带占据的区域

之外的第二高度值之中的最小值。

[0014] 在根据本公开的视频信号解码方法中,所述变量*i*可被更新为与从所述条带的数量减去1得到的值相加而获得的值。

[0015] 上面针对本公开简单概述的特征仅仅是之后描述的本公开的详细描述的说明确性方面,并且不限制本公开的范围。

[0016] 技术效果

[0017] 根据本公开,可通过将画面分区为多个并行块来提高编码/解码效率。

[0018] 根据本公开,可通过基于并行块分区结构确定条带分区结构来提高编码/解码效率。

[0019] 根据本公开,可通过将画面分区为多个子画面来提高编码/解码效率。

[0020] 可从本公开获得的效果可不受上述效果的限制,并且本公开所属技术领域的普通技术人员可从以下描述清楚地理解其他未提及的效果。

附图说明

[0021] 图1是示出根据本公开的实施例的视频编码装置(编码器)的框图的示意图。

[0022] 图2是示出根据本公开的实施例的视频解码装置(解码器)的框图的示意图。

[0023] 图3是示出根据本公开的实施例的基本编码树单元的示意图。

[0024] 图4是示出编码块的各种分区类型的示意图。

[0025] 图5是示出对CTU进行分区的方面的示意图。

[0026] 图6是根据本公开的实施例的帧间预测方法的流程图。

[0027] 图7是在合并模式下推导当前块的运动信息的处理的流程图。

[0028] 图8是示出用于推导合并候选的候选块的示意图。

[0029] 图9是示出用于推导合并候选的候选块的示意图。

[0030] 图10是解释运动信息表的更新方面的示意图。

[0031] 图11是示出运动信息表的更新方面的示意图。

[0032] 图12是示出更新保存的运动信息候选的索引的示意图。

[0033] 图13是示出代表性子块的位置的示意图。

[0034] 图14是示出仅针对一部分合并候选执行冗余校验的示意图。

[0035] 图15是示出对特定合并候选的冗余校验被省略的示意图。

[0036] 图16是示出与当前块包括在同一合并处理区域中的候选块被设置为不可用作合并候选的示意图。

[0037] 图17是示出当当前块被包括在合并处理区域中时推导针对当前块的合并候选的示意图。

[0038] 图18是示出临时运动信息表的示意图。

[0039] 图19是示出使运动信息表和临时运动信息表一致的示意图。

[0040] 图20是根据本公开的实施例的帧内预测方法的流程图。

[0041] 图21是示出帧内预测模式的示意图。

[0042] 图22和图23是示出参考样点被排列为一行的一维阵列的示意图。

[0043] 图24是示出由方向帧内预测模式与平行于x轴的直线形成的角度的示意图。

- [0044] 图25是示出在当前块具有非正方形形状的情况下获得预测样点的方面的示意图。
- [0045] 图26是示出广角帧内预测模式的示意图。
- [0046] 图27是示出垂直方向上的分区和水平方向上的分区的示例的示意图。
- [0047] 图28是示出确定编码块的分区形状的示例的示意图。
- [0048] 图29是示出确定编码块的分区形状的示例的示意图。
- [0049] 图30是示出按每个子块确定变换跳过是否被执行的示例的示意图。
- [0050] 图31是示出子块使用相同变换类型的示例的示意图。
- [0051] 图32和图33是示出子变换块编码方法的应用方面的示意图。
- [0052] 图34和图35示出根据作为变换的目标的子块的位置的水平方向变换类型和垂直方向变换类型。
- [0053] 图36是示出当缩减因子为16时的变换系数的编码方面的示意图。
- [0054] 图37和图38是示出作为第二变换的目标的区域的示意图。
- [0055] 图39是示出各种不可分离变换矩阵候选的示例。
- [0056] 图40是示出根据本公开的实施例的画面分区方法的示意图。
- [0057] 图41示出画面被分区为多个并行块的示例。
- [0058] 图42是用于描述条带的尺寸信息被用信号发送的方面的示意图。
- [0059] 图43和图44是用于描述条带的高度信息的编码方面的示意图。
- [0060] 图45是用于描述可被应用于画面的分区形状的示意图。
- [0061] 图46是根据本公开的实施例的将画面分区为至少一个子画面的方法的流程图。

具体实施方式

- [0062] 在下文中,将参照附图详细描述本公开的实施例。
- [0063] 基于块来执行图像编码和解码。在示例中,对于编码块、变换块或预测块,可执行编码/解码处理,诸如变换、量化、预测、环内滤波、重建等。
- [0064] 在下文中,编码/解码目标块被称为“当前块”。在示例中,当前块可表示根据编码/解码的当前处理的编码块、变换块或预测块。
- [0065] 此外,本说明书中使用的术语“单元”表示用于执行特定编码/解码处理的基本单元,并且“块”可被理解为表示具有预定尺寸的样点阵列。除非另有说明,否则“块”和“单元”可被可互换地使用。在示例中,在稍后描述的示例中,编码块和编码单元可被理解为具有彼此相同的含义。
- [0066] 图1是示出根据本公开的实施例的图像编码设备(编码器)的框图的示意图。
- [0067] 参照图1,图像编码设备100可包括画面分区单元110、预测单元120和125、变换单元130、量化单元135、重排单元160、熵编码单元165、反量化单元140、逆变换单元145、滤波器单元150和存储器155。
- [0068] 图1中描述的组件被独立地示出,以便示出图像编码设备中的不同特征功能,并且该图不表示每个组件由单独的硬件或一个软件单元构成。也就是说,每个组件仅是为了便于解释而被列举出的,各个组件中的至少两个组件可构成一个组件,或者一个组件可被划分为可执行其功能的多个组件。甚至对各个组件进行集成的实施例以及对组件进行划分的实施例也被包括在本公开的范围中,除非它们脱离本公开的精神。

[0069] 此外,一些组件不是执行本公开的必要功能的必需部件,而是仅用于提升性能的可选组件。本公开可利用除了仅用于提升性能的组件之外的用于实现本公开的精神的必要组件来实现,并且仅包括除了仅用于提升性能的可选组件之外的必要组件的结构也被包括在本公开的范围之内。

[0070] 画面分区单元110可将输入画面分区为至少一个处理单元。就此而言,处理单元可以是预测单元(PU)、变换单元(TU)或编码单元(CU)。在画面分区单元110中,可将单个画面分区为多个编码单元、预测单元和变换单元的组合,并且可通过根据预定条件(例如,代价函数)选择编码单元、预测单元和变换单元的组合来对画面进行编码。

[0071] 例如,可将单个画面分区为多个编码单元。为了将画面分区为编码单元,可使用诸如四叉树结构的递归树结构,并且源自诸如单个图像或最大编码单元的根的编码单元可被分区为其他编码单元,并且可具有与分区出的编码单元一样多的子节点。根据特定限制不再被分区的编码单元成为叶节点。也就是说,当假设仅正方形分区可用于单个编码单元时,单个编码单元可被分区为最多四个其他编码单元。

[0072] 在下文中,在本公开的实施例中,编码单元可被用作用于编码的单元或可被用作用于解码的单元。

[0073] 可通过将单个编码单元分区为具有相同尺寸的至少一个正方形或矩形来获得预测单元,或者可以以一个预测单元在形状和/或尺寸上可与另一预测单元不同的方式将单个编码单元分区为预测单元。

[0074] 在基于正被执行帧内预测的编码块生成预测单元时,当编码单元不是最小编码单元时,可在不执行分区为多个 $N \times N$ 的预测单元的情况下执行帧内预测。

[0075] 预测单元120和125可包括执行帧间预测的帧间预测单元120以及执行帧内预测的帧内预测单元125。可确定是对预测单元执行帧间预测还是帧内预测,并且可确定根据每种预测方法的详细信息(例如,帧内预测模式、运动矢量、参考画面等)。就此而言,被执行预测的处理单元可与确定了预测方法及其细节的处理单元不同。例如,可基于预测单元来确定预测方法、预测模式等,并且可基于变换单元来执行预测。所生成的预测块与原始块之间的残差值(残差块)可被输入到变换单元130。此外,用于预测的预测模式信息、运动矢量信息等可由熵编码单元165使用残差值来编码,且可被发送到解码器。当使用特定编码模式时,原始块在不通过预测单元120或125生成预测块的情况下被按原样编码并被发送到解码单元。

[0076] 帧间预测单元120可基于关于当前画面的先前画面和后续画面中的至少一个的信息来对预测单元进行预测,或者在一些情况下,可基于关于当前画面中的一些编码区域的信息来对预测单元进行预测。帧间预测单元120可包括参考画面插值单元、运动预测单元和运动补偿单元。

[0077] 参考画面插值单元可从存储器155接收参考画面信息,并且从参考画面生成整数像素或更小像素的像素信息。在亮度像素的情况下,可使用具有不同系数的8抽头的基于DCT的插值滤波器,以便生成关于整数像素或针对 $1/4$ 像素单位的更小像素的像素信息。在色度信号的情况下,可使用具有不同滤波器系数的4抽头的基于DCT的插值滤波器,以便生成关于整数像素或针对 $1/8$ 像素单位的更小像素的像素信息。

[0078] 运动预测单元可基于由参考画面插值单元插值的参考画面来执行运动预测。作为

用于计算运动矢量的方法,可使用各种方法,诸如基于全搜索的块匹配算法(FBMA)、三步搜索(TSS)算法、新三步搜索(NTS)算法等。基于经过插值的像素,运动矢量可具有以1/2像素或1/4像素为单位的运动矢量值。运动预测单元可通过改变运动预测方法来对当前预测单元进行预测。作为运动预测方法,可使用各种方法,诸如跳过方法、合并方法、高级运动矢量预测(AMVP)方法、帧内块复制方法等。

[0079] 帧内预测单元125可基于关于当前块周围的参考像素的信息(是当前画面中的像素信息)生成预测单元。当当前预测单元的邻近块是被执行帧间预测的块,并且因此参考像素是被执行帧间预测的像素时,可通过关于被执行帧内预测的邻近块的参考像素的信息来替换包括在被执行帧间预测的块中的参考像素。换言之,当参考像素不可用时,可使用可用参考像素中的至少一个参考像素来替换不可用参考像素信息。

[0080] 帧内预测中的预测模式可包括根据预测方向使用参考像素信息的方向预测模式以及在执行预测时不使用方向信息的非方向模式。用于对亮度信息进行预测的模式可不同于用于对色度信息进行预测的模式。为了对色度信息进行预测,可使用关于用于对亮度信息进行预测的帧内预测模式的信息或关于预测的亮度信号的信息。

[0081] 在执行帧内预测时,当预测单元的尺寸与变换单元相同时,可基于位于预测单元的左侧、左上方和上方的像素对预测单元执行帧内预测。然而,在执行帧内预测时,当预测单元的尺寸与变换单元不同时,可通过使用参考像素基于变换单元来执行帧内预测。此外,使用 $N \times N$ 分区的帧内预测可仅被用于最小编码单元。

[0082] 在帧内预测方法中,可在根据预测模式将自适应帧内平滑(AIS)滤波器应用于参考像素之后生成预测块。应用于参考像素的AIS滤波器的类型可变化。为了执行帧内预测方法,可从当前预测单元周围存在的预测单元的帧内预测模式对针对当前预测单元的帧内预测模式进行预测。在通过使用从邻近预测单元预测出的模式信息对针对当前预测单元的预测模式进行预测时,当针对当前预测单元的帧内预测模式与邻近预测单元的帧内预测模式相同时,可通过使用预定标志信息来发送指示当前预测单元和邻近预测单元具有相同预测模式的信息。当针对当前预测单元的预测模式与邻近预测单元的预测模式不同时,可执行熵编码以对关于针对当前块的预测模式的信息进行编码。

[0083] 此外,可生成包括关于残差值的信息的残差块,其中,所述残差值是通过预测单元120或125而被执行预测的预测单元与该预测单元的原始块之间的差值。所生成的残差块可被输入到变换单元130。

[0084] 变换单元130可通过使用诸如离散余弦变换(DCT)或离散正弦变换(DST)的变换方法来对残差块执行变换,其中,所述残差块包括关于原始块与由预测单元120或125生成的预测单元之间的残差值的信息。就此而言,DCT变换核包括DCT2或DCT8中的至少一个,并且DST变换核包括DST7。可基于关于用于生成残差块的预测单元的帧内预测模式的信息来确定是应用DCT还是DST以便对残差块执行变换。可跳过针对残差块的变换。可对指示是否跳过针对残差块的变换的标志进行编码。对于尺寸小于或等于阈值的残差块、亮度分量的残差块或4:4:4格式下的色度分量的残差块,可允许变换跳过。

[0085] 量化单元135可对由变换单元130变换到频域的值执行量化。量化系数可根据块或图像的重要性而变化。在量化单元135中计算出的值可被提供给反量化单元140和重排单元160。

[0086] 重排单元160可对针对量化的残差值的系数值执行重排。

[0087] 重排单元160可通过系数扫描方法将二维块形式的系数改变为一维矢量形式的系数。例如,重排单元160可通过使用Z字形扫描方法从DC系数扫描到高频域中的系数,以便将系数改变为一维矢量的形式。根据变换单元的尺寸和帧内预测模式,可使用垂直方向扫描或水平方向扫描,而不是Z字形扫描,其中,在垂直方向扫描中在列方向上对二维块形式的系数进行扫描,在水平方向扫描中在行方向上对二维块形式的系数进行扫描。换言之,可根据变换单元的尺寸和帧内预测模式来确定使用Z字形扫描、垂直方向扫描和水平方向扫描中的哪种扫描方法。

[0088] 熵编码单元165可基于由重排单元160计算出的值来执行熵编码。熵编码可使用各种编码方法,例如,指数哥伦布编码、上下文自适应可变长度编码(CAVLC)或上下文自适应二进制算术编码(CABAC)。

[0089] 熵编码单元165可对从重排单元160以及预测单元120和125获得的各种类型的信息(诸如关于残差值系数的信息和关于编码单元的块类型的信息、关于预测模式的信息、关于分区单元的信息、关于预测单元的信息和关于变换单元的信息、关于运动矢量的信息、关于参考帧的信息、关于块插值的信息、滤波信息等)进行编码。

[0090] 熵编码单元165可对从重排单元160输入的编码单元的系数进行熵编码。

[0091] 反量化单元140可对在量化单元135中量化的值执行反量化,并且逆变换单元145可对在变换单元130中变换的值执行逆变换。由反量化单元140和逆变换单元145生成的残差值可与由包括在预测单元120和125中的运动估计单元、运动补偿单元或帧内预测单元预测出的预测单元相加,以便生成重建块。

[0092] 滤波器单元150可包括去块滤波器、偏移校正单元和自适应环路滤波器(ALF)中的至少一个。

[0093] 去块滤波器可去除由于重建画面中的块之间的边界而发生的块失真。为了确定是否执行去块,可基于块中包括的若干行和若干列中所包括的像素来确定是否将去块滤波器应用于当前块。当去块滤波器被应用于块时,根据所需的去块滤波强度应用强滤波器或弱滤波器。此外,在应用去块滤波器时,当执行水平方向滤波和垂直方向滤波时,水平方向滤波和垂直方向滤波可被配置为被并行处理。

[0094] 偏移校正单元可通过相对于被执行去块的图像的以像素为单位的偏移来校正原始图像。为了对特定画面执行偏移校正,将偏移应用于在将图像的像素分区为预定数量的区域之后确定的区域的方法或者根据每个像素的边缘信息应用偏移的方法可被使用。

[0095] 可基于通过将经过滤波的重建图像与原始图像进行比较而获得的值来执行自适应环路滤波(ALF)。可将图像中包括的像素分区为预定组,将被应用于所述组中的每个组的滤波器可被确定,并且可对每个组单独执行滤波。可针对亮度信号的每个编码单元(CU)发送关于是否应用ALF的信息,并且将被应用的ALF滤波器的形状和滤波器系数可基于每个块而变化。可选地,可应用具有相同形状(固定的形状)的ALF滤波器,而不管将被应用滤波器的块的特征如何。

[0096] 在存储器155中,可存储通过滤波器单元150计算出的重建块或画面。当执行帧内预测时,可将所存储的重建块或画面提供给预测单元120或125。

[0097] 图2是示出根据本公开的实施例的图像解码设备(解码器)的框图的示图。

[0098] 参照图2,图像解码设备200可包括:熵解码单元210、重排单元215、反量化单元220、逆变换单元225、预测单元230和235、滤波器单元240和存储器245。

[0099] 当从编码器输入了图像比特流时,可根据图像编码设备的逆处理对输入的比特流进行解码。

[0100] 熵解码单元210可根据由图像编码器的熵编码单元进行的熵编码的逆处理来执行熵解码。例如,与由图像编码器设备执行的方法相关联,可应用各种方法,诸如指数哥伦布编码、上下文自适应可变长度编码(CAVLC)或上下文自适应二进制算术编码(CABAC)。

[0101] 熵解码单元210可对关于由编码器执行的帧内预测和帧间预测的信息进行解码。

[0102] 重排单元215可基于在编码器中使用的重排方法对由熵解码单元210熵解码的比特流执行重排。以一维矢量形式表示的系数可被重建并被重排为二维块形式的系数。重排单元215可通过下述方法来执行重排:接收与在编码器中执行的系数扫描相关的信息并基于在编码器中执行的扫描顺序进行逆扫描。

[0103] 反量化单元220可基于从编码器接收到的量化参数和经过重排的块的系数值来执行反量化。

[0104] 逆变换单元225可执行与由图像编码器中的变换单元对量化结果执行的变换(即,DCT或DST)相反的逆变换(即,逆DCT或逆DST)。就此而言,DCT变换核可包括DCT2或DCT8中的至少一个,并且DST变换核可包括DST7。可选地,当在图像编码器中跳过了变换时,也不在逆变换单元225中执行逆变换。可基于由图像编码器确定的变换单元来执行逆变换。图像解码器的逆变换单元225可根据诸如预测方法、当前块的尺寸、预测方向等的多条信息选择性地执行变换方法(例如,DCT或DST)。

[0105] 预测单元230或235可基于从熵解码单元210接收到的与预测块相关的信息以及从存储器245接收到的关于先前解码的块或画面的信息来生成预测块。

[0106] 如上所述,作为图像编码器的操作,在执行帧内预测时,当预测单元的尺寸与变换单元相同时,可基于位于预测单元的左侧、左上方和上方的像素对预测单元执行帧内预测。然而,在执行帧内预测时,当预测单元的尺寸与变换单元不同时,可通过使用参考像素基于变换单元来执行帧内预测。此外,使用 $N \times N$ 分区的帧内预测可仅被用于最小编码单元。

[0107] 预测单元230和235可包括PU确定模块、帧间预测单元和帧内预测单元。PU确定模块可接收从熵解码单元210输入的各种类型的信息(诸如关于预测单元的信息、关于帧内预测方法的预测模式的信息、关于帧间预测方法的运动预测的信息等),对当前编码单元中的预测单元进行划分,并且确定对预测单元执行帧间预测还是帧内预测。通过使用从图像编码器接收到的在当前预测单元的帧间预测中所需的信息,帧间预测单元230可基于关于包括当前预测单元的当前画面的先前画面和后续画面中的至少一个的信息来对当前预测单元执行帧间预测。可选地,可基于关于包括当前预测单元的当前画面中的一些预重建区域的信息来执行帧间预测。

[0108] 为了执行帧间预测,可基于编码单元确定将跳过模式、合并模式、AMVP模式或帧内块复制模式中的哪种方法用作针对包括在编码单元中的预测单元的运动预测方法。

[0109] 帧内预测单元235可基于关于当前画面内的像素的信息来生成预测块。当预测单元是已被执行帧内预测的预测单元时,可基于从图像编码器接收到的关于预测单元的帧内预测模式的信息来执行帧内预测。帧内预测单元235可包括自适应帧内平滑(AIS)滤波器、

参考像素插值模块或DC滤波器。AIS滤波器可对当前块的参考像素执行滤波,并且可根据针对当前预测单元的预测模式来确定是否应用滤波器。当对当前块的参考像素执行AIS滤波时,可使用从图像编码器接收到的关于AIS滤波器的信息和预测单元的预测模式。当针对当前块的预测模式是不应用AIS滤波的模式时,可不应用AIS滤波器。

[0110] 当预测单元的预测模式是基于通过对参考像素进行插值而获得的像素值执行帧内预测的预测模式时,参考像素插值单元可对参考像素进行插值以便生成具有整数或更小的单位的参考像素。当针对当前预测单元的预测模式是在不对参考像素进行插值的情况下生成预测块的预测模式时,可不对参考像素进行插值。当针对当前块的预测模式是DC模式时,DC滤波器可通过滤波来生成预测块。

[0111] 可将重建块或重建画面提供给滤波器单元240。滤波器单元240可包括去块滤波器、偏移校正模块和ALF。

[0112] 可从图像编码器接收关于去块滤波器是否已被应用于对应块或对应画面的信息以及关于当去块滤波器被应用时是应用强滤波器还是弱滤波器的信息。图像解码器的去块滤波器可从图像编码器接收关于去块滤波器的信息,并且图像解码器可对对应块执行去块滤波。

[0113] 偏移校正单元可基于在执行编码时应用于图像的偏移校正的类型、关于偏移值的信息等对重建图像执行偏移校正。

[0114] 可基于从编码器接收到的关于是否应用ALF的信息、关于ALF系数的信息等将ALF应用于编码单元。以上ALF信息可通过被包括在特定参数集中而被提供。

[0115] 在存储器245中,可存储重建画面或重建块以使用作参考画面或参考块,并且可将重建画面提供给输出单元。

[0116] 图3是示出根据本公开的实施例的基本编码树单元的示意图。

[0117] 最大编码块可被定义为编码树块。可将单个画面分区为多个编码树单元(CTU)。CTU可以是最大尺寸的编码单元,并且可被称作最大编码单元(LCU)。图3是示出将单个画面分区为多个CTU的示例的示意图。

[0118] 可在画面级或序列级定义CTU的尺寸。同样,可通过画面参数集或序列参数集用信号发送表示CTU的尺寸的信息。

[0119] 在示例中,针对序列内的整个画面的CTU的尺寸可被设置为 128×128 。可选地,可将 128×128 或 256×256 中的任意一个确定为画面级的CTU的尺寸。在示例中,CTU可被设置为在第一画面具有 128×128 的尺寸,并且在第二画面具有 256×256 的尺寸。

[0120] 可通过对CTU进行分区来生成编码块。编码块表示用于执行编码/解码的基本单元。在示例中,可针对每个编码块执行预测或变换,或者可针对每个编码块确定预测编码模式。就此而言,预测编码模式表示生成预测图像的方法。在示例中,预测编码模式可包括帧内预测、帧间预测、当前画面参考(CPR)、帧内块复制(IBC)或组合预测。对于编码块,可通过使用帧内预测、帧间预测、当前画面参考或组合预测中的至少一个的预测编码模式来生成编码块的预测块。

[0121] 可以以比特流用信号发送表示针对当前块的预测编码模式的信息。在示例中,该信息可以是表示预测编码模式是帧内模式还是帧间模式的1比特标志。当针对当前块的预测编码模式被确定为帧间模式时,当前画面参考或组合预测可以是可用的。

[0122] 当前画面参考将当前画面设置为参考画面,并且从当前画面内的已被编码/解码的区域获得当前块的预测块。就此而言,当前画面表示包括当前块的画面。可以以比特流用信号发送表示当前画面参考是否被应用于当前块的信息。在示例中,该信息可以是1比特标志。当该标志为真时,可将针对当前块的预测编码模式确定为当前画面参考,并且当该标志为假时,可将针对当前块的预测编码模式确定为帧间预测。

[0123] 可选地,可基于参考画面索引来确定当前块的预测编码模式。在示例中,当参考画面索引指示当前画面时,可将针对当前块的预测编码模式确定为当前画面参考。当参考画面索引指示除了当前画面之外的画面时,可将针对当前块的预测编码模式确定为帧间预测。换言之,当前画面参考是使用关于当前画面内的已被编码/解码的区域的信息的预测方法,并且帧间预测是使用关于已被编码/解码的另一画面的信息的预测方法。

[0124] 组合预测表示将帧内预测、帧间预测和当前画面参考中的至少两个进行组合的组编码模式。在示例中,当组合预测被应用时,可基于帧内预测、帧间预测或当前画面参考中的任意一个来生成第一预测块,并且可基于帧内预测、帧间预测或当前画面参考中的另一个来生成第二预测块。当生成了第一预测块和第二预测块时,可通过计算第一预测块和第二预测块的平均值或加权和来生成最终预测块。可以以比特流用信号发送表示是否将组合预测应用于当前块的信息。该信息可以是1比特标志。

[0125] 图4是示出编码块的各种分区类型的示图。

[0126] 可基于四叉树分区、二叉树分区或三叉树分区将编码块分区为多个编码块。可基于四叉树分区、二叉树分区或三叉树分区将分区出的编码块再次分区为多个编码块。

[0127] 四叉树分区表示将当前块分区为四个块的方法。作为四叉树分区的结果,当前块可被分区为四个正方形分区(参照图4的“SPLIT_QT”)。

[0128] 二叉树分区表示将当前块分区为两个块的方法。沿着垂直方向(即,使用穿过当前块的垂直线)将当前块分区为两个块的操作可被称为垂直方向二叉树分区,并且沿着水平方向(即,使用穿过当前块的水平线)将当前块分区为两个块的操作可被称为水平方向二叉树分区。作为二叉树分区的结果,当前块可被分区为两个非正方形分区。图4的“SPLIT_BT_VER”是示出垂直方向二叉树分区的结果的示图,并且图4的“SPLIT_BT_HOR”是示出水平方向二叉树分区的结果的示图。

[0129] 三叉树分区表示将当前块分区为三个块的方法。沿着垂直方向(即,使用穿过当前块的两条垂直线)将当前块分区为三个块的操作可被称为垂直方向三叉树分区,并且沿着水平方向(即,使用穿过当前块的两条水平线)将当前块分区为三个块的操作可被称为水平方向三叉树分区。作为三叉树分区的结果,当前块可被分区为三个非正方形分区。就此而言,位于当前块的中心的分区的宽度/高度可以是其他分区的宽度/高度的两倍。图4的“SPLIT_TT_VER”是示出垂直方向三叉树分区的结果的示图,而图4的“SPLIT_TT_HOR”是示出水平方向三叉树分区的结果的示图。

[0130] CTU的分区次数可被定义为分区深度。CTU的最大分区深度可在序列级或画面级被确定。因此,CTU的最大分区深度可基于序列或画面而变化。

[0131] 可选地,可针对每种分区方法独立地确定最大分区深度。在示例中,允许四叉树分区的最大分区深度可与允许二叉树分区和/或三叉树分区的最大分区深度不同。

[0132] 编码器可以以比特流用信号发送表示当前块的分区类型和分区深度中的至少一

个的信息。解码器可基于通过对比特流进行解析而获得的该信息来确定CTU的分区类型和分区深度。

[0133] 图5是示出对CTU进行分区的方面的示例的示意图。

[0134] 通过使用四叉树分区、二叉树分区和/或三叉树分区来对编码块进行分区可被称为多树分区。

[0135] 通过应用多树分区对编码块进行分区而生成的编码块可被称为子编码块。当编码块的分区深度为 k 时,子编码块的分区深度被设置为 $k+1$ 。

[0136] 相反,对于分区深度为 $k+1$ 的编码块,分区深度为 k 的编码块可被称为父编码块。

[0137] 可基于父编码块的分区类型和邻近编码块的分区类型中的至少一个来确定当前编码块的分区类型。就此而言,邻近编码块可以是与当前编码块相邻的块,并且包括上方邻近块、左侧邻近块或与当前编码块的左上角相邻的邻近块中的至少一个。就此而言,所述分区类型可包括是否应用四叉树分区、是否应用二叉树分区、二叉树分区的方向、是否应用三叉树分区或三叉树分区的方向。

[0138] 为了确定编码块的分区类型,可以以比特流用信号发送表示编码块是否被分区的信息。该信息是1比特标志“split_cu_flag”,并且当该标志为真时,它可表示编码块通过多树分区方法被分区。

[0139] 当split_cu_flag为真时,可以以比特流用信号发送表示编码块是否通过四叉树分区被分区的信息。该信息是1比特标志split_qt_flag,并且当该标志为真时,编码块可被分区为四个块。

[0140] 在示例中,在图5中所示的示例中,CTU通过四叉树分区被分区,并且因此生成分区深度为1的四个编码块。此外,示出了再次将四叉树分区应用于通过四叉树分区生成的四个编码块中的第一编码块和第四编码块。结果,可生成分区深度为2的四个编码块。

[0141] 此外,通过再次将四叉树分区应用于分区深度为2的编码块,可生成分区深度为3的编码块。

[0142] 当四叉树分区不被应用于编码块时,可根据编码块的尺寸、编码块是否位于画面边界、最大分区深度或邻近块的分区类型中的至少一个来确定是否针对编码块执行二叉树分区或三叉树分区。当确定针对编码块执行二叉树分区或三叉树分区时,可以以比特流用信号发送表示分区方向的信息。该信息可以是1比特标志mtt_split_cu_vertical_flag。可基于该标志确定分区方向是垂直方向还是水平方向。此外,可以以比特流用信号发送表示二叉树分区或三叉树分区中的哪一个被应用于编码块的信息。该信息可以是1比特标志mtt_split_cu_binary_flag。可基于该标志确定是二叉树分区被应用于编码块还是三叉树分区被应用于编码块。

[0143] 在示例中,在图5中所示的示例中,垂直方向二叉树分区被应用于分区深度为1的编码块,垂直方向三叉树分区被应用于通过分区生成的编码块中的左侧编码块,并且垂直方向二叉树分区被应用于右侧编码块。

[0144] 帧间预测是一种通过使用关于先前画面的信息对当前块进行预测的预测编码模式。在示例中,先前画面内的与当前块相同位置处的块(在下文中,同位块)可被设置为当前块的预测块。在下文中,基于当前块的同位块生成的预测块可被称为同位预测块。

[0145] 相反,当存在于先前画面中的对象已移动到当前画面中的另一位置时,可通过使

用该对象的运动来有效地预测当前块。例如,当通过将先前画面与当前画面进行比较来确定对象的运动方向和尺寸时,可根据对象的运动信息生成当前块的预测块(或预测图像)。在下文中,通过使用运动信息生成的预测块可被称为运动预测块。

[0146] 可通过从当前块减去预测块来生成残差块。就此而言,在对象移动的情况下,可通过使用运动预测块而不是使用同位预测块来减小残差块的能量,因此可提高残差块的压缩性能。

[0147] 如上,通过使用运动信息生成预测块的操作可被称为运动估计预测。在大多数帧间预测中,可基于运动补偿预测来生成预测块。

[0148] 运动信息可包括运动矢量、参考画面索引、预测方向和双向加权因子索引中的至少一个。运动矢量表示对象的运动方向和大小。参考画面索引指定参考画面列表中包括的参考画面中的当前块的参考画面。预测方向指示单向L0预测、单向L1预测或双向预测(L0预测和L1预测)中的任意一个。可根据当前块的预测方向使用L0方向运动信息和L1方向运动信息中的至少一个。双向加权因子索引指定应用于L0预测块的加权因子和应用于L1预测块的加权因子。

[0149] 图6是根据本公开的实施例的帧间预测方法的流程图。

[0150] 参照图6,帧间预测方法包括:确定针对当前块的帧间预测模式(S601),根据所确定的帧间预测模式获得当前块的运动信息(S602),并且基于所获得的运动信息执行针对当前块的运动补偿预测(S603)。

[0151] 就此而言,帧间预测模式可表示用于确定当前块的运动信息的各种方法,并且包括使用平移运动信息的帧间预测模式、使用仿射运动信息的帧间预测模式。在示例中,使用平移运动信息的帧间预测模式可包括合并模式和运动矢量预测模式,并且使用仿射运动信息的帧间预测模式可包括仿射合并模式和仿射运动矢量预测模式。可基于与当前块邻近的邻近块或通过对比特流进行解析而获得的信息来确定关于当前块的运动信息。

[0152] 可从另一块的运动信息推导当前块的运动信息。就此而言,另一块可以是在当前块之前通过帧间预测被编码/解码的块。将当前块的运动信息设置为与另一块的运动信息相同的操作可被定义为合并模式。此外,将另一块的运动矢量设置为当前块的运动矢量的预测值的操作可被定义为运动矢量预测模式。

[0153] 图7是在合并模式下推导当前块的运动信息的处理的流程图。

[0154] 可推导当前块的合并候选S701。可在当前块之前通过帧间预测被编码/解码的块推导当前块的合并候选。

[0155] 图8是示出用于推导合并候选的候选块的示图。

[0156] 候选块可包括邻近块或非邻近块中的至少一个,其中,所述邻近块包括与当前块相邻的样点,所述非邻近块包括与当前块不相邻的样点。在下文中,确定候选块的样点被定义为基本样点。此外,与当前块相邻的基本样点被称作邻近基本样点,并且与当前块不相邻的基本样点被称作非邻近基本样点。

[0157] 邻近基本样点可被包括在当前块的最左侧列的邻近列或当前块的最上方行的邻近行中。在示例中,当当前块的左上样点的坐标为(0,0)时,包括位置(-1,H-1)、(W-1,-1)、(W,-1)、(-1,H)或(-1,1)处的基本样点的块中的至少一个可被用作候选块。参照示图,索引0至4的邻近块可被用作候选块。

[0158] 非邻近基本样点表示距与当前块相邻的基本样点的x轴距离或y轴距离中的至少一个具有预定义值的样点。在示例中,包括距左基本样点的x轴距离是预定义值的基本样点的块、包括距上基本样点的y轴距离是预定义值的非邻近样点的块或者包括距左上基本样点的x轴距离和y轴距离是预定义值的非邻近样点的块中的至少一个块可被用作候选块。预定义值可以是诸如4、8、12、16等的自然数。参照附图,索引5至26的块中的至少一个块可被用作候选块。

[0159] 可将不位于与邻近基本样点相同的垂直线、水平线或对角线上的样点设置为非邻近基本样点。

[0160] 在下文中,候选块中的包括邻近基本样点的候选块被称为邻近块,并且包括非邻近基本样点的块被称为非邻近块。

[0161] 当当前块与候选块之间的距离等于或大于阈值时,候选块可被设置为不可用作合并候选。可基于编码树单元的尺寸来确定所述阈值。在示例中,阈值可被设置为编码树单元的高度(ctu_height)或者将偏移加上编码树单元的高度或从编码树单元的高度减去偏移得到的值(例如,ctu_height $\pm$ N)。作为编码器和解码器中的预定义值,偏移N可被设置为4、8、16、32或ctu_height。

[0162] 当当前块的y轴坐标与包括在候选块中的样点的y轴坐标之间的差大于阈值时,候选块可被确定为不可用作合并候选。

[0163] 可选地,与当前块不属于相同的编码树单元的候选块可被设置为不可用作合并候选。在示例中,当基本样点在当前块所属的编码树单元的上边界之外时,包括该基本样点的候选块可被设置为不可用作合并候选。

[0164] 如果当前块的上边界与编码树单元的上边界相接,则多个候选块可被确定为不可用作合并候选,从而降低当前块的编码/解码效率。为了解决这样的问题,候选块可被设置为使得当前块的左侧位置处的候选块的数量大于当前块的上方位置处的候选块的数量。

[0165] 图9是示出用于推导合并候选的候选块的示图。

[0166] 如在图9中所示的示例中,属于当前块的N个上方块列的上方块和属于当前块的M个左侧块列的左侧块可被设置为候选块。在这种情况下,可通过将M设置为大于N来将左侧候选块的数量设置为大于上方候选块的数量。

[0167] 在示例中,当前块中的基本样点的y轴坐标与可被用作候选块的上方块的y轴坐标之间的差可被设置为不超过当前块的高度的N倍。此外,当前块中的基本样点的y轴坐标与可被用作候选块的左侧块的y轴坐标之间的差可被设置为不超过当前块的宽度的M倍。

[0168] 在示例中,在图9中所示的示例中,示出了属于当前块的2个上方块列的块和属于当前块的5个左侧块列的块被设置为候选块。

[0169] 可从不同于当前块的画面中包括的时间邻近块推导合并候选。在示例中,可从同位画面中包括的同位块推导合并候选。参考画面列表中包括的参考画面中的任意一个参考画面可被设置为同位画面。可以以比特流用信号发送标识参考画面中的同位画面的索引信息。可选地,参考画面中的具有预定义索引的参考画面可被确定为同位画面。

[0170] 合并候选的运动信息可被设置为与候选块的运动信息相同。在示例中,候选块的运动矢量、参考画面索引、预测方向或双向权重索引中的至少一个可被设置为合并候选的运动信息。

[0171] 可生成包括合并候选的合并候选列表S702。

[0172] 可根据预定顺序分配合并候选列表中的合并候选的索引。在示例中,可按照从左侧邻近块推导出的合并候选、从上方邻近块推导出的合并候选、从右上方邻近块推导出的合并候选、从左下方邻近块推导出的合并候选、从左上方邻近块推导出的合并候选和从时间邻近块推导出的合并候选的顺序分配索引。

[0173] 当多个合并候选被包括在合并候选列表中时,可选择多个合并候选中的至少一个合并候选S703。具体地,可以以比特流用信号发送用于指定多个合并候选中的任意一个合并候选的信息。在示例中,可以以比特流用信号发送表示合并候选列表中包括的合并候选中的任意一个合并候选的索引的信息merge_idx。

[0174] 当包括在合并候选列表中的合并候选的数量小于阈值时,包括在运动信息表中的运动信息候选可作为合并候选被添加到合并候选列表。就此而言,所述阈值可以是可被包括在合并候选列表中的合并候选的最大数量或者从所述合并候选的最大数量减去偏移的值。偏移可以是诸如1或2等的自然数。

[0175] 运动信息表包括从当前画面中的基于帧间预测被编码/解码的块推导出的运动信息候选。在示例中,包括在运动信息表中的运动信息候选的运动信息可被设置为与基于帧间预测被编码/解码的块的运动信息相同。就此而言,运动信息可包括运动矢量、参考画面索引、预测方向或双向权重索引中的至少一个。

[0176] 包括在运动信息表中的运动信息候选也可被称作帧间区域合并候选或预测区域合并候选。

[0177] 可在编码器和解码器中预定义可被包括在运动信息表中的运动信息候选的最大数量。在示例中,可被包括在运动信息表中的运动信息候选的最大数量可以是1、2、3、4、5、6、7、8或更大(例如,16)。

[0178] 可选地,可以以比特流用信号发送表示可被包括在运动信息表中的运动信息候选的最大数量的信息。可以以序列级、画面级或条带级用信号发送所述信息。所述信息可表示可被包括在运动信息表中的运动信息候选的最大数量。可选地,所述信息可表示可被包括在运动信息表中的运动信息候选的最大数量与可被包括在合并候选列表中的合并候选的最大数量之间的差。

[0179] 可选地,可根据画面尺寸、条带尺寸或编码树单元尺寸来确定可被包括在运动信息表中的运动信息候选的最大数量。

[0180] 运动信息表可以以画面、条带、并行块(tile)、分块(brick)、编码树单元或编码树单元线(行或列)为单位被初始化。在示例中,当条带被初始化时,运动信息表也被初始化,因此运动信息表可不包括任何运动信息候选。

[0181] 可选地,可以以比特流用信号发送表示运动信息表是否将被初始化的信息。可在条带级、并行块级、分块级或块级用信号发送所述信息。可使用预配置的运动信息表,直到所述信息指示运动信息表的初始化为止。

[0182] 可选地,关于初始运动信息候选的信息可在画面参数集或条带头中被用信号发送。尽管条带被初始化,但是运动信息表可包括初始运动信息候选。因此,可针对作为条带中的第一编码/解码目标的块使用初始运动信息候选。

[0183] 可选地,可将包括在先前编码树单元的运动信息表中的运动信息候选设置为初始

运动信息候选。在示例中,可将先前编码树单元的运动信息表中包括的运动信息候选中的具有最小索引或具有最大索引的运动信息候选设置为初始运动信息候选。

[0184] 按照编码/解码顺序对块进行编码/解码,并且可按照编码/解码顺序将基于帧间预测被编码/解码的块顺序地设置为运动信息候选。

[0185] 图10是用于解释运动信息表的更新方面的示意图。

[0186] 对于当前块,当执行帧间预测S1001时,可基于当前块推导运动信息候选S1002。运动信息候选的运动信息可被设置为与当前块的运动信息相同。

[0187] 当运动信息表为空时S1003,可将基于当前块推导出的运动信息候选添加到运动信息表S1004。

[0188] 当运动信息表已包括运动信息候选S1003时,可执行针对当前块的运动信息(或基于当前块推导出的运动信息候选)的冗余校验S1005。冗余校验在于确定运动信息表中的预存储的运动信息候选的运动信息是否与当前块的运动信息相同。可针对运动信息表中的所有预存储的运动信息候选执行冗余校验。可选地,可针对运动信息表中的预存储的运动信息候选中的索引超过或低于阈值的运动信息候选执行冗余校验。可选地,可针对预定义数量的运动信息候选执行冗余校验。在示例中,可将具有最小索引或具有最大索引的2个运动信息候选确定为用于冗余校验的目标。

[0189] 当不包括与当前块具有相同运动信息的运动信息候选时,可将基于当前块推导出的运动信息候选添加到运动信息表S1008。可基于运动信息候选的运动信息(例如,运动矢量/参考画面索引等)是否相同来确定运动信息候选是否相同。

[0190] 就此而言,当最大数量的运动信息候选已被存储在运动信息表中时S1006,可删除最老的运动信息候选S1007,并且可将基于当前块推导出的运动信息候选添加到运动信息表S1008。就此而言,最老的运动信息候选可以是具有最大索引或最小索引的运动信息候选。

[0191] 运动信息候选可由相应索引标识。当从当前块推导出的运动信息候选被添加到运动信息表时,可将最小索引(例如,0)分配给该运动信息候选,并且预存储的运动信息候选的索引可增加1。就此而言,当最大数量的运动信息候选已被存储在运动信息表中时,去除具有最大索引的运动信息候选。

[0192] 可选地,当将从当前块推导出的运动信息候选添加到运动信息表时,可将最大索引分配给该运动信息候选。在示例中,当运动信息表中的预存储的运动信息候选的数量小于最大值时,可将与预存储的运动信息候选的数量具有相同的值的索引分配给该运动信息候选。可选地,当运动信息表中的预存储的运动信息候选的数量等于最大值时,可将从最大值减去1得到的索引分配给该运动信息候选。可选地,去除具有最小索引的运动信息候选,并且将其余的预存储的运动信息候选的索引减小1。

[0193] 图11是示出运动信息表的更新方面的示意图。

[0194] 假设当将从当前块推导出的运动信息候选添加到运动信息表时,将最大索引分配给该运动信息候选。此外,假设最大数量的运动信息候选已被存储在运动信息表中。

[0195] 当将从当前块推导出的运动信息候选HmvpCand[n+1]添加到运动信息表HmvpCandList时,可删除预存储的运动信息候选中的具有最小索引的运动信息候选HmvpCand[0],并且其余的运动信息候选的索引可减小1。此外,可将当前块推导出的运动

信息候选HmvpCand[n+1]的索引设置为最大值(对于图11中所示的示例,n)。

[0196] 当预存储了与基于当前块推导出的运动信息候选相同的运动信息候选时S1005,可不将基于当前块推导出的运动信息候选添加到运动信息表S1009。

[0197] 可选地,在将基于当前块推导出的运动信息候选添加到运动信息表时,可去除与该运动信息候选相同的预存储的运动信息候选。在这种情况下,产生与预存储的运动信息候选的索引被重新更新时相同的效果。

[0198] 图12是示出预存储的运动信息候选的索引被更新的示例的示意图。

[0199] 当与从当前块推导出的运动信息候选mvCand相同的预存储的运动信息候选的索引为hIdx时,可去除该预存储的运动信息候选,并且索引大于hIdx的运动信息候选的索引可减小1。在示例中,图12中所示的示例示出了在运动信息表HmvpCandList中删除与mvCand相同的HmvpCand[2],并且从HmvpCand[3]到HmvpCand[n]的索引减小1。

[0200] 而且,可将基于当前块推导出的运动信息候选mvCand添加到运动信息表的末尾。

[0201] 可选地,分配给与基于当前块推导出的运动信息候选相同的预存储的运动信息候选的索引可被更新。例如,可将预存储的运动信息候选的索引改变为最小值或最大值。

[0202] 可将包括在预定区域中的块的运动信息设置为不被添加到运动信息表。在示例中,可不将基于包括在合并处理区域中的块的运动信息推导出的运动信息候选添加到运动信息表。由于针对包括在合并处理区域中的多个块的编码/解码顺序未被定义,因此将所述多个块中的任意一个块的运动信息用于所述多个块中的另一块的帧间预测是不合适的。因此,可不将基于包括在合并处理区域中的块推导出的运动信息候选添加到运动信息表。

[0203] 可选地,可将小于预设尺寸的块的运动信息设置为不被添加到运动信息表。在示例中,可不将基于宽度或高度小于4或8的编码块的运动信息或者4×4尺寸的编码块的运动信息推导出的运动信息候选添加到运动信息表。

[0204] 当以子块为单位执行运动补偿预测时,可基于当前块中包括的多个子块中的代表性子块的运动信息推导运动信息候选。在示例中,当子块合并候选被用于当前块时,可基于子块中的代表性子块的运动信息推导运动信息候选。

[0205] 可按照以下顺序推导子块的运动矢量。首先,可选择当前块的合并候选列表中包括的合并候选中的一个合并候选,并且可基于所选的合并候选的运动矢量推导初始移位矢量(shVector)。并且,可通过将初始移位矢量与编码块中的每个子块的基本样点(例如,左上样点或中心样点)的位置(xSb,ySb)相加来推导基本样点的位置为(xColSb,yColSb)的移位子块。下面的等式1示出用于推导移位子块的等式。

[0206] 【等式1】

[0207] $(xColSb, yColSb) = (xSb + shVector[0] >> 4, ySb + shVector[1] >> 4)$

[0208] 然后,可将与包括(xColSb,yColSb)的子块的中心位置对应的同位块的运动矢量设置为包括(xSb,ySb)的子块的运动矢量。

[0209] 代表性子块可表示包括当前块的左上样点、中心样点、右下样点、右上样点或左下样点的子块。

[0210] 图13是示出代表性子块的位置的示意图。

[0211] 图13(a)表示当前块的左上方位置处的子块被设置为代表性子块的示例,并且图13(b)表示当前块的中心位置处的子块被设置为代表性子块的示例。当以子块为单位执行

运动补偿预测时,可基于包括当前块的左上样点的子块或包括当前块的中心样点的子块的运动矢量推导当前块的运动信息候选。

[0212] 基于当前块的帧间预测模式,可确定当前块是否将被用作运动信息候选。在示例中,基于仿射运动模型被编码/解码的块可被设置为不可用作运动信息候选。因此,尽管通过帧间预测对当前块进行编码/解码,但是当当前块的帧间预测模式是仿射预测模式时,可不基于当前块更新运动信息表。

[0213] 可选地,可基于以下项中的至少一项来确定当前块是否将被用作运动信息候选:当前块的运动矢量分辨率、合并偏移编码方法是否被应用、组合预测是否被应用、或者三角形分区是否被应用。在示例中,对于当前块的运动信息分辨率等于或大于2整数像素时的情况、组合预测被应用于当前块时的情况、三角形分区被应用于当前块时的情况或者合并偏移编码方法被应用于当前块时的情况中的至少一个情况,当前块可被设置为不可用作运动信息候选。

[0214] 可选地,可根据基于仿射运动模型被编码/解码的块中包括的子块中的至少一个子块的子块矢量推导运动信息候选。在示例中,可通过使用当前块的左上方位置处的子块、中心位置处的子块或右上方位置处的子块来推导运动信息候选。可选地,多个子块的子块矢量的平均值可被设置为运动信息候选的运动矢量。

[0215] 当包括在当前块的合并候选列表中的合并候选的数量小于阈值时,包括在运动信息表中的运动信息候选可作为合并候选被添加到合并候选列表。按照按升序或降序反映运动信息候选的索引的排序的顺序执行附加处理。在示例中,首先可将具有最大索引的运动信息候选添加到当前块的合并候选列表。

[0216] 当包括在运动信息表中的运动信息候选被添加到合并候选列表时,可执行运动信息候选与合并候选列表中的预存储的合并候选之间的冗余校验。作为冗余校验的结果,可不将与预存储的合并候选具有相同的运动信息的运动信息候选添加到合并候选列表。

[0217] 可仅针对包括在运动信息表中的运动信息候选中的一部分运动信息候选执行冗余校验。在示例中,可仅针对索引超过或低于阈值的运动信息候选执行冗余校验。可选地,可仅针对具有最大索引或最小索引的N个运动信息候选执行冗余校验。可选地,可仅针对合并候选列表中的预存储的合并候选中的一部分合并候选执行冗余校验。在示例中,可仅针对索引超过或低于阈值的合并候选或从特定位置处的块推导出的合并候选执行冗余校验。就此而言,特定位置可包括当前块的左侧邻近块、上方邻近块、右上方邻近块或左下方邻近块中的至少一个。

[0218] 图14是示出仅针对一部分合并候选执行冗余校验的示例的示意图。

[0219] 当运动信息候选HmvpCand[j]被添加到合并候选列表时,可针对运动信息候选执行对具有最大索引的2个合并候选mergeCandList[NumMerge-2]和mergeCandList[NumMerge-1]的冗余校验。就此而言,NumMerge可示出可用空间合并候选和时间合并候选的数量。

[0220] 不同于所示的示例,当运动信息候选HmvpCand[j]被添加到合并候选列表时,可针对运动信息候选执行对具有最小索引的2个合并候选的冗余校验。例如,可检查mergeCandList[0]和mergeCandList[1]是否与HmvpCand[j]相同。

[0221] 可选地,可仅针对从特定位置推导出的合并候选执行冗余校验。在示例中,可针对

从位于当前块的左侧或当前块的上方的邻近块推导出的合并候选中的至少一个执行冗余校验。当在合并候选列表中不存在从特定位置推导出的合并候选时,可在不进行冗余校验的情况下将运动信息候选添加到合并候选列表。

[0222] 当运动信息候选HmvpCand[j]被添加到合并候选列表时,可针对运动信息候选执行对具有最大索引的2个合并候选mergeCandList[NumMerge-2]和mergeCandList[NumMerge-1]的冗余校验。就此而言,NumMerge可示出可用空间合并候选和时间合并候选的数量。

[0223] 可仅针对运动信息候选中的一部分运动信息候选执行对合并候选的冗余校验。在示例中,可仅针对运动信息表中包括的运动信息候选中的具有大索引或小索引的N个运动信息候选执行冗余校验。在示例中,可仅针对具有包括在运动信息表中的运动信息候选的编号和差低于阈值的索引的运动信息候选执行冗余校验。当所述阈值为2时,可仅针对运动信息表中包括的运动信息候选中的具有最大索引值的3个运动信息候选执行冗余校验。对于除了上述3个运动信息候选之外的运动信息候选,可省略冗余校验。当省略冗余校验时,可将运动信息候选添加到合并候选列表,而不管是否存在与合并候选相同的运动信息。

[0224] 相反,将冗余校验设置为仅针对具有包括在运动信息表中的运动信息候选的编号和差超过所述阈值的索引的运动信息候选来执行。

[0225] 可在编码器和解码器中重新定义执行冗余校验的运动信息候选的数量。在示例中,所述阈值可以是诸如0、1或2的整数。

[0226] 可选地,可基于包括在合并候选列表中的合并候选的数量或包括在运动信息表中的运动信息候选的数量中的至少一个来确定所述阈值。

[0227] 当发现与第一运动信息候选相同的合并候选时,可在针对第二运动信息候选的冗余校验中省略对与第一运动信息候选相同的合并候选的冗余校验。

[0228] 图15是示出省略对特定合并候选的冗余校验的示例的示意图。

[0229] 当索引为i的运动信息候选HmvpCand[i]被添加到合并候选列表时,执行该运动信息候选与合并候选列表中的预存储的合并候选之间的冗余校验。就此而言,当发现与运动信息候选HmvpCand[i]相同的合并候选mergeCandList[j]时,可在不将运动信息候选HmvpCand[i]添加到合并候选列表的情况下执行索引为i-1的运动信息候选HmvpCand[i-1]与合并候选之间的冗余校验。就此而言,可省略运动信息候选HmvpCand[i-1]与合并候选mergeCandList[j]之间的冗余校验。

[0230] 在示例中,在图15中所示的示例中,确定HmvpCand[i]和mergeCandList[2]相同。因此,可在不将HmvpCand[i]添加到合并候选列表的情况下执行针对HmvpCand[i-1]的冗余校验。就此而言,可省略HmvpCand[i-1]与mergeCandList[2]之间的冗余校验。

[0231] 当包括在当前块的合并候选列表中的合并候选的数量小于阈值时,除了运动信息候选之外,可另外包括成对合并候选或零合并候选中的至少一个。成对合并候选表示具有从对多于2个合并候选的运动矢量求平均而获得的值作为运动矢量的合并候选,并且零合并候选表示运动矢量为0的合并候选。

[0232] 对于当前块的合并候选列表,可按照以下顺序添加合并候选。

[0233] 空间合并候选-时间合并候选-运动信息候选-(仿射运动信息候选)-成对合并候选-零合并候选

[0234] 空间合并候选表示从邻近块或非邻近块中的至少一个推导出的合并候选,并且时间合并候选表示从先前参考画面推导出的合并候选。仿射运动信息候选表示从通过仿射运动模型被编码/解码的块推导出的运动信息候选。

[0235] 还可在运动矢量预测模式下使用运动信息表。在示例中,当包括在当前块的运动矢量预测候选列表中的运动矢量预测候选的数量小于阈值时,包括在运动信息表中的运动信息候选可被设置为针对当前块的运动矢量预测候选。具体地,运动信息候选的运动矢量可被设置为运动矢量预测候选。

[0236] 如果包括在当前块的运动矢量预测候选列表中的运动矢量预测候选中的任意一个被选择,则可将所选的候选设置为当前块的运动矢量预测因子。然后,在对当前块的运动矢量残差值进行解码之后,可通过将运动矢量预测因子和运动矢量残差值相加来获得当前块的运动矢量。

[0237] 可按照以下顺序配置当前块的运动矢量预测候选列表。

[0238] 空间运动矢量预测候选-时间运动矢量预测候选-运动信息候选-(仿射运动信息候选)-零运动矢量预测候选

[0239] 空间运动矢量预测候选表示从邻近块或非邻近块中的至少一个推导出的运动矢量预测候选,并且时间运动矢量预测候选表示从先前参考画面推导出的运动矢量预测候选。仿射运动信息候选表示从通过仿射运动模型被编码/解码的块推导出的运动信息候选。零运动矢量预测候选表示运动矢量的值为0的候选。

[0240] 大于编码块的合并处理区域可被定义。合并处理区域中包括的编码块可被并行地处理,而无需被顺序地编码/解码。就此而言,不被顺序地编码/解码表示编码/解码的顺序未被定义。因此,可独立地对合并处理区域中包括的块的编码/解码处理进行处理。可选地,包括在合并处理区域中的块可共享合并候选。就此而言,可基于合并处理区域推导合并候选。

[0241] 根据上述特征,合并处理区域可被称为并行处理区域、共享合并区域(SMR)或合并估计区域(MER)。

[0242] 可基于编码块推导当前块的合并候选。但是,当当前块被包括在大于当前块的合并处理区域中时,包括在与当前块相同的合并处理区域中的候选块可被设置为不可用作合并候选。

[0243] 图16是示出包括在与当前块相同的合并处理区域中的候选块被设置为不可用作合并候选的示例的示图。

[0244] 在图16(a)中所示的示例中,在CU5的编码/解码中,包括与CU5相邻的基本样点的块可被设置为候选块。就此而言,包括在与CU5相同的合并处理区域中的候选块x3和x4可被设置为不可用作CU5的合并候选。但是,未包括在与CU5相同的合并处理区域中的候选块x0、x1和x2可被设置为可用作合并候选。

[0245] 在图16(b)中所示的示例中,在CU8的编码/解码中,包括与CU8相邻的基本样点的块可被设置为候选块。就此而言,包括在与CU8相同的合并处理区域中的候选块x6、x7和x8可被设置为不可用作合并候选。但是,未包括在与CU8相同的合并处理区域中的候选块x5和x9可被设置为可用作合并候选。

[0246] 可选地,当当前块被包括在合并处理区域中时,可将与当前块和合并处理区域相

邻的邻近块设置为候选块。

[0247] 图17是示出当当前块被包括在合并处理区域中时推导针对当前块的合并候选的示例的示意图。

[0248] 如在图17 (a) 所示的示例中,与当前块相邻的邻近块可被设置为用于推导当前块的合并候选的候选块。就此而言,包括在与当前块相同的合并处理区域中的候选块可被设置为不可用作合并候选。在示例中,在推导针对编码块CU3的合并候选时,包括在与编码块CU3相同的合并处理区域中的上方邻近块y3和右上方邻近块y4可被设置为不可用作编码块CU3的合并候选。

[0249] 通过按照预定义顺序对与当前块相邻的邻近块进行扫描,可推导合并候选。在示例中,所述预定义顺序可以是y1、y3、y4、y0和y2的顺序。

[0250] 当可从与当前块相邻的邻近块推导的合并候选的数量小于从合并候选的最大数量减去偏移的值或所述最大数量时,可如图17 (b) 中所示的示例通过使用与合并处理区域相邻的邻近块来推导针对当前块的合并候选。在示例中,与包括编码块CU3的合并处理区域相邻的邻近块可被设置为针对编码块CU3的候选块。就此而言,与合并处理区域相邻的邻近块可包括左侧邻近块x1、上方邻近块x3、左下方邻近块x0、右上方邻近块x4或左上方邻近块x2中的至少一个。

[0251] 通过按照预定义顺序对与合并处理区域相邻的邻近块进行扫描,可推导合并候选。在示例中,所述预定义顺序可以是x1、x3、x4、x0和x2的顺序。

[0252] 总之,可通过按照以下扫描顺序对候选块进行扫描来推导针对包括在合并处理区域中的编码块CU3的合并候选。

[0253] (y1, y3, y4, y0, y2, x1, x3, x4, x0, x2)

[0254] 但是,上面示出的候选块的扫描顺序仅示出了本公开的示例,并且可按照与以上示例不同的顺序对候选块进行扫描。可选地,可基于当前块或合并处理区域的尺寸或形状中的至少一个来自适应地确定扫描顺序。

[0255] 合并处理区域可以是正方形或非正方形的。可以以比特流用信号发送用于确定合并处理区域的信息。该信息可包括表示合并处理区域的形状的信息或表示合并处理区域的尺寸的信息中的至少一个信息。当合并处理区域是非正方形时,可以以比特流用信号发送表示合并处理区域的尺寸的信息、表示合并处理区域的宽度或高度的信息或者表示合并处理区域的宽高比的信息中的至少一个信息。

[0256] 可基于以比特流用信号发送的信息、画面分辨率、条带的尺寸或并行块的尺寸中的至少一个来确定合并处理区域的尺寸。

[0257] 如果针对包括在合并处理区域中的块执行运动补偿预测,则可将基于被执行运动补偿预测的块的运动信息推导出的运动信息候选添加到运动信息表。

[0258] 但是,如果从包括在合并处理区域中的块推导出的运动信息候选被添加到运动信息表,则可能发生在对合并处理区域中的编码/解码实际上比该块慢的其它块的编码/解码中使用从该块推导出的运动信息候选的情况。换言之,虽然应该在合并处理区域中包括的块的编码/解码中排除块之间的依赖性,但是可能发生通过使用合并处理区域中包括的其他块的运动信息来执行运动预测补偿的情况。为了解决这样的问题,虽然完成了对合并处理区域中包括的块的编码/解码,但是可不将完成了编码/解码的块的运动信息添加到运动

信息表。

[0259] 可选地, 可仅使用合并处理区域内的预定义位置处的块来更新运动信息表。所述预定义位置的示例可包括位于合并处理区域的左上方的块、位于合并处理区域的右上方的块、位于合并处理区域的左下方的块、位于合并处理区域的右下方的块、位于合并处理区域的中心的块、与合并处理区域的右边界相邻的块和与合并处理区域的下边界相邻的块中的至少一个。作为示例, 可仅利用与合并处理区的右下角相邻的块的运动信息来更新运动信息表, 并且可不利用其它块的运动信息来更新运动信息表。

[0260] 可选地, 在完成了对包括在合并处理区域中的所有块的解码之后, 可将从所述块推导出的运动信息候选添加到运动信息表。也就是说, 虽然包括在合并处理区域中的块被编码/解码, 但可不更新运动信息表。

[0261] 在示例中, 如果针对合并处理区域中包括的块执行运动补偿预测, 则可按照预定义顺序将从所述块推导出的运动信息候选添加到运动信息表。就此而言, 可按照合并处理区域或编码树单元中的编码块的扫描顺序来确定所述预定义顺序。扫描顺序可以是光栅扫描、水平扫描、垂直扫描或Z字形扫描中的至少一个。可选地, 可基于每个块的运动信息或具有相同运动信息的块的数量来确定所述预定义顺序。

[0262] 可选地, 包括单向运动信息的运动信息候选可在包括双向运动信息的运动信息候选之前被添加到运动信息表。相反, 包括双向运动信息的运动信息候选可在包括单向运动信息的运动信息候选之前被添加到运动信息表。

[0263] 可选地, 可按照在合并处理区域或编码树单元中高使用频率或低使用频率的顺序将运动信息候选添加到运动信息表。

[0264] 当当前块被包括在合并处理区域中且包括在当前块的合并候选列表中的合并候选的数量小于最大数量时, 可将包括在运动信息表中的运动信息候选添加到合并候选列表。就此而言, 可将从包括在与当前块相同的合并处理区域中的块推导出的运动信息候选设置为不被添加到当前块的合并候选列表。

[0265] 可选地, 当当前块被包括在合并处理区域中时, 可设置为不使用包括在运动信息表中的运动信息候选。换言之, 虽然包括在当前块的合并候选列表中的合并候选的数量小于最大数量, 但是可不将包括在运动信息表中的运动信息候选添加到合并候选列表。

[0266] 在另一示例中, 可配置关于合并处理区域或编码树单元的运动信息表。这个运动信息表起到临时存储合并处理区域中包括的块的运动信息的作用。为了将一般运动信息表与针对合并处理区域或编码树单元的运动信息表区分开, 针对合并处理区域或编码树单元的运动信息表被称为临时运动信息表。而且, 存储在临时运动信息表中的运动信息候选被称为临时运动信息候选。

[0267] 图18是示出临时运动信息表的示意图。

[0268] 可配置针对编码树单元或合并处理区域的临时运动信息表。当对包括在编码树单元或合并处理区域中的当前块执行运动补偿预测时, 可不将该块的运动信息添加到运动信息表HmvpCandList。作为替代, 可将从该块推导出的临时运动信息候选添加到临时运动信息表HmvpMERCandList。换言之, 添加到临时运动信息表的临时运动信息候选可不被添加到运动信息表。因此, 运动信息表可不包括基于包括当前块的编码树单元或合并处理区中所包括的块的运动信息推导出的运动信息候选。

[0269] 可选地, 可将合并处理区域中包括的块中的一些块的运动信息添加到临时运动信息表。作为示例, 仅合并处理区域内的预定义位置处的块可被用于更新运动信息表。所述预定义位置可包括位于合并处理区域的左上方的块、位于合并处理区域的右上方的块、位于合并处理区域的左下方的块、位于合并处理区域的右下方的块、位于合并处理区域的中心的块、与合并处理区域的右边界相邻的块和与合并处理区域的下边界相邻的块中的至少一个。作为示例, 可将与合并处理区域的右下角相邻的块的运动信息添加到临时运动信息表, 并且可不将其它块的运动信息添加到临时运动信息表。

[0270] 临时运动信息表能够包括的临时运动信息候选的最大数量可被设置为等于运动信息表能够包括的运动信息候选的最大数量。可选地, 可根据编码树单元或合并处理区域的尺寸确定临时运动信息表能够包括的临时运动信息候选的最大数量。可选地, 临时运动信息表能够包括的临时运动信息候选的最大数量可被设置为小于运动信息表能够包括的运动信息候选的最大数量。

[0271] 包括在编码树单元或合并处理区域中的当前块可被设置为不使用关于对应的编码树单元或合并处理区域的临时运动信息表。换言之, 当包括在当前块的合并候选列表中的合并候选的数量小于阈值时, 可将包括在运动信息表中的运动信息候选添加到合并候选列表, 并且可不将包括在临时运动信息表中的临时运动信息候选添加到合并候选列表。因此, 包括在与当前块相同的编码树单元或相同的合并处理区域中的其他块的运动信息可不被用于当前块的运动补偿预测。

[0272] 如果完成了对编码树单元或合并处理区域中包括的所有块的编码/解码, 则可使运动信息表和临时运动信息表一致。

[0273] 图19是示出使运动信息表和临时运动信息表一致的示例的示意图。

[0274] 如果完成了对包括在编码树单元或合并处理区域中的所有块的编码/解码, 则如在图19中所示的示例中, 可在运动信息表中更新包括在临时运动信息表中的临时运动信息候选。

[0275] 就此而言, 包括在临时运动信息表中的临时运动信息候选可按照插入在临时运动信息表中的顺序(换言之, 按照索引值的升序或降序)被添加到运动信息表。

[0276] 在另一示例中, 包括在临时运动信息表中的临时运动信息候选可按照预定义顺序被添加到运动信息表。就此而言, 可按照合并处理区域或编码树单元中的编码块的扫描顺序来确定所述预定义顺序。扫描顺序可以是光栅扫描、水平扫描、垂直扫描或Z字形扫描中的至少一个。可选地, 可基于每个块的运动信息或具有相同运动信息的块的数量来确定所述预定义顺序。

[0277] 可选地, 包括单向运动信息的临时运动信息候选可在包括双向运动信息的临时运动信息候选之前被添加到运动信息表。相反, 包括双向运动信息的临时运动信息候选可在包括单向运动信息的临时运动信息候选之前被添加到运动信息表。

[0278] 可选地, 可按照在合并处理区域或编码树单元中高使用频率或低使用频率的顺序将临时运动信息候选添加到运动信息表。

[0279] 在包括在临时运动信息表中的临时运动信息候选被添加到运动信息表的情况下, 可执行针对临时运动信息候选的冗余校验。在示例中, 当与包括在临时运动信息表中的临时运动信息候选相同的运动信息候选被预存储在运动信息表中时, 可不将该临时运动信息

候选添加到运动信息表。就此而言,可针对包括在运动信息表中的运动信息候选中的一部分运动信息候选执行冗余校验。在示例中,可针对索引超过或低于阈值的运动信息候选执行冗余校验。在示例中,当临时运动信息候选等于索引高于预定义值的运动信息候选时,可不将该临时运动信息候选添加到运动信息表。

[0280] 这可限制将从与当前块相同的编码树单元或相同的合并处理区域中包括的块推导出的运动信息候选用作当前块的合并候选。为此,可针对运动信息候选另外存储块的地址信息。块的地址信息可包括块的位置、块的地址、块的索引、包括块的合并处理区域的位置、包括块的合并处理区域的地址、包括块的合并处理区域的索引、包括块的编码树区域的位置、包括块的编码树区域的地址或包括块的编码树区域的索引中的至少一个。

[0281] 帧内预测通过使用已经被编码/解码且在当前块周围的重建样点来对当前块进行预测。就此而言,在应用环内滤波器之前的重建样点可被用于当前块的帧内预测。

[0282] 帧内预测方法包括基于矩阵的帧内预测以及利用邻近重建样点的根据方向的帧内预测。可以以比特流用信号发送指示当前块的帧内预测方法的信息。所述信息可以是1比特标志。可选地,可基于当前块的位置、当前块的尺寸、当前块的形状或邻近块的帧内预测方法中的至少一个来确定当前块的帧内预测。在示例中,当当前块跨越画面边界时,可设置为使得基于矩阵的帧内预测方法不被应用于当前块。

[0283] 基于矩阵的帧内预测方法是一种基于存储在编码器和解码器中的矩阵与当前块周围的重建样点的矩阵乘积来获得当前块的预测块的方法。可以以比特流用信号发送用于指定多个预存储的矩阵中的任意一个矩阵的信息。解码器可基于上述信息和当前块的尺寸来确定用于对当前块执行帧内预测的矩阵。

[0284] 一般帧内预测是一种基于非方向帧内预测模式或方向帧内预测模式获得当前块的预测块的方法。在下文中,将参照附图详细描述基于一般帧内预测的帧内预测的处理。

[0285] 图20是根据本公开的实施例的帧内预测方法的流程图。

[0286] 可确定当前块的参考样点线S2001。参考样点线表示与当前块的上方和/或左侧间隔的第k条线中包括的一组参考样点。可从当前块周围的被编码/解码的重建样点推导参考样点。

[0287] 可以以比特流用信号发送标识多条参考样点线中的针对当前块的参考样点线的索引信息。在示例中,可以以比特流用信号发送用于指定当前块的参考样点线的索引信息intra_luma_ref_idx。所述索引信息可按编码块被用信号发送。

[0288] 多条参考样点线可包括当前块的上方和/或左侧处的第一条线、第二条线或第三条线中的至少一个。多条参考样点线中的包括与当前块的上方相邻的行和与当前块的左侧相邻的列的参考样点线可被称为相邻参考样点线,并且其它参考样点线可被称为非相邻参考样点线。

[0289] 表1示出分配给每条候选参考样点线的索引。

[0290] 【表1】

[0291]	索引 (intra_luma_ref_idx)	参考样点线
	0	相邻参考样点线

[0292]	1	第一非相邻参考样点线
	2	第二非相邻参考样点线

[0293] 基于当前块的位置、尺寸、形状或邻近块的预测编码模式中的至少一个,可确定当前块的参考样点线。在一个示例中,当当前块与画面、并行块、条带或编码树单元的边界邻接时,相邻参考样点线可被确定为当前块的参考样点线。参考样点线可包括位于当前块的上方的上参考样点和位于当前块的左侧的左参考样点。可从当前块周围的重建样点推导上参考样点和左参考样点。重建样点可处于应用环内滤波器之前的状态。

[0294] 接下来,可确定当前块的帧内预测模式S2002。对于当前块的帧内预测模式,可将非方向帧内预测模式或方向帧内预测模式中的至少一个确定为当前块的帧内预测模式。非方向帧内预测模式包括平面和DC,并且方向帧内预测模式包括从左下对角线方向到右上对角线方向的33或65个模式。

[0295] 图21是示出帧内预测模式的示图。

[0296] 图21(a)示出35个帧内预测模式。图21(b)示出67个帧内预测模式。

[0297] 可定义比图21中所示的帧内预测模式的数量更多或更少数量的帧内预测模式。

[0298] 基于与当前块相邻的邻近块的帧内预测模式,可设置MPM(最可能模式)。就此而言,邻近块可包括与当前块的左侧相邻的左侧邻近块和与当前块的上方相邻的上方邻近块。

[0299] 可在编码器和解码器中预先设置包括在MPM列表中的MPM的数量。在示例中,MPM的数量可以是3、4、5或6。可选地,可以以比特流用信号发送表示MPM的数量的信息。可选地,基于邻近块的预测编码模式、当前块的尺寸、形状或参考样点线索引中的至少一个,可确定MPM的数量。在示例中,当相邻参考样点线被确定为当前块的参考样点线时N个MPM可被使用,而当非相邻参考样点线被确定为当前块的参考样点线时M个MPM可被使用。在M是小于N的自然数时,在示例中,N可以是6并且M可以是5、4或3。因此,当当前块的参考样点线的索引为0且MPM标志为真时,当前块的帧内预测模式可被确定为6个候选帧内预测模式中的任意一个,而当当前块的参考样点线的索引大于0且MPM标志为真时,当前块的帧内预测模式可被确定为5个候选帧内预测模式中的任意一个。

[0300] 可选地,可使用固定数量(例如,6或5)个MPM候选,而不管当前块的参考样点线的索引如何。

[0301] 当基于矩阵的帧内预测被应用于邻近块时,可通过将邻近块的帧内预测模式推断为平面来推导MPM候选。

[0302] 当帧内BDPCM被应用于邻近块时,可通过将邻近块的帧内预测模式推断为默认模式来推导MPM候选。在这种情况下,默认模式可以是DC、平面、垂直方向或水平方向中的至少一个。

[0303] 可选地,基于邻近块的帧内BDPCM应用方向,可确定邻近块的帧内预测模式。在示例中,当水平方向上的帧内BDPCM被应用于邻近块时,邻近块的帧内预测模式可被推断为在水平方向上。另一方面,当垂直方向上的帧内BDPCM被应用于邻近块时,邻近块的帧内预测模式可被推断为在垂直方向上。

[0304] 可生成包括多个MPM的MPM列表,并且可以以比特流用信号发送指示与当前块的帧

内预测模式相同的MPM是否被包括在MPM列表中的信息。在所述信息是1比特标志时,它可被称为MPM标志。当MPM标志表示与当前块相同的MPM被包括在MPM列表中时,可以以比特流用信号发送标识MPM中的一个MPM的索引信息。在示例中,可以以比特流用信号发送指定多个MPM中的任意一个MPM的索引信息`mpm_idx`。由索引信息指定的MPM可被设置为当前块的帧内预测模式。当MPM标志表示与当前块相同的MPM未被包括在MPM列表中时,可以以比特流用信号发送指示除MPM以外的其余帧内预测模式中的任意一个帧内预测模式的其余模式信息。其余模式信息表示当向除MPM以外的其余帧内预测模式重新分配索引时与当前块的帧内预测模式对应的索引值。解码器可通过按照升序排列MPM并且将其余模式信息与MPM进行比较来确定当前块的帧内预测模式。在示例中,当其余模式信息等于或小于MPM时,可通过将1与其余模式信息相加来推导当前块的帧内预测模式。

[0305] 在推导当前块的帧内预测模式时,可省略将一部分MPM与其余模式信息进行比较的操作。在示例中,可从比较目标排除MPM中的非方向帧内预测模式中的MPM。当非方向帧内预测模式被设置为MPM时,其余模式信息显然指示方向帧内预测模式,所以可通过将除了非方向帧内预测模式以外的其余MPM与其余模式信息进行比较来推导当前块的帧内预测模式。作为从比较目标排除非方向帧内预测模式的替代,在将非方向帧内预测模式的数量与其余模式信息相加之后,可将结果值与其余MPM进行比较。

[0306] 作为将默认模式设置为MPM的替代,可以以比特流用信号发送指示当前块的帧内预测模式是否是默认模式的信息。所述信息是1比特标志,并且所述标志可被称为默认模式标志。仅当MPM标志表示与当前块相同的MPM被包括在MPM列表中时,可用信号发送默认模式标志。如上所述,默认模式可包括平面、DC、垂直方向模式或水平方向模式中的至少一个。在示例中,当平面被设置为默认模式时,默认模式标志可指示当前块的帧内预测模式是否为平面。当默认模式标志指示当前块的帧内预测模式不是默认模式时,MPM中的由索引信息指示的一个MPM可被设置为当前块的帧内预测模式。

[0307] 当使用默认模式标志时,可将与默认模式相同的帧内预测模式设置为不被设置为MPM。在示例中,当默认模式标志指示当前块的帧内预测模式是否是平面时,可通过使用除了与平面对应的MPM之外的5个MPM来推导当前块的帧内预测模式。

[0308] 当多个帧内预测模式被设置为默认模式时,还可用信号发送指示默认模式中的任意一个默认模式的索引信息。当前块的帧内预测模式可被设置为由索引信息指示的默认模式。

[0309] 当当前块的参考样点线的索引不为0时,可设置为不使用默认模式。在示例中,当非相邻参考样点线被确定为当前块的参考样点线时,可设置为不使用诸如DC模式或平面模式的非方向帧内预测模式。因此,当参考样点线的索引不为0时,可不用信号发送默认模式标志,并且可将默认模式标志的值推断为预定义值(即,假)。

[0310] 当当前块的帧内预测模式被确定时,可基于所确定的帧内预测模式来获得针对当前块的预测样点`S2003`。

[0311] 当DC模式被选择时,可基于参考样点的平均值来生成针对当前块的预测样点。详细地,可基于参考样点的平均值来生成预测块内的所有样点的值。可使用与当前块的上方相邻的上参考样点和与当前块的左侧相邻的左参考样点中的至少一个来推导平均值。

[0312] 在推导平均值时使用的参考样点的数量或范围可基于当前块的形状而变化。在示

例中,当当前块是宽度大于高度的非正方形块时,可通过使用上参考样点来计算平均值。反之,当当前块是宽度小于高度的非正方形块时,可通过使用左参考样点来计算平均值。换言之,当当前块的宽度和高度不同时,可使用与更大长度相邻的参考样点来计算平均值。可选地,可基于当前块的宽高比来确定是通过使用上参考样点还是通过使用左参考样点来计算平均值。

[0313] 当平面模式被选择时,可通过使用水平方向预测样点和垂直方向预测样点来获得预测样点。就此而言,可基于与预测样点位于同一水平线的左参考样点和右参考样点来获得水平方向预测样点,并且可基于与预测样点位于同一垂直线的上参考样点和下参考样点来获得垂直方向预测样点。就此而言,可通过复制与当前块的右上角相邻的参考样点来生成右参考样点,并且可通过复制与当前块的左下角相邻的参考样点来生成下参考样点。可基于左参考样点和右参考样点的加权和来获得水平方向预测样点,并且可基于上参考样点和下参考样点的加权和来获得垂直方向预测样点。就此而言,可根据预测样点的位置来确定分配给每个参考样点的加权因子。可基于水平方向预测样点和垂直方向预测样点的平均值或加权和来获得预测样点。当使用加权和时,可基于预测样点的位置来确定分配给水平方向预测样点和垂直方向预测样点的加权因子。

[0314] 当方向预测模式被选择时,可确定表示所选的方向预测模式的预测方向(或预测角度)的参数。下面的表2表示针对每个帧内预测模式的帧内方向参数intraPredAng。

[0315] 【表2】

[0316]	PredModeIntra	1	2	3	4	5	6	7
	IntraPredAng	-	32	26	21	17	13	9
	PredModeIntra	8	9	10	11	12	13	14
	IntraPredAng	5	2	0	-2	-5	-9	-13
	PredModeIntra	15	16	17	18	19	20	21
	IntraPredAng	-17	-21	-26	-32	-26	-21	-17
	PredModeIntra	22	23	24	25	26	27	28
	IntraPredAng	-13	-9	-5	-2	0	2	5
	PredModeIntra	29	30	31	32	33	34	
	IntraPredAng	9	13	17	21	26	32	

[0317] 表2表示每个帧内预测模式的帧内方向参数,其中,当35个帧内预测模式被定义时,每个帧内预测模式的索引是2至34中的一个。当方向帧内预测模式被定义为多于33时,可通过对表2进行细分来设置每个帧内预测模式的帧内方向参数。针对当前块的上参考样点和左参考样点被排列为一条线,然后可基于帧内方向参数的值获得预测样点。就此而言,当帧内方向参数的值为负值时,左参考样点和上参考样点可被排列为一条线。

[0318] 图22和图23是分别示出参考样点被排列为一条线的一维排列的示例的示图。

[0319] 图22是示出参考样点在垂直方向上被排列的垂直方向一维排列的示图,并且图23是示出参考样点在水平方向上被排列的水平方向一维排列的示图。将在定义了35个帧内预测模式的假设下描述图22和图23的示例。

[0320] 当帧内预测模式索引是11至18中的任意一个时,可应用水平方向一维排列,其中,上参考样点被逆时针旋转,并且当帧内预测模式索引是19至25中的任意一个时,可应用垂

直方向一维排列,其中,左参考样点被顺时针旋转。当将参考样点排列为一条线时,可考虑帧内预测模式角度。

[0321] 可基于帧内方向参数确定参考样点确定参数。参考样点确定参数可包括用于指定样点的参考样点索引以及用于确定应用于参考样点的加权因子的加权因子参数。

[0322] 可通过下面的等式2和等式3分别获得参考样点索引*iIdx*和加权因子参数*iFact*。

[0323] **【等式2】**

$$[0324] \quad iIdx = (y+1) * P_{ang} / 32$$

[0325] **【等式3】**

$$[0326] \quad i_{fact} = [(y+1) * P_{ang}] \& 31$$

[0327] 在等式2和等式3中, P_{ang} 表示帧内方向参数。由参考样点索引*iIdx*指定的参考样点对应于整数像素。

[0328] 为了推导预测样点,可指定至少一个参考样点。详细地,根据预测模式的斜率,可指定用于推导预测样点的参考样点的位置。在示例中,可通过使用参考样点索引*iIdx*来指定用于推导预测样点的参考样点。

[0329] 就此而言,当帧内预测模式的斜率不由一个参考样点表示时,可通过对多个参考样点执行插值来生成预测样点。在示例中,当帧内预测模式的斜率是预测样点与第一参考样点之间的斜率和预测样点与第二参考样点之间的斜率之间的值时,可通过对第一参考样点和第二参考样点执行插值来获得预测样点。换言之,当根据帧内预测角度的角度线不穿过位于整数像素的参考样点时,可通过对位置与角度线穿过的位置的左侧和右侧或者上方和下方相邻的参考样点执行插值来获得预测样点。

[0330] 下面的等式4表示基于参考样点获得预测样点的示例。

[0331] **【等式4】**

$$[0332] \quad P(x,y) = ((32 - i_{fact}) / 32) * Ref_1D(x + iIdx + 1) + (i_{fact} / 32) * Ref_1D(x + iIdx + 2)$$

[0333] 在等式4中, P 表示预测样点,并且 Ref_1D 表示被排列为一条线的参考样点中的任意一个参考样点。就此而言,可通过预测样点的位置(x,y)和参考样点索引*iIdx*来确定参考样点的位置。

[0334] 当帧内预测模式的斜率可能由一个参考样点来表示时,加权因子参数*iFact*被设置为0。因此,等式4可被简化为下面的等式5。

[0335] **【等式5】**

$$[0336] \quad P(x,y) = Ref_1D(x + iIdx + 1)$$

[0337] 可基于多个帧内预测模式执行针对当前块的帧内预测。在示例中,可针对每个预测样点推导帧内预测模式,并且可基于分配给每个预测样点的帧内预测模式推导预测样点。

[0338] 可选地,可针对每个区域推导帧内预测模式,可基于分配给每个区域的帧内预测模式执行针对每个区域的帧内预测。就此而言,所述区域可包括至少一个样点。可基于当前块的尺寸、当前块的形状和当前块的帧内预测模式中的至少一个自适应地确定所述区域的尺寸和形状中的至少一个。可选地,可在编码器和解码器中独立于当前块的尺寸或形状预先定义所述区域的尺寸和形状中的至少一个。

[0339] 图24是示出在方向帧内预测模式与平行于x轴的直线之间形成的度数的示图。

[0340] 作为图24中所示的示例,方向预测模式可存在于左下对角线方向与右上对角线方向之间。在描述在x轴与方向预测模式之间形成的度数时,方向预测模式可存在于从45度(左下对角线方向)到-135度(右上对角线方向)。

[0341] 当当前块是非正方形时,可能存在如下情况:通过使用位于根据帧内预测角度的角度线处的参考样点中的位置比与根据针对当前块的帧内预测模式的预测样点接近的参考样点更远的参考样点来推导预测样点。

[0342] 图25是示出当当前块是非正方形时获得预测样点的方面的示意图。

[0343] 在示例中,作为图27(a)中所示的示例,假设当前块是宽度大于高度的非正方形,并且针对当前块的帧内预测模式是具有从0度至45度的角度的方向帧内预测模式。在上述情况下,当在以根据上述度数的角度模式定位的参考样点中推导当前块的右侧列周围的预测样点A时,可能存在远离该预测样点的左参考样点L被使用的情况,而非使用与该预测样点接近的上参考样点T。

[0344] 在另一示例中,作为图25(b)中所示的示例,假设当前块是高度大于宽度的非正方形,并且针对当前块的帧内预测模式是从-90度到-135度的方向帧内预测模式。在上述情况下,当在以根据上述度数的角度模式定位的参考样点中推导当前块的下方行周围的预测样点A时,可能存在远离该预测样点的上参考样点T被使用的情况,而非使用与该预测样点接近的左参考样点L。

[0345] 为了解决上述问题,当当前块是非正方形时,可利用相反方向的方向帧内预测模式替换针对当前块的方向帧内预测模式。因此,对于非正方形块,可使用角度大于或小于图21中所示的方向预测模式的角度的方向帧内预测模式。上述方向帧内预测模式可被定义为广角帧内预测模式。广角帧内预测模式表示不属于45度到-135度的范围的方向帧内预测模式。

[0346] 图26是示出广角帧内预测模式的示意图。

[0347] 在图26中所示的示例中,索引从-1到-14的帧内预测模式以及索引从67到80的帧内预测模式表示广角帧内预测模式。

[0348] 在图26中,示出了角度大于45度的14个广角帧内预测模式(从-1到-14)以及角度小于-135度的14个广角帧内预测模式(从67到80)。然而,可定义更多或更少数量的广角帧内预测模式。

[0349] 当使用广角帧内预测模式时,上参考样点的长度可被设置为 $2W+1$,并且左参考样点的长度可被设置为 $2H+1$ 。

[0350] 通过使用广角帧内预测模式,可通过使用参考样点T来预测图25(a)中所示的样点A,并且可通过使用参考样点L来预测图25(b)中所示的样点A。

[0351] 加上传统帧内预测模式和N个广角帧内预测模式,可使用总共 $67+N$ 个帧内预测模式。在示例中,表3表示当定义了20个广角帧内预测模式时针对帧内预测模式的帧内方向参数。

[0352] 【表3】

[0353]	PredModeIntra	-10	-9	-8	-7	-6	-5	-4	-3	-2
--------	---------------	-----	----	----	----	----	----	----	----	----

[0354]

intraPredAngle	114	93	79	68	60	54	49	45	39
PredModeIntra	-1	2	3	4	5	6	7	8	9
intraPredAngle	35	32	29	26	23	21	19	17	15
PredModeIntra	10	11	12	13	14	15	16	17	18
intraPredAngle	13	11	9	7	5	3	2	1	0
PredModeIntra	19	20	21	22	23	24	25	26	27
intraPredAngle	-1	-2	-3	-5	-7	-9	-11	-13	-15
PredModeIntra	28	29	30	31	32	33	34	35	36
intraPredAngle	-17	-19	-21	-23	-26	-29	-32	-29	-26
PredModeIntra	37	38	39	40	41	42	43	44	45
intraPredAngle	-23	-21	-19	-17	-15	-13	-11	-9	-7
PredModeIntra	46	47	48	49	50	51	52	53	54
intraPredAngle	-5	-3	-2	-1	0	1	2	3	5
PredModeIntra	55	56	57	58	59	60	61	62	63
intraPredAngle	7	9	11	13	15	17	19	21	23
PredModeIntra	64	65	66	67	68	69	70	71	72
intraPredAngle	26	29	32	35	39	45	49	54	60
PredModeIntra	73	74	75	76					
intraPredAngle	68	79	93	114					

[0355] 当当前块是非正方形并且在S2502获得的针对当前块的帧内预测模式属于变换范围时,可将针对当前块的帧内预测模式变换为广角帧内预测模式。可基于当前块的尺寸、形状或比率中的至少一个来确定变换范围。就此而言,所述比率可表示当前块的宽高比。当当前块是宽度大于高度的非正方形时,可将变换范围设置为从右上对角线方向的帧内预测模式索引(例如,66)到(右上对角线方向的帧内预测模式索引-N)。就此而言,可基于当前块的比率来确定N。当针对当前块的帧内预测模式属于变换范围时,该帧内预测模式可被变换为广角帧内预测模式。变换可以是帧内预测模式减去预定义值,并且所述预定义值可以是排除广角帧内预测模式的帧内预测模式的总数(例如,67)。

[0356] 在以上示例中,从编号66到编号53的帧内预测模式可分别被变换为从编号-1到编号-14的广角帧内预测模式。

[0357] 当当前块是高度大于宽度的非正方形时,可将变换范围设置为从左下对角线方向的帧内预测模式索引(例如,2)到(左下对角线方向的帧内预测模式索引+M)。就此而言,可基于当前块的比率确定M。当针对当前块的帧内预测模式属于变换范围时,该帧内预测模式可被变换为广角帧内预测模式。变换可以是将预定义值与帧内预测模式相加,并且所述预定义值可以是排除广角帧内预测模式的方向帧内预测模式的总数(例如,65)。

[0358] 在以上示例中,从编号2到编号15的帧内预测模式可分别被变换为从编号67到编号80的广角帧内预测模式。

[0359] 在下文中,属于变换范围的帧内预测模式被称为广角帧内预测替换模式。

[0360] 可基于当前块的比率确定变换范围。在示例中,表4和表5分别示出定义了除了广角帧内预测模式之外的35个帧内预测模式的情况以及定义了除了广角帧内预测模式之外的67个帧内预测模式的情况的变换范围。

[0361] 【表4】

条件	替换帧内预测模式
$W/H=2$	模式2,3,4
$W/H>2$	模式2,3,4,5,6
$W/H=1$	无
$H/W=1/2$	模式32,33,34
$H/W<1/2$	模式30,31,32,33,34

[0363] 【表5】

条件	替换帧内预测模式
$W/H=2$	模式2,3,4,5,6,7
$W/H>2$	模式2,3,4,5,6,7,8,9,10,11
$W/H=1$	无
$H/W=1/2$	模式61,62,63,64,65,66
$H/W<1/2$	模式57,58,59,60,61,62,63,64,65,66

[0365] 如表4和表5中所示的示例,包括在变换范围中的广角帧内预测替换模式的数量可根据当前块的比率而变化。可进一步对当前块的比率进行细分以设置如下表6中所示的变换范围。

[0366] 【表6】

条件	替换帧内预测模式
$W/H=16$	模式12,13,14,15
$W/H=8$	模式12,13
$W/H=4$	模式2,3,4,5,6,7,8,9,10,11
$H/W=2$	模式2,3,4,5,6,7
$H/W=1$	无
$W/H=1/2$	模式61,62,63,64,65,66
$W/H=1/4$	模式57,58,59,60,61,62,63,64,65,66
$W/H=1/8$	模式55,56
$H/W=1/16$	模式53,54,55,56

[0368] 当非相邻参考样点线被确定为针对当前块的参考样点线时,或者当用于选择多条参考样点线中的一条参考样点线的多线帧内预测编码方法被使用时,预测方法可被配置为不使用广角帧内预测模式。也就是说,虽然当前块具有非正方形形状并且针对当前块的帧内预测模式属于变换范围,但是针对当前块的帧内预测模式可不被变换为广角帧内预测模

式。可选地,当针对当前块的帧内预测模式被确定为广角帧内预测模式时,预测方法可被配置为使得非相邻参考样点线不可用作针对当前块的参考样点线,或者可被配置为不用于选择多条参考样点线中的一条参考样点线的多线帧内预测编码方法。当不使用多线帧内预测编码方法时,可将相邻参考样点线确定为当前块的参考样点线。

[0369] 当不使用广角帧内预测模式时,可将 refW 和 refH 中的每一个设置为 nTbW 和 nTbH 的总和。因此,与当前块相距 i 条线的非相邻参考样点线可包括除了左上参考样点之外的 $(\text{nTbW}+\text{nTbH}+\text{offsetX}[i])$ 个上参考样点和 $(\text{nTbW}+\text{nTbH}+\text{offsetY}[i])$ 个左参考样点。也就是说,与当前块相距 i 条线的非相邻参考样点线可包括 $(2\text{nTbW}+2\text{nTbH}+\text{offsetX}[i]+\text{offsetY}[i]+1)$ 个参考样点。例如,当 whRatio 的值大于1时, offsetX 的值可被设置为大于 offsetY 的值。在一个示例中,当 offsetX 的值可被设置为1时, offsetY 的值可被设置为0。反之,当 whRatio 的值小于1时, offsetY 的值可被设置为大于 offsetX 的值。在一个示例中, offsetX 的值可被设置为0,并且 offsetY 的值可被设置为1。

[0370] 由于除了传统帧内预测模式之外还使用广角帧内预测模式,因此用于对广角帧内预测模式进行编码的资源可能增加,并且因此编码效率可能降低。因此,不是按原样对广角帧内预测模式进行编码,而是对针对广角帧内预测模式的替换帧内预测模式进行编码,以便提高编码效率。

[0371] 在示例中,当通过使用编号67的广角帧内预测模式对当前块进行编码时,可将编号2(编号67的广角替换帧内预测模式)编码为针对当前块的帧内预测模式。此外,当通过使用编号-1的广角帧内预测模式对当前块进行编码时,可将编号66(编号-1的广角替换帧内预测模式)编码为针对当前块的帧内预测模式。

[0372] 解码器可对针对当前块的帧内预测模式进行解码,并确定经过解码的帧内预测模式是否属于变换范围。当经过解码的帧内预测模式是广角替换帧内预测模式时,该帧内预测模式可被变换为广角帧内预测模式。

[0373] 可选地,当通过广角帧内预测模式对当前块进行编码时,该广角帧内预测模式可按原样被编码。

[0374] 可基于上述MPM列表执行帧内预测模式的编码。具体地,当在广角帧内预测模式下对邻近块进行编码时,可基于与该广角帧内预测模式对应的广角替换帧内预测模式来设置MPM。

[0375] 编码块或变换块可被分区为多个子块(或子分区)。当编码块或变换块被分区为多个子块时,可针对每个子块执行预测、变换和量化。将编码块或变换块分区为多个子块的操作可被定义为子分区帧内编码方法。

[0376] 可以以比特流用信号发送表示子分区帧内编码方法是否被应用的信息。该信息可以是1比特标志。在示例中,可以以比特流用信号发送“ $\text{intra_subpartitions_mode_flag}$ ”(表示编码块或变换块是否被分区为多个子块的语法因子)。

[0377] 可选地,可基于编码块或变换块的尺寸、形状或帧内预测模式中的至少一个来确定子分区帧内编码方法是否被应用。在示例中,当编码块的帧内预测模式是非方向帧内预测模式(例如,平面或DC)或者预定义方向帧内预测模式(例如,水平方向上的帧内预测模式、垂直方向上的帧内预测模式或对角线方向上的帧内预测模式)时,可不应用子分区帧内编码方法。可选地,当编码块的尺寸小于阈值时,子分区帧内编码方法可被设置为不被使

用。

[0378] 可选地,当基于编码块的帧内预测模式执行针对子块的帧内预测时,可基于包括在邻近子块中的重建样点是否应被用作子块的帧内预测中的参考样点来确定子分区帧内编码方法是否被应用。在示例中,当编码块的帧内预测模式是对角线方向上的帧内预测模式或广角帧内预测模式,并且在基于该帧内预测模式针对子块执行帧内预测的操作中邻近子块可不被用作参考样点时,子分区帧内编码方法可被设置为不被使用。

[0379] 可选地,当编码块的高宽比等于或大于阈值或者等于或小于阈值时,子分区帧内编码方法可被设置为不被使用。可选地,当编码块的高度或宽度中的至少一个等于或小于阈值时,子分区帧内编码方法可不被使用。在示例中,当编码块的宽度或高度等于或小于阈值时,或者当编码块的高度和宽度两者等于或小于阈值时,子分区帧内编码方法可不被使用。可选地,当编码块中包括的样点的数量等于或小于阈值时,子分区帧内编码方法可不被使用。阈值可在编码器和解码器中具有预定义值。可选地,可以以比特流用信号发送用于确定阈值的信息。

[0380] 可选地,可基于编码块或变换块的尺寸、形状或帧内预测模式中的至少一个来确定表示子分区帧内编码方法是否被应用的标志是否被用信号发送。在示例中,仅当编码块的高度和宽度两者等于或小于阈值时以及/或者当编码块的尺寸等于或大于阈值时,表示子分区帧内编码方法是否被应用的标志可被编码并被用信号发送。当表示子分区帧内编码方法是否被应用的标志未被编码时,可不应用子分区帧内编码方法。

[0381] 当子分区帧内编码方法未被使用时,可省略用信号发送语法因子 `intra_subpartitions_mode_flag` 的操作。当省略用信号发送该标志的操作时,该标志可被视为表示子分区帧内编码方法未被应用。

[0382] 当子分区帧内编码方法被应用时,可确定编码块或变换块的分区形状。在这种情况下,分区形状表示编码块或变换块的分区方向。在示例中,垂直方向上的分区可表示通过使用至少一条垂直线对编码块或变换块进行分区,并且水平方向上的分区可表示通过使用至少一条水平线对编码块或变换块进行分区。

[0383] 图27是示出垂直方向上的分区和水平方向上的分区的示例的示意图。

[0384] 图27(a)表示编码块被分区为2个子块的示例,图27(b)表示编码块被分区为4个子块的示例。

[0385] 可以以比特流用信号发送用于确定编码块或变换块的分区形状的信息。在示例中,可用信号发送表示垂直方向上的分区是否被应用于编码块或变换块或者水平方向上的分区是否被应用于编码块或变换块的信息。该信息可以是1比特标志 `intra_subpart_type_flag`。当该标志的值为1时,这表示编码块或变换块在水平方向上被分区,并且当该标志的值为0时,这表示编码块或变换块在垂直方向上被分区。

[0386] 可选地,可基于编码块或变换块的尺寸、形状或帧内预测模式来确定编码块或变换块的分区形状。在示例中,可基于编码块的宽高比来确定编码块的分区形状。例如,当表示编码块的高宽比的值 `whRatio` 等于或大于第一阈值时,可将垂直方向上的分区应用于编码块。否则,可将水平方向上的分区应用于编码块。

[0387] 图28是示出确定编码块的分区形状的示例的示意图。

[0388] 为了便于描述,假设第一阈值为2。在图28(a)中所示的示例中,编码块的 `whRatio`

为小于第一阈值的1。因此,可省略对表示编码块的分区形状的信息的编码,并且可将水平方向上的分区应用于编码块。

[0389] 在图28(b)中所示的示例中,编码块的whRatio为与第一阈值相同的2。因此,可省略对表示编码块的分区形状的信息的编码,并且可将垂直方向上的分区应用于编码块。

[0390] 可通过使用符号与第一阈值相反的第二阈值来确定编码块的分区形状。在示例中,当whRatio的值等于或小于第二阈值时,可将水平方向上的分区应用于编码块,否则,可将垂直方向上的分区应用于编码块。第一阈值和第二阈值的绝对值可相同,并且它们的符号可不同。在示例中,当第一阈值是N(在这种情况下,N是整数,诸如1、2、4等)时,第二阈值可以是-N。

[0391] 图29是示出确定编码块的分区形状的示例的示意图。

[0392] 为了便于描述,假设第二阈值为-2。在图29(a)中所示的示例中,编码块的whRatio为大于第二阈值的-1。因此,可省略对表示编码块的分区形状的信息的编码,并且可将垂直方向上的分区应用于编码块。

[0393] 在图29(b)中所示的示例中,编码块的whRatio为与第二阈值相同的-2。因此,可省略对表示编码块的分区形状的信息的编码,并且可将水平方向上的分区应用于编码块。

[0394] 可选地,可基于第一阈值和第二阈值确定编码块的分区形状。在示例中,当whRatio的值等于或大于第一阈值时,可将水平方向上的分区应用于编码块,并且当whRatio的值等于或小于第二阈值时,可将垂直方向上的分区应用于编码块。当whRatio的值在第一阈值与第二阈值之间时,可通过对比特流中的信息进行解析来确定当前块的分区形状。

[0395] 可在编码器和解码器中预定义第一阈值和第二阈值。可选地,可按每个序列、画面或条带定义第一阈值和第二阈值。

[0396] 可选地,可基于编码块或变换块的尺寸来确定分区形状。在示例中,当编码块的尺寸为 $N \times n$ 时,可应用垂直方向上的分区,并且当编码块的尺寸为 $n \times N$ 时,可应用水平方向上的分区。在这种情况下,n可以是小于N的自然数。N和/或n可以是在编码器和解码器中的预定义值。可选地,可以以比特流用信号发送用于确定N和/或n的信息。在示例中,N可以是32、64、128或256等。因此,当编码块的尺寸是 $128 \times n$ (在这种情况下,n是自然数,诸如16、32或64等)时,可应用垂直方向上的分区,并且当编码块的尺寸是 $n \times 128$ 时,可应用水平方向上的分区。

[0397] 可选地,可基于编码块或变换块的帧内预测模式来确定编码块或变换块的分区形状。在示例中,当编码块的帧内预测模式在水平方向上或在与水平方向相似的方向上时,可将垂直方向上的分区应用于编码块。在这种情况下,与水平方向相似的方向上的帧内预测模式表示与水平方向上的帧内预测模式(例如,图21(b)中所示的INTRA_ANGULAR18)的索引差值等于或小于阈值的帧内预测模式(例如,INTRA_ANGULAR18 $\pm$ N)。另一方面,当编码块的帧内预测模式在垂直方向上或在与垂直方向相似的方向上时,可将水平方向上的分区应用于编码块。在这种情况下,与垂直方向相似的方向上的帧内预测模式表示与垂直方向上的帧内预测模式(例如,图21(b)中所示的INTRA_ANGULAR50)的索引差值等于或小于阈值的帧内预测模式(例如,INTRA_ANGULAR50 $\pm$ N)。在这种情况下,阈值N可以是编码器和解码器中的预定义值。可选地,可在序列级、画面级或条带级用信号发送用于确定阈值N的信息。

[0398] 当垂直方向上的分区和水平方向上的分区都可用时,可通过对表示编码块的划分形状的信息进行解析来确定编码块的划分形状。

[0399] 可基于编码块或变换块的尺寸或形状中的至少一个确定子块的数量。在示例中,当编码块的宽度或高度中的任意一个为8且另一个为4时,可将编码块分区为2个子块。另一方面,当编码块的宽度和高度都等于或大于8时,或者当编码块的宽度或高度中的任意一个大于8时,可将编码块分区为4个子块。总之,当编码块具有 4×4 尺寸时,可不将编码块分区为子块。当编码块具有 4×8 或 8×4 尺寸时,可将编码块分区为2个子块。否则,可将编码块分区为4个子块。

[0400] 可选地,可以以比特流用信号发送表示子块的尺寸或形状或者子块的数量的信息。子块的尺寸或形状可由表示子块的数量的信息确定。可选地,子块的数量可由表示子块的尺寸或形状的信息确定。

[0401] 当子分区帧内编码方法被应用时,通过对编码块或变换块进行分区而生成的子块可使用相同的帧内预测模式。在示例中,可基于与编码块邻近的邻近块的帧内预测模式推导用于编码块的MPM,并且可基于推导出的MPM确定用于编码块的帧内预测模式。当编码块的帧内预测模式被确定时,每个子块可通过使用所确定的帧内预测模式来执行帧内预测。

[0402] 当子分区帧内编码方法被应用时,MPM中的任意一个MPM可被确定为编码块的帧内预测模式。换言之,虽然没有用信号发送MPM标志,但是当子分区帧内编码方法被应用时,MPM标志可被视为真。

[0403] 可选地,当子分区帧内编码方法被应用时,预定义候选帧内预测模式中的任意一个预定义候选帧内预测模式可被确定为编码块的帧内预测模式。在示例中,水平方向上的帧内预测模式、垂直方向上的帧内预测模式、对角线方向上的帧内预测模式(例如,左上帧内预测模式、右上帧内预测模式或左下帧内预测模式中的至少一个)或非方向帧内预测模式(例如,平面或DC中的至少一个)中的任意一个可被确定为编码块的帧内预测模式。可以以比特流用信号发送指定预定义候选帧内预测模式中的任意一个预定义候选帧内预测模式的索引信息。可选地,根据编码块的分区方向,候选帧内预测模式的数量和/或类型可不同。在示例中,当水平方向上的分区被应用于编码块时,非方向帧内预测模式、垂直方向上的帧内预测模式、左上对角线方向上的帧内预测模式或右上对角线方向上的帧内预测模式中的至少一个可被设置为候选帧内预测模式。另一方面,当垂直方向上的分区被应用于编码块时,非方向帧内预测模式、水平方向上的帧内预测模式、左上对角线方向上的帧内预测模式或左下对角线方向上的帧内预测模式中的至少一个可被设置为候选帧内预测模式。

[0404] 可单独确定子块的量化参数。因此,可不同地设置每个子块的量化参数的值。表示与先前子块的量化参数的差值的信息可被编码以确定每个子块的量化参数。在示例中,针对第N子块,第N子块的量化参数与第N-1子块的量化参数之间的差值可被编码。

[0405] 可通过使用参考样点来执行子块的帧内预测。在这种情况下,可从与子块相邻的邻近块的重建样点推导参考样点。当与子块相邻的邻近块是与所述子块包括在相同的编码块中的其它子块时,可基于所述其它子块的重建样点推导所述子块的参考样点。在示例中,当第一子块位于第二子块的左侧或上方时,可从第一子块的重建样点推导第二子块的参考样点。对此,可不在子块之间应用并行帧内预测。换言之,可针对包括在编码块中的子块依次进行编码/解码。因此,在完成第一子块的编码/解码之后,可执行针对第二子块的帧内预

测。

[0406] 当子分区帧内编码方法被应用时,可设置为不使用选择多条参考样点线候选中的任意一条参考样点线候选的多线帧内预测编码方法。当多线帧内预测编码方法未被使用时,可将与每个子块相邻的相邻参考样点线确定为每个子块的参考样点线。可选地,当当前块的参考样点线的索引大于0时,可省略对表示子分区帧内编码方法是否被应用的语法因子intra_subpartitions_mode_flag的编码。当省略了对语法intra_subpartitions_mode_flag的编码时,子分区帧内编码方法可不被应用。

[0407] 可针对与子块之间的边界相邻的样点执行滤波。可针对预测样点或重建样点执行滤波。在示例中,当假设第二子块与第一子块的右侧相邻时,可通过使用与第一子块的右边界相接的重建样点来对与第二子块的左边界相接的预测样点或重建样点进行滤波。

[0408] 可基于帧内预测模式、子块的尺寸或子块的数量中的至少一个确定滤波器是否将被应用于与子块之间的边界相邻的样点。

[0409] 当通过执行帧内预测来生成预测块时,可基于包括在预测块中的每个预测样点的位置来更新预测样点。这样的更新方法可被称为基于样点位置的帧内加权预测方法(或位置依赖预测组合PDPC)。

[0410] 可通过考虑当前块的尺寸或形状、帧内预测模式、当前块的参考样点线、当前块的尺寸或颜色分量来确定PDPC是否将被使用。在示例中,当当前块的帧内预测模式为平面、DC、垂直方向、水平方向、索引值小于垂直方向的模式或索引值大于水平方向的模式中的至少一个时,可使用PDPC。可选地,仅当当前块的宽度或高度中的至少一个大于4时,可使用PDPC。可选地,仅当当前块的参考样点线的索引为0时,可使用PDPC。可选地,仅当当前块的参考样点线的索引等于或大于预定义值时,可使用PDPC。可选地,可仅针对亮度分量使用PDPC。可选地,可根据是否满足所列条件中的两个或更多个条件来确定是否使用PDPC。

[0411] 可选地,可根据子分区帧内编码方法是否被使用来确定是否使用PDPC。在示例中,当子分区帧内编码方法被应用于编码块或变换块时,可将PDPC设置为不被使用。可选地,当子分区帧内编码方法被应用于编码块或变换块时,可将PDPC应用于多个子块中的至少一个子块。在这种情况下,可基于编码块或子块的尺寸、形状、位置、帧内预测模式或参考样点线索引中的至少一个来确定作为PDPC的目标的子块。在示例中,PDPC可被应用于与编码块的上边界和/或左边界相邻的子块或者与编码块的下边界和/或右边界相邻的子块。可选地,基于子块的尺寸或形状,可设置为将PDPC应用于编码块中包括的所有子块,或者可设置为不将PDPC应用于编码块中包括的所有子块。在示例中,当子块的宽度或高度中的至少一个小于阈值时,可省略PDPC的应用。在另一示例中,PDPC可被应用于编码块中的所有子块。

[0412] 可选地,可根据通过对编码块或变换块进行分区而生成的子块的尺寸、形状、帧内预测模式或参考画面索引中的至少一个是否满足预设条件来按每个子块确定是否应用PDPC。在示例中,当子块的宽度或高度中的至少一个大于4时,可将PDPC应用于子块。

[0413] 在另一示例中,可以以比特流用信号发送表示PDPC是否被应用的信息。

[0414] 可选地,可基于以下项中的至少一项来确定应用PDPC的区域:当前块的尺寸、形状或帧内预测模式、或者预测样点的位置。在示例中,当当前块的帧内预测模式具有大于垂直方向的索引时,可不对x轴坐标或y轴坐标中的至少一个大于阈值的预测样点进行修改,并且可仅针对x轴坐标或y轴坐标等于或小于阈值的预测样点执行修改。可选地,当当前块的

帧内预测模式具有小于水平方向的索引时,可不对x轴坐标或y轴坐标中的至少一个大于阈值的预测样点进行修改,并且可仅针对x轴坐标或y轴坐标等于或小于阈值的预测样点执行修改。在这种情况下,可基于当前块的尺寸、形状或帧内预测模式中的至少一个来确定阈值。

[0415] 当通过帧内预测样点获得了预测样点时,可基于所获得的预测样点的位置确定用于修改预测样点的参考样点。

[0416] 可通过在针对当前块执行帧内预测之后对预测样点与当前块的原始样点进行求差来获得残差信号。在这种情况下,可在推导特定位置处的残差信号与邻近残差信号之间的差之后对推导出的差进行编码,而不是按原样对特定位置处的残差信号进行编码。在这种情况下,残差信号可表示残差样点、通过对残差样点进行变换而生成的变换系数或通过跳过变换而生成的系数。

[0417] 例如,可在对属于第一线的残差信号和属于第二线的残差信号进行求差之后执行针对差分残差值的编码,而不是按原样对属于第二线的残差信号进行编码。在这种情况下,对于第一线和第二线,x轴坐标或y轴坐标中的至少一个可以是不同的。

[0418] 在示例中,编码器可通过对残差样点进行变换来生成变换系数,并且对所生成的变换系数与通过对邻近变换系数进行求差而推导出的变换系数之间的差进行编码。解码器可将属于第一线的变换系数设置为针对第二残差信号的变换系数预测值,并通过将经过解码的差变换系数与变换系数预测值相加来推导第二变换系数。

[0419] 如上所述,当在对残差信号进行求差之后对残差差值进行编码/解码时,这可被称为帧内BDPCM(基于块的差量脉冲编码调制)。仅当当前块的预测编码模式被确定为帧内预测时,可使用帧内BDPCM。

[0420] 当帧内BDPCM被应用时,当前块的预测样点可被设置为0。换言之,当帧内BDPCM被应用时,残差样点可被设置为重建样点。

[0421] 可选地,当帧内BDPCM被应用时,可基于帧内预测推导当前块的预测样点。在这种情况下,可根据帧内BDPCM方向来确定当前块的帧内预测模式。在示例中,当帧内BDPCM方向为水平时,可基于水平方向上的帧内预测模式来获得预测样点。当帧内BDPCM方向为垂直时,可基于垂直方向上的帧内预测模式获得预测样点。

[0422] 可选地,可通过使用默认帧内预测模式来推导当前块的预测样点。默认帧内预测模式可以是DC、平面、水平或垂直中的任意一个。可在编码器和解码器中预定义默认帧内预测模式。指定默认帧内预测模式中的一个默认帧内预测模式的信息可被编码并被用信号发送。

[0423] 可选地,可从多个MPM候选中的一个MPM候选推导帧内预测模式。

[0424] 当帧内BDPCM被应用于当前块时,可强制使用相邻的参考样点线。换言之,可通过省略用信号发送指定参考样点线中的一条参考样点线的索引信息的操作并通过使用相邻的参考样点线来获得预测样点。

[0425] 当帧内BDPCM方法被应用时,可以以比特流用信号发送用于确定帧内BDPCM方向的信息。在示例中,可以以比特流用信号发送表示帧内BDPCM方向的标志intra_bdpcm_dir_flag。当语法intra_bdpcm_dir_flag为0时,这表示水平方向上的BDPCM被应用,并且当语法intra_bdpcm_dir_flag为1时,这表示垂直方向上的BDPCM被应用。

[0426] 可选地,可基于当前块的尺寸或形状来确定帧内BDPCM方向。在示例中,当当前块具有宽度大于高度的非正方形形状时,可确定水平方向上的BDPCM被应用。另一方面,当当前块具有高度大于宽度的非正方形形状时,可确定垂直方向上的BDPCM被应用。

[0427] 可选地,可通过考虑与当前块邻近的邻近块的帧内预测模式来确定帧内BDPCM方向。在示例中,当当前块的左侧块和上方块的至少一个帧内预测模式在水平方向上或在与水平方向相似的方向上时,可将水平方向上的BDPCM应用于当前块。在这种情况下,与水平方向相似的方向表示与水平方向上的帧内预测模式的差等于或小于阈值的帧内预测模式。可选地,当当前块的左侧块和上方块的至少一个帧内预测模式在垂直方向上或在与垂直方向相似的方向上时,可将垂直方向上的BDPCM应用于当前块。在这种情况下,与垂直方向相似的方向表示与垂直方向上的帧内预测模式的差等于或小于阈值的帧内预测模式。

[0428] 当水平方向上的BDPCM被应用时,可对残差信号与和该残差信号的上方邻近的残差信号之间的差值进行编码。解码器可通过将上方残差信号与经过解码的差值相加来推导残差信号。

[0429] 当垂直方向上的BDPCM被应用时,可对残差信号与和该残差信号的左侧邻近的残差信号之间的差值进行编码。解码器可通过将左侧残差信号与经过解码的差值相加来推导残差信号。

[0430] 可选地,可应用非方向BDPCM。在示例中,DC BDPCM表示对预定位置处的残差信号与预定位置处的邻近残差信号的平均值之间的差进行编码/解码。平面BDPCM表示对水平方向上的差值(预定位置处的残差信号与位于该预定位置处的残差信号的水平方向上的残差信号之间的差)和垂直方向上的差值(预定位置处的残差信号与位于该预定位置处的残差信号的垂直方向上的残差信号之间的差)的平均或加权求和结果进行编码/解码。

[0431] 可以以比特流用信号发送用于指定可用BDPCM模式的信息。该信息可以是表示非方向BDPCM是否被应用的信息或者用于指定可被应用于当前块的多个BDPCM候选中的任意一个BDPCM候选的信息。

[0432] 可以以比特流用信号发送表示帧内BDPCM是否将被应用于当前块的信息。在示例中,可以以比特流用信号发送标志intra_bdpcm_flag。当语法intra_bdpcm_flag为1时,这表示帧内BDPCM被应用于当前块。当语法intra_bdpcm_flag为0时,这表示帧内BDPCM未被应用于当前块。

[0433] 在画面级或序列级,可用信号发送表示帧内BDPCM的可用性的信息。在示例中,表示帧内BDPCM的可用性的标志sps_intra_bdpcm_flag可通过序列参数集(SPS)被用信号发送。当语法sps_intra_bdpcm_flag为1时,这表示参考序列参数集的画面可使用帧内BDPCM。当语法sps_intra_bdpcm_flag为0时,这表示参考序列参数集的画面可不使用帧内BDPCM。仅当sps_intra_bdpcm_flag为1时,可用信号发送表示帧内BDPCM是否被应用于当前块的intra_bdpcm_flag。

[0434] 当帧内BDPCM被应用于当前块时,PDPC可被设置为不被使用。

[0435] 当帧内BDPCM被应用于当前块时,可设置为强制应用变换跳过。换言之,虽然未用信号发送表示变换跳过是否被应用的transform_skip_flag,但是当帧内BDPCM被应用于当前块时,该值可被视为1。

[0436] 组合预测模式是通过两个或更多个预测模式进行组合来生成预测图像的方法。

在示例中,当组合预测模式被应用时,可通过对基于第一预测模式生成的第一预测块和基于第二预测模式生成的第二预测块求平均或者通过使用它们进行加权求和运算来生成预测块。预测模式可包括帧内预测模式、合并模式、AMVP模式、跳过模式、帧内块复制模式或调色板模式中的至少一个。在示例中,第一预测模式可以是合并模式,并且第二预测模式可以是帧内预测模式。

[0437] 当组合了合并模式和帧内预测模式的组合预测模式被使用时,可通过基于运动信息获得的第一预测块和基于预定帧内预测模式获得的第二预测块的加权预测来生成当前块的预测块。可从由merge_idx (以比特流被用信号发送的索引) 指定的合并候选推导当前块的运动信息。当前块的帧内预测模式可被设置为预定义帧内预测模式。预定义帧内预测模式可以是平面模式、DC模式、水平模式或垂直模式。可选地,邻近块的帧内预测模式可被设置为当前块的帧内预测模式。

[0438] 可以以比特流用信号发送表示组合预测模式是否被应用于当前块的标志。在示例中,可以以比特流用信号发送语法ciip_flag。当语法ciip_flag的值为1时,这表示组合预测模式被应用于当前块。当语法ciip_flag的值为0时,这表示组合预测模式未被应用于当前块。当组合预测模式未被应用于当前块时,可应用合并偏移编码方法或三角形分区方法中的至少一个。

[0439] 可通过从原始图像减去预测图像来推导残差图像。就此而言,当将残差图像转换到频域时,即使从频率分量去除了高频分量,图像的主观图像质量也不会显著下降。因此,当高频分量的值被变换为小值时,或者当高频分量的值被设置为0时,可在不导致大的视觉失真的情况下提高压缩效率。反映上述特征,可对当前块执行变换,以便将残差图像分解为二维频率分量。可通过使用诸如DCT (离散余弦变换)、DST (离散正弦变换) 等的变换方法来执行变换。

[0440] DCT用于通过使用余弦变换将残差图像分解(或变换)为二维频率分量,并且DST用于通过使用正弦变换将残差图像合成(或变换)为二维频率分量。作为对残差图像进行变换的结果,频率分量可被表示为基本图像。在示例中,当针对 $N \times N$ 尺寸的块执行DCT变换时,可获得 N^2 个基本模式分量。可通过变换获得 $N \times N$ 尺寸的块中包括的基本模式分量中的每个基本模式分量的尺寸。根据所使用的变换方法,基本模式分量的尺寸可被称为DCT系数或DST系数。

[0441] 变换方法DCT主要被用于对分布有许多非零低频分量的图像进行变换。变换方法DST主要被用于分布有许多高频分量的图像。

[0442] 还可通过使用除了DCT或DST之外的变换方法来对残差图像进行变换。

[0443] 在下文中,将残差图像变换为二维频率分量的操作被称为二维图像变换。此外,通过变换获得的基本模式分量的尺寸被称为变换系数。在示例中,变换系数可表示DCT系数或DST系数。当之后描述的第一变换和第二变换两者被应用时,变换系数可表示通过第二变换的结果生成的基本模式分量。此外,应用变换跳过的残差样点也被称为变换系数。

[0444] 可以以块为单位确定变换方法。可基于当前块的预测编码模式、当前块的尺寸或当前块的形状中的至少一个来确定变换方法。在示例中,当通过帧内预测模式对当前块进行编码并且当前块的尺寸小于 $N \times N$ 时,可通过使用DST变换方法来执行变换。另一方面,当不满足所述条件时,可通过使用DCT变换方法来执行变换。

[0445] 对于残差图像的一些块,可不执行二维图像变换。不执行二维图像变换可被称为变换跳过。变换跳过表示第一变换和第二变换不被应用于当前块。当变换跳过被应用时,可将量化应用于未执行变换的残差值。

[0446] 可基于当前块的尺寸或形状中的至少一个来确定针对当前块是否允许变换跳过。在示例中,仅当当前块的尺寸小于阈值时,可应用变换跳过。所述阈值与当前块的宽度、高度或样点的数量中的至少一个相关,并且可被定义为 32×32 等。可选地,可仅针对正方形块允许变换跳过。在示例中,针对 32×32 、 16×16 、 8×8 或 4×4 尺寸的正方形块,可允许变换跳过。可选地,仅当子分区帧内编码方法不被使用时,可允许变换跳过。

[0447] 可选地,当子分区帧内编码方法被应用于当前块时,可确定针对每个子分区是否应用变换跳过。

[0448] 图30是示出按子块确定是否执行变换跳过的示例的示意图。

[0449] 变换跳过可仅被应用于多个子块中的一部分子块。在示例中,如在图30中所示的示例中,可设置为将变换跳过应用于当前块的上方位置处的子块并且不将变换跳过应用于下方位置处的子块。

[0450] 可基于以比特流用信号发送的信息来确定不允许变换跳过的子块的变换类型。在示例中,可基于稍后将描述的`tu_mts_idx`来确定变换类型。

[0451] 可选地,可基于子块的尺寸来确定子块的变换类型。在示例中,可基于子块的宽度是否等于或大于以及/或者等于或小于阈值来确定水平方向变换类型,并且可基于子块的高度是否等于或大于以及/或者等于或小于阈值来确定垂直方向变换类型。

[0452] 可选地,可针对编码块用信号发送用于确定变换类型的信息,并且可将由该信息指定的变换类型共同应用于包括在编码块中的子块。换言之,可等地设置编码块中的子块的变换类型。

[0453] 在通过使用DCT或DST对当前块执行变换之后,可再次对经过变换的当前块执行变换。就此而言,基于DCT或DST的变换可被定义为第一变换,并且再次对应用了第一变换的块执行变换可被定义为第二变换。

[0454] 可通过使用多个变换核候选中的任意一个来执行第一变换。在示例中,可通过使用DCT2、DCT8或DST7中的任意一个来执行第一变换。

[0455] 不同变换核可被用于水平方向和垂直方向。可以以比特流用信号发送表示水平方向上的变换核和垂直方向上的变换核的组合的信息。在示例中,上述`tu_mts_idx`可指示水平方向上的变换核和垂直方向上的变换核的组合中的一个组合。

[0456] 第一变换的处理单元可不同于第二变换。在示例中,可对 8×8 的块执行第一变换,并且可对经过变换的 8×8 的块内的 4×4 尺寸的子块执行第二变换。可选地,可针对属于3个 4×4 尺寸的子块的变换系数执行第二变换。所述3个子块可包括位于当前块的左上方的子块、与该子块的右侧邻近的子块和与该子块的下方邻近的子块。可选地,可针对 8×8 尺寸的块执行第二变换。

[0457] 还可将不执行第二变换的其余区域中的变换系数设置为0。

[0458] 可选地,可对 4×4 的块执行第一变换,并且可对包括经过变换的 4×4 的块的尺寸为 8×8 的区域执行第二变换。

[0459] 可以以比特流用信号发送表示当前块的变换类型的信息。所述信息可以是表示针

对水平方向的变换类型和针对垂直方向的变换类型的组合中的一个组合的索引信息tu_mts_idx。

[0460] 基于由索引信息tu_mts_idx指定的变换类型候选,可确定针对垂直方向的变换核和针对水平方向的变换核。表7表示根据tu_mts_idx的变换类型组合。

[0461] 【表7】

tu_mts_idx	变换类型	
	水平	垂直
0	DCT-II	DCT-II
1	DST-VII	DST-VII
2	DCT-VIII	DST-VII
3	DST-VII	DCT-VIII
4	DCT-VIII	DCT-VIII

[0463] 变换类型可被确定为DCT2、DST7或DCT8中的一个。可选地,变换跳过可被插入在变换类型候选中。

[0464] 当使用表7时,当tu_mts_idx为0时可在水平方向上和在垂直方向上应用DCT2。当tu_mts_idx为2时,可在水平方向上应用DCT8,并且可在垂直方向上应用DCT7。

[0465] 当子分区帧内编码方法被应用时,可独立地确定子块的变换核。在示例中,用于指定变换类型组合候选的信息可按子块被编码并被用信号发送。因此,子块之间的变换核可不同。

[0466] 可选地,子块可使用相同的变换类型。在这种情况下,可仅针对第一子块用信号发送指定变换类型组合候选的tu_mts_idx。可选地,可在编码块级用信号发送tu_mts_idx,并且可通过参考在编码块级被用信号发送的tu_mts_idx来确定子块的变换类型。可选地,可基于子块中的一个子块的尺寸、形状或帧内预测模式中的至少一个来确定变换类型,并且可将所确定的变换类型设置为用于所有子块。

[0467] 图31是示出子块使用相同的变换类型的示例的示意图。

[0468] 当编码块在水平方向上被分区时,可将编码块的上方位置处的子块(Sub-CU0)的变换类型设置为与下方位置处的子块(Sub-CU1)的变换类型相同。在示例中,如在图31(a)中所示的示例中,当基于针对上方子块而被用信号发送的tu_mts_idx确定了水平变换类型和垂直变换类型时,所确定的变换类型也可被应用于下方子块。

[0469] 当编码块在垂直方向上被分区时,可将编码块的左侧位置处的子块(Sub-CU0)的变换类型设置为与右侧位置处的子块(Sub-CU1)的变换类型相同。在示例中,如在图31(b)中所示的示例中,当基于针对左侧子块而被用信号发送的tu_mts_idx确定了水平变换类型和垂直变换类型时,所确定的变换类型也可被应用于右侧子块。

[0470] 可基于以下项中的至少一项来确定索引信息是否被编码:当前块的尺寸或形状、非零系数的数量、第二变换是否被执行、或者子分区帧内编码方法是否被应用。在示例中,当子分区帧内编码方法被应用于当前块时,或者当非零系数的数量等于或小于阈值时,可省略用信号发送索引信息的操作。当省略用信号发送索引信息的操作时,可将默认变换类

型应用于当前块。

[0471] 默认变换类型可包括DCT2或DST7中的至少一个。当存在多个默认变换类型时,可通过考虑以下项中的至少一项来选择多个默认变换类型中的一个默认变换类型:当前块的尺寸、形状或帧内预测模式、第二变换是否被执行、或者子分区帧内编码方法是否被应用。在示例中,可基于当前块的宽度是否在预设范围中来将多个变换类型中的一个变换类型确定为水平方向变换类型,并且可基于当前块的高度是否在预设范围中来将多个变换类型中的一个变换类型确定为垂直方向变换类型。可选地,可根据当前块的尺寸、形状或帧内预测模式或者第二变换是否被执行来不同地确定默认模式。

[0472] 可选地,当在当前块中仅存在DC分量的变换系数时,水平方向变换类型和垂直方向变换类型可被设置为默认变换类型。在示例中,当在当前块中仅存在DC分量的变换系数时,水平方向变换类型和垂直方向变换类型可被设置为DCT2。

[0473] 可基于当前块的尺寸或形状确定阈值。在示例中,当当前块的尺寸等于或小于 32×32 时,可将阈值设置为2,并且当当前块大于 32×32 时(例如,当当前块为 32×64 或 64×32 尺寸的编码块时),可将阈值设置为4。

[0474] 多个查找表可被预先存储在编码装置/解码装置中。分配给变换类型组合候选的索引值、变换类型组合候选的类型或变换类型组合候选的数量中的至少一个对于多个查找表中的每个查找表可以是不同的。

[0475] 可基于以下项中的至少一项来选择针对当前块的查找表:当前块的尺寸、形状或帧内预测模式、第二变换是否被应用、或者变换跳过是否被应用于邻近块。

[0476] 在示例中,当当前块的尺寸等于或小于 4×4 时,或者当当前块通过帧间预测被编码时,可使用第一查找表,并且当当前块的尺寸大于 4×4 时,或者当当前块通过帧内预测被编码时,可使用第二查找表。

[0477] 可选地,可以以比特流用信号发送指示多个查找表中的一个查找表的信息。解码装置可基于所述信息选择针对当前块的查找表。

[0478] 在另一示例中,可基于以下项中的至少一项来自适应地确定分配给变换类型组合候选的索引:当前块的尺寸、形状、预测编码模式或帧内预测模式、第二变换是否被应用、或者变换跳过是否被应用于邻近块。在示例中,在当前块的尺寸为 4×4 时分配给变换跳过的索引可小于在当前块的尺寸大于 4×4 时分配给变换跳过的索引。具体地,当当前块的尺寸为 4×4 时,可将索引0分配给变换跳过,并且当当前块大于 4×4 且等于或小于 16×16 时,可将大于0的索引(例如,索引1)分配给变换跳过。当当前块大于 16×16 时,可将最大值(例如,5)分配给变换跳过的索引。

[0479] 可选地,当当前块通过帧间预测被编码时,可将索引0分配给变换跳过。当当前块通过帧内预测被编码时,可将大于0的索引(例如,索引1)分配给变换跳过。

[0480] 可选地,当当前块是通过帧间预测被编码的 4×4 尺寸的块时,可将索引0分配给变换跳过。另一方面,当当前块不是通过帧间预测被编码时,或者当当前块大于 4×4 时,可将大于0的索引(例如,索引1)分配给变换跳过。

[0481] 还可使用与表7中列出的变换类型组合候选不同的变换类型组合候选。在示例中,可使用由应用于水平方向变换或垂直方向变换中的一个的变换跳过以及应用于水平方向变换或垂直方向变换中的另一个的变换核(诸如DCT2、DCT8或DST7等)组成的变换类型组合

候选。在这种情况下,可基于当前块的尺寸(例如,宽度和/或高度)、形状、预测编码模式或帧内预测模式中的至少一个来确定变换跳过是否将被用作针对水平方向或垂直方向的变换类型候选。

[0482] 可以以比特流用信号发送表示用于确定当前块的变换类型的索引信息是否被显式地用信号发送的信息。在示例中,可在序列级用信号发送表示针对通过帧内预测被编码的块是否允许显式变换类型确定的信息sps_explicit_intra_mts_flag以及/或者表示针对通过帧间预测被编码的块是否允许显式变换类型确定的信息sps_explicit_inter_mts_flag。

[0483] 当允许显式变换类型确定时,可基于以比特流被用信号发送的索引信息tu_mts_idx来确定当前块的变换类型。另一方面,当不允许显式变换类型确定时,可基于以下项中的至少一项来确定变换类型:当前块的尺寸或形状、是否允许以子块为单位执行变换、包括非零变换系数的子块的位置、第二变换是否被执行、或者子分区帧内编码方法是否被应用。在示例中,可基于当前块的宽度确定当前块的水平方向变换类型,并且可基于当前块的高度确定当前块的垂直方向变换类型。例如,当当前块的宽度小于4或大于16时,可将水平方向变换类型确定为DCT2。否则,可将水平方向变换类型确定为DST7。当当前块的高度小于4或大于16时,可将垂直方向变换类型确定为DCT2。否则,可将垂直方向变换类型确定为DST7。在这种情况下,可基于当前块的尺寸、形状或帧内预测模式中的至少一个来确定将与宽度和高度进行比较的阈值,以确定水平方向变换类型和垂直方向变换类型。

[0484] 可选地,当当前块具有高度和宽度相同的正方形形状时,可将水平方向变换类型和垂直方向变换类型设置为相同,但是当当前块具有高度和宽度彼此不同的非正方形形状时,可不同地设置水平方向变换类型和垂直方向变换类型。在示例中,当当前块的宽度大于高度时,可将水平方向变换类型确定为DST7,并且可将垂直方向变换类型确定为DCT2。当当前块的高度大于宽度时,可将垂直方向变换类型确定为DST7,并且可将水平方向变换类型确定为DCT2。

[0485] 变换类型候选的数量和/或类型或者变换类型组合候选的数量和/或类型可根据显式变换类型确定是否被允许而不同。在示例中,当显式变换类型确定被允许时,DCT2、DST7和DCT8可被用作变换类型候选。因此,水平方向变换类型和垂直方向变换类型中的每一个可被设置为DCT2、DST7或DCT8。当显式变换类型确定不被允许时,仅DCT2和DST7可被用作变换类型候选。因此,可将水平方向变换类型和垂直方向变换类型中的每一个确定为DCT2或DST7。

[0486] 编码块或变换块可被分区为多个子块,并且可仅针对多个子块中的部分子块执行变换。仅将变换应用于多个子块中的部分子块的操作可被定义为子变换块编码方法。

[0487] 图32和图33是示出子变换块编码方法的应用方面的示意图。

[0488] 图32是示出仅针对4个子块中的一个子块执行变换的示例的示意图,并且图33是示出仅针对2个子块中的任意一个子块执行变换的示例的示意图。在图32和图33中,假设仅针对标记了“目标”的子块执行变换。

[0489] 如在图32中所示的示例中,在通过使用相互正交的垂直线和水平线将编码块分区为4个子块之后,可仅针对它们中的一个子块执行变换和量化。未执行变换的子块中的变换系数可被设置为0。

[0490] 可选地,如在图33中所示的示例中,在通过使用垂直线或水平线将编码块分区为2个子块之后,可仅针对它们中的一个子块执行变换和量化。未执行变换的子块中的变换系数可被设置为0。

[0491] 可以以比特流用信号发送表示子变换块编码方法是否被应用于编码块的信息。所述信息可以是1比特标志`cu_sbt_flag`。当所述标志为1时,这表示仅针对通过对编码块或变换块进行分区而生成的多个子块中的部分子块执行变换,并且当所述标志为0时,这表示在未将编码块或变换块分区为子块的情况下执行变换。

[0492] 可基于以下项中的至少一项来确定子变换块编码方法是否可被用于编码块:编码块的尺寸、形状或预测编码模式、或者组合预测模式是否被用于编码块。在示例中,当以下情况中的至少一个情况被满足时,子变换块编码方法可用于编码块:编码块的宽度或高度中的至少一个等于或大于阈值的情况、帧间预测被应用于编码块的情况、或者组合预测模式未被应用于编码块的情况。在这种情况下,阈值可以是诸如4、8或16等的自然数。

[0493] 可选地,当编码块的宽高比大于阈值时,可不允许应用于子变换块编码方法。

[0494] 当帧内预测被应用于编码块时或者当帧内块复制模式被应用时,子变换块编码方法可被确定为不可用。

[0495] 可选地,可基于子分区帧内编码方法是否被应用于编码块来确定子变换块编码方法是否可用于编码块。在示例中,当子分区帧内编码方法被应用时,子变换块编码方法可被确定为可用。

[0496] 当子变换块编码方法被确定可用于编码块时,可以以比特流用信号发送语法`cu_sbt_flag`。根据经过解析的`cu_sbt_flag`的值,可确定子变换块编码方法是否被应用。

[0497] 另一方面,当子变换块编码方法被确定为不可用于编码块时,可省略用信号发送语法`cu_sbt_flag`的操作。当省略用信号发送语法`cu_sbt_flag`的操作时,可确定不将子变换块编码方法应用于编码块。

[0498] 当子变换编码方法被应用于编码块时,可以以比特流用信号发送表示编码块的分区形状的信息。表示编码块的分区形状的信息可包括表示编码块是否被分区为包括1/4尺寸的子块的信息、表示编码块的分区方向的信息或表示子块的数量中的至少一个。

[0499] 在示例中,当语法`cu_sbt_flag`为1时,可用信号发送表示编码块是否被分区为包括1/4尺寸的子块的标志`cu_sbt_quadtree_flag`。

[0500] 当语法`cu_sbt_quadtree_flag`为1时,这表示编码块被分区为包括1/4尺寸的子块。在示例中,可通过使用1条垂直线将编码块分区为宽度为编码块的宽度的1/4的子块和宽度为编码块的宽度的3/4的子块,或者可通过使用1条水平线将编码块分区为高度为编码块的高度的1/4的子块和高度为编码块的高度的3/4的子块。可选地,编码块可被分区为包括宽度和高度皆为编码块的宽度和高度的1/2的子块。当编码块被分区为包括1/4尺寸的子块时,这可被称为四型分区。当语法`cu_sbt_quad_tree_flag`为1时,编码块的1/4尺寸的子块可被设置为变换目标。

[0501] 当语法`cu_sbt_quadtree_flag`为0时,这表示编码块被分区为包括1/2尺寸的子块。在示例中,可通过使用1条垂直线或1条水平线将编码块分区为2个1/2尺寸的子块。将编码块分区为2个1/2尺寸的子块的操作可被称为二型分区。当语法`cu_sbt_quad_tree_flag`的值为0时,编码块的1/2尺寸的子块可被包括在该编码块中。

[0502] 此外,可以以比特流用信号发送表示编码块的分区方向的标志。在示例中,表示水平方向分区是否被应用于编码块的标志`cu_sbt_horizontal_flag`可被编码并被用信号发送。当`cu_sbt_horizontal_flag`的值为1时,这表示使用与编码块的上侧和下侧平行的至少一条分区线的水平方向分区被应用。当`cu_sbt_horizontal_flag`的值为0时,这表示使用与编码块的左侧和右侧平行的至少一条分区线的垂直方向分区被应用。

[0503] 根据编码块的尺寸或形状,可确定编码块的分区形状。在示例中,当编码块的宽度或高度中的至少一个等于或大于第一阈值时,四型分区可以是可用的。在示例中,第一阈值可以是诸如4、8或16的自然数。第一阈值可被称为四型阈值。

[0504] 当四型分区被确定为可用时,可以以比特流用信号发送语法`cu_sbt_quadtrees_flag`。根据经过解析的`cu_sbt_quadtrees_flag`的值,可确定四型分区是否被应用于编码块。

[0505] 当四型分区被确定为不可用时,可省略用信号发送语法`cu_sbt_quadtrees_flag`的操作。当省略用信号发送语法`cu_sbt_quadtrees_flag`的操作时,可确定将二型分区应用于编码块。

[0506] 表8示出用于确定是否对语法`cu_sbt_quadtrees_flag`进行解析的语法结构。

[0507] **【表8】**

[0508]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ && allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	

	}	
	transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
[0509]	}	
	}	
	}	

[0510] 在表8中,变量allowSbtVerQ表示是否允许垂直方向上的四型分区,并且变量allowSbtHorQ表示是否允许水平方向上的四型分区。可基于四型阈值来确定变量allowSbtVerQ和allowSbtHorQ。在示例中,当四型阈值为16时,可基于编码块的宽度是否等于或大于16来确定allowSbtVerQ,并且可基于编码块的高度是否等于或大于16来确定allowSbtHorQ。

[0511] 如在表8中所示的示例中,当变量allowSbtVerQ和allowSbtHorQ都为真时,可从比特流对语法cu_sbt_quad_flag进行解析。在示例中,当编码块是 16×8 时,变量allowSbtHorQ被设置为假,所以可省略对语法cu_sbt_quad_flag的解析。可选地,当编码块是 8×16 时,变量allowSbtVerQ被设置为假,所以可省略对语法cu_sbt_quad_flag的解析。当省略对语法cu_sbt_quad_flag的解析时,二型分区可被应用于编码块。

[0512] 可选地,与表8中所示的示例不同,当变量allowSbtVerQ或变量allowSbtHorQ中的任意一个为真时,可对语法cu_sbt_quad_flag进行解析。换言之,当编码块的宽度和高度中的仅任意一个等于或大于四型阈值时,四型分区可以是可用的。

[0513] 可选地,虽然编码块的宽度或高度中的任意一个等于或大于第一阈值,但是当编码块的宽度或高度中的另一个等于或小于第二阈值时,编码块的四型分区可被确定为不可用。在这种情况下,第二阈值可具有小于第一阈值的值。在示例中,第二阈值可以是诸如2、4或8的自然数。

[0514] 变量allowSbtHorH表示水平方向上的二型分区是否可用。当编码块的高度等于或大于阈值时,水平方向上的二型分区可被设置为可用。变量allowSbtVerH表示垂直方向上的二型分区是否可用。当编码块的宽度等于或大于阈值时,垂直方向上的二型分区可被设置为可用。在这种情况下,阈值可以是诸如4、8或16的自然数。

[0515] 当水平方向上的四型分区/二型分区和垂直方向上的四型分区/二型分区都可用时,可以以比特流用信号发送语法cu_sbt_horizontal_flag。根据语法cu_sbt_horizontal_flag的值,可将水平方向上的分区或垂直方向上的分区应用于编码块。

[0516] 另一方面,当水平方向上的四型分区/二型分区和垂直方向上的四型分区/二型分区中的仅一个可用时,可省略用信号发送语法cu_sbt_horizontal_flag的操作。当省略用信号发送语法cu_sbt_horizontal_flag的操作时,水平方向上的四型分区/二型分区和垂直方向上的四型分区/二型分区中的可用的一个可被应用。

[0517] 当水平方向上的二型分区或垂直方向上的二型分区不可用时,可省略用信号发送语法cu_sbt_horizontal_flag的操作。

[0518] 当省略用信号发送语法cu_sbt_horizontal_flag的操作并且变量allowSbtHorH为真时,水平方向上的二型分区可被应用于编码块。

[0519] 当省略用信号发送语法`cu_sbt_horizontal_flag`的操作并且变量`allowSbtVerH`为真时,垂直方向上的二型分区可被应用于编码块。

[0520] 可以以比特流用信号发送用于指定多个子块中的作为变换的目标的子块的信息。在示例中,可以以比特流用信号发送语法`cu_sbt_pos_flag`。语法`cu_sbt_pos_flag`表示变换目标是否是编码块中的第一子块。在示例中,当水平方向上的四型分区/二型分区被应用于编码块时,当`cu_sbt_flag`为1时,最左侧子块被确定为变换目标,并且当`cu_sbt_flag`为0时,最右侧子块被确定为变换目标。当垂直方向上的四型分区/二型分区被应用于编码块时,当`cu_sbt_pos_flag`为1时,最上方子块被确定为变换目标,并且当`cu_sbt_pos_flag`为0时,最下方子块被确定为变换目标。

[0521] 可通过考虑编码块的分区方向和子块的位置来确定子块的变换类型。在示例中,当编码块在垂直方向上被分区并且针对子块中的左侧位置处的子块执行变换时,可不同地设置水平方向变换类型和垂直方向变换类型。

[0522] 图34和图35示出根据作为变换的目标的子块的位置的水平方向变换类型和垂直方向变换类型。

[0523] 在图34中所示的示例中,当作为变换的目标的子块包括编码块的左上样点或右下样点时,水平方向变换类型和垂直方向变换类型可被设置为相同。在示例中,图34中所示的示例示出了当作为变换的目标的子块包括编码块的左上样点时,水平方向变换类型和垂直方向变换类型被设置为DCT8,并且当作为变换的目标的子块包括编码块的右下样点时,水平方向变换类型和垂直方向变换类型被设置为DST7。

[0524] 当作为变换的目标的子块包括编码块的右上样点或左下样点时,水平方向变换类型和垂直方向变换类型可被不同地设置。在示例中,图34中所示的示例示出了当作为变换的目标的子块包括编码块的右上样点时,水平方向变换类型被设置为DST7,并且垂直方向变换类型被设置为DCT8。当作为变换的目标的子块包括编码块的左下样点时,水平方向变换类型被设置为DCT8,并且垂直方向变换类型被设置为DST7。

[0525] 与图34中所示的示例不同,当编码块中的包括左上样点的子块或包括右下样点的子块被确定为变换目标时,水平方向变换类型和垂直方向变换类型可被不同地设置,并且当编码块中的包括右上样点的子块或包括左下样点的子块被确定为变换目标时,水平方向变换类型和垂直方向变换类型可被设置为相同。

[0526] 图34示出了高度和宽度分别是编码块的1/2的子块被设置为变换目标。与示出的示例不同,宽度与编码块相同但高度为编码块的1/4的子块或者高度与编码块相同但宽度为编码块的1/4的子块可被设置为变换目标。

[0527] 在图35中所示的示例中,当作为变换的目标的子块包括编码块的左上样点时,水平方向变换类型和垂直方向变换类型可被不同地设置。在示例中,在图35中所示的示例中,当水平方向上的二型分区被应用并且上方子块被确定为变换目标时,水平方向变换类型可被设置为DST7,并且垂直方向变换类型可被设置为DCT7。当垂直方向上的二型分区被应用并且左侧子块被确定为变换目标时,水平方向变换类型可被设置为DCT8,并且垂直方向变换类型可被设置为DST7。

[0528] 与图35中所示的示例不同,当作为变换的目标的子块包括编码块的左上样点时,水平方向变换类型和垂直方向变换类型可被设置为相同,并且当作为变换的目标的子块包

括编码块的右下样点时,水平方向变换类型和垂直方向变换类型可被不同地设置。

[0529] 当作为变换的目标的子块包括编码块的右下样点时,水平方向变换类型和垂直方向变换类型可被设置为相同。在示例中,在图35中所示的示例中,当水平方向上的二型分区被应用并且下方子块被确定为变换目标时,水平方向变换类型和垂直方向变换类型可被设置为DST7。当垂直方向上的二型分区被应用并且右侧子块被确定为变换目标时,水平方向变换类型和垂直方向变换类型可被设置为DST7。

[0530] 如在上述示例中,可根据编码块中的作为变换的目标的子块的位置来确定水平方向变换类型和垂直方向变换类型是否被设置为相同。此外,可根据编码块中的作为变换的目标的子块的位置来确定水平方向变换类型和垂直方向变换类型。

[0531] 对于子块,可省略对表示是否存在非零系数的信息(例如,CBF)的编码。当省略对CBF的编码时,可基于执行了变换的块的位置来确定非零残差系数是否被包括在每个子块中。在示例中,当应用了二型分区的编码块中的右侧或下方位置处的子块被确定为变换目标时,针对左侧或上方位置处的子块的CBF值可被推断为0,并且针对右侧或下方位置处的子块的CBF值可被推断为1。可选地,当应用了二型分区的编码块中的左侧或下方位置处的子块被确定为变换目标时,针对左侧或上方位置处的子块的CBF值可被推断为1,并且针对右侧或下方位置处的子块的CBF值可被推断为0。

[0532] 可针对执行了第一变换的块执行第二变换。可针对已应用了第一变换的变换块中的左上方区域执行第二变换。

[0533] 如果已执行了第一变换和第二变换的残差系数被编码,则解码装置可针对变换块执行作为第二变换的逆处理的第二逆变换,并且针对已执行了第二逆变换的变换块执行作为第一变换的逆处理的第一逆变换。

[0534] 可基于以下项中的至少一项来确定第二变换是否被应用于当前块:当前块的尺寸、残差系数的数量、编码模式或帧内预测模式、或者子分区帧内编码方法是否被应用。编码器可对表示第二变换是否被应用的信息进行编码并将所述信息用信号发送到解码器。可选地,编码器和解码器可基于相同的条件确定第二变换是否被执行。

[0535] 在示例中,可以以比特流用信号发送表示第二变换是否被执行的信息。具体地,可用信号发送表示第二变换是否被执行的标志或指定第二变换是否被执行的索引信息以及用于第二变换的变换核。

[0536] 表9表示以比特流用信号发送表示第二变换是否被执行的标志1fnst_flag的示例。当标志1fnst_flag的值为0时,这表示针对当前块未执行第二变换。另一方面,当标志1fnst_flag的值为1时,这表示针对当前块执行了第二变换。

[0537] 【表9】

[0538]	<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</code>	描述符
--------	---	-----

[0539]

...	
numSigCoeff = 0	
numZeroOutSigCoeff = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : cbWidth	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : cbHeight	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && !intra_mip_flag[x0][y0] && !intra_bdpcm_flag) {	
if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && numZeroOutSigCoeff == 0)	
lfnst_flag[x0][y0]	u(1)
}	
}	
}	
}	

[0540] 可选地,可以以比特流用信号发送语法lfnst_idx。当索引lfnst_idx的值为0时,这表示针对当前块未执行第二变换。另一方面,当索引lfnst_idx大于0时,这表示针对当前块执行了第二变换。当索引lfnst_idx的值大于0时,lfnst_idx可被用于指定用于执行第二变换的变换核。

[0541] 可选地,可通过将当前块的宽度或高度中的至少一个与阈值进行比较来确定是否执行第二变换。在示例中,当当前块的宽度和高度之中的最小值小于阈值时,可不执行第二变换。在这种情况下,阈值可以是自然数,诸如4、8或16。

[0542] 可选地,当通过帧间预测对当前块进行编码时,可不应用第二变换。

[0543] 可选地,虽然通过帧内预测对当前块进行编码,但是当执行基于矩阵的帧内预测

时,可不应用第二变换。

[0544] 可选地,可基于水平方向上的变换核是否与垂直方向上的变换核相同来确定第二变换是否被执行。在示例中,仅当水平方向上的变换核与垂直方向上的变换核相同时,可执行第二变换。可选地,仅当水平方向上的变换核不同于垂直方向上的变换核时,可执行第二变换。

[0545] 可选地,仅当预定义变换核被用于水平方向变换和垂直方向变换时,可允许第二变换。在示例中,当DCT2变换核被用于水平方向上的变换和垂直方向上的变换时,可允许第二变换。

[0546] 可选地,在子分区帧内编码方法被应用于当前块时,仅当DCT2变换核被用于水平方向上的变换和垂直方向上的变换时,允许第二变换。

[0547] 可选地,可基于当前块的非零变换系数的数量确定第二变换是否被执行。在示例中,当当前块的非零变换系数的数量小于或等于阈值时,可设置为不使用第二变换,并且当当前块的非零变换系数的数量大于阈值时,可设置为使用第二变换。仅当通过帧内预测对当前块进行编码时,可允许使用第二变换。

[0548] 可选地,可基于当前块的最后非零变换系数的位置确定第二变换是否被执行。在示例中,当当前块的最后非零变换系数的x轴坐标或y轴坐标中的至少一个大于阈值时,或者当当前块的最后非零变换系数所属的子块的x轴坐标或y轴坐标中的至少一个大于阈值时,可不执行第二变换。在这种情况下,可在编码器和解码器中预定义阈值。可选地,可基于当前块的尺寸或形状确定阈值。

[0549] 可选地,当在当前块中仅存在DC分量的变换系数时,可设置为不执行第二变换。在这种情况下,DC分量表示当前块中左上方位置处的变换系数。

[0550] 可基于组合预测编码模式是否被应用于当前块来确定是否执行第二变换。在示例中,当组合预测编码模式被应用于当前块时,可设置为不执行第二变换。

[0551] 可选地,当组合预测编码模式被应用时,可基于当前块的尺寸或形状、帧内预测模式或者权重中的至少一个来确定是否执行第二变换。

[0552] 可选地,当组合预测编码模式被应用时,可设置为用信号发送表示针对当前块是否执行了第二变换的信息。例如,当组合预测编码模式被应用于当前块时,可用信号发送索引`lfnst_idx`。

[0553] 可选地,可确定是否执行第二变换,而不管是否应用了组合预测编码模式。

[0554] 可选地,可基于子分区帧内编码方法是否被应用于当前块来确定是否允许第二变换。在示例中,当子分区帧内编码方法被应用于当前块时,可设置为不将第二变换应用于当前块。

[0555] 可选地,当子分区帧内编码方法被应用于当前块时,可设置为应用第二变换。例如,当垂直方向分区或水平方向分区被应用于当前块时,可用信号发送表示第二变换是否被应用于当前块的索引信息。可通过该索引信息确定第二变换是否被应用和/或用于第二变换的变换核。另一方面,当子分区帧内编码方法未被应用于当前块时,可省略对索引信息的编码。可选地,当子分区帧内编码方法未被应用于当前块时,可基于预设条件是否被满足来确定是否对表示第二变换被是否应用的索引信息进行编码。在这种情况下,所述预设条件可与非零系数的位置、非零系数的数量或当前块的尺寸中的至少一个相关。在示例中,在

非零系数的数量等于或大于1时,当非零系数的数量为1但未被包括在当前块的 4×4 的左上方区域中时,或者当非零系数(非零系数未被包括在当前块的 4×4 的区域中)的扫描顺序等于或小于阈值时,可对索引信息进行编码并用信号发送所述索引信息。否则,可省略对索引信息的编码。

[0556] 当索引信息未被用信号发送时,索引信息的值可被推断为指示第二变换未被应用。

[0557] 可选地,可基于子块的尺寸、宽度、高度或形状中的至少一个来确定是否应用第二变换。在示例中,当子块的尺寸、宽度或高度中的至少一个小于阈值时,可设置为不应用第二变换。具体地,当子块的尺寸、宽度或高度中的至少一个小于阈值时,可省略对索引信息的编码。可在编码块级用信号发送表示第二变换是否被执行的信息。可基于在编码块级用信号发送的信息来确定第二变换是否将被应用于属于编码块的子块。

[0558] 阈值可在编码器和解码器中具有预定义值。在示例中,可将阈值设置为2、4或8等。可选地,可基于当前块的分区方向或包括在当前块中的子块的数量中的至少一个来确定阈值。

[0559] 可选地,可通过使用上面列举的与当前块相关的用于确定第二变换是否被应用的参数来确定是否对表示第二变换是否被应用的语法进行编码。

[0560] 在示例中,当当前块的尺寸、残差系数的数量、编码模式或帧内预测模式、或者子分区帧内编码方法是否被应用中的至少一个不满足预设条件时,可省略对表示第二变换是否被应用的语法的编码。在示例中,可省略对`lfnst_flag`或`lfnst_idx`的编码,并且其值可被推断为0。换言之,当省略对语法的编码时,可不应用第二变换。

[0561] 当帧内BDPCM被应用于当前块时,可不执行第二变换。当帧内BDPCM被应用于当前块时,可省略用信号发送表示第二变换是否被执行的语法元素的操作,并且其值可被推导为0。表10表示针对应用了帧内BDPCM的块省略用信号发送`lfnst_idx`的操作的示例。

[0562] 【表10】

	<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</code>	描述符
	<code>...</code>	
	<code>numSigCoeff = 0</code>	
	<code>numZeroOutSigCoeff = 0</code>	
	<code>transform_tree(x0, y0, cbWidth, cbHeight, treeType)</code>	
	<code>lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : cbWidth</code>	
[0563]	<code>lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : cbHeight</code>	
	<code>if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT &&</code>	
	<code>!intra_mip_flag[x0][y0] && !intra_bdpcm_flag) {</code>	
	<code>if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && numZeroOutSigCoeff == 0)</code>	
[0564]	<code>lfnst_idx[x0][y0]</code>	ae(v)
	<code>}</code>	
	<code>}</code>	
	<code>}</code>	
	<code>}</code>	

[0565] 可选地,当帧内BDPCM被应用于当前块时,可省略用信号发送表示变换跳过是否被执行的语法元素(例如,transform_skip_flag)的操作。当帧内BDPCM被应用于当前块时,可省略用信号发送标志transform_skip_flag的操作,并且其值可被推断为1。换言之,当帧内BDPCM被应用时,可不将变换应用于当前块。

[0566] 当确定变换未被应用于当前块时,可不应用第一变换和第二变换两者。因此,当当前块的变换被跳过时,可省略用信号发送表示第二变换是否被执行的语法元素的操作,并且其值可被推断为0。最后,当帧内BDPCM被应用于当前块时,可省略用信号发送表示第二变

换是否被执行的语法元素`lfnst_idx`的操作。

[0567] 基于上述描述,将详细描述在编码器和解码器中执行第二变换的方法。

[0568] 可对当前块中的左上方区域执行第二变换。应用了第二变换的目标区域可具有预定义尺寸或预定义形状。应用了第二变换的目标区域可具有诸如 4×4 或 8×8 的正方形块形状或者诸如 4×8 或 8×4 的非正方形块形状。

[0569] 可选地,当当前块被均匀地分区为 N 个区域时, N 个区域中的至少一个区域可被设置为目标区域。在这种情况下, N 可以是诸如2、4、8或16的自然数。可在编码器和解码器中预定义变量 N 。可选地,可基于当前块的尺寸和/或形状来确定变量 N 。

[0570] 可选地,可基于变换系数的数量来确定目标区域。在示例中,可根据预定扫描顺序将预定数量的变换系数确定为目标区域。

[0571] 可选地,用于指定目标区域的尺寸和/或形状的信息可被编码并以比特流被发送。所述信息可包括表示目标区域的尺寸的信息或表示目标区域所包括的 4×4 的块的数量信息中的至少一个。

[0572] 可选地,可将整个当前块设置为目标区域。在示例中,当当前块的尺寸与目标区域的最小尺寸(例如, 4×4)相同时,整个当前块可被设置为用于第二变换的目标。

[0573] 第二变换可以以不可分离形状被应用。因此,第二变换也可被称为不可分离第二变换(NSST)。

[0574] 应用第二变换的区域中的变换系数可被排列在单个列中。在示例中,当针对 $N \times N$ 尺寸的目标区域执行第二变换时,可将包括在目标区域中的变换系数转换为 $N^2 \times 1$ 尺寸的输入矩阵。当 4×4 尺寸的块被设置为目标区域时,包括在目标区域中的变换系数可被转换为 16×1 尺寸的输入矩阵。当 8×8 尺寸的块被设置为目标区域时,包括在目标区域中的变换系数可被转换为 64×1 尺寸的输入矩阵。

[0575] 不可分离变换矩阵可被应用于通过将包括在目标区域中的变换系数排列为一号线而生成的输入矩阵。不可分离变换矩阵的尺寸可根据输入矩阵的尺寸而不同地确定。

[0576] 在示例中,当输入矩阵的尺寸为 $N^2 \times 1$ 时,可基于 $N^2 \times N^2$ 尺寸的不可分离变换矩阵执行第二变换。例如,当输入矩阵的尺寸为 16×1 时,可使用 16×16 尺寸的不可分离变换矩阵,并且当输入矩阵的尺寸为 64×1 时,可使用 64×64 尺寸的不可分离变换矩阵。

[0577] 多个不可分离变换矩阵可被存储在编码器和解码器中。可以以比特流用信号发送用于指定多个不可分离变换矩阵中的任意一个不可分离变换矩阵的信息。

[0578] 可选地,可基于在第一变换中使用的当前块的尺寸、形状、量化参数、帧内预测模式或变换类型中的至少一个来指定不可分离变换矩阵。

[0579] 可选地,可基于在第一变换中使用的当前块的尺寸、形状、量化参数、帧内预测模式或变换类型中的至少一个来指定可由当前块使用的不可分离变换矩阵候选。当存在多个可由当前块使用的不可分离变换矩阵候选时,指示所述多个不可分离变换矩阵候选中的一个不可分离变换矩阵候选的信息可被编码并被用信号发送。

[0580] 可通过将不可分离变换矩阵与输入矩阵相乘来获得变换矩阵。在示例中,等式6示出获得变换矩阵 A' 的示例。

[0581] 【等式6】

[0582] $A' = T * A$

[0583] 在等式6中, T 表示不可分离变换矩阵, 并且 A 表示输入矩阵。当矩阵 T 的尺寸是 $N^2 \times N^2$ 并且矩阵 A 的尺寸是 $N^2 \times 1$ 时, 可获得 $N^2 \times 1$ 尺寸的变换矩阵 A' 。在示例中, 当使用 16×1 尺寸的输入矩阵和 16×16 尺寸的不可分离变换矩阵时, 可获得 16×1 尺寸的变换矩阵 A' 。可选地, 当使用 64×1 尺寸的输入矩阵和 64×64 尺寸的不可分离变换矩阵时, 可获得 64×1 尺寸的变换矩阵 A' 。

[0584] 当获得了变换矩阵 A' 时, 变换矩阵 A' 中的分量可被设置为当前块中的 $N \times N$ 尺寸的块的变换系数。不包括所述 $N \times N$ 尺寸的块的其余区域中的变换系数可被设置为默认值。在示例中, 未执行第二变换的区域中的变换系数可被设置为0。

[0585] 可通过使用行的数量小于列的数量的不可分离变换矩阵来执行第二变换。在示例中, $(k \times N^2)$ 尺寸的不可分离变换矩阵可被应用于 $(N^2 \times 1)$ 尺寸的输入矩阵。在这种情况下, k 可具有小于 N^2 的值。在示例中, k 可以是 $N^2/2$ 、 $N^2/4$ 或 $3N^2/4$ 等。 k 可被称为缩减因子。

[0586] 结果, 可获得小于输入矩阵的 $(k \times 1)$ 尺寸的变换矩阵。这样, 输出尺寸小于输入矩阵的变换矩阵的第二变换可被称为缩减第二变换。

[0587] 等式7表示应用缩减第二变换的示例。

[0588] 【等式7】

[0589] $A_R = R * A$

[0590] 在等式7中, R 表示 $k \times N^2$ 尺寸的不可分离变换矩阵。行的数量 k 小于列的数量 N^2 的不可分离变换矩阵可被称为缩减不可分离变换矩阵。 A_R 表示 $k \times 1$ 尺寸的变换矩阵。尺寸小于输入矩阵 A 的变换矩阵 A_R 可被称为缩减变换矩阵。

[0591] 当获得了缩减变换矩阵 A_R 时, 可将缩减变换矩阵 A_R 中的分量设置为当前块中的至少一个或更多个 $M \times M$ 尺寸的块的变换系数。在这种情况下, M 可以是小于 N 的自然数。可根据缩减因子 k 来确定 $M \times M$ 尺寸的块的数量。可将不包括至少一个 $M \times M$ 尺寸的块的其余区域的变换系数设置为默认值。在示例中, 所述其余区域中的变换系数可被设置为0。

[0592] 图36是示出当缩减因子为16时变换系数的编码方面的示图。

[0593] 可将包括在 8×8 尺寸的目标区域中的变换系数变换为 64×1 尺寸的输入矩阵, 并且可通过使用 16×64 尺寸的不可分离变换矩阵来获得 16×1 尺寸的变换矩阵。

[0594] 16×1 尺寸的变换矩阵可被设置为 4×4 的块的变换系数, 并且其他区域中的变换系数可被设置为0。

[0595] 虽然未示出, 但是当缩减因子 k 为32时, 32×1 尺寸的变换矩阵可被设置为 8×4 的块或 4×8 的块的变换系数, 并且其他区域中的变换系数可被设置为0。

[0596] 当缩减因子 k 为48时, 可将 48×1 尺寸的变换矩阵设置为三个 4×4 的块的变换系数, 并且可将其它区域中的变换系数设置为0。具体地, 可将变换矩阵设置为当前块的左上方位置处的 4×4 的块、与所述左上方块的右侧相邻的 4×4 的块和与所述左上方块的下方相邻的 4×4 的块的变换系数。

[0597] 可基于以下项中的至少一项来确定变换矩阵: 当前块的尺寸或形状、或者帧内预测模式。在示例中, 基于当前块的帧内预测模式, 可确定变换矩阵集, 并且可选择包括在变换矩阵集中的多个变换矩阵候选中的一个变换矩阵候选。可对指定多个变换矩阵候选中的应用于当前块的变换矩阵的索引信息进行编码并用信号发送所述索引信息。

[0598] 当不包括由第二变换生成的变换系数的其余变换系数被设置为0时, 解码器可基

于最后非零残差系数的位置确定是否执行第二变换。在示例中,当最后残差系数位于存储由第二变换生成的变换系数的块之外时,可确定不执行第二变换。换言之,仅当最后残差系数位于存储由第二变换生成的变换系数的块中时,解码器可执行针对第二变换的逆变换。

[0599] 可基于当前块的尺寸或形状中的至少一个来确定是否执行缩减第二变换。在示例中,当当前块的宽度或高度中的至少一个大于阈值时,可应用缩减第二变换,否则,可应用一般第二变换。在这种情况下,阈值可以是诸如4、8或16的自然数。

[0600] 可选地,可根据目标区域的尺寸来确定是否执行缩减第二变换。在示例中,当针对 4×4 尺寸的目标区域执行第二变换时,可应用一般第二变换。在示例中,对于 4×4 尺寸的目标区域,可通过使用 16×16 尺寸的不可分离变换矩阵来执行第二变换。

[0601] 另一方面,当针对 8×8 尺寸的目标区域执行第二变换时,可应用缩减第二变换。在示例中,对于 8×8 尺寸的目标区域,可通过使用 48×64 、 32×64 或 16×64 尺寸的不可分离变换矩阵来执行第二变换。

[0602] 当执行针对缩减第二变换的逆变换时,输出矩阵的尺寸具有大于输入矩阵的尺寸的值。在示例中,当缩减因子 k 是16时,可通过针对 16×1 尺寸的输入矩阵执行逆变换来获得 64×1 尺寸的输出矩阵。

[0603] 当子分区帧内编码方法被应用于编码块时,可将编码块分区为多个子块。当子分区帧内编码方法被应用于编码块时,可设置为不应用第二变换。

[0604] 可选地,可基于子分区的形状或尺寸来确定是否执行第二变换。在示例中,当编码块被分区为宽度或高度为4的子分区时,可应用第二变换。换言之,仅当子分区具有 $4 \times L$ 或 $L \times 4$ 的形状时,可应用第二变换。在这种情况下, L 表示等于或大于4的整数。

[0605] 可选地,仅当子分区的宽度和高度的最小值等于或大于预定义阈值时,可应用第二变换。在这种情况下,阈值可以是诸如4、8或16的整数。

[0606] 如在以上示例中,当子分区的形状或尺寸满足预定义条件时,可以以比特流用信号发送表示第二变换是否被应用的信息。在示例中,当子分区为 $4 \times L$ 或 $L \times 4$ 时,或者当子分区的宽度和高度之中的最小值等于或大于阈值时,可用信号发送指示第二变换是否被执行的语法`lfnst_idx`。

[0607] 另一方面,当子分区的形状或尺寸不满足预定义条件时,可省略对表示第二变换是否被应用的信息的编码/解码。在示例中,当子分区的宽度和高度之中的最小值小于阈值时,可省略对语法`lfnst_idx`的编码/解码。当省略对语法`lfnst_idx`的编码/解码时,其值可被推导为0。

[0608] 当确定第二变换被应用时,可将第二变换应用于编码块中的预定义位置处的子块或分区索引小于阈值的子块。在这种情况下,左侧子块的分区索引可被设置为具有小于右侧子块的分区索引的值,或者上方子块的分区索引可被设置为具有小于下方子块的分区索引的值。在示例中,第二变换可仅被应用于编码块中的第一子块。

[0609] 可选地,可将第二变换应用于所有子块中的每个子块。

[0610] 可选地,当确定在编码块级执行第二变换时,可基于每个子块的属性自适应地确定是否将第二变换应用于每个子块。在这种情况下,子块的属性可包括以下项中的至少一项:子块所包括的残差系数的数量、变换跳过是否被应用于子块、或者变换核是否被应用于子块。

[0611] 可基于子块的尺寸确定用于第二变换的目标区域的尺寸。在示例中,可将包括16个样点的 4×4 尺寸的块、包括32个样点的两个 4×4 尺寸的块、包括48个样点的三个 4×4 尺寸的块或者包括64个样点的四个 4×4 尺寸的块设置为用于第二变换的目标区域。根据目标区域的尺寸,可应用一般第二变换,或者可应用缩减第二变换。

[0612] 可基于子块的尺寸确定用于第二变换的目标区域的尺寸。在示例中,当子块的宽度或高度中的至少一个小于阈值时,可将包括N个样点的区域设置为用于第二变换的目标区域。另一方面,当子块的宽度和高度等于或大于阈值时,可将包括M个样点的区域设置为用于第二变换的目标区域。在这种情况下,M可以是大于N的自然数。在示例中,N可以是16或32,并且M可以是48或64。此外,阈值可以是诸如2、4、8或16等的自然数。

[0613] 可将用于第二变换的目标区域设置为不延伸到子块的边界之外。换言之,当用于第二变换的目标区域跨越两个或更多个子块时,可设置为不执行第二变换。

[0614] 图37和图38是示出用于第二变换的目标区域的示意图。

[0615] 当水平方向上的分区被应用于 16×16 尺寸的编码块时,可将编码块分区为 16×4 尺寸的子块。如在所示示例中,当子块具有 $N \times 4$ 尺寸(N是大于4的整数)时,可将用于第二变换的目标区域的高度设置为不超过4。

[0616] 换言之,如在图37(a)中所示的示例中,可将用于第二变换的目标区域设置为 4×4 或 8×4 尺寸的区域。

[0617] 另一方面,如在图37(b)中所示的示例中,可不允许设置用于第二变换的目标区域,诸如 4×8 或 8×8 。

[0618] 当垂直方向上的分区被应用于 16×16 尺寸的编码块时,可将编码块分区为 4×16 尺寸的子块。如在所示示例中,当子块具有 $4 \times N$ 尺寸(N是大于4的整数)时,可将用于第二变换的目标区域的宽度设置为不超过4。

[0619] 换言之,如在图38(a)中所示的示例中,可将用于第二变换的目标区域设置为 4×4 或 4×8 尺寸的区域。

[0620] 另一方面,如在图38(b)中所示的示例中,可不允许设置用于第二变换的目标区域,诸如 8×4 或 8×8 。

[0621] 解码器可对比特流中的残差系数进行解码并且通过对残差系数执行反量化来推导变换系数。当通过第一变换和第二变换生成变换系数时,可通过针对变换系数执行第二逆变换和第一逆变换来推导残差样点。

[0622] 当确定对当前块执行第二变换时,可确定作为第二逆变换的目标的目标区域。当一般第二变换被应用时,可将用于第二逆变换的目标区域的尺寸设置为与用于第二变换的目标区域的尺寸相同。在示例中,当通过使用 16×16 尺寸的不可分离变换矩阵针对 4×4 尺寸的区域执行了第二变换时,可将第二逆变换应用于 4×4 尺寸的区域。

[0623] 另一方面,当缩减第二变换被应用时,用于第二逆变换的目标区域的尺寸可具有小于用于第二变换的目标区域的尺寸的值。在示例中,当通过使用 64×48 尺寸的缩减变换矩阵针对 8×8 尺寸的区域执行了第二变换时,可针对包括48个样点(例如,三个 4×4 尺寸的块)的区域执行第二逆变换。

[0624] 在解码器中,可基于当前块的尺寸确定用于第二逆变换的目标区域。在这种情况下,当前块可表示作为执行第二逆变换的目标的编码块或变换块。在示例中,当当前块的宽

度或高度中的至少一个小于阈值时,目标区域可被配置为包括16个样点。另一方面,当当前块的宽度和高度等于或大于阈值时,目标区域可被配置为包括48个样点。

[0625] 可通过将包括在目标区域中的变换系数排列为一行来生成输入矩阵。在这种情况下,当缩减第二变换被应用于当前块时,可基于与缩减因子 k 一样多的变换系数来生成输入矩阵。在示例中,当缩减因子 k 为16时,可基于包括在 4×4 尺寸的左上方块中的变换系数来生成输入矩阵。当缩减因子 k 为32时,可基于包括在左上方块以及与该左上方块的右侧或下方相邻的 4×4 尺寸的邻近块中的变换系数来生成输入矩阵。当缩减因子 k 为48时,可基于左上方块、与该左上方块的右侧相邻的 4×4 尺寸的邻近块以及与该左上方块的下方相邻的 4×4 尺寸的邻近块的变换系数来生成输入矩阵。

[0626] 可在编码器和解码器中预定义缩减因子 k 。可选地,可以以比特流用信号发送用于确定缩减因子 k 的信息。可选地,可基于当前块的尺寸或形状来确定缩减因子 k 。

[0627] 可通过将输入矩阵与不可分离逆变换矩阵相乘来获得变换矩阵。不可分离逆变换矩阵可以是等式6至等式7中所示的不可分离变换矩阵的对称矩阵。等式8和等式9示出通过使用不可分离逆变换矩阵来获得变换矩阵的示例。

[0628] 【等式8】

[0629] $A' = T^T * A$ 当一般第二变换被应用于当前块时,可通过将输入矩阵 A 乘以不可分离逆变换矩阵 T^T 来推导变换矩阵。在示例中,可通过将 16×16 尺寸的逆变换矩阵 T^T 与 16×1 尺寸的输入矩阵 A 相乘来推导 16×1 尺寸的变换矩阵。

[0630] 当获得了变换矩阵 A' 时,变换矩阵 A' 中的分量可被设置为当前块中的 $N \times N$ 尺寸的块的变换系数。在示例中, 16×1 尺寸的变换矩阵可被设置为 4×4 的块的变换系数。

[0631] 【等式9】

[0632] $A' = R^T * A$

[0633] 当缩减第二变换被应用于当前块时,可通过将输入矩阵 A 乘以缩减不可分离逆变换矩阵 R^T 来推导变换矩阵 A' 。在示例中,可通过将 64×16 尺寸的缩减不可分离逆变换矩阵 R^T 与 16×1 尺寸的输入矩阵 A 相乘来推导 64×1 尺寸的变换矩阵。

[0634] 可选地,可通过将 64×32 尺寸的缩减不可分离逆变换矩阵 R^T 与 32×1 尺寸的输入矩阵 A 相乘来推导 64×1 尺寸的变换矩阵。

[0635] 可选地,可通过将 64×48 尺寸的缩减不可分离逆变换矩阵 R^T 与 48×1 尺寸的输入矩阵 A 相乘来推导 48×1 尺寸的变换矩阵。

[0636] 当获得了变换矩阵 A' 时,变换矩阵 A' 中的分量可被设置为当前块中的 $N \times N$ 尺寸的块的变换系数。在示例中, 64×1 尺寸的变换矩阵可被设置为 8×8 的块的变换系数。

[0637] 可基于以比特流用信号发送的索引信息或当前块的尺寸或帧内预测模式中的至少一个来确定不可分离逆变换矩阵。在示例中,可基于当前块的尺寸确定不可分离逆变换矩阵的尺寸,并且可基于当前块的帧内预测模式确定不可分离逆变换矩阵集。当通过当前块的帧内预测模式确定不可分离逆变换矩阵集时,不可分离逆变换矩阵集中包括的与当前块的尺寸对应的多个不可分离逆变换矩阵候选中的至少一个不可分离逆变换矩阵候选可由索引`lfnst_idx`指定。

[0638] 不可分离逆变换矩阵集可包括多个不可分离逆变换矩阵候选。在这种情况下,逆变换矩阵候选的类型或数量中的至少一个在具有不同索引的不可分离逆变换矩阵集之间

可不同。

[0639] 表11表示基于帧内预测模式确定不可分离逆变换集的示例。在表11中, predModeIntra表示帧内预测模式的索引,并且lfnstTrSetIdx表示不可分离逆变换集的索引。

[0640] 【表11】

predModeIntra	lfnstTrSetIdx
predModeIntra<0	1
0<=predModeIntra<=1	0
2<=predModeIntra<=12	1
13<=predModeIntra<=23	2
24<=predModeIntra<=44	3
45<=predModeIntra<=55	2
56<=predModeIntra<=80	1
81<=predModeIntra<=83	0

[0642] 在示例中,当当前块的帧内预测模式为平面时,可选择索引为0的不可分离逆变换矩阵集。之后,可基于当前块的尺寸和lfnst_idx的值来确定不可分离逆变换矩阵。

[0643] 可通过进一步简化表11来确定不可分离逆变换集。在示例中,表11可被简化为表12中那样。

[0644] 【表12】

predModeIntra	lfnstTrSetIdx
predModeIntra<0	1
0<=predModeIntra<=1	0
2<=predModeIntra<50	1
50<=predModeIntra<=83	2

[0646] 可选地,可通过仅考虑帧内预测模式是方向还是非方向来选择不可分离逆变换矩阵集。表13表示基于帧内预测模式是方向还是非方向来确定不可分离逆变换矩阵集的示例。

[0647] 【表13】

predModeIntra	lfnstTrSetIdx
0<=predModeIntra<=1	0
其他	1

[0649] 可选地,在存储了定义帧内预测模式与不可分离逆变换矩阵集之间的映射关系的多个查找表之后,可通过使用多个查找表中的一个查找表来确定不可分离逆变换矩阵集。在示例中,当不可分离逆变换矩阵集被确定时,可选择性地使用表11至表13中的一个表。

[0650] 可以以比特流用信号发送指定多个查找表中的一个查找表的信息。可在序列级、画面级、条带级或块级用信号发送所述信息。

[0651] 可选地,可基于当前块的尺寸或形状或者在第一变换中应用的变换核中的至少一个来选择多个查找表中的一个查找表。

[0652] 当使用预定义尺寸的逆变换矩阵时,可设置为不使用基于帧内预测模式确定不可分离逆变换矩阵集的方法。在示例中,当使用 48×16 尺寸的逆变换矩阵时,可省略基于帧内预测模式确定不可分离逆变换矩阵集的处理。

[0653] 如在上面所提及的描述中,多个不可分离变换矩阵候选中的一个不可分离变换矩阵候选可被应用于当前块。在这种情况下,不可分离变换矩阵候选中的每个不可分离变换矩阵候选可具有至少一个不同的尺寸或系数。

[0654] 图39是示出各种不可分离变换矩阵候选的示例。

[0655] 假设 4×4 的子块是用于第二逆变换的目标。

[0656] 如在图39(a)中所示的示例中,可通过使用 64×16 尺寸的不可分离逆变换矩阵推导 64×1 尺寸的变换矩阵。推导出的变换矩阵可被设置为 8×8 尺寸的块的变换系数。

[0657] 如在图39(b)中所示的示例中,可通过使用 32×16 尺寸的不可分离逆变换矩阵推导 32×1 尺寸的变换矩阵。推导出的变换矩阵可被设置为 4×8 或 8×4 尺寸的块的变换系数。

[0658] 如在图39(c)中所示的示例中,可通过使用 48×16 尺寸的不可分离逆变换矩阵推导 48×1 尺寸的变换矩阵。推导出的变换矩阵可被设置为三个 4×4 尺寸的块的变换系数。在这三个 4×4 尺寸的块中的左上方块中存在非零变换系数,并且在其它块中,块中的变换系数的值可被设置为0。

[0659] 可基于子分区帧内编码方法是否被应用于当前块来确定将被应用于当前块的不可分离逆变换矩阵。在示例中,可选地,当子分区帧内编码方法被应用于当前块时,可设置为应用 32×16 尺寸的不可分离逆变换矩阵。

[0660] 可选地,不可分离逆变换矩阵候选的类型或数量可基于子分区帧内编码方法是否被应用于当前块而被不同地确定。在示例中,仅当子分区帧内编码方法被应用于当前块时, 32×16 尺寸的不可分离逆变换矩阵可被用作候选。

[0661] 当子分区帧内编码方法被应用于当前块并且 32×16 尺寸的不可分离逆变换矩阵被应用时,通过执行第二逆变换而推导出的块的尺寸或形状可根据子分区的尺寸或形状而不同。

[0662] 在示例中,当垂直方向上的分区被应用于当前块时(例如,当子分区具有 $4 \times L$ 形状时),可将通过执行第二逆变换而获得的 32×1 尺寸的变换矩阵设置为 4×8 尺寸的块的变换系数。另一方面,当水平方向上的分区被应用于当前块时(例如,当子分区具有 $L \times 4$ 形状时),可将通过执行第二逆变换而获得的 32×1 尺寸的变换矩阵设置为 8×4 尺寸的块的变换系数。

[0663] 虽然未应用子分区帧内预测编码方法,但是当通过帧内预测对当前块进行编码并且其尺寸为 $4 \times L$ 或 $L \times 4$ 时,可应用 32×16 尺寸的不可分离矩阵。

[0664] 当子分区帧内编码方法被应用于当前块时,可将第二逆变换和第一逆变换应用于当前块所包括的多个子块中的每个子块。

[0665] 可选地,可将第二逆变换和第一逆变换应用于多个子块中的预定义位置处的子块或分区索引小于阈值的子块。在示例中,第二变换可仅被应用于当前块的最上方位置处的子分区或当前块的最左侧位置处的子分区,并且可不被应用于其余子分区。

[0666] 当子分区帧内编码方法被应用时,用于第二逆变换的目标区域可被确定为 4×4 尺寸。可选地,可根据子块的尺寸自适应地确定目标区域的尺寸。在示例中,当子分区的宽度

和高度之中的最小值为4时,可将 4×4 尺寸的区域设置为目标区域。

[0667] 可选地,对于 $4 \times N$ 或 $N \times 4$ 形状的子块,可将 8×4 尺寸的区域设置为目标区域。在这种情况下, N 表示等于或大于8的整数。

[0668] 可通过将子块中的目标区域中包括的变换系数排列为一行来生成输入矩阵。在示例中,当目标区域被设置为 4×4 尺寸时,包括在目标区域中的变换系数可被变换为 16×1 形状的输入矩阵。可选地,当目标区域被设置为 4×8 或 8×4 尺寸时,包括在目标区域中的变换系数可被变换为 32×1 形状的输入矩阵。

[0669] 可通过将输入矩阵乘以不可分离变换矩阵来推导变换矩阵。在示例中,可通过将 16×16 尺寸的不可分离变换矩阵与 16×1 尺寸的输入矩阵相乘来获得 16×1 尺寸的变换矩阵。可选地,可通过将 32×32 尺寸的不可分离变换矩阵与 32×1 尺寸的输入矩阵相乘来获得 32×1 尺寸的变换矩阵。

[0670] 当获得了变换矩阵时,可将变换矩阵中的分量设置为子块的变换系数。在示例中, 16×1 尺寸的变换矩阵可被设置为 4×4 尺寸的块的变换系数。可选地, 32×1 尺寸的变换矩阵可被设置为 4×8 或 8×4 尺寸的左上方块的变换系数。

[0671] 当第二逆变换被应用于子块时,可将预定义变换核应用于子块的水平方向上的变换和垂直方向上的变换。在示例中,可将应用第二逆变换的子块的水平方向上的变换核和垂直方向上的变换核设置为DCT2。

[0672] 可基于子变换块编码方法是否被应用于编码块来确定是否允许第二变换。在示例中,当子变换块编码方法被应用于编码块时,可设置为不应用第二变换。

[0673] 可选地,当子变换块编码方法被应用于编码块时,可设置为仅针对多个子块中的至少一个可用子块应用第二变换。在这种情况下,可用子块可表示多个子块中的执行了第一变换的块。

[0674] 可根据子块的尺寸或形状来确定子块中的用于第二变换的目标区域的尺寸。在示例中,当子块的高度或宽度中的至少一个小于阈值时,可针对 4×4 的区域执行第二变换。另一方面,当子块的高度或宽度中的至少一个等于或大于阈值时,可针对 8×8 的区域执行第二变换。

[0675] 可基于子块的尺寸、形状、位置或分区索引中的至少一个来确定第二变换是否被应用于子块。在示例中,第二变换可仅被应用于包括编码块的左上样点的子块。可选地,仅当子块的高度或宽度中的至少一个大于阈值时,可应用第二变换。

[0676] 可选地,可以以比特流用信号发送表示第二变换是否被应用于子块的信息。

[0677] 当子变换块编码方法被应用时,可设置为不允许缩减第二变换。可选地,虽然应用了子变换块编码方法,但是可基于子块的尺寸或形状中的至少一个来确定缩减第二变换是否被执行。

[0678] 可通过预测样点与残差样点之和来推导执行了变换的子块的重建样点。另一方面,预测样点可被设置为省略了变换的子块中的重建样点。量化用于降低块的能量,并且量化处理包括将变换系数除以特定常数值的处理。所述常数值可由量化参数来推导,并且量化参数可被定义为从1到63的值。

[0679] 当获得了当前块的重建块时,可经由环内滤波来减少在量化和编码处理中产生的信息损失。环内滤波器可包括去块滤波器、样点自适应偏移滤波器(SAO)或自适应环路滤波

器 (ALF) 中的至少一个。在下文中,应用环内滤波器之前的重建块被称为第一重建块,并且应用环内滤波器之后的重建块被称为第二重建块。

[0680] 可通过将去块滤波器、SAO或ALF中的至少一个应用于第一重建块来获得第二重建块。在这种情况下,可在应用去块滤波器之后应用SAO或ALF。

[0681] 可考虑将画面划分为多个区域并且并行地对多个区域进行编码/解码以用于高分辨率图像 (诸如全景视频、360度视频或4K/8K UHD (超高清) 图像等) 的实时或低延迟编码的方法。具体地,出于处理目的,可将画面分区为并行块或条带 (或并行块组)。

[0682] 并行块表示并行编码/解码的基本单元。可并行地对每个并行块进行处理。并行块可具有矩形形状。在另一实施例中,可允许非矩形并行块。

[0683] 可以以比特流用信号发送表示非矩形并行块是否被允许或非矩形并行块是否存在的信息。

[0684] 当一并行块被编码/解码时,不能使用另一并行块的数据。可通过去除并行块之间的编码/解码依赖性来支持并行块的并行处理。具体地,可以以并行块为单位对CABAC (上下文自适应二进制算术编码) 上下文的概率表进行初始化,并且可将环内滤波器设置为不被应用于并行块之间的边界。此外,另一并行块中的数据可不被用于推导运动矢量的候选。例如,另一并行块中的数据可不被用作合并候选、运动矢量预测候选 (AMVP候选) 或运动信息候选。此外,另一并行块中的数据可不被用于符号的上下文计算。

[0685] 图像编码/解码信息可通过条带头被用信号发送。通过条带头被用信号发送的信息可被共同应用于包括在条带中的并行块或编码树单元。条带也可被称为并行块组。

[0686] 图40是示出根据本公开的实施例的画面分区方法的示意图。

[0687] 首先,可确定是否将当前画面分区为多个处理单元 (S4001)。在这种情况下,处理单元可包括并行块或条带中的至少一个。在示例中,可以以比特流用信号发送表示当前画面是否被分区为多个并行块或条带的语法no_pic_partition_flag。当语法no_pic_partition_flag的值为0时,这表示当前画面被分区为至少一个并行块或至少一个条带。另一方面,当语法no_pic_partition_flag的值为1时,这表示当前画面未被分区为多个并行块或条带。

[0688] 当确定当前画面未被分区为多个处理单元时,可终止当前画面的分区处理。在这种情况下,可理解的是,当前画面利用一个并行块和一个条带 (或一个并行块组) 来配置。

[0689] 可选地,可以以比特流用信号发送表示画面中是否存在多个并行块的信息。所述信息可包括表示画面中是否存在多个并行块的1比特标志或指定画面中的并行块的数量信息中的至少一个。仅当确定画面中存在多个并行块时,可执行后面描述的画面分区处理。

[0690] 当确定当前画面被分区为多个处理单元时,可以以比特流用信号发送并行块分区信息。基于用信号发送的并行块分区信息,可将画面分区为至少一个并行块 (S4002)。

[0691] 当当前画面被分区为多个并行块时,可通过合并多个并行块或通过并行块进行分区来确定条带 (S4003)。

[0692] 在下文中,根据本公开,详细描述对并行块进行分区的方法和确定条带的方法。

[0693] 图41示出将画面分区为多个并行块的示例。

[0694] 并行块可包括至少一个编码树单元。并行块的边界可与编码树单元的边界匹配。换言之,可不允许将单个编码树单元分区为多个部分的分区类型。

[0695] 当画面被分区为多个并行块时,相邻并行块的高度或相邻并行块的宽度可被设置为具有相同的值。

[0696] 在示例中,如在图41中所示的示例中,属于相同并行块行的并行块的高度和/或属于相同并行块列的并行块的宽度可被设置为相同。属于相同并行块行的并行块可被称为水平方向上的并行块集,并且属于相同并行块列的并行块可被称为垂直方向上的并行块集。

[0697] 可选地,可用信号发送表示将被编码/解码的并行块的宽度和/或高度是否与先前并行块的宽度和/或高度相同的信息。

[0698] 可以以比特流用信号发送表示画面的分区类型的信息。所述信息可通过画面参数集、序列参数集或条带头而被编码并被用信号发送。

[0699] 表示画面的分区类型的信息可包括表示并行块是否按统一尺寸被分区的信息、表示并行块列的数量的信息或表示并行块行的数量的信息中的至少一个。在这种情况下,并行块列的数量表示垂直方向上的并行块集的数量,并且并行块行的数量表示水平方向上的并行块集的数量。

[0700] 当通过使用跨画面的垂直线或水平线中的至少一个来对画面进行分区时,每个并行块属于不同的列和/或行。为了确定画面的分区类型,可用信号发送表示并行块列的数量和/或并行块行的数量的信息。在示例中,可以以比特流用信号发送表示通过对画面进行分区而生成的并行块行的数量的信息`num_tile_row_minus1`以及表示通过对画面进行分区而生成的并行块列的数量的信息`num_tile_column_minus1`。语法`num_tile_row_minus1`表示从并行块行的数量减去1得到的值,语法`num_tile_column_minus1`表示从并行块列的数量减去1得到的值。

[0701] 在图41中所示的示例中,并行块列的数量是4,并且并行块行的数量是3。因此,`num_tile_columns_minus1`可表示3,并且`num_tile_rows_minus1`可表示2。

[0702] 可以以比特流用信号发送表示每个并行块列的宽度的语法和表示每个并行块行的高度的语法。在示例中,`tile_cols_width_minus1[i]`可表示第*i*并行块列的宽度,并且`tile_rows_height_minus1[j]`可表示第*j*并行块行的高度。

[0703] 语法`tile_cols_width_minus1[i]`表示从配置第*i*并行块列的编码树单元列的数量减去1得到的值。对于最后并行块列,可省略用信号发送语法`tile_cols_width_minus1[i]`的操作。可通过从当前画面的宽度减去先前并行块列的宽度来推导最后并行块列的宽度。

[0704] 语法`tile_rows_height_minus1[j]`表示从配置第*j*并行块行的编码树单元行的数量减去1得到的值。对于最后并行块行,可省略用信号发送语法`tile_rows_height_minus1[i]`的操作。可通过从当前画面的高度减去先前并行块行的高度来推导最后并行块行的高度。

[0705] 在另一示例中,可以以比特流用信号发送表示宽度信息被显式地用信号发送的并行块列的数量的信息和/或表示高度信息被显式地用信号发送的并行块行的数量的信息。表14表示包括所述信息的语法表。

[0706] 【表14】

[0707]

pic_parameter_set_rbsp() {	描述符
...	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	

[0708]

log2_pps_ctu_size_minus5	u(2)
num_exp_tile_columns_minus1	ue(v)
num_exp_tile_rows_minus1	ue(v)
for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
rect_slice_flag	u(1)
if(rect_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1 num_slices_in_tile_minus1[i]	=
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i][j]	ue(v)
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)	
tile_idx_delta[i]	se(v)
}	
}	
...	
}	

	}	
[0709]	...	
	}	

[0710] 可以以比特流用信号发送通过从宽度信息被显式地用信号发送的并行块列的数量减去1而推导出的语法num_exp_tile_columns_minus1。

[0711] 当通过语法num_exp_tile_columns_minus1确定宽度信息被显式地用信号发送的并行块列的数量时,宽度信息tile_column_width_minus1[i]可与所确定的数量一样多地被用信号发送。在这种情况下,i可具有0与num_exp_tile_columns_minus1之间的值。

[0712] 可基于针对对应并行块列被用信号发送的语法tile_column_width_minus1[k]来确定索引k等于或小于num_exp_tile_columns_minus1的并行块列的宽度。具体地,可通过将通过将语法tile_column_width_minus1[k]的值加1而推导出的值乘以编码树单元的宽度来确定索引k等于或小于num_exp_tile_columns_minus1的并行块列的宽度。

[0713] 另一方面,可基于最后被用信号发送的宽度信息和画面中的其余编码树单元列的数量来确定索引k大于num_exp_tile_columns_minus1的并行块列的宽度。在示例中,当通过与最后被用信号发送的宽度相关的语法tile_column_width_minus1[num_exp_tile_columns_minus1]推导出的并行块列的宽度是LastColWidth(即,tile_column_width_minus1[num_exp_tile_columns_minus1]+1)并且当前画面中的除了由先前并行块列占据的区域之外的其余编码树单元列的数量是remainingWidthInCtbY时,可将第k并行块列的宽度设置为LastColWidth和remainingWidthInCtbY之中的较小值。

[0714] 表15是描述确定并行块列的宽度的处理的示例。

[0715] 【表15】

[0716]	<pre>remainingWidthInCtbsY = PicWidthInCtbsY for(i = 0; i <= num_exp_tile_columns_minus1; i++) { colWidth[i] = tile_column_width_minus1[i] + 1 remainingWidthInCtbsY -= colWidth[i] } uniformTileColWidth</pre>	=
[0717]	<pre>tile_column_width_minus1[num_exp_tile_columns_minus1] + 1 while(remainingWidthInCtbsY >= uniformTileColWidth) { colWidth[i++] = uniformTileColWidth remainingWidthInCtbsY -= uniformTileColWidth } if(remainingWidthInCtbsY > 0) colWidth[i++] = remainingWidthInCtbsY NumTileColumns = i</pre>	

[0718] 在表15中,变量PicWidthInCtbsY表示画面中的编码树单元列的总数。在示例中,可基于以下等式10推导变量PicWidthInCtbsY。

[0719] 【等式10】

[0720]
$$\text{PicWidthInCtbsY} = \text{Ceil}(\text{pic_width_in_luma_samples} / \text{CtbSizeY})$$

[0721] 在等式10中,pic_width_in_luma_samples表示画面中的水平方向上的亮度样点的数量。变量CtbSizeY可以是表示编码树单元的值的值。

[0722] 可通过从变量PicWidthInCtbsY减去先前并行块列的累加宽度来推导变量remainingWidthInCtbsY。

[0723] 可以以比特流用信号发送通过从高度信息被显式地用信号发送的并行块行的数量减去1而推导出的语法num_exp_tile_rows_minus1。

[0724] 当通过语法num_exp_tile_rows_minus1确定高度信息被显式地用信号发送的并行块行的数量时,高度信息tile_row_height_minus1[i]可与所确定的数量一样多地被用信号发送。在这种情况下,i可具有0与num_exp_tile_rows_minus1之间的值。

[0725] 可基于针对对应并行块行被用信号发送的语法tile_row_Height_minus1[k]来确定索引k等于或小于num_exp_tile_rows_minus1的并行块行的高度。具体地,可通过将通过将语法tile_row_Height_minus1[k]的值加1而推导出的值乘以编码树单元的高度来确定索引k等于或小于num_exp_tile_rows_minus1的并行块行的高度。

[0726] 另一方面,可基于最后被用信号发送的高度信息和画面中的其余编码树单元行的数量来确定索引k大于num_exp_tile_rows_minus1的并行块行的高度。在示例中,当通过与最后被用信号发送的高度信息相关的语法tile_row_height_minus1[num_exp_tile_rows_minus1]推导出的并行块行的高度为LastRowHeight (即,tile_row_height_minus1[num_exp_tile_rows_minus1]+1)并且当前画面中的除了由先前并行块行占据的区域之外的其余编码树单元行的数量为remainingHeightInCtbsY时,可将第k并行块行的高度设置为LastColHeight和remainingHeightInCtbsY之中的较小值。

[0727] 表16是描述确定并行块行的高度的处理的示例。

[0728] 【表16】

[0729]

```

remainingHeightInCtbsY = PicHeightInCtbsY
for( j = 0; j < num_exp_tile_rows_minus1; j++ ) {
    RowHeight[ j ] = tile_row_height_minus1[ j ] + 1
    remainingHeightInCtbsY -= RowHeight[ j ]
}
uniformTileRowHeight = tile_row_height_minus1[ num_exp_tile_rows_minus1 ] +
1
while( remainingHeightInCtbsY >= uniformTileRowHeight ) {
    RowHeight[ j++ ] = uniformTileRowHeight
    remainingHeightInCtbsY -= uniformTileRowHeight
}
if( remainingHeightInCtbsY > 0 )
    RowHeight[ j++ ] = remainingHeightInCtbsY
NumTileRows = j

```

[0730] 在表16中,变量PicHeightInCtbsY表示画面中的编码树单元行的总数。在示例中,可基于以下等式11推导变量PicHeightInCtbsY。

[0731] 【等式11】

[0732] $\text{PicHeightInCtbsY} = \text{Ceil}(\text{pic_height_in_lima_samples} / \text{CtbSizeY})$

[0733] 在等式11中,pic_height_in_luma_samples表示画面中的垂直方向上的亮度样点的数量。变量CtbSizeY可以是表示编码树单元的值的值。

[0734] 可通过从变量PicHeightInCtbsY减去先前并行块行的累加高度来推导变量remainingheightInCtbsY。

[0735] 通过将图41用作示例来描述4个并行块列中的每个并行块列利用2个编码树单元列来配置。因此,仅针对第一并行块列,语法tile_column_width_minus1[0]可作为值1被用信号发送,并且其余并行块列的宽度可被设置为与tile_column_width_minus1[0]相同。由于将被用信号发送的语法tile_column_width_minus1的数量为1,因此可将语法num_exp_tile_columns_minus1的值设置为0。

[0736] 在图41中,第一并行块行利用3个编码树单元行来配置,并且第二并行块行和第三并行块行利用2个编码树单元行来配置。由于可基于第二并行块行的高度信息推导第三并行块行的高度,因此可仅针对第一并行块行和第二并行块行用信号发送高度信息。在示例中,针对第一并行块行的语法tile_row_height_minus1[0]可按值2被用信号发送,并且针对第二并行块行的语法tile_row_height_minus1[1]可按值1被用信号发送。由于将被用信号发送的语法tile_row_width_minus1的数量为2,因此可将语法num_exp_tile_rows_minus1的值设置为1。

[0737] 另一方面,可通过序列参数集或画面参数集用信号发送表示编码树单元的值的值的信息。

[0738] 一个并行块可利用至少一个编码树单元配置。除了与画面的右边界或下边界相邻

的并行块之外的其余并行块可被设置为不通过包括小于编码树单元的区域配置。换言之，并行块的边界与编码树单元的边界匹配。

[0739] 语法loop_filter_across_tiles_enabled_flag表示是否允许对参考画面参数集的画面中的并行块的边界应用环内滤波器。在这种情况下，环内滤波器可包括去块滤波器、ALF或SA0中的至少一个。当标志loop_filter_across_tiles_enabled_flag的值为1时，这表示可应用跨参考画面参数集的画面中的并行块的边界的环内滤波器。另一方面，当标志loop_filter_across_tiles_enabled_flag的值为0时，这表示不允许对参考画面参数集的画面中的并行块的边界应用环内滤波器。

[0740] 语法loop_filter_across_slices_enabled_flag表示是否允许对参考画面参数集的画面中的条带的边界应用环内滤波器。在这种情况下，环内滤波器可包括去块滤波器、ALF或SA0中的至少一个。当标志loop_filter_across_slices_enabled_flag的值为1时，这表示可应用跨参考画面参数集的画面中的条带的边界的环内滤波器。另一方面，当标志loop_filter_across_slices_enabled_flag的值为0时，这表示不允许对参考画面参数集的画面中的条带的边界应用环内滤波器。

[0741] 表17示出包括表示画面是否被分区为多个区域的标志no_pic_partition_flag的语法表。

[0742] 【表17】

[0743]	no_pic_partition_flag	u(1)
	if(!no_pic_partition_flag){	
	pps_log2_ctu_size_minus5	u(2)
	num_exp_tile_columns_minus1	ue(v)
	num_exp_tile_rows_minus1	ue(v)
	for(i=0; i<=num_exp_tile_columns_minus1; i++)	
	tile_column_width_minus1[i]	ue(v)
	for(i=0; i<=num_exp_tile_rows_minus1; i++)	
	tile_row_height_minus1[i]	ue(v)

[0744] 当标志no_pic_partition_flag为1时，这表示画面或子画面未被分区为多个并行块或多个条带。当标志no_pic_partition_flag的值为1时，可省略对与并行块分区结构和/或条带分区结构相关的语法的编码。

[0745] 另一方面，当no_pic_partition_flag为0时，这表示画面或子画面可被分区为多个并行块或多个条带。当标志no_pic_partition_flag的值为0时，可用信号发送用于确定编码树单元的尺寸的语法pps_log2_ctu_size_minus5。

[0746] 利用多个编码树单元配置的矩形区域可被定义为并行块。换言之，单个编码树单元可能无法被定义为并行块。如上所述，当多个编码树单元被定义为并行块时，可根据语法tile_column_width_minus1的值来确定语法tile_row_height_minus1的值的范围，或者可根据语法tile_row_height_minus1的值来确定语法tile_column_width_minus1的值的范围。

[0747] 在示例中,当语法tile_column_width_minus1的值为0时,这表示一个编码树单元列被包括在并行块中。如上所述,由于一个并行块利用多个编码树单元来配置,因此表示并行块的高度的tile_row_height_minus1应具有大于0的值。换言之,当语法tile_column_width_minus1在语法tile_row_height_minus1之前被编码并被用信号发送时,并且如果语法tile_column_width_minus1的值为0,则语法tile_row_height_minus1可具有等于或大于1的值。

[0748] 另一方面,当语法tile_row_height_minus1在语法tile_column_width_minus1之前被编码并被用信号发送时,并且如果语法tile_row_height_minus1的值为0,则语法tile_column_width_minus1可具有等于或大于1的值。

[0749] 如在所述示例中,当并行块利用一个编码树单元列来配置时,该并行块应包括多个编码树行。换言之,当表示针对第i并行块的宽度的语法tile_column_width_minus1[i]的值为0时,表示针对第i并行块的高度的语法tile_row_height_minus1[i]的值应等于或大于1。

[0750] 此外,当并行块利用一个编码树单元行来配置时,该并行块应包括多个编码树列。换言之,当表示针对第i并行块的高度的语法tile_row_height_minus1[i]的值为0时,表示针对第i并行块的宽度的语法tile_row_width_minus1[i]的值应等于或大于1。

[0751] 因此,当语法tile_column_width_minus1[i]在语法tile_row_height_minus1[i]之前被编码时,可如下表18中那样定义每个语法。

[0752] 【表18】

	tile_column_width_minus1[i]加1指定以CTB为单位的第i并行块列的宽度,i在0至num_exp_tile_columns_minus1-1的范围(包括0和num_exp_tile_columns_minus1-1)中。如第6.5.1节中所说明的,tile_column_width_minus1[num_exp_tile_columns_minus1]被用于推导索引大于或等于num_exp_tile_columns_minus1的并行块列的宽度。当不存在时,tile_column_width_minus1[0]的值被推断为等于PicWidthInCtbsY-1。
[0753]	tile_row_height_minus1[i]加1指定以CTB为单位的第i并行块行的高度,i在0至num_exp_tile_rows_minus1-1的范围(包括0和num_exp_tile_rows_minus1-1)中。如第6.5.1节中说明的,tile_row_height_minus1[num_exp_tile_rows_minus1]被用于推导索引大于或等于num_exp_tile_rows_minus1的并行块行的高度。当不存在时,tile_row_height_minus1[0]的值被推断为等于PicHeightInCtbsY-1。当tile_column_width_minus1[i]的值之一等于0时,tile_row_height_minus1[i]的所有值应大于或等于1。

[0754] 反之,当语法tile_row_height_minus1[i]在语法tile_column_width_minus1[i]之前被编码时,可如下表19中那样定义每个语法。

[0755] 【表19】

[0756]

tile_column_width_minus1[i]加 1 指定以 CTB 为单位的第 i 并行块列的宽度, i 在 0 至 num_exp_tile_columns_minus1 - 1 (包括 0 和

[0757]

num_exp_tile_columns_minus1 - 1) 的范围中。如第 6.5.1 条中所说明的, tile_column_width_minus1[num_exp_tile_columns_minus1] 被用于推导索引大于或等于 num_exp_tile_columns_minus1 的并行块列的宽度。当不存在时, tile_column_width_minus1[0] 的值被推断为等于 PicWidthInCtbsY - 1。当 tile_row_height_minus1[i] 的值之一等于 0 时, tile_column_width_minus1[i] 的所有值应大于或等于 1。

tile_row_height_minus1[i]加 1 指定以 CTB 为单位的第 i 并行块行的高度, i 在 0 至 num_exp_tile_rows_minus1 - 1 的范围 (包括 0 和 num_exp_tile_rows_minus1 - 1) 中。如第 6.5.1 节中所说明的, tile_row_height_minus1[num_exp_tile_rows_minus1] 被用于推导索引大于或等于 num_exp_tile_rows_minus1 的并行块行的高度。当不存在时, tile_row_height_minus1[0] 的值被推断为等于 PicHeightInCtbsY - 1。

[0758] 在另一示例中,当语法tile_column_width_minus1[i]的值为0时,可对语法tile_row_height_minus2[i]进行编码,而不是语法tile_row_height_minus1[i]。可通过从第i并行块所包括的编码树单元行的数量减去2来推导语法tile_row_height_minus2[i]。

[0759] 可选地,当语法tile_row_height_minus1[i]的值为0时,可对语法tile_column_width_minus2[i]进行编码,而不是语法tile_column_width_minus1[i]。可通过从第i并行块所包括的编码树单元列的数量减去2来推导语法tile_column_width_minus2[i]。

[0760] 至少一个或更多个并行块可被定义为一个处理单元。在示例中,多个并行块可被定义为一个条带。条带可被称为并行块组。

[0761] 可选地,一个并行块可被分区为多个处理单元。在示例中,一个并行块可被分区为多个条带。在这种情况下,一个条带可包括至少一个编码树单元列。当一个并行块被分区为多个条带时,可以以比特流用信号发送表示每个条带的高度的信息。

[0762] 配置条带的四个边中的至少一个边可与画面的边界和/或并行块的边界匹配。在示例中,条带的左边界或上边界可被设置为与画面的左边界或上边界匹配。可选地,条带的四个边中的至少一个边可位于并行块的边界。

[0763] 图像编码/解码信息可通过条带头被用信号发送。通过条带头被用信号发送的信息可被共同应用于属于条带的并行块和/或块。

[0764] 可以以比特流用信号发送表示条带类型的信息。该信息表示当前画面中的条带的定义方法。在示例中,可以以比特流用信号发送表示条带类型的语法rect_slice_flag。

[0765] 语法rect_slice_flag表示条带是否基于并行块的光栅扫描顺序被定义或者条带是否按矩形形状被定义。在示例中,当rect_slice_flag为0时,这表示条带基于并行块的光栅扫描顺序被定义。另一方面,当rect_slice_flag为1时,这表示条带按矩形形状被定义。

[0766] 在下文中,详细描述用于确定条带的两种方法。

[0767] 基于光栅扫描的定义方法是根据光栅扫描顺序指定至少一个或更多个并行块,并将至少一个或更多个指定的并行块定义为条带。当基于光栅扫描的定义方法被应用时,可将一个或更多个连续并行块定义为条带。在这种情况下,可根据光栅扫描顺序确定连续并行块。当光栅扫描条带被应用时,可生成非矩形条带。

[0768] 当条带基于光栅扫描顺序被定义时,可用信号发送表示每个条带所包括的并行块的数量信息。对于最后条带,可省略用信号发送表示条带所包括的并行块的数量信息的操作。

[0769] 当条带包括多个并行块时,条带中包括的并行块的宽度或高度可不同。

[0770] 矩形形状的条带的定义方法是仅允许矩形形状的条带的分区方法。当矩形形状的条带的定义方法被应用时,位于条带的四个角的并行块属于同一行或同一列。

[0771] 可选地,当矩形形状的条带的定义方法被应用时,可将一个并行块分区为多个矩形条带。

[0772] 当矩形形状的条带的定义方法被应用时(例如,当`rect_slice_flag`为1时),可用信号发送表示画面是否利用单个条带来配置的信息。在示例中,可以以比特流用信号发送表示画面中的条带的数量是否为1的语法`one_slice_in_pic_flag`。当标志`one_slice_in_pic_flag`为1时,这表示画面利用单个条带来配置。另一方面,当标志`one_slice_in_pic_flag`为0时,这表示画面利用至少两个或更多个条带来配置。

[0773] 可选地,可用信号发送表示每个子画面是否利用一个条带来配置的信息。在示例中,可用信号发送表示每个子画面是否利用单个条带来配置的标志`single_slice_per_subpic`。当标志`single_slice_per_subpic`为1时,这表示每个子画面利用单个条带来配置。在这种情况下,可确定条带分区结构与通过参考序列参数集确定的子画面分区结构相同。另一方面,当`single_slice_per_subpic`为0时,这表示子画面分区结构不同于条带分区结构。当标志`single_slice_per_subpic`的值为0时,可另外用信号发送用于确定条带分区结构的信息。

[0774] 基于条带所包括的并行块列的数量和/或并行块行的数量,可确定条带的尺寸。通过参考表14描述可以以比特流用信号发送表示第*i*条带的高度的语法`slice_height_in_tiles_minus1[i]`。语法`slice_height_in_tiles_minus1[i]`表示从第*i*条带所包括的并行块行的数量减去1得到的值。在这种情况下,可通过将第*i*条带所包括的每个并行块行的高度相加来推导第*i*条带的高度。

[0775] 可以以比特流用信号发送表示第*i*条带的宽度的语法`slice_width_in_tiles_minus1[i]`。语法`slice_width_in_tiles_minus1[i]`表示从第*i*条带所包括的并行块列的数量减去1得到的值。在这种情况下,可通过将第*i*条带所包括的每个并行块列的宽度相加来推导第*i*条带的宽度。

[0776] 图42是用于描述用信号发送条带的尺寸信息的方面的示意图。

[0777] 在图42中所示的示例中,索引为4的条带(条带4)利用2个并行块列和2个并行块行来配置。因此,针对条带4的表示条带的宽度的语法`slice_width_in_tiles_minus1[4]`和表示条带的高度的语法`slice_height_in_tiles_minus1[4]`可被分别设置为1。

[0778] 一个并行块可被分区为多个条带。在示例中,可通过使用至少一条垂直线将一个并行块分区为多个条带。

[0779] 当一个并行块被分区为多个条带时,针对第*i*条带的宽度信息和高度信息可被分别设置为0。在示例中,当表示第*i*条带的宽度的语法slice_width_in_tiles_minus1[*i*]和表示第*i*条带的高度的语法slice_height_in_tiles_minus1[*i*]两者为0时,可用信号发送表示对应并行块所包括的条带的数量的语法num_slices_in_tiles_minus1[*i*]。语法num_slices_in_tiles_minus1[*i*]表示从并行块所包括的条带的数量减去1得到的值。

[0780] 当语法num_slices_in_tiles_minus1[*i*]的值为0时,这表示第*i*条带利用一个并行块来配置。另一方面,当语法num_slices_in_tiles_minus1[*i*]的值大于0时,这表示包括第*i*条带的并行块被分区为多个条带。当语法num_slices_in_tiles_minus1[*i*]的值大于0时,可另外用信号发送表示每个并行块的高度的信息。

[0781] 在示例中,可通过从对应并行块中的第*j*条带所包括的编码树单元行的数量减去1来推导语法slice_height_in_ctu_minus1[*i*][*j*]。对于对应并行块中的最后条带,可省略对语法slice_height_in_ctu_minus1的编码。可基于并行块中的其余编码树单元行的高度来确定最后条带的高度。

[0782] 可用信号发送用于标识每个条带所包括的并行块的信息。

[0783] 具体地,用于指定条带中包括的并行块的索引的差信息可被编码并被用信号发送。在示例中,表示减去第(*i*+1)条带中的预定位置处的并行块的索引和第*i*条带中的预定位置处的并行块的索引的值的语法tile_idx_delta[*i*]可被编码。在这种情况下,预定位置处的并行块可以是条带中的左上方并行块或右下方并行块。

[0784] 对于最后条带,可省略对差信息的编码。

[0785] 利用差信息,表示差信息的符号的信息还可被编码并被用信号发送。在示例中,可以以比特流用信号发送表示语法tile_idx_delta[*i*]的值是正数还是负数的语法tile_idx_delta_sign[*i*]。在这种情况下,由于第一条带(即,*i*为0的条带)与第二条带(即,*i*为1的条带)之间的差值总是正数,因此可针对第一条带省略对语法tile_idx_delta_sign[*i*]的编码。

[0786] 当第*i*条带和第(*i*+1)条带未被包括在一个并行块中时,第*i*条带与第(*i*+1)条带之间的并行块索引差的绝对值总是具有大于1的值。因此,通过从第*i*条带与第(*i*+1)条带之间的并行块索引差的绝对值减去1推导出的语法tile_idx_delta_minus1[*i*]可被编码并被用信号发送。

[0787] 表20示出包括差信息的语法表。

[0788] 【表20】

[0789]	pic_parameter_set_rbsp() {	描述符
	...	
	single_tile_in_pic_flag	u(1)
	if(!single_tile_in_pic_flag) {	
	log2_pps_ctu_size_minus5	u(2)
	num_exp_tile_columns_minus1	ue(v)
	num_exp_tile_rows_minus1	ue(v)

[0790]

for(i = 0; i <= num_exp_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i <= num_exp_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
loop_filter_across_tiles_enabled_flag	u(1)
loop_filter_across_slices_enabled_flag	u(1)
rect_slice_flag	u(1)
if(rect_slice_flag) {	
num_slices_in_pic_minus1	ue(v)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0) {	
num_slices_in_tile_minus1[i]	ue(v)
numSlicesInTileMinus1	=
num_slices_in_tile_minus1[i]	
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i][j]	ue(v)
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1){	
tile_idx_delta_minus1[i]	ue(v)
if(i>0)	
tile_idx_delta_sign[i]	u(1)
}	
}	
}	
...	
}	
}	

[0791]

...	
}	

[0792] 当一个并行块包括多个条带时,所述多个条带被包括在同一并行块中,由此所述多个条带之间的并行块索引差值变为0。因此,可省略对包括在一个并行块中的多个条带之间的差值的编码。由于省略了对属于同一并行块的条带之间的差信息的编码,因此语法 `tile_idx_delta_minus1[i]` 可总是被设置为等于或大于0的值。

[0793] 此外,当一个并行块包括多个条带时,可仅针对多个条带中的一个条带对差信息进行编码。在示例中,可仅针对一个并行块中包括的多个条带中的第一条带或最后条带对语法 `tile_idx_delta_minus1[i]` 进行编码。在示例中,由语法 `tile_idx_delta_minus1[i]` 指定的第*i*条带表示属于第一并行块的多个条带中的第一条带或最后条带,并且第(*i*+1)条带表示属于与第一并行块不同的第二并行块的条带。

[0794] 语法 `tile_idx_delta_sign[i]` 表示并行块索引差值的符号。在示例中,可如在以下等式12中确定并行块索引差值的符号 `TileIdxDeltaSign`。

[0795] 【等式12】

[0796]
$$\text{TileIdxDeltaSign}[i] = \text{tile_idx_delta_sign}[i] ? 1 : -1$$

[0797] 如等式12中所示,当语法 `tile_idx_delta_sign[i]` 的值为1时,这表示由语法 `tile_idx_delta_minus1[i]` 推导出的并行块索引差值为正数。当语法 `tile_idx_delta_sign[i]` 的值为0时,这表示由语法 `tile_idx_delta_minus1[i]` 推导出的并行块索引差值为负数。

[0798] 可通过将由语法 `tile_idx_delta_minus1[i]` 推导出的并行块索引差值(例如, `tile_idx_delta_minus1[i]+1`)乘以由语法 `tile_idx_delta_sign[i]` 推导出的符号值来推导并行块索引差。

[0799] 对于第一条带(例如,*i*为0的条带)、最后条带(例如,*i*为 `num_slices_in_pic_minus1` 的条带)或最后条带的前一条带(例如,*i*为 `num_slices_in_pic_minus1-1` 的条带)中的至少一个,可省略对语法 `tile_idx_delta_sign[i]` 的编码。当省略了对语法 `tile_idx_delta_sign[i]` 的编码时,该值可被推断为1。

[0800] 可选地,可省略对并行块索引差值的符号信息的编码,并且可将并行块索引差值设置为始终具有正值。

[0801] 基于条带的位置,可确定并行块索引差信息是否被编码。表21示出基于条带的位置来确定并行块索引差信息是否被解析的示例。

[0802] 【表21】

[0803]

rect_slice_flag	u(1)
if(rect_slice_flag)	
single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
tile_idx_delta_present_flag	u(1)
if(num_slices_in_pic_minus1)	
multiple_slice_in_tile_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if(multiple_slice_in_tile_present_flag)	
multiple_slice_in_tile_flag[i]	u(1)
if ((!multiple_slice_in_tile_flag[i] !multiple_slice_in_tile_present_flag)) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
} else {	
num_slices_in_tile_minus2[i]	ue(v)
numSlicesInTileMinus1 num_slices_in_tile_minus2[i] + 1	=
for(j = 0; j < numSlicesInTileMinus1; j++)	
slice_height_in_ctu_minus1[i][j]	ue(v)
}	
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1) {	
if ((tileIdx / NumTileColumns) != NumTileRow - 1)	
tile_idx_delta_minus1[i]	ue(v)

[0804]	if (i>0){	
	if ((tileIdx % NumTileColumns) != NumTileColumn) (tileIdx% NumTileRows) != NumTileRows)	
	tile_idx_delta_sign[i]	u(1)
	}	
	}	
	}	
	}	

[0805] 在表21中,变量tileIdx表示第i条带中的左上方并行块的索引。

[0806] 当第i条带所包括的左上方并行块属于画面中的最后并行块行时,可省略对表示并行块索引差值的语法tile_idx_delta_minus1[i]的编码/解码。换言之,仅当第i条带的左上方并行块不属于画面中的最后并行块行时,语法tile_idx_delta_minus1[i]可被编码。

[0807] 当省略了对语法tile_idx_delta_minus1[i]的编码时,语法tile_idx_delta_minus1[i]的值可被推断为与语法slice_width_in_tiles_minus1[i]的值相同。

[0808] 当第i条带所包括的左上方并行块属于画面中的最后并行块列时,或者当第i条带所包括的左上方并行块属于画面中的最后并行块行时,可省略对表示并行块索引差值的符号的语法tile_idx_delta_sign[i]的编码/解码。

[0809] 当省略了对语法tile_idx_delta_sign[i]的编码时,语法tile_idx_delta_sign[i]的值可被推断为1。

[0810] 在上述示例中,示出了当第i条带所属的并行块包括多个条带时,表示属于该并行块的条带的数量的信息和条带中的每个条带的高度信息被用信号发送。

[0811] 在另一示例中,当第i条带所属的并行块包括多个条带时,可用信号发送高度信息被显式地用信号发送的条带的数量信息,并且高度信息可与由所述数量信息指定的数量一样多地被用信号发送。表22示出包括高度信息应被显式地用信号发送的数量信息的语法表。

[0812] 【表22】

[0813]	rect_slice_flag	u(1)
	if(rect_slice_flag)	

[0814]

single_slice_per_subpic_flag	u(1)
if(rect_slice_flag && !single_slice_per_subpic_flag) {	
num_slices_in_pic_minus1	ue(v)
tile_idx_delta_present_flag	u(1)
multiple_slices_in_tile_present_flag	u(1)
for(i = 0; i < num_slices_in_pic_minus1; i++) {	
if (multiple_slice_in_tile_present_flag)	
multiple_slices_in_tile_flag[i]	u(1)
if (!multiple_slices_in_tile_flag[i]	
!multiple_slices_in_tile_present_flag) {	
slice_width_in_tiles_minus1[i]	ue(v)
slice_height_in_tiles_minus1[i]	ue(v)
}else{	
num_exp_slices_in_tile_minus1[i]	ue(v)
numSlicesInTile	=
num_exp_slices_in_tile_minus1[i]+1	
for(j = 0; j < numSlicesInTile; j++)	
slice_height_in_ctu_minus1[i][j]	ue(v)
if(numExpSlicesInTile)	
i += RemNumSlicesMinus1	
}	
if(tile_idx_delta_present_flag && i <	
num_slices_in_pic_minus1) {	
tile_idx_delta_minus1[i]	ue(v)
if (i>0)	
tile_idx_delta_sign[i]	u(1)
}	
}	
}	

[0815] 在表22中,语法exp_num_slices_in_tile_minus1[i]表示包括第i条带的并行块中的高度信息被显式地用信号发送的条带的数量。具体地,可通过从高度信息被显式地用信号发送的条带的数量减去1来推导语法exp_num_slices_in_tile_minus1[i]。

[0816] 变量numExpSlicesInTile表示通过将语法exp_num_slices_in_tile_minus1[i]

的值加1而推导出的显式数量。

[0817] 当确定了高度信息被显式地用信号发送的条带的数量时,条带的高度信息可根据所确定的数量被编码并被用信号发送。在示例中,语法slice_height_in_ctu_minus1[i][j]表示从包括第i条带的并行块中的第j条带所包括的编码树单元行的数量减去1得到的值。在这种情况下,变量i是基于画面中的条带的数量计算出的值,并且变量j是基于并行块中的条带的数量计算出的值。

[0818] 可基于以比特流用信号发送的语法slice_height_in_ctu_minus1[k]来确定索引k小于显式数量的条带的高度。

[0819] 可基于最后被用信号发送的高度信息和并行块中的其余编码树单元行的数量来确定索引k等于或大于显式数量的条带的高度。在示例中,当通过与最后被用信号发送的高度信息相关的语法slice_height_in_ctu_minus1[exp_num_slices_in_tile_minus1]推导出的条带的宽度是LastSliceHeight,并且除了由并行块中的先前条带占据的区域之外的其余编码树单元行的数量是remainingHeightInCtbsY时,可将第k条带的高度设置为LastSliceHeight和remainingHeightInCtbsY之中的较小值。

[0820] 在表22中,示出了对通过从高度信息被显式地用信号发送的条带的数量减去1而推导出的语法num_exp_slices_in_tile_minus1[i]进行编码/解码。在这种情况下,由语法num_exp_slices_in_tile_minus1[i]确定的显式数量至少为1或更大。因此,即使当一个并行块利用一个条带来配置时,也可能出现应显式地用信号发送并行块的高度的问题。

[0821] 为了解决所述问题,可对被设置为高度信息被显式地用信号发送的条带的数量的语法num_exp_slices_in_tile[i]进行编码/解码。表23示出包括语法num_exp_slices_in_tile[i]的语法表。

[0822] 【表23】

[0823]

rect_slice_flag	u(1)
if(rect_slice_flag)	

	single_slice_per_subpic_flag	u(1)
	if(rect_slice_flag && !single_slice_per_subpic_flag) {	
	num_slices_in_pic_minus1	ue(v)
	tile_idx_delta_present_flag	u(1)
	for(i = 0; i < num_slices_in_pic_minus1; i++) {	
	slice_width_in_tiles_minus1[i]	ue(v)
	slice_height_in_tiles_minus1[i]	ue(v)
	if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] ==	
[0824]	0) {	
	num_exp_slices_in_tile[i]	ue(v)
	numExpSlicesInTile = num_exp_slices_in_tile[i]	
	for(j = 0; j < numExpSlicesInTile; j++)	
	slice_height_in_ctu_minus1[i][j]	ue(v)
	i += (RemNumSlicesMinus1+ numExpSlicesInTile)	
	}	
	if(tile_idx_delta_present_flag && i <	
	num_slices_in_pic_minus1)	
	tile_idx_delta[i]	se(v)
	}	
	}	

[0825] 仅当显式数量 (即, numExpSlicesInTile) 大于0时, 可对表示条带的高度的语法 slice_height_in_ctu_minus1 进行编码/解码。

[0826] 当显式数量为0时, 可确定对应并行块利用一个条带来配置。

[0827] 图43和图44是用于描述条带的高度信息的编码方面的示意图。

[0828] 在图43中, 示出了一个并行块被分区为四个条带。在所示例中, 示出了除了第一条带之外的3个条带具有相同高度 (即, 三个编码树单元行)。

[0829] 在这种情况下, 表示并行块中的第一条带 (条带0) 的高度的语法 slice_height_in_ctu_minus1[0] 可按值2被编码, 并且表示第二条带 (条带1) 的高度的语法 slice_height_in_ctu_minus1[1] 可按值1被编码。由于第三条带 (条带2) 和第四条带 (条带3) 具有与最后通过最后被编码的语法 slice_height_in_ctu_minus1[1] 编码的高度相同的高度, 因此它们的高度信息可不被进一步编码。

[0830] 因此, 高度信息被显式地用信号发送的条带的数量可被确定为2。例如, 语法 num_exp_slices_in_tile[i] 的值可被设置为2。

[0831] 在图44中,示出了一个并行块被分区为五个条带。在所示示例中,示出了第一条带利用三个编码树单元行来配置,第二条带至第四条带利用三个编码树单元行来配置,并且最后条带利用一个编码树单元行来配置。

[0832] 在这种情况下,表示并行块中的第一条带(条带0)的高度的语法slice_height_in_ctu_minus1[0]可按值2被编码,并且表示第二条带(条带1)的高度的语法slice_height_in_ctu_minus1[1]可按值1被编码。第三条带(条带2)和第四条带(条带3)具有与最后通过语法slice_height_in_ctu_minus1[1]编码的高度相同的高度。此外,对于最后条带(条带4),并行块中的除了由先前条带占据的区域之外的其余高度remainingHeightInCtbsY具有比最后通过语法slice_height_in_ctu_minus1[1]编码的高度值LastSliceHeight小的值。因此,虽然第三条带、第四条带和第五条带的高度信息未被单独编码,但是可推导它们中的每一个的高度。

[0833] 因此,高度信息被显式地用信号发送的条带的数量可被确定为2。例如,语法num_exp_slices_in_tile[i]的值可被设置为2。

[0834] 在表22和表23的示例中,变量i被设置在0与从画面中的条带的数量减去1得到的值之间,并且变量j被设置在0与从并行块中的条带的数量减去1得到的值之间。

[0835] 变量i与画面中的条带的总数相关,因此如果预定并行块中的n个条带的分区结构被确定,则当确定关于下一并行块的条带的尺寸时,变量i应增加(n-1)。

[0836] 因此,当确定包括预定并行块中的多个条带时,可在确定了所述多个条带中的每个条带的高度之后将变量i改变为预定并行块中的最后条带的索引。

[0837] 在示例中,在表23中,可通过从预定并行块中的条带的总数减去1和由语法num_exp_slices_in_tile确定的显式数量来推导变量RemNumSlicesMinus1。具体地,可如下推导变量RemNumSlicesMinus1。首先,可通过从确定的并行块的高度减去高度信息被显式地用信号发送的条带的高度之和来推导其余编码树单元行的数量(RemHeight)。随后,从基于其余编码树单元行的数量RemHeight和最后被用信号发送的高度LastSliceHeight计算出的Ceil(RemHeight/LastSliceHeight)减去1得到的值可被设置为变量RemNumSliceMinus1的值。

[0838] 在示例中,在图43的示例中,并行块的高度为9(9个CTU)并且高度信息被显式地用信号发送的条带的高度之和为5。因此,其余编码树行的数量RemHeight被设置为4。此外,高度信息最后被用信号发送的条带的高度LastSliceHeight为2。因此,可通过从将变量RemHeight除以变量LastSliceHeight得到的值减去1来将变量RemNumSliceMinus1的值推导为1。

[0839] 当变量RemHeight不是变量LastSliceHeight的倍数时,通过除法推导出的商可被设置为变量RemNumSliceMinus1。在示例中,在图44的示例中,变量RemHeight是5,并且变量LastSliceHeight是2。因此,变量RemNumSlice可被设置为在变量RemHeight除以变量LastSliceHeight时的商2。

[0840] 表24表示推导变量RemNumSlicesMinus1的示例。

[0841] 【表24】

[0842]

```

tileIdx = 0
for( i = 0; i <= num_slices_in_pic_minus1; i++ )
    NumCtuInSlice[ i ] = 0
for( i = 0; i <= num_slices_in_pic_minus1; i++ ) {
    tileX = tileIdx % NumTileColumns
    tileY = tileIdx / NumTileColumns
    if( i == num_slices_in_pic_minus1 ) {
        slice_width_in_tiles_minus1[ i ] = NumTileColumns - 1 - tileX
        slice_height_in_tiles_minus1[ i ] = NumTileRows - 1 - tileY
        num_slices_in_tile_minus1[ i ] = 0
    }
    if( slice_width_in_tiles_minus1[ i ] == 0 &&
slice_height_in_tiles_minus1[ i ] == 0 ) {
        ctbY = tileRowBd[ tileY ]
        SumExpCtuHeight = 0 ;
        numExpSlicesInTile = num_exp_slices_in_tile[ i ]
    }
}

```

[0843]

```

        for( j = 0; j < numExpSlicesInTile; j++ ) {
            AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
                           ctbY, ctbY + slice_height_in_ctu_minus1[ i ] + 1 )
            ctbY += slice_height_in_ctu_minus1[ i ] + 1
            SumExpCtuHeight += slice_height_in_ctu_minus1[ i ] + 1
            i++
        }
        RemNumSlicesMinus1 = (RowHeight[TileY] - SumExpCtuHeight+
slice_height_in_ctu_minus1[i-1]) / (slice_height_in_ctu_minus1[i-1] + 1)
        for( j = 0; j < RemNumSlicesMinus1; j++ ) {
            AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
                           ctbY, ctbY + slice_height_in_ctu_minus1[ i-1 ] + 1 )
            ctbY += slice_height_in_ctu_minus1[ i-1 ] + 1
        }
    } else
        for( j = 0; j <= slice_height_in_tiles_minus1[ i ]; j++ )
            for( k = 0; k <= slice_width_in_tiles_minus1[ i ]; k++ )
                AddCtbsToSlice( i, tileColBd[ tileX + k ],
tileColBd[ tileX + k + 1 ],
                               tileRowBd[ tileY + j ], tileRowBd[ tileY + j + 1 ] )
            if( tile_idx_delta_present_flag )
                tileIdx += tile_idx_delta[ i ]
            else {
                tileIdx += slice_width_in_tiles_minus1[ i ] + 1
                if( tileIdx % NumTileColumns == 0 )
                    tileIdx += slice_height_in_tiles_minus1[ i ] * NumTileColumns
            }
        }
    }
}

```

[0844] 可选地,如在表25的示例中,可推导变量RemNumSlicesMinus1。

[0845] 【表25】

[0846] tileIdx = 0

[0847]

```

for( i = 0; i  <=  num_slices_in_pic_minus1; i++ )
    NumCtuInSlice[ i ] = 0
for( i = 0; i  <=  num_slices_in_pic_minus1; i++ ) {
    tileX = tileIdx % NumTileColumns
    tileY = tileIdx / NumTileColumns
    if( i  ==  num_slices_in_pic_minus1 ) {
        slice_width_in_tiles_minus1[ i ] = NumTileColumns - 1 - tileX
        slice_height_in_tiles_minus1[ i ] = NumTileRows - 1 - tileY    }
    if( (multiple_slice_in_tile_flag[i] && multiple_slice_in_tile_present_flag) {
(29)
        ctbY = tileRowBd[ tileY ]
        SumExpCtuHeight = 0 ;
        numExpSlicesInTile = num_exp_slices_in_tile_minus1[ i ]+1
        for( j = 0; j < numExpSlicesInTile; j++ ) {
            AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
                           ctbY, ctbY + slice_height_in_ctu_minus1[ i ] + 1 )
            ctbY  +=  slice_height_in_ctu_minus1[ i ] + 1
            SumExpCtuHeight += slice_height_in_ctu_minus1[ i ] + 1
            i++
        }
        RemNumSlicesMinus1 = (RowHeight[TileY] - SumExpCtuHeight+
slice_height_in_ctu_minus1[i-1]) / (slice_height_in_ctu_minus1[i-1] + 1)
        for( j = 0; j < RemNumSlicesMinus1; j++ ) {
            AddCtbsToSlice( i, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
                           ctbY, ctbY + slice_height_in_ctu_minus1[ i-1 ] + 1 )
            ctbY += slice_height_in_ctu_minus1[ i-1 ] + 1
        }
    }
}

```

[0848] 一个画面可被分区为至少一个或更多个子画面。子画面可以是包括至少一个条带的矩形区域。

[0849] 可以以子画面为单位执行编码或解码。可选地,可按每个子画面生成部分比特流。在解码器中,可仅对复用比特流的部分进行解析。在示例中,可对与用户的观看区域对应的部分比特流进行解析,并且基于此,可渲染图像。

[0850] 可选地,可按每个子画面对画面参数集进行编码/解码。因此,按每个子画面参考的画面参数集可不同。可选地,可按每个子画面独立地执行并行块分区。

[0851] 子画面可利用连续区域来配置。这表示子画面不能利用在空间上彼此不相邻的两个条带来配置。例如,彼此邻近的条带可被定义为一个子画面,而彼此不邻近的条带可不被定义为一个子画面。

[0852] 子画面可被定义为包括一个或更多个条带的矩形形状。换言之,子画面具有非矩形形状的分区分形状不被允许。

[0853] 图45是用于描述可被应用于画面的分区分形状的示意图。

[0854] 在图45(a)至(c)中,每个四边形表示条带。此外,分配给每个条带的数字表示对应条带所属的子画面的索引。

[0855] 通过条带头,可用信号发送每个条带所属的子画面的索引。在图45的示例中,可理解的是,被分配了相同索引的条带被包括在同一子画面中。

[0856] 参照图45(a),示出了每个子画面被定义为矩形形状。因此,可将与图45(a)中所示的示例中相同的分区分形状应用于画面。

[0857] 另一方面,在图45(b)中,示出了标记为数字3的子画面被定义为非矩形形状。由于不允许将子画面定义为非矩形形状,因此不能将与图45(b)中所示的示例中相同的分区分形状应用于画面。

[0858] 在图45(c)中,示出了标记为数字0的子画面被定义为在空间上分离的两个区域。不允许一个子画面包括空间上不连续的多个区域。因此,不能将与图45(c)中所示的示例中相同的分区分形状应用于画面。

[0859] 基于所述描述,将根据本公开的实施例详细描述将画面分区分至少一个子画面的方法。

[0860] 图46是根据本公开的实施例的将画面分区分至少一个子画面的方法的流程图。

[0861] 与子画面相关的信息可在序列级被用信号发送。在示例中,表示下面将描述的子画面是否被分区的语法、与画面分区分信息相关的语法或与子画面独立性相关的语法中的至少一个可被包括在序列参数集中。

[0862] 与子画面相关的信息可被共同应用于参考序列参数集的画面。因此,参考序列参数集的画面的分区分形状可以是相同的。

[0863] 可选地,与子画面相关的信息中的一些信息可在序列级被用信号发送,并且一些信息可在画面级被用信号发送。在示例中,表示子画面分区分是否被执行的语法可被包括在序列参数集中,而与画面分区分信息相关的语法和与子画面独立性相关的语法可被包括在画面参数集中。可选地,表示子画面分区分是否被执行的语法和与画面分区分信息相关的语法可被包括在序列参数集中,而与子画面独立性相关的语法可被包括在画面参数集中。在这种情况下,分区分形状或子画面的独立性中的至少一个可按每个画面而不同。

[0864] 在下面描述的实施例中,假设通过序列参数集用信号发送与子画面相关的信息。

[0865] 参照图46,可确定画面是否将被分区分至少一个子画面(S4601)。可以以比特流用信号发送表示画面是否将被分区分至少一个子画面的信息。在示例中,可以以比特流用信号发送语法subpics_present_flag。当语法subpics_present_flag为1时,这表示画面可被分区分至少一个子画面。当语法subpics_present_flag为1时,子画面参数可被包括在比特流(例如,序列参数集)中。另一方面,当语法subpics_present_flag为0时,这表示画面不被分区分子画面。当语法subpics_present_flag为0时,子画面参数可不被包括在比特流(例

如,序列参数集)中。在这种情况下,子画面参数可包括表示子画面的数量的信息(例如,max_subpics_minus1)、表示子画面的尺寸的信息(例如,subpic_grid_col_width_minus1、subpic_grid_row_height_minus1)、表示子画面的索引的信息(例如,subpic_grid_idx)、表示子画面是否被视为画面的信息(例如,subpic_treated_as_pic_flag)或表示是否允许对子画面的边界应用环内滤波器的信息(例如,loop_filter_across_subpic_enabled_flag)中的至少一个。在下文中,详细描述子画面参数。

[0866] 当画面可被分区为至少一个子画面时,可获得子画面分区信息(S4602)。画面分区信息可包括表示画面中包括的子画面的数量的信息、表示每个子画面的位置的信息或表示每个子画面的尺寸的信息中的至少一个。

[0867] 在示例中,可以以比特流用信号发送表示子画面的数量的语法sps_num_subpics_minus1。语法sps_num_subpics_minus1可表示从画面中包括的子画面的数量减去1得到的值。

[0868] 可选地,可以以比特流用信号发送表示子画面的数量的语法max_subpics_minus1。当针对序列中的每个画面独立地执行子画面分区时,语法max_subpics_minus1表示参考序列参数集的每个画面的子画面的数量的最大值。在示例中,当序列中的第一画面所包括的子画面的数量是3并且除了第一画面之外的其余画面包括2个或更少的子画面时,语法max_subpics_minus1的值可被设置为2(从最大值3减去1得到的值)。

[0869] 可基于正方形块确定子画面的尺寸。正方形块可以是通过将网格结构应用于画面而推导出的预定义尺寸的块。成为用于确定子画面的尺寸的基本单元的正方形块可被称为网格。

[0870] 可在编码器和解码器中预定义网格的尺寸。在示例中,网格可具有 8×8 、 16×16 、 32×32 或 64×64 尺寸。可选地,可以以比特流用信号发送表示网格的尺寸的信息。

[0871] 可选地,可将单个编码树单元或多个编码树单元设置为网格。可在序列级用信号发送编码树单元的尺寸。换言之,可对包括在序列参数集中的编码树单元尺寸信息进行解析,并且可通过使用经过解析的编码树单元尺寸信息来确定子画面的位置和尺寸。

[0872] 可通过将每个网格所属的子画面的索引分配给每个网格来定义子画面。

[0873] 表26表示子画面的索引被分配给每个网格的示例。

[0874] **【表26】**

[0875]	seq_parameter_set_rbsp(){	描述符
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag){	
	max_subpics_minus1	u(8)
	subpic_grid_col_width_minus1	u(v)
	subpic_grid_row_height_minus1	u(v)
	for(i=0; i<NumSubPicGridRows; i++)	
	for(j=0; j<NumSubPicGridCols; j++)	
	subpic_grid_idx[i][j]	u(v)
	for(i=0; i<=NumSubPics; i++)	
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
[0876]	}	
	}	
	...	
	}	

[0877] 语法subpic_grid_col_width_minus1表示网格的宽度。在示例中,可通过从将网格的宽度除以预定常数值N得到的值减去1来推导语法subpic_grid_col_width_muns1。N可以是诸如2、4、8或16等的整数。可选地,可根据编码树单元的尺寸来确定N。

[0878] 语法subpic_grid_row_height_minus1表示网格的高度。在示例中,可通过从将网格的高度除以预定常数值N得到的值减去1来推导语法subpic_grid_row_height_minus1。

[0879] 可将子画面索引分配给每个网格。在示例中,语法subpic_grid_idx[i][j]表示属于第i列第j行的网格的子画面索引。

[0880] 分配了相同的子画面索引的网格可被理解为属于相同的子画面。

[0881] 基于子画面的最大数量,可确定网格尺寸信息是否被用信号发送。在示例中,表27表示基于语法max_subpics_minus1的值来确定是否对网格尺寸信息进行解析的示例。

[0882] 【表27】

[0883]	seq_parameter_set_rbsp() {	描述符
	...	
	subpics_present_flag	u(1)
	if(subpics_present_flag) {	
	max_subpics_minus1	u(8)
	if (max_subpics_minus1>0){	
	subpic_grid_col_width_minus1	u(v)
	subpic_grid_row_height_minus1	u(v)
	}	
	for(i = 0; i < NumSubPicGridRows; i++)	
	for(j = 0; j < NumSubPicGridCols; j++)	
	subpic_grid_idx[i][j]	u(v)
[0884]	for(i = 0; i < NumSubPics; i++) {	
	subpic_treated_as_pic_flag[i]	u(1)
	loop_filter_across_subpic_enabled_flag[i]	u(1)
	}	
	}	
	...	ue(v)
	}	

[0885] 如表27中所示,仅当语法max_subpics_minus1的值大于0时(即,仅当最大子画面的数量等于或大于2时),可用信号发送表示网格尺寸的信息,即语法subpic_grid_col_width_minus1和语法subpic_grid_col_height_minus1。

[0886] 在另一示例中,可基于子画面的位置信息或子画面的尺寸信息中的至少一个来确定子画面的尺寸。

[0887] 子画面的位置信息可包括子画面的水平方向位置信息(即,X轴坐标)和子画面的垂直方向位置信息(即,Y轴坐标)。当多个子画面被包括在画面中时,可针对子画面中的每个子画面用信号发送位置信息。

[0888] 在示例中,可以以比特流用信号发送用于确定子画面的水平方向位置的语法subpic_ctu_top_left_x[i]。语法subpic_ctu_top_left_x[i]表示画面中的第i子画面的水平方向位置。具体地,subpic_ctu_top_left_x[i]表示第i子画面的左上方位位置处的网格(例如,编码树单元)的水平方向位置。在这种情况下,水平方向位置可以是基于编码树单元的尺寸确定的值。在示例中,可通过将第i子画面的左上方位位置处的编码树单元的x坐标除以编码树单元的尺寸(例如,宽度)来推导水平方向位置。因此,可通过将语法subpic_ctu_top_left_x[i]的值乘以编码树单元的尺寸来推导第i编码树单元的左上方位位置处的x坐

标。

[0889] 可以以比特流用信号发送用于确定子画面的垂直方向位置的语法subpic_ctu_top_left_y[i]。语法subpic_ctu_top_left_y[i]表示画面中的第i子画面的垂直方向位置。具体地,subpic_ctu_top_left_y[i]表示第i子画面的左上方位位置处的网格(例如,编码树单元)的垂直方向位置。在这种情况下,垂直方向位置可以是基于编码树单元的尺寸确定的值。在示例中,可通过将第i子画面的左上方位位置处的编码树单元的y坐标除以编码树单元的尺寸(例如,高度)来推导垂直方向位置。因此,可通过将语法subpic_ctu_top_left_y[i]的值乘以编码树单元的尺寸来推导第i编码树单元的左上方位位置处的y坐标。

[0890] 当第一子画面的左上边界邻接画面的左上边界时,对于第一子画面,可省略用信号发送表示子画面的位置的信息的操作。

[0891] 子画面的尺寸信息可包括子画面的宽度信息和子画面的高度信息。当多个子画面被包括在画面中时,可针对子画面中的每个子画面用信号发送尺寸信息。

[0892] 在示例中,可以以比特流用信号发送用于确定子画面的宽度的语法subpic_width_minus1[i]。语法subpic_width_minus1[i]可表示从将画面中的第i子画面的宽度除以网格的尺寸(例如,编码树单元的宽度)得到的值减去1而得到的值。在示例中,当网格的尺寸是 4×4 时,语法subpic_width_minus1[i]可被设置为通过从将第i子画面的宽度除以4得到的结果值减去1而推导出的值。可选地,当网格的尺寸与编码树单元的尺寸相同时,语法subpic_width_minus1[i]可被设置为通过从将第i子画面的宽度除以编码树单元的尺寸得到的结果值减去1而推导出的值。换言之,语法subpic_width_minus1表示从第i子画面所包括的网格列(例如,编码树单元列)的数量减去1得到的值。

[0893] 当网格的尺寸与编码树单元的尺寸相同时,解码器可如在以下等式13中那样推导子画面的宽度。

[0894] 【等式13】

[0895]
$$\text{subpicWidth} = (\text{subpic_width_minus1}[i] + 1) * \text{CtbSize}$$

[0896] 在等式13中,subpicWidth表示第i子画面的宽度,并且CtbSize表示编码树单元的尺寸。在示例中,变量CtbSize可以是通过将编码树单元的宽度和高度的乘积取Log₂而推导出的值。

[0897] 可以以比特流用信号发送用于确定子画面的高度的语法subpic_height_minus1[i]。语法subpic_height_minus1[i]可表示从将画面中的第i子画面的高度除以网格的尺寸(例如,编码树单元的高度)得到的值减去1而得到的值。在示例中,当网格的尺寸是 4×4 时,语法subpic_height_minus1[i]可被设置为通过从将第i子画面的高度除以4得到的结果值减去1而推导出的值。可选地,当网格的尺寸与编码树单元的尺寸相同时,语法subpic_height_minus1[i]可被设置为通过从将第i子画面的高度除以编码树单元的尺寸得到的结果值减去1而推导出的值。换言之,语法subpic_height_minus1表示从第i子画面所包括的网格列(例如,编码树单元列)的数量减去1得到的值。

[0898] 当网格的尺寸与编码树单元的尺寸相同时,解码器可如在以下等式14中那样推导子画面的高度。

[0899] 【等式14】

[0900]
$$\text{subpicHeight} = (\text{subpic_height_minus1}[i] + 1) * \text{CtbSize}$$

[0901] 在等式14中,subpicHeight表示第i子画面的高度。

[0902] 当画面可被分区为至少一个子画面时,可获得子画面独立性信息。子画面独立性信息表示子画面是否可被独立地编码/解码。在示例中,子画面独立性信息可包括表示子画面是否像画面一样被处理的信息或表示是否允许对子画面边界应用环路滤波器的信息中的至少一个。

[0903] 在示例中,在表26和表27的示例中,语法subpic_treated_as_pic_flag[i]表示第i子画面是否被视为画面。当语法subpic_treated_as_pic_flag[i]的值为1时,这表示当对子画面进行不包括环路滤波器处理过程的编码/解码时可独立地对子画面进行编码/解码。

[0904] 当子画面被视为画面时,在不包括环内滤波处理的编码/解码中可不允许参考其他子画面的信息。换言之,子画面的边界可被视为画面边界。在示例中,在推导时间运动预测矢量的处理中或在诸如插值等的编码/解码处理中,可在假设子画面的边界为画面边界的情况下处理编码/解码。

[0905] 在示例中,通常,可基于同位画面中的同位块的运动矢量推导时间运动预测矢量候选或时间合并候选。在这种情况下,同位画面中的包括与当前块具有相同位置和尺寸的块的右下角的坐标的块可被设置为同位块。如果包括右下角的坐标的块不可用,则同位画面中的包括与当前块具有相同位置和尺寸的块的中心坐标的块可被设置为同位块。在推导时间运动预测矢量候选或时间合并候选时,当同位画面中的与当前块具有相同位置和尺寸的块的右下角在画面边界上或在画面边界之外时,可从同位画面中的包括与当前块具有相同位置和尺寸的块的中心坐标的块的运动矢量推导时间运动预测矢量候选或时间合并候选。

[0906] 当子画面未被视为画面时,可基于画面边界而不是子画面边界来确定右下角是否在画面边界上或者右下角是否在画面边界之外。

[0907] 另一方面,当子画面被视为画面时,可基于子画面边界来确定右下角是否在画面边界上或者右下角是否在画面边界之外。在示例中,当同位画面中的与当前块具有相同位置和尺寸的块的右下角在子画面边界上或在子画面边界之外时,同位画面中的包括与当前块具有相同位置和尺寸的块的中心坐标的块可被设置为同位块。

[0908] 此外,通常,在帧间预测中,在对十进制像素进行插值时,存在当前画面和参考画面的整数位置处的亮度分量像素不在画面的左边界、右边界、上边界和下边界之外的限制。当子画面被视为画面时,通过将子画面的边界视为画面边界,可能存在整数位置处的亮度分量像素不在子画面的左边界、右边界、上边界和下边界之外的限制。

[0909] 在表26和表27的示例中,语法loop_filter_across_subpic_enabled_flag[i]表示是否允许在第i子画面中应用环内滤波器。在示例中,当语法loop_filter_across_subpic_enabled_flag[i]的值为1时,这表示允许对第i子画面的边界应用环内滤波器。环内滤波器可包括去块滤波器、SAO或ALF中的至少一个。另一方面,当语法loop_filter_across_subpic_enabled_flag[i]的值为0时,这表示不允许对第i子画面的边界应用环内滤波器。

[0910] 在表26和表27的示例中,变量NumSubPics表示子画面的数量。变量NumSubPics可具有与将语法max_subpics_minus1加1得到的值相同的值或比将语法max_subpics_minus1加1得到的值更小的值。

[0911] 可针对i的值在0至NumSubPics之间的子画面用信号发送子画面独立性信息,例如,subpic_treated_as_pic_flag[i]和loop_filter_across_subpic_enabled_flag[i])。

[0912] 一个条带或多个条带可被设置为网格。在示例中,当条带被设置为网格时,对应条带所属的子画面索引可按每个条带被用信号发送。表28和表29表示当使用矩形形状的条带的定义方法时用信号发送分配给每个条带的子画面信息的示例。

[0913] 【表28】

[0914]	slice_header() {	描述符
	slice_pic_parameter_set_id	ue(v)
	if(rect_slice_flag NumBricksInPic > 1) {	
	slice_address	u(v)
	subpic_grid_idx[slice_address]	u(v)
	}	
[0915]	if(!rect_slice_flag && !single_brick_per_slice_flag)	
	num_bricks_in_slice_minus1	ue(v)
	...	u(1)
	byte_alignment()	
	}	

[0916] 【表29】

[0917]	pic_parameter_set_rbsp() {	描述符
	...	
	if(rect_slice_flag) {	
	signalled_slice_id_flag	u(1)
	if(signalled_slice_id_flag) {	
	signalled_slice_id_length_minus1	ue(v)
	for(i = 0; i <= num_slices_in_pic_minus1; i++){	
	slice_id[i]	u(v)
	subpic_grid_idx[slice_id[i]]	u(v)
	}	
	}	
	}	
	...	u(1)
	rbp_trailing_bits()	
	}	

[0918] 在表28中,语法slice_address表示分配给画面中的第i条带的地址。语法subpic_grid_idx[slice_address]表示分配给地址值为slice_address的条带的子画面索引值。分配了相同的子画面索引的条带可被理解为属于相同的子画面。

[0919] 在表29中,语法slice_id[i]表示分配给画面中的第i条带的标识符。语法subpic_grid_idx[slice_id[i]]表示分配给第i条带(即,标识符为slice_id[i]的条带)的子画面索引值。分配了相同的子画面索引的条带可被理解为属于相同的子画面。

[0920] 仅当基于矩形形状的条带分区方法被应用时(例如,仅当语法rect_slice_flag为1时),可基于条带设置网格。另一方面,当基于光栅扫描顺序的条带分区方法被应用时(例如,当语法rect_slice_flag为0时),可不允许基于条带设置网格。

[0921] 一个并行块或多个并行块可被设置为网格。在示例中,当并行块被设置为网格时,对应并行块所属的子画面索引可按每个并行块被用信号发送。

[0922] 将如关于解码处理或编码处理所描述的实施例分别应用于编码处理或解码处理的操作可被包括在本公开的范围中。在本公开的范围中,操作按照预定顺序发生的实施例可被修改为操作按照与所述预定顺序不同的顺序发生的实施例。

[0923] 虽然基于一系列操作或流程图描述了上述实施例,但是所述实施例不将方法的操作的时间序列顺序限制于此。在另一示例中,操作可根据需要同时执行或按照与其不同的顺序来执行。此外,在上述实施例中,构成框图的组件中的每个组件(例如,单元、模块等)可以以硬件装置或软件的形式来实现。多个组件可被彼此组合为可使用单个硬件装置或软件来实现的单个组件。可使用可经由各种计算机组件执行的程序指令来实现上述实施例。指令可被记录在计算机可读存储介质中。计算机可读存储介质可在其中单独地或彼此组合地包括程序指令、数据文件、数据结构等。计算机可读存储介质的示例包括磁介质(诸如硬盘、

软盘和磁带)、光学存储介质(诸如CD-ROM、DVD)和磁光介质(诸如软光盘)以及被专门配置为在其中存储并执行程序指令的硬件装置(诸如ROM、RAM、闪存等)。硬件装置可被配置为如一个或多个软件模块进行操作以执行根据本公开的处理,反之亦可。

[0924] 【工业实用性】

[0925] 本公开可被应用于对视频进行编码/解码的电子装置。

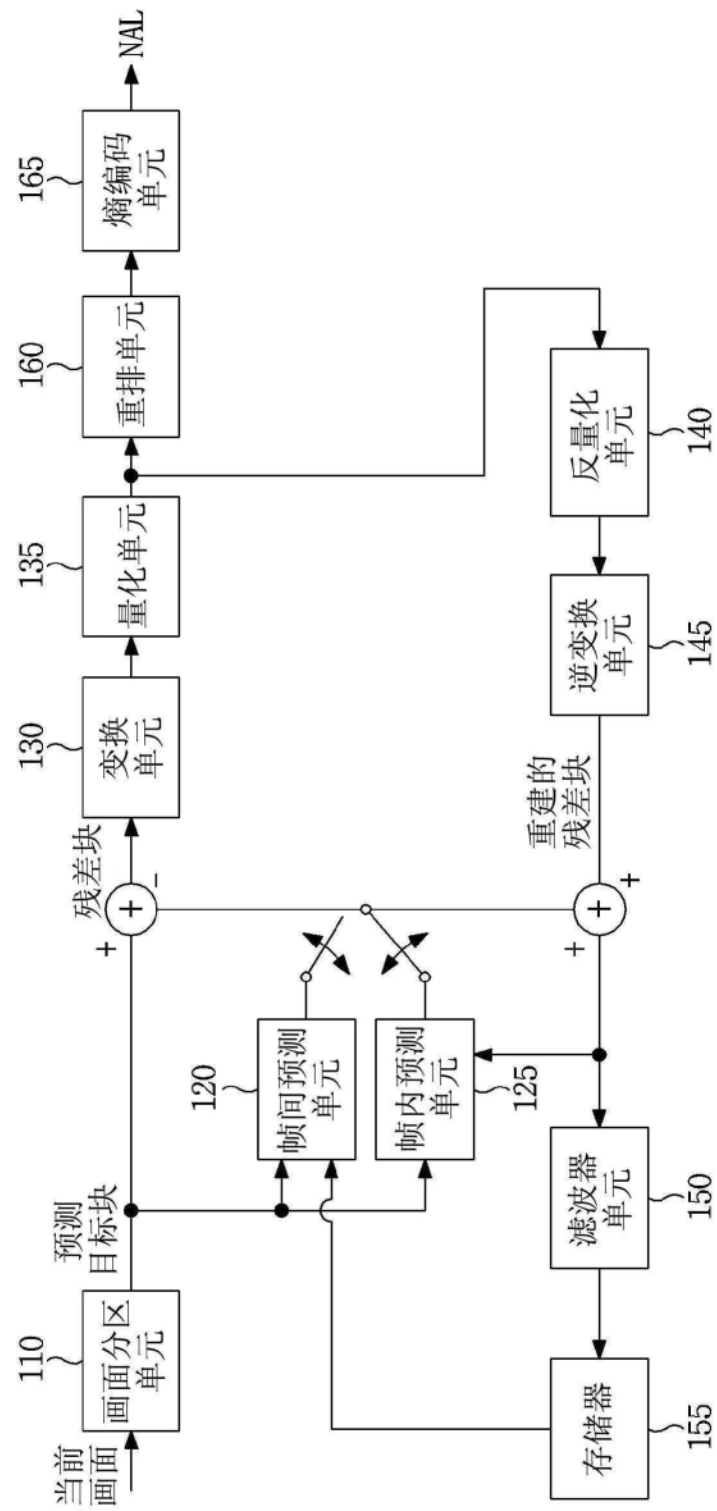


图1

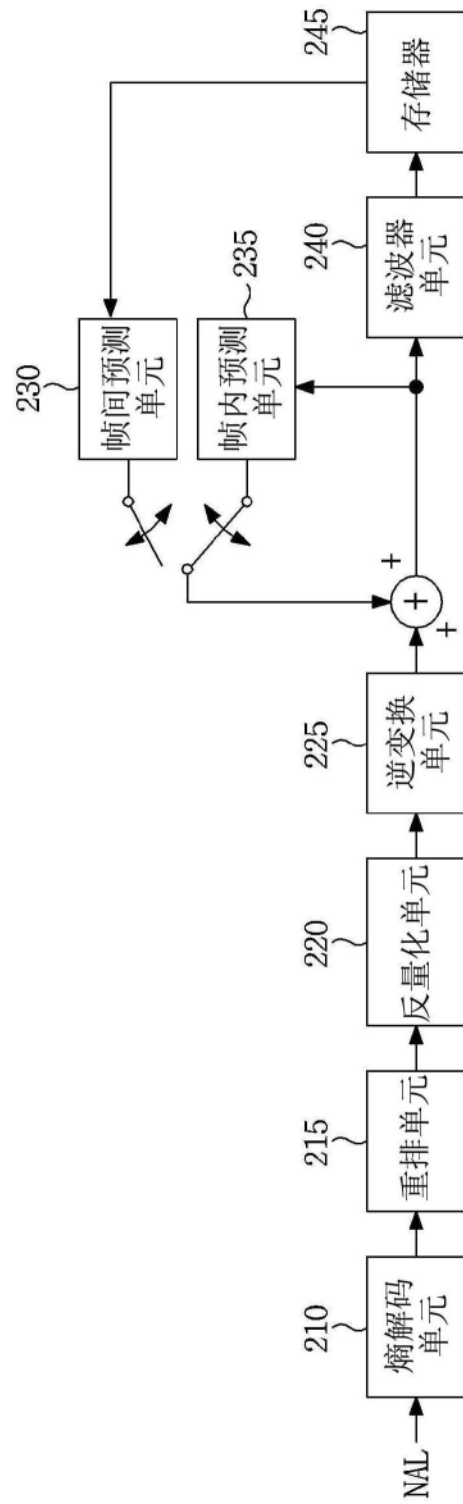


图2

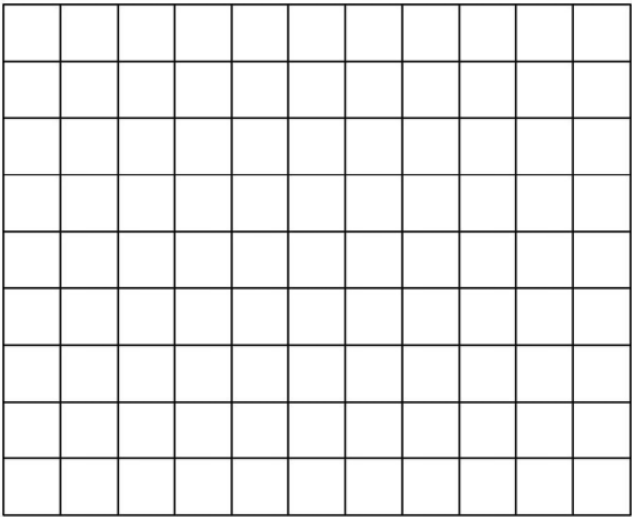


图3

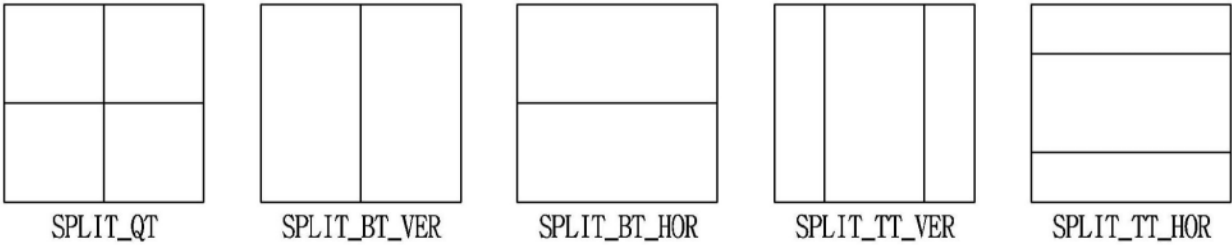


图4

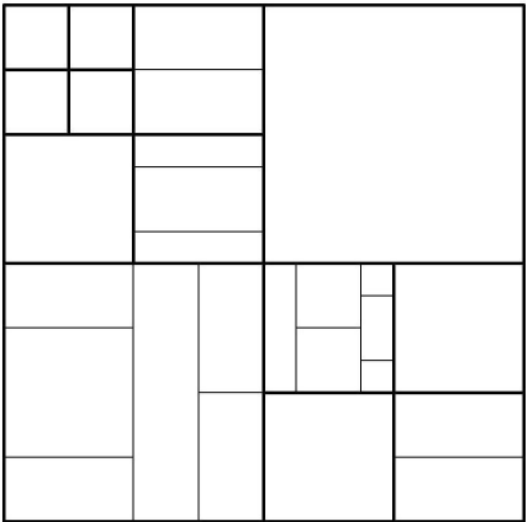


图5

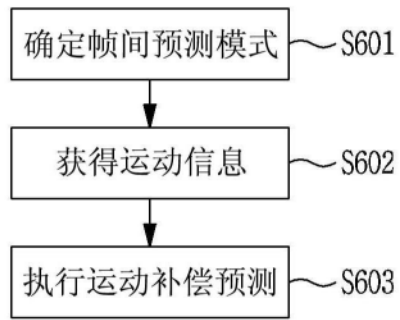


图6

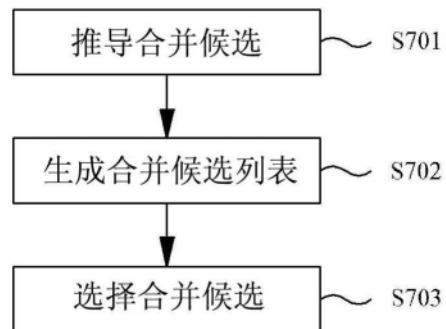


图7

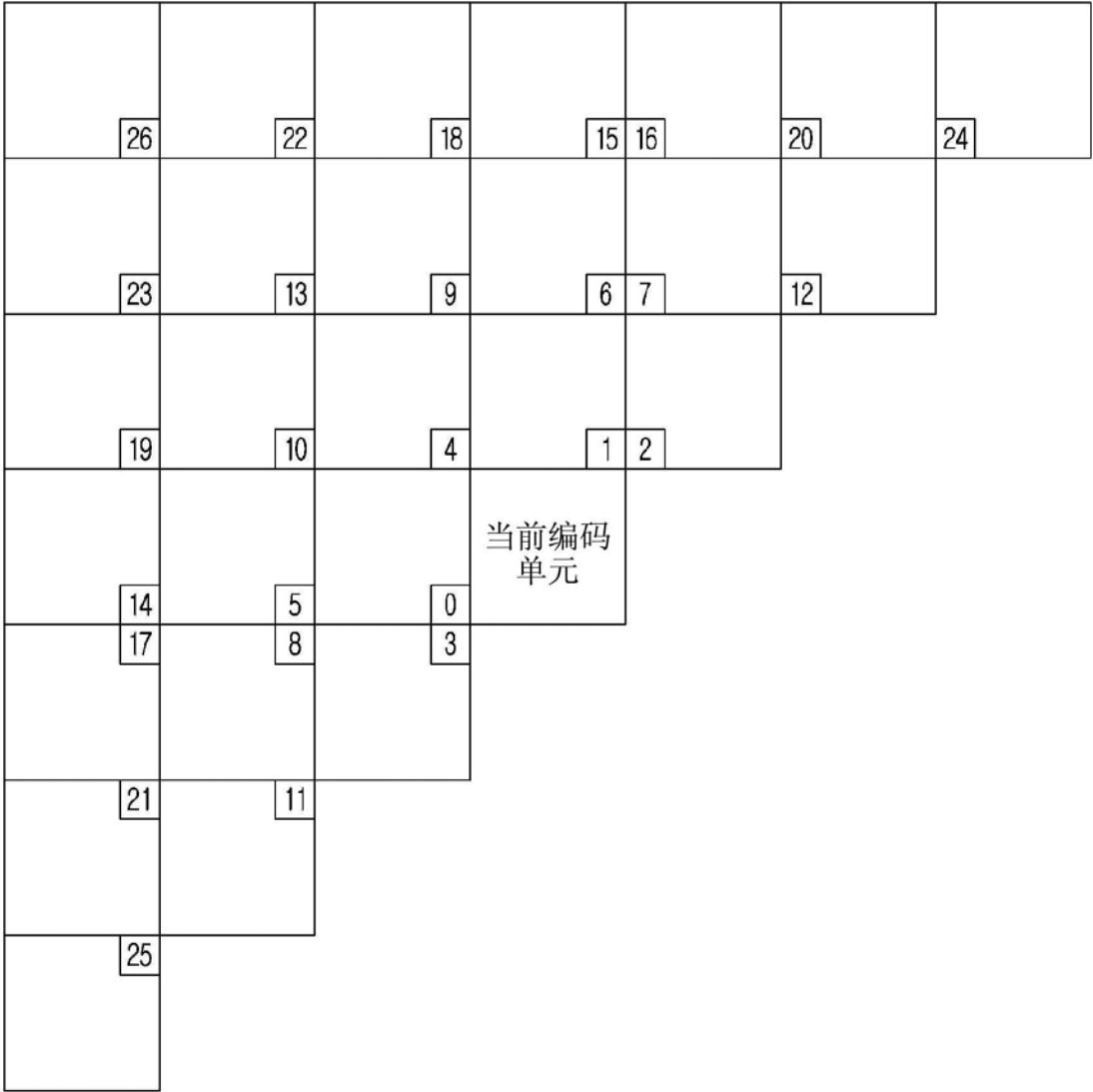


图8

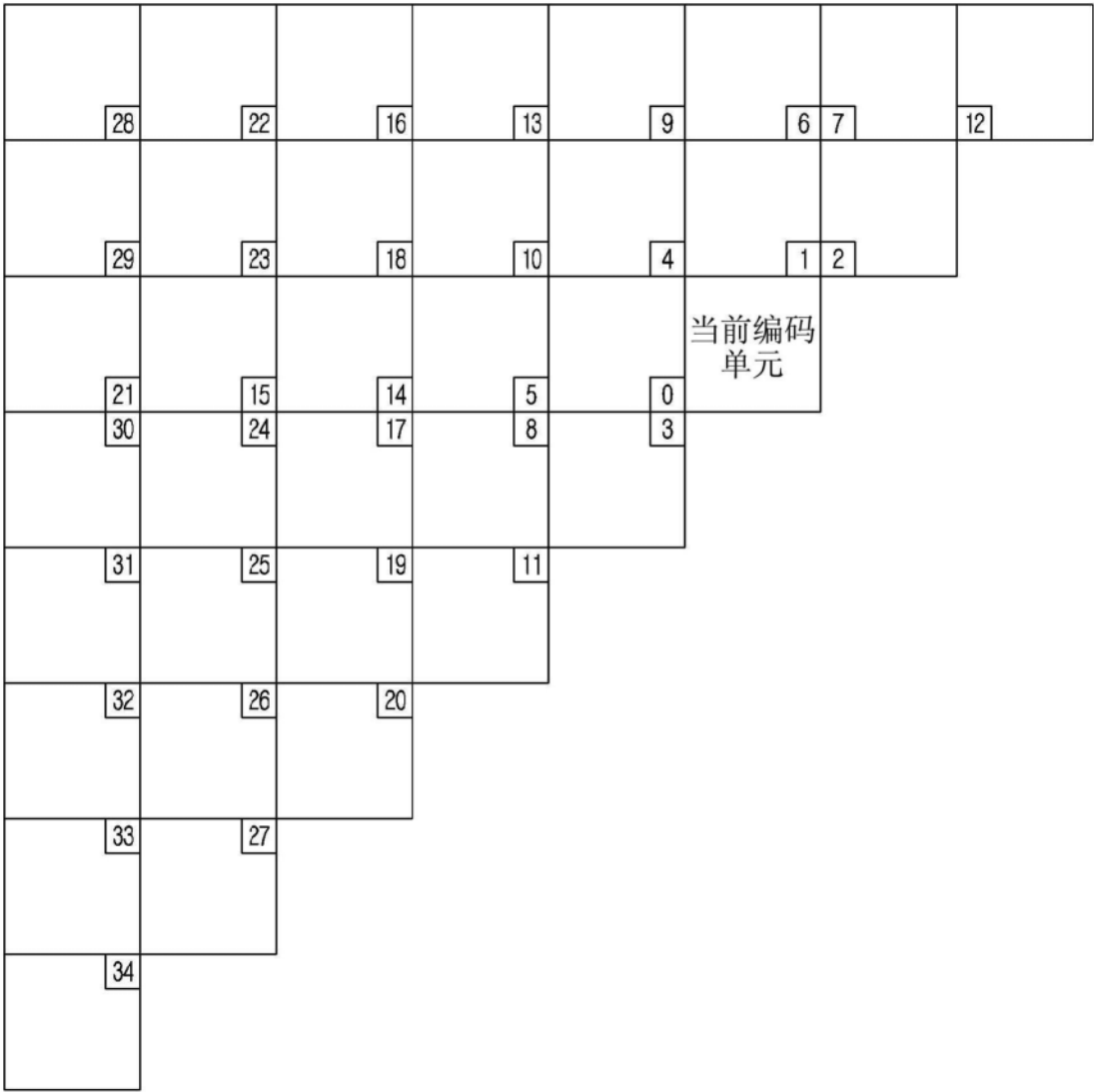


图9

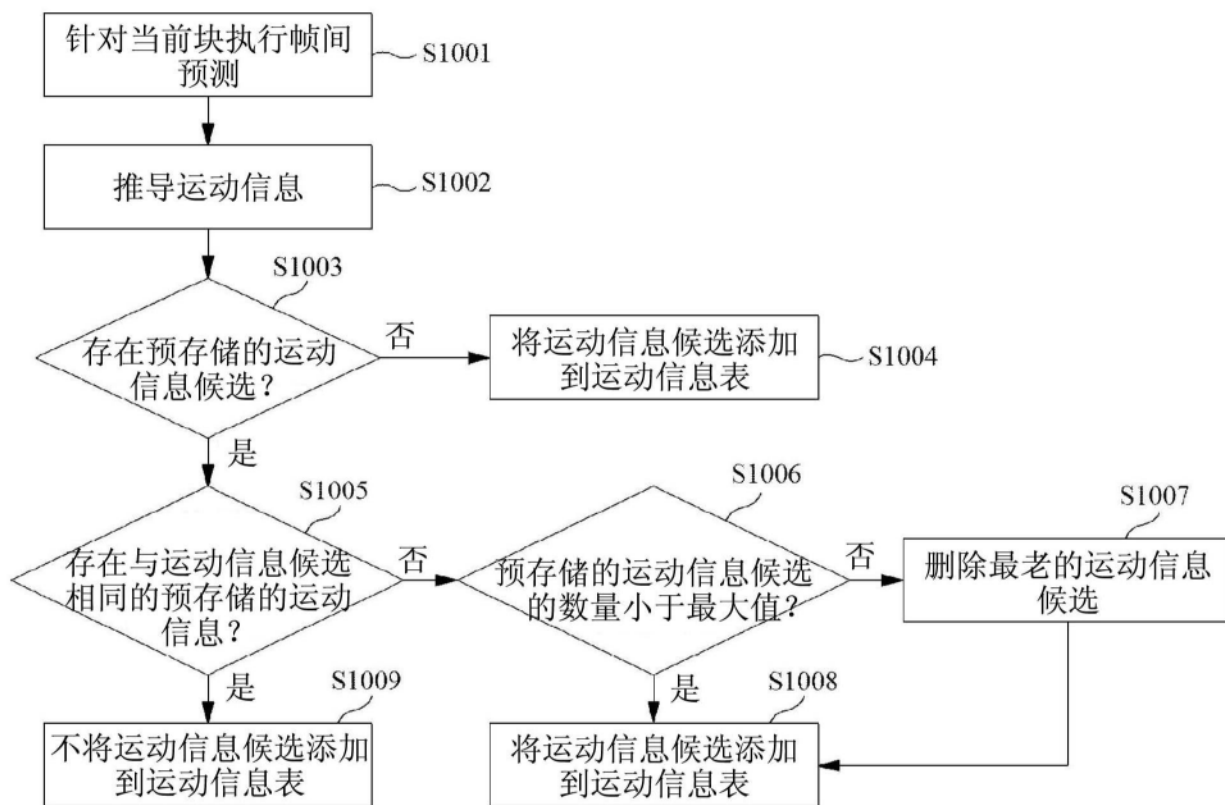


图10

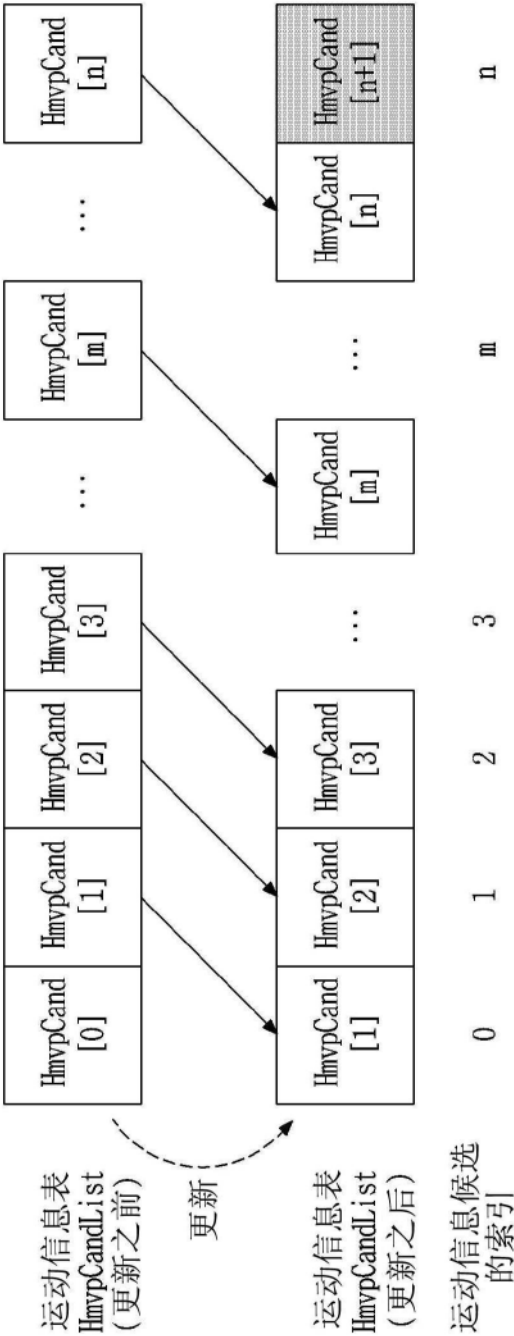


图11

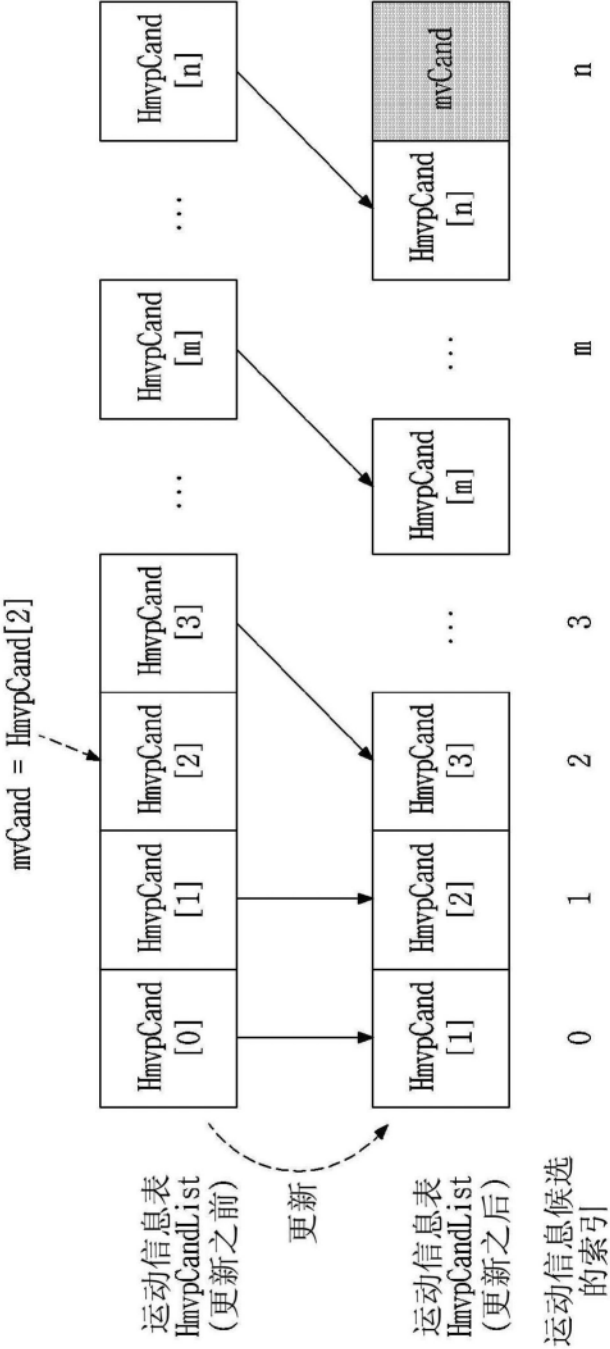


图12

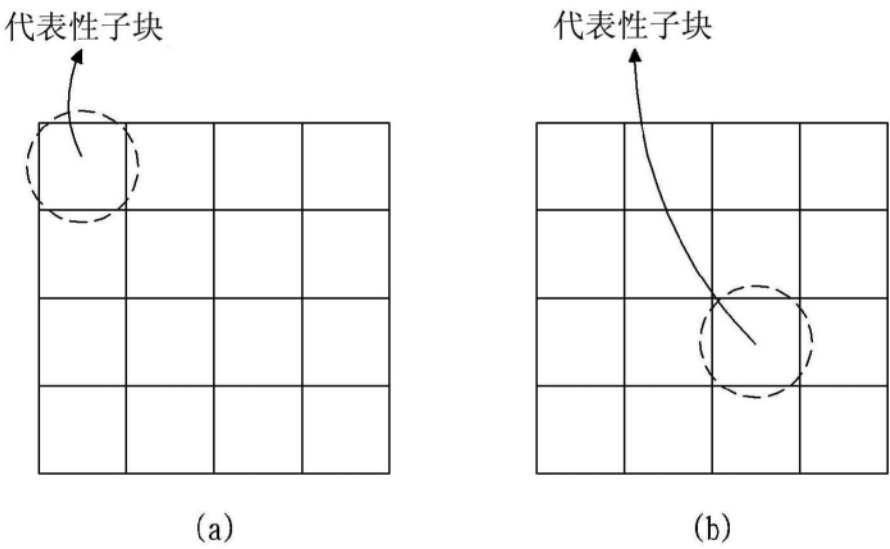


图13

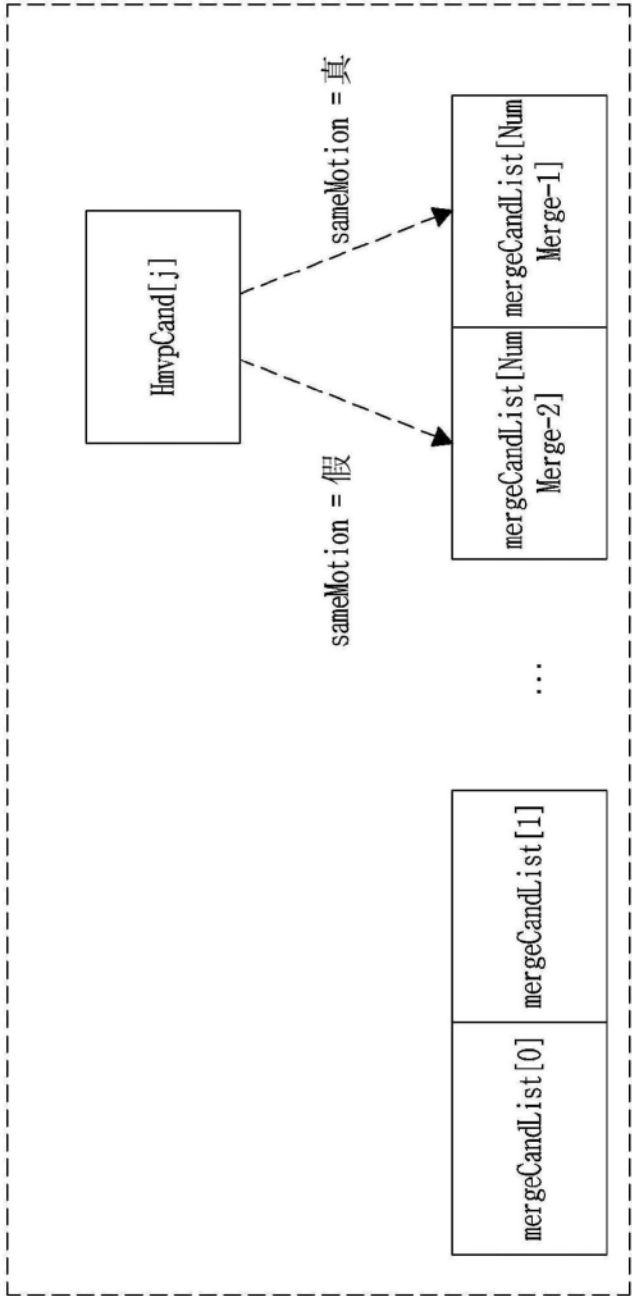


图14

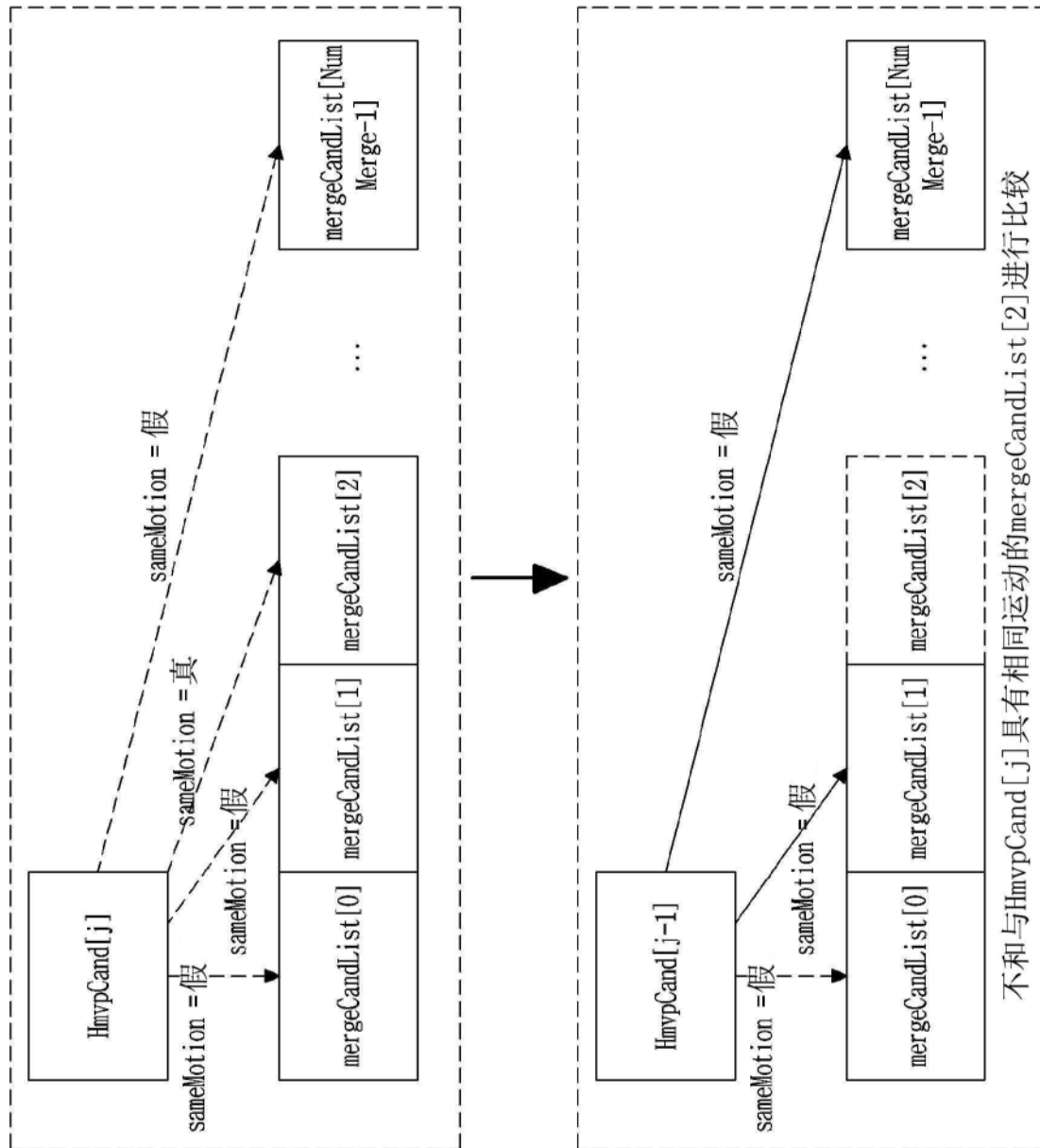


图15

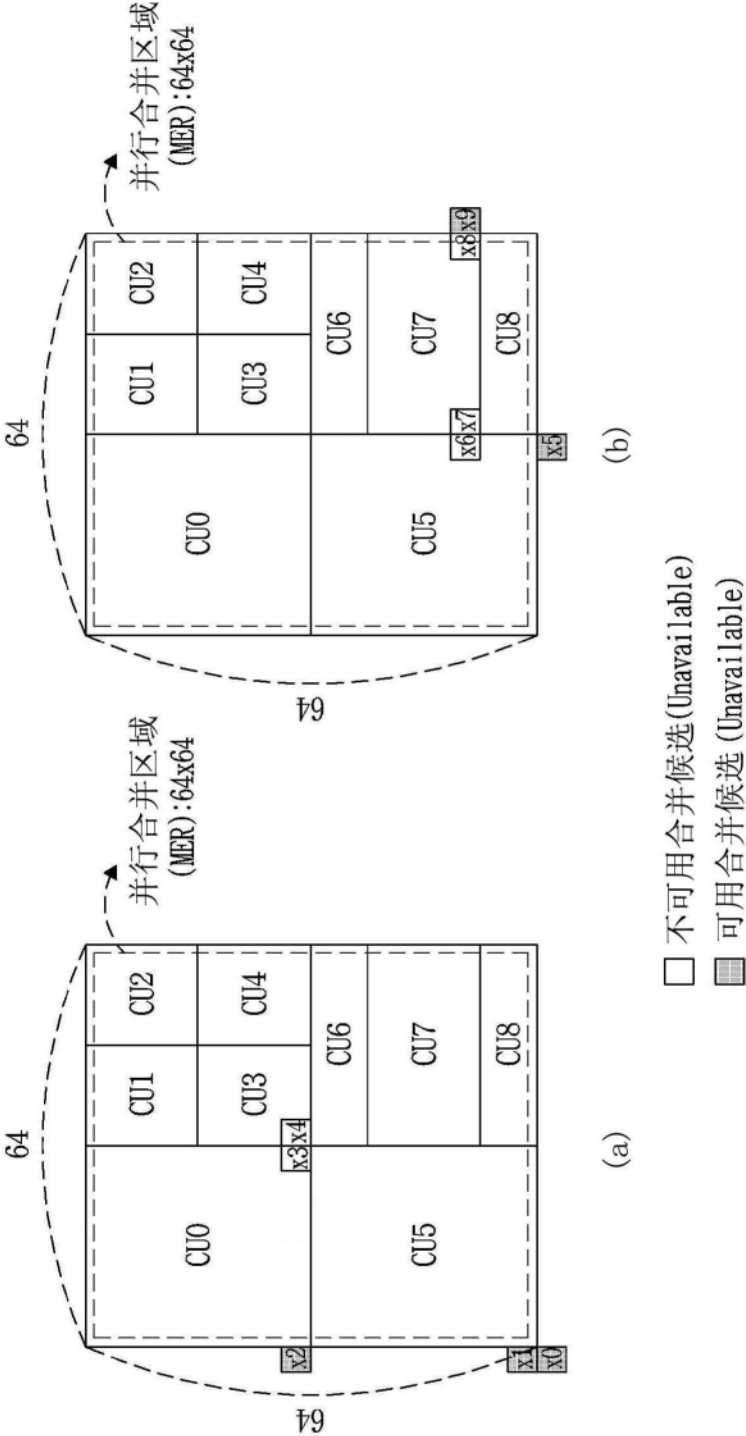


图16

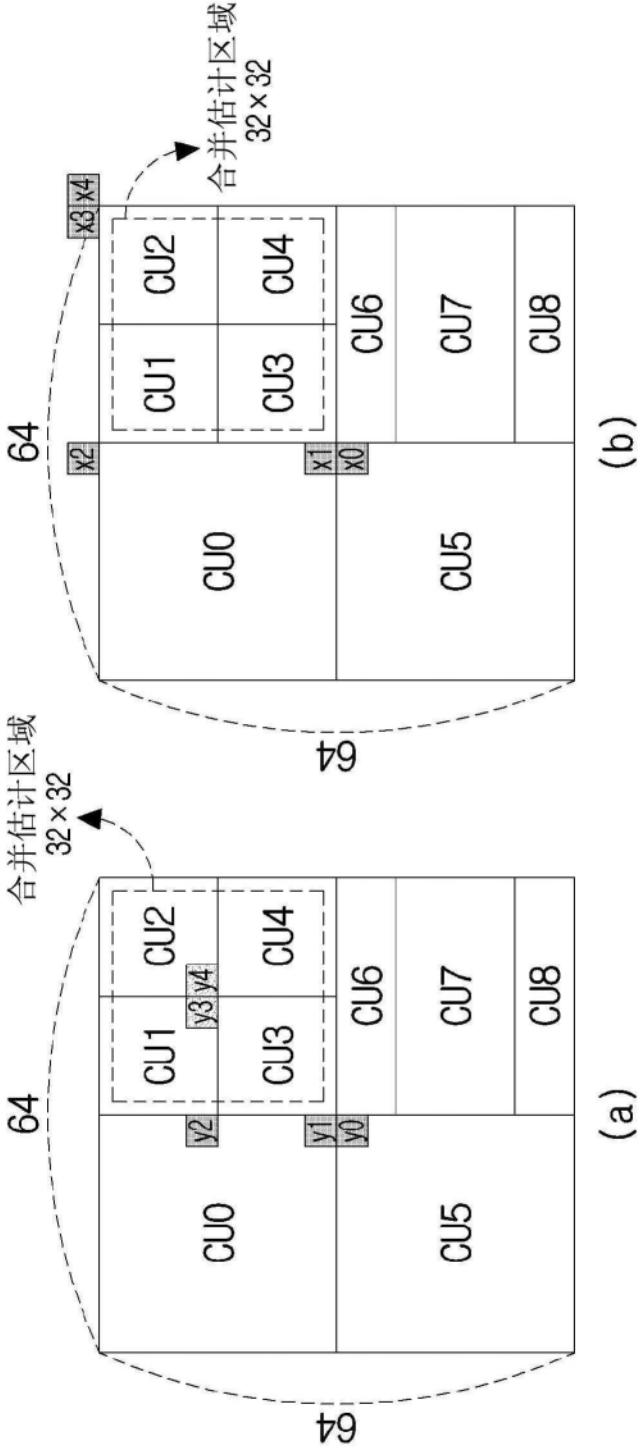


图17

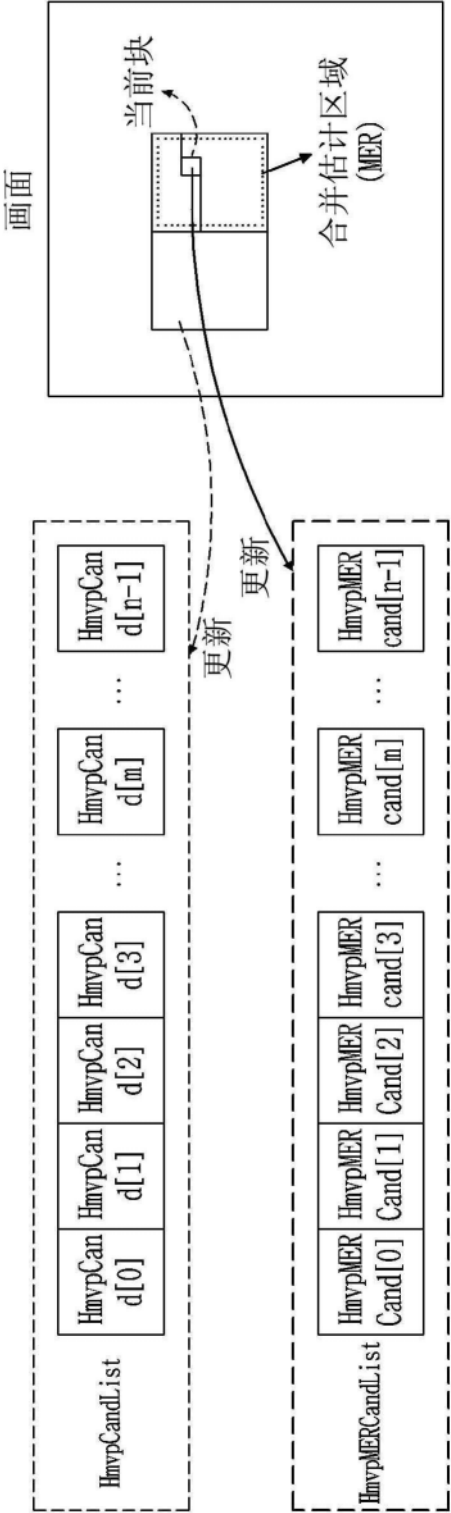


图18

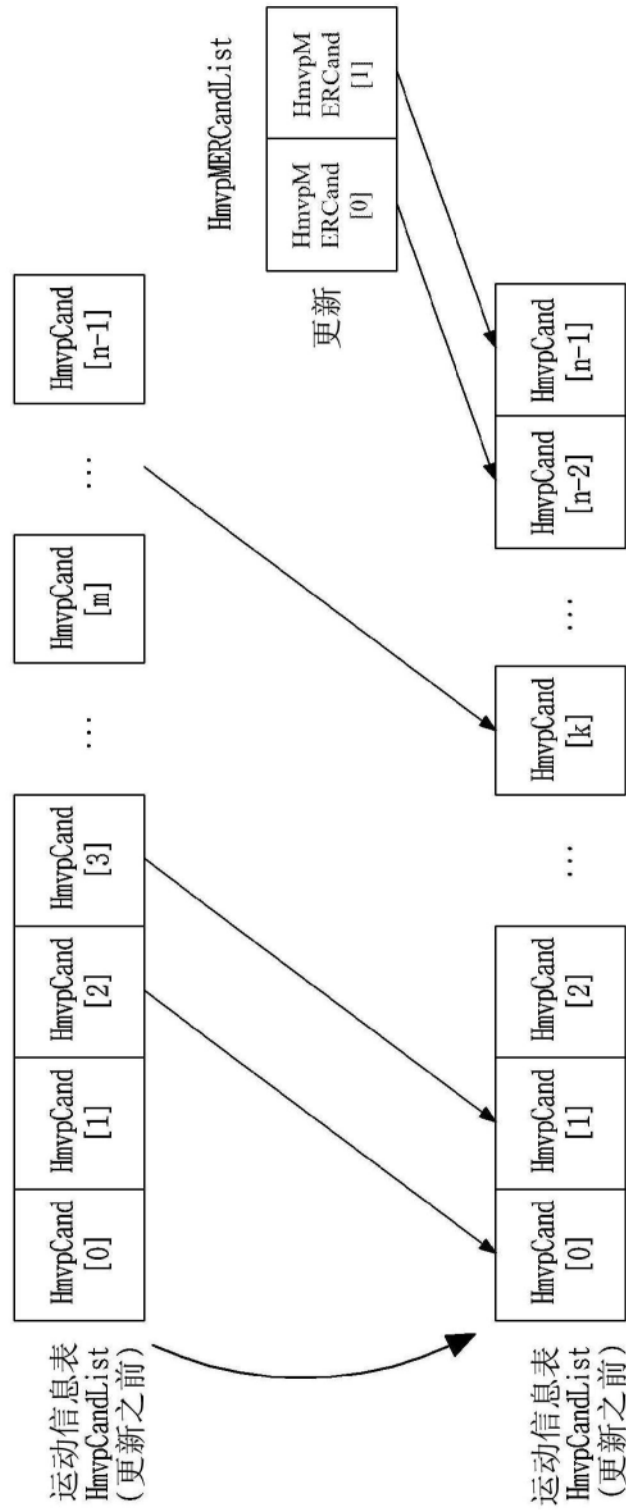


图19

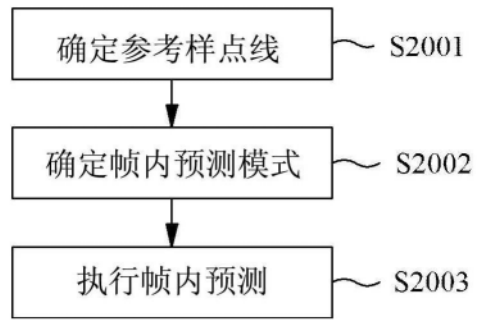


图20

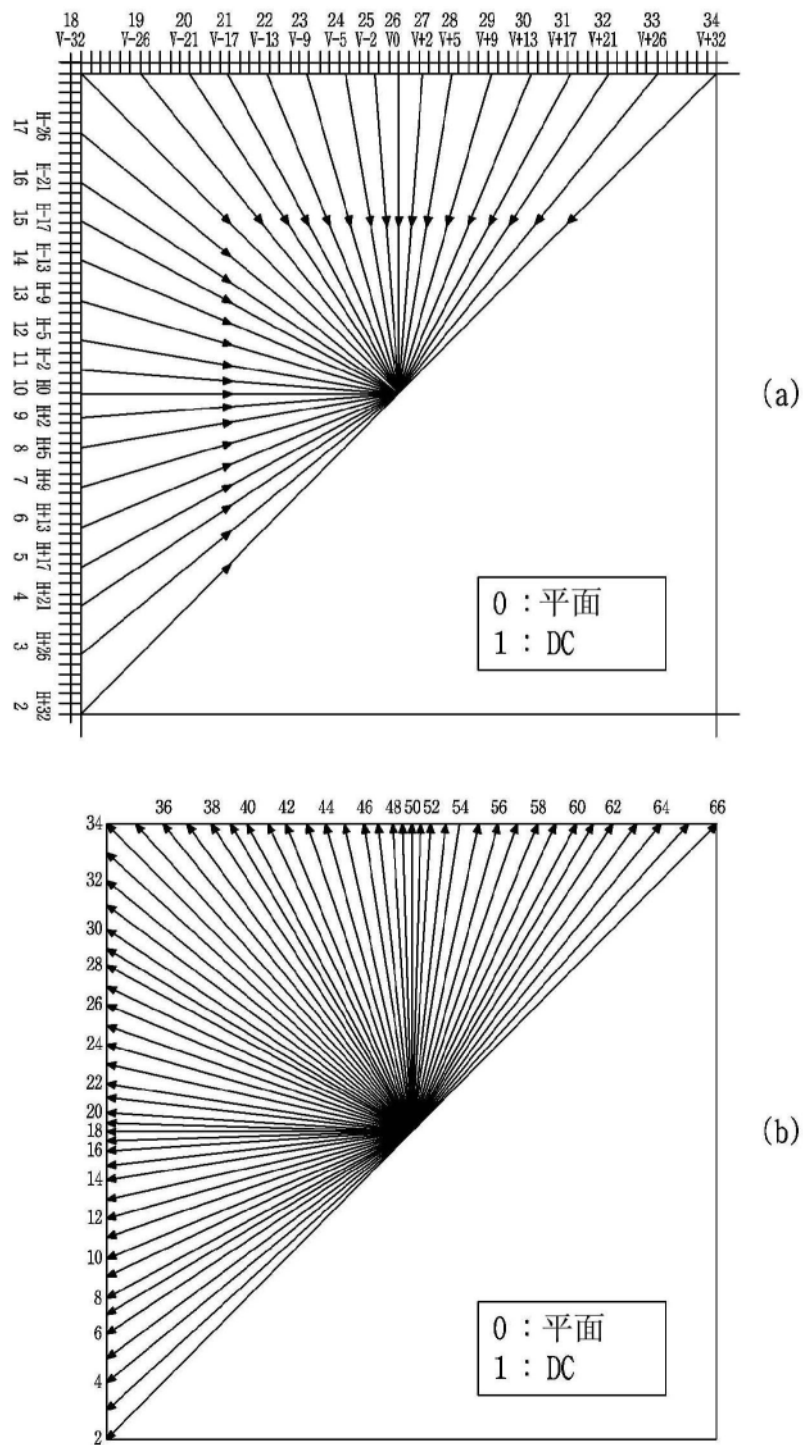


图21

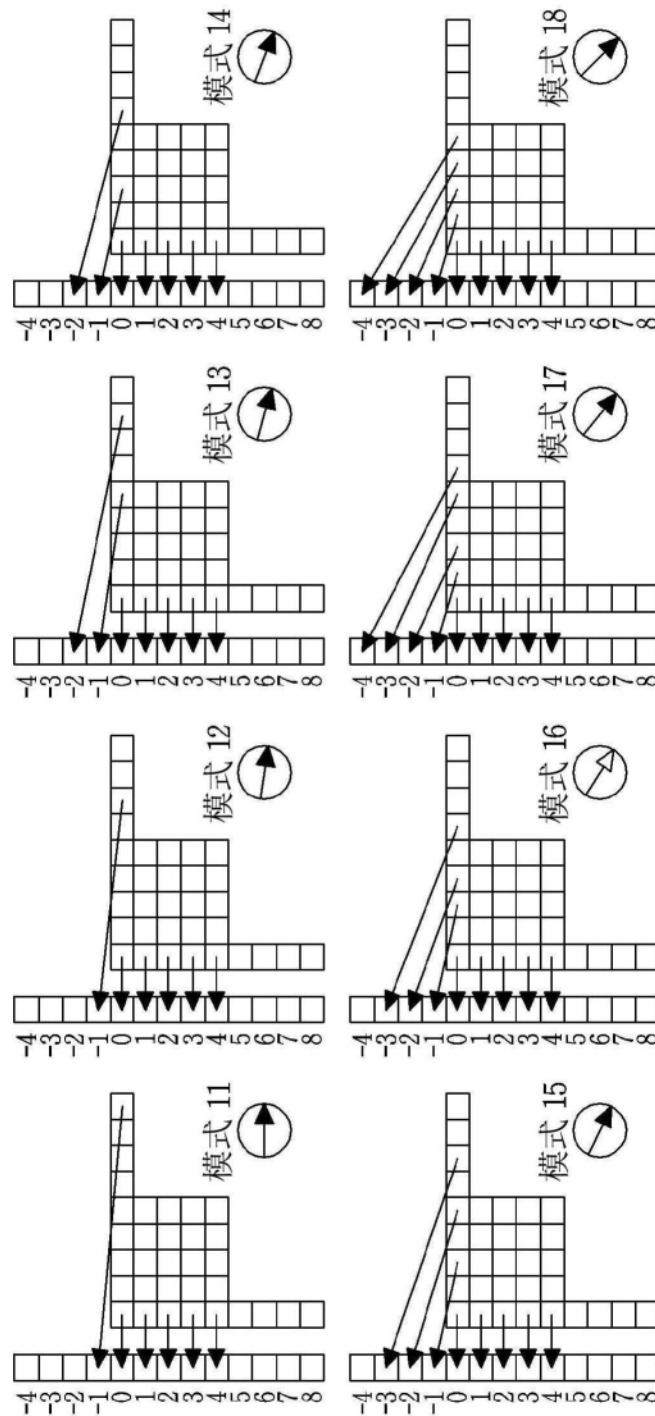


图22

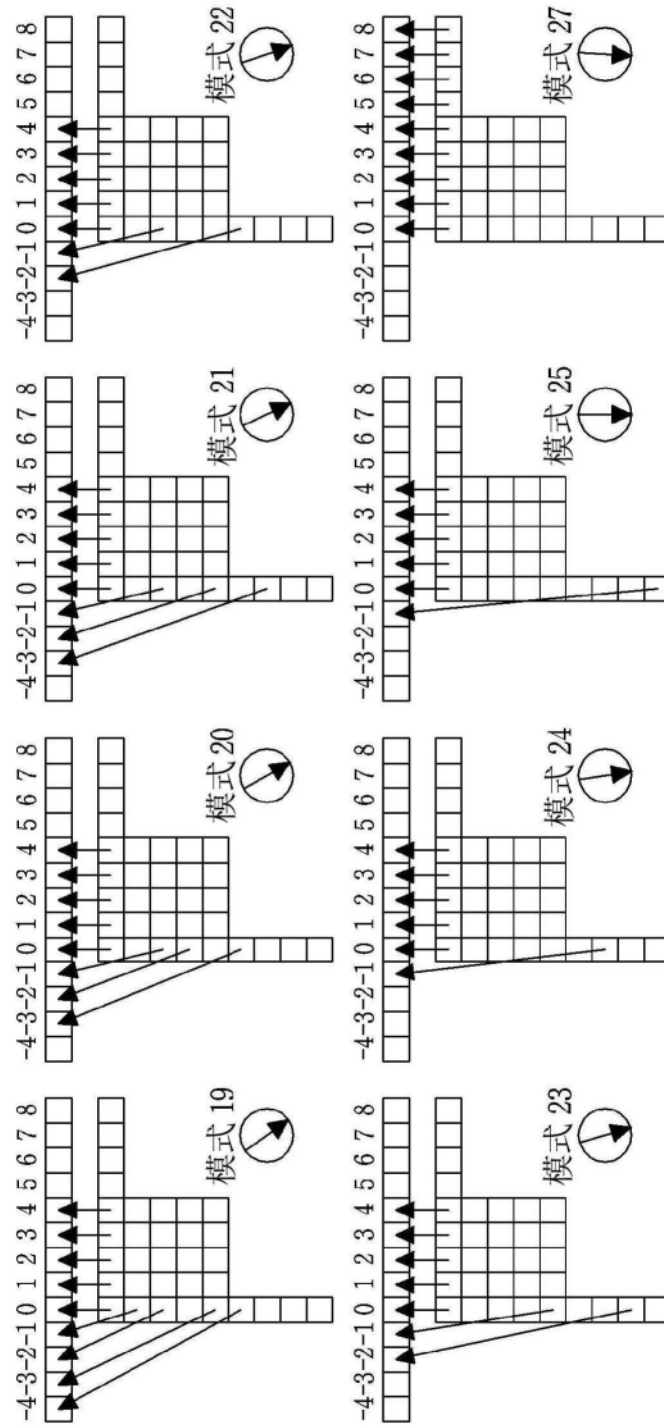


图23

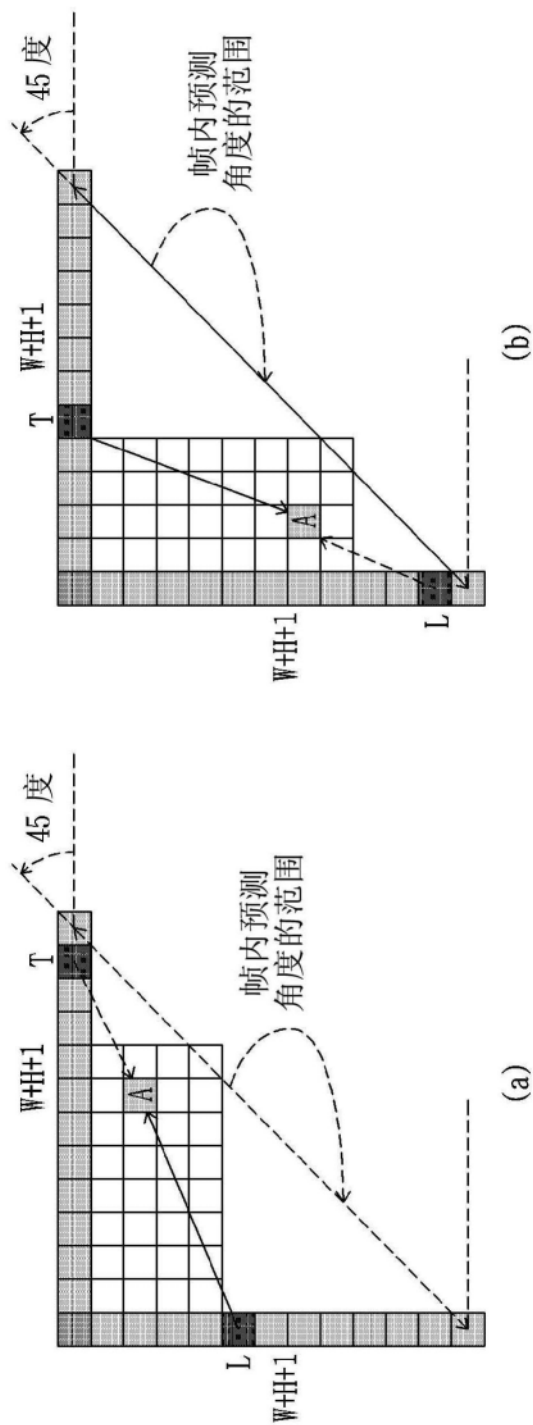


图25

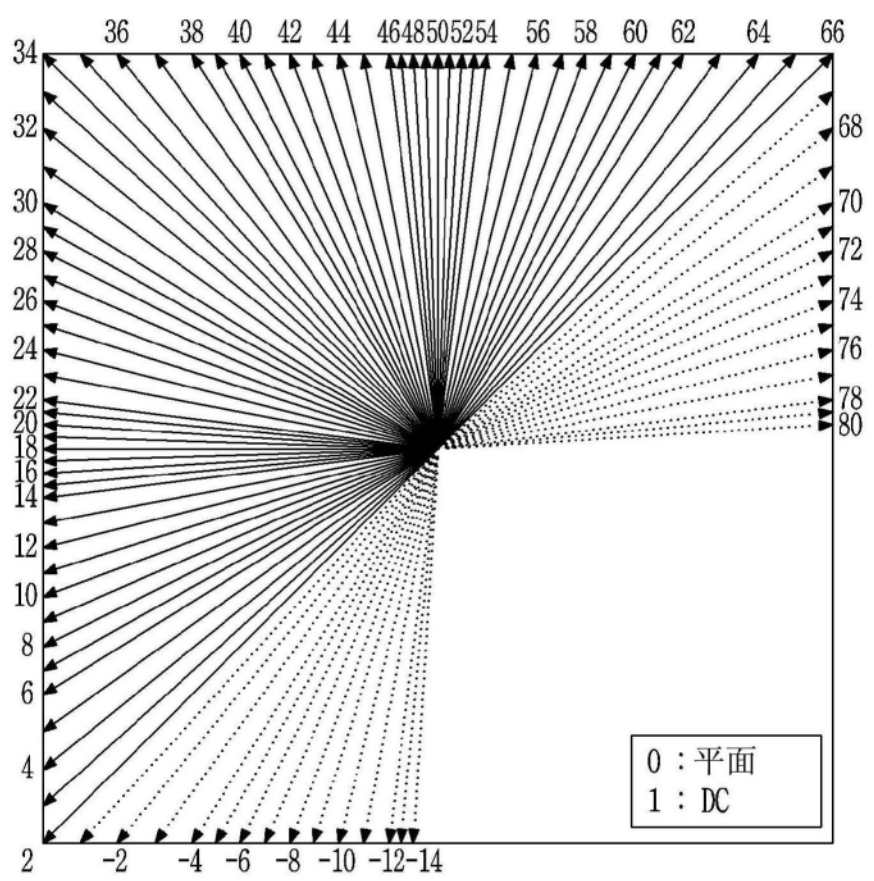


图26

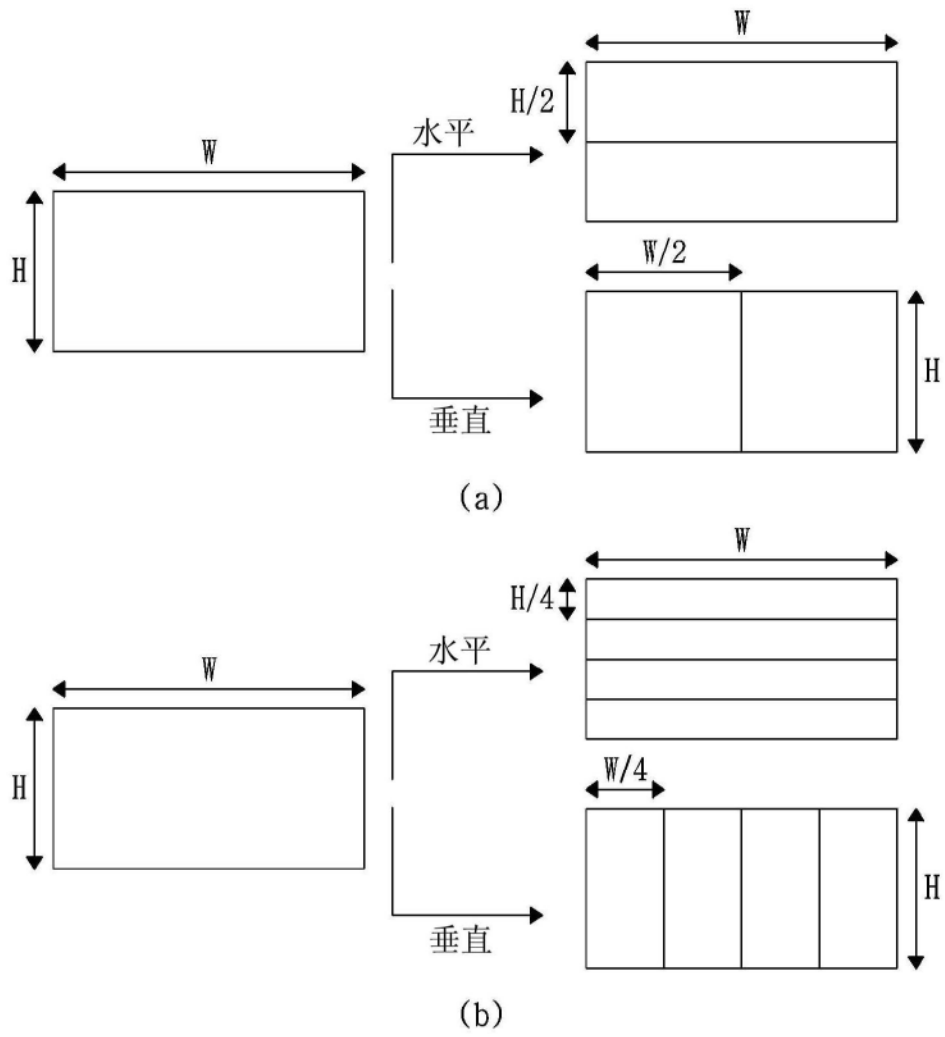


图27

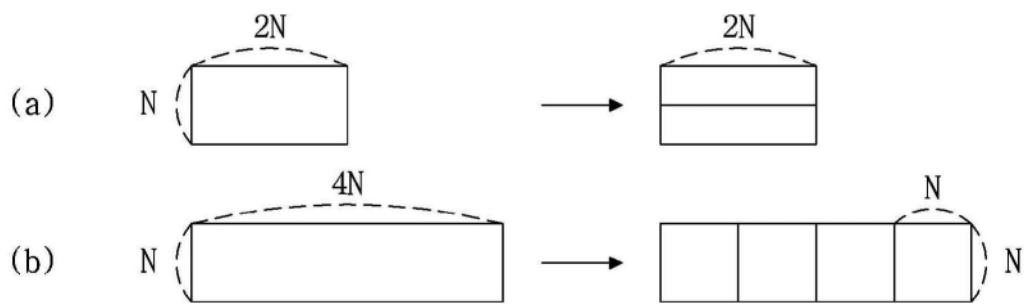


图28

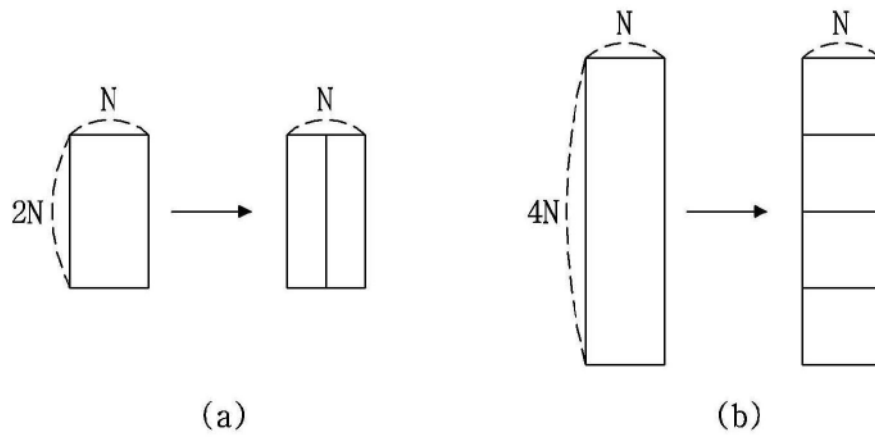


图29

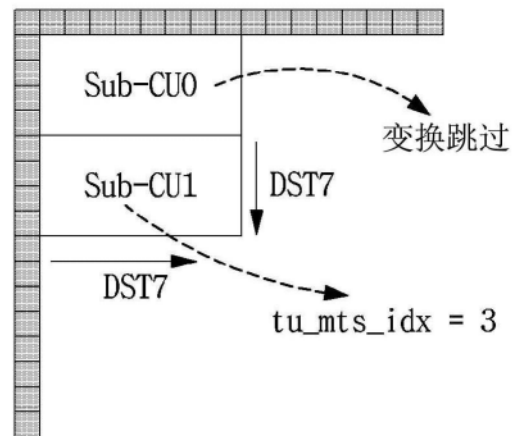


图30

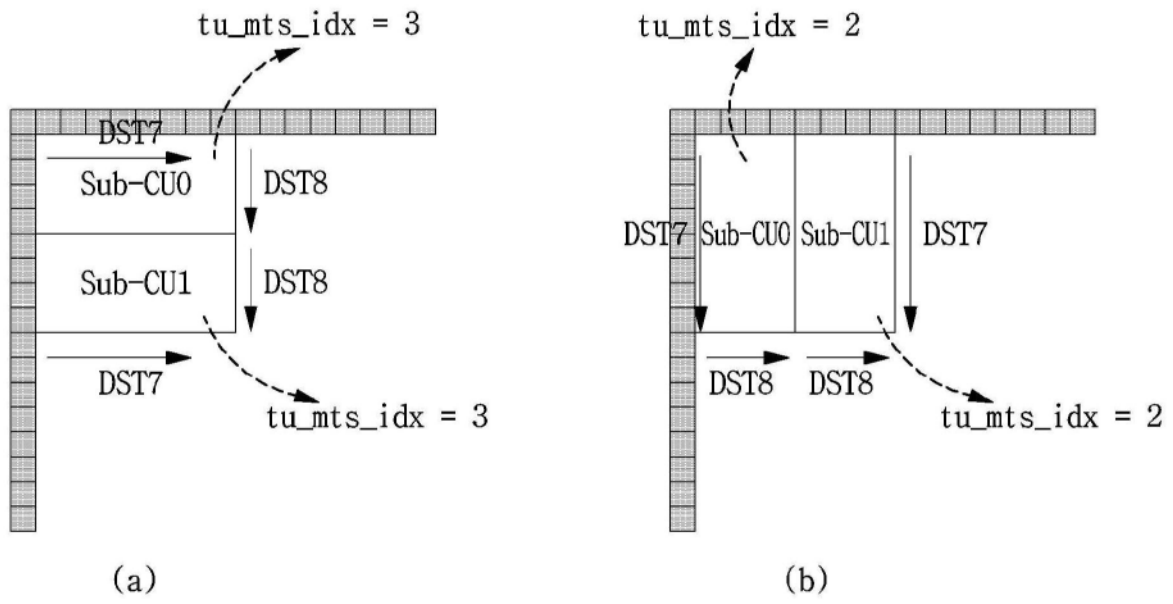


图31



图32



图33

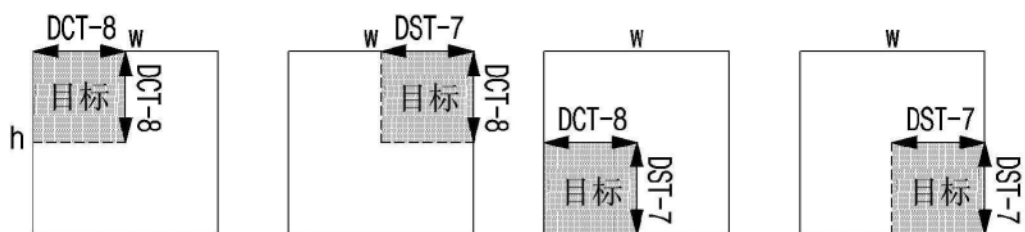


图34

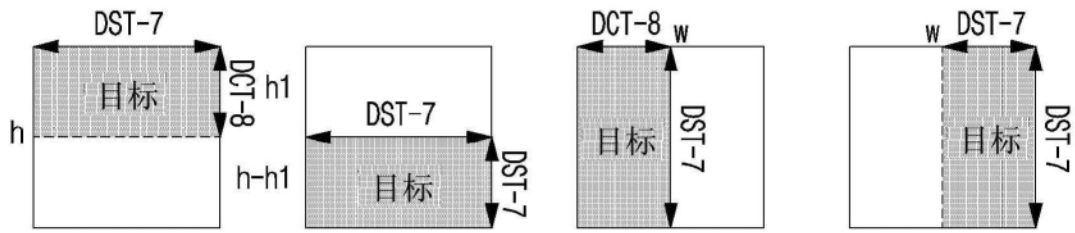


图35

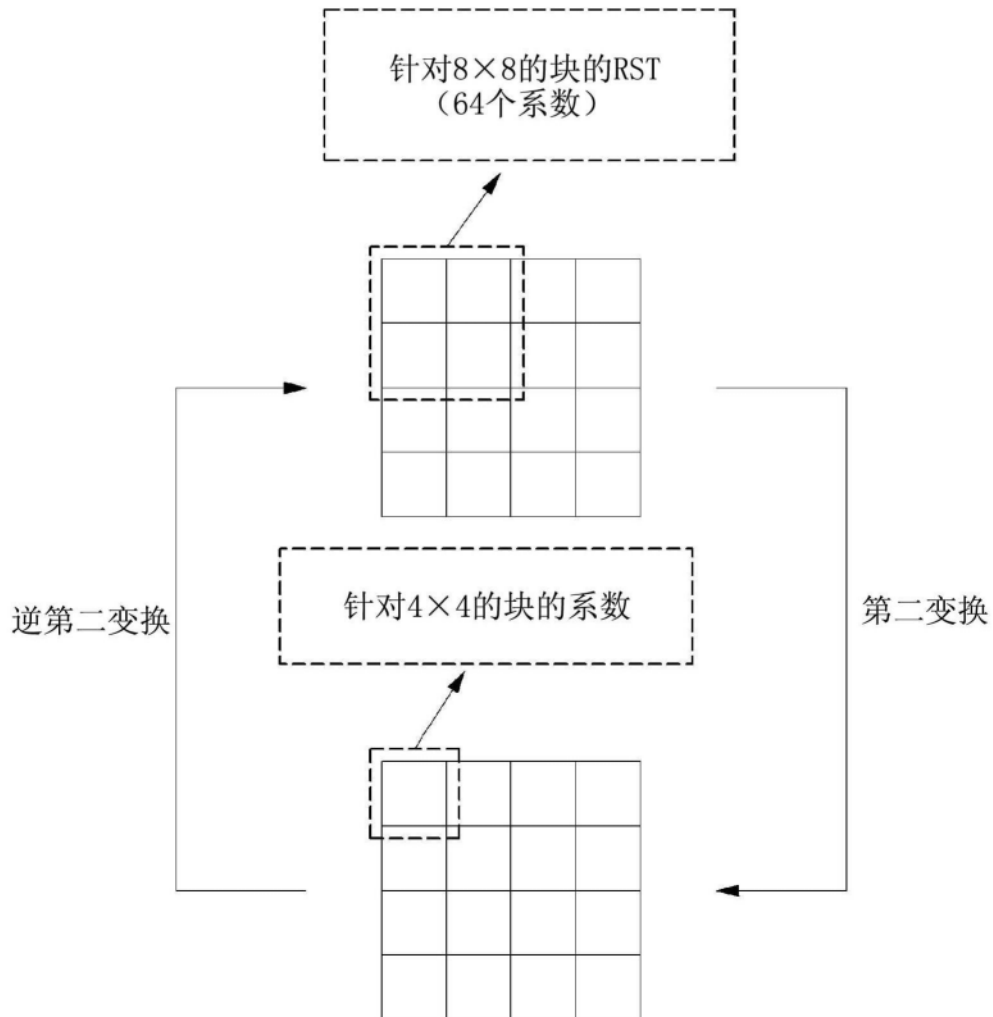


图36

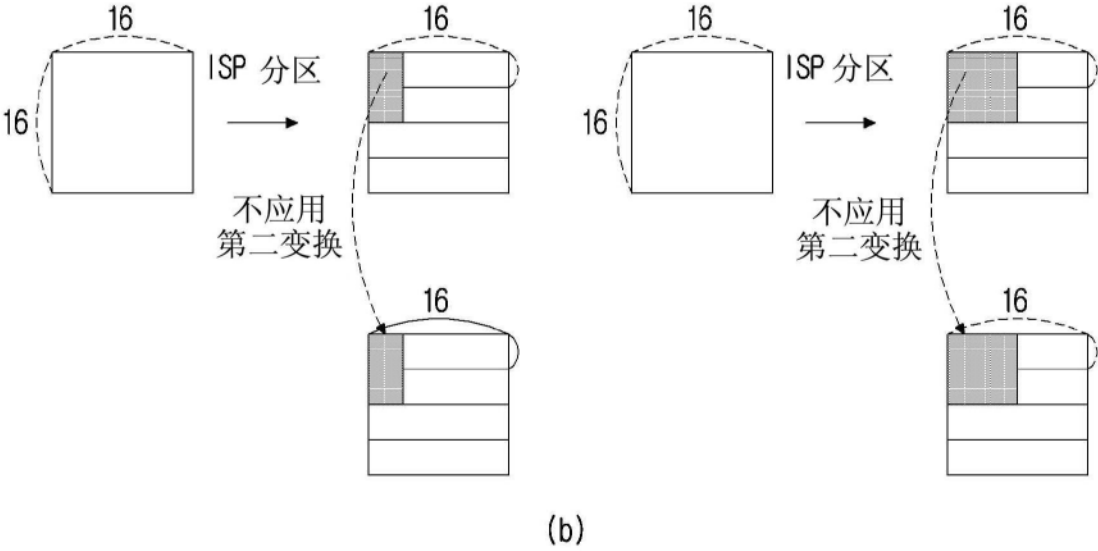
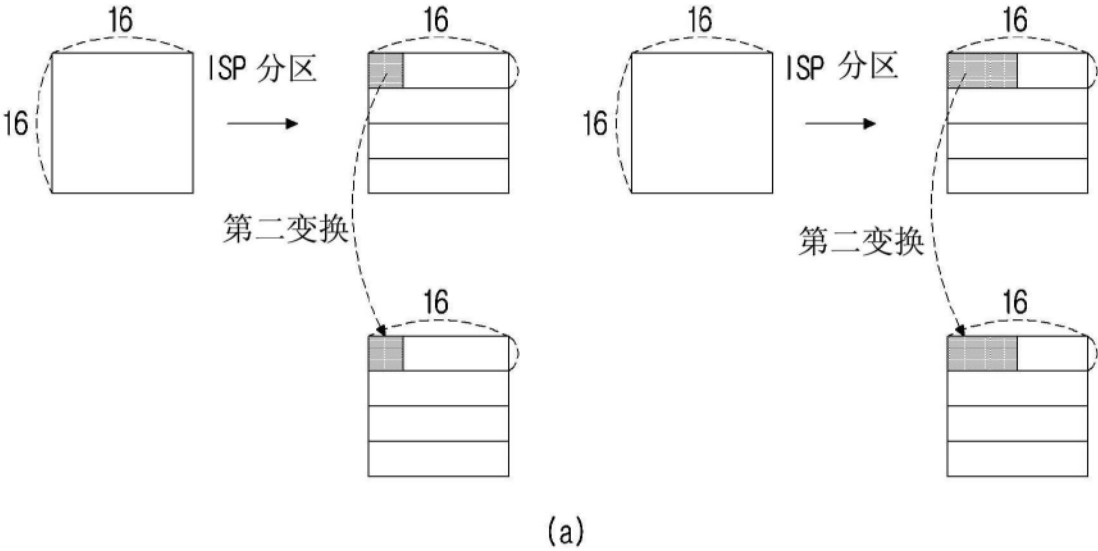


图37

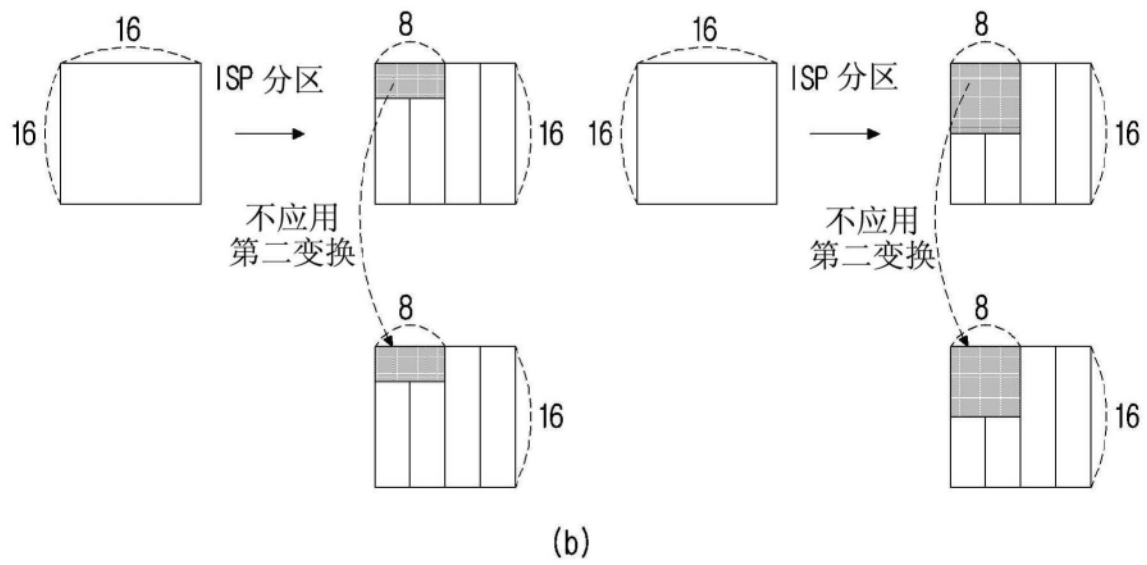
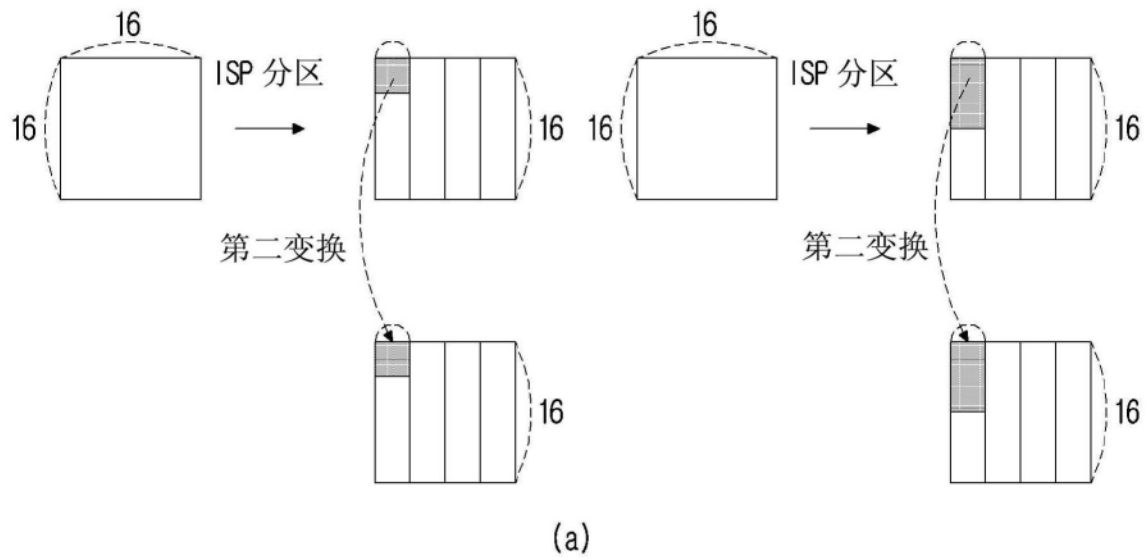


图38

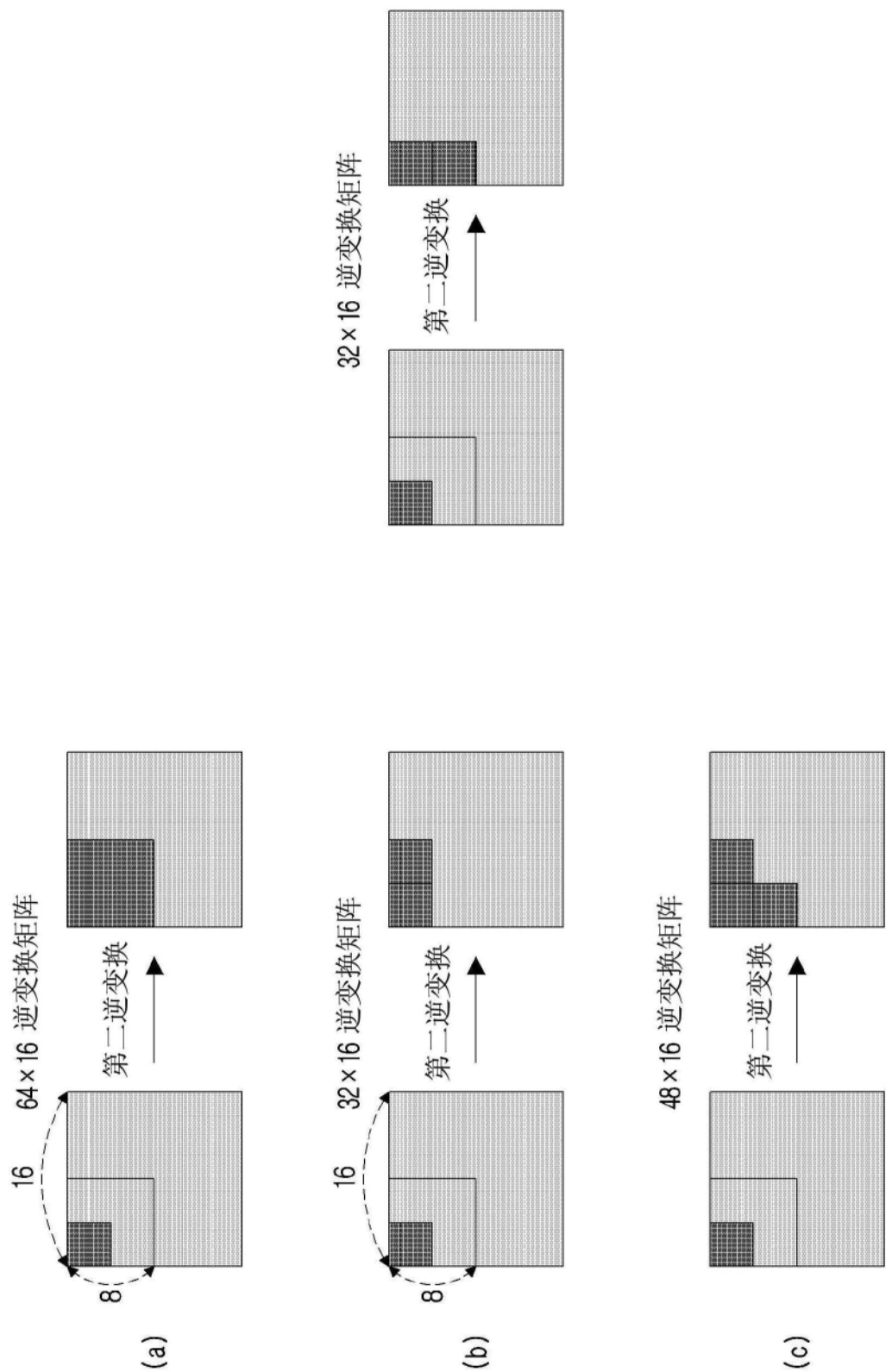


图39

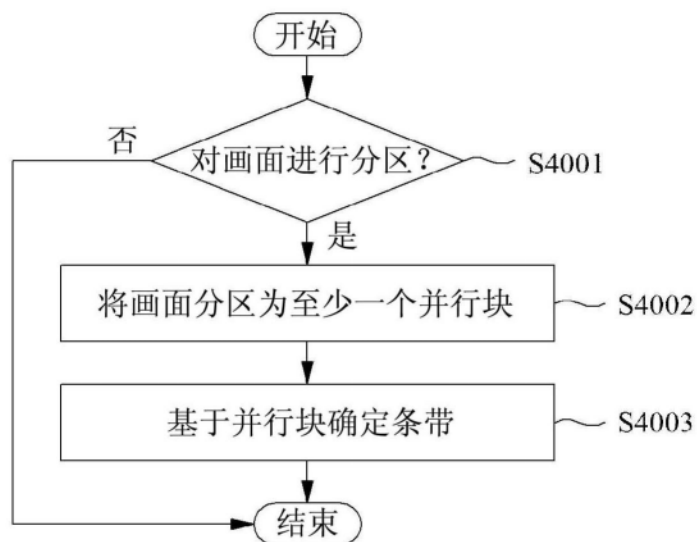


图40

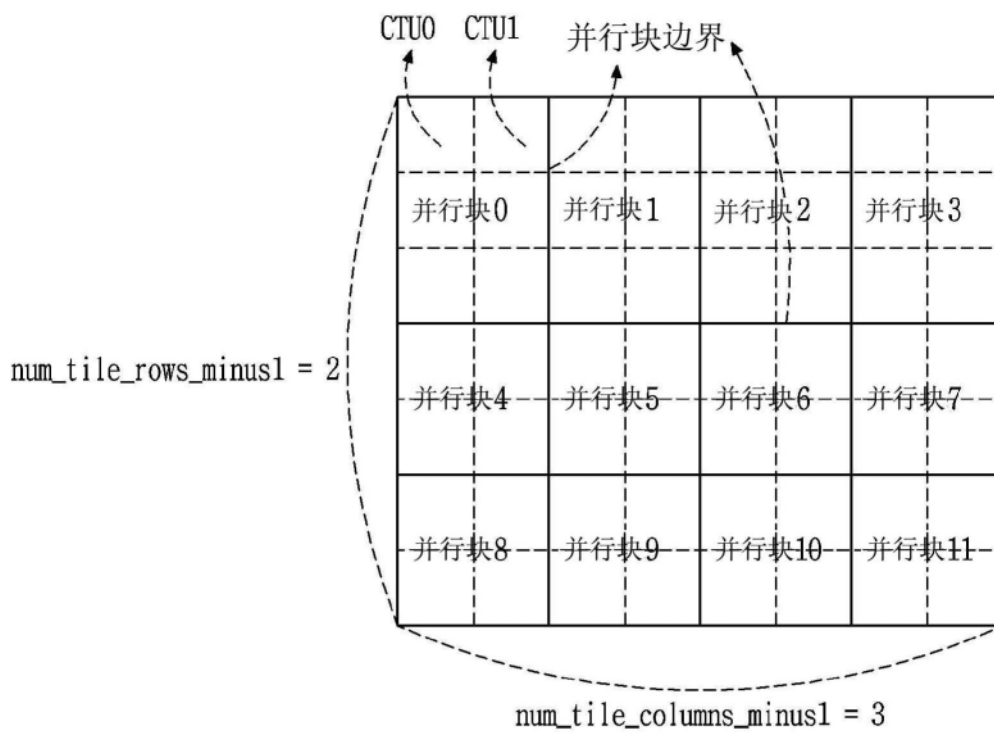


图41

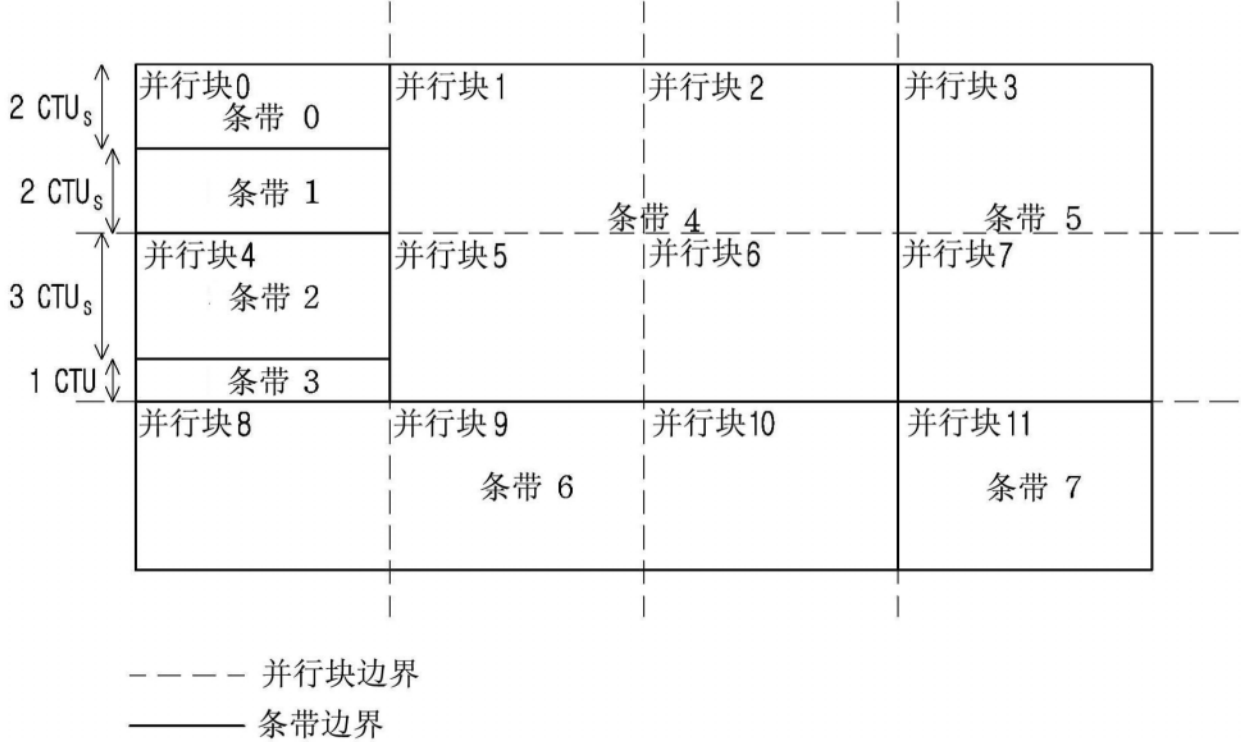


图42

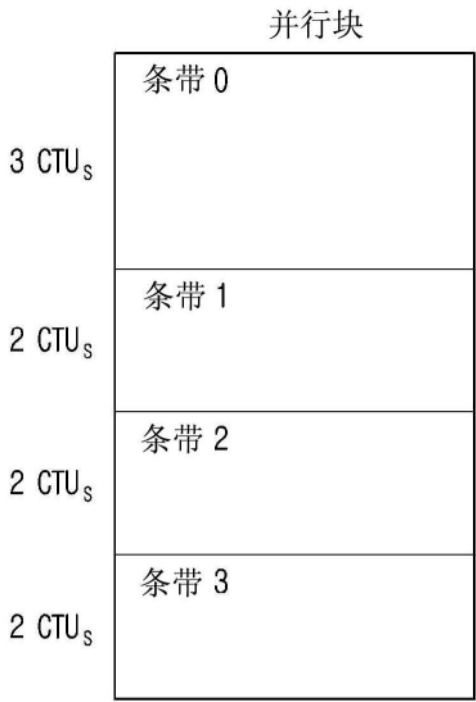


图43



图44

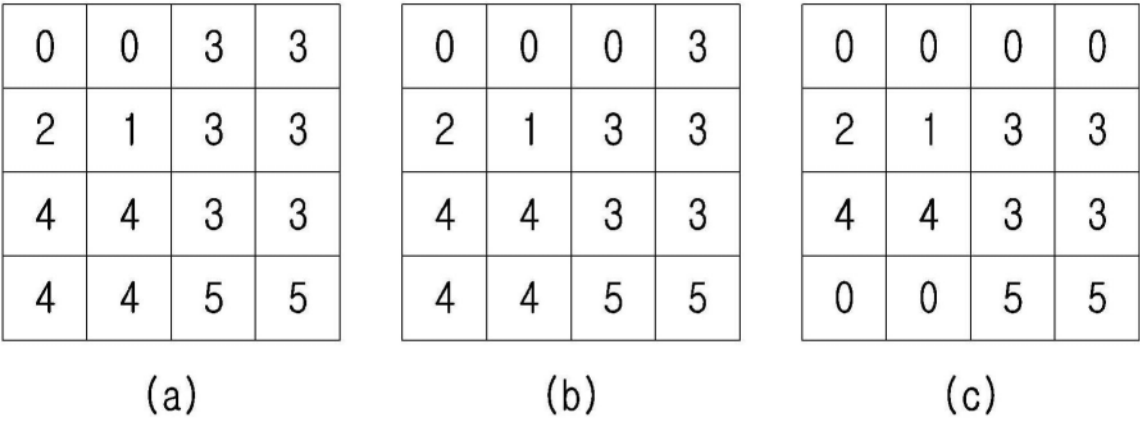


图45

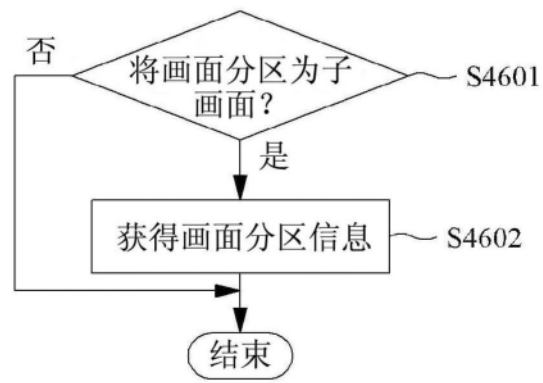


图46