



(12)发明专利

(10)授权公告号 CN 105100797 B

(45)授权公告日 2018.05.29

(21)申请号 201510387296.1

H04N 19/176(2014.01)

(22)申请日 2010.12.09

H04N 19/119(2014.01)

(65)同一申请的已公布的文献号

H04N 19/46(2014.01)

申请公布号 CN 105100797 A

H04N 19/134(2014.01)

(43)申请公布日 2015.11.25

H04N 19/122(2014.01)

(30)优先权数据

H04N 19/61(2014.01)

10-2009-0121935 2009.12.09 KR

H04N 19/11(2014.01)

H04N 19/85(2014.01)

(62)分案原申请数据

(56)对比文件

201080063360.6 2010.12.09

CN 101569201 A, 2009.10.28,

(73)专利权人 三星电子株式会社

CN 101502119 A, 2009.08.05,

地址 韩国京畿道水原市

EP 1841230 A1, 2007.10.03,

(续)

(72)发明人 艾琳娜·阿尔辛娜

审查员 曹珊珊

亚历山大·阿尔辛

范迪姆·谢廖金

尼古雷·谢利亚霍夫

(74)专利代理机构 北京铭硕知识产权代理有限公司

公司 11286

代理人 曾世骁 常桂珍

(51)Int.Cl.

H04N 19/105(2014.01)

权利要求书1页 说明书31页 附图21页

(54)发明名称

对视频解码的设备

(57)摘要

公开了一种对视频编码的方法和设备以及对视频解码的方法和设备。对视频编码的方法包括：产生将被编码的当前编码单元的第一预测编码单元；确定当前编码单元是否包括位于当前画面的边界之外的部分；当当前编码单元不包括位于当前画面的边界之外的部分时，通过使用第一预测编码单元的像素和第一预测编码单元的所述像素的邻近像素来改变第一预测编码单元的所述像素的值，从而产生第二预测编码单元。因此，作为当前编码单元和第二预测编码单元之间的差的残差块可被编码，从而提高视频预测效率。

预测模式	名称
0	垂直
1	水平
2	DC
3	下_左
4	下_右
5	垂直_右
6	水平_下
7	垂直_左
8	水平_上

[转续页]

[接上页]

(56)对比文件

Gianni Vernazza.a new splitting strategy for adaptive transform coding.《Signal Processing》.1993,第34卷(第3期),335-340.

ThomasWiegand等.Overview of the H.264/AVC Video Coding Standard.《IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY》.2003,第13卷(第7期),560-576.

1. 一种对视频解码的设备,所述设备包括:

划分器,基于关于最大编码单元的尺寸的信息将图像划分为多个最大编码单元,并通过基于划分信息对所述多个最大编码单元中的当前最大编码单元进行划分来确定所述多个最大编码单元中的所述当前最大编码单元所包括的至少一个编码单元;

熵解码器,从接收到的比特流提取与所述至少一个编码单元中包括的将被解码的当前块的预测模式有关的信息;

预测器,当提取的信息指示当前块的预测模式为帧内预测时,通过使用当前块的可用邻近像素来确定用于帧内预测的当前块的邻近像素,通过计算与当前块邻近的所述邻近像素的平均值来产生当前块的第一预测值,并通过使用第一预测值以及与当前块邻近的所述邻近像素中的至少一个邻近像素的像素值来产生位于当前块的上边界和左边界的像素的第二预测值,

其中:

当前块的像素的第二预测值是通过使用获得的第一预测值的加权值、与位于当前块的上边界和左边界的像素中的左上像素处于同一列的第一邻近像素以及与所述左上像素处于同一行的第二邻近像素而获得的。

对视频解码的设备

[0001] 本申请是申请日为2010年12月9日、申请号为201080063360.6、题为“对视频编码的方法和设备以及对视频解码的方法和设备”的专利申请的分案申请。

技术领域

[0002] 一个或多个示例性实施例涉及一种能够通过根据预测的视频数据的位置执行后处理来提高视频压缩效率的视频编码方法和设备以及视频解码方法和设备。

背景技术

[0003] 在图像压缩方法(诸如运动图像专家组(MPEG)-1、MPEG-2、MPEG-4或者H.264/MPEG-4先进视频编码(AVC))中,画面被划分为多个宏块以对图像进行编码。以能够在帧间预测或帧内预测中使用的所有编码模式对每个宏块编码,然后以这样的编码模式对所述每个宏块编码:所述编码模式根据用于对宏块编码的比特率和解码的宏块基于原始宏块的失真度而被选择。

[0004] 随着用于再现和存储高分辨率或高质量视频内容的硬件正被开发和提供,越来越需要一种对高分辨率或高质量视频内容进行有效的编码或解码的视频编解码器。在现有技术的视频编解码器中,以多个宏块为单位对视频进行编码,每个宏块具有预定大小。

发明内容

[0005] 技术问题

[0006] 在现有技术的视频编解码器中,以多个宏块为单位对视频进行编码,每个宏块具有有限的大小。

[0007] 技术方案

[0008] 一个或多个示例性实施例提供了一种视频编码方法和设备以及视频解码方法和设备,所述视频编码方法和设备以及视频解码方法和设备通过根据画面中的预测块的位置,经过后处理改变预测块中每个像素的值来产生新的预测块,从而提高视频压缩效率。

[0009] 有益效果

[0010] 根据本发明,提高了编码效率。

附图说明

[0011] 图1是根据示例性实施例的用于对视频编码的设备的框图;

[0012] 图2是根据示例性实施例的用于对视频解码的设备的框图;

[0013] 图3是用于描述根据示例性实施例的编码单元的构思的示图;

[0014] 图4是根据示例性实施例的基于编码单元的图像编码器的框图;

[0015] 图5是根据示例性实施例的基于编码单元的图像解码器的框图;

[0016] 图6是示出根据示例性实施例的根据深度的较深层编码单元以及分区的示图;

[0017] 图7是用于描述根据示例性实施例的编码单元和变换单元之间的关系

- [0018] 图8是用于描述根据示例性实施例的与编码深度对应的编码单元的编码信息的示意图；
- [0019] 图9是根据示例性实施例的根据深度的较深层编码单元的示意图；
- [0020] 图10至图12是用于描述根据示例性实施例的编码单元、预测单元和变换单元之间的关系示意图；
- [0021] 图13是用于描述根据表1的编码模式信息的编码单元、预测单元或分区与变换单元之间的关系示意图；
- [0022] 图14是根据示例性实施例的帧内预测设备的框图；
- [0023] 图15是显示根据示例性实施例的根据编码单元的大小的帧内预测模式的数量表；
- [0024] 图16A至图16C是用于解释根据示例性实施例的可对具有预定大小的编码单元执行的帧内预测模式的示意图；
- [0025] 图17是用于解释根据另外的示例性实施例的可对具有预定大小的编码单元执行的帧内预测模式的示意图；
- [0026] 图18是用于解释根据示例性实施例的具有不同方向性的帧间预测模式的参考图；
- [0027] 图19是用于解释根据示例性实施例的双线性模式的参考图；
- [0028] 图20是用于解释根据示例性实施例的第一预测编码单元的后处理的参考图；
- [0029] 图21是用于解释根据示例性实施例的后处理器的操作的参考图；
- [0030] 图22是用于解释根据示例性实施例的由后处理器使用的邻近像素的参考图；
- [0031] 图23是示出根据示例性实施例的对视频编码的方法的流程图；
- [0032] 图24是用于解释根据示例性实施例的对编码单元进行后处理的索引处理的参考图；
- [0033] 图25是用于解释根据另一示例性实施例的对编码单元进行后处理的索引处理的参考图；
- [0034] 图26是示出根据示例性实施例的对视频解码的方法的流程图；
- [0035] 图27是用于解释当前像素和位于具有(dx,dy)方向的延长线上的邻近像素之间的关系示意图；
- [0036] 图28是用于解释根据示例性实施例的位于根据当前像素的位置的(dx, dy)方向的延长线上的邻近像素的改变示意图；
- [0037] 图29和图30是用于解释根据示例性实施例的确定帧内预测模式方向的方法示意图。
- [0038] 最佳模式
- [0039] 根据本发明的一方面,提供了一种对视频编码的方法,所述方法包括:产生将被编码的当前编码单元的第一预测编码单元;确定当前编码单元是否包括位于当前画面的边界之外的部分;当当前编码单元不包括位于当前画面的边界之外的部分时,通过使用第一预测编码单元的像素和第一预测编码单元的像素的邻近像素改变第一预测编码单元的像素的值,来产生第二预测编码单元;当当前编码单元包括位于当前画面的边界之外的部分时,跳过产生第二预测编码单元的步骤。
- [0040] 根据另一示例性实施例的一方面,提供了一种对视频编码的设备,所述设备包括:

预测器,产生将被编码的当前编码单元的第一预测编码单元;确定器,确定当前编码单元是否包括位于当前画面的边界之外的部分;后处理器,当当前编码单元不包括位于当前画面的边界之外的部分时,通过使用第一预测编码单元的像素和第一预测编码单元的像素的邻近像素改变第一预测编码单元的像素的值,来产生第二预测编码单元;当当前编码单元包括位于当前画面的边界之外的部分时,跳过产生第二预测编码单元。

[0041] 根据另一示例性实施例的一方面,提供了一种对视频解码的方法,所述方法包括:从接收的比特流提取关于将被解码的当前解码单元的预测模式的信息;基于提取的信息产生当前解码单元的第一预测解码单元;确定当前解码单元是否包括位于当前画面的边界之外的部分;当当前解码单元不包括位于当前画面的边界之外的部分时,通过使用第一预测解码单元的像素和第一预测解码单元的像素的邻近像素改变第一预测解码单元的像素的值来产生第二预测解码单元,当当前解码单元包括位于当前画面的边界之外的部分时,跳过产生第二预测解码单元的步骤。

[0042] 根据另一示例性实施例,提供了一种对视频解码的设备,所述设备包括:熵解码器,从接收的比特流提取关于将被解码的当前解码单元的预测模式的信息;预测器,基于提取的信息产生当前解码单元的第一预测解码单元;确定器,确定当前解码单元是否包括位于当前画面的边界之外的部分;后处理器,当当前解码单元不包括位于当前画面的边界之外的部分时,通过使用第一预测解码单元的像素和第一预测解码单元的像素的邻近像素改变第一预测解码单元的像素的值来产生第二预测解码单元,当当前解码单元包括位于当前画面的边界之外的部分时,跳过产生第二预测解码单元的步骤。

具体实施方式

[0043] 以下,将参照附图更完整地描述示例性实施例,在附图中示出了示例性实施例。在示例性实施例中,根据上下文,单元可以表示大小的单位,也可以不表示大小的单位。

[0044] 将参照图1到图13来描述根据示例性实施例的视频编码方法和设备以及视频解码方法和设备。

[0045] 以下,根据示例性实施例,编码单元是在编码器端图像数据被编码的编码数据单元以及在解码器端图像数据被解码的编码数据单元。另外,编码深度表示编码单元被编码的深度。此外,图像可表示视频的静止图像或运动图像(即,视频本身)。

[0046] 图1是根据示例性实施例的视频编码设备100的框图。视频编码设备100 包括最大编码单元划分器110、编码单元确定器120和输出单元130。

[0047] 最大编码单元划分器110可基于针对图像的当前画面的最大编码单元来划分当前画面。如果当前画面大于最大编码单元,则可将当前画面的图像数据划分为至少一个最大编码单元。根据示例性实施例的最大编码单元可以是大小为 32×32 、 64×64 、 128×128 、 256×256 等的单元,其中,数据单元的形状是高度和宽度为2的若干次方的正方形。可根据至少一个最大编码单元将图像数据输出到编码单元确定器120。

[0048] 根据示例性实施例的编码单元可由最大大小和深度表征。深度表示编码单元从最大编码单元被空间划分的次数,随着深度加深或增加,根据深度的较深层编码单元可从最大编码单元被划分到最小编码单元。最大编码单元的深度是最高深度,最小编码单元的深度是最低深度。由于与每个深度相应的编码单元的大小随着最大编码单元的深度加深而减

小,因此与较高深度相应的编码单元可包括多个与较低深度相应的多个编码单元。

[0049] 如上所述,根据编码单元的最大大小将当前画面的图像数据划分为多个最大编码单元,每个最大编码单元可包括根据深度划分的较深层编码单元。由于根据示例性实施例的最大编码单元根据深度被划分,因此包括在最大编码单元的空间域中的图像数据可根据深度被分层地分类。

[0050] 可预先确定编码单元的最大深度和最大大小,最大深度和最大大小限制最大编码单元的高度和宽度被分层划分的总次数。

[0051] 编码单元确定器120对通过根据深度划分最大编码单元的区域而获得的至少一个划分区域进行编码,并根据所述至少一个划分区域确定用于输出最终编码的图像数据的深度。换言之,编码单元确定器120通过根据当前画面的最大编码单元对根据深度的较深层编码单元中的图像数据进行编码,并选择具有最小编码误差的深度来确定编码深度。因此,最终输出与确定的编码深度对应的编码单元的编码图像数据。另外,与编码深度对应的编码单元可被认为是编码的编码单元。

[0052] 确定的编码深度和根据确定的编码深度的编码图像数据被输出到输出单元130。

[0053] 基于与等于或低于最大深度的至少一个深度对应的较深层编码单元对最大编码单元中的图像数据进行编码,并基于每个较深层编码单元来比较对图像数据进行编码的结果。可在比较较深层编码单元的编码误差之后选择具有最小编码误差的深度。可针对每个最大编码单元选择至少一个编码深度。

[0054] 随着编码单元根据深度被分层划分并随着编码单元的数量增加,最大编码单元的大小被划分。另外,即使一个最大编码单元中的多个编码单元对应于相同深度,仍通过单独测量每个编码单元的图像数据的编码误差来确定是否将对应于相同深度的每个编码单元划分为更低深度。因此,即使图像数据被包括在一个最大编码单元中,图像数据被划分为根据深度的多个区域并且在一个最大编码单元中编码误差可根据区域而不同,因此,编码深度可根据图像数据中的区域而不同。因此,可在一个最大编码单元中确定一个或多个编码深度,并可根据至少一个编码深度的编码单元来划分最大编码单元的图像数据。

[0055] 因此,编码单元确定器120可确定包括在最大编码单元中的具有树结构的编码单元。根据示例性实施例的具有树结构的编码单元包括最大编码单元中所包括的所有较深层编码单元中与确定为编码深度的深度对应的编码单元。可在最大编码单元的相同区域中根据深度分层地确定编码深度的编码单元,并可在不同区域中独立地确定编码深度的编码单元。类似地,当前区域中的编码深度与另一区域中的编码深度可被独立地确定。

[0056] 根据示例性实施例的最大深度是与从最大编码单元到最小编码单元的划分次数相关的索引。根据示例性实施例的第一最大深度可表示从最大编码单元到最小编码单元的总划分次数。根据示例性实施例的第二最大深度可表示从最大编码单元划分到最小编码单元的深度等级的总数。例如,当最大编码单元的深度是0时,最大编码单元被划分一次的编码单元的深度可被设置为1,最大编码单元被划分两次的编码单元的深度可被设置为2。这里,如果最小编码单元是最大编码单元被划分4次的编码单元,则存在深度0、深度1、深度2、深度3、深度4这5个深度等级,因此,第一最大深度可被设置为4,第二最大深度可被设置为5。

[0057] 可根据最大编码单元执行预测编码和变换。还根据最大编码单元,基于根据等于

最大深度的深度或小于最大深度的深度的较深层编码单元执行预测编码和变换。可根据正交变换或整数变换的方法执行变换。

[0058] 由于每当最大编码单元根据深度被划分时较深层编码单元的数量就增加,因此对随着深度加深而产生的所有较深层编码单元执行包括预测编码和变换的编码。为了便于描述,现在将基于最大编码单元中的当前深度的编码单元描述预测编码和变换。

[0059] 视频编码设备100可不同地选择用于对图像数据编码的数据单元的大小或形状。为了对图像数据编码,执行诸如预测编码、变换和熵编码的操作,并且同时,可针对所述操作使用相同的数据单元,或者可针对每个操作使用不同的数据单元。

[0060] 例如,视频编码设备100不仅可选择用于对图像数据编码的编码单元,还可选择与该编码单元不同的数据单元以对编码单元中的图像数据执行预测编码。

[0061] 为了在最大编码单元中执行预测编码,可基于与编码深度对应的编码单元(即,基于不再被划分为与更低深度对应的编码单元的编码单元)执行预测编码。以下,不再被划分并且成为用于预测编码的基本单元的编码单元将被称为预测单元。通过划分预测单元获得的分区(partition)可包括预测单元或通过划分预测单元的高度和宽度中的至少一个而获得的数据单元。

[0062] 例如,当 $2N \times 2N$ (其中, N 是正整数)的编码单元不再被划分并且成为 $2N \times 2N$ 的预测单元时,分区的大小可以是 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 或 $N \times N$ 。分区类型的示例包括通过对称地划分预测单元的高度或宽度而获得的对称分区、通过不对称地划分预测单元的高度或宽度而获得的分区(诸如, $1:n$ 或 $n:1$)、通过几何地划分预测单元而获得的分区以及具有任意形状的分区。

[0063] 预测单元的预测模式可以是帧内模式、帧间模式和跳过模式中的至少一个。例如,可对 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 或 $N \times N$ 的分区执行帧内模式或帧间模式。另外,可仅对 $2N \times 2N$ 的分区执行跳过模式。可独立地对编码单元中的一个预测单元执行编码,从而选择具有最小编码误差的预测模式。

[0064] 视频编码设备100可不仅基于用于对图像数据编码的编码单元对编码单元中的图像数据执行变换,还可基于与编码单元不同的数据单元对编码单元中的图像数据执行变换。

[0065] 为了在编码单元中执行变换,可基于具有小于或等于编码单元的大小的数据单元执行变换。例如,用于变换的数据单元可包括用于帧内模式的数据单元和用于帧间模式的数据单元。

[0066] 用作变换的基础的数据单元现在将被称为变换单元。还可在变换单元中设置指示通过划分编码单元的宽度和高度而达到变换单元的划分次数的变换深度。例如,在 $2N \times 2N$ 的当前编码单元中,当变换单元的大小也是 $2N \times 2N$ 时,变换深度可以是0,当当前编码单元的高度和宽度均被划分为两等分,总共被划分为4个变换单元并且因此变换单元的大小是 $N \times N$ 时,变换深度可以是1,当当前编码单元的高度和宽度均被划分为四等分,总共被划分为 4^2 个变换单元并且因此变换单元的大小是 $N/2 \times N/2$ 时,变换深度可以是2。例如,可根据分层树结构设置变换单元,在该结构中,根据变换深度的分层特性将上层变换深度的变换单元划分为下层变换深度的四个变换单元。

[0067] 与编码单元类似,编码单元中的变换单元可被递归地划分为大小更小的区域,从

而可以以区域为单位独立地确定变换单元。因此,可根据变换深度按照具有树结构的变换来划分编码单元中的残差数据。

[0068] 根据与编码深度对应的编码单元的信息编码不仅需要关于编码深度的信息,还需要与预测编码和变换相关的信息。因此,编码单元确定器120不仅确定具有最小编码误差的编码深度,还确定预测单元中的分区类型、根据预测单元的预测模式以及用于变换的变换单元的大小。

[0069] 稍后将参照图3到图12详细描述根据示例性实施例的根据最大编码单元的树结构的编码单元和确定分区的方法。

[0070] 编码单元确定器120可通过使用基于拉格朗日乘子(Lagrangian multiplier) 的率失真最优化,测量根据深度的较深层编码单元的编码误差。

[0071] 输出单元130在比特流中输出最大编码单元的图像数据和关于根据编码深度的编码模式的信息,其中,基于由编码单元确定器120确定的至少一个编码深度对最大编码单元的图像数据编码。

[0072] 可通过对图像的残差数据编码来获得编码的图像数据。

[0073] 关于根据编码深度的编码模式的信息可包括关于编码深度的信息、关于预测单元中的分区类型、预测模式和变换单元的大小的信息。

[0074] 可通过使用根据深度的划分信息定义关于编码深度的信息,所述根据深度的划分信息指示是否对较低深度而非当前深度的编码单元执行编码。如果当前编码单元的当前深度是编码深度,则当前编码单元中的图像数据被编码和输出,并且因此划分信息可被定义为不将当前编码单元划分至较低深度。可选择地,如果当前编码单元的当前深度不是编码深度,则对较低深度的编码单元执行编码,并且因此划分信息可被定义为对当前编码单元进行划分以获得较低深度的编码单元。

[0075] 如果当前深度不是编码深度,则对被划分为较低深度的编码单元的编码单元执行编码。因为在当前深度的一个编码单元中存在较低深度的至少一个编码单元,所以对较低深度的每个编码单元重复执行编码,从而可针对具有相同深度的编码单元递归地执行编码。

[0076] 因为针对一个最大编码单元确定具有树结构的编码单元,并且针对编码深度的编码单元确定关于至少一个编码模式的信息,所以可针对一个最大编码单元确定关于至少一个编码模式的信息。另外,因为根据深度分层地划分图像数据,所以最大编码单元的图像数据的编码深度可根据位置而不同,从而可针对图像数据设置关于编码深度和编码模式的信息。

[0077] 因此,输出单元130可将关于相应的编码深度和编码模式的编码信息分配给包括在最大编码单元中的编码单元、预测单元和最小单元中的至少一个。

[0078] 根据示例性实施例的最小单元是通过将组成最低深度的最小编码单元划分为4份而获得的矩形数据单元。可选择地,最小单元可以是最大矩形数据单元,所述最大矩形数据单元可被包括在最大编码单元中包括的所有编码单元、预测单元、分区单元和变换单元中。

[0079] 例如,通过输出单元130输出的编码信息可被分类为根据编码单元的编码信息和根据预测单元的编码信息。根据编码单元的编码信息可包括关于预测模式的信息以及关于分区的大小的信息。根据预测单元的编码信息可包括关于帧间模式的估计方向的信息、关

于帧间模式的参考图像索引的信息、关于运动矢量的信息、关于帧内模式的色度分量的信息以及关于帧内模式的插值方法的信息。另外,关于根据画面、像条或GOP定义的编码单元的最大大小的信息以及关于最大深度的信息可被插入到比特流的SPS(序列参数集)或头中。

[0080] 在视频编码设备100中,较深层编码单元可以通过将作为上一层的较高深度的编码单元的高度或宽度除以2而获得的编码单元。换言之,在当前深度的编码单元的大小是 $2N \times 2N$ 时,较低深度的编码单元的大小是 $N \times N$ 。另外,大小为 $2N \times 2N$ 的当前深度的编码单元可包括最多4个较低深度的编码单元。

[0081] 因此,视频编码设备100可基于最大编码单元的大小和考虑当前画面的特性而确定的最大深度,通过针对每个最大编码单元确定具有最佳形状和最佳大小的编码单元,来形成具有树结构的编码单元。另外,因为可通过使用各种预测模式和变换中的任意一种来对每个最大编码单元执行编码,所以可考虑各种图像大小的编码单元的特性来确定最佳编码模式。

[0082] 因此,如果以传统的宏块对具有高分辨率或大数据量的图像编码,则每个画面的宏块的数量会急剧增加,针对每个宏块产生的压缩信息的条数增加,从而难以发送压缩的信息并且数据压缩效率下降。然而,通过使用视频编码设备100,因为在考虑图像的大小而增大编码单元的最大大小的同时,考虑图像的特性调整了编码单元,所以图像压缩效率可提高。

[0083] 图2是根据示例性实施例的视频解码设备200的框图。视频解码设备200 包括接收器210、图像数据和编码信息提取器220以及图像数据解码器230。用于视频解码设备200的各种操作的各种术语(诸如编码单元、深度、预测单元、变换单元以及关于各种编码模式的信息)的定义与参照图1和视频编码设备100所描述的术语的定义相同。

[0084] 接收器210接收并解析编码的视频的比特流。图像数据和编码信息提取器220从解析的比特流中提取每个编码单元的编码的图像数据,并将提取的图像数据输出到图像数据解码器230,其中,编码单元具有根据每个最大编码单元的树结构。图像数据和编码信息提取器220可从关于当前画面的头或 SPS提取关于当前画面的编码单元的最大大小的信息。

[0085] 此外,图像数据和编码信息提取器220从解析的比特流提取关于编码单元的编码深度和编码模式的信息,其中,编码单元具有根据每个最大编码单元的树结构。提取的关于编码深度和编码模式的信息被输出到图像数据解码器230。换言之,比特流中的图像数据被划分为最大编码单元,从而图像数据解码器230对每个最大编码单元的图像数据解码。

[0086] 可针对关于与编码深度相应的至少一个编码单元的信息设置关于根据最大编码单元的编码深度和编码模式的信息,关于编码模式的信息可包括关于与编码深度相应的相应编码单元的分区类型的信息、关于预测模式的信息以及变换单元的大小的信息。另外,根据深度的划分信息可被提取作为关于编码深度的信息。

[0087] 由图像数据和编码信息提取器220提取的关于根据每个最大编码单元的编码模式和编码深度的信息是关于这样的编码深度和编码模式的信息,即:所述编码深度和编码模式被确定为当编码器(诸如视频编码设备100)根据每个最大编码单元针对根据深度的每个较深层编码单元重复执行编码时产生最小编码误差。因此,视频解码设备200可通过根据产生最小编码误差的编码深度和编码模式对图像数据解码来恢复图像。

[0088] 由于关于编码深度和编码模式的编码信息可被分配给相应的编码单元、预测单元和最小单元中的预定数据单元,因此图像数据和编码信息提取器220 可提取关于根据预定数据单元的编码深度和编码模式的信息。被分配有相同的关于编码深度和编码模式的信息的预定数据单元可被推断为是包括在同一最大编码单元中的数据单元。

[0089] 图像数据解码器230通过基于关于根据最大编码单元的编码深度和编码模式的信息,对每个最大编码单元中的图像数据解码来恢复当前画面。换言之,图像数据解码器230可基于提取的与包括在每个最大编码单元中的具有树结构的编码单元中的每个编码单元的分区类型、预测模式和变换单元有关的信息,来对编码的图像数据解码。解码处理可包括预测和反变换,所述预测包括帧内预测和运动补偿。可根据反正交变换或反整数变换的方法来执行反变换。

[0090] 图像数据解码器230可基于关于根据编码深度的编码单元的预测单元的分区类型和预测模式的信息,根据每个编码单元的分区和预测模式执行帧内预测或运动补偿。

[0091] 此外,图像数据解码器230可基于关于根据编码深度的编码单元的变换单元的大小的信息,根据编码单元中的每个变换单元来执行反变换,以根据最大编码单元执行反变换。

[0092] 图像数据解码器230可通过使用根据深度的划分信息来确定当前最大编码单元的至少一个编码深度。如果所述划分信息指示在当前深度不再划分图像数据,则当前深度是编码深度。因此,图像数据解码器230可通过使用关于针对与编码深度相应的每个编码单元的预测单元的分区类型、预测模式、变换单元的大小的信息,对当前最大编码单元中的与每个编码深度相应的至少一个编码单元的编码数据进行解码,并输出当前最大编码单元的图像数据。

[0093] 换言之,可通过观察为编码单元中的预定数据单元分配的编码信息集,收集包含编码信息(所述编码信息包括相同的划分信息)的数据单元,并且收集的数据单元可被认为是将由图像数据解码器230以相同编码模式解码的一个数据单元。

[0094] 视频解码设备200可获得与在对每个最大编码单元递归地执行编码时产生最小编码误差的至少一个编码单元有关的信息,并且视频解码设备200可使用所述信息来对当前画面解码。换言之,可以对在每个最大编码单元中被确定为最佳编码单元的具有树结构的编码单元进行解码。此外,考虑图像数据的分辨率和数据量来确定编码单元的最大大小。

[0095] 因此,即使图像数据具有高分辨率和大数据量,也可通过使用编码单元的大小和编码模式对所述图像数据进行有效地解码和恢复,其中,通过使用从编码器接收的关于最佳编码模式的信息,根据图像数据的特性来自适应地确定编码单元的大小和编码模式。

[0096] 现在将参照图3至图13描述根据示例性实施例的确定具有树结构的编码单元、预测单元和变换单元的方法。

[0097] 图3是用于描述根据示例性实施例的编码单元的构思的示图。可以以宽度×高度来表示编码单元的大小,并且编码单元的大小可以是 64×64 、 32×32 、 16×16 和 8×8 。 64×64 的编码单元可被划分为 64×64 、 64×32 、 32×64 或 32×32 的分区, 32×32 的编码单元可被划分为 32×32 、 32×16 、 16×32 或 16×16 的分区, 16×16 的编码单元可被划分为 16×16 、 16×8 、 8×16 或 8×8 的分区, 8×8 的编码单元可被划分为 8×8 、 8×4 、 4×8 或 4×4 的分区。

[0098] 在视频数据310中,分辨率是 1920×1080 ,编码单元的最大大小是64,最大深度是

2。在视频数据320中,分辨率是 1920×1080 ,编码单元的最大大小是64,最大深度是3。在视频数据330中,分辨率是 352×288 ,编码单元的最大大小是16,最大深度是1。图3中示出的最大深度表示从最大编码单元到最小编码单元的划分的总数。

[0099] 如果分辨率高或者数据量大,则编码单元的最大大小可以很大,从而不仅提高编码效率,还精确地反映图像的特性。因此,具有比视频数据330更高的分辨率的视频数据310和视频数据320的编码单元的最大大小可以为64。

[0100] 由于视频数据310的最大深度是2,所以视频数据310的编码单元315 可包括具有64的长轴(long axis)大小的最大编码单元,还包括由于通过将最大编码单元划分两次使深度加深了两层而具有32和16的长轴大小的编码单元。同时,因为视频数据330的最大深度是1,所以视频数据330的编码单元335可包括具有16的长轴大小的最大编码单元,还包括由于通过将最大编码单元划分一次使深度加深了一层而具有8的长轴大小的编码单元。

[0101] 因为视频数据320的最大深度是3,所以所述视频数据320的编码单元 325可包括具有64的长轴大小的最大编码单元,并且由于通过划分最大编码单元三次将深度加深了3层而具有32、16和8的长轴大小的编码单元。随着深度加深,详细信息可被精确地表示。

[0102] 图4是根据示例性实施例的基于编码单元的图像编码器400的框图。图像编码器400执行视频编码设备100的编码单元确定器120的操作,以对图像数据编码。换言之,帧内预测器410对当前帧405中的帧内模式的编码单元执行帧内预测,运动估计器420和运动补偿器425通过使用当前帧405 和参考帧495,对当前帧405中的帧间模式的编码单元执行帧间估计和运动补偿。

[0103] 从帧内预测器410、运动估计器420和运动补偿器425输出的数据通过变换器430和量化器440被输出为量化的变换系数。量化的变换系数通过反量化器460和反变换器470被恢复为空间域中的数据,并且恢复的空间域中的数据在通过去块单元480和环路滤波单元490进行后处理之后,被输出为参考帧495。量化的变换系数可通过熵编码器450被输出为比特流455。

[0104] 为了在视频编码设备100中应用图像编码器400,图像编码器400的所有部件(即,帧内预测器410、运动估计器420、运动补偿器425、变换器430、量化器440、熵编码器450、反量化器460、反变换器470、去块单元480、环路滤波器490)在考虑每个最大编码单元的最大深度的同时,基于具有树结构的编码单元中的每个编码单元执行操作。

[0105] 具体地说,帧内预测器410、运动估计器420以及运动补偿器425在考虑当前最大编码单元的最大大小和最大深度的同时,确定具有树结构的编码单元中的每个编码单元的分区和预测模式,变换器430确定在具有树结构的编码单元中的每个编码单元中的变换单元的大小。

[0106] 图5是根据示例性实施例的基于编码单元的图像解码器500的框图。解析器510从比特流505解析将被解码的编码图像数据和解码所需的关于编码的信息。编码图像数据通过熵解码器520和反量化器530被输出为反量化的数据,并且反量化的数据通过反变换器540被恢复为空间域中的图像数据。

[0107] 帧内预测器550针对空间域中的图像数据,对帧内模式的编码单元执行帧内预测,运动补偿器560通过使用参考帧585,对帧间模式的编码单元执行运动补偿。

[0108] 经过帧内预测器550和运动补偿器560的空间域中的图像数据可在通过去块单元

570和环路滤波单元580进行后处理之后,被输出为恢复的帧595。另外,通过去块单元570和环路滤波单元580进行后处理的图像数据可被输出为参考帧585。

[0109] 为了在视频解码设备200的图像数据解码器230中对图像数据解码,图像解码器500可执行在解析器510之后执行的操作。

[0110] 为了在视频解码设备200中应用图像解码器500,图像解码器500的所有部件(即,解析器510、熵解码器520、反量化器530、反变换器540、帧内预测器550、运动补偿器560、去块单元570以及循环滤波单元580)针对每个最大编码单元基于具有树结构的编码单元来执行操作。

[0111] 具体地说,帧内预测器550和运动补偿器560基于每个具有树结构的编码单元的分区和预测模式来执行操作,反变换器540基于每个编码单元的变换单元的大小来执行操作。

[0112] 图6是示出根据示例性实施例的根据深度的较深层编码单元以及分区的示图。视频编码设备100和视频解码设备200使用分层的编码单元以考虑图像的特性。根据图像的特性可自适应地确定编码单元的最大高度、最大宽度和最大深度,或可由用户不同地设置编码单元的最大高度、最大宽度和最大深度。可根据编码单元的预定最大大小来确定根据深度的较深层编码单元的大小。

[0113] 根据示例性实施例,在编码单元的分层结构600中,编码单元的最大高度和最大宽度均为64,最大深度是4。随着深度随分层结构600的垂直轴加深,较深层编码单元的高度和宽度均被划分。此外,沿分层结构600的水平轴示出了作为用于每个较深层编码单元的预测编码的基础的预测单元和分区。

[0114] 换言之,编码单元610是分层结构600中的最大编码单元,其中,深度是0,大小(即高度 $\times$ 宽度)是 64×64 。深度沿着垂直轴加深,并存在大小为 32×32 和深度为1的编码单元620、大小为 16×16 和深度为2的编码单元630、大小为 8×8 和深度为3的编码单元640和大小为 4×4 和深度为4的编码单元650。大小为 4×4 和深度为4的编码单元650是最小编码单元。

[0115] 编码单元的预测单元和分区根据每个深度沿着水平轴布置。换言之,如果大小为 64×64 和深度为0的编码单元610是预测单元,则所述预测单元可被划分为包括在编码单元610中的多个分区,即:大小为 64×64 的分区610、大小为 64×32 的分区612、大小为 32×64 的分区614和大小为 32×32 的分区616。

[0116] 类似地,大小为 32×32 和深度为1的编码单元620的预测单元可以被划分为包括在编码单元620中的多个分区,即:大小为 32×32 的分区620、大小为 32×16 的分区622、大小为 16×32 的分区624和大小为 16×16 的分区626。

[0117] 类似地,大小为 16×16 和深度为2的编码单元630的预测单元可以被划分为包括在编码单元630中的多个分区,即:包括在编码单元630中的大小为 16×16 的分区、大小为 16×8 的分区632、大小为 8×16 的分区634和大小为 8×8 的分区636。

[0118] 类似地,大小为 8×8 和深度为3的编码单元640的预测单元可以被划分为包括在编码单元640中的多个分区,即:包括在编码单元640中的大小为 8×8 的分区、大小为 8×4 的分区642、大小为 4×8 的分区644和大小为 4×4 的分区646。

[0119] 大小为 4×4 和深度为4的编码单元650是最小编码单元和最低深度的编码单元。编码单元650的预测单元只被分配给大小为 4×4 的分区。

[0120] 为了确定组成最大编码单元610的编码单元的至少一个编码深度,视频编码设备100的编码单元确定器120对包括在最大编码单元610中的与各个深度相应的编码单元执行编码。

[0121] 随着深度加深,以相同范围和相同大小包括数据的根据深度的较深层编码单元的数量增加。例如,需要4个与深度2相应的编码单元以覆盖与深度1相应的一个编码单元中包括的数据。因此,为了比较根据深度的相同数据的编码结果,与深度1相应的编码单元和四个与深度2相应的编码单元均被编码。

[0122] 为了对深度中的当前深度执行编码,可通过沿着分层结构600的水平轴对与当前深度相应的编码单元中的每个预测单元执行编码来针对当前深度选择最小编码误差。作为选择,可通过随着深度沿分层结构600的垂直轴加深对每个深度执行编码来比较根据深度的最小编码误差,从而搜索最小编码误差。编码单元610中具有最小编码误差的深度和分区可被选择为编码单元610 的编码深度和分区类型。

[0123] 图7是用于描述根据示例性实施例的在编码单元710和变换单元720之间的关系的示意图。

[0124] 视频编码设备100或200针对每个最大编码单元根据具有小于或等于最大编码单元的大小的编码单元对图像进行编码或解码。可基于不大于相应编码单元的数据单元来选择编码期间用于变换的变换单元的大小。

[0125] 例如,在视频编码设备100或200中,如果编码单元170的大小是 64×64 ,则可通过使用大小为 32×32 的变换单元720来执行变换。

[0126] 此外,可通过对大小为小于 64×64 的 32×32 、 16×16 、 8×8 、 4×4 的变换单元中的每一个执行变换来对大小为 64×64 的编码单元710的数据进行编码,然后可选择具有最小编码误差的变换单元。

[0127] 图8是用于描述根据示例性实施例的与编码深度相应的编码单元的编码信息的示意图。视频编码设备100的输出单元130可将与编码深度相应的每个编码单元的关于分区类型的信息800、关于预测模式的信息810和关于变换单元的大小的信息820编码为关于编码模式的信息,并将其发送。

[0128] 信息800指示关于通过划分当前编码单元的预测单元而获得的分区的形状的信息,其中,所述分区是用于对当前编码单元进行预测编码的数据单元。例如,大小为 $2N \times 2N$ 的当前编码单元CU_0可被划分为大小为 $2N \times 2N$ 的分区802、大小为 $2N \times N$ 的分区804、大小为 $N \times 2N$ 的分区806和大小为 $N \times N$ 的分区808中的任意一个。这里,关于分区类型的信息800被设置为指示大小为 $2N \times N$ 的分区804、大小为 $N \times 2N$ 的分区806和大小为 $N \times N$ 的分区808 中的一个。

[0129] 信息810指示每个分区的预测模式。例如,信息810可指示对由信息800 指示的分区执行预测编码的模式,即:帧内模式812、帧间模式814或跳过模式816。

[0130] 信息820指示当对当前编码单元执行变换时作为基础的变换单元。例如,变换单元可以是第一帧内变换单元822、第二帧内变换单元824、第一帧间变换单元826或第二帧间变换单元828。

[0131] 视频解码设备200的图像数据和编码信息提取器220可根据每个较深层编码单元提取和使用用于解码的信息800、810和820。

[0132] 图9是根据示例性实施例的根据深度的较深层编码单元的示图。划分信息可被用于指示深度的改变。划分信息指示当前深度的编码单元是否被划分为较低深度的编码单元。

[0133] 用于对深度为0大小为 $2N_0 \times 2N_0$ 的编码单元900进行预测编码的预测单元910可包括以下分区类型的分区:大小为 $2N_0 \times 2N_0$ 的分区类型912、大小为 $2N_0 \times N_0$ 的分区类型914、大小为 $N_0 \times 2N_0$ 的分区类型916和大小为 $N_0 \times N_0$ 的分区类型918。图9只示出了通过对称地划分预测单元910 获得的分区类型912至分区类型918,但是分区类型不限于此,预测单元910 的分区可包括非对称分区、具有预定形状的分区和具有几何形状的分区。

[0134] 根据每个分区类型,可对大小为 $2N_0 \times 2N_0$ 的一个分区、大小为 $N_0 \times 2N_0$ 的两个分区、大小为 $2N_0 \times N_0$ 的两个分区、大小为 $N_0 \times N_0$ 的四个分区重复地执行预测编码。可对大小为 $2N_0 \times 2N_0$ 、 $N_0 \times 2N_0$ 、 $2N_0 \times N_0$ 和 $N_0 \times N_0$ 的分区执行帧内模式下和帧间模式下的预测编码。仅对大小为 $N_0 \times N_0$ 的分区执行跳过模式下的预测编码。

[0135] 对包括分区类型912至分区类型918中的预测编码的编码误差进行比较,在分区类型中确定最小编码误差。如果在分区类型912至分区类型916之一中,编码误差最小,则预测单元910可不被划分为较低深度。

[0136] 如果在分区类型918中编码误差最小,则在操作920,深度从0改变为1 以划分分区类型918,并对深度为2和大小为 $N_0 \times N_0$ 的编码单元930重复执行编码以搜索最小编码误差。

[0137] 用于对深度为1和大小为 $2N_1 \times 2N_1 (=N_0 \times N_0)$ 的编码单元930进行预测编码的预测单元940可包括以下分区类型的分区:大小为 $2N_1 \times 2N_1$ 的分区类型942、大小为 $2N_1 \times N_1$ 的分区类型944、大小为 $N_1 \times 2N_1$ 的分区类型946和大小为 $N_1 \times N_1$ 的分区类型948。

[0138] 如果在分区类型948中编码误差最小,则在操作950,深度从1改变为2 以划分分区类型948,并对深度为2和大小为 $N_2 \times N_2$ 的编码单元960重复执行编码以搜索最小编码误差。

[0139] 当最大深度为d时,根据每个深度的划分操作可被执行直到深度变为 $d-1$,并且划分信息可被编码直到深度为0至 $d-2$ 中的一个。换言之,当在操作970将与深度 $d-2$ 相应的编码单元划分之后,执行编码直到深度为 $d-1$ 时,用于对深度为 $d-1$ 和大小为 $2N_{(d-1)} \times 2N_{(d-1)}$ 的编码单元980进行预测编码的预测单元990可包括以下分区类型的分区:大小为 $2N_{(d-1)} \times 2N_{(d-1)}$ 的分区类型992、大小为 $2N_{(d-1)} \times N_{(d-1)}$ 的分区类型994、大小为 $N_{(d-1)} \times 2N_{(d-1)}$ 的分区类型996以及大小为 $N_{(d-1)} \times N_{(d-1)}$ 的分区类型 998。

[0140] 可对分区类型992至分区类型998中的大小为 $2N_{(d-1)} \times 2N_{(d-1)}$ 的一个分区、大小为 $2N_{(d-1)} \times N_{(d-1)}$ 的两个分区、大小为 $N_{(d-1)} \times 2N_{(d-1)}$ 的两个分区、大小为 $N_{(d-1)} \times N_{(d-1)}$ 的四个分区重复执行预测编码,以搜索具有最小编码误差的分区类型。

[0141] 即使当分区类型998具有最小编码误差时,由于最大深度为d,所以深度为 $d-1$ 的编码单元 $CU_{(d-1)}$ 不再被划分为较低深度,构成当前最大编码单元900的编码单元的编码深度被确定为 $d-1$,并且当前最大编码单元900的分区类型可被确定为 $N_{(d-1)} \times N_{(d-1)}$ 。此外,由于最大深度为d,并且最低深度为 $d-1$ 的最小编码单元980不再被划分为较低深度,所以不设置最小编码单元980的划分信息。

[0142] 数据单元999可以是用于当前最大编码单元的“最小单元”。根据示例性实施例的最小单元可以通过将最小编码单元980划分为4部分而获得的矩形数据单元。通过重复地执行编码,视频编码设备100可通过比较根据编码单元900的深度的编码误差来选择具有最小编码误差的深度以确定编码深度,并且可将相应的分区类型和预测模式设置为编码深度的编码模式。

[0143] 这样,在深度1至深度d的所有深度中比较根据深度的最小编码误差,并且具有最小编码误差的深度可被确定为编码深度。预测单元的分区类型、编码深度和预测模式可作为关于编码模式的信息被编码和发送。此外,由于从深度0至编码深度来划分编码单元,所以只有编码深度的划分信息被设置为0,并且除了编码深度之外的深度的划分信息被设置为1。

[0144] 视频解码设备200的图像数据和编码信息提取器220可提取并使用关于编码单元900的编码深度以及预测单元的信息,以对分区912解码。视频解码设备200可通过使用根据深度的划分信息将划分信息为0的深度确定为编码深度,并且可使用关于相应深度的编码模式的信息进行解码。

[0145] 图10至图12是描述根据示例性实施例的编码单元1010、预测单元1060 和变换单元1070之间的关系的示图。编码单元1010是最大编码单元中的与由视频编码设备100确定的编码深度相应的具有树结构的编码单元。预测单元1060是每个编码单元1010的预测单元的分区,变换单元1070是每个编码单元1010的变换单元。

[0146] 当编码单元1010中的最大编码单元的深度为0时,编码单元1012和1054 的深度为1,编码单元1014、1016、1018、1028、1050和1052的深度为2,编码单元1020、1022、1024、1026、1030、1032和1048的深度为3,编码单元1040、1042、1044和1046的深度为4。

[0147] 在预测单元1060中,通过在编码单元1010中划分编码单元来获得一些编码单元1014、1016、1022、1032、1048、1050、1052和1054。换言之,编码单元1014、1022、1050和1054中的分区类型的大小为 $2N \times N$,编码单元1016、1048和1052的分区类型的大小为 $N \times 2N$,编码单元1032的分区类型的大小为 $N \times N$ 。编码单元1010的预测单元和分区小于或等于每个编码单元。

[0148] 以小于编码单元1052的数据单元对变换单元1070中的编码单元1052 的图像数据执行变换或反变换。另外,变换单元1070中的编码单元1014、1016、1022、1032、1048、1050和1052与预测单元1060中的那些编码单元在大小和形状上不同。换言之,视频编码设备100和视频解码设备200可单独地对相同的编码单元中的数据单元执行帧内预测、运动估计、运动补偿、变换和反变换。

[0149] 因此,对最大编码单元的每个区域中具有分层结构的每个编码单元递归地执行编码以确定最佳编码单元,因此可获得具有递归树结构的编码单元。编码信息可包括关于编码单元的划分信息、关于分区类型的信息、关于预测模式的信息以及关于变换单元的大小的信息。表1示出了可由视频编码和解码设备100和200设置的编码信息。

[0150] [表1]

[0151]	划分信息0 (对大小为 $2N \times 2N$ 且当前深度为 d 的编码单元编码)				划分信息1
	预测模式	分区类型		变换单元的大小	
	帧内 帧间 跳过(仅 $2N \times 2N$)	对称分区 类型	非对称分 区类型	变换单元 的划分信 息0	变换单元 的划分信 息1
		$2N \times 2N$ $2N \times N$ $N \times 2N$ $N \times N$	$2N \times nU$ $2N \times nD$ $nL \times 2N$ $nR \times 2N$	$2N \times 2N$	$N \times N$ (对称 类型) $N/2 \times N/2$ (非对称类 型)

[0152] 视频编码设备100的输出单元130可输出关于具有树结构的编码单元的编码信息，视频解码设备200的图像数据和编码信息提取器220可从接收的比特流提取关于具有树结构的编码单元的编码信息。

[0153] 划分信息指示当前编码单元是否被划分为较低深度的编码单元。如果当前深度 d 的划分信息为0，则当前编码单元不再被划分为较低深度的深度是编码深度，因此可针对编码深度定义关于分区类型、预测模式、变换单元的大小的信息。如果根据划分信息将当前编码单元进一步划分，则对较低深度的四个划分的编码单元独立执行编码。

[0154] 预测模式可以是帧内模式、帧间模式和跳过模式中的一种。可在所有分区类型中定义帧内模式和帧间模式，仅在大小为 $2N \times 2N$ 的分区类型中定义跳过模式。

[0155] 关于分区类型的信息可指示通过对称划分预测单元的高度或宽度获得的大小为 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 和 $N \times N$ 的对称分区类型以及通过非对称划分预测单元的高度或宽度获得的大小为 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 和 $nR \times 2N$ 的非对称分区类型。可通过以1:3和3:1划分预测单元的高度来分别获得大小为 $2N \times nU$ 和 $2N \times nD$ 的非对称分区类型，可通过以1:3和3:1划分预测单元的宽度来分别获得大小为 $nL \times 2N$ 和 $nR \times 2N$ 的非对称分区类型。

[0156] 变换单元的大小可被设置为帧内模式中的两种类型以及帧间模式中的两种类型。换言之，如果变换单元的划分信息为0，则变换单元的大小可以是 $2N \times 2N$ ($2N \times 2N$ 是当前编码单元的大小)。如果变换单元的划分信息为1，则可通过划分当前编码单元来获得变换单元。另外，如果大小为 $2N \times 2N$ 的当前编码单元的分区类型是对称分区类型，则变换单元的大小可以是 $N \times N$ ，如果当前编码单元的分区类型是非对称分区类型，则变换单元的大小可以是 $N/2 \times N/2$ 。

[0157] 关于具有树结构的编码单元的编码信息可包括与编码深度相应的编码单元、预测单元以及最小单元中的至少一个。与编码深度相应的编码单元可包括包含相同编码信息的最小单元和预测单元中的至少一个。

[0158] 因此，通过比较邻近数据单元的编码信息来确定邻近数据单元是否包括在与编码深度相应的相同编码单元中。另外，通过使用数据单元的编码信息来确定与编码深度相应的相应编码单元，因此可确定最大编码单元中的编码深度的分布。

[0159] 因此，如果基于邻近数据单元的编码信息来预测当前编码单元，则可直接参照和使用与当前编码单元邻近的较深层编码单元中的数据单元的编码信息。

[0160] 或者，如果基于邻近数据单元的编码信息来预测当前编码单元，则使用所述数据

单元的编码信息来搜索与当前编码单元邻近的数据单元,并且可参照搜索到的邻近的编码单元来预测当前编码单元。

[0161] 图13是描述根据表1的编码模式信息的编码单元、预测单元或分区、变换单元之间的关系示意图。最大编码单元1300包括多个编码深度的编码单元 1302、1304、1306、1312、1314、1316以及1318。这里,由于编码单元1318 是编码深度的编码单元,所以划分信息可被设置为0。关于大小为 $2N \times 2N$ 的编码单元1318的分区类型的信息可被设置为大小为 $2N \times 2N$ 的分区类型1322、大小为 $2N \times N$ 的分区类型1324、大小为 $N \times 2N$ 的分区类型1326、大小为 $N \times N$ 的分区类型1328、大小为 $2N \times nU$ 的分区类型1332、大小为 $2N \times nD$ 的分区类型1334、大小为 $nL \times 2N$ 的分区类型1336以及大小为 $nR \times 2N$ 的分区类型1338 中的一个。

[0162] 当分区类型被设置为对称(即,分区类型1322、1324、1326或1328)时,如果变换单元的划分信息(TU大小标志)为0,则设置大小为 $2N \times 2N$ 的变换单元1342;如果TU大小标志为1,则设置大小为 $N \times N$ 的变换单元1344。

[0163] 当分区类型被设置为非对称(即,分区类型1332、1334、1336或1338)时,如果TU大小标志为0,则设置大小为 $2N \times 2N$ 的变换单元1352;如果TU大小标志为1,则设置大小为 $N/2 \times N/2$ 的变换单元1354。

[0164] 参照图13,TU大小标志是具有值0或1的标志,但是TU大小标志并不限于1比特,在TU大小标志从0增加时,变换单元可被分层地划分为具有树结构。

[0165] 在这种情况下,根据示例性实施例,可通过使用变换单元的TU大小标志以及变换单元的最大大小和最小大小一起来表示已经实际使用的变换单元的大小。根据示例性实施例,视频编码设备100能够对最大变换单元大小信息、最小变换单元大小信息和最大TU大小标志编码。对最大变换单元大小信息、最小变换单元大小信息和最大TU大小标志编码的结果可被插入SPS。根据示例性实施例,视频解码设备200可通过使用最大变换单元大小信息、最小变换单元大小信息和最大TU大小标志信息来对视频解码。

[0166] 例如,如果当前编码单元的大小是 64×64 并且最大变换单元大小是 32×32 ,则当TU大小标志是0时,变换单元的大小可以是 32×32 ;当TU大小标志是1时,变换单元的大小可以是 16×16 ;并且当TU大小标志是2时,变换单元的大小可以是 8×8 。

[0167] 作为另一示例,如果当前编码单元的大小是 32×32 并且最小变换单元是 32×32 ,则当TU大小标志是0时,变换单元的大小可以是 32×32 。这里,TU大小标志不能被设置为除0之外的值,这是因为变换单元的大小不能小于 32×32 。

[0168] 作为另一示例,如果当前编码单元的大小是 64×64 并且最大TU大小标志是1,则TU大小标志可被设置为0或1。这里,TU大小标志不能被设置为除了0或1之外的值。

[0169] 因此,当TU大小标志是0时,如果定义最大TU大小标志为“MaxTransformSizeIndex”,定义最小变换单元大小为“MinTransformSize”,定义变换单元大小为“RootTuSize”,则可通过等式(1)来定义可在当前编码单元中确定的当前最小编码单元大小“CurrMinTuSize”:

[0170]
$$\text{CurrMinTuSize} = \max(\text{MinTransformSize}, \text{RootTuSize} / (2^{\text{MaxTransformSizeIndex}})) \cdots \cdots (1)$$

[0171] 与可在当前编码单元中确定的当前最小变换单元大小“CurrMinTuSize”相比,当TU大小标志是0时的变换单元大小“RootTuSize”可表示可在系统中选择的最大变换单元。

在等式(1)中,“ $\text{RootTuSize}/(2^{\text{MaxTransformSizeIndex}})$ ”表示当在TU大小标志是0时的变换单元大小“RootTuSize”被划分与最大TU 大小标志相应的次数时的变换单元大小,“MinTransformSize”表示最小变换大小。因此,“ $\text{RootTuSize}/(2^{\text{MaxTransformSizeIndex}})$ ”和“MinTransformSize”中的较小值可以是可在当前编码单元中确定的当前最小变换单元大小“CurrMinTuSize”。

[0172] 根据示例性实施例,最大变换单元大小RootTuSize可根据预测模式的类型而改变。

[0173] 例如,如果当前预测模式是帧间模式,则可通过使用下面的等式(2)来确定RootTuSize。在等式(2)中,“MaxTransformSize”表示最大变换单元大小,“PUSize”表示当前预测单元大小。

[0174] $\text{RootTuSize} = \min(\text{MaxTransformSize}, \text{PUSize}) \cdots \cdots (2)$

[0175] 也就是说,如果当前预测模式是帧间模式,则当TU大小标志是0时的变换单元大小“RootTuSize”可以是最大变换单元大小和当前预测单元大小中的较小值。

[0176] 如果当前分区单元的预测模式是帧内模式,则可通过使用下面的等式(3) 来确定“RootTuSize”。在等式(3)中,“PartitionSize”表示当前分区单元的大小。

[0177] $\text{RootTuSize} = \min(\text{MaxTransformSize}, \text{PartitionSize}) \cdots \cdots (3)$

[0178] 也就是说,如果当前预测模式是帧内模式,则当TU大小标志是0时的变换单元大小“RootTuSize”可以是最大变换单元大小和当前分区单元的大小中的较小值。

[0179] 然而,在分区单元中根据预测模式的类型变化的当前最大变换单元大小“RootTuSize”只是示例,当前最大变化单元大小“RootTuSize”不限于此。

[0180] 现在将详细描述图4中示出的图像编码器400的帧内预测单元410和在图5中示出的图像解码器500的帧内预测单元550执行的帧内预测。在下面的描述中,编码单元表示图像的编码处理中的当前编码的块,解码单元表示在图像的解码处理中的当前解码的块。编码单元和解码单元的不同之处在于编码单元用于编码处理而解码单元用于解码。为了术语一致,除非特定的情况,编码单元和解码单元在编码和解码处理中都被称作编码单元。此外,通过本说明书,本领域的普通技术人员应该理解,根据示例性实施例的帧内预测方法和设备还可被用于在普通的视频编解码器中执行帧内预测。

[0181] 图14是根据示例性实施例的帧内预测设备1400的框图。参照图14,帧内预测设备1400包括预测器1410、确定器1415和后处理器1420。预测器 1410通过使用根据当前编码单元的大小确定的帧内预测模式来对当前编码单元进行帧内预测,并输出第一预测编码单元。确定器1415确定当前编码单元是否具有位于当前画面的边界之外的部分,并根据确定结果产生索引 MPI_PredMode。

[0182] 索引MPI_PredMode指示将执行哪种类型的多参数帧内预测(MPI),将在稍后详细描述。参照表2,如果索引MPI_PredMode是0,则指示不执行用于产生第二预测编码单元的MPI,如果索引MPI_PredMode大于0,则指示将执行MPI以产生第二预测编码单元。

[0183] [表2]

[0184]

MPI_PredMode	MPI模式名称	含义
0	MPI_Mode()	不执行MPI

1	MPI_Model	执行MPI

[0185]

MPI_PredMode1MAX	MPI_ModelMAX	执行MPI
------------------	--------------	-------

[0186] 根据表2,索引MPI_PredMode是0还是1取决于是否执行MPI。然而,在存在N个模式作为MPI模式的情况下,MPI_PredMode可具有范围从0到N的整数值以表示将不执行MPI以及N个模式的情况。

[0187] 如果确定器1415确定当前编码单元不包括位于画面的边界之外的部分即,当索引MPI_PredMode不是0时,则后处理器1420通过使用构成第一预测编码单元的像素的邻近像素执行MPI来产生第二预测编码单元,从而改变第一预测编码单元的像素的像素值。

[0188] 图15是示出根据示例性实施例的根据编码单元的大小的帧内预测模式的数量表。根据示例性实施例,可根据编码单元(在解码处理的情况下的解码单元)的大小确定帧内预测模式的数量。参照图15,如果将被执行帧内预测的编码单元的大小例如是 $N \times N$,则实际对大小为 $N_{Min} \times N_{Min}$ 、 $N_{Max} \times N_{Max}$ (N_{Min} 可以是2, N_{Max} 可以是128)的预测单元执行的帧内预测模式的数量可取决于预测单元大小。在示例2中,预测单元大小为 4×4 、 8×8 、 16×16 、 32×32 、 64×64 和 128×128 。在该示例中,帧内预测模式的数量分别是5、9、9、17、33、5和5。对于另一示例,当将被执行帧内预测的编码单元的大小是 $N \times N$ 时,实际对大小为 2×2 、 4×4 、 8×8 、 16×16 、 32×32 、 64×64 和 128×128 的编码单元执行的帧内预测模式的数量可以被设置为3、17、34、34、34、5和5。因为用于对预测模式信息进行编码的开销根据编码单元的大小而不同,所以可根据编码单元的大小确定实际将被执行的帧内预测模式的数量。换言之,虽然大小较小的编码单元占据整个图像数据的一小部分,但是用于发送关于大小较小的编码单元的附加信息(例如,预测模式)的开销可能很大。因此,当使用太多的预测模式来对大小较小的编码单元进行编码时,比特数会增加,因此会降低压缩效率。此外,大小较大的编码单元(例如,大小为 64×64 或更大的编码单元)很可能被选择作为用于图像的平面区域的编码单元。当使用过多的预测模式对被选择用于对这样的平面区域进行编码的大小较大的编码单元编码时,压缩效率也会降低。

[0189] 因此,根据示例性实施例,编码单元被大致划分为至少三种大小: $N_1 \times N_1$ ($2 = N_1 = 4$, N_1 表示整数), $N_2 \times N_2$ ($8 = N_2 = 32$, N_2 表示整数) 和 $N_3 \times N_3$ ($64 = N_3$, N_3 表示整数)。如果对大小为 $N_1 \times N_1$ 的每个编码单元执行帧内预测模式的数量是 A_1 (A_1 表示正整数),对大小为 $N_2 \times N_2$ 的每个编码单元执行帧内预测模式的数量是 A_2 (A_2 是正整数),并且对大小为 $N_3 \times N_3$ 的每个编码单元执行帧内预测模式的数量是 A_3 (A_3 表示正整数),则根据编码单元的大小执行的帧内预测模式的数量可被确定为满足 $A_3 = A_1 = A_2$ 。也就是说,如果当前画面被划分为大小较小的编码单元、大小中等的编码单元和大小较大的编码单元,则将对大小中等的编码单元执行的预测模式的数量可大于将对大小较小的编码单元执行的预测模式和将对大小较大的编码单元执行的预测模式的数量。然而,另外的示例性实施例并不限于此,也可设置为对大小较小的编码单元和大小中等的编码单元执行大量的预测模式。在图15中示出的根据每个编码单元的大小的预测模式的数量仅仅是示例,并且可以是可变的。

[0190] 图16A到图16C是用于解释根据示例性实施例的可对具有预定大小的编码单元执行的帧内预测模式的图。图16A是显示根据示例性实施例的可对具有预定大小的编码单元

执行的帧内预测模式的表。参照图15和图16A,例如,如果对大小为 4×4 的编码单元执行帧内预测,则可执行垂直模式(模式0)、水平模式(模式1)、直流(DC)模式(模式2)、斜的下-左模式(模式3)、斜的下-右模式(模式4)、垂直-右模式(模式5)、水平-下模式(模式6)、垂直-左模式(模式7)和水平-上模式(模式8)。

[0191] 图16B示出了根据示例性实施例的图16A中示出的帧内预测模式的方向。在图16B中,分配给箭头的值分别表示当按照箭头指示的方向执行预测时的模式值。这里,模式2是不具有方向性的DC模式,因此没有在图16B中示出。

[0192] 图16C示出了根据示例性实施例的可对图16A中示出的编码单元执行的帧内预测方法。参照图16C,按照根据当前编码单元的大小确定的可用帧内预测模式,使用当前编码单元的邻近像素A至邻近像素M来产生预测编码单元。例如,将描述根据图16A的垂直模式(模式0)对大小为 4×4 的当前编码单元执行预测编码的方法。首先,与 4×4 编码单元的顶部邻近的像素A到像素D的像素值被预测为 4×4 编码单元的像素值。具体地,像素A的像素值被预测为 4×4 编码单元的第一列的四个像素的像素值,像素B的像素值被预测为 4×4 编码单元的第二列的四个像素的像素值,像素C的像素值被预测为 4×4 编码单元的第三列的四个像素的像素值,像素D的像素值被预测为 4×4 编码单元的第四列的四个像素的像素值。然后,使用像素A到像素D预测的 4×4 预测编码单元中包括的像素的实际像素值和原始的 4×4 编码单元中包括的像素的实际像素值之间的误差值被计算和编码。

[0193] 图17是用于解释根据另一示例性实施例的可对具有预定大小的编码单元执行的帧内预测模式的图。参照图15和图17,例如,如果对大小为 2×2 的编码单元执行帧内预测,则可执行总共五个模式,例如,垂直模式、水平模式、DC模式、平面模式、斜的下-右模式。

[0194] 如图15所示,如果大小为 32×32 的编码单元具有33个帧内预测模式,则应该设置33个帧内预测模式的方向。根据示例性实施例,通过使用dx参数和dy参数来设置用于基于包括在编码单元中的像素选择将被用作参考像素的邻近像素的预测方向,从而设置除了以上参照图16和图17描述的帧内预测模式之外的具有不同方向性的帧内预测模式。例如,当33个预测模式的每一个预测模式被定义为模式N(N是从0到32的整数)时,模式0可被设置为垂直模式,模式1可被设置为水平模式,模式2可被设置为DC模式,模式3可被设置为平面模式,并通过使用被表示为如表3中所示的 $(1, -1)$ 、 $(1, 1)$ 、 $(1, 2)$ 、 $(2, 1)$ 、 $(1, -2)$ 、 $(2, 1)$ 、 $(1, -2)$ 、 $(2, -1)$ 、 $(2, -11)$ 、 $(5, -7)$ 、 $(10, -7)$ 、 $(11, 3)$ 、 $(4, 3)$ 、 $(1, 11)$ 、 $(1, -1)$ 、 $(12, -3)$ 、 $(1, -11)$ 、 $(1, -7)$ 、 $(3, -10)$ 、 $(5, -6)$ 、 $(7, -6)$ 、 $(7, -4)$ 、 $(11, 1)$ 、 $(6, 1)$ 、 $(8, 3)$ 、 $(5, 3)$ 、 $(5, 7)$ 、 $(2, 7)$ 、 $(5, -7)$ 和 $(4, -3)$ 之一的 (dx, dy) 来将模式4至模式31中的每一个设置为具有 $\tan^{-1}(dy/dx)$ 的方向的预测模式。

[0195] [表3]

[0196]

模式#	dx	dy	模式#	dx	dy
模式 4	1	-1	模式 18	1	-11
模式 5	1	1	模式 19	1	-7
模式 6	1	2	模式 20	3	-10
模式 7	2	1	模式 21	5	-6
模式 8	1	-2	模式 22	7	-6
模式 9	2	-1	模式 23	7	-4
模式 10	2	-11	模式 24	11	1
模式 11	5	-7	模式 25	6	1
模式 12	10	-7	模式 26	8	3
模式 13	11	3	模式 27	5	3
模式 14	4	3	模式 28	5	7

[0197]

模式 15	1	11	模式 29	2	7
模式 16	1	-1	模式 30	5	-7
模式 17	12	-3	模式 31	4	-3
模式 0 表示垂直模式，模式 1 表示水平模式，模式 2 表示 DC 模式，模式 3 表示平面模式，模式 32 表示双线性模式。					

[0198] 模式32可被设置为使用双线性插值的双线性模式，稍后将参照图19进行描述。

[0199] 图18A至图18C是用于解释根据示例性实施例的具有多种方向性的帧内预测模式的参考图。如上面参照表3所描述的，根据示例性实施例的每个帧内预测模式可通过使用多个(dx,dy)参数而具有多个 $\tan^{-1}(dy/dx)$ 的方向。

[0200] 参照图18A，从将被预测的当前编码单元中的当前像素P延伸出的直线 180 (其角度为通过表3中显示的根据模式的(dx,dy)参数的值确定的 $\tan^{-1}(dy/dx)$)上的邻近像素A和B可用作当前像素P的预测值。在这种情况下，邻近像素A和B可以是已经被编码和恢复的像素，并且属于位于当前编码单元之上和左侧的先前编码单元。此外，当直线180没有穿过具有整数值的位置上的每个邻近像素而是在这些邻近像素之间穿过时，较为接近直线180的邻近像素可用作当前像素P的预测值。如果存在与直线180相交的两个像素（例如，位于当前像素P的上面的邻近像素A和位于当前像素P的左侧的邻近像素B），则邻近像素A和邻近像素B的像素值的平均可用作当前像素的预测值。否则，如果dx和dy参数的值的乘积是正值，则可使用邻近像素A，如果dx和dy参数的值的乘积是负值，则可使用邻近像素B。

[0201] 图18B和图18C是用于解释当图18A的延长线180在整数位置的邻近像素之间穿过而不通过邻近像素时产生预测值的处理的参考图。

[0202] 参照图18B，如果如上所述，具有根据每个模式的(dx,dy)值确定的 $\tan^{-1}(dy/dx)$ 的角度的延长线180穿过整数位置的邻近像素A 181和邻近像素B 182之间，则考虑延长线180的交点与接近延长线180的邻近像素A 181和邻近像素B 182之间的距离而计算的加权

平均值可被用作预测值。例如,如果具有 $\tan^{-1}(dy/dx)$ 的角度的延长线180的交点和邻近像素A 181之间的距离为f,并且延长线180的交点和邻近像素B 182之间的距离是g,则当前像素P的预测值可被获得为 $(A \times g + B \times f) / (f + g)$ 。这里,f和g可以被归一化为使用整数的距离。如果使用软件或硬件,则可通过如(比如 $(A \times g + B \times f + 2) \gg 2$)的移位运算获得当前像素P的预测值。如图18B所示,如果延长线180穿过将整数位置的邻近像素A 181和邻近像素B 182之间的距离四等分而获得的四个部分中的接近于邻近像素A 181的1/4处,则当前像素P的预测值可被获得为 $(3 \times A + B) / 4$ 。可通过考虑取整至最接近的整数的移位运算(例如, $(3 \times A + B + 2) \gg 2$)来执行该操作。

[0203] 同时,如果具有根据每个模式的(dx,dy)确定的 $\tan^{-1}(dy/dx)$ 的角度的延长线180在整数位置的邻近像素A 181和邻近像素B 182之间穿过,则邻近像素A 181和邻近像素B 182之间的线段(section)可被划分为预定数量的区间,并且考虑每个划分的区间中的交点与邻近像素A 181和邻近像素B 182之间的距离而计算的加权平均值可被用作预测值。例如,参照图18C,邻近像素A 181和邻近像素B 182之间的线段被划分为五个线段P1至P5,考虑每个线段中的交点与邻近像素A 181和邻近像素B 182之间的距离而计算的加权平均值可被确定,并可被用作当前像素P的预测值。具体地,如果延长线180穿过线段P1,则邻近像素A 181的值可被确定为当前像素P的预测值。具体地,如果延长线180穿过线段P1,则邻近像素A的值可被确定为当前像素P的预测值。如果延长线180穿过线段P2,则考虑线段P2的中点与邻近像素A 181和邻近像素B 182之间的距离计算的加权平均值(即 $(3 \times A + 1 \times B + 2) \gg 2$)可被确定为当前像素P的预测值。如果延长线180穿过线段P3,则考虑线段P3的中点与邻近像素A 181和邻近像素B 182之间的距离计算的加权平均值(即 $(2 \times A + 2 \times B + 2) \gg 2$)可被确定为当前像素P的预测值。如果延长线180穿过线段P4,则考虑线段P4的中心与邻近像素A 181和邻近像素B 182之间的距离计算的加权平均值(即 $(1 \times A + 3 \times B + 2) \gg 2$)可被确定为当前像素P的预测值。如果延长线180穿过线段P5,则邻近像素B 182的值可被确定为当前像素P的预测值。

[0204] 此外,如图18A所示,如果两个邻近像素(即,上侧的邻近像素A和左侧的邻近像素B)与延长线180相交,则邻近像素A和邻近像素B的平均值可被用作当前像素P的预测值。作为选择,如果 $(dx \times dy)$ 的值是正值,则可使用邻近像素A,如果 $(dx \times dy)$ 是负值,则可使用邻近像素B。

[0205] 可在编码器侧和解码器侧预先确定表3中显示的具有多种方向的帧内预测模式,可仅发送与每个编码单元的帧内预测模式的索引。

[0206] 图19是用于解释根据示例性实施例的双线性模式的参考图。参照图19,在双线性模式下,通过考虑将被预测的当前编码单元中的当前像素P 190的值、当前编码单元的上、下、左、右边界上的像素的值以及当前像素P 190和当前编码单元的上、下、左、右边界之间的距离来计算几何平均,几何平均随后被用作当前像素P 190的预测值。例如,在双线性模式下,使用位于当前像素P 190的上、下、左、右侧的虚拟像素A 191、虚拟像素B 192、像素D 196和像素E 197以及当前像素P 190和当前编码单元的上、下、左、右边界之间的距离计算的几何平均被用作当前像素P的预测值。由于双线性是帧内预测模式之一,因此已经被编码和恢复且属于先前编码单元的邻近像素被用于预测的参考像素。因此,当前编码单元中的像素值不被使用,而是使用位于当前编码单元的上侧和左侧的邻近像素计算的虚拟像

素值被用作像素A 191和像素B 192。

[0207] 具体地,首先,通过计算当前编码单元的最右上点上的邻近像素(右上像素)194和当前编码单元的最左下点上的邻近像素(左下像素)195的值的平均来计算当前编码单元的最右上点上的虚拟像素C 193的值,如以下的等式所示:

$$[0208] \quad C = 0.5 (\text{LeftDownPixel} + \text{RightUpPixel}) \quad (4)$$

[0209] 可通过如等式4的移位运算获得的虚拟像素C 193可以是用于当前像素 P 的预测值,可通过如下的移位运算获得虚拟像素C 193:

$$[0210] \quad C = 0.5 (\text{LeftDownPixel} + \text{RightUpPixel} + 1) \gg 1。$$

[0211] 接下来,当通过考虑当前像素P 190和当前编码单元的左边界之间的距离W1以及当前像素P 190和当前编码单元的右边界之间的距离W2而使当前像素P 190向下延长时位于当前编码单元的最底边界上的虚拟像素A 191的值可通过使用以下等式来计算:

$$[0212] \quad A = (C \times W1 + \text{LeftDownPixel} \times W2) / (W1 + W2)$$

$$[0213] \quad A = (C \times W1 + \text{LeftDownPixel} \times W2 + ((W1 + W2) / 2)) / (W1 + W2) \quad (5)$$

[0214] 当等式5中的W1+W2的值是2的若干次方(如 2^n)时,可通过如 $A = (C \times W1 + \text{LeftDownPixel} \times W2 + 2^{(n-1)}) \gg n$ 的移位运算而不进行除法来计算 $A = (C \times W1 + \text{LeftDownPixel} \times W2 + ((W1 + W2) / 2)) / (W1 + W2)$ 。

[0215] 类似地,当通过考虑当前像素P 190和当前编码单元的上边界之间的距离h1以及当前像素P 190和当前编码单元的下边界之间的距离h2而使当前像素P 190向右延长时位于当前编码单元的最右边界上的虚拟像素B 192的值可通过使用以下等式来计算:

$$[0216] \quad B = (C \times h1 + \text{RightUpPixel} \times h2) / (h1 + h2)$$

$$[0217] \quad B = (C \times h1 + \text{RightUpPixel} \times h2 + ((h1 + h2) / 2)) / (h1 + h2) \quad (6)$$

[0218] 当等式6中的h1+h2的值是2的若干次方(如 2^m)时,可通过如 $B = (C \times h1 + \text{RightUpPixel} \times h2 + 2^{(m-1)}) \gg m$ 的移位运算而不进行除法来计算 $B = (C \times h1 + \text{RightUpPixel} \times h2 + ((h1 + h2) / 2)) / (h1 + h2)$ 。

[0219] 一旦通过使用等式(4)到等式(6)确定了当前像素P 190的右边界上的虚拟像素B 192和下边界上的虚拟像素A的值,可通过使用A+B+D+E的平均值确定当前像素P 190的预测值。具体地,考虑当前像素P 190与虚拟像素A 191、虚拟像素B 192、像素D 196和像素E 197之间的距离的加权平均值或者 A+B+D+E的平均值可被用作当前像素P 190的预测值。例如,如果使用加权平均值且块的大小为 16×16 ,则当前像素P的预测值可获得为 $(h1 \times A + h2 \times D + W1 \times B + W2 \times E + 16) \gg 5$ 。这样的双线性预测被应用于当前编码单元中的所有像素,产生双线性预测模式下的当前编码单元的预测编码单元。

[0220] 根据示例性实施例,按照根据编码单元的大小确定的多种帧内预测模式执行预测编码,从而允许基于图像的特性进行有效的视频压缩。

[0221] 同时,如参照图18A到图18C所述,如果通过使用延长线180上或接近延长线180的邻近像素来产生当前像素P的预测值,则延长线180实际具有 $\tan^{-1}(dy/dx)$ 的方向。为了计算方向,由于除法(dy/dx)是必要的,因此当使用硬件或软件时,计算进行到小数位,从而增加了计算量。因此,当通过以与参照表3描述的方式相似的方式,使用dx和dy参数来设置用于选择将用作关于编码单元中的像素的参考像素的邻近像素的预测方向时,使用设置dx和dy的处理以减少计算量。

[0222] 图27是用于解释根据示例性实施例的当前像素和位于具有 (dy/dx) 的方向的延长线上的邻近像素之间的关系的示意图。

[0223] 参照图27,假设当前像素P的位置是 $P(j, i)$,位于延长线2710上的上邻近像素和左邻近像素B分别是A和B,其中,延长线2710具有倾斜的 $\tan^{-1}(dy/dx)$ 的方向且穿过当前像素P。当假设上邻近像素的位置对应于坐标平面上的x轴,左邻近像素的位置对应于坐标平面上的y轴,上邻近像素A位于 $(j+i \times dx/dy, 0)$,左邻近像素B位于 $(0, i+j \times dy/dx)$ 。因此,为了确定用于预测当前像素P的上邻近像素A和左邻近像素B中的任意一个,需要进行除法,诸如 dx/dy 或者 dy/dx 。如上所述,这样的除法非常复杂,从而降低了软件或硬件的计算速度。

[0224] 因此,表示用于确定邻近像素的预测模式的方向的 dx 和 dy 中的任意一个的值可被确定为2的若干次方。也就是说,当 n 和 m 是整数时, dx 和 dy 可以分别是 2^n 和 2^m 。

[0225] 参照图27,如果左邻近像素B用作当前像素P的预测值且 dx 具有值 2^n ,则确定作为左邻近像素B的位置的 $(0, i+j \times dy/dx)$ 所需的 $j \times dy/dx$ 变为 $(j \times dy / (2^n))$,通过如 $(j \times dy) \gg n$ 的移位运算容易获得使用这样的2的若干次方的除法,从而减少了计算量。

[0226] 同样地,如果上邻近像素A用作当前像素P的预测值且 dy 具有值 2^m ,则确定作为左邻近像素A的位置的 $(j+i \times dx/dy, 0)$ 所需的 $i \times dx/dy$ 变为 $(i \times dx) / (2^m)$,通过如 $(i \times dx) \gg m$ 的移位运算容易获得使用这样的2的若干次方的除法。

[0227] 图28是用于解释根据示例性实施例的位于具有根据当前像素的位置的 (dx, dy) 的方向的延长线上的邻近像素的改变的示意图。

[0228] 由于邻近像素对于根据当前像素的位置的预测是必须的,因此,选择上邻近像素和左邻近像素中的任意一个。

[0229] 参照图28,当当前像素2810是 $P(j, i)$,并通过使用位于预测方向上的邻近像素来预测当前像素2810时,上像素A用于预测当前像素P 2810。当当前像素2810是 $Q(b, a)$ 时,左像素B用于预测当前像素Q 2820。

[0230] 如果代表预测方向的 (dx, dy) 中仅y轴方向的 dy 分量具有2的若干次方(如 2^m),而可通过不需要除法的移位运算(诸如, $(j+(i \times dx) \gg m), 0)$ 确定图24中的上像素A,则左像素B需要除法,诸如 $(0, a+b \times 2^m/dx)$ 。因此,为了在针对当前块的所有像素产生预测值时排除除法,所有的 dx 和 dy 可具有2的若干次方的类型。

[0231] 图29和图30是用于解释根据示例性实施例的确定帧内预测模式方向的方法的示意图。

[0232] 通常,在很多情况下,图像或视频信号中显示的线性样式是垂直或水平的。因此,当通过使用参数 dx 和 dy 定义具有多种方向的帧内预测模式时,可通过如下定义 dx 和 dy 的值来提高图像编码效率。

[0233] 具体地,如果 dy 具有固定值 2^m ,则 dx 的绝对值可被设置为使得接近垂直方向的多个预测方向之间的距离很窄,而更接近于水平方向的多个预测模式之间的距离较宽。例如,参照图29,如果 dy 具有值 2^4 (即,16),则 dx 的值可被设置为1、2、3、4、6、9、12、16、0、-1、-2、-3、-4、-6、-9、-12和-16,从而使接近于垂直方向的多个预测方向之间的距离很窄而更接近于水平方向的多个预测模式之间的距离较宽。

[0234] 同样地,如果 dx 具有固定值 2^n ,则 dy 的绝对值可被设置为使得接近于水平方向的多个预测方向之间的距离很窄,而更接近于垂直方向的多个预测模式之间的距离较宽。例

如,参照图30,如果dx具有值 2^4 (即,16),则 dy的值可被设置为1、2、3、4、6、9、12、16、0、-1、-2、-3、-4、-6、-9、-12和-16,从而使接近于水平方向的多个预测方向之间的距离很窄而更接近于垂直方向的多个预测模式之间的距离较宽。

[0235] 另外,当dx和dy的值之一为固定时,剩余的值可被设置为根据预测模式增加。例如,如果dy固定,则dx之间的距离可被设置为根据预定值增加。另外,可按照预定单位划分水平方向和垂直方向的角度,并且可在每个划分的角度中设置增量。例如,如果dy固定,则dx的值可被设置为在小于15度的区间中具有增量a,在15度到30度之间的区间中具有增量b,在大于30度的区间中具有增加宽度c。在这种情况下,为了具有如图27所示的形状,可将dx的值设置为满足关系 $a < b < c$ 。

[0236] 例如,可通过使用如表4到表6中示出的(dx,dy)将参照图27到图30描述的预测模式定义为具有 $\tan^{-1}(dy/dx)$ 的方向的预测模式。

[0237] [表4]

[0238]

dx	Dy	dx	dy	dx	dy
-32	32	21	32	32	13
-26	32	26	32	32	17
-21	32	32	32	32	21
-17	32	32	-26	32	26
-13	32	32	-21	32	32
-9	32	32	-17		
-5	32	32	-13		
-2	32	32	-9		
0	32	32	-5		

[0239]

2	32	32	-2		
5	32	32	0		
9	32	32	2		
13	32	32	5		
17	32	32	9		

[0240] 表5

[0241]

dx	Dy	dx	dy	dx	dy
-32	32	19	32	32	10
-25	32	25	32	32	14
19	32	32	32	32	19
-14	32	32	-25	32	25
-10	32	32	-19	32	32
-6	32	32	-14		
-3	32	32	-10		

-1	32	32	-6		
0	32	32	-3		
1	32	32	-1		
3	32	32	0		
6	32	32	1		
10	32	32	3		
14	32	32	6		

[0242] 表6

[0243]

dx	Dy	dx	dy	dx	dy
-32	32	23	32	32	15
-27	32	27	32	32	19
-23	32	32	32	32	23
-19	32	32	-27	32	27
-15	32	32	-23	32	32
-11	32	32	-19		
-7	32	32	-15		

[0244]

-3	32	32	-11		
0	32	32	-7		
3	32	32	-3		
7	32	32	0		
11	32	32	3		
15	32	32	7		
19	32	32	11		

[0245] 如上所述,使用帧内预测模式产生的预测编码单元具有根据帧内预测模式的方向,其中,由图14的帧内预测设备1400的预测器1410根据当前编码单元的大小确定所述帧内预测模式。当将被预测的当前编码单元的像素具有预定方向时,预测编码单元中的方向可改进预测效率,但是当这些像素不具有预定方向时,会导致预测效率降低。因此,作为针对通过帧内预测产生的预测编码单元的后处理,后处理器1420可通过使用预测编码单元中的像素和至少一个邻近像素改变预测编码单元中的像素的值来产生新的预测编码单元。在这种情况下,后处理器1420不对所有预测编码单元执行后处理,而是可仅当确定器1415确定当前预测编码单元不包括位于画面的边界之外的部分时(即,当索引MPI_PredMode不为0时)执行后处理。

[0246] 图24是用于解释根据示例性实施例的用于后处理的索引处理的参考图。参照图24,如果当前画面2410被划分并编码为具有预定大小的编码单元,并且当前画面2410的宽度Frame_width不是每个编码单元的水平长度的倍数或者当前画面2410的高度Frame_height不是每个编码单元的垂直长度的倍数,则编码单元的某些部分(用斜线指示)延伸超出当前画面2410的右边界和下边界,如图24所示。图14的确定器1415可将延伸超出当前画

面的边界的编码单元的预定索引MPI_PredMode设置为0,从而图14的后处理器1420 可跳过这些编码单元的后处理。

[0247] 在当前预测编码单元具有位于当前编码单元的边界之外的部分时不执行后处理的原因在于每个像素的邻近像素用于后处理,当前预测编码单元中的像素缺少邻近像素。即使通过填充或外推产生邻近像素来执行后处理,但由于产生的邻近像素是原来不存在的像素,因此预测效率不高。

[0248] 图25是用于解释根据另一示例性实施例的用于后处理编码单元的索引处理的参考图。参照图24,如果图14的确定器1415确定编码单元2520延伸超出了画面的边界2510,则编码单元2520可被划分为比编码单元2520的深度更深的深度的较深层编码单元,并且可确定每个较深层编码单元是否具有位于画面的边界之外的部分,而不对全部编码单元2520进行后处理。可重复执行这样的划分处理直到延伸超出画面的边界2510的编码单元(例如,编码单元2524和2528)是最小编码单元且不能再被划分为止,即,直到当前深度为最大深度为止。参照图25,位于边界2510之外的最小编码单元2524和 2528的索引MPI_PredMode被设置为0,从而最小编码单元2524和2528不会被后处理,位于边界2510之内的最小编码单元2522和2526的索引 MPI_PredMode被设置为1,从而可对最小编码单元2522和2526进行后处理。

[0249] 现在将描述根据示例性实施例的由图14的后处理器1420对预测编码单元执行后处理的方法。

[0250] 如果图14的确定器1415确定当前编码单元不包括位于画面的边界之外的部分,则后处理器1420通过使用由图14的预测器140产生的第一预测编码单元的像素和至少一个邻近像素执行后处理来改变构成第一预测编码单元的像素的值,从而产生第二预测编码单元。如上所述,预测器1410通过使用根据当前编码单元的大小确定的帧内预测模式来产生第一预测编码单元。

[0251] 图20是用于解释根据示例性实施例的第一预测编码单元的后处理的参考图。在图20中,标号2010到2060指示按时间顺序由后处理器1420改变第一预测编码单元中的像素的值的处理。

[0252] 参照图20,图14的后处理器1420通过计算将被改变的第一预测编码单元2010中的像素的值和所述像素的邻近像素的值的加权平均来改变第一预测编码单元2010中的像素的值。例如,参照图20,如果将被改变的第一预测编码单元2010的像素2021的值为 $f[1][1]$,位于像素2021的上面的像素2022 的值是 $f[0][1]$,位于像素2021的左侧的像素2023为 $f[1][0]$,改变像素2021 的值 $f[1][1]$ 的结果是 $f'[1][1]$,则可使用以下等式来计算 $f'[1][1]$:

[0253] $f[1][1] = (f[0][1] + f[1][0] + (f[1][1] \ll 1) + 2) \gg 2$

[0254]
$$f'[1][1] = \frac{f[0][1] + f[1][0] + 2 \times f[1][1]}{4} \quad (7)$$

[0255] 如图20所示,后处理器1420通过按照从第一预测编码单元的最左上点到第一预测编码单元的最右下点的方向,计算第一预测编码单元的每个像素的值以及位于所述像素的上面的像素和位于所述像素左侧的像素的值的加权平均,来改变第一预测编码单元2010中所包括的像素的值。然而,根据另一示例性实施例的这样的后处理操作不限于此,并且可按照从第一预测编码单元的最右上点到第一预测编码单元的最左下点的方向,或者按照从第一预测编码单元的最右下点到第一预测编码单元的最左上点的方向,对第一预测编码单元

的像素顺序地执行后处理。例如,如果后处理器1420按照如图20所示的从最右上点到最左下点的方向改变第一预测编码单元中的像素的值,则通过计算第一预测编码单元的每个像素的值和位于第一预测编码单元的下侧和右侧的像素的值的加权平均来改变第一预测编码单元的像素的值。

[0256] 图21是用于解释根据示例性实施例的图14的后处理器1420的操作的参考图。图22是用于解释根据示例性实施例的由后处理器使用的邻近像素的参考图。在图21中,标号2110表示将被改变的第一预测编码单元2100的第一像素,标号2111到2118表示第一像素2110的邻近像素。

[0257] 在当前示例性实施例(第一示例性实施例)中,与图20中示出的不同,第一像素2110的邻近像素不限于位于第一预测编码单元的上面和左侧的像素。参照图21,后处理器1420可通过使用从邻近像素2111到2118选出的预定数量的邻近像素来对第一像素2110进行后处理。也就是说,参照图22,从当前编码单元的第一像素c的邻近像素P1到P8中选择预定数量的像素,并通过对选择的邻近像素和第一像素c执行预定操作来改变第一像素c的值。例如,如果第一预测编码单元2100的大小是 $m \times n$,将被改变且位于第一预测编码单元2100的第i列第j行的第一像素2110的值是 $f[i][j]$,则从第一像素2110的邻近像素2111到2118中选择以对第一像素2110进行后处理的n个像素的值分别是f1到fn,则后处理器1420通过使用以下等式将第一像素2110的值从 $f[i][j]$ 改变为 $f'[i][j]$ 。这里,m表示正整数,n是2或3,i表示从0到m-1的整数,j表示从0到n-1的整数。

$$[0258] \quad f'[i][j] = \frac{f1 + f2 + 2 \times [i][j] + 2}{4} \quad (n=2)$$

$$[0259] \quad f'[i][j] = \frac{f1 + f2 + f3 + f[i][j]}{4} \quad (n=3) \quad (8)$$

[0260] 后处理器1420通过使用等式(8)改变包括在第一预测编码单元2100中的所有像素的值来产生第二预测编码单元。在等式(8)中,使用三个邻近像素,但是另一示例性实施例不限于此,后处理器1420可通过使用四个或更多邻近像素来执行后处理。

[0261] 根据第二示例性实施例,后处理器1420通过使用将被改变的第一预测编码单元2100的像素的值和所述像素的邻近像素的值的加权调和平均,来改变第一预测编码单元的每个像素的值,从而产生第二预测编码单元。

[0262] 例如,后处理器1420通过使用位于第一预测编码单元2100的第i列第j行的像素的上面和左侧的邻近像素,将该像素的值从 $f[i][j]$ 改变为 $f'[i][j]$,如以下等式所示:

$$[0263] \quad f'[i][j] = \frac{\alpha + \beta + \gamma}{\frac{\alpha}{f[i-1][j]} + \frac{\beta}{f[i][j-1]} + \frac{\gamma}{f[i][j]}} \quad (9)$$

[0264] 其中, α, β, γ 表示正整数,例如, $\alpha=2, \beta=2, \gamma=1$ 。

[0265] 根据第三示例性实施例,后处理器1420通过使用将被改变的第一预测编码单元的像素的值和该像素的邻近像素的值的加权几何平均来改变第一预测编码单元2100的每个像素的值,从而产生第二预测编码单元。

[0266] 例如,后处理器1420通过使用位于第一预测编码单元2100的第i列第j行的像素的上面和左侧的邻近像素,将该像素的值从 $f[i][j]$ 改变为 $f'[i][j]$,如以下等式所示:

$$[0267] \quad f'[i][j] = (f[i-1][j]^{\alpha} \times f[i][j-1]^{\beta} \times f[i][j]^{\gamma})^{\frac{1}{(\alpha+\beta+\gamma)}} \quad (10)$$

[0268] 其中, α, β, γ 表示正整数, 例如, $\alpha=2, \beta=2, \gamma=1$ 。在等式 (8) 到等式 (10) 中, 相对大的权重被分配给将被改变的像素的值 $f[i][j]$ 。

[0269] 如上所述, 在第一示例性实施例到第三示例性实施例中, 后处理器1420 可通过不仅使用位于将被改变的像素的上面和左侧的邻近像素还使用从如图 21所示的邻近像素2111到2118中选择的预定数量的邻近像素来执行后处理。

[0270] 根据第四示例性实施例, 后处理器1420通过使用将被改变的第一预测编码单元中的像素的值和从所述像素的邻近像素中选择一个邻近像素的值的平均来改变第一预测编码单元中的每个像素的值, 从而产生第二预测编码单元。

[0271] 例如, 后处理器1420通过使用位于第一预测编码单元2100的第*i*列第*j*行的像素的上面的邻近像素, 将该像素的值从 $f[i][j]$ 改变为 $f'[i][j]$, 如以下等式所示:

$$[0272] \quad f[i][j] = (f[i-1][j] + f[i][j-1] + 1) \gg 1 \quad (11)$$

[0273] 类似地, 根据第五示例性实施例, 后处理器1420通过使用将被改变的第一预测编码单元中的像素的值和位于该像素的左侧的邻近像素的值的平均来改变第一预测编码单元中的每个像素的值, 从而产生第二预测编码单元。

[0274] 换言之, 后处理器1420将第一预测编码单元2100的第*i*列第*j*行的像素的值从 $f[i][j]$ 改变为 $f'[i][j]$, 如以下等式所示:

$$[0275] \quad f[i][j] = (f[i-1][j] + f[i][j-1] + 1) \gg 1 \quad (12)$$

[0276] 根据第六示例性实施例, 后处理器1420通过使用将被改变的第一预测编码单元中的像素的值和该像素的邻近像素的值之间的中间值来改变第一预测编码单元中的每个像素的值, 从而产生第二预测编码单元。参照图21, 例如, 假设在块大小方面, 位于第一预测编码单元2100的第*i*列第*j*行的第一像素 2110的值 $f[i][j]$ 、第二像素2112的值 $f[i][j-1]$ 和第三像素2111的值 $f[i-1][j]$ 具有关系 $f[i][j-1] > f[i-1][j] > f[i][j]$ 。在这种情况下, 后处理器1420将第一像素 2110的值 $f[i][j]$ 改变为第一像素2110到第三像素2112中的中间值 $f[i-1][j]$ 。

[0277] 在第七示例性实施例中, 后处理器1420通过使用与当前编码单元邻近的已被编码和恢复的先前编码单元来产生第二预测编码单元, 而不是通过使用将被改变的像素的邻近像素来产生第二预测编码单元。

[0278] 参照回图21, 在第七示例性实施例中, 后处理器1420通过计算位于第一预测编码单元2100的第*i*列第*j*行的第一像素2110的值和位于与第一像素 2110相同列且包括在与当前编码单元的顶部邻近的编码单元中的像素2121 的值的平均, 来将第一像素2110的值改变为 $f[i][j]$, 如以下等式所示:

$$[0279] \quad f[i][j] = (f[i][j] + f[-1][j] + 1) \gg 1 \quad (13)$$

[0280] 其中, $f[-1][j]$ 表示像素2121的值。

[0281] 类似地, 在第八示例性实施例, 后处理器1420通过计算位于第一预测编码单元2100的第*i*列第*j*行的第一像素2110的值和位于与第一像素2110相同行且包括在与当前编码单元的左侧邻近的编码单元中的像素2122的值的平均, 来将第一像素2110的值改变为 $f[i][j]$, 如以下等式所示:

$$[0282] \quad f[i][j] = (f[i][j] + f[j][-1] + 1) >> 1 \quad (14)$$

[0283] 其中, $f[j][-1]$ 表示像素2122的值。

[0284] 在第九示例性实施例中, 后处理器1420通过计算位于第一预测编码单元 2100的第*i*列第*j*行的第一像素2110的值、位于与第一像素2110相同列且包括在与当前编码单元的顶部邻近的编码单元中的像素2121的值以及位于与第一像素2110相同行且包括在与当前编码单元的左侧邻近的编码单元中的像素2122的值的加权平均, 来将第一像素2110的值改变为 $f[i][j]$, 如以下等式所示:

$$[0285] \quad f[i][j] = ((f[i][j] << 1) + f[j][-1] + f[i][j-1] + 2) >> 2 \quad (15)$$

[0286] 在第十示例性实施例中, 后处理器1420通过使用以下等式之一来使将被改变的第一预测编码单元2100的第一像素2110的值从 $f[i][j]$ 改变为 $f'[i][j]$ 。

$$[0287] \quad f'[i][j] = \min(f[i][j] + i, 255) \quad (16)$$

$$[0288] \quad f'[i][j] = \min(f[i][j] + j, 255) \quad (17)$$

$$[0289] \quad f'[i][j] = \max(f[i][j] - i, 0) \quad (18)$$

$$[0290] \quad f'[i][j] = \max(f[i][j] - j, 0) \quad (19)$$

[0291] 在等式(16)中, 第一预测编码单元2100的像素值被改变为: 在第一预测编码单元2100的列单元中, 从顶部到底部逐渐增加。在等式(17)中, 第一预测编码单元2100的像素值被改变为: 在第一预测编码单元2100的行单元中, 以向右的方向逐渐增加。在等式(18)中, 第一预测编码单元2100的像素值被改变为: 在第一预测编码单元2100的列单元中, 从顶部到底部逐渐减小。在等式(19)中, 第一预测编码单元2100的像素值被改变为: 在第一预测编码单元2100的行单元中, 以向右的方向逐渐减小。

[0292] 在第十一示例性实施例中, 如果位于第一预测编码单元2100的第*i*列第 *j*行且将被改变的第一像素2110的值是 $f[i][j]$, 位于第一预测编码单元2100 的最左上点的像素的值是 $f[0][0]$, 与第一像素2110位于相同的第*j*列并且位于第一预测编码单元2100的最顶点的像素的值是 $f[0][j]$, 与第一像素位于相同的第*i*行并且位于在第一预测编码单元的最左点的像素的值是 $f[i][0]$, 并且 $G[i][j] = f[i][0] + f[0][j] - f[0][0]$, 则后处理器1420将第一像素2110的值改变为 $f'[i][j]$, 如以下的等式所示:

$$[0293] \quad f'[i][j] = (f[i][j] + G[i][j]) / 2 \quad (20)$$

[0294] 等式(20)基于波动方程, 其中, 通过以下的方式来改变第一预测编码单元2100中的每个像素的值: 通过将第一预测编码单元2100的最顶点的像素和最左边的点的像素的值设置为使第一预测编码单元2100中的每个像素的值平滑的边界条件来计算值 $G[i][j]$, 然后计算值 $G[i][j]$ 和 $f[i][j]$ 的平均。

[0295] 另外, 如果将被改变的第一预测编码单元的第*x*列和第*y*行的第一像素的值是 $f[x][y]$, 位于第一像素的上、下、左、右侧的邻近像素的值分别是 $f[x-1][y]$ 、 $f[x+1][y]$ 、 $f[x][y-1]$ 和 $f[x][y+1]$, 则后处理器1420可通过使用以下移位预算之一将第一像素的值改变为 $f[x][y]$:

$$[0296] \quad f[x, y] = (f[x, y] + f[x-1, y] + f[x, y-1] + f[x, y+1] + 2) >> 2$$

$$[0297] \quad f[x, y] = (f[x, y] + f[x-1, y] + f[x, y-1] + f[x-1, y-1] + 2) >> 2$$

$$[0298] \quad f[x, y] = (2 \times f[x, y] + f[x+1, y] + f[x, y-1] + 2) >> 2$$

$$[0299] \quad f[x, y] = (2 \times f[x, y] + f[x-1, y] + f[x, y-1] + 2) >> 2$$

[0300] $f[x,y] = (f[x,y] + f[x+1,y] + f[x,y+1] + f[x,y-1] + 2) >> 2$

[0301] $f[x,y] = (f[x,y] + f[x-1,y] + f[x,y+1] + f[x,y-1] + 2) >> 2$

[0302] 另外,后处理器1420可通过使用第一像素和第一像素的邻近像素产生中间值,并通过使用中间值改变第一像素的值。例如,可通过使用等式 $t[x,y] = (2 \times f[x,y] + f[x-1,y] + f[x,y-1] + 2) >> 2$, $f[x,y] = t[x,y]$ 设置中间值 $t[x,y]$ 来改变第一像素的值。类似地,可通过使用等式 $t[x,y] = \text{median}(f[x,y], f[x-1,y], f[x,y-1])$ 来计算第一像素和邻近像素之间的中间值 $t[x,y]$,并可将中间值 $t[x][y]$ 确定为第一像素的改变的值。

[0303] 另外,后处理器1420可通过使用以下等式改变第一像素的值:

```
[0304] {
[0305]   t[x][y] = f[x][y]
[0306]   for (Int iter = 0; iter < iterMax; iter++)
[0307]   {
[0308]     laplacian[x,y] = (t[x,y] << 2) - t[x-1,y] - t[x+1,y] - t[x,y-1] - t[x,y+1]
[0309]     t[x,y] = (a × t[x,y] + laplacian[x,y]) / a
[0310]   }
[0311]   f[x,y] = t[x,y]
[0312] }
```

[0313] 这里,iterMax可以是5,a可以是16。

[0314] 将包含对第二预测编码单元进行编码的结果的比特流的代价进行相互比较,并随后将多个后处理模式中具有最小代价的后处理模式添加到比特流的头,其中,分别使用根据以上的第一实施例到第十一实施例的后处理模式来产生第二预测编码单元。当后处理模式被添加到比特流时,可以通过使用可变长度编码来将不同的后处理模式表示为互不相同,在可变长度编码中,基于在对预定数量的编码单元的编码完成之后确定的后处理模式的分布,将少量的比特分配给最频繁使用的后处理模式。例如,如果根据第一示例性实施例的后处理模式是形成大部分编码单元的最小代价的最优操作,则将最少数量的比特分配给指示该后处理模式的索引,从而可使该后处理模式与其它后处理模式相区别。

[0315] 当编码单元被划分为子编码单元并在子编码单元中执行预测时,可通过将不同的后处理模式分别应用于子编码单元来产生第二预测编码单元,或者通过将相同的后处理模式应用于属于相同编码单元的子编码单元来产生第二预测编码单元,从而简化计算并减少开销率(overhead rate)。

[0316] 率失真优化方法可用作确定最优后处理模式的代价。由于对用作另一编码单元的参考数据的帧内预测编码单元执行根据示例性实施例的视频编码方法,因此与率失真优化方法相比,可通过将高权重分配给失真来计算代价。也就是说,在率失真优化方法中,基于作为编码图像和原始图像之间的差异的失真以及产生的比特率来计算代价,如以下的等式所示:

[0317] 代价 = 失真 + 比特率 (21)

[0318] 相反,在根据示例性实施例的视频编码方法中,与率失真优化方法相比,通过将高权重分配给失真来确定最优后处理模式,如以下的等式所示:

[0319] 代价 = $a \times$ 失真 + 比特率 (a表示等于或大于2的实数) (22)

[0320] 图23是示出根据示例性实施例的对视频编码的方法的流程图。参照图 23,在操作2310,产生将被编码的当前编码单元的第一预测编码单元。第一预测编码单元是通过执行一般帧内预测方法以及根据编码单元的大小确定的具有多个方向的多种帧内预测模式之一而产生的帧内预测块。

[0321] 在操作2320,确定当前编码单元是否具有位于当前画面的边界之外的部分。可按照这样的方式根据确定结果产生预定索引MPI_PredMode:当预定索引MPI_PredMode是0时将不执行产生第二预测编码单元的后处理,当预定索引MPI_PredMode是1时将执行产生第二预测编码单元的后处理。

[0322] 如果在操作2320确定当前编码单元不具有位于当前画面的边界之外的部分,则在操作2330,通过使用第一预测编码单元的每个像素和至少一个邻近像素来改变第一预测编码单元的每个像素的值,从而产生第二预测编码单元。如上所述,在关于后处理器1420的操作的第一到第十一示例性实施例中,可通过对将被改变的第一预测编码单元的像素及其邻近像素执行多个后处理模式之一来改变第一预测编码单元中的每个像素的值,从而产生第二预测编码单元。然后,作为第二预测编码单元和当前编码单元之间的差的残差块被变换、量化和熵编码以产生比特流。关于用于产生第二预测编码单元的后处理模式的信息可被添加到比特流的预定区域,从而解码设备可再现当前编码单元的第二预测编码单元。

[0323] 如果在操作2320确定当前编码单元具有位于当前画面的边界之外的部分,则在操作2340,不产生第二预测编码单元,第一预测编码单元被直接输出作为关于当前编码单元的预测信息。然后,作为第一预测编码单元和当前编码单元之间的差的残差块被变换、量化和熵编码以产生比特流。

[0324] 图26是示出根据示例性实施例的对视频解码的方法的流程图。参照图 26,在操作2610,从接收的比特流提取关于将被解码的当前解码单元的预测模式的信息。

[0325] 在操作2620,根据提取的信息产生当前解码单元的第一预测解码单元。

[0326] 在操作2630,确定当前解码单元是否具有位于当前画面的边界之外的部分。可按照这样的方式根据确定结果产生预定索引MPI_PredMode:当预定索引MPI_PredMode是0时将不执行产生第二预测编码单元的后处理,当预定索引MPI_PredMode是1时将执行产生第二预测编码单元的后处理。

[0327] 如果在操作2630确定当前解码单元不具有位于当前画面的边界之外的部分,则在操作2640,通过使用第一预测解码单元的每个像素和每个像素的邻近像素来改变第一预测解码单元的每个像素的值,从而产生第二预测解码单元。如上所述,在关于后处理器1420的操作的第一到第十一示例性实施例中,可通过对将被改变的第一预测解码单元的像素及其邻近像素执行多个后处理模式之一来改变第一预测解码单元中的每个像素的值,从而产生第二预测解码单元。

[0328] 如果在操作2630确定当前解码单元具有位于当前画面的边界之外的部分,则在操作2650,不执行产生第二预测解码单元的后处理,并将第一预测解码单元直接输出为关于当前解码单元的预测信息。第一预测解码单元与从比特流提取的当前解码单元的残差块组合以产生当前解码单元。

[0329] 示例性实施例还可被实施为计算机可读记录介质上的计算机可读代码。计算机可读记录介质是能够存储其后由计算机系统读取的数据的任何数据存储装置。计算机可读记

录介质的示例包括只读存储器 (ROM)、随机存取存储器 (RAM)、CD-ROM、磁带、软盘和光数据存储装置。计算机可读记录介质还可分布于联网的计算机系统,从而以分布的方式存储和执行计算机可读代码。

[0330] 示例性实施例还可被实现为计算机处理器和硬件装置。

[0331] 虽然上面已经具体显示和描述了示例性实施例,但本领域的普通技术人员应理解,在不脱离由权利要求限定的本发明构思的精神和范围的情况下,可对其在形式和细节上进行各种修改。示例性实施例应被认为仅是示意性而不是出于限制性目的。因此,本发明构思的范围不是由示例性实施例的具体描述限定,而是由权利要求限定,在所述范围内的所有差异应被理解为包括在本发明构思之中。

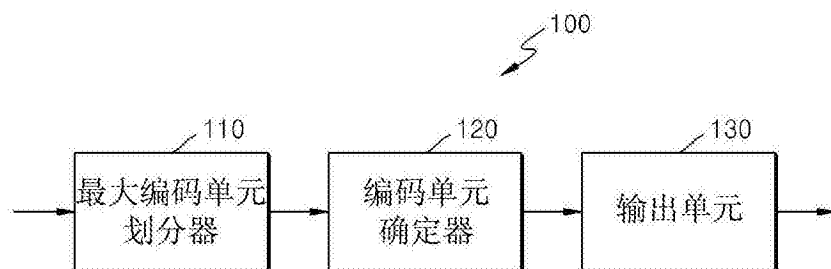


图1

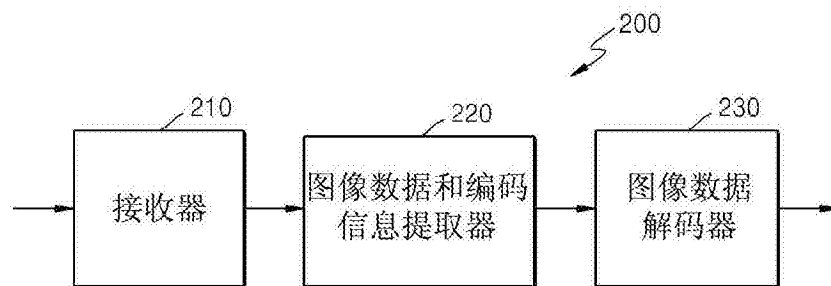


图2

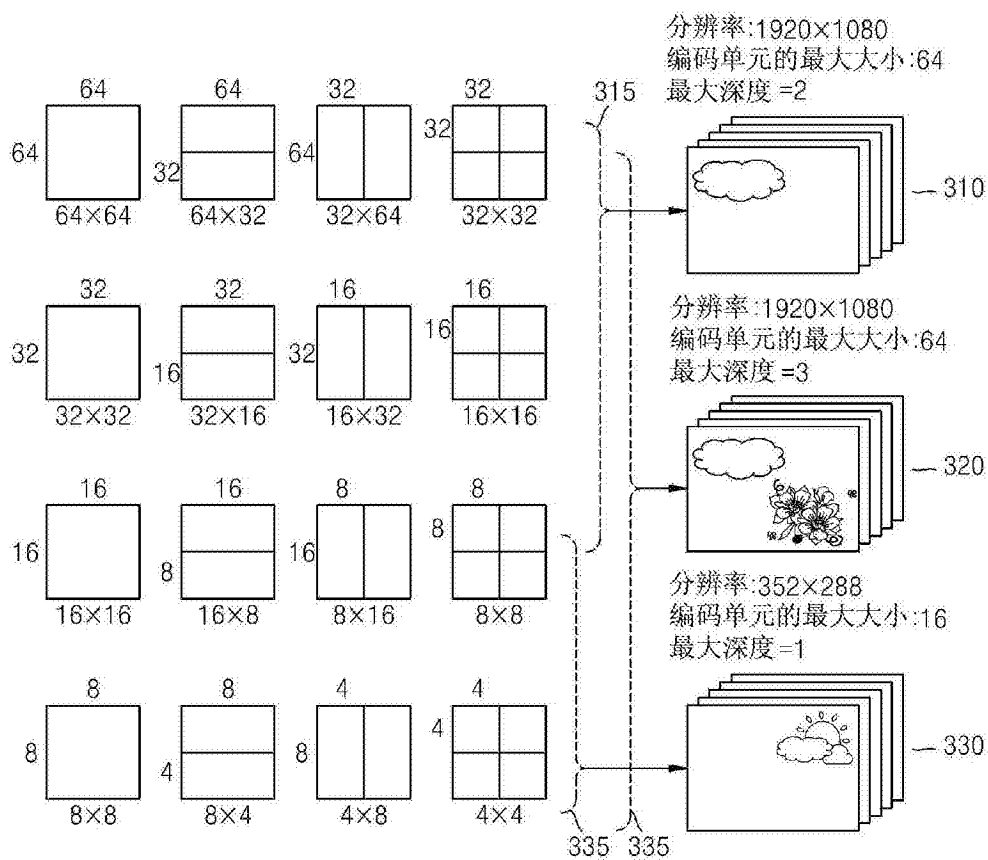


图3

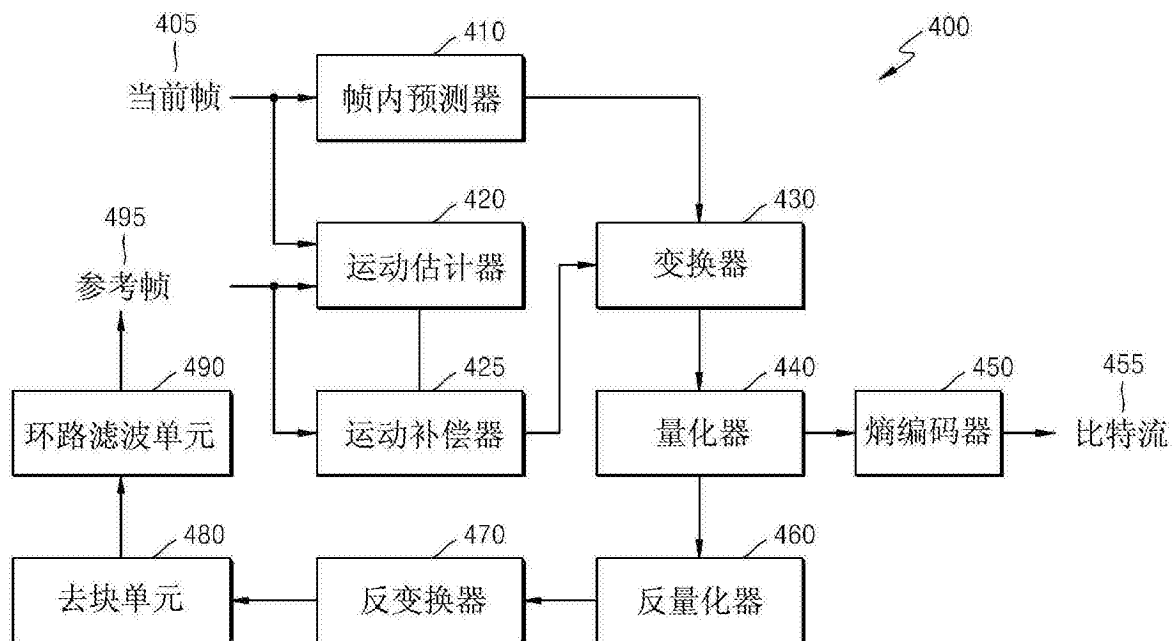


图4

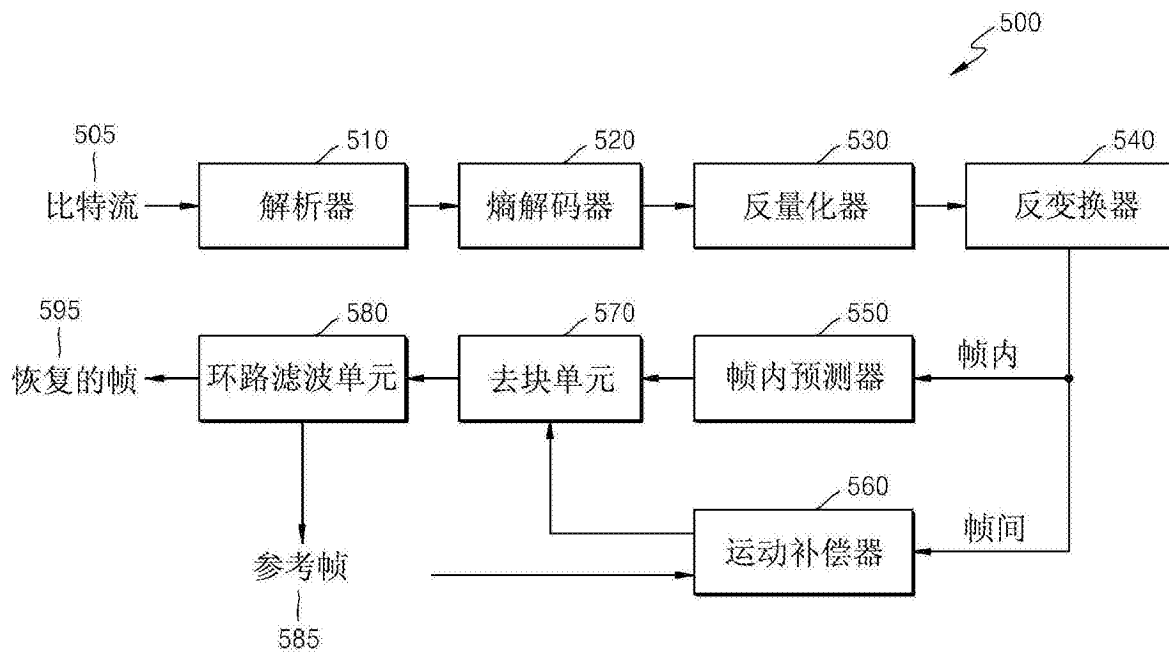


图5

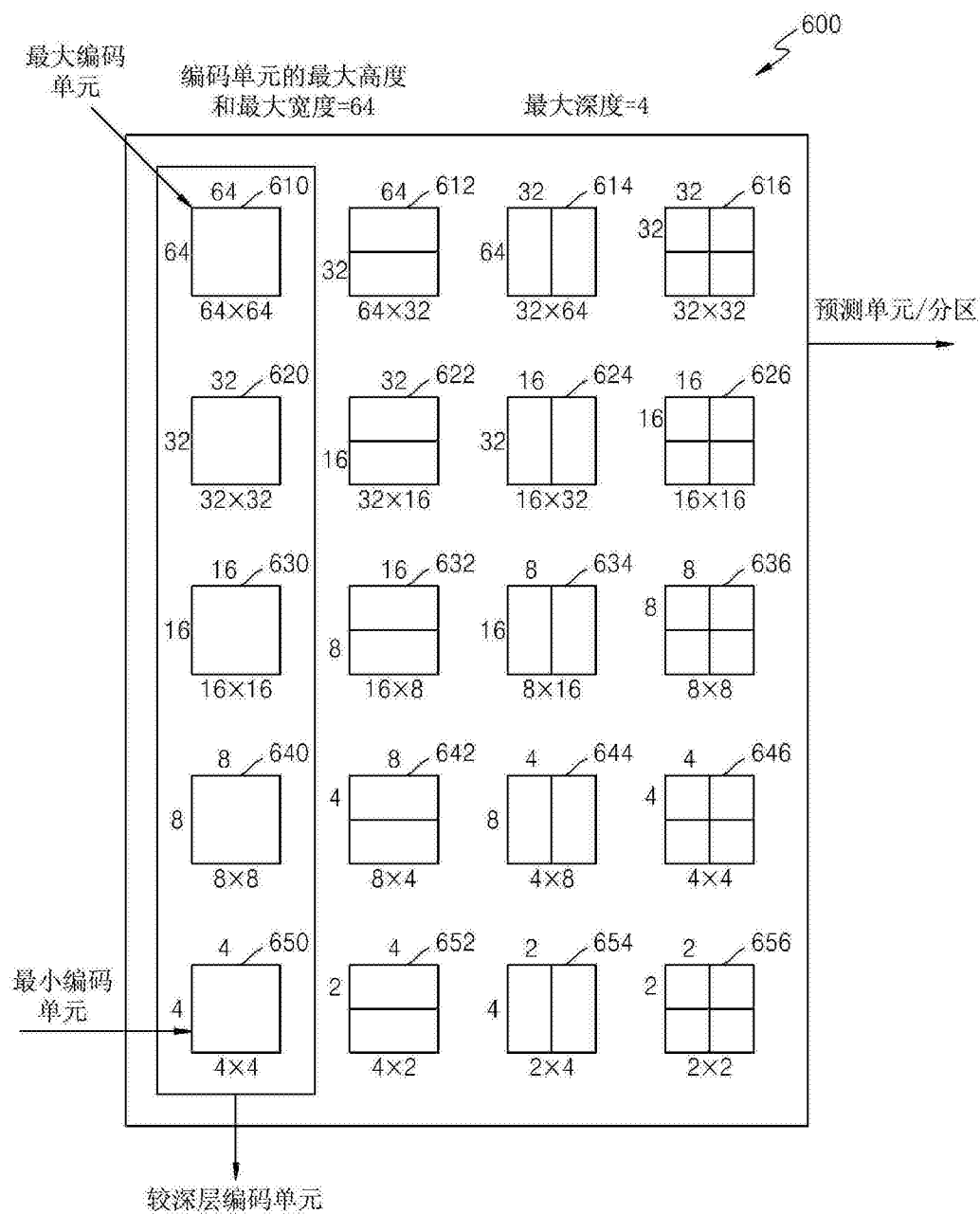


图6

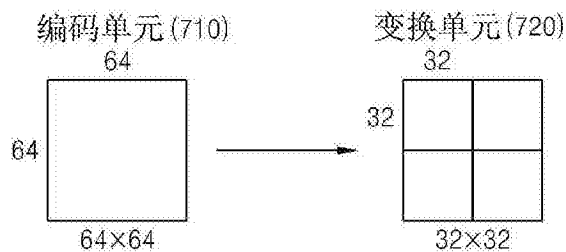


图7

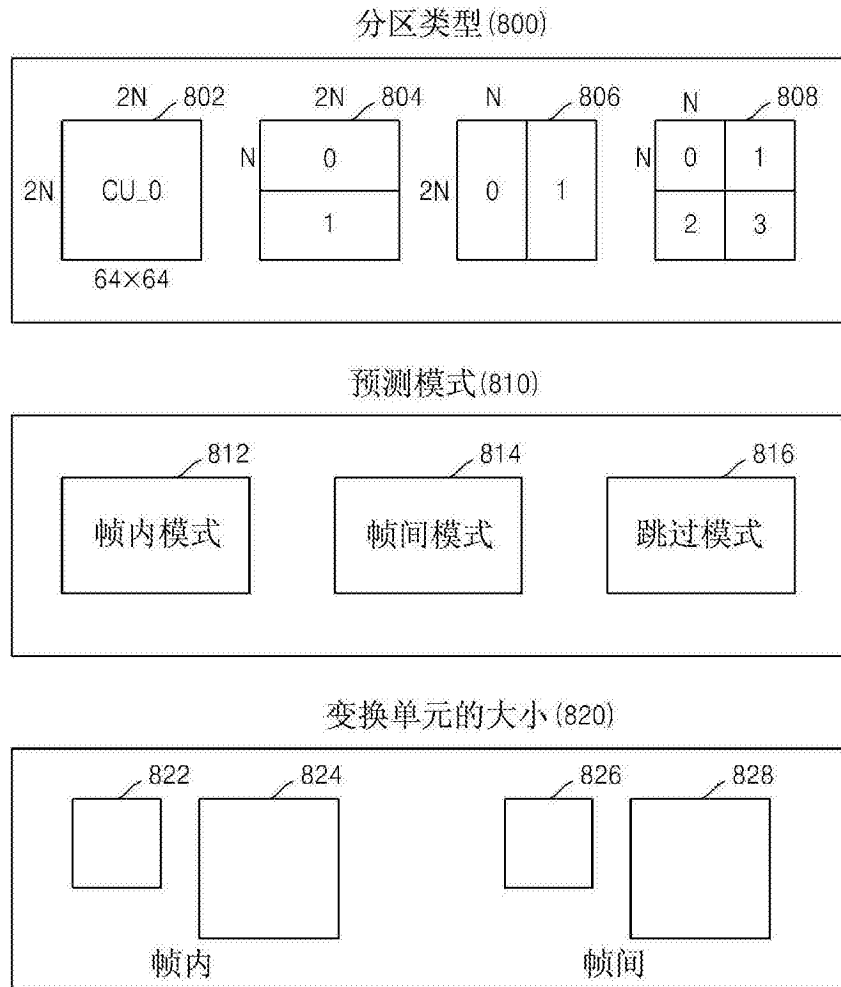


图8

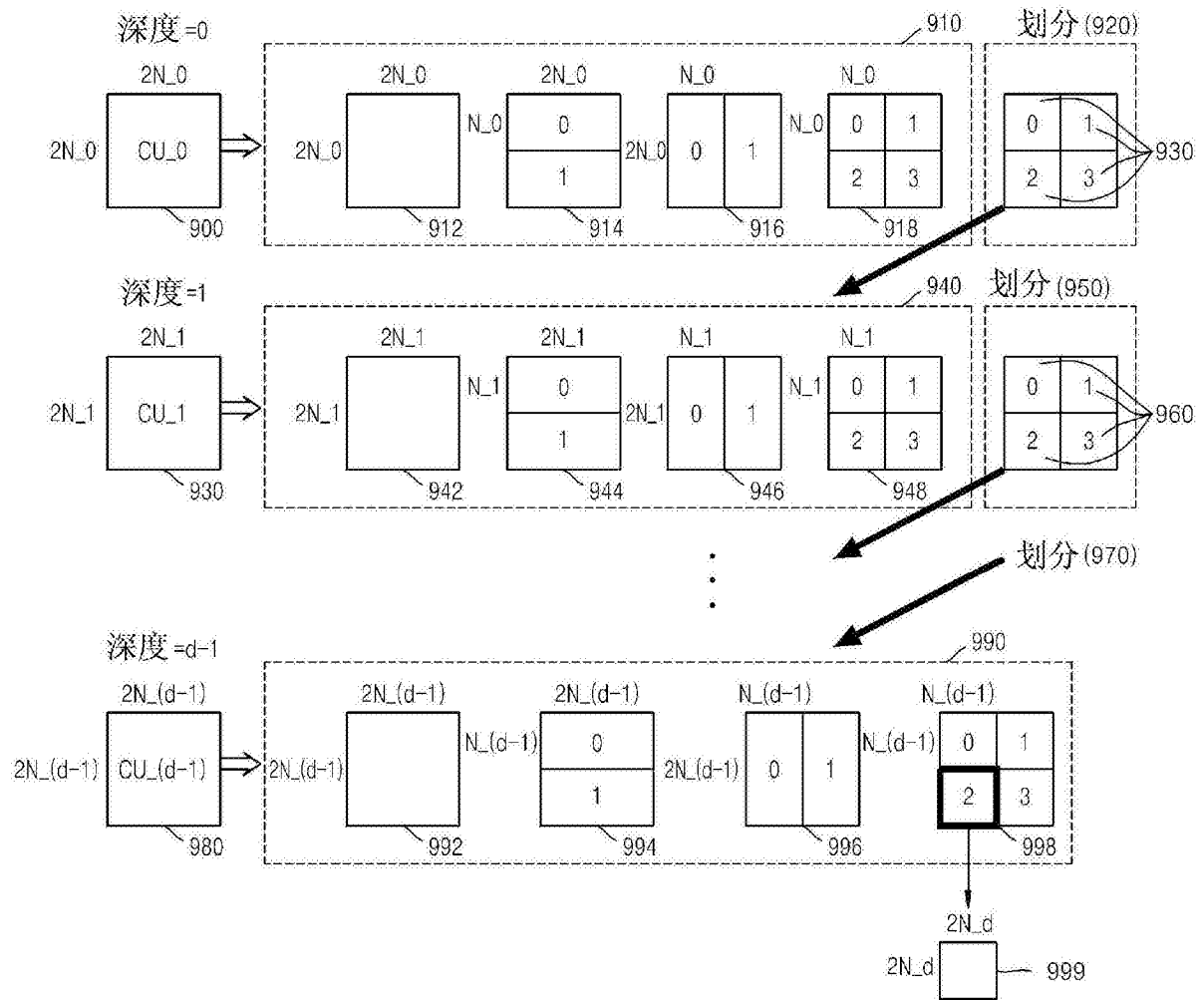
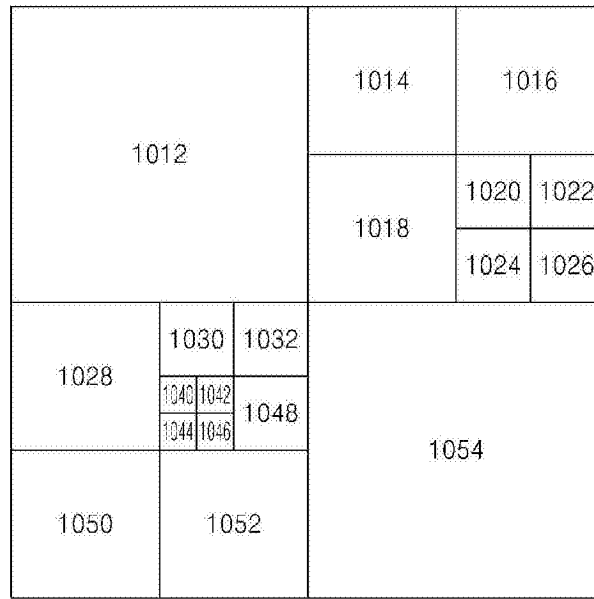
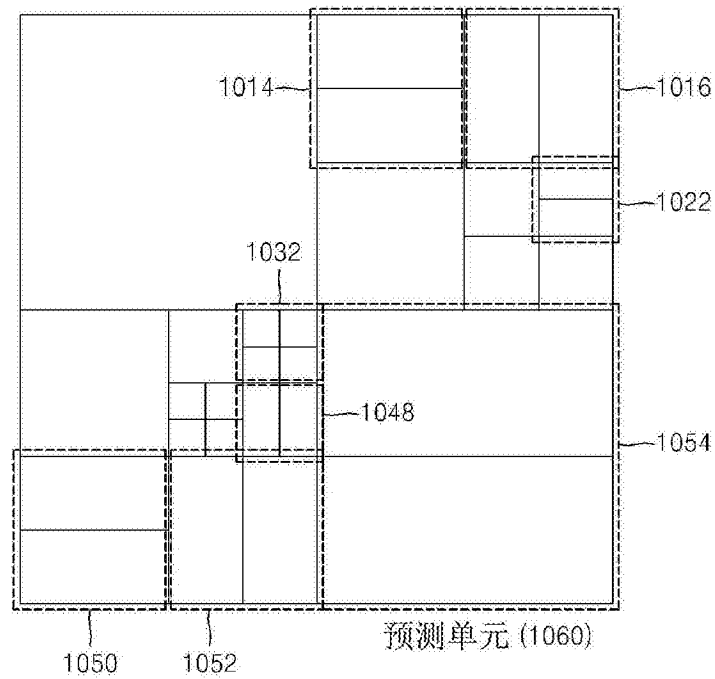


图9



编码单元 (1010)

图10



预测单元 (1060)

图11

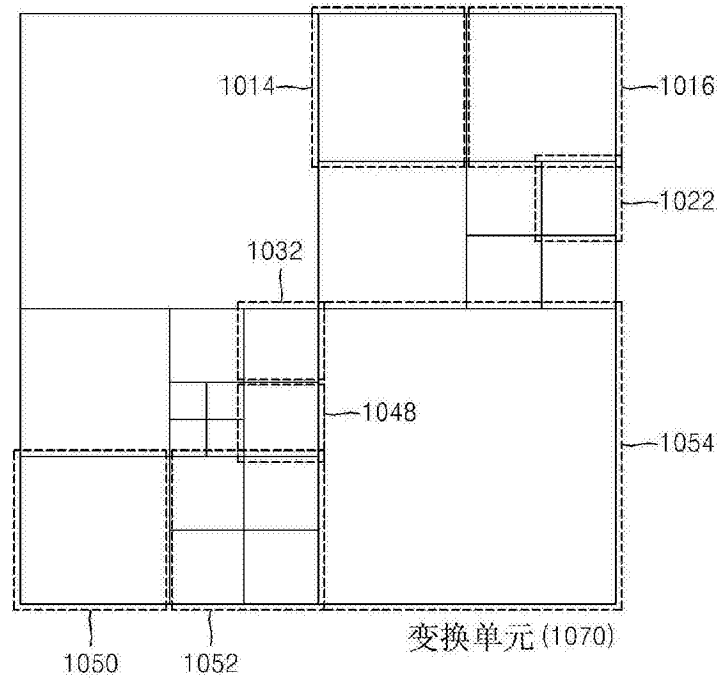


图12

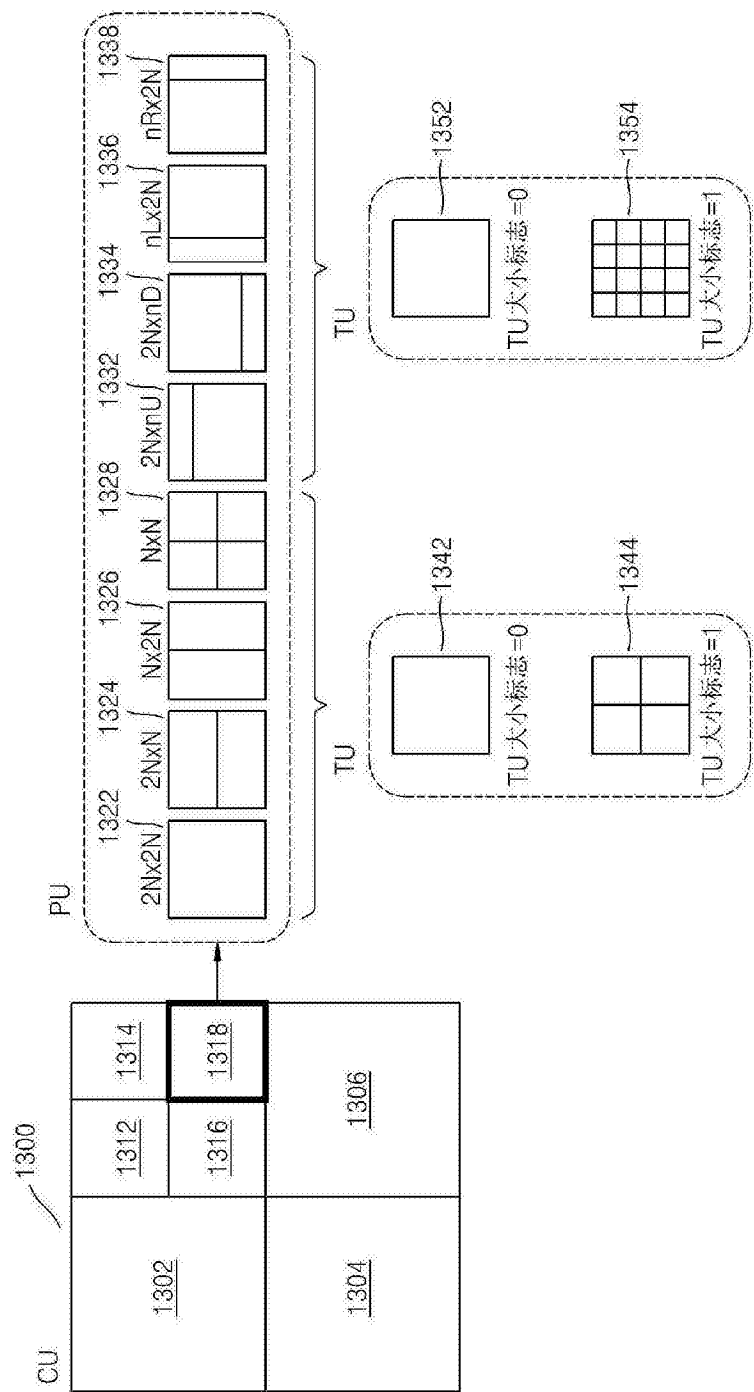


图13

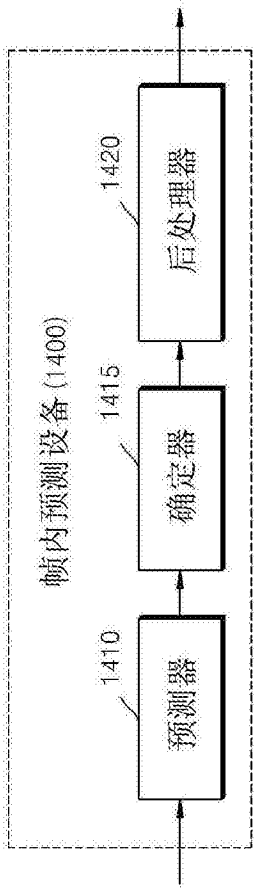


图14

编码单元 的大小	预测模式的数量		
	示例 1	示例 2	示例 3
2	—	5	5
4	9	9	9
8	9	9	9
16	33	17	11
32	33	33	33
64	5	5	9
128	5	5	5

图15

预测模式	名称
0	垂直
1	水平
2	DC
3	下_左
4	下_右
5	垂直_右
6	水平_下
7	垂直_左
8	水平_上

图16A

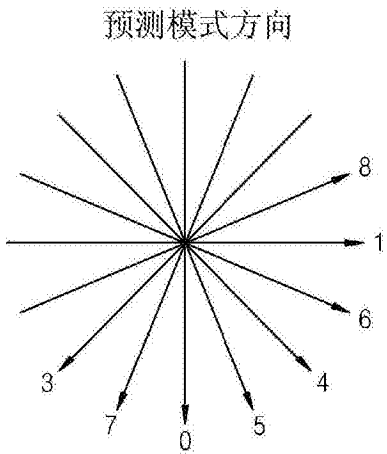


图16B

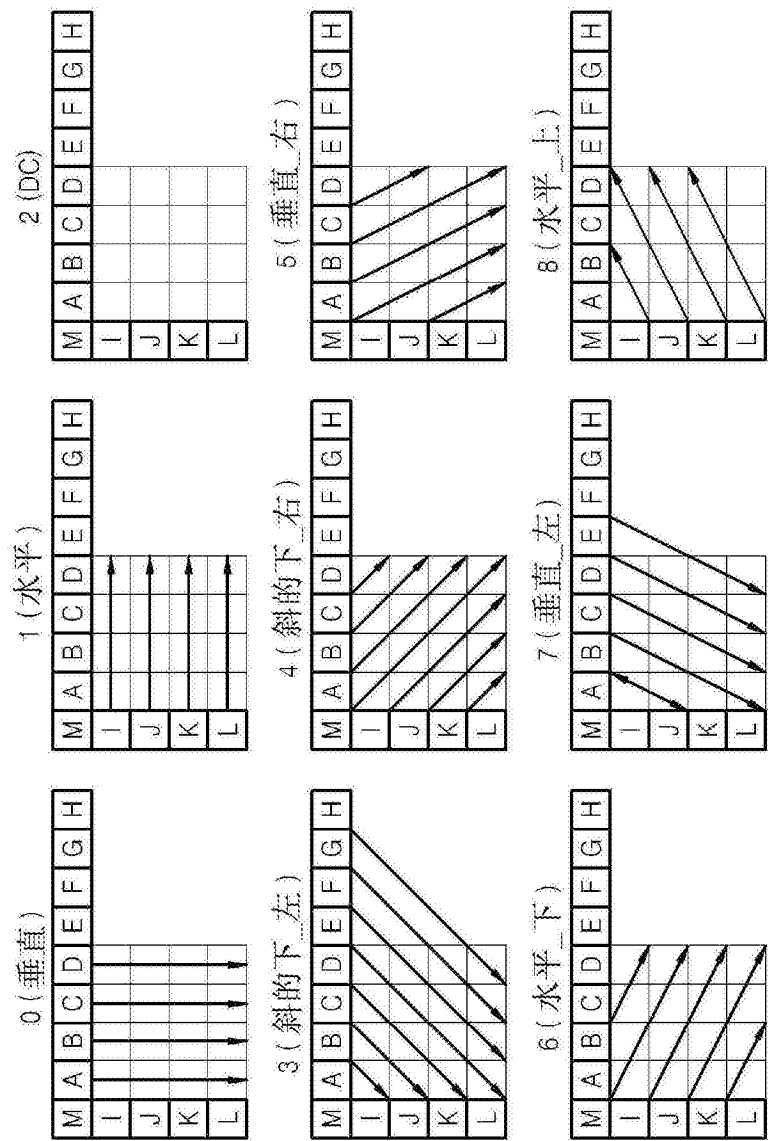


图16C

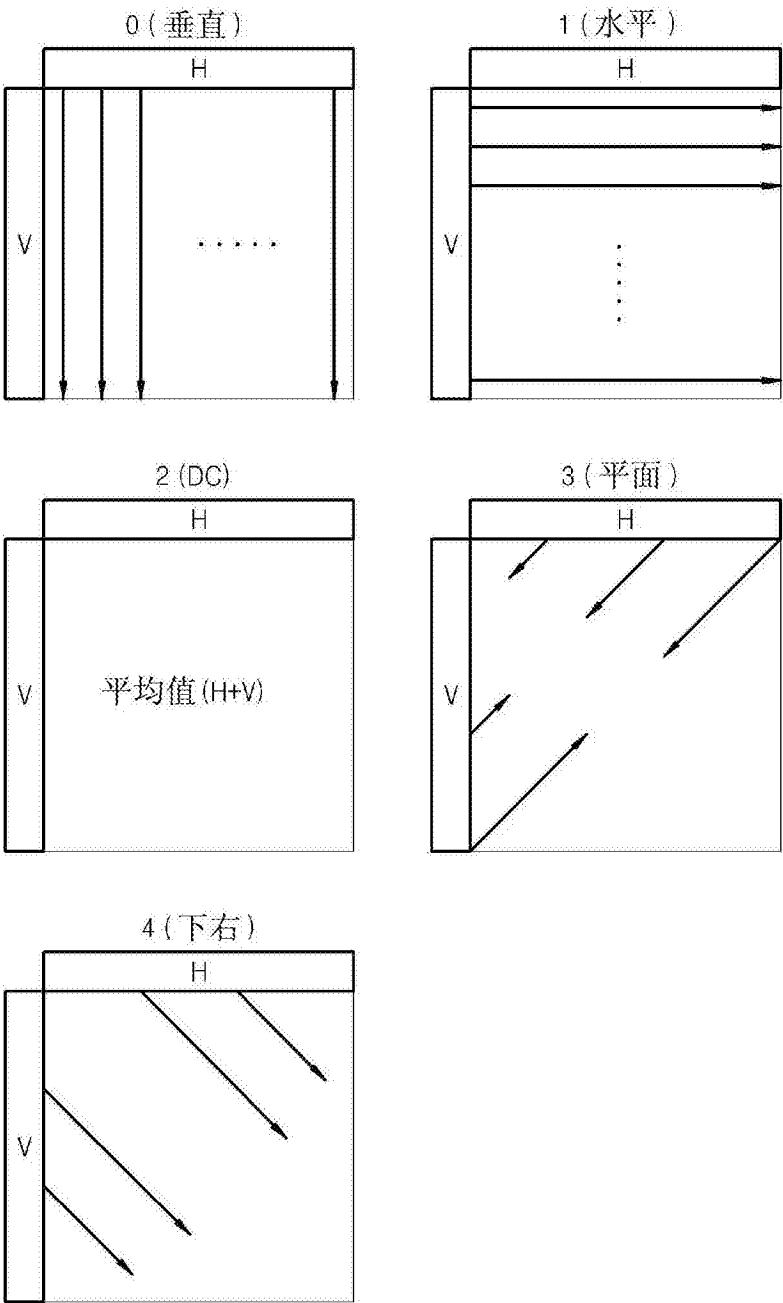


图17

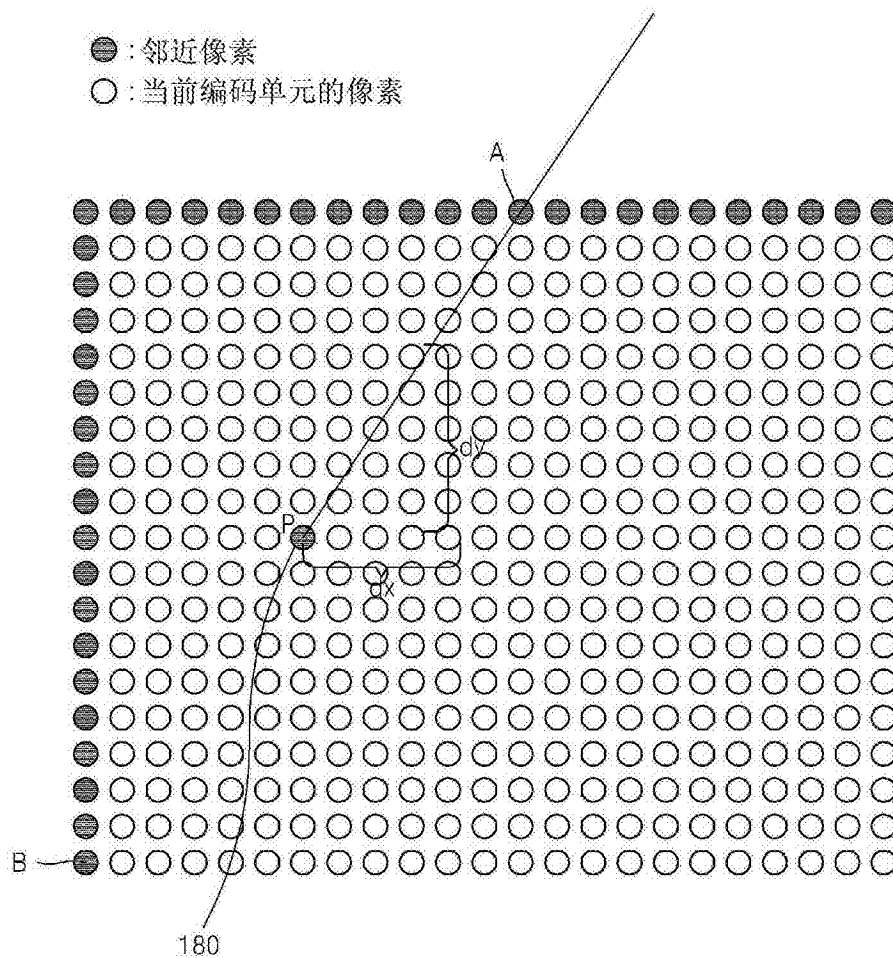


图18A

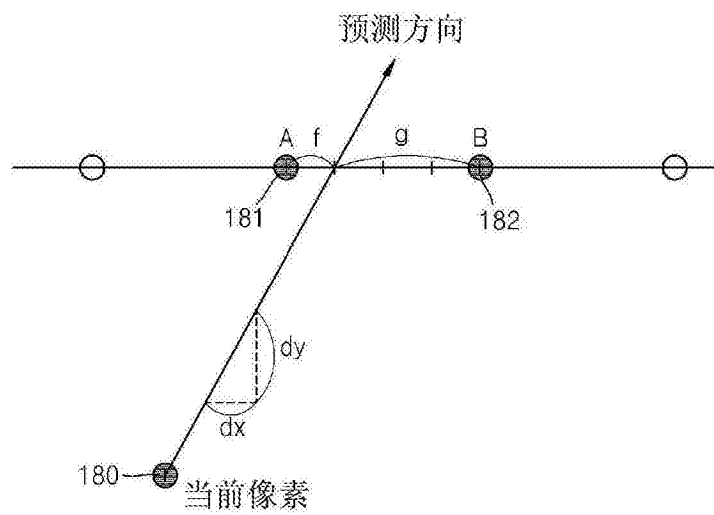


图18B

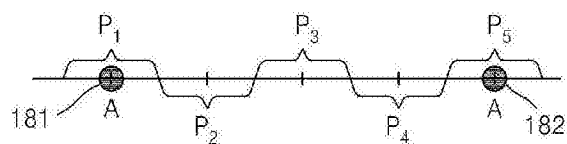
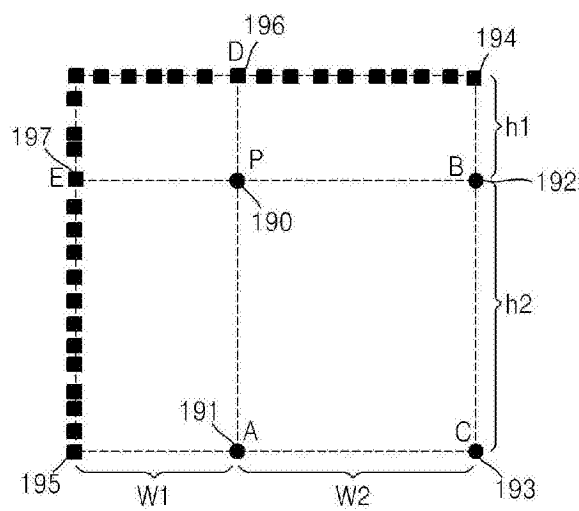


图18C



■ : 邻近像素

图19

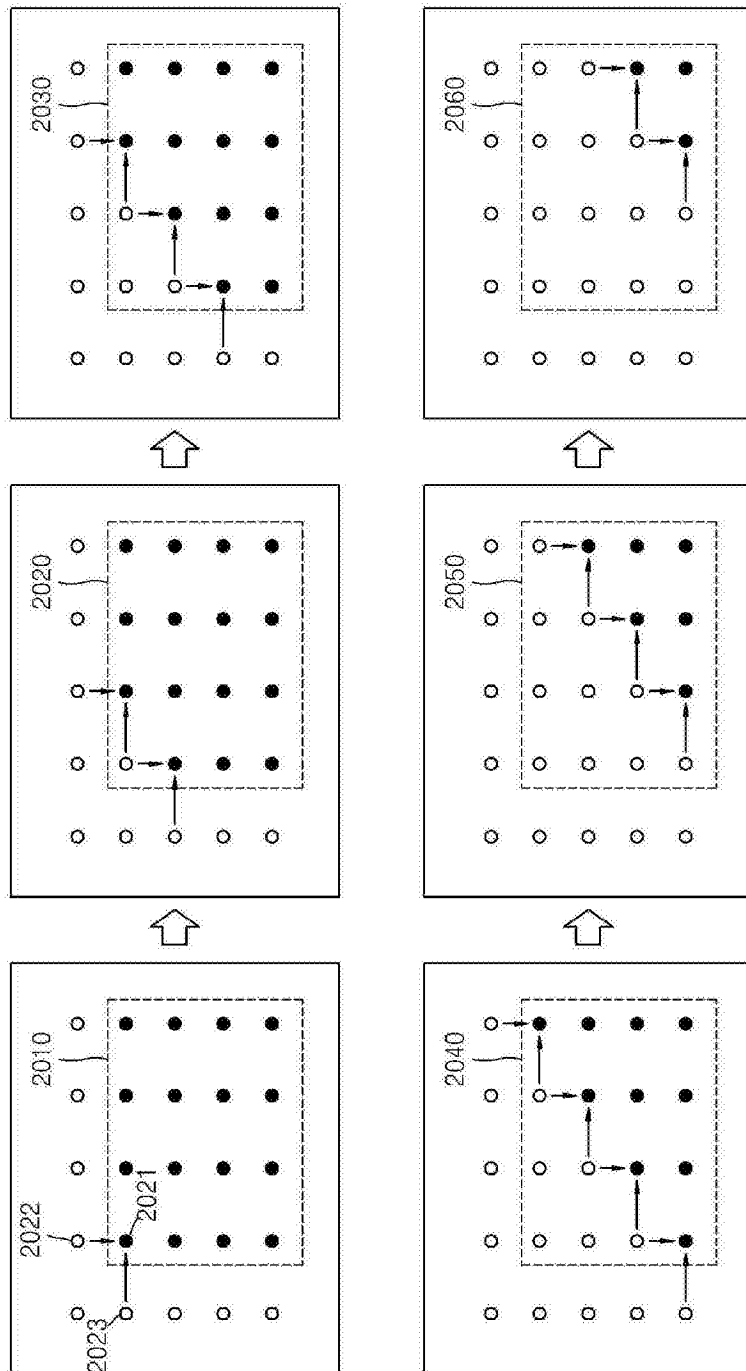


图20

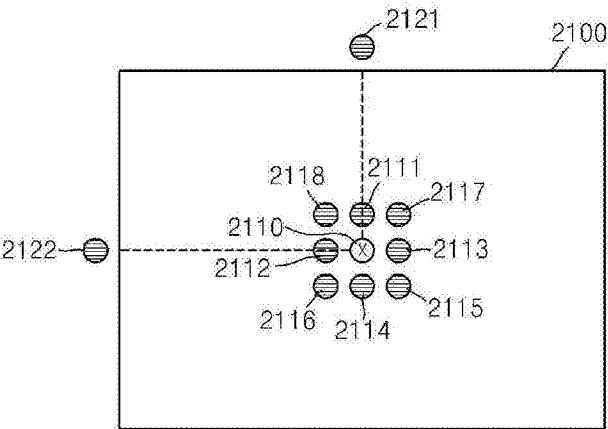


图21

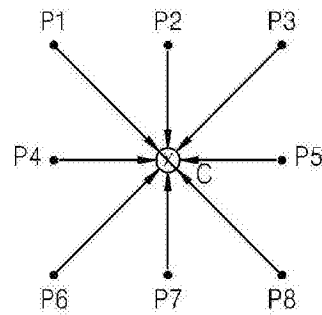


图22

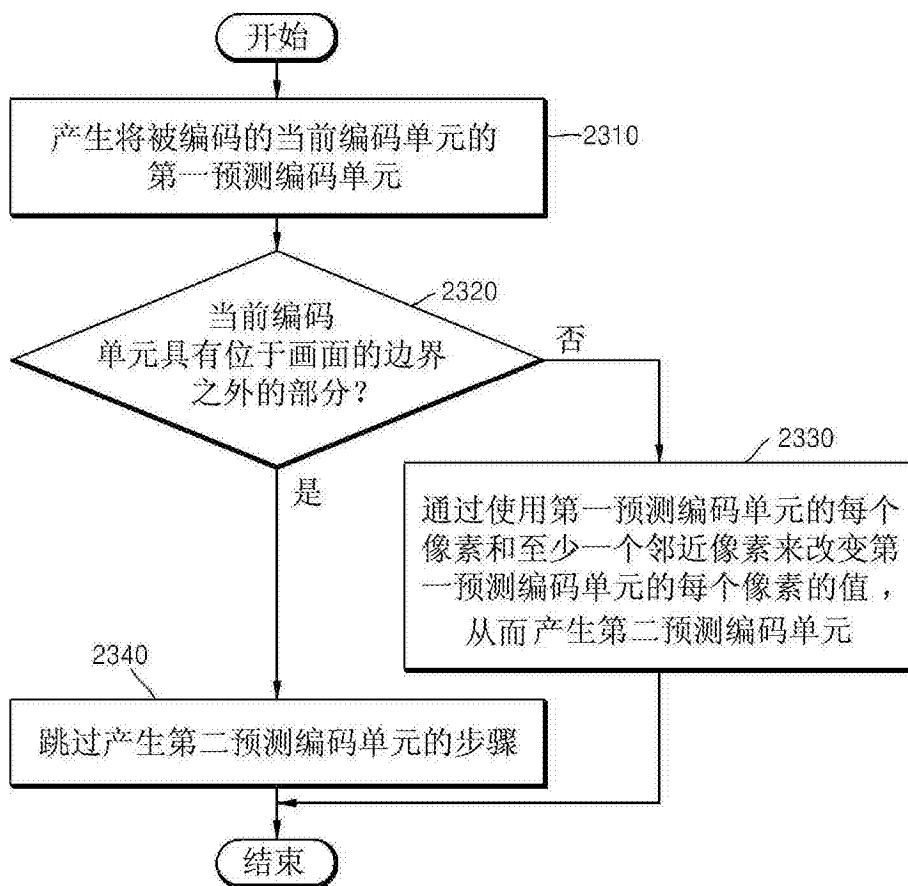


图23

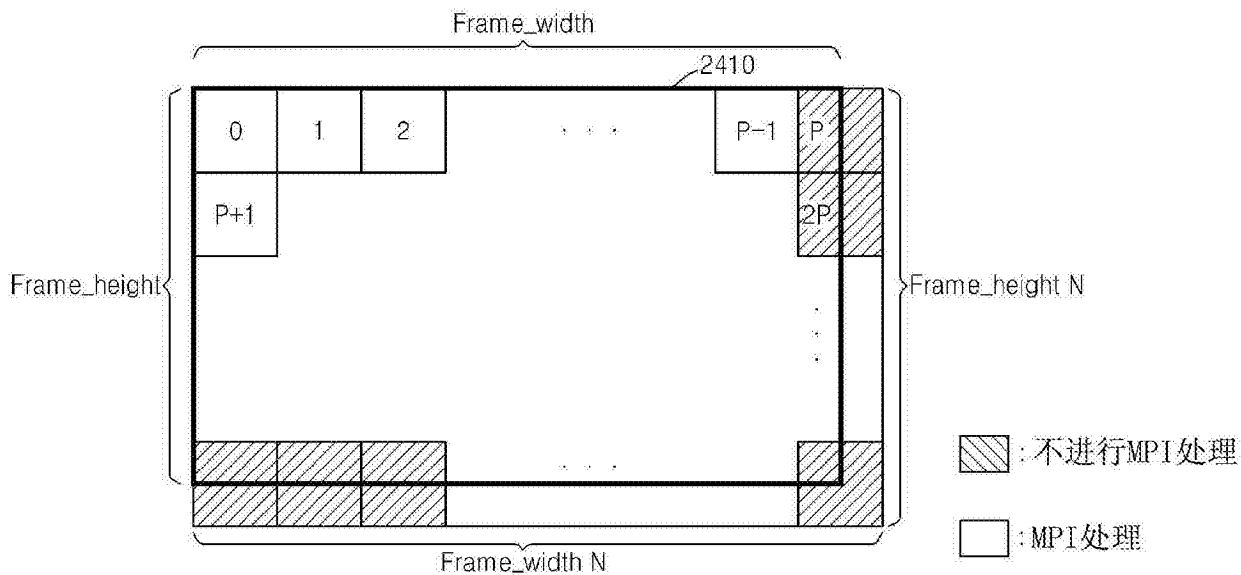


图24

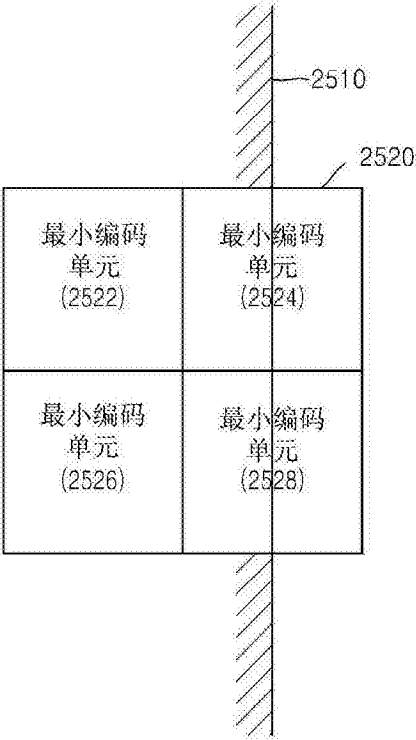


图25

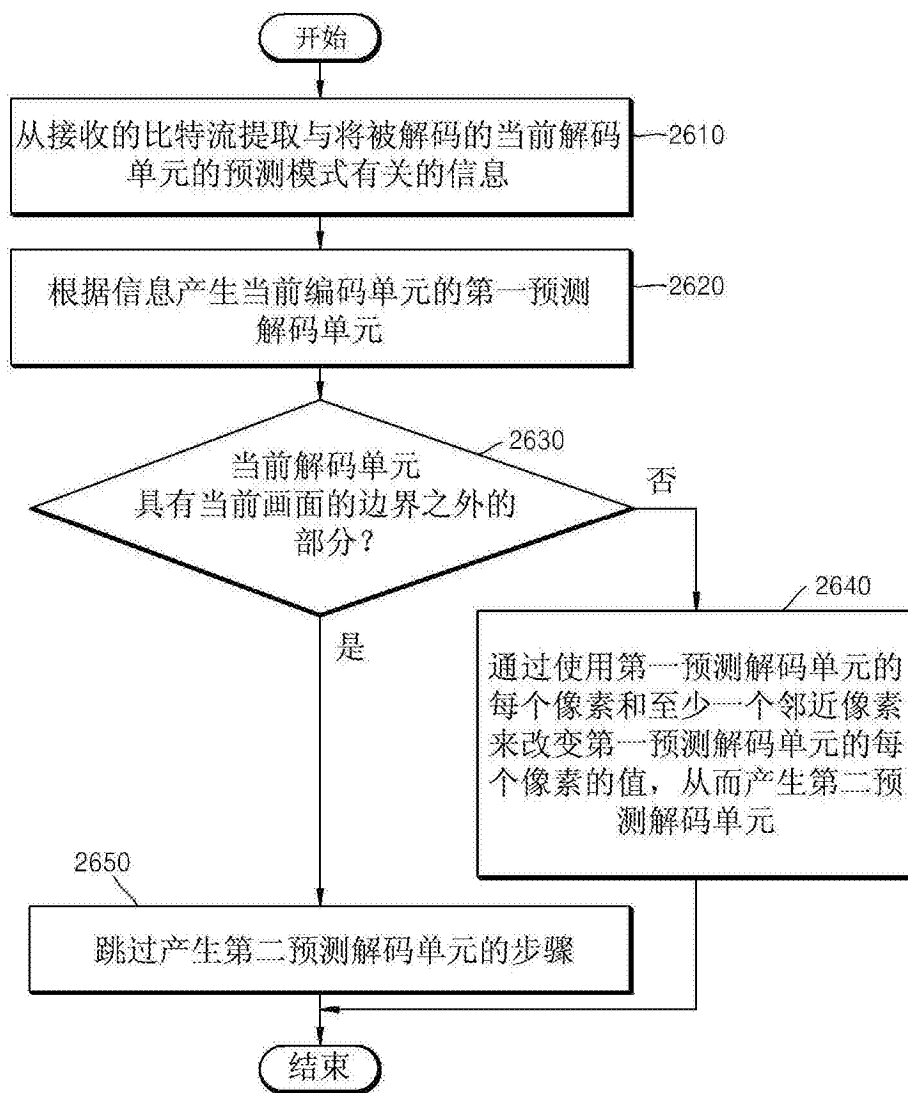


图26

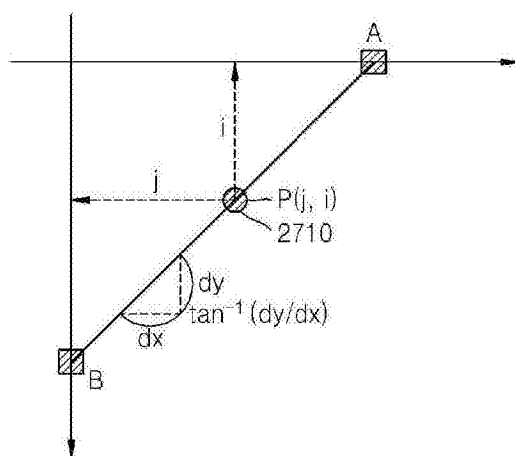


图27

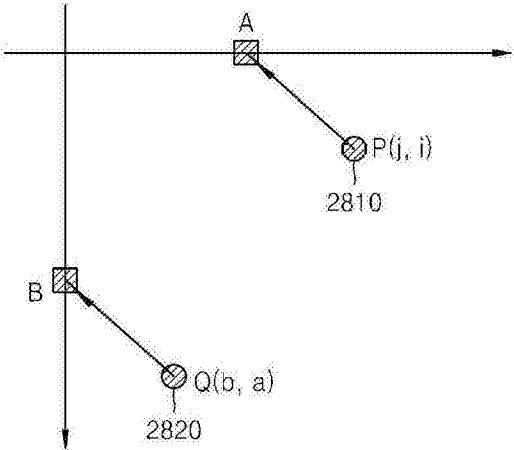


图28

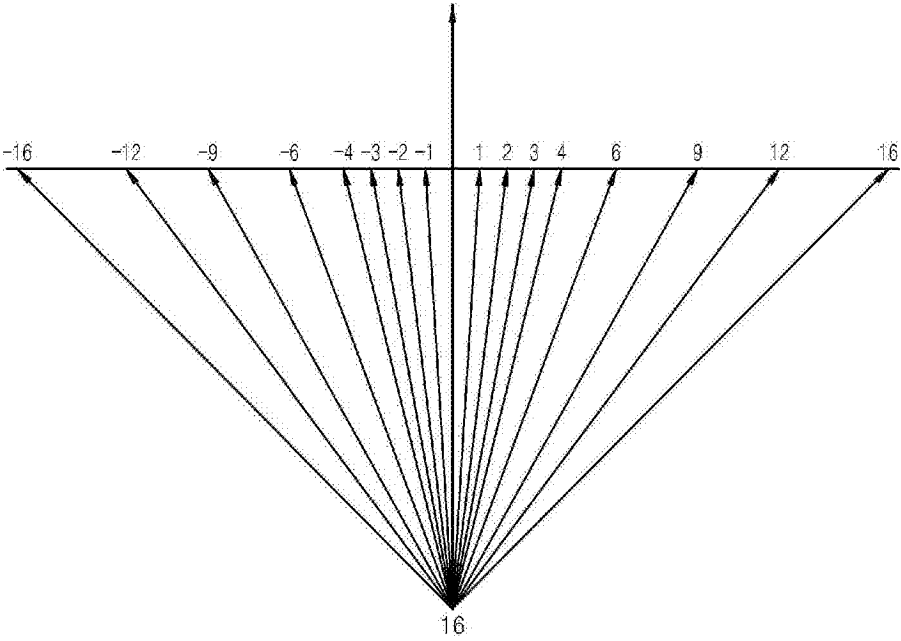


图29

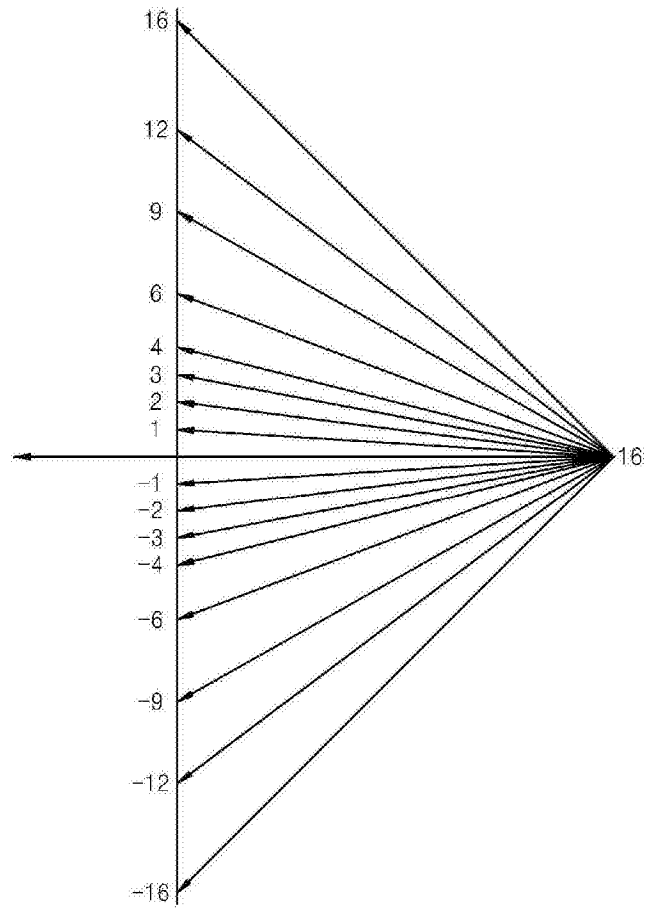


图30