



(12) 发明专利

(10) 授权公告号 CN 113691807 B

(45) 授权公告日 2023. 05. 09

(21) 申请号 202110872234.5

(22) 申请日 2019.11.08

(65) 同一申请的已公布的文献号
申请公布号 CN 113691807 A

(43) 申请公布日 2021.11.23

(30) 优先权数据
10-2018-0136261 2018.11.08 KR

(62) 分案原申请数据
201980071283.X 2019.11.08

(73) 专利权人 OPPO广东移动通信有限公司
地址 523860 广东省东莞市长安镇乌沙海
滨路18号

(72) 发明人 李培根

(74) 专利代理机构 北京派特恩知识产权代理有
限公司 11270
专利代理师 王军红 张颖玲

(51) Int.Cl.
H04N 19/109 (2014.01)
H04N 19/513 (2014.01)
H04N 19/176 (2014.01)

(56) 对比文件
JP 2012186759 A, 2012.09.27
KR 20180061060 A, 2018.06.07
US 2014286429 A1, 2014.09.25
Xu Chen等.CET4:Merge Offset
Extension.《Joint Video Experts Team(JEVT)
of ITU-T SG 16 WP 3 and ISO/IEC JTC1/SC
29/WG11》.2018,第1-6页.

审查员 杨欣怡

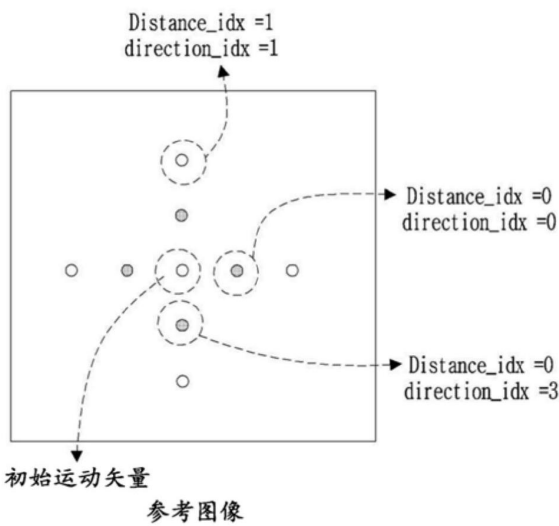
权利要求书2页 说明书28页 附图12页

(54) 发明名称

视频信号编码/解码方法以及用于所述方法
的设备

(57) 摘要

根据本发明的视频解码方法包括以下步骤：
生成当前块的合并候选列表；从在所述合并候选
列表中包括的合并候选中确定所述当前块的合
并候选；导出所述当前块的偏移矢量；以及通过
将所述偏移矢量加上所述合并候选的运动矢量
来导出所述当前块的运动矢量。



1. 一种视频解码方法,包括以下步骤:
生成当前块的合并候选列表;
从在所述合并候选列表中包括的合并候选中确定所述当前块的合并候选;
导出所述当前块的偏移矢量;以及
通过将所述偏移矢量加上所述合并候选的运动矢量来导出所述当前块的运动矢量;
其中,所述偏移矢量的大小基于指定运动大小候选中的任何一个的第一索引信息确定;

其中,所述运动大小候选的最大值和最小值中的至少一个根据所述当前块的运动矢量精度而不同地设置。

2. 根据权利要求1所述的视频解码方法,其中,
所述偏移矢量的大小是通过对由所述第一索引信息指定的所述运动大小候选所表示的值进行移位操作而获得的。

3. 根据权利要求1所述的视频解码方法,其中,
基于指定矢量方向候选中的任何一个的第二索引信息来确定所述偏移矢量的方向。

4. 根据权利要求1所述的视频解码方法,其中,根据所述运动矢量精度而地不同地设置运动矢量偏移大小候选的范围。

5. 根据权利要求1所述的视频解码方法,其中,所述当前块的所述运动矢量精度为分数像素,索引信息的数值被设置为1、2、4、8或16。

6. 根据权利要求5所述的视频解码方法,其中,所述分数像素包括四分之一像素。

7. 根据权利要求1所述的视频解码方法,其中,所述当前块的所述运动矢量精度为整数像素,索引信息的数值被设置为4、8、16、32或64。

8. 根据权利要求1所述的视频解码方法,其中,用于确定所述运动矢量精度的信息以比特流的图像级用信号发送。

9. 一种视频编码方法,包括以下步骤:生成当前块的合并候选列表;
从在所述合并候选列表中包括的合并候选中选择所述当前块的合并候选;
导出所述当前块的偏移矢量;以及
通过将所述偏移矢量加上所述合并候选的运动矢量来导出所述当前块的运动矢量;
其中,对第一索引信息进行编码,所述第一索引信息用于指定多个运动大小候选中指示所述偏移矢量的大小的运动大小候选;

其中,所述运动大小候选的最大值和最小值中的至少一个根据所述当前块的运动矢量精度而不同地设置。

10. 根据权利要求9所述的视频编码方法,其中,
所述运动大小候选具有通过对所述偏移矢量的大小进行移位操作而导出的值。

11. 根据权利要求9所述的视频编码方法,其中,
对第二索引信息进行编码,所述第二索引信息用于指定多个矢量方向候选中指示所述偏移矢量的方向的矢量方向候选。

12. 一种视频解码设备,包括:帧间预测部,所述帧间预测部用于:生成当前块的合并候选列表;从在所述合并候选列表中包括的合并候选中确定所述当前块的合并候选;导出所述当前块的偏移矢量;并且通过将所述偏移矢量加上所述合并候选的运动矢量来导出所述

当前块的运动矢量;其中,所述偏移矢量的大小基于指定运动大小候选中的任何一个的第一索引信息确定;其中,所述运动大小候选的最大值和最小值中的至少一个根据所述当前块的运动矢量精度而不同地设置。

视频信号编码/解码方法以及用于所述方法的设备

[0001] 本案为2019年11月8日递交的国际专利申请PCT/KR2019/015194的中国国家阶段专利申请201980071283.X的分案申请。

技术领域

[0002] 本发明涉及视频信号编码/解码方法及用于所述方法的设备。

背景技术

[0003] 随着显示面板越来越大的趋势,逐渐需要画质更高的视频服务。高清视频服务的最大问题在于数据量大幅增加,为了解决这种问题,正在积极进行用于提高视频压缩率的研究。作为代表性示例,在2009年,运动图像专家组(Motion Picture Experts Group, MPEG)和国际电信联盟-电信(International Telecommunication Union-Telecommunication, ITU-T)旗下的视频编码专家组(Video Coding Experts Group, VCEG)建立了视频编码联合协作团队JCT-VC(Joint Collaborative Team on Video Coding)。JCT-VC提出了视频压缩标准HEVC(高效视频编码, High Efficiency Video Coding)并于2013年1月25日获得批准,其压缩性能约是H.264/AVC的2倍。随着高清视频服务的急剧发展,HEVC的性能也逐渐暴露出其局限性。

发明内容

[0004] 要解决的技术问题

[0005] 本发明的目的是提供一种用于在对视频信号进行编码/解码时基于偏移矢量来细化从合并候选导出的运动矢量的方法以及一种用于执行该方法的设备。

[0006] 本发明的目的是提供一种在对视频信号进行编码/解码时用信号发送偏移矢量的方法以及一种用于执行该方法的设备。

[0007] 本发明要实现的技术问题不限于以上提及的技术问题,并且本发明所属技术领域的普通技术人员将通过以下描述清楚理解未提到的其他技术问题。

[0008] 技术方案

[0009] 根据本发明的视频信号解码/编码方法包括以下步骤:生成当前块的合并候选列表;从在所述合并候选列表中包括的合并候选中确定所述当前块的合并候选;导出所述当前块的偏移矢量;以及通过将所述偏移矢量加上所述合并候选的运动矢量来导出所述当前块的运动矢量。

[0010] 在根据本发明的视频信号解码/编码方法中,可以基于指定运动大小候选中的任何一个的第一索引信息来确定所述偏移矢量的大小。

[0011] 在根据本发明的视频信号解码/编码方法中,可以根据指示所述运动大小候选的范围的标志的值而不同地设置所述运动大小候选的最大值或最小值中的至少一个。

[0012] 在根据本发明的视频信号解码/编码方法中,可以以图像级用信号发送所述标志。

[0013] 在根据本发明的视频信号解码/编码方法中,可以根据所述当前块的运动矢量精

度而不同地设置所述运动大小候选的最大值或最小值中的至少一个。

[0014] 在根据本发明的视频信号解码/编码方法中,可以通过对由所述第一索引信息指定的所述运动大小候选所表示的值进行移位操作来获得所述偏移矢量的大小。

[0015] 在根据本发明的视频信号解码/编码方法中,可以基于指定矢量方向候选中的任何一个的第二索引信息来确定所述偏移矢量的方向。

[0016] 以上对本发明简要概述的特征仅是稍后将描述的本发明的详细说明书的示例性实施方式,并不限定本发明的范围。

[0017] 发明效果

[0018] 根据本发明,可以通过基于偏移矢量对合并候选的运动矢量进行细化来提高帧间预测效率。

[0019] 根据本发明,可以通过自适应地确定偏移矢量的大小和方向来提高帧间预测效率。

[0020] 在本发明中可获得的效果不限于上述效果,并且本发明所属技术领域的普通技术人员将通过以下描述清楚理解未提到的其他效果。。

附图说明

[0021] 图1是本发明实施例的视频编码器(encoder)的框图。

[0022] 图2是本发明实施例的视频解码器(decoder)的框图。

[0023] 图3是示出本发明实施例的基本编码树单元的图。

[0024] 图4(a) - 图4(e)是示出编码块的多种划分类型的图。

[0025] 图5是示出编码树单元的划分方面的图。

[0026] 图6是本发明实施例的帧间预测方法的流程图。

[0027] 图7是示出对象的非线性运动的图。

[0028] 图8是展示了根据本发明实施例的基于仿射运动的帧间预测方法的流程图。

[0029] 图9是示出每个仿射运动模型的仿射种子矢量的示例的图。

[0030] 图10是示出在4参数运动模型下的子块的仿射矢量的示例的图。

[0031] 图11是在合并模式下导出当前块的运动信息的过程的流程图。

[0032] 图12是示出了用于导出合并候选的候选块的示例的图。

[0033] 图13是示出了参考样本的位置的图。

[0034] 图14是示出了用于导出合并候选的候选块的示例的图。

[0035] 图15是示出了改变参考样本的位置的示例的图。

[0036] 图16是示出改变参考样本的位置的示例的图。

[0037] 图17是示出根据指示偏移矢量的大小的distance_idx和指示偏移矢量的方向的direction_idx的值的偏移矢量的图。

[0038] 图18是示出根据指示偏移矢量的大小的distance_idx和指示偏移矢量的方向的direction_idx的值的偏移矢量的图。

具体实施方式

[0039] 以下,参照附图详细说明本发明的实施例。

[0040] 视频的编码及解码以块为单位来执行。例如,可以对编码块、变换块或预测块执行变换、量化、预测、环内滤波或重建等编码/解码处理。

[0041] 以下,将要编码/解码的块称为“当前块”。例如,根据当前编码/解码处理步骤,当前块可以表示编码块、变换块或预测块。

[0042] 另外,本说明书中使用的术语“单元”表示用于执行特定编码/解码过程的基本单元,并且“块”可以被理解为表示预定大小的样本阵列。除非另有说明,“块”和“单元”可互换使用。例如,在稍后描述的实施例中,编码块和编码单元可被理解为具有相同的含义。

[0043] 图1是本发明实施例的视频编码器(encoder)的框图。

[0044] 参照图1,视频编码设备100可以包括图像划分部110、预测部120、125、变换部130、量化部135、重排部160、熵编码部165、逆量化部140、逆变换部145、滤波器部150及存储器155。

[0045] 图1所示的各部件单独示出以便表示视频编码设备中相互不同的特征性功能,并不表示各部件由分离的硬件或单一软件组件构成。即,对于各部件而言,为了便于说明,以排列各个部件的方式使得各部件中的至少两个部件合并成一个部件或者使得一个部件划分成多个部件,并由此执行功能,只要不脱离本发明的本质,这种整合各部件的实施例及使得各部件分离的实施例也属于本发明的权利范围。

[0046] 而且,部分结构要素并不是在本发明中执行本质功能的必要结构要素,而是仅用于提高性能的可选结构要素。本发明可以通过仅包括除了仅用于提高性能的结构要素之外的实现本发明的本质所需的部件来实施,并且仅包括除了仅用于提高性能的选择性结构要素之外的必要结构要素的结构也属于本发明的权利范围。

[0047] 图像划分部110可以将输入的图像划分为至少一个处理单元。在这种情况下,处理单元可以是预测单元(Prediction Unit:PU),也可以是变换单元(Transform Unit:TU),还可以是编码单元(Coding Unit:CU)。图像划分部110将一个图像划分为多个编码单元、预测单元及变换单元的组合,可以基于预定标准(例如,成本函数)选择一个编码单元、预测单元及变换单元组合来对图像进行编码。

[0048] 例如,可以将一个图像划分为多个编码单元。为了将图像分割为编码单元,可以使用如四叉树结构(Quad Tree Structure)的递归树状结构,可以将一个视频或最大编码单元(largest coding unit)作为根(root)而将编码单元划分为可具有相当于划分的编码单元数量的子节点的另外编码单元。根据某些限制不再被划分的编码单元将成为叶节点。即,当假设一个编码单元只能实现正方形划分时,一个编码单元可最多被划分为4个其他编码单元。

[0049] 以下,在本发明的实施例中,编码单元可意味着执行编码的单元,还可意味着执行解码的单元。

[0050] 可以将一个编码单元内的预测单元分割为相同大小的正方形或矩形中的至少一个等形状,也可以将一个编码单元内的一个预测单元中分割为具有与另一个预测单元不同的形状和/或大小。

[0051] 当基于编码单元执行帧内预测的预测单元并非是最小编码单元的情况下,可以执行帧内预测,而无需分割为多个预测单元 $N \times N$ 。

[0052] 预测部120、125可以包括执行帧间预测的帧间预测部120和执行帧内预测的帧内

预测部125。可以确定对于预测单元使用帧间预测还是执行帧内预测,并基于各预测方法确定具体信息(例如,帧内预测模式、运动矢量、参考图像等)。在这种情况下,执行预测的处理单元有可能与确定预测方法及具体内容的处理单元不同。例如,可以由预测单元确定预测方法和预测模式等,并且可以由变换单元执行预测。所生成的预测块与原始块之间的残差值(残差块)可以输入到变换部130。并且,可以将用于预测的预测模式信息、运动矢量信息等与残差值一同在熵编码部165中进行编码并传送至解码器。当使用特定的编码模式时,也可以通过直接对原始块进行编码并传送至解码器,而不通过预测部120、125生成预测块。

[0053] 帧间预测部120可以基于当前图像的上一个图像或下一个图像中的至少一个图像的信息来预测出预测单元,在一些情况下,还可以基于当前图像内已编码的一部分区域的信息来预测出预测单元。帧间预测部120可以包括参考图像插值部、运动预测部、运动补偿部。

[0054] 参考图像插值部从存储器155接收参考图像信息,并且可以从参考图像生成整数像素或以下的像素信息。对于亮度像素而言,为了以1/4像素单元生成整数像素以下的像素信息,可以使用滤波器系数不同的基于DCT的8阶插值滤波器(DCT-based Interpolation Filter)。对于色度信号而言,为了以1/8像素单元生成整数像素以下的像素信息,可以使用滤波器系数不同的基于DCT的4阶插值滤波器(DCT-based Interpolation Filter)。

[0055] 运动预测部可以基于通过参考图像插值部进行插值的参考图像来执行运动预测。用于计算运动矢量的方法可以使用全搜索块匹配算法(FBMA, Full search-based Block Matching Algorithm)、三步搜索法(TSS, Three Step Search)、新三步搜索算法(NTS, New Three-Step Search Algorithm)等多种方法。运动矢量可以基于插值的像素来具有以1/2像素或1/4像素为单位的运动矢量值。可以在运动预测部中通过采取不同的运动预测方法来预测当前预测单元。运动预测方法可以使用跳过(Skip)方法、合并(Merge)方法、高级运动矢量预测(AMVP, Advanced Motion Vector Prediction)方法、帧内块复制(Intra Block Copy)方法等多种方法。

[0056] 帧内预测部125可以基于作为当前图像内的像素信息的当前块周边的参考像素信息来生成预测单元。在当前预测单元的相邻块是已执行帧间预测的块并且在参考像素是已执行帧间预测的像素的情况下,可以将已执行帧间预测的块中所包括的参考像素用作周边的已执行帧内预测的块的参考像素信息。即,在参考像素不可用的情况下,可以使用可用的参考像素中的至少一个参考像素来代替不可用的参考像素信息。

[0057] 在帧内预测中,预测模式可以具有按预测方向使用参考像素信息的角度预测模式和在执行预测时不使用方向信息的非角度模式。用于预测亮度信息的模式和用于预测色度信息的模式可以不同,为了预测色度信息,可以应用用于预测亮度信息而使用的帧内预测模式信息或所预测到的亮度信号信息。

[0058] 当执行帧内预测时,若预测单元的大小与变换单元的大小相同,则可以基于存在于预测单元的左侧的像素、存在于左上方的像素、存在于上方的像素来对预测单元执行帧内预测。但是,当执行帧内预测时,若预测单元的大小与变换单元的大小不同,则可以使用基于变换单元的参考像素来执行帧内预测。并且,可以应用仅对最小编码单元使用 $N \times N$ 划分的帧内预测。

[0059] 帧内预测方法可以在根据预测模式来对参考像素应用自适应帧内平滑(AIS,

Adaptive Intra Smoothing)滤波器之后生成预测块。用于参考像素的自适应帧内平滑滤波器的类型可能不同。为了执行帧内预测方法,可以从存在于当前预测单元周边的预测单元的帧内预测模式预测出当前预测单元的帧内预测模式。在使用从周边预测单元预测到的模式信息来预测当前预测单元的预测模式的情况下,若当前预测单元与周边预测单元的帧内预测模式相同,则可以使用预定的标志信息来传送表示当前预测单元与周边预测单元的预测模式相同的信息,若当前预测单元与周边预测单元的预测模式不同,则可以通过执行熵编码来对当前块的预测模式信息进行编码。

[0060] 并且,可以生成将包括作为基于在预测部120、125中所生成的预测单元来执行预测的预测单元与预测单元的原始块之间的差值的残差值信息的残差块。可以向变换部130输入所生成的残差块。

[0061] 在变换部130中,可以使用离散余弦变换(DCT,Discrete Cosine Transform)或离散正弦变换(DST,Discrete Sine Transform)等的变换方法来对原始块和包括通过预测部120、125生成的预测单元之间的残差值(residual)信息的残差块进行变换。其中,DCT变换核包括DCT2或DCT8中的至少一个,DST变换核包括DST7。可以基于为了生成残差块而使用的预测单元的帧内预测模式信息来确定要应用DCT还是应用DST来变换残差块。还可以跳过对残差块的变换。可以对表示是否对于残差块的变换进行跳过的标志进行编码。对于大小在阈值以下的残差块、亮度分量或色度分量(4:4:4格式以下),可允许变换跳过。

[0062] 量化部135可以对在变换部130中变换为频域的值进行量化。量化系数可以根据块或视频的重要性而改变。可以向逆量化部140和重排部160提供在量化部135中计算的值。

[0063] 重排部160可以对量化的残差值执行系数值的重排。

[0064] 重排部160可以通过系数扫描(Coefficient Scanning)方法来将2维的块形状系数变更为1维的矢量形式。例如,重排部160可以使用Z字形扫描(Zig-Zag Scan)方法扫描DC系数乃至高频域的系数,并将其改变为1维矢量形式。根据变换单元的大小及帧内预测模式,代替Z字形扫描,还可以使用沿着列方向扫描2维块形状系数的垂直扫描和沿着行方向扫描2维块形状系数的水平扫描。即,可以根据变换单元的大小及帧内预测模式来确定使用Z字形扫描、垂直方向扫描及水平方向扫描中的哪一种扫描方法。

[0065] 熵编码部165可以基于通过重排部160计算出的值来执行熵编码。例如,熵编码可以使用指数哥伦布码(Exponential Golomb)、上下文自适应变长编码(CAVLC,Context-Adaptive Variable Length Coding)、上下文自适应二进制算术编码(CABAC,Context-Adaptive Binary Arithmetic Coding)等的多种编码方法。

[0066] 熵编码部165可以对源自重排部160及预测部120、125的编码单元的残差值系数信息及块类型信息、预测模式信息、划分单元信息、预测单元信息及传送单元信息、运动矢量信息、参考帧信息、块的插值信息、滤波信息等多种信息进行编码。

[0067] 可以在熵编码部165对从重排部160输入的编码单元的系数值进行熵编码。

[0068] 逆量化部140及逆变换部145对在量化部135量化的多个值进行逆量化,并对在变换部130变换的值进行逆变换。在逆量化部140及逆变换部145中生成的残差值可以通过与由预测部120、125中包括的运动预测部、运动补偿部及帧内预测部所预测的预测单元合并来生成重建块(Reconstructed Block)。

[0069] 滤波器部150可以包括去块滤波器、偏移校正部、自适应环路滤波器(ALF,

Adaptive Loop Filter)中的至少一个。

[0070] 去块滤波器可移除因块之间的边界而在重建的图像中生成的块失真。为了判断是否执行去块,能够基于块中所包括的几个列或行所包括的像素来判断是否对当前块应用去块滤波器。在对块应用去块滤波器的情况下,可以根据所需的去块滤波强度来应用强滤子(Strong Filter)或弱滤子(Weak Filter)。并且,在使用去块滤波器的过程中,当执行垂直滤波及水平滤波时,可以同步处理水平方向滤波及垂直方向滤波。

[0071] 偏移校正部可以按像素单位来对执行去块的视频中的与原本视频之间的偏移进行校正。为了对指定图像进行偏移校正,可以使用如下方法,即,在将视频中所包括的像素划分为预定数量的区域之后,确定要执行偏移的区域,并对相应区域应用偏移,或者将各个像素的边缘信息考虑在内来应用偏移。

[0072] 自适应环路滤波(ALF, Adaptive Loop Filtering)可以基于对经过滤波的重建图像和原来的视频进行比较的值来执行。将视频中包括的像素划分成预定的组之后,可以通过确定要用于相应组的一个滤波器来对每个组有区别地执行滤波。与是否应用自适应环路滤波相关的信息和亮度信号可按编码单元(Coding Unit, CU)进行传送,所要应用的自适应环路滤波器的形状及滤波器系数可以根据各个块而不同。并且,还能够应用相同类型(固定类型)的自适应环路滤波器,而不管被应用的块的特性如何。

[0073] 存储器155可以存储通过滤波器部150计算的重建块或图像,可以在执行帧间预测时向预测部120、125提供所存储的重建块或图像。

[0074] 图2是本发明实施例的视频解码器(decoder)的框图。

[0075] 参照图2,视频解码器200可以包括熵解码部210、重排部215、逆量化部220、逆变换部225、预测部230、预测部235、滤波器部240、存储器245。

[0076] 在从视频编码器输入视频比特流时,可以按与视频编码器相反的步骤对所输入的比特流进行解码。

[0077] 熵解码部210可按与视频编码器的熵编码部中执行熵编码的步骤相反的步骤进行熵解码。例如,与视频编码器中执行的方法相对应地,可以应用指数哥伦布码(Exponential Golomb)、上下文自适应变长编码(CAVLC, Context-Adaptive Variable Length Coding)、上下文自适应二进制算术编码(CABAC, Context-Adaptive Binary Arithmetic Coding)等多种方法。

[0078] 熵解码部210可以对与编码器执行的帧内预测及帧间预测相关的信息进行解码。

[0079] 重排部215能够基于在编码部中对借助熵解码部210进行熵解码的比特流进行重排的方法来执行重排。可以将由1维矢量形式表示的多个系数重建成2维的块形状的系数来进行重排。重排部215接收与在编码部执行的系数扫描相关的信息,可以通过基于在相应编码部执行的扫描顺序来进行逆向扫描的方法执行重排。

[0080] 逆量化部220可以基于编码器所提供的量化参数和重排的块的系数值来执行逆量化。

[0081] 逆变换部225可以对视频编码器执行的量化结果执行逆离散余弦变换、逆离散正弦变换,所述逆离散余弦变换、逆离散正弦变换属于在变换部执行的变换的逆变换,即对于离散余弦变换、离散正弦变换的逆变换。其中,DCT变换核可以包括DCT2或DCT8中的至少一个,DST变换核可以包括DST7。可替代地,若在视频编码器中跳过变换,则在逆变换部225可

以不执行逆变换。逆变换可以基于在视频编码器中确定的传送单元来执行。在视频解码器的逆变换部225中,可以根据预测方法、当前块的大小及预测方向等多个信息选择性地执行变换方法(例如,DCT或DST)。

[0082] 预测部230、235能够基于由熵解码部210提供的与预测块生成相关的信息和由存储器245提供的之前解码的块或图像信息来生成预测块。

[0083] 如上所述,当以与视频编码器中的操作相同的方式执行帧内预测时,若预测单元的大小与变换单元的大小相同,则基于存在于预测单元的左侧的像素、存在于左上方的像素、存在于上方的像素来对预测单元执行帧内预测,若执行帧内预测时的预测单元的大小与变换单元的大小不同,则可以使用基于变换单元的参考像素来执行帧内预测。并且,还可以应用仅对最小编码单元使用 $N \times N$ 划分的帧内预测。

[0084] 预测部230、235可以包括预测单元判定部、帧间预测部及帧内预测部。预测单元判定部接收从熵解码部210输入的预测单元信息、帧内预测方法的预测模式信息、帧间预测方法的运动预测相关信息等多种信息,根据当前编码单元对预测单元进行分类,可确定预测单元是在执行帧间预测还是在执行帧内预测。帧间预测部230可以使用由视频编码器提供的当前预测单元的帧间预测所需的信息,并基于当前预测单元所属的当前图像的上一个图像或下一个图像中的至少一个图像中所包括的信息来对当前预测单元执行帧间预测。可替代地,还可以基于在当前预测单元所属的当前图像内已重建的一部分区域的信息来执行帧间预测。

[0085] 为了执行帧间预测,能够基于编码单元来判断相应编码单元中所包括的预测单元的运动预测方法是跳过模式(Skip Mode)、合并模式(Merge Mode)、高级运动矢量预测模式(AMVP Mode)、帧内块复制模式中的哪种模式。

[0086] 帧内预测部235可以基于当前图像内的像素信息来生成预测块。当预测单元为已执行帧内预测的预测单元时,能够基于由视频编码器提供的预测单元的帧内预测模式信息来执行帧内预测。帧内预测部235可以包括自适应帧内平滑(AIS, Adaptive Intra Smoothing)滤波器、参考像素插值部、DC滤波器。自适应帧内平滑滤波器是对当前块的参考像素执行滤波的部分,并且可以根据当前预测单元的预测模式确定是否应用滤波器。可以使用由视频编码器提供的预测单元的预测模式及自适应帧内平滑滤波器信息来对当前块的参考像素执行自适应帧内平滑滤波。若当前块的预测模式为不执行自适应帧内平滑滤波的模式,则可以不应用自适应帧内平滑滤波器。

[0087] 对于参考像素插值部而言,若预测单元的预测模式为基于对参考像素进行插值的像素值来执行帧内预测的预测单元,则可以通过对参考像素进行插值来生成整数值或以下的像素单位的参考像素。若当前预测单元的预测模式为以不对参考像素进行插值的方式生成预测块的预测模式,则可不对参考像素进行插值。若当前块的预测模式为DC模式,则DC滤波器可以通过滤波来生成预测块。

[0088] 可以向滤波器部240提供重建的块或图像。滤波器部240可以包括去块滤波器、偏移校正部、ALF。

[0089] 可以从视频编码器接收与是否对相应块或图像应用去块滤波器相关的信息及与在应用去块滤波器时应用强滤子还是弱滤子相关的信息。从视频解码器的去块滤波器接收由视频编码器提供的与去块滤波器相关的信息,并且可以在视频解码器对相应块执行去块

滤波。

[0090] 偏移校正部能够基于编码时用于视频的偏移校正的类型及偏移量信息等来对已重建的视频执行偏移校正。

[0091] 可以基于由编码器提供的与是否应用ALF相关的信息、ALF系数信息等来将ALF应用于编码单元。这种ALF信息可以通过被包括在特定的参数集中来提供。

[0092] 存储器245存储已重建的图像或块,使得所述图像或块可用作参考图像或参考块,并且可以向输出部提供已重建的图像。

[0093] 图3是示出本发明实施例的基本编码树单元的图。

[0094] 可以将大小最大的编码块定义为编码树块。一个图像可被划分为多个编码树单元(Coding Tree Unit,CTU)。编码树单元是大小最大的编码单元,还可被称为最大编码单元(LCU,Largest Coding Unit)。图3示出了将一个图像划分为多个编码树单元的示例。

[0095] 编码树单元的大小可以以图像级或序列级定义。为此,可以通过图像参数集或序列参数集来用信号发送表示编码树单元的大小的信息。

[0096] 例如,序列内整个图像的编码树单元的大小可以设置为 128×128 。可替代地,可以将图像级的 128×128 或 256×256 中的任一个确定为编码树单元的大小。例如,可以将第一图像中编码树单元的大小设置为 128×128 ,并且可以将第二图像中编码树单元的大小设置为 256×256 。

[0097] 可以通过划分编码树单元来生成编码块。编码块表示用于进行编码/解码处理的基本单元。例如,可按不同编码块执行预测或者变换,或者可按不同编码块确定预测编码模式。其中,预测编码模式表示生成预测图像的方法。例如,预测编码模式可以包括画面内预测(Intra Prediction,帧内预测)、画面间预测(Inter Prediction,帧间预测)、当前图像参考(Current Picture Referencing,CPR,或帧内块复制(Intra Block Copy,IBC))或组合预测(Combined Prediction)。针对编码块,可以使用帧内预测、帧间预测、当前图像参考或组合预测中的至少一种预测编码模式来生成与编码块相关的预测块。

[0098] 可以通过比特流用信号发送表示当前块的预测编码模式的信息。例如,所述信息可以是指示预测编码模式为帧内模式还是帧间模式的1比特标志。可仅在当前块的预测编码模式被确定为帧间模式的情况下使用当前图像参考或组合预测。

[0099] 当前图像参考用于将当前图像设置为参考图像,并从当前图像内的已编码/解码的区域获得当前块的预测块。其中,当前图像意味着包括当前块的图像。可以通过比特流用信号发送表示是否对当前块应用当前图像参考的信息。例如,所述信息可以是1比特标志。当所述标志为真时,可以将当前块的预测编码模式确定为当前图像参考,当所述标志为假时,可以将当前块的预测模式确定为帧间预测。

[0100] 可替代地,可以基于参考图像索引确定当前块的预测编码模式。例如,当参考图像索引指向当前图像时,可以将当前块的预测编码模式确定为当前图像参考。当参考图像索引指向其他图像而不是当前图像时,可以将当前块的预测编码模式确定为帧间预测。即,当前图像参考是使用当前图像内已编码/解码的区域的信息的预测方法,并且帧间预测是使用已编码/解码的其他图像的信息的预测方法。

[0101] 组合预测表示由帧内预测、帧间预测及当前图像参考中的两种以上组合而成的编码模式。例如,在应用组合预测的情况下,可以基于帧内预测、帧间预测或当前图像参考中

的一种来生成第一预测块,并且可以基于另一种来生成第二预测块。若生成第一预测块和第二预测块,则可以通过第一预测块和第二预测块的平均运算或加权和运算来生成最终预测块。可以通过比特流用信号发送表示是否应用组合预测的信息。所述信息可以是1比特标志。

[0102] 图4(a)-图4(e)是示出编码块的多种划分类型的图。

[0103] 可以基于四叉树划分、二叉树划分或三叉树划分将编码块划分为多个编码块。还可以基于四叉树划分、二叉树划分或三叉树划分将所划分的编码块再次划分为多个编码块。

[0104] 四叉树划分是指将当前块划分为4个块的划分技术。作为四叉树划分的结果,可以将当前块划分为4个正方形分区(参考图4(a)部分中的“SPLIT_QT”)。

[0105] 二叉树划分是指将当前块划分为2个块的划分技术。可以将沿着垂直方向(即,使用横穿当前块的垂直线)将当前块划分为两个块的过程称为垂直方向二叉树划分,可以将沿着水平方向(即,使用横穿当前块的水平线)将当前块划分为两个块的过程称为水平方向二叉树划分。可以在经过二叉树划分后将当前块划分为2个非正方形分区。图4(b)部分中的“SPLIT_BT_VER”表示垂直方向二叉树划分结果,并且图4(c)部分中的“SPLIT_BT_HOR”表示水平方向二叉树划分结果。

[0106] 三叉树划分是指将当前块划分为3个块的划分技术。可以将沿着垂直方向(即,使用横穿当前块的两个垂直线)将当前块划分为三个块的过程称为垂直方向三叉树划分,可以将沿着水平方向(即,使用横穿当前块的两个水平线)将当前块划分为三个块的过程称为水平方向三叉树划分。可以在经过三叉树划分后将当前块划分为3个非正方形分区。在这种情况下,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2倍。图4(d)部分中的“SPLIT_TT_VER”表示垂直方向三叉树划分结果,并且图4(e)部分中的“SPLIT_TT_HOR”表示水平方向三叉树划分结果。

[0107] 可以将编码树单元的划分次数定义为划分深度(Partitioning Depth)。可以以序列或图像级确定编码树单元的最大划分深度。由此,编码树单元的最大划分深度可根据不同序列或图像而变得不同。

[0108] 可替代地,可单独确定对于多个划分技术的每一个的最大划分深度。例如,允许四叉树划分的最大划分深度可以与允许二叉树划分和/或三叉树划分的最大划分深度不同。

[0109] 编码器可以通过比特流用信号发送表示当前块的划分形状或划分深度中的至少一个的信息。解码器可以基于从比特流中解析的信息来确定编码树单元的划分形状及划分深度。

[0110] 图5是示出编码树单元的划分方面的图。

[0111] 可以将使用四叉树划分、二叉树划分和/或三叉树划分等划分技术来划分编码块的过程称为多叉树划分(Multi Tree Partitioning)。

[0112] 可以将通过对编码块应用多叉树划分而生成的编码块称为多个下游编码块。当编码块的划分深度为 k 时,多个下游编码块的划分深度设置为 $k+1$ 。

[0113] 另一方面,针对划分深度为 $k+1$ 的多个编码块,可以将划分深度为 k 的编码块称为上游编码块。

[0114] 可以基于上游编码块的划分形状或相邻编码块的划分类型中的至少一个来确定

当前编码块的划分类型。其中,相邻编码块与当前编码块相邻,其可以包括当前编码块的上相邻块、左相邻块或与左上角相邻的相邻块中的至少一个。其中,划分类型可以包括是否划分四叉树、是否划分二叉树、二叉树划分方向、是否划分三叉树或三叉树划分方向中的至少一个。

[0115] 为了确定编码块的划分形状,可以通过比特流用信号发送表示编码块是否被划分的信息。所述信息为1比特标志“split_cu_flag”,当所述标志为真时,表示通过多叉树划分技术来划分编码块。

[0116] 当“split_cu_flag”为真时,可以通过比特流用信号发送表示编码块是否被四叉树划分的信息。所述信息为1比特标志“split_qt_flag”,当所述标志为真时,编码块可被划分为4个块。

[0117] 例如,在图5所示的示例中图示了随着编码树单元被四叉树划分来生成划分深度为1的4个编码块。并且,图示了对作为四叉树划分的结果而生成的4个编码块中的第一个编码块及第四个编码块再次应用四叉树划分。最终,可以生成划分深度为2的4个编码块。

[0118] 并且,可以通过对划分深度为2的编码块再次应用四叉树划分来生成划分深度为3的编码块。

[0119] 在未对编码块应用四叉树划分时,可以通过考虑编码块的大小、编码块是否位于图像边界处、最大划分深度或相邻块的划分形状中的至少一个来确定对所述编码块执行二叉树划分还是三叉树划分。当确定对所述编码块执行二叉树划分或三叉树划分时,可以通过比特流用信号发送表示划分方向的信息。所述信息可以是1比特标志“mtt_split_cu_vertical_flag”。可以基于所述标志来确定划分方向是垂直方向还是水平方向。另外,可以通过比特流用信号发送表示对所述编码块应用二叉树划分或三叉树划分中的哪一个的信息。所述信息可以是1比特标志“mtt_split_cu_binary_flag”。可以基于所述标志来确定对所述编码块应用二叉树划分还是三叉树划分。

[0120] 例如,在图5所示的示例中图示了对划分深度为1的编码块应用垂直方向二叉树划分,对作为所述划分结果而生成的编码块中的左侧编码块应用垂直方向三叉树划分,对右侧编码块应用垂直方向二叉树划分。

[0121] 帧间预测是指使用前一图像的信息来预测当前块的预测编码模式。例如,可以将前一图像内与当前块相同位置的块(以下称为同位块,Collocated block)设置为当前块的预测块。以下,将基于位置与当前块相同的块生成的预测块称为同位预测块(Collocated Prediction Block)。

[0122] 另一方面,若存在于前一图像中的对象已经移动至当前图像中的其他位置,则可以使用对象的运动来有效预测当前块。例如,若可以通过比较前一图像和当前图像来知晓对象的移动方向及大小,则可以考虑对象的运动信息来生成当前块的预测块(或预测图像)。以下,可以将使用运动信息生成的预测块称为运动预测块。

[0123] 可以通过从当前块减去预测块来生成残差块(residual block)。在这种情况下,当存在对象的运动时,可以通过使用运动预测块来代替同位预测块,由此减少残差块的能量,从而可提高残差块的压缩性能。

[0124] 如上所述,可以将使用运动信息生成预测块的过程称为运动补偿预测。在大部分的帧间预测中,可以基于运动补偿预测来生成预测块。

[0125] 运动信息可以包括运动矢量、参考图像索引、预测方向或双向加权值索引中的至少一个。运动矢量表示对象的移动方向及大小。参考图像索引指定参考图像列表中包括的多个参考图像中的当前块的参考图像。预测方向是指单向L0预测、单向L1预测或双向预测(L0预测及L1预测)中的任一个。可以根据当前块的预测方向来使用L0方向的运动信息或L1方向的运动信息中的至少一个。双向加权值索引指定用于L0预测块的加权值及应用于L1预测块的加权值。

[0126] 图6是本发明实施例的帧间预测方法的流程图。

[0127] 参考图6,帧间预测方法包括如下的步骤:确定当前块的帧间预测模式(S601);根据所确定的帧间预测模式来获得当前块的运动信息(S602);以及基于所获得的运动信息来对当前块执行运动补偿预测(S603)。

[0128] 其中,帧间预测模式表示用于确定当前块的运动信息的多种技术,可以包括使用平移(Translation)运动信息的帧间预测模式和使用仿射(Affine)运动信息的帧间预测模式。例如,使用平移运动信息的帧间预测模式可以包括合并模式及高级运动矢量预测模式,使用仿射运动信息的帧间预测模式可以包括仿射合并模式及仿射运动矢量预测模式。根据帧间预测模式,可以基于与当前块相邻的相邻块或从比特流中解析的信息来确定当前块的运动信息。

[0129] 以下,详细说明使用仿射运动信息的帧间预测方法。

[0130] 图7是示出了对象的非线性运动的图。

[0131] 视频内对象的运动有可能是非线性运动。例如,如图7所示的示例,可能发生摄像机放大(Zoom-in)、缩小(Zoom-out)、旋转(Rotation)或仿射变换等对象的非线性运动。当发生对象的非线性运动时,无法通过平移运动矢量来有效表示对象的运动。由此,在发生对象的非线性运动的部分,可以通过使用仿射运动来代替平移运动,由此提高编码效率。

[0132] 图8是展示了根据本发明实施例的基于仿射运动的帧间预测方法的流程图。

[0133] 可以基于从比特流解析的信息来确定是否对当前块应用基于仿射运动的帧间预测技术。具体地,可以基于表示是否对当前块应用仿射合并模式的标志或表示是否对当前块应用仿射运动矢量预测模式的标志中的至少一个来确定是否对当前块应用基于仿射运动的帧间预测技术。

[0134] 当将基于仿射运动的帧间预测技术应用于当前块时,可以确定当前块的仿射运动模型(S801)。仿射运动模型可以由6参数仿射运动模型或4参数仿射运动模型中的至少一个来确定。6参数仿射运动模型使用6个参数来表示仿射运动,4参数仿射运动模型使用4个参数来表示仿射运动。

[0135] 等式1为使用6个参数表示仿射运动的情况。仿射运动表示与通过仿射种子矢量确定的预定区域的平移运动。

[0136] **【等式1】**

[0137] 当使用6个参数表示仿射运动时,可以表示复杂的运动,但是对各个参数进行编码时所需的比特数增加,这会降低编码效率。由此,还可以使用4个参数表示仿射运动。等式2为使用4个参数表示仿射运动的情况。

[0138] **【等式2】**

[0139] 可以对用于确定当前块的仿射运动模型的信息进行编码并且可以通过比特流用

信号发送。例如,所述信息可以是1比特标志“affine_type_flag”。所述标志的值为0表示应用4参数仿射运动模型,所述标志的值为1表示应用6参数仿射运动模型。所述标志可以以条带、图块或块(例如,编码块或编码树单元)为单位进行编码。当以条带级用信号发送标志时,可以对所述条带中所属的所有块应用以所述条带级确定的仿射运动模型。

[0140] 可替代地,可以基于当前块的仿射帧间预测模式来确定当前块的仿射运动模型。例如,当应用仿射合并模式时,可以将当前块的仿射运动模型确定为4参数运动模型。另一方面,当应用仿射运动矢量预测模式时,可以对用于确定当前块的仿射运动模型的信息进行编码并且可以通过比特流用信号发送。例如,在对当前块应用仿射运动矢量预测模式时,可以基于1比特标志“affine_type_flag”来确定当前块的仿射运动模型。

[0141] 接下来,可以导出当前块的仿射种子矢量(S802)。当选择4参数仿射运动模型时,可以导出当前块的两个控制点上的运动矢量。另一方面,当选择6参数仿射运动模型时,可以导出当前块的三个控制点上的运动矢量。可以将控制点上的运动矢量称为仿射种子矢量。控制点可以包括当前块的左上角、右上角或左下角中的至少一个。

[0142] 图9是示出每个仿射运动模型的仿射种子矢量的示例的图。

[0143] 在4参数仿射运动模型中,可以导出与左上角、右上角或左下角中的两个相关的仿射种子矢量。例如,如图9(a)部分所示的示例,当选择4参数仿射运动模型时,可以通过使用与当前块的左上角(例如,左上样本(x0,y0))相关的仿射种子矢量sv0和与当前块的右上角(例如,右上样本(x1,y1))相关的仿射种子矢量sv1来导出仿射矢量。还可以使用与左下角相关的仿射种子矢量来代替与左上角相关的仿射种子矢量,或者还可以使用与左下角相关的仿射种子矢量来代替与右上角相关的仿射种子矢量。

[0144] 在6参数仿射运动模型中,可以导出与左上角、右上角及左下角相关的仿射种子矢量。例如,如图9(b)部分所示的示例,当选择6参数仿射运动模型时,可以通过使用与当前块的左上角(例如,左上样本(x0,y0))相关的仿射种子矢量sv0、与当前块的右上角(例如,右上样本(x1,y1))相关的仿射种子矢量sv1及与当前块的左下角(例如,左下样本(x2,y2))相关的仿射种子矢量sv2来导出仿射矢量。

[0145] 在稍后所述的实施例中,在4参数仿射运动模型下,将左上控制点及右上控制点的仿射种子矢量分别称为第一仿射种子矢量及第二仿射种子矢量。在稍后所述的使用第一仿射种子矢量及第二仿射种子矢量的实施例中,第一仿射种子矢量及第二仿射种子矢量中的至少一个可替换为左下控制点的仿射种子矢量(第三仿射种子矢量)或右下控制点的仿射种子矢量(第四仿射种子矢量)。

[0146] 并且,在6参数仿射运动模型下,将左上控制点、右上控制点及左下控制点的仿射种子矢量分别称为第一仿射种子矢量、第二仿射种子矢量及第三仿射种子矢量。在稍后所述的使用第一仿射种子矢量、第二仿射种子矢量及第三仿射种子矢量的实施例中,第一仿射种子矢量、第二仿射种子矢量及第三仿射种子矢量中的至少一个可替换为右下控制点的仿射种子矢量(第四仿射种子矢量)。

[0147] 可以使用仿射种子矢量来按不同子块导出仿射矢量(S803)。其中,仿射矢量表示基于仿射种子矢量导出的平移运动矢量。可以将子块的仿射矢量称为仿射子块运动矢量或子块运动矢量。

[0148] 图10是示出在4参数运动模型下的子块的仿射矢量的示例的图。

[0149] 可以基于控制点的位置、子块的位置及仿射种子矢量而导出子块的仿射矢量。例如,等式3表示导出仿射子块矢量的示例。

[0150] 【等式3】

[0151] 在所述等式3中, (x, y) 表示子块的位置。其中,子块的位置表示子块中所包括的参考样本的位置。参考样本可以是位于子块的左上角的样本,或者是x轴或y轴坐标中的至少一个位于中心位置的样本。 (x_0, y_0) 表示第一控制点的位置,并且 (sv_0x, sv_0y) 表示第一仿射种子矢量。另外, (x_1, y_1) 表示第二控制点的位置,并且 (sv_1x, sv_1y) 表示第二仿射种子矢量。

[0152] 当第一控制点及第二控制点分别与当前块的左上角及右上角相对应时,可以将 $x_1 - x_0$ 设置为与当前块的宽度相同的值。

[0153] 之后,可以使用各个子块的仿射矢量来对各个子块执行运动补偿预测(S804)。在执行运动补偿预测后,可以生成与各个子块相关的预测块。可以将子块的预测块设置为当前块的预测块。

[0154] 接下来,详细说明使用平移运动信息的帧间预测方法。

[0155] 可从当前块的其他块的运动信息中导出当前块的运动信息。其中,其他块可以是比起当前块更优先地以帧间预测进行编码/解码的块。可以将当前块的运动信息设置为与其他块的运动信息相同的情况定义为合并模式。并且,可以将其他块的运动矢量设置为当前块的运动矢量的预测值情况定义为运动矢量预测模式。

[0156] 图11是在合并模式下导出当前块的运动信息的过程的流程图。

[0157] 可以导出当前块的合并候选(S1101)。当前块的合并候选可从在当前块之前以帧间预测进行编码/解码的块导出。

[0158] 图12是示出了用于导出合并候选的候选块的示例的图。

[0159] 候选块可以包括包含与当前块相邻的样本的相邻块或包含不与当前块相邻的样本的非相邻块中的至少一者。在下文中,将用于确定候选块的样本指定为参考样本。另外,将与当前块相邻的参考样本称为相邻参考样本,将不与当前块相邻的参考样本称为非相邻参考样本。

[0160] 相邻参考样本可以包括在当前块的最左侧列的相邻列中、或当前块的最上方行的相邻行中。例如,若当前块的左上样本的坐标为 $(0, 0)$, 则包括 $(-1, H-1)$ 位置上的参考样本的块、包括 $(W-1, -1)$ 位置上的参考样本的块、包括 $(W, -1)$ 位置上的参考样本的块、包括 $(-1, H)$ 位置上的参考样本的块或包括 $(-1, -1)$ 位置上的参考样本的块中的至少一个可用作候选块。参考附图,可以将索引为0至4的相邻块用作候选块。

[0161] 非相邻参考样本表示与当前块相邻的参考样本之间的x轴距离或y轴距离中的至少一者具有预定义的值值的样本。例如,可以将包含与左侧参考样本之间的x轴距离为预定义的值值的参考样本的块、包含与上方参考样本之间的y轴距离为预定义的值值的非相邻样本的块、或包含与左上参考样本之间的x轴距离及y轴距离为预定义的值值的非相邻样本的块中的至少一者用作候选块。预定义的值可以是4、8、12、16等整数。参考附图,可以将索引为5至26的块中的至少一个用作候选块。

[0162] 可以将与相邻参考样本不在同一垂直线、水平线或对角线上的样本设置为非相邻参考样本。

[0163] 图13是示出了参考样本的位置的图。

[0164] 如图13所示的示例,可以将上方非相邻参考样本的x坐标设置为与上方相邻参考样本的x坐标不同。例如,当上相邻参考样本的位置为 $(W-1, -1)$ 时,可以将沿y轴与上相邻参考样本相隔N的上非相邻参考样本的位置设置为 $((W/2) - 1, -1-N)$,并且可以将沿y轴与上相邻参考样本相隔2N的上非相邻参考样本的位置设置为 $(0, -1-2N)$ 。即,非相邻参考样本的位置可以基于相邻参考样本的位置和与相邻参考样本之间的距离来确定。

[0165] 在下文中,将候选块中的包含相邻参考样本的候选块称为相邻块,包含非相邻参考样本的块称为非相邻块。

[0166] 在当前块与候选块之间的距离大于或等于阈值时,可以将所述候选块设置为不可用作合并候选。所述阈值可以基于编码树单元的大小来确定。例如,可以将所述阈值设置为编码树单元的高度(ctu_height)、或在编码树单元的高度加上或减去偏移值而得的值(例如, $ctu_height \pm N$)。偏移值N为编码器及解码器中预定义的值,可以设置为4、8、16、32或ctu_height。

[0167] 在当前块的y轴坐标与候选块中包括的样本的y轴坐标之间的差值大于阈值时,可以将候选块确定为不可用作合并候选。

[0168] 可替代地,可以将不属于与当前块相同的编码树单元的候选块设置为不可用作合并候选。例如,在参考样本超出当前块所属的编码树单元的上边界时,可以将包括参考样本的候选块设置为不可用作合并候选。

[0169] 如果,在当前块的上边界与编码树单元的上边界相邻时,多个候选块被确定为不可用作合并候选,由此会降低当前块的编码/解码效率。为了解除如上问题,可以设置候选块,以使位于当前块的上方的候选块的数量多于位于当前块的左侧的候选块的数量。

[0170] 图14是示出了用于导出合并候选的候选块的示例的图。

[0171] 如图14所示的示例,可以将属于当前块的上方N个块列的上方块和属于当前块的左侧M个块列的左侧块设置为候选块。此时,通过将M设置为大于N,可以将左侧候选块的数量设置为大于上方候选块的数量。

[0172] 例如,可以将当前块内参考样本的y轴坐标与可用作候选块的上方块的y轴坐标之间的差值设置为不超过当前块的高度的N倍。另外,可以将当前块内参考样本的x轴坐标和可用作候选块的左侧块的x轴坐标之间的差值设置为不超过当前块的宽度的M倍。

[0173] 例如,如图14所示的示例中,示出了将属于当前块的上方两个块列的块和属于当前块的左侧五个块列的块设置为候选块。

[0174] 作为另一例,当候选块不属于与当前块相同的编码树单元时,可以代替所述候选块而利用属于与当前块相同的编码树单元的块、或包含与所述编码树单元的边界相邻的参考样本的块来导出合并候选。

[0175] 图15是示出了改变参考样本的位置的示例的图。

[0176] 当参考样本包括在与当前块不同的编码树单元,并且所述参考样本与所述编码树单元的边界不相邻时,可以代替所述参考样本而利用与所述编码树单元的边界相邻的参考样本来确定候选块参考样本。

[0177] 例如,在图15(a)和图15(b)所示的示例中,在当前块的上边界与编码树单元的上边界彼此接触时,当前块上方的参考样本属于与当前块不同的编码树单元。可以用与编码

树单元的上边界相邻的样本来替换属于与当前块不同的编码树单元的参考样本中不与编码树单元的上边界相邻的参考样本。

[0178] 例如,如图15(a)的示例所示,位置6处的参考样本被替换为位于编码树单元的上边界的位置6'处的样本,如图15(b)的示例所示,位置15处的参考样本被替换为位于编码树单元的上边界的位置15'处的样本。此时,替换样本的y坐标可以改变为编码树单元的相邻位置,替换样本的x坐标可以设置为与参考样本相同。例如,6'位置的样本可以具有与6位置的样本相同的x坐标,15'位置的样本可以具有与15位置的样本相同的x坐标。

[0179] 可替代地,可以将在参考样本的x坐标中加上或减去偏移值而得到的值设置为替换样本的x坐标。例如,当位于当前块的上方的相邻参考样本和非相邻参考样本的x坐标相同时,可以将在参考样本的x坐标中加上或减去偏移值而得到的值设置为替换样本的x坐标。这是为了防止用于替换非相邻参考样本的替换样本与其他非相邻参考样本或相邻参考样本处于相同的位置。

[0180] 图16是示出改变参考样本的位置的示例的图。

[0181] 在用位于编码树单元的边界的样本来替换包括在与当前块不同的编码树单元中且与编码树单元的边界不相邻的参考样本时,可以将在参考样本的x坐标中加上或减去偏移值而得到的值设置为替换样本的x坐标。

[0182] 例如,在图16所示的示例中,位置6处的参考样本和位置15处的参考样本可以被分别替换为具有同与编码树单元的上边界相邻的行相同的y坐标的、位置6'处的样本和位置15'处的样本。此时,6'位置的样本的x坐标可以设置为与6位置的参考样本的x坐标的差值为 $W/2$ 的值,15'位置的样本的x坐标可以设置为与15位置的参考样本的x坐标的差值为 $W-1$ 的值。

[0183] 不同于图15和图16所示的示例,位于当前块的最上方行的上方的行的y坐标或编码树单元的上边界的y坐标还可以被设置为替换样本的y坐标。

[0184] 虽然未图示,但是还可以基于编码树单元的左边界来确定代替参考样本的样本。例如,当参考样本不包括在与当前块相同的编码树单元中,并且不与编码树单元的左边界相邻时,可以将所述参考样本替换为与编码树单元的左边界相邻的样本。此时,替换样本可以具有与参考样本相同的y坐标,或者可以具有在参考样本的y坐标中加上或差值偏移值而获取的y坐标。

[0185] 之后,可以将包含替换样本的块设置为候选块,并基于所述候选块导出当前块的合并候选。

[0186] 还可从与当前块不同的图像中所包括的时间相邻块导出合并候选。例如,可从同位图像中所包括的同位块导出合并候选。

[0187] 可以将合并候选的运动信息设置为与候选块的运动信息相同。例如,可以将候选块的运动矢量、参考图像索引、预测方向或双向加权值索引中的至少一个设置为合并候选的运动信息。

[0188] 可以生成包括合并候选的合并候选列表(S1102)。所述合并候选可以分类为从与当前块相邻的相邻块中导出的相邻合并候选、以及从非相邻块中导出的非相邻合并候选。

[0189] 可按照预定顺序分配合并候选列表内的多个合并候选的索引。例如,分配给相邻合并候选的索引可以具有比分配给非相邻合并候选的索引更小的值。可替代地,基于图12

或图14所示的每个块的索引,可以将索引分配给每个合并候选。

[0190] 当合并候选列表中包括多个合并候选时,可以选择所述多个合并候选中的至少一个(S1103)。此时,可以通过比特流用信号发送指示当前块的运动信息是否是从相邻合并候选中导出的信息。所述信息可以是1比特标志。例如,可以通过比特流用信号发送指示当前块的运动信息是否是从相邻合并候选中导出的语法元素isAdjacentMergeFlag。当语法元素isAdjacentMergeFlag的值为1时,当前块的运动信息可以基于相邻合并候选来导出。另一方面,当语法元素isAdjacentMergeFlag的值为0时,当前块的运动信息可以基于非相邻合并候选来导出。

[0191] 表1示出了包括语法元素isAdjacentMergeFlag的语法表。

[0192] 【表1】

```

coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {  描述符
    if( slice_type != I ) {
        pred_mode_flag ae(v)
    }
    if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {
        if( treeType == SINGLE_TREE | | treeType ==
[0193] DUAL_TREE_LUMA ) {
            intra_luma_mpm_flag[ x0 ][ y0 ]
            if( intra_luma_mpm_flag[ x0 ][ y0 ] )
                intra_luma_mpm_idx[ x0 ][ y0 ] ae(v)
            else
                intra_luma_mpm_remainder[ x0 ][ y0 ] ae(v)
        }
        if( treeType == SINGLE_TREE | | treeType ==
DUAL_TREE_CHROMA )
            intra_chroma_pred_mode[ x0 ][ y0 ] ae(v)
    }
}

```

```

    } else {
        if (cu_skip_falg[x0][y0]) {
            if (MaxNumMergeCand > 1){
                isAdjacentMergeflag ae(v)
                if (isAdjcanetMergeflag){
                    merge_idx[x0][y0] ae(v)
                } else{
                    NA_merge_idx[x0][y0] ae(v)
                }
            }
        } else { /* MODE_INTER*/
            merge_flag[x0][y0] ae(v)
            if (merge_flag[x0][y0]){
[0194]         if (MaxNumMergeCand > 1){
                    isAdjacentMergeflag ae(v)
                    if (isAdjcanetMergeflag){
                        merge_idx[x0][y0] ae(v)
                    } else{
                        NA_merge_idx[x0][y0] ae(v)
                    }
                }
            }
        }
    }
    if( CuPredMode[ x0 ][ y0 ] != MODE_INTRA )
        cu_cbf ae(v)
    if( cu_cbf ) {
        transform_tree( x0, y0, cbWidth, cbHeight, treeType )
    }
}

```

[0195] 可以通过比特流用信号发送用于指定多个合并候选中的任一个的信息。例如，可以通过比特流用信号发送指示合并候选列表中包括的合并候选中的任一个索引的信息。

[0196] 当isAdjacentMergeflag为1时，可以用信号发送用于确定相邻合并候选中的任一个的语法元素merge_idx。可以将语法元素merge_idx的最大值设置为与相邻合并候选的数量的差值为1的值。

[0197] 当isAdjacentMergeflag为0时，可以用信号发送用于确定非相邻合并候选中的任一个的语法元素NA_merge_idx。语法元素NA_merge_idx指示将非相邻合并候选的索引与相邻合并候选的数量求差而获得的值。解码器可以通过将根据NA_merge_idx确定的索引加上相邻合并候选的数量来选择非相邻合并候选。

[0198] 在合并候选列表所包括的合并候选的数量小于阈值时，可以将帧间运动信息列表中所包括的合并候选添加到合并候选列表中。其中，阈值可以从合并候选列表可以包括的最大合并候选的数量或最大合并候选的数量减去偏移量的值。偏移量可以是1或2等整数。帧间运动信息列表可以包括基于在当前块之前进行编码/解码的块来导出的合并候选。

[0199] 帧间运动信息列表包括从于当前图像内基帧间预测进行编码/解码的块导出的合

并候选。例如，帧间运动信息列表中包括的合并候选的运动信息可以设置为与基于帧间预测进行编码/解码的块的运动信息相同。其中，运动信息可以包括运动矢量、参考图像索引、预测方向或双向加权值索引中的至少一个。为了便于说明，将帧间运动信息列表中包括的合并候选称为帧间合并候选。

[0200] 当选择当前块的合并候选时，所选合并候选的运动矢量被设置为初始运动矢量，并且可以使用通过将初始运动矢量加上或减去偏移矢量导出的运动矢量来执行当前块的运动补偿预测。通过将合并候选的运动矢量加上或减去偏移矢量来导出新的运动矢量可以被定义为合并运动差值编码方法。

[0201] 可以通过比特流用信号发送指示是否使用合并偏移编码方法的信息。所述信息可以是1位的标志merge_offset_vector_flag。例如，merge_offset_vector_flag的值为1指示将合并运动差值编码方法应用于当前块。当将合并运动差值编码方法应用于当前块时，可以通过将合并候选的运动矢量加上或减去偏移矢量来导出当前块的运动矢量。merge_offset_vector_flag的值为0指示未将合并运动差值编码方法应用于当前块。当未应用合并偏移编码方法时，可以将合并候选的运动矢量设置为当前块的运动矢量。

[0202] 所述标志仅当指示是否应用跳过模式的跳过标志的值为真或指示是否应用合并模式的合并标志的值为真时才可以用信号发送。例如，当指示是否将跳过模式应用于当前块的skip_flag的值是1时，或者当指示是否将合并模式应用于当前块的merge_flag的值是1时，可以对merge_offset_vector_flag进行编码并用信号发送。

[0203] 当确定将合并偏移编码方法应用于当前块时，可以另外用信号发送指定合并候选列表中包括的合并候选中的任何一个的信息、指示偏移矢量的大小的信息、以及指示偏移矢量的方向的信息中的至少一项。

[0204] 可以通过比特流来用信号发送用于确定合并候选列表可以包括的合并候选的最大数量的信息。例如，可以将合并候选列表可以包括的合并候选的最大数量设置为6或更小的整数。

[0205] 当确定将合并偏移编码方法应用于当前块时，仅预设的最大数量的合并候选可以被设置为当前块的初始运动矢量。即，取决于是否应用合并偏移编码方法，可以自适应地确定当前块可以使用的合并候选的数量。例如，当merge_offset_vector_flag的值设置为0时，当前块可以使用的合并候选的最大数量可以设置为M，而当merge_offset_vector_flag的值设置为1时，当前块可以使用的合并候选的最大数量可以设置为N。在此，M表示合并候选列表可以包括的合并候选的最大数量，并且N表示等于或小于M的整数。

[0206] 例如，当M为6并且N为2时，可以将合并候选列表中包括的合并候选中具有最小索引的两个合并候选设置为可用于当前块。因此，可以将索引值为0的合并候选的运动矢量或索引值为1的合并候选的运动矢量设置为当前块的初始运动矢量。当M和N相同时（例如，当M和N为2时），可以将合并候选列表中包括的所有合并候选设置为可用于当前块。

[0207] 可替代地，可以基于是否将合并运动差值编码方法应用于当前块来确定是否可以将相邻块用作合并候选。例如，当merge_offset_vector_flag的值为1时，与当前块的右上角相邻的相邻块、与左下角相邻的相邻块、以及与左下角相邻的相邻块中的至少一个可以被设置为不可用作合并候选。因此，当将合并运动差值编码方法应用于当前块时，与当前块的右上角相邻的相邻块、与左下角相邻的相邻块、以及与左下角相邻的相邻块中的至少一

个的运动矢量不能被设置为初始运动矢量。可替代地,当merge_offset_vector_flag的值是1时,当前块的时间相邻块可以被设置为不可用作为合并候选。

[0208] 当将合并运动差值编码方法应用于当前块时,可以设置为不使用成对合并候选和零合并候选中的至少一个。因此,当merge_offset_vector_flag的值是1时,尽管合并候选列表中包括的合并候选的数量小于最大数量,也可能不会将成对合并候选或零合并候选中的至少一个添加到合并候选列表。

[0209] 合并候选的运动矢量可以被设置为当前块的初始运动矢量。在这种情况下,当当前块可以使用的合并候选的数量为多个时,可以通过比特流用信号发送指定多个合并候选中的任何一个的信息。例如,当合并候选列表可以包括的合并候选的最大数量大于1时,可以通过比特流用信号发送指示多个合并候选中的任何一个的信息merge_idx。即,在合并偏移编码方法下,可以通过用于指定多个合并候选中的任何一个的信息merge_idx来指定合并候选。当前块的初始运动矢量可以被设置为由merge_idx指示的合并候选的运动矢量。

[0210] 另一方面,当当前块可以使用的合并候选的数量为1时,可以省略用信号发送用于指定合并候选的信息。例如,当合并候选列表可以包括的合并候选的最大数量不大于1时,可以省略用信号发送用于指定合并候选的信息merge_idx。即,在合并偏移编码方法下,当在合并候选列表中包括一个合并候选时,可以省略对用于指定合并候选的信息merge_idx的编码,并且可以基于合并候选列表中包括的合并候选来确定初始运动矢量。合并候选的运动矢量可以被设置为当前块的初始运动矢量。

[0211] 作为另一示例,在确定当前块的合并候选之后,可以确定是否将合并运动差值编码方法应用于当前块。例如,当合并候选列表可以包括的合并候选的最大数量大于1时,可以用信号发送用于指定合并候选中的任何一个的信息merge_idx。在基于merge_idx选择合并候选之后,可以对指示是否将合并运动差值编码方法应用于当前块的merge_offset_vector_flag进行解码。表2是示出根据上述实施例的语法表。

[0212] 【表2】

```
coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {  描述符
    if( slice_type != I ) {
        cu_skip_flag[ x0 ][ y0 ]  ae(v)
[0213]     if( cu_skip_flag[ x0 ][ y0 ] == 0 )
        pred_mode_flag  ae(v)
    }
    if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {
```

```

        if( treeType == SINGLE_TREE | | treeType ==
DUAL_TREE_LUMA ) {
            intra_luma_mpm_flag[ x0 ][ y0 ]
            if( intra_luma_mpm_flag[ x0 ][ y0 ] )
                intra_luma_mpm_idx[ x0 ][ y0 ] ae(v)
            else
                intra_luma_mpm_remainder[ x0 ][ y0 ] ae(v)
        }
        if( treeType == SINGLE_TREE | | treeType ==
DUAL_TREE_CHROMA )
            intra_chroma_pred_mode[ x0 ][ y0 ] ae(v)
    } else { /* MODE_INTER */
        if( cu_skip_flag[ x0 ][ y0 ] ) {
            if( merge_affine_flag[ x0 ][ y0 ] == 0 && MaxNumMergeCand >
1 ) {
                merge_idx[ x0 ][ y0 ] ae(v)
                merge_offset_vector_flag ae(v)
                if (merge_idx < 2 && merge_offset_vector_flag) {
                    distance_idx[ x0 ][ y0 ] ae(v)
                    direction_idx[ x0 ][ y0 ] ae(v)
                }
            }
        } else {
            merge_flag[ x0 ][ y0 ] ae(v)
            if( merge_flag[ x0 ][ y0 ] ) {
                if( merge_affine_flag[ x0 ][ y0 ] == 0 && MaxNumMergeCand >
1 ) {
                    merge_idx[ x0 ][ y0 ] ae(v)
                    merge_offset_vector_flag ae(v)
                    if (merge_idx < 2 && merge_offset_vector_flag) {
                        distance_idx[ x0 ][ y0 ] ae(v)
                        direction_idx[ x0 ][ y0 ] ae(v)
                    }
                }
            }
        } else {
            if( slice_type == B )
                inter_pred_idc[ x0 ][ y0 ] ae(v)
            if( sps_affine_enabled_flag && cbWidth >= 16 &&
cbHeight >= 16 ) {
                inter_affine_flag[ x0 ][ y0 ] ae(v)
                if( sps_affine_type_flag && inter_affine_flag[ x0 ][ y0 ] )
                    cu_affine_type_flag[ x0 ][ y0 ] ae(v)
            }
        }
    }
}

```

[0215] 作为另一示例,在确定当前块的合并候选之后,仅当所确定的合并候选的索引小于在应用合并运动差值编码方法时可以使用的合并候选的最大数量时,才可以确定是否将合并运动差值编码方法应用于当前块。例如,仅当索引信息merge_idx的值小于N时,才可以对指示是否对当前块应用合并运动差值编码方法的merge_offset_vector_flag进行编码

并用信号发送。当索引信息merge_idx的值等于或大于N时,可以省略对merge_offset_vector_flag的编码。如果省略了对merge_offset_vector_flag编码,则可以确定未将合并运动差值编码方法应用于当前块。

[0216] 可替代地,在确定当前块的合并候选之后,可以考虑所确定的合并候选具有双向运动信息还是单向运动信息来确定是否将合并运动差值编码方法应用于当前块。例如,仅当索引信息merge_idx的值小于N并且由索引信息选择的合并候选具有双向运动信息时,才对表示是否将合并运动差值编码方法应用于当前块的merge_offset_vector_flag进行编码并用信号发送。可选地,仅当索引信息merge_idx的值小于N并且由索引信息选择的合并候选具有单向运动信息时,才对表示是否将合并运动差值编码方法应用于当前块的merge_offset_vector_flag进行编码并用信号发送。

[0217] 可替代地,可以基于当前块的大小、当前块的形状、以及当前块是否与编码树单元的边界接触中的至少一项来确定是否应用合并运动差值编码方法。当当前块的大小、当前块的形状、以及当前块是否与编码树单元的边界接触中的至少一项不满足预设条件时,可以省略对指示是否将合并运动差值编码方法应用于当前块的merge_offset_vector_flag的编码。

[0218] 当选择合并候选时,可以将合并候选的运动矢量设置为当前块的初始运动矢量。然后,可以对指示偏移矢量的大小的信息和指示偏移矢量的方向的信息进行解码以确定偏移矢量。偏移矢量可以具有水平方向分量或垂直方向分量。

[0219] 指示偏移矢量的大小的信息可以是指示运动大小候选中的任何一个的索引信息。例如,可以通过比特流用信号发送指示运动大小候选中的任何一个的索引信息distance_idx。表3示出了索引信息distance_idx的二值化以及用于根据distance_idx确定偏移矢量的大小的变量DistFromMergeMV的值。

[0220] 【表3】

	distance_idx[x][y] 二值化	DistFromMergeMV[x0][y0]
	0 0 1	
	1 10 2	
	2 1104	
[0221]	3 1110 8	
	4 11110 16	
	5 11111032	
	6 1111110 64	
	7 1111111 128	

[0222] 可以通过将变量DistFromMergeMV除以预设值来导出偏移矢量的大小。等式4示出了确定偏移矢量的大小的示例。

[0223] 【等式4】

[0224] 根据等式4,可以将通过将变量DistFromMegeMV除以4而获得的值或通过将变量DistFromMergeMV向左移位2而获得的值设置为偏移矢量的大小。

[0225] 可以使用与表3所示的示例相比更多的运动大小候选或更少的运动大小候选,或者可以将运动矢量偏移大小候选的范围设置为与表3所示的示例不同。例如,可以将偏移矢

量的水平方向分量或垂直方向分量的大小设置为不大于2个样本距离。表4示出了索引信息distance_idx的二值化以及用于根据distance_idx确定偏移矢量的大小的变量DistFromMergeMV的值。

[0226] 【表4】

	distance_idx[x][y] 二值化	DistFromMergeMV[x0][y0]
	0 0 1	
[0227]	1 10 2	
	2 1104	
	3 1118	

[0228] 可替代地,可以基于运动矢量精度来不同地设置运动矢量偏移大小候选的范围。例如,当当前块的运动矢量精度为分数像素(fractional-pel)时,可以将与索引信息distance_idx的值相对应的变量DistFromMergeMV的值设置为1、2、4、8、16等。在此,分数像素包括1/16像素、八分之一像素、四分之一像素或半像素中的至少一个。另一方面,当当前块的运动矢量精度是整数像素时,可以将与索引信息distance_idx的值相对应的变量DistFromMergeMV的值设置为4、8、16、32、64等。即,可以根据当前块的运动矢量精度不同地设置被用来确定变量DistFromMergeMV的表。

[0229] 例如,当当前块或合并候选的运动矢量精度是四分之一像素时,可以使用表3导出由distance_idx表示的变量DistFromMergeMV。另一方面,当当前块或合并候选的运动矢量精度是整数像素时,可以通过将表3中distance_idx指示的变量DistFromMergeMV的值取N倍(例如4倍)而导出的值作为变量DistFromMergeMV的值。

[0230] 可以通过比特流用信号发送用于确定运动矢量精度的信息。例如,可以以序列级、图像级、条带(slice)级或块级用信号发送信息。因此,可以根据通过比特流用信号发送的与运动矢量精度有关的信息来不同地设置运动大小候选的范围。可替代地,可以基于当前块的合并候选来确定运动矢量精度。例如,可以将当前块的运动矢量精度设置为与合并候选的运动矢量精度相同。

[0231] 可替代地,可以通过比特流用信号发送用于确定偏移矢量的搜索范围的信息。可以基于搜索范围来确定运动大小候选的数量、运动大小候选中的最小值、以及运动大小候选中的最大值中的至少一个。例如,可以通过比特流来用信号发送用于确定偏移矢量的搜索范围的标志merge_offset_vector_flag。所述信息可以通过序列报头、图像报头或条带头来用信号发送。

[0232] 例如,当merge_offset_extend_range_flag的值为0时,可以将偏移矢量的大小设置为不超过2。因此,DistFromMergeMV的最大值可以设置为8。另一方面,当merge_offset_extend_range_flag的值是1时,可以将偏移矢量的大小设置为不超过32个样本距离。因此,DistFromMergeMV的最大值可以设置为128。

[0233] 可以使用指示偏移矢量的大小是否大于阈值的标志来确定偏移矢量的大小。例如,可以通过比特流用信号发送指示偏移矢量的大小是否大于阈值的标志distance_flag。阈值可以是1、2、4、8或16。例如,distance_flag为1指示偏移矢量的大小大于4。另一方面,distance_flag为0指示偏移矢量的大小为4或更小。

[0234] 当偏移矢量的大小大于阈值时,可以使用索引信息distance_idx来导出偏移矢量

的大小与阈值之间的差值。可替代地,当偏移矢量的大小小于或等于阈值时,可以通过使用索引信息distance_idx来确定偏移矢量的大小。表5是示出了对distance_flag和distance_idx进行编码的过程的语法表。

[0235] 【表5】

```

coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {  描述符
    if( slice_type != I ) {
        cu_skip_flag[ x0 ][ y0 ] ae(v)
        if( cu_skip_flag[ x0 ][ y0 ] == 0 )
            pred_mode_flag ae(v)
    }
    if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {
        if( treeType == SINGLE_TREE | | treeType ==
DUAL_TREE_LUMA ) {
            intra_luma_mpm_flag[ x0 ][ y0 ]
            if( intra_luma_mpm_flag[ x0 ][ y0 ] )
                intra_luma_mpm_idx[ x0 ][ y0 ] ae(v)
            else
                intra_luma_mpm_remainder[ x0 ][ y0 ] ae(v)
        }
        if( treeType == SINGLE_TREE | | treeType ==
DUAL_TREE_CHROMA )
            intra_chroma_pred_mode[ x0 ][ y0 ] ae(v)
    } else { /* MODE_INTER */
        if( cu_skip_flag[ x0 ][ y0 ] ) {
            if( merge_affine_flag[ x0 ][ y0 ] == 0 && MaxNumMergeCand >
1 ) {
                merge_idx[ x0 ][ y0 ] ae(v)
                merge_offset_vector_flag ae(v)
                if( merge_idx < 2 && merge_offset_vector_flag ) {
                    distance_flag[ x0 ][ y0 ] ae(v)
                    distance_idx[ x0 ][ y0 ]
                    direction_idx[ x0 ][ y0 ] ae(v)
                }
            }
        }
    }
}

```

```

    }
  }
  } else {
    merge_flag[ x0 ][ y0 ] ae(v)
    if( merge_flag[ x0 ][ y0 ] ) {
      if( merge_affine_flag[ x0 ][ y0 ] == 0 && MaxNumMergeCand >
1 ) {
        merge_idx[ x0 ][ y0 ] ae(v)
        merge_offset_vector_flag ae(v)
        if (merge_idx < 2 && merge_offset_vector_flag) {
          distance_flag[ x0 ][ y0 ] ae(v)
          distance_idx[ x0 ][ y0 ] ae(v)
[0237] direction_idx[ x0 ][ y0 ] ae(v)
        }
      }
    } else {
      if( slice_type == B )
        inter_pred_idc[ x0 ][ y0 ] ae(v)
      if( sps_affine_enabled_flag && cbWidth >= 16 &&
cbHeight >= 16 ) {
        inter_affine_flag[ x0 ][ y0 ] ae(v)
        if( sps_affine_type_flag && inter_affine_flag[ x0 ][ y0 ] )
          cu_affine_type_flag[ x0 ][ y0 ] ae(v)
      }
    }
  }

```

[0238] 等式5示出了使用distance_flag和distance_idx导出用于确定偏移矢量的大小的变量DistFromMergeMV的示例。

[0239] 【等式5】

[0240] 在等式5中,distance_flag的值可以设置为1或0.distance_idx的值可以设置为1、2、4、8、16、32、64、128等。N表示由阈值确定的系数。例如,当阈值是4时,N可以设置为16。

[0241] 指示偏移矢量的方向的信息可以是指示矢量方向候选中的任何一个的索引信息。例如,可以通过比特流用信号发送指示矢量方向候选中的任何一个的索引信息direction_idx。表6示出了索引信息direction_idx的二值化以及根据direction_idx的偏移矢量的方向。

[0242] 【表6】

	direction_idx[x][y]	二值化	sign[x][y][0]	sign[x][y][1]
	0 00	+1 0		
[0243]	1 01	-1 0		
	2 10	0 +1		
	3 11	0 -1		

[0244] 在表6中,sign[0]指示水平方向,并且sign[1]指示垂直方向。+1指示偏移矢量的x分量或y分量的值为正(+),并且-1指示偏移矢量的x分量或y分量的值为负(-)。等式6示出了基于偏移矢量的大小和方向来确定偏移矢量的示例。

[0245] 【等式6】

[0246] 在等式6中,offsetMV[0]指示偏移矢量的垂直方向分量,并且offsetMV[1]指示偏移矢量的水平方向分量。

[0247] 图17是示出根据指示偏移矢量的大小的distance_idx和指示偏移矢量的方向的direction_idx的值的偏移矢量的图。

[0248] 如在图17所示的示例中,可以根据distance_idx和direction_idx的值确定偏移矢量的大小和方向。偏移矢量的最大大小可以被设置为不超过阈值。在此,阈值可以具有由编码器和解码器预定义的值。例如,阈值可以是32个样本距离。可替代地,可以根据初始运动矢量的大小来确定阈值。例如,可以基于初始运动矢量的水平分量的大小设置水平方向的阈值,并且可以基于初始运动矢量的垂直分量的大小设置垂直方向的阈值。

[0249] 当合并候选具有双向运动信息时,可以将合并候选的L0运动矢量设置为当前块的L0初始运动矢量,并且可以将合并候选的L1运动矢量设置为当前块的L1初始运动矢量。在这种情况下,可以考虑到合并候选的L0参考图像与当前图像之间的输出顺序差值(以下称为L0差值)和合并候选的L1参考图像与当前图像之间的输出顺序差值(以下称为L1差值)来确定L0偏移矢量和L1偏移矢量。

[0250] 首先,当L0差值和L1差值的符号相同时,可以将L0偏移矢量和L1偏移矢量设置为相同。另一方面,当L0差值和L1差值的符号不同时,可以在与L0偏移矢量相反的方向上设置L1偏移矢量。

[0251] L0偏移矢量的大小和L1偏移矢量的大小可以设置为相同。可替代地,可以通过基于L0差值和L1差值缩放L0偏移矢量来确定L1偏移矢量的大小。

[0252] 例如,等式7示出当L0差值和L1差值的符号相同时的L0偏移矢量和L1偏移矢量。

[0253] 【等式7】

[0254] 在等式7中,offsetMVL0[0]指示L0偏移矢量的水平方向分量,并且offsetMVL0[1]指示L0偏移矢量的垂直方向分量。offsetMVL1[0]指示L1偏移矢量的水平方向分量,并且offsetMVL1[1]指示L1偏移矢量的垂直方向分量。

[0255] 等式8示出了当L0差值和L1差值的符号不同时的L0偏移矢量和L1偏移矢量。

[0256] 【等式8】

[0257] 也可以定义多于四个的矢量方向候选。表7和表8示出了定义八个矢量方向候选的示例。

[0258] 【表7】

	direction_idx[x][y]	二值化	sign[x][y][0]	sign[x][y][1]
	0 000	+1	0	
	1 001	-1	0	
	2 010	0	+1	
[0259]	3 0110	-1		
	4 100	+1	+1	
	5 101	+1	-1	
	6 110-1	+1		
	7 111-1	-1		

[0260] 【表8】

	direction_idx[x][y]	二值化	sign[x][y][0]	sign[x][y][1]
	0 000	+1	0	
	1 001	-1	0	
	2 010	0	+1	
[0261]	3 0110	-1		
	4 100	+1/2	+1/2	
	5 101	+1/2	-1/2	
	6 110-1/2	+1/2		
	7 111-1/2	-1/2		

[0262] 在表7和表8中,sign[0]和sign[1]的绝对值大于0指示偏移矢量在对角线方向上。当使用表6时,对角线偏移矢量的x轴分量和y轴分量的大小设置为abs(offsetMV),而当使用表7时,对角线偏移矢量的x轴分量和y轴分量的大小设置为abs(offsetMV/2)。

[0263] 图18是示出根据指示偏移矢量的大小的distance_idx和指示偏移矢量的方向的direction_idx的值的偏移矢量的图。

[0264] 图18(a)是应用表6时的示例,并且图18(b)是应用表7时的示例。

[0265] 可以通过比特流用信号发送用于确定矢量方向候选的数量或大小中的至少一个的信息。例如,可以通过比特流用信号发送用于确定矢量方向候选的标志merge_offset_direction_range_flag。可以以序列级、图像级或条带级用信号发送标志。例如,当标志的值为0时,可以使用表6所例示的四个矢量方向候选。另一方面,当标志的值为1时,可以使用表7或表8所例示的八个矢量方向候选。

[0266] 可替代地,可以基于偏移矢量的大小来确定矢量方向候选的数量或大小中的至少一个。例如,当用于确定偏移矢量的大小的变量DistFromMergeMV的值等于或小于阈值时,可以使用表7或表8所例示的八个矢量方向候选。另一方面,当变量DistFromMergeMV的值大于阈值时,可以使用表6所例示的四个矢量方向候选。

[0267] 可替代地,可以基于初始运动矢量的x分量的值MVx和y分量的值MVy确定矢量方向候选的数量或大小中的至少一个。例如,当MVx与MVy之差或差的绝对值小于或等于阈值时,可以使用表7或表8所例示的八个矢量方向候选。另一方面,当MVx和MVy之差或差的绝对值大于阈值时,可以使用表6所例示的四个矢量方向候选。

[0268] 可以通过将偏移矢量加上初始运动矢量来导出当前块的运动矢量。等式9示出了确定当前块的运动矢量的示例。

[0269] **【等式9】**

[0270] 在等式9中,mvL0指示当前块的L0运动矢量,并且mvL1指示当前块的L1运动矢量。mergeMVL0指示当前块的L0初始运动矢量(即,合并候选的L0运动矢量),并且mergeMVL1表示当前块的L1初始运动矢量。[0]指示运动矢量的水平方向分量,并且[1]指示运动矢量的垂直方向分量。

[0271] 帧内预测是使用当前块周边已编码/解码的重建样本来预测当前块。在这种情况下,当前块的帧内预测可以使用应用环内滤波器之前的重建样本。

[0272] 帧内预测技术包括基于矩阵(Matrix)的帧内预测及考虑与周边重建样本的方向性的一般帧内预测。可以通过比特流用信号发送指示当前块的帧内预测技术的信息。所述信息可以是1比特标志。可替代地,基于当前块的位置、大小、形状或相邻块的帧内预测技术

中的至少一个,可以确定当前块的帧内预测技术。例如,当当前块跨过图像边界存在时,当前块设置为不应用基于矩阵的帧内预测。

[0273] 基于矩阵的帧内预测是基于编码器及解码器中已存储的矩阵与当前块周边的重建样本之间的矩阵乘法来获得当前块的预测块的方法。可以通过比特流用信号发送用于指定已存储的多个矩阵中的任一个的信息。解码器可以基于所述信息及当前块的大小来确定用于当前块的帧内预测的矩阵。

[0274] 一般帧内预测是基于非角度帧内预测模式或角度帧内预测模式来获得与当前块相关的预测块的方法。

[0275] 可以通过将原始图像与预测图像相减来导出所导出的残差图像。在这种情况下,将残差图像变更为频域时,即使移除频率分量中的高频率分量,也不会大幅降低视频的主观画质。由此,若将高频率分量的值变小或者将高频率分量的值设置为0,则具有在不引起明显视觉失真的情况下提高压缩效率的效果。反映以上特性,可以对当前块进行变换以将残差图像分解为2维频率分量。可以使用离散余弦变换(DCT,Discrete Cosine Transform)或离散正弦变换(DST,Discrete Sine Transform)等变换技术来执行所述变换。

[0276] 使用DCT或DST变换当前块之后,可以再次对变换的当前块进行变换。在这种情况下,基于DCT或DST的变换可定义为第一变换,再次对应用第一变换的块进行变换的过程可称为第二变换。

[0277] 第一变换可以使用多个变换核候选中的任一个来执行。例如,可以使用DCT2、DCT8或DCT7中的任一个来执行第一变换。

[0278] 针对水平方向及垂直方向还可以使用不同的变换核。还可以通过比特流用信号发送表示水平方向的变换核及垂直方向的变换核的组合的信息。

[0279] 第一变换及第二变换的执行单元会不同。例如,可以对 8×8 块执行第一变换,并且可以对变换的 8×8 块中的大小为 4×4 的子块执行第二变换。在这种情况下,还可以将不执行第二变换的剩余区域的变换系数设置为0。

[0280] 可替代地,可以对 4×4 块执行第一变换,并且可以对包括变换的 4×4 块的大小为 8×8 的区域可执行第二变换。

[0281] 可以通过比特流用信号发送表示是否执行第二变换的信息。

[0282] 在解码器中可执行第二变换的逆变换(第二逆变换),并且可以对其结果执行第一变换的逆变换(第一逆变换)。作为第二逆变换及第一逆变换的执行结果,可以获得当前块的残差信号。

[0283] 量化用于减少块的能量,并且量化过程包括将变换系数除以特定常数的过程。所述常数可由量化参数来导出,并且量化参数可定义为1至63之间的值。

[0284] 若在编码器中执行变换及量化,则解码器可以通过逆量化及逆变换来获得残差块。在解码器中将预测块和残差块加起来,可以获得当前块的重建块。

[0285] 若获得当前块的重建块,则可以通过环内滤波(In-loop filtering)来减少在量化及编码过程中产生的信息丢失。环内滤波器可以包括去块滤波器(Deblocking filter)、样本自适应偏移滤波器(Sample Adaptive Offset filter,SAO)或自适应环路滤波器(Adaptive Loop Filter,ALF)中的至少一个。

[0286] 将以解码过程或编码过程为中心描述的实施例用于编码过程或解码过程也包括

在本发明的范围内。以与所描述的不同的顺序来改变按预定顺序描述的多个实施例也包括在本发明的范围内。

[0287] 已经基于一系列的步骤或流程图来描述实施例,但这并不限定发明的时间序列顺序,并且根据需要可同时执行或者按另外顺序执行。并且,在上述实施例中,构成框图的结构要素(例如,单元、模块等)还可分别实现为硬件设备或软件,并且多个结构要素可以组合实施为单个硬件设备或软件。所述实施例可以通过程序指令的形式实施,所述程序指令可以由各种计算机部件执行并记录在计算机可读记录介质中。所述计算机可读记录介质可单独包括或组合包括程序指令、数据文件、数据结构等。计算机可读记录介质的示例可以包括硬盘、软盘及磁带等磁性介质、CD-ROM、DVD等光学记录介质、如软式光盘(floptical disk)的磁-光介质(magneto-optical media)及ROM、RAM、闪速存储器等等以存储程序指令并执行该指令的方式专门配置的硬件设备。所述硬件设备可以被配置为作为一个或多个软件模块进行操作以执行根据本发明的处理,另一方面亦然。

[0288] **【工业实用性】**

[0289] 本发明可以应用于对视频进行编码/解码的电子设备。

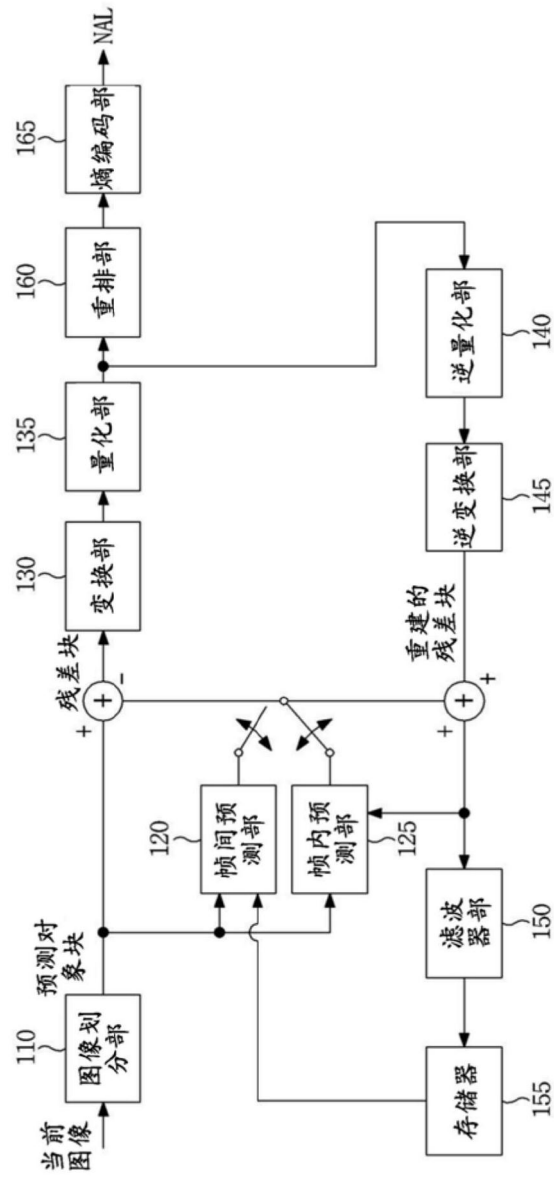


图1

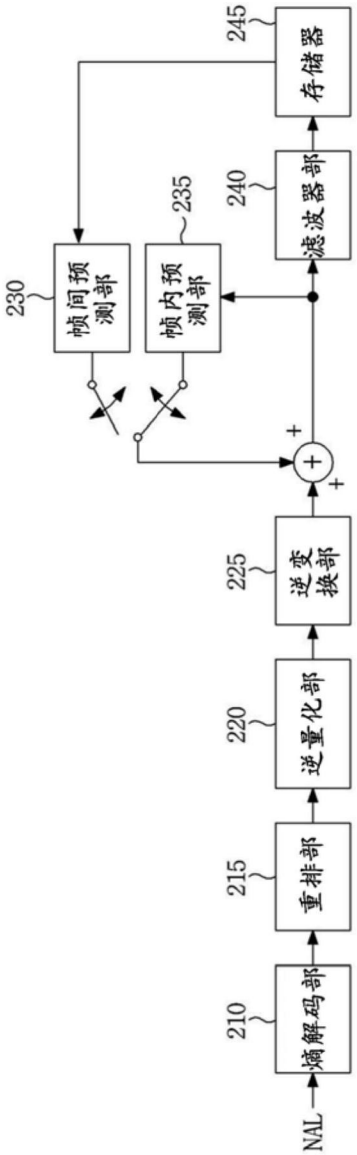


图2

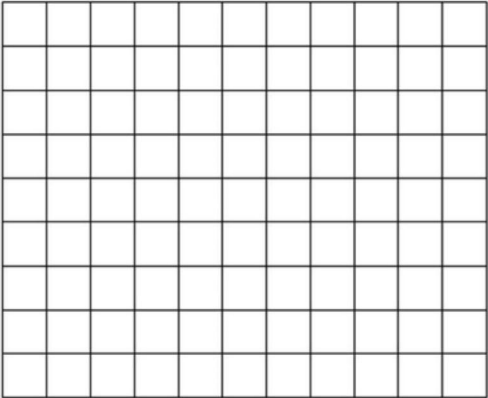


图3

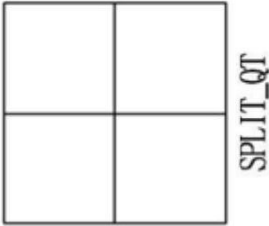


图4(a)

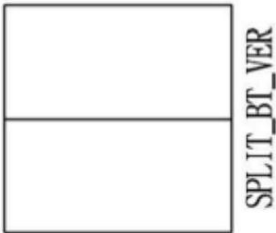


图4(b)

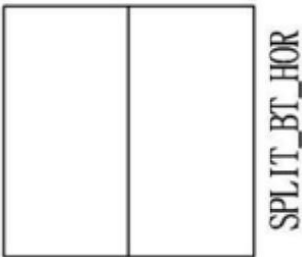


图4(c)

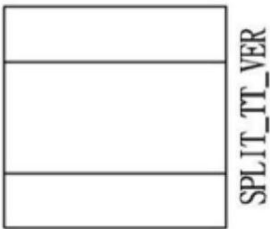


图4(d)

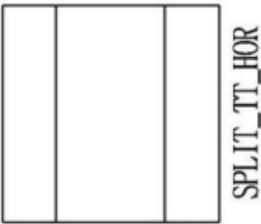


图4(e)

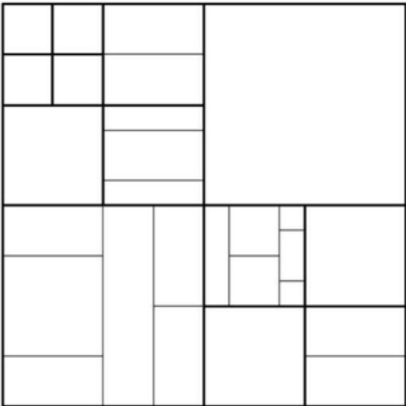


图5

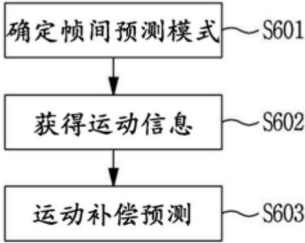


图6

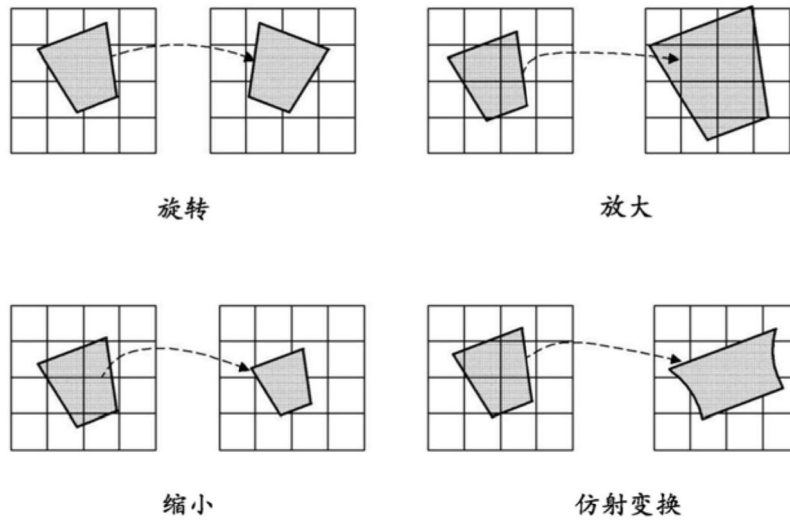


图7

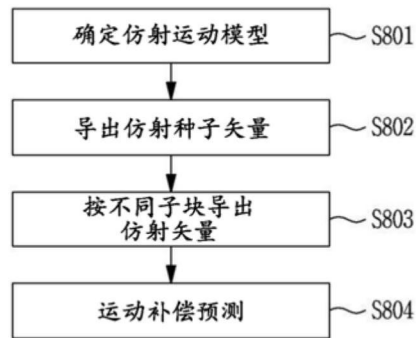


图8

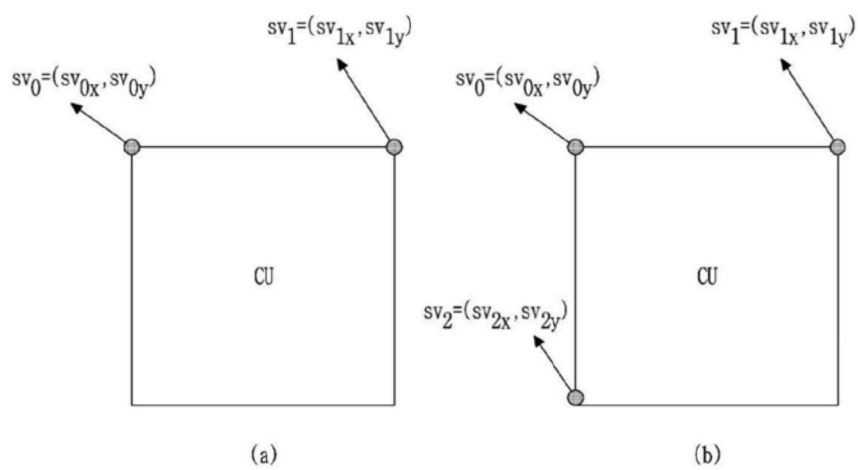


图9

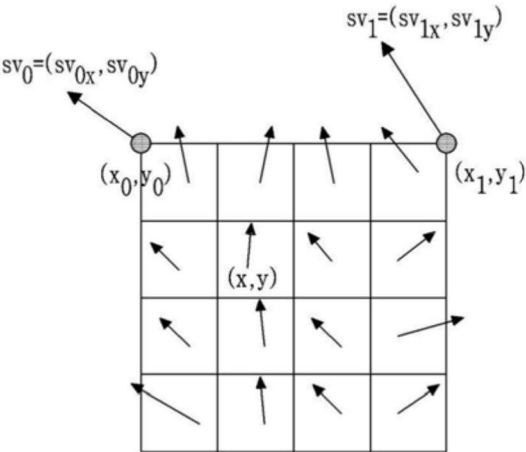


图10

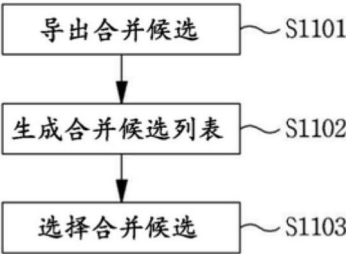


图11

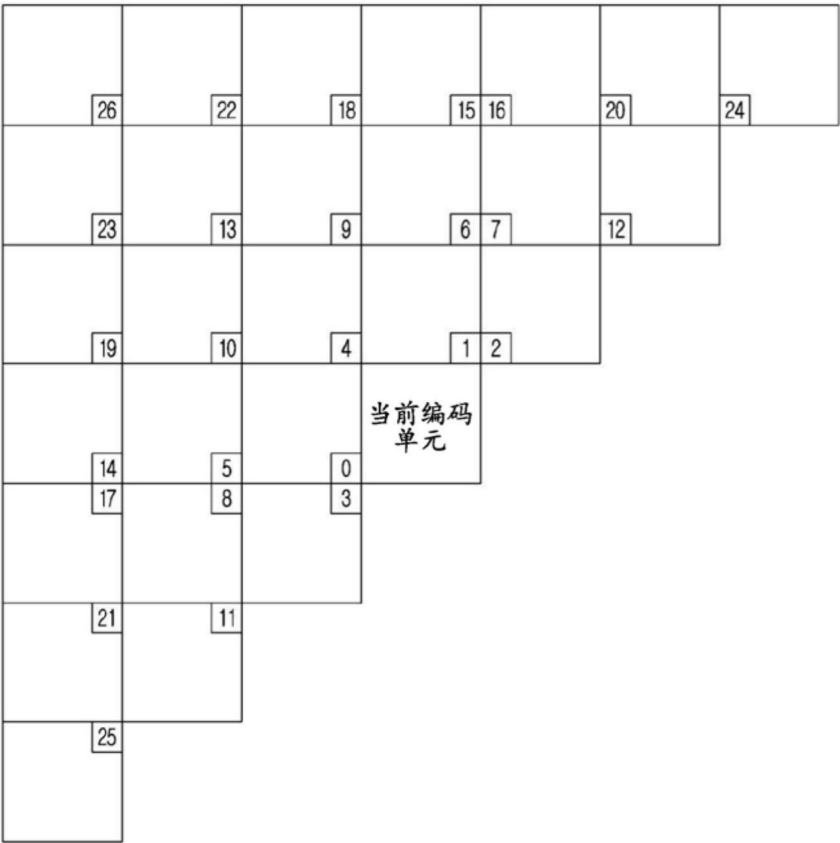


图12

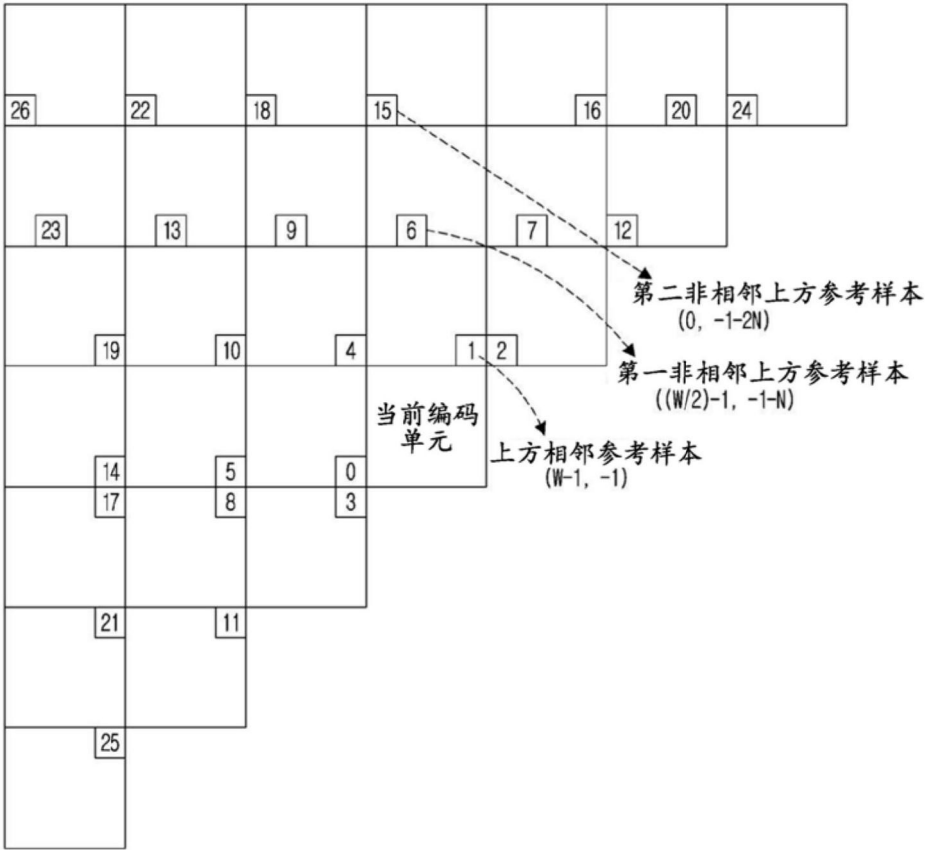


图13

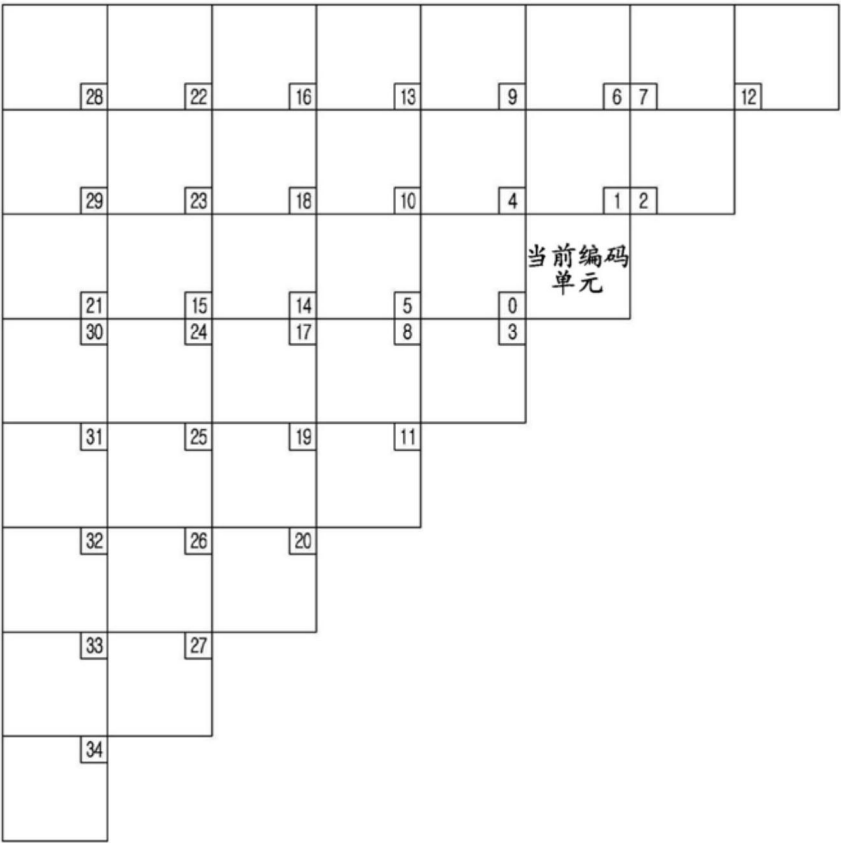


图14

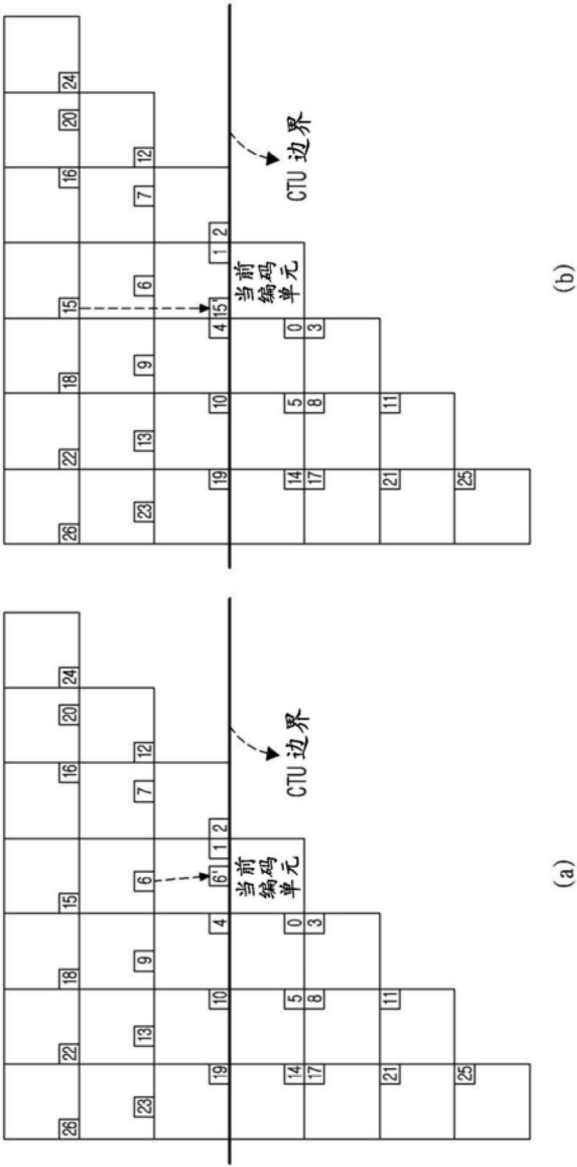


图15

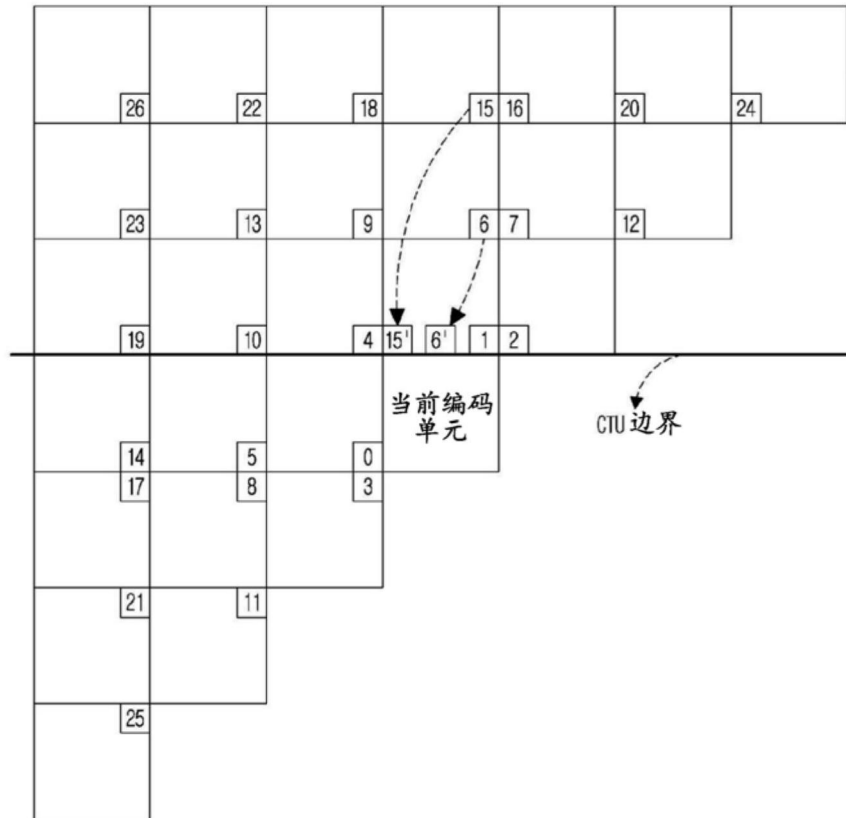


图16

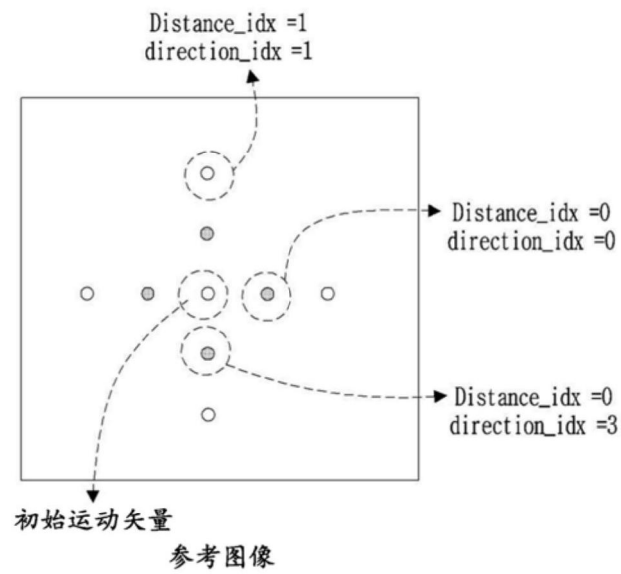


图17

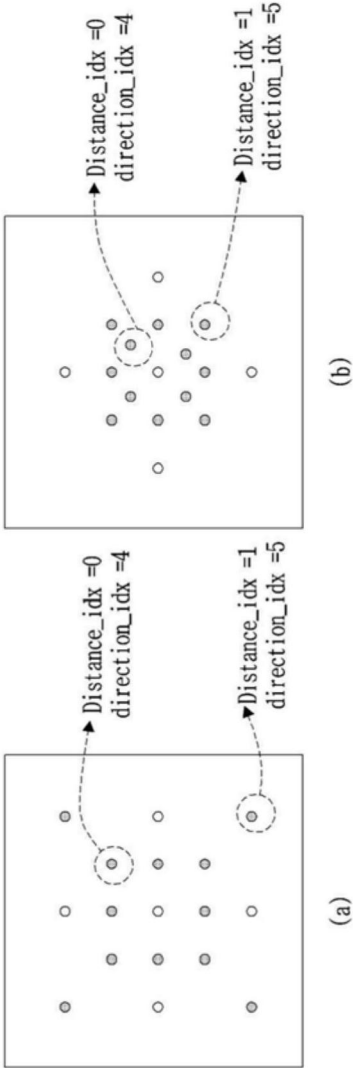


图18