



(21) 申请号 202310922346.6

(22) 申请日 2019.09.20

(30) 优先权数据

62/734,233 2018.09.20 US

62/734,266 2018.09.21 US

(62) 分案原申请数据

201980005874.7 2019.09.20

(71) 申请人 LG电子株式会社

地址 韩国首尔

(72) 发明人 崔情娥 金昇焕 柳先美 崔璋元

许镇

(74) 专利代理机构 北京三友知识产权代理有限公司

公司 11127

专利代理人 刘久亮 黄纶伟

(51) Int.Cl.

H04N 19/513 (2014.01)

H04N 19/124 (2014.01)

H04N 19/129 (2014.01)

H04N 19/13 (2014.01)

H04N 19/132 (2014.01)

H04N 19/176 (2014.01)

H04N 19/18 (2014.01)

H04N 19/46 (2014.01)

H04N 19/70 (2014.01)

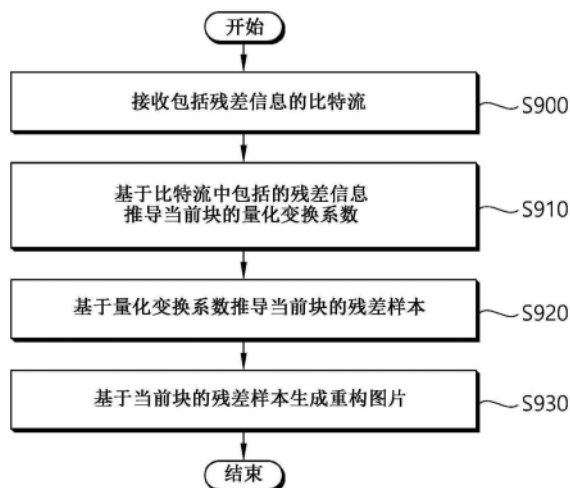
权利要求书3页 说明书36页 附图10页

(54) 发明名称

图像解码设备、图像编码设备和比特流发送设备

(57) 摘要

本申请涉及图像解码设备、图像编码设备和比特流发送设备。根据本公开的用于由解码装置对图像进行解码的方法包括以下步骤：接收包括残差信息的比特流，基于比特流中包括的残差信息来推导当前块的量化变换因子，基于量化变换因子推导当前块的残差样本，并基于当前块的残差样本生成重构图片。



1. 一种图像解码设备,该图像解码设备包括:
存储器;以及
连接到所述存储器的至少一个处理器,所述至少一个处理器被配置为:
接收包括残差信息的比特流;
基于所述比特流中包含的所述残差信息,推导当前块的量化变换系数;
基于所述量化变换系数,推导所述当前块的残差样本;以及
基于所述当前块的所述残差样本生成重构图片,
其中,所述残差信息包括用于所述量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于所述变换系数级别是否大于第一阈值的第一变换系数级别标志,
其中,所述量化变换系数是通过以下步骤推导的:
对所述第一变换系数级别标志进行解码,并对所述奇偶校验级别标志进行解码;以及
基于经解码的所述奇偶校验级别标志的值和经解码的所述第一变换系数级别标志的值,推导所述量化变换系数,并且
其中,在对所述奇偶校验级别标志进行解码之前,执行对所述第一变换系数级别标志的解码,并且
所述奇偶校验级别标志是基于所述第一变换系数级别标志等于1被解码的。
2. 根据权利要求1所述的图像解码设备,其中,所述残差信息还包括:表示所述量化变换系数是否为非零有效系数的有效系数标志;以及关于所述量化变换系数的所述变换系数级别是否大于第二阈值的第二变换系数级别标志。
3. 根据权利要求2所述的图像解码设备,其中,所述残差信息中所包含的所述当前块内的所述量化变换系数的有效系数标志的数量、所述第一变换系数级别标志的数量、所述奇偶校验级别标志的数量和所述第二变换系数级别标志的数量之和为预定阈值或更小。
4. 根据权利要求3所述的图像解码设备,其中,基于所述当前块的尺寸来确定所述预定阈值。
5. 根据权利要求3所述的图像解码设备,其中,所述残差信息中所包含的所述有效系数标志的数量、所述第一变换系数级别标志的数量和所述奇偶校验级别标志的数量之和为第三阈值或更小,
其中,所述残差信息中所包含的所述第二变换系数级别标志的数量为第四阈值或更小,并且
其中,所述预定阈值是所述第三阈值和所述第四阈值之和。
6. 根据权利要求3所述的图像解码设备,其中,当基于由系数扫描顺序确定的第0量化变换系数到第n量化变换系数而推导的所述有效系数标志的数量、所述第一变换系数级别标志的数量、所述奇偶校验级别标志的数量和所述第二变换系数级别标志的数量之和达到所述预定阈值时,针对由所述系数扫描顺序确定的第(n+1)量化变换系数,省略所述有效系数标志、所述第一变换系数级别标志、所述奇偶校验级别标志和所述第二变换系数级别标志的显式信令,并且基于所述残差信息中包含的系数级别信息的值来推导所述第(n+1)量化变换系数的值。
7. 根据权利要求6所述的图像解码设备,其中,所述残差信息中所包含的所述有效系数标志、所述第一变换系数级别标志、所述奇偶校验级别标志和所述第二变换系数级别标志

是基于上下文被编码的,并且

其中,所述系数级别信息是以旁路为基础被编码的。

8. 一种图像编码设备,该图像编码设备包括:

存储器;以及

连接到所述存储器的至少一个处理器,所述至少一个处理器被配置为:

推导当前块的残差样本;

基于所述当前块的所述残差样本推导量化变换系数;以及

对包含用于所述量化变换系数的信息的残差信息进行编码,

其中,所述残差信息包括用于所述量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于所述变换系数级别是否大于第一阈值的第一变换系数级别标志,

其中,所述残差信息是通过以下步骤编码的:

基于所述量化变换系数来推导所述奇偶校验级别标志的值和所述第一变换系数级别标志的值;以及

对所述第一变换系数级别标志进行编码并且对所述奇偶校验级别标志进行编码,并且

其中,在对所述奇偶校验级别标志进行编码之前,执行对所述第一变换系数级别标志的编码,并且

所述奇偶校验级别标志是基于所述第一变换系数级别标志等于1被编码的。

9. 根据权利要求8所述的图像编码设备,其中,所述残差信息还包括:表示所述量化变换系数是否为非零有效系数的有效系数标志;以及关于所述量化变换系数的所述变换系数级别是否大于第二阈值的第二变换系数级别标志。

10. 根据权利要求9所述的图像编码设备,其中,所述残差信息中所包含的所述当前块内的所述量化变换系数的有效系数标志的数量、所述第一变换系数级别标志的数量、所述奇偶校验级别标志的数量和所述第二变换系数级别标志的数量之和为预定阈值或更小。

11. 根据权利要求10所述的图像编码设备,其中,基于所述当前块的尺寸来确定所述预定阈值。

12. 根据权利要求10所述的图像编码设备,其中,所述残差信息中所包含的所述有效系数标志的数量、所述第一变换系数级别标志的数量和所述奇偶校验级别标志的数量之和为第三阈值或更小,

其中,所述残差信息中所包含的所述第二变换系数级别标志的数量为第四阈值或更小,并且

其中,所述预定阈值是所述第三阈值和所述第四阈值之和。

13. 根据权利要求10所述的图像编码设备,其中,当基于由系数扫描顺序确定的第0量化变换系数到第n量化变换系数而推导的所述有效系数标志的数量、所述第一变换系数级别标志的数量、所述奇偶校验级别标志的数量和所述第二变换系数级别标志的数量之和达到所述预定阈值时,针对由所述系数扫描顺序确定的第(n+1)量化变换系数,省略所述有效系数标志、所述第一变换系数级别标志、所述奇偶校验级别标志和所述第二变换系数级别标志的显式信令,并且基于所述残差信息中包含的系数级别信息的值来推导所述第(n+1)量化变换系数的值。

14. 根据权利要求13所述的图像编码设备,其中,所述残差信息中所包含的所述有效系

数标志、所述第一变换系数级别标志、所述奇偶校验级别标志和所述第二变换系数级别标志是基于上下文被编码的,并且

其中,所述系数级别信息是以旁路为基础被编码的。

15. 一种用于发送比特流的设备,所述设备包括:

被配置为获得由图像编码设备生成的所述比特流的至少一个处理器;以及

被配置为发送所述比特流的发送器,

所述比特流是通过以下操作生成的:

推导当前块的残差样本;

基于所述当前块的所述残差样本推导量化变换系数;以及

对包含关于所述量化变换系数的信息的残差信息进行编码,

其中,所述残差信息包括用于所述量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于所述变换系数级别是否大于第一阈值的第一变换系数级别标志,

其中,所述残差信息是通过以下步骤编码的:

基于所述量化变换系数来推导所述奇偶校验级别标志的值和所述第一变换系数级别标志的值;以及

对所述第一变换系数级别标志进行编码并且对所述奇偶校验级别标志进行编码,

其中,在对所述奇偶校验级别标志进行编码之前,执行对所述第一变换系数级别标志的编码,并且

其中,所述奇偶校验级别标志是基于所述第一变换系数级别标志等于1被编码的。

图像解码设备、图像编码设备和比特流发送设备

[0001] 本申请是原案申请号为201980005874.7的发明专利申请(国际申请号:PCT/KR2019/012226,申请日:2019年9月20日,发明名称:变换因子级别编码方法及其装置)的分案申请。

技术领域

[0002] 本发明涉及图像编码技术,尤其涉及一种用于图像编码系统的变换系数级别的编码方法及其装置。

背景技术

[0003] 近来,在各个领域中对诸如4K或8K超高清(UHD)图像/视频之类的高分辨率、高质量图像/视频的需求正在增长。随着图像/视频分辨率或质量变得更高,比传统图像/视频数据传输相对更多量的信息或比特。因此,如果图像/视频数据经由诸如现有的有线/无线宽带线路之类的介质传输或存储在传统存储介质中,则传输和存储的成本容易增加。

[0004] 此外,对于虚拟现实(VR)和人工现实(AR)内容以及诸如全息图之类的沉浸式媒体的兴趣和需求正在增长;并且展现出与实际图像/视频不同的图像/视频特性的图像/视频(诸如游戏图像/视频)的广播也在增长。

[0005] 因此,需要高效的图像/视频压缩技术来有效地压缩和传输、存储或播放如上所述的表现出各种特性的高分辨率、高质量的图像/视频。

发明内容

[0006] 技术任务

[0007] 本公开的一个目的是提供一种用于增强图像编码效率的方法及装置。

[0008] 本公开的另一目的是提供一种用于增强残差编码的效率的方法及装置。

[0009] 本公开的又一目的是提供一种用于增强变换系数级别的编码效率的方法及装置。

[0010] 本公开的再一目的是提供一种用于通过基于莱斯(Rice)参数对残差信息执行二值化处理来增强残差编码效率的方法及装置。

[0011] 本公开的再一目的是提供一种用于通过确定(或改变)用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志的解码顺序来增强编码效率的方法及装置。

[0012] 本公开的进一步目的是提供一种用于通过将残差信息中所包括的当前块内的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和限制为预定阈值或更小来增强编码效率的方法及装置。

[0013] 本公开的另一进一步目的是提供一种用于通过将残差信息中包括的当前块内的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和限制为预定阈值或更小来减少基于上下文编

码的数据的方法及装置。

[0014] 解决方案

[0015] 本公开的实施方式提供了一种由解码设备执行的用于对图像进行解码的方法。该方法包括：接收包括残差信息的比特流，基于比特流中包括的残差信息来推导当前块的量化变换系数，基于量化变换系数推导当前块的残差样本，并基于当前块的残差样本生成重构图片；并且残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志，量化变换系数的推导包括对第一变换系数级别标志进行解码并对奇偶校验级别标志进行解码，以及基于经解码的奇偶校验级别标志的值和经解码的第一变换系数级别标志的值推导量化变换系数，并且在奇偶校验级别标志进行解码之前执行对第一变换系数级别标志的解码。

[0016] 本公开的另一实施方式提供了一种用于执行图像解码的解码设备。解码设备包括：熵解码器，其接收包括残差信息的比特流，并且基于比特流中包括的残差信息来推导当前块的量化变换系数；逆变换器，其基于量化变换系数推导当前块的残差样本；以及加法器，其基于当前块的残差样本生成重构图片，残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志，由熵解码器执行的推导量化变换系数包括：对第一变换系数级别标志进行解码，并对奇偶校验级别标志进行解码；以及基于经解码的奇偶校验级别标志的值和经解码的第一变换系数级别标志的值推导量化变换系数，并且在奇偶校验级别标志进行解码之前，执行对第一变换系数级别标志的解码。

[0017] 本公开的又一个实施方式提供了一种由编码设备执行的用于对图像进行编码的方法。该方法包括：推导当前块的残差样本；基于当前块的残差样本推导量化变换系数；以及对包括关于量化变换系数的信息的残差信息进行编码，并且残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志，残差信息的编码包括：基于量化变换系数来推导奇偶校验级别标志的值和第一变换系数级别标志的值，以及对第一变换系数级别标志进行编码并且对奇偶校验级别标志进行编码，并且在奇偶校验级别标志进行编码之前执行对第一变换系数级别标志的编码。

[0018] 本公开的又一实施方式提供了一种用于执行图像编码的编码设备。编码设备包括：减法器，其推导当前块的残差样本；量化器，其基于当前块的残差样本，推导量化变换系数；以及熵编码器，其对包括关于量化变换系数的信息的残差信息进行编码，并且残差信息包括：用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志，由熵编码器执行的残差信息的编码包括：基于量化变换系数来推导奇偶校验级别标志的值和第一变换系数级别标志的值；以及对第一变换系数级别标志进行编码并且对奇偶校验级别标志进行编码，并且在奇偶校验级别标志进行编码之前执行对第一变换系数级别标志的编码。

[0019] 本公开的再一实施方式提供了一种解码器可读存储介质，其存储关于使视频解码设备执行根据一些实施方式的解码方法的指令的信息。

[0020] 本公开的进一步实施方式提供了一种解码器可读存储介质，其存储关于使视频解码设备执行根据一些实施方式的解码方法的指令的信息。根据实施方式的解码方法包括：

接收包括残差信息的比特流,基于比特流中包括的残差信息来推导当前块的量化变换系数,基于量化变换系数推导当前块的残差样本,并基于当前块的残差样本生成重构图片;并且残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志,量化变换系数的推导包括对第一变换系数级别标志进行解码并对奇偶校验级别标志进行解码,以及基于经解码的奇偶校验级别标志的值和经解码的第一变换系数级别标志的值推导量化变换系数,并且在奇偶校验级别标志进行解码之前执行对第一变换系数级别标志的解码。

[0021] 技术效果

[0022] 根据本公开,可以增强整体图像/视频压缩效率。

[0023] 根据本公开,可以增强残差编码的效率。

[0024] 根据本公开,可以通过基于莱斯参数对残差信息执行二值化处理来增强残差编码效率。

[0025] 根据本公开,可以增强变换系数级别编码的效率。

[0026] 根据本公开,可以通过基于莱斯参数对残差信息执行二值化处理来增强残差编码效率。

[0027] 根据本公开,可以通过确定(或改变)用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志的解码顺序来增强编码效率。

[0028] 根据本公开,可以通过将残差信息中所包括的当前块内的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和限制为预定阈值或更小来增强编码效率。

[0029] 根据本公开,可以通过将残差信息中包括的当前块内的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和限制为预定阈值或更小来减少基于上下文编码的数据。

附图说明

[0030] 图1是示意性地例示了可以应用本公开的视频/图像编码系统的示例的图。

[0031] 图2是示意性地说明可以应用本公开的视频/图像编码设备的配置的图。

[0032] 图3是示意性地说明可以应用本公开的视频/图像解码设备的配置的图。

[0033] 图4是根据实施方式的CABAC编码系统的框图。

[0034] 图5是例示了 4×4 块内的变换系数的示例的图。

[0035] 图6是例示了 2×2 块内的变换系数的示例的图。

[0036] 图7是例示了根据实施方式的编码设备的操作的流程图。

[0037] 图8是例示了根据实施方式的编码设备的配置的框图。

[0038] 图9是例示了根据实施方式的解码设备的操作的流程图。

[0039] 图10是例示了根据实施方式的解码设备的配置的框图。

[0040] 图11是例示了可以应用本文件中公开的公开内容的上下文流系统的示例的图。

具体实施方式

[0041] 本公开的实施方式提供了一种由解码设备执行的用于解码图像的方法。该方法包括:接收包括残差信息的比特流;基于比特流中包括的残差信息,推导当前块的量化变换系数;基于量化变换系数,推导当前块的残差样本;以及基于当前块的残差样本生成重构图片,并且残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志,量化变换系数的推导包括:对第一变换系数级别标志进行解码;对奇偶校验级别标志进行解码;以及基于经解码的奇偶校验级别标志和经解码的第一变换系数级别标志推导量化变换系数,并且在奇偶校验级别标志进行解码之前执行对第一变换系数级别标志的解码。

[0042] 本公开可以以各种方式改变并且可以具有各种实施方式,并且将详细描述并在附图中例示出其特定实施方式。然而,这并非将本公开限制于特定实施方式。本说明书中使用的术语仅用于描述特定实施方式,并非旨在限制本公开的技术精神。单数表述包括复数表述,只要清楚地在此上下文中不同地理解它即可。本说明书中的诸如“包括”和“具有”之类的术语旨在表示说明书中使用的特征、数量、步骤、操作、组件、部件或其组合的存在,并且应当理解,并没有预先排除存在或增加一个或多个不同特征、数量、步骤、操作、组件、部件或其组合的可能性。

[0043] 此外,为了便于描述不同特征功能而独立地例示了本公开中描述的附图中的每个组件,并且并不意味着以单独的硬件或单独的软件来实现每个组件。例如,两个或多个组件可以组合以形成一个组件,或者一个组件可以划分为多个组件。在不脱离本公开的精神的情况下,每个组件被集成和/或分离的实施方式也包括在本公开的范围之内。

[0044] 在下文中,将参照附图更详细地描述本公开的优选实施方式。在下文中,在附图中,相同的附图标记用于相同的组件,并且可以省略相同组件的冗余描述。

[0045] 图1例示了可以应用本公开的视频/图像编码系统的示例。

[0046] 本文件涉及视频/图像编码。例如,本文件中公开的方法/实施方式可以应用于在多功能视频编码(VVC)标准、基本视频编码(EVC)标准、AOMedia Video 1(AV1)标准、第2代音频视频编码标准(AVS2)或下一代视频/图像编码标准(例如,H.267、H.268等)中公开的方法。

[0047] 本文件提出了视频/图像编码的各种实施方式,并且除非另外指明,否则以上实施方式也可以彼此组合地执行。

[0048] 在本文件中,视频可以是指一段时间上的一系列图像。图片通常是指特定时间帧表示一幅图像的单位,而切片/瓦片是指就编码而言组成图片的单位。切片/瓦片可以包括一个或多个编码树单元(CTU)。一幅图片可以包括一个或多个切片/瓦片。一幅图片可以包括一个或多个瓦片组。一个瓦片组可以包括一个或多个瓦片。砖块可以表示图片中瓦片内CTU行的矩形区域。一个瓦片可以分割为多个砖块,每个砖块包括瓦片内的一个或多个CTU行。没有分割为多个砖块的瓦片也可以称为砖块。砖块扫描是在砖块的CTU光栅扫描中连续排序的、对图片进行分割的CTU的特定顺序排序,瓦片内的砖块在瓦片的光栅扫描中是连续排列的,并且图片中的瓦片在图片的瓦片的光栅扫描中是连续排序的。瓦片是图片中特定瓦片列和特定瓦片行内的CTU的矩形区域。瓦片列是高度等于图片的高度并且宽度由图片参数集中的语法元素指定的CTU的矩形区域。瓦片行是高度由图片参

数集中的语法元素指定并且宽度等于图片宽度的CTU的矩形区域。瓦片扫描是指在瓦片的CTU光栅扫描中连续排序的、对图片进行分割的CTU的特定顺序排序,而图片中的瓦片在图片的瓦片的光栅扫描中是连续排序的。切片包括可以唯一地包含在单个NAL单元中的图片的整数个砖块。切片可以包括多个完整的瓦片,也可以包括仅一个瓦片的完整砖块的连续序列。在该文件中,瓦片组和切片可以互换使用。例如,在本文件中,瓦片组/瓦片组头也可以被称为切片/切片头。

[0049] 像素或画素可以表示组成一幅图片(或图像)的最小单位。另外,“样本”可以用作与像素相对应的术语。样本通常可以表示像素或像素值,并且可以仅表示亮度分量的像素/像素值或仅表示色度分量的像素/像素值。

[0050] 单元可以表示图像处理的基本单位。单元可以包括图片的特定区域和与该区域有关的信息中的至少一个。一个单元可以包括一个亮度块和两个色度(例如,cb、cr)块。在一些情况下,单元可以与诸如块或区域之类的术语互换使用。在一般情况下, $M \times N$ 块可以包括M列和N行的样本(或样本阵列)或变换系数的集合(或阵列)。

[0051] 在本文中,术语“/”和“、”应解释为表示“和/或”。例如,表述“A/B”可以表示“A和/或B”。此外,“A、B”可以表示“A和/或B”。此外,“A/B/C”可以表示“A、B和/或C中的至少一个”。另外,“A/B/C”可以表示“A、B和/或C中的至少一个”。

[0052] 此外,在本文中,术语“或”应解释为表示“和/或”。例如,表述“A或B”可包括1)仅A, 2)仅B,和/或3)A和B两者。换句话说,本文中的术语“或”应解释为表示“附加或另选地。”

[0053] 参照图1,视频/图像编码系统可以包括第一设备(源装置)和第二设备(接收装置)。源设备可以通过数字存储介质或网络以文件或流的形式向接收设备发送编码视频/图像信息或数据。

[0054] 源设备可以包括视频源、编码设备和发送器。接收器可以包括接收器、解码设备和渲染器。编码设备可以称为视频/图像编码设备,并且解码设备可以称为视频/图像解码设备。发送器可以包括在编码设备中。接收器可以包括在解码设备中。渲染器可以包括显示器,并且显示器可以被配置为单独的装置或外部组件。

[0055] 视频源可以通过捕获、合成或生成视频/图像的除了来获取视频/图像。视频源可以包括视频/图像捕获装置和/或视频/图像生成装置。视频/图像捕获装置可以包括例如一个或多个相机、包括先前捕获的视频/图像的视频/图像档案等。视频/图像生成装置可以包括例如计算机、平板电脑和智能手机,并且可以(电子地)生成视频/图像。例如,可以通过计算机等生成虚拟视频/图像。在这种情况下,视频/图像捕获处理可以由生成相关数据的处理代替。

[0056] 编码设备可以对输入的视频/图像进行编码。编码设备可以执行一系列过程,诸如预测、变换和量化以实现压缩和编码效率。编码数据(编码视频/图像信息)可以以比特流的形式输出。

[0057] 发送器可以通过数字存储介质或网络以文件或流的形式向接收装置的接收器发送以比特流形式输出的编码后的图像/图像信息或数据。数字存储介质可以包括诸如USB、SD、CD、DVD、蓝光、HDD、SSD等的各种存储介质。发送器可以包括用于通过预定文件格式生成媒体文件的元件,并且可以包括用于通过广播/通信网络进行发送的元件。接收器可以接收/提取比特流,并向解码设备发送接收到的比特流。

[0058] 解码设备可以通过执行与编码设备的操作相对应的诸如反量化、逆变换和预测之类的一系列过程,来对视频/图像进行解码。

[0059] 渲染器可以渲染经解码的视频/图像。经渲染的视频/图像可以通过显示器显示。

[0060] 图2例示了可以应用本公开的视频/图像编码设备的结构。在下文中,视频编码设备可以包括图像编码设备。

[0061] 参照图2,编码设备200包括图像分割器210、预测器220、残差处理器230和熵编码器240、加法器250、滤波器260和存储器270。预测器220可以包括帧间预测器221和帧内预测器222。残差处理器230可以包括变换器232、量化器233、反量化器234和逆变换器235。残差处理器230还可以包括减法器231。加法器250可以称为重构器或重构块生成器。根据实施方式,图像分割器210、预测器220、残差处理器230、熵编码器240、加法器250和滤波器260可以由至少一个硬件组件(例如,编码器芯片组或处理器)构成。此外,存储器270可以包括解码图片缓冲器(DPB)或者可以由数字存储介质构成。硬件组件还可以包括作为内部/外部组件的存储器270。

[0062] 图像分割器210可以将输入到编码设备200的输入图像(或图片或帧)分割到一个或多个处理器中。例如,处理器可以被称为编码单元(CU)。在这种情况下,可以根据四叉树二叉树三叉树(QTBT)结构从编码树单元(CTU)或最大编码单元(LCU)来递归地分割编码单元。例如,一个编码单元可以基于四叉树结构、二叉树结构和/或三元结构,被分割为深度更深的多个编码单元。在这种情况下,例如,可以首先应用四叉树结构,随后可以应用二叉树结构和/或三元结构。另选地,可以首先应用二叉树结构。可以基于不再分割的最终编码单元来执行根据本公开的编码过程。在这种情况下,可以根据图像特性基于编码效率将最大编码单元用作最终编码单元,或者如果需要,可以将编码单元递归地分割为深度更深的编码单元并且具有最佳尺寸的编码单元可以用作最终编码单元。这里,编码过程可以包括预测、变换和重构的过程,这将在后面描述。作为另一示例,处理器还可以包括预测单元(PU)或变换单元(TU)。在这种情况下,可以从上述最终编码单元来分离或分割预测单元和变换单元。预测单元可以是样本预测的单元,并且变换单元可以是用于推导变换系数的单元和/或用于从变换系数推导残差信号的单元。

[0063] 在一些情况下,单元可以与诸如块或区域之类的术语互换使用。在一般情况下, $M \times N$ 块可以表示由M列和N行组成的样本或变换系数的集合。样本通常可以表示像素或像素值,可以仅表示亮度分量的像素/像素值,或者仅表示色度分量的像素/像素值。样本可用作与像素或画素的一幅图片(或图像)相对应的术语。

[0064] 在编码设备200中,从输入图像信号(原始块、原始样本阵列)中减去从帧间预测器221或帧内预测器222输出的预测信号(预测块、预测样本阵列),以生成残差信号(残差块、残差样本阵列)并且向变换器232发送所生成的残差信号。在这种情况下,如图所示,在编码设备200中用于从输入图像信号(原始块、原始样本阵列)减去预测信号(预测块、预测样本阵列)的单元可以称为减法器231。预测器可以对要处理的块(在下文中称为当前块)执行预测,并生成包括当前块的预测样本的预测块。预测器可以以当前块或CU为基础来确定是应用帧内预测还是应用帧间预测。如稍后在每个预测模式的描述中所述,预测器可以生成与预测有关的、诸如预测模式信息之类的信息各种信息,并向熵编码器240发送所生成的信息。关于预测的信息可以在熵编码器240中编码并以比特流的形式输出。

[0065] 帧内预测器222可以通过参考当前图片中的样本来预测当前块。根据预测模式,参考的样本可以位于当前块的附近,或者可以远离当前块。在帧内预测中,预测模式可以包括多个非定向模式和多个定向模式。非定向模式可以包括例如DC模式和平面模式。根据预测方向的详细程度,定向模式可以包括例如33个定向预测模式或65个定向预测模式。然而,这仅是示例,依据设置可以使用更多或更少的定向预测模式。帧内预测器222可以通过使用应用于相邻块的预测模式来确定应用于当前块的预测模式。

[0066] 帧间预测器221可以基于由参考图片上的运动矢量指定的参考块(参考样本阵列)来推导当前块的预测块。这里,为了减少在帧间预测模式下发送的运动信息的量,可以基于相邻块和当前块之间的运动信息的相关性以块、子块或样本为单位来预测运动信息。运动信息可以包括运动矢量和参考图片索引。运动信息还可包括帧间预测方向(L0预测、L1预测、Bi预测等)信息。在帧间预测的情况下,相邻块可以包括存在于当前图片中的空间相邻块和存在于参考图片中的时间相邻块。包括参考块的参考图片和包括时间相邻块的参考图片可以相同或不同。时间相邻块可以称为并置参考块、共位CU(colCU)等,并且包括时间相邻块的参考图片可以称为并置图片(colPic)。例如,帧间预测器221可以基于相邻块来配置运动信息候选者列表,并且生成表示哪个候选者用于推导当前块的运动矢量和/或参考图片索引的信息。可以基于各种预测模式来执行帧间预测。例如,在跳过模式和合并模式的情况下,帧间预测器221可以将相邻块的运动信息用作当前块的运动信息。在跳过模式下,与合并模式不同,可能无法发送残差信号。在运动矢量预测(MVP)模式的情况下,可以将相邻块的运动矢量用作运动矢量预测器,并且可以通过发信号通知运动矢量差来指示当前块的运动矢量。

[0067] 预测器220可以基于以下描述的各种预测方法来生成预测信号。例如,预测器不仅可以应用帧内预测或帧间预测来预测一个块,而且可以同时应用帧内预测和帧间预测。这可以称为帧间帧内组合预测(CIIP)。此外,预测器可以基于帧内块复制(IBC)预测模式或调色板模式来预测块。IBC预测模式或调色板模式可用于游戏等的内容图像/视频编码,例如,屏幕内容编码(SCC)。IBC基本上在当前图片中执行预测,但是可以类似于帧间预测来执行IBC,因为在当前图片中推导参考块。即,IBC可以使用本文中描述的帧间预测技术中的至少一种。调色板模式可以被视为帧内编码或帧内预测的示例。当应用调色板模式时,可以基于关于调色板表和调色板索引的信息来发信号通知图片内的样本值。

[0068] 由预测器(包括帧间预测器221和/或帧内预测器222)生成的预测信号可以用于生成重构信号或生成残差信号。变换器232可以通过向残差信号应用变换技术来生成变换系数。例如,变换技术可以包括离散余弦变换(DCT)、离散正弦变换(DST)、Karhunen-Loève变换(KLT)、基于图的变换(GBT)或条件非线性变换(CNT)中的至少一种。这里,GBT表示当像素之间的关系信息由图表示时从图获得的变换。CNT是指基于使用所有先前重构的像素生成的预测信号而生成的变换。此外,变换处理可以应用于具有相同尺寸的正方形像素块,或者可以应用于具有可变尺寸而非正方形的块。

[0069] 量化器233可以对变换系数进行量化,并且将它们发送给熵编码器240,并且熵编码器240可以对量化信号(关于量化变换系数的信息)进行编码并且输出比特流。关于量化变换系数的信息可以称为残差信息。量化器233可以基于系数扫描顺序将块类型量化变换系数重新布置为一维矢量形式,并且基于一维矢量形式的量化变换系数来生成关于量化变

换系数的信息。可以生成关于变换系数的信息。熵编码器240可以执行各种编码方法,诸如,例如指数哥伦布(Golomb)、上下文自适应变长编码(CAVLC)、上下文自适应二进制算术编码(CABAC)等。熵编码器240可以对除了量化变换系数以外的视频/图像重构所需的信息(例如,语法元素的值等)一起或分开地进行编码。可以以比特流的形式以NAL(网络抽象层)为单位发送或存储编码信息(例如,编码视频/图像信息)。视频/图像信息还可以包括关于诸如自适应参数集(APS)、图片参数集(PPS)、序列参数集(SPS)或视频参数集(VPS)之类的各种参数集的信息。此外,视频/图像信息还可包括一般约束信息。在本文件中,从编码设备向解码设备发送/发信号通知的信息和/或语法元素可以包括在视频/图片信息中。视频/图像信息可以通过上述编码过程被编码并且包括在比特流中。比特流可以通过网络发送,或者可以存储在数字存储介质中。网络可以包括广播网络和/或通信网络,并且数字存储介质可以包括诸如USB、SD、CD、DVD、蓝光、HDD、SSD等的各种存储介质。可以包括发送从熵编码器240输出的信号的发送器(未示出)和/或存储该信号的存储单元(未示出)作为编码设备200的内部/外部元件,另选地,发送器可以包括在熵编码器240中。

[0070] 从量化器233输出的量化变换系数可以用于生成预测信号。例如,可以通过利用反量化器234和逆变换器235对量化变换系数应用反量化和逆变换,来重构残差信号(残差块或残差样本)。加法器250将重构的残差信号与从帧间预测器221或帧内预测器222输出的预测信号相加,以生成重构信号(重构图片、重构块、重构样本阵列)。如果要处理的块没有残差(诸如应用了跳过模式的情况),则可以将预测块用作重构块。加法器250可以称为重构器或重构块生成器。所生成的重构信号可以用于在当前图片中要处理的下一块的帧内预测,并且可以通过如下所述的滤波用于下一图片的帧间预测。

[0071] 此外,在图片编码和/或重构期间,可以应用具有色度缩放的亮度映射(LMCS)。

[0072] 滤波器260可以通过对重构信号应用滤波来改善主观/客观图像质量。例如,滤波器260可以通过对重构图片应用各种滤波方法来生成修改后的重构图片,并将修改后的重构图片存储在存储器270(具体地,存储器270的DPB)中。各种滤波方法可包括例如去块滤波、样本自适应偏移、自适应环路滤波器、双边滤波器等。滤波器260可以生成与滤波有关的各种信息,并且将生成的信息发送给熵编码器240,如稍后在各种滤波方法的描述中所述。与滤波有关的信息可以由熵编码器240编码并且以比特流的形式输出。

[0073] 发送给存储器270的修改后的重构图片可以用作帧间预测器221中的参考图片。当通过编码设备应用帧间预测时,可以避免编码设备200与解码设备之间的预测不匹配,并且可以提高编码效率。

[0074] 存储器270的DPB可以存储用作帧间预测器221中的参考图片的修改后的重构图片。存储器270可以存储从中推导(或编码)当前图片中的运动信息的块的运动信息和/或图片中已重构的块的运动信息。所存储的运动信息可以发送给帧间预测器221,并且用作空间相邻块的运动信息或时间相邻块的运动信息。存储器270可以存储当前图片中的重构块的重构样本,并且可以将重构样本传送给帧内预测器222。

[0075] 图3例示了可以应用本公开的视频/图像解码设备的结构。

[0076] 参照图3,解码设备300可以包括熵解码器310、残差处理器320、预测器330、加法器340、滤波器350、存储器360。预测器330可以包括帧间预测器332和帧内预测器331。残差处理器320可以包括反量化器321和逆变换器322。根据实施方式,熵解码器310、残差处理器

320、预测器330、加法器340和滤波器350可以由硬件组件(例如,解码器芯片组或处理器)构成。此外,存储器360可以包括解码图片缓冲器(DPB),或者可以由数字存储介质构成。硬件组件还可以包括存储器360作为内部/外部组件。

[0077] 当输入了包括视频/图像信息的比特流时,解码设备300可以与在图2的编码设备中处理视频/图像信息的处理相对应地重构图像。例如,解码设备300可以基于从比特流获得的块分割有关信息来推导单元/块。解码设备300可以使用在编码设备中应用的处理器来执行解码。因此,解码的处理器例如可以是编码单元,并且可以根据二叉树结构、二叉树结构和/或三叉树结构,从编码树单元或最大编码单元对编码单元进行分割。可以从编码单元推导一个或更多个变换单元。可以通过再现设备来再现通过解码设备300解码并输出的重构图像信号。

[0078] 解码设备300可以以比特流形式接收从图2的编码设备输出的信号,并且可以通过熵解码器310对接收到的信号进行解码。例如,熵解码器310可以解析比特流,以推导图像重构(或图片重构)所需的信息(例如,视频/图像信息)。视频/图像信息还可以包括关于诸如自适应参数集(APS)、图片参数集(PPS)、序列参数集(SPS)或视频参数集(VPS)之类的各种参数集的信息。此外,视频/图像信息还可以包括一般约束信息。解码设备还可以基于关于参数集的信息和/或一般约束信息来对图片进行解码。本文中稍后描述的发信号通知的/接收的信息和/或语法元素可以通过解码过程被解码,并从比特流中获取。例如,熵解码器310基于诸如指数哥伦布编码、CAVLC或CABAC之类的编码方法对比特流中的信息进行解码,并输出图像重构所需的语法元素和残差的变换系数的量化值。更具体地,CABAC熵解码方法可以接收与比特流中的每个语法元素相对应的bin(位),使用解码目标语法元素信息、解码目标块的解码信息或在先前级中解码的符号/bin的信息来确定上下文模型,并通过根据所确定的上下文模型预测bin的出现概率来对该bin进行算术解码,并且生成与每个语法元素的值相对应的符号。在这种情况下,在确定上下文模型之后,CABAC熵解码方法可以通过将经解码的符号/bin的信息用于下一符号/bin的上下文模型来更新上下文模型。由熵解码器310解码的信息当中与预测有关的信息可以提供给预测器(帧间预测器332和帧内预测器331),并且在熵解码器310中对其执行了熵解码的残差值(也就是说,量化变换系数和相关参数信息)可以输入到残差处理器320。残差处理器320可以推导残差信号(残差块、残差样本、残差样本阵列)。此外,由熵解码器310解码的信息当中关于滤波的信息可以提供给滤波器350。此外,用于接收从编码设备输出的信号的接收器(未示出)可以进一步被配置为解码设备300的内部/外部元件,或者接收器可以是熵解码器310的组件。此外,根据本文件的解码设备可以称为视频/图像/图片解码设备,并且解码设备可以分类为信息解码器(视频/图像/图片信息解码器)和样本解码器(视频/图像/图片样本解码器)。信息解码器可以包括熵解码器310,并且样本解码器可以包括反量化器321、逆变换器322、加法器340、滤波器350、存储器360、帧间预测器332和帧内预测器331中的至少一个。

[0079] 反量化器321可以对量化变换系数进行反量化并且输出变换系数。反量化器321可以以二维块的形式重新布置量化变换系数。在这种情况下,可以基于在编码设备中执行的系数扫描顺序来执行重新布置。反量化器321可以通过使用量化参数(例如,量化步长信息)对量化变换系数执行反量化,并且获得变换系数。

[0080] 逆变换器322对变换系数进行逆变换以获得残差信号(残差块、残差样本阵列)。

[0081] 预测器可以对当前块执行预测,并生成包括当前块的预测样本的预测块。预测器可以基于从熵解码器310输出的关于预测的信息来确定向当前块应用帧内预测还是帧间预测,并且可以确定具体的帧内/帧间预测模式。

[0082] 预测器可以基于以下描述的各种预测方法来生成预测信号。例如,预测器不仅可以应用帧内预测或帧间预测来预测一个块,而且可以同时应用帧内预测和帧间预测。这可以称为帧间和帧内组合预测(CIIP)。此外,预测器可以基于帧内块复制(IBC)预测模式或调色板模式来预测块。IBC预测模式或调色板模式可以用于游戏等的内容图像/视频编码,例如,屏幕内容编码(SCC)。IBC基本上在当前图片中执行预测,但是可以类似于帧间预测来执行IBC,因为在当前图片中推导参考块。即,IBC可以使用本文件中描述的帧间预测技术中的至少一种。调色板模式可以被视为帧内编码或帧内预测的示例。当应用调色板模式时,可以基于关于调色板表和调色板索引的信息来发信号通知图片内的样本值。

[0083] 帧内预测器331可以通过参考当前图片中的样本来预测当前块。根据预测模式,参考的样本可以位于当前块的附近,或者可以远离当前块。在帧内预测中,预测模式可以包括多个非定向模式和多个定向模式。帧内预测器331可以通过使用应用于相邻块的预测模式来确定应用于当前块的预测模式。

[0084] 帧间预测器332可以基于由参考图片上的运动矢量指定的参考块(参考样本阵列)来推导当前块的预测块。在这种情况下,为了减少在帧间预测模式中发送的运动信息的量,可以基于相邻块和当前块之间的运动信息的相关性,以块、子块或样本为单位来预测运动信息。运动信息可以包括运动矢量和参考图片索引。运动信息还可包括帧间预测方向(L0预测、L1预测、Bi预测等)信息。在帧间预测的情况下,相邻块可以包括存在于当前图片中的空间相邻块和存在于参考图片中的时间相邻块。例如,帧间预测器332可以基于相邻块来配置运动信息候选者列表,并基于接收到的候选者选择信息来推导当前块的运动矢量和/或参考图片索引。可以基于各种预测模式来执行帧间预测,并且关于预测的信息可以包括表示针对当前块的帧间预测的模式的信息。

[0085] 加法器340可以通过将所获得的残差信号与从预测器(包括帧间预测器332和/或帧内预测器331)输出的预测信号(预测块、预测样本阵列)相加来生成重构信号(重构图片、重构块、重构样本阵列)。如果要处理的块没有残差(例如当应用跳过模式时),则可以将预测块用作重构块。

[0086] 加法器340可以称为重构器或重构块生成器。所生成的重构信号可以用于当前图片中要处理的下一块的帧内预测,可以通过如下所述的滤波输出,或者可以用于下一图片的帧间预测。

[0087] 此外,在图片解码处理中可以应用具有色度缩放的亮度映射(LMCS)。

[0088] 滤波器350可以通过向重构信号应用滤波来改善主观/客观图像质量。例如,滤波器350可以通过对重构图片应用各种滤波方法来生成修改后的重构图片,并将修改后的重构图片存储在存储器360(具体地,存储器360的DPB)中。各种滤波方法可包括例如去块滤波、样本自适应偏移、自适应环路滤波器、双边滤波器等。

[0089] 存储器360的DPB中存储的(修改后的)重构图片可以用作帧间预测器332中的参考图片。存储器360可以存储从中推导(或解码)当前图片中的运动信息的块的运动信息和/或图片中已重构的块的运动信息。所存储的运动信息可以发送给帧间预测器332,以作为空间

相邻块的运动信息或时间相邻块的运动信息来利用。存储器360可以存储当前图片中的重构块的重构样本,并且可以将重构样本传送给帧内预测器331。

[0090] 在本公开中,在编码设备200的滤波器260、帧间预测器221和帧内预测器222中描述的实施方式可以与解码设备300的滤波器350、帧间预测器332和帧内预测器331相同或分别被应用以对应于解码设备300的滤波器350、帧间预测器332和帧内预测器331。相同的内容也可以应用于帧间预测器332和帧内预测器331。

[0091] 如上所述,在执行视频编码时,执行预测以增强压缩效率。因此,可以生成包括作为编码目标块的当前块的预测样本的预测块。在此,预测块包括空间域(或像素域)中的预测样本。从编码设备和解码设备同样地推导预测块,并且编码设备可以向解码设备发信号通知关于原始块和预测块之间的残差的信息(残差信息),而不是原始块的原始样本值本身,从而增强图像编码效率。解码设备可以基于残差信息来推导包括残差样本的残差块,通过将残差块和预测块相加来生成包括重构样本的重构块,并且生成包括重构块的重构图片。

[0092] 可以通过变换和量化过程来生成残差信息。例如,编码设备可以通过推导原始块和预测块之间的残差块、通过对残差块中包括的残差样本(残差样本阵列)执行变换过程推导变换系数以及通过对变换系数执行量化过程推导量化变换系数,来向解码设备(通过比特流)发信号通知相关的残差信息。这里,残差信息可以包括诸如值信息、位置信息、变换技术、变换核以及量化变换系数的量化参数之类的信息。解码设备可以基于残差信息执行反量化/逆变换过程,并推导残差样本(或残差块)。解码设备可以基于预测块和残差块来生成重构图片。编码设备还可以通过对量化变换系数进行反量化/逆变换来推导残差块以用于后面对图片的帧间预测,并基于此生成重构图片。

[0093] 图4是根据实施方式的CABAC编码系统的框图。

[0094] 参照图4,例示了用于对单个语法元素进行编码的CABAC的框图。在对CABAC进行编码的处理中,首先,如果输入信号是非二进制语法元素,则可以通过二值化将输入信号转换为二进制值。如果输入信号已经是二进制值,则输入信号可以被旁路而不进行二值化。在此,构成二进制值的每个二进制0或1可以被称为bin。例如,当二值化后的二进制字符串为110时,1、1和0中的每一个都可以称为一个bin。

[0095] 二值化的bin可以输入到常规编码引擎或旁路编码引擎。常规编码引擎可以为相应bin指派反映概率值的上下文模型,并基于指派的上下文模型对相应的bin进行编码。常规编码引擎可以在对每个bin进行编码之后更新相应bin的概率模型。如此编码的bin称为上下文编码bin。旁路编码引擎可以省略估计输入的bin的概率的过程以及在编码之后更新应用于相应bin的概率模型的过程。通过应用均匀的概率分布而不是指派上下文来对输入的bin进行编码,可以提高编码速度。如此编码的bin可以称为旁路bin。

[0096] 熵编码可以确定是通过常规编码引擎进行编码还是通过旁路编码引擎进行编码,并切换编码路径。熵解码可以反向地执行与编码相同的处理。

[0097] 在实施方式中,可以基于诸如transform_skip_flag、last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix、coded_sub_block_flag、sig_coeff_flag、par_level_flag、rem_abs_gt1_flag、rem_abs_gt2_flag、abs_remainder、coeff_sign_flag和mts_idx之类的语法元素来对(量化)变换系

数进行编码和/或解码。下表1表示与残差数据的编码有关的语法元素。

[0098] 表1

[0099]

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	描述符
if(transform_skip_enabled_flag && (cIdx != 0 cu_mts_flag[x0][y0] == 0) && (log2TbWidth <= 2) && (log2TbHeight <= 2))	
transform_skip_flag[x0][y0][cIdx]	ac(v)
last_sig_coeff_x_prefix	ac(v)
last_sig_coeff_y_prefix	ac(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ac(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ac(v)
log2SbSize = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
numSbCoeff = 1 << (log2SbSize << 1)	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - 2 * log2SbSize)) - 1	
do {	
if(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][0]	
yS = DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][1]	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][lastScanPos][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][lastScanPos][1]	
} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
QState = 0	
for(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	

[0100]

$xS =$ DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][0]	
$yS =$ DiagScanOrder[log2TbWidth - log2SbSize][log2TbHeight - log2SbSize] [lastSubBlock][1]	
inferSbDcSigCoeffFlag = 0	
if((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
for(n = (i == lastSubBlock) ? lastScanPos - 1 : numSbCoeff - 1; n >= 0; n--) {	
$xC = (xS << \log_2SbSize) +$ DiagScanOrder[log2SbSize][log2SbSize][n][0]	
$yC = (yS << \log_2SbSize) +$ DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
}	
if(sig_coeff_flag[xC][yC]) {	
par_level_flag[n]	ae(v)
rem_abs_gt1_flag[n]	ae(v)
if(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] + par_level_flag[n] + 2 * rem_abs_gt1_flag[n]	
if(dep_quant_enabled_flag)	
QState = QStateTransTable[QState][par_level_flag[n]]	
}	
for(n = numSbCoeff - 1; n >= 0; n--) {	
if(rem_abs_gt1_flag[n])	
rem_abs_gt2_flag[n]	ae(v)
}	
for(n = numSbCoeff - 1; n >= 0; n--) {	
$xC = (xS << \log_2SbSize) +$ DiagScanOrder[log2SbSize][log2SbSize][n][0]	
$yC = (yS << \log_2SbSize) +$ DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(rem_abs_gt2_flag[n])	
abs_remainder[n]	

[0101]

AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * (rem_abs_gt2_flag[n] + abs_remainder[n])	
}	
if(dep_quant_enabled_flag !sign_data_hiding_enabled_flag)	
signHidden = 0	
else	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0)	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC] && (!signHidden (n != firstSigScanPosSb)))	
coeff_sign_flag[n]	ac(v)
}	
if(dep_quant_enabled_flag) {	
QState = startQStateSb	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
if(sig_coeff_flag[xC][yC])	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1 : 0)) * (1 - 2 * coeff_sign_flag[n])	
QState = QStateTransTable[QState][par_level_flag[n]]	
} else {	
sumAbsLevel = 0	
for(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	

[0102]

if(sig_coeff_flag[xC][yC]) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff_sign_flag[n])	
if(signHidden) {	
sumAbsLevel += AbsLevel[xC][yC]	
if((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1))	
TransCoeffLevel[x0][y0][cIdx][xC][yC] = -TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	
}	
}	
}	
if(cu_mts_flag[x0][y0] && (cIdx == 0) && !transform_skip_flag[x0][y0][cIdx] && ((CuPredMode[x0][y0] == MODE_INTRA && numSigCoeff > 2) (CuPredMode[x0][y0] == MODE_INTER))) {	
mts_idx[x0][y0]	ae(v)
}	

[0103] transform_skip_flag表示在相关联块中是否省略了变换。相关联块可以是编码块(CB)或变换块(TB)。关于变换(和量化)和残差编码过程,CB和TB可以互换使用。例如,如上所述,可以针对CB推导残差样本,可以通过残差样本的变换和量化来推导(量化)变换系数,并且可以通过残差编码过程生成并发信号通知有效地表示(量化)变换系数的位置、尺寸、符号等的信息(例如,语法元素)。量化变换系数可以简单地称为变换系数。通常,如果CB不大于最大TB,则CB的尺寸可以等于TB的尺寸,并且在这种情况下,要变换(和量化)和残差编码的目标块可以称为CB或TB。此外,如果CB大于最大TB,则要变换(和量化)和残差编码的目标块可以称为TB。在下文中,尽管将描述以变换块(TB)为单位发信号通知与残差编码有关的语法元素,但是这是示例,并且如上所述,TB可以与编码块(CB)互换使用。

[0104] 在实施方式中,可以基于语法元素last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix和last_sig_coeff_y_suffix来对变换块内的最后非零变换系数的(x,y)位置信息进行编码。更具体地说,last_sig_coeff_x_prefix表示变换块内按扫描顺序的最后一个有效系数的列位置的前缀,last_sig_coeff_y_prefix表示变

变换块内按扫描顺序的最后一个有效系数的行位置的前缀, `last_sig_coeff_x_suffix`表示变换块内按扫描顺序的最后一个有效系数的列位置的后缀, 并且`last_sig_coeff_y_suffix`表示变换块内按扫描顺序的最后一个有效系数的行位置的后缀。在此, 有效系数可以表示非零系数。扫描顺序可以是右上对角线扫描顺序。另选地, 扫描顺序可以是水平扫描顺序或垂直扫描顺序。可以基于是否向目标块(CB或包括TB的CB)应用帧内/帧间预测和/或特定帧内/帧间预测模式来确定扫描顺序。

[0105] 随后, 在将变换块划分为 4×4 子块之后, 可以每 4×4 子块使用1比特语法元素`coded_sub_block_flag`来表示当前子块内是否存在非零系数。

[0106] 如果`coded_sub_block_flag`的值为0, 则没有更多的信息要发送, 从而可以终止当前子块的编码处理。相反, 如果`coded_sub_block_flag`的值为1, 则可以继续执行`sig_coeff_flag`的编码处理。由于包括最后一个非零系数的子块不需要对`coded_sub_block_flag`进行编码并且包括变换块的DC信息的子块包括非零系数的可能性高, 因此, 不对`coded_sub_block_flag`进行编码, 并且可以假设其值为1。

[0107] 如果`coded_sub_block_flag`的值为1并且确定当前子块内存在非零系数, 则可以根据反扫描顺序对具有二进制值的`sig_coeff_flag`进行编码。可以根据扫描顺序针对每个系数, 对1比特语法元素`sig_coeff_flag`进行编码。如果当前扫描位置处的变换系数的值是非零, 则`sig_coeff_flag`的值可以为1。这里, 在包括最后一个非零系数的子块的情况下, 由于最后一个非零系数不需要`sig_coeff_flag`, 因此可以省略子块的编码处理。仅当`sig_coeff_flag`为1时, 才可以对级别信息进行编码, 并且在级别信息编码处理中可以使用四个语法元素。更具体地, 每个`sig_coeff_flag[xC][yC]`可以表示在当前TB内的每个变换系数位置(xC, yC)处的相应变换系数的级别(值)是否为非零。在实施方式中, `sig_coeff_flag`可以对应于表示量化变换系数是否为非零有效系数的有效系数标志的示例。

[0108] 对`sig_coeff_flag`进行编码之后的其余级别值可以由下式1表示。即, 表示要编码的级别值的语法元素`remAbsLevel`可以由下式1所示。在此, `coeff`表示实际的变换系数值。

[0109] 式1

[0110] $\text{remAbsLevel} = |\text{coeff}| - 1$

[0111] 如下式2所表示的, 可以通过`par_level_flag`对由式1表示的`remAbsLevel`的最低有效系数(LSB)的值进行编码。在此, `par_level_flag[n]`可以表示扫描位置(n)处的变换系数级别(值)的奇偶校验。在对`par_level_flag`进行编码之后, 可以更新要编码的变换系数级别值`remAbsLevel`, 如下式3所表达的。

[0112] 式2

[0113] $\text{par_level_flag} = \text{remAbsLevel} \& 1$

[0114] 式3

[0115] $\text{remAbsLevel}' = \text{remAbsLevel} \gg 1$

[0116] `rem_abs_gt1_flag`可以表示相应扫描位置(n)处的`remAbsLevel'`是否大于1, 并且`rem_abs_gt2_flag`可以表示相应扫描位置(n)处的`remAbsLevel'`是否大于2。可以仅当`rem_abs_gt2_flag`为1时才对`abs_remainder`进行编码。例如, 如下式4所表示, 将实际变换系数值(`coeff`)与各个语法元素之间的关系进行总结, 并且下表2表示与式4有关的示例。每个系数的符号可以通过使用1比特符号`coeff_sign_flag`来编码。`|coeff|`表示变换系数级别

(值),也可以被表示为变换系数的AbsLevel。

[0117] 式4

[0118] $|\text{coeff}| = \text{sig_coeff_flag} + \text{par_level_flag} + 2 * (\text{rem_abs_gt1_flag} + \text{rem_abs_gt2_flag} + \text{abs_remainder})$

[0119] 表2

[0120]

coeff	sig_coeff_flag	par_level_flag	rem_abs_gt1_flag	rem_abs_gt2_flag	abs_remainder
0	0				
1	1	0	0		
2	1	1	0		
3	1	0	1	0	
4	1	1	1	0	
5	1	0	1	1	0
6	1	1	1	1	0
7	1	0	1	1	1
8	1	1	1	1	1
9	1	0	1	1	2
10	1	1	1	1	2
11	1	0	1	1	3
...	...	...	...	...	...

[0121] 此外,在实施方式中,par_level_flag可以表示量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志的示例,rem_abs_gt1_flag可以表示关于变换系数级别是否大于第一阈值的第一变换系数级别标志的示例,并且rem_abs_gt2_flag可以表示关于变换系数级别是否大于第二阈值的第二变换系数级别标志的示例。

[0122] 此外,在另一实施方式中,rem_abs_gt2_flag也可以称为rem_abs_gt3_flag,并且在又一实施方式中,可以基于abs_level_gtx_flag[n][j]来表示rem_abs_gt1_flag和rem_abs_gt2_flag.abs_level_gtx_flag[n][j]可以是表示扫描位置(n)处的变换系数级别的绝对值(或者通过将变换系数级别向右移动1位而获得的值)是否大于(j<<1)+1的标志。在示例中,rem_abs_gt1_flag可以执行与abs_level_gtx_flag[n][0]相同和/或相似的功能,并且rem_abs_gt2_flag可以执行与abs_level_gtx_flag[n][1]相同和/或相似的功能。即,abs_level_gtx_flag[n][0]可以对应于第一变换系数级别标志的示例,而abs_level_gtx_flag[n][1]可以对应于第二变换系数级别标志的示例。在一些情况下,也可以用诸如第一阈值和第二阈值之类的预定阈值来替换(j<<1)+1。

[0123] 图5是例示了4×4块内的变换系数的示例的图。

[0124] 图5所示的4×4块例示了量化系数的示例。图5中所示的块可以是4×4变换块,或者是8×8、16×16、32×32或64×64变换块的4×4子块。图5中所示的4×4块可以表示亮度块或色度块。例如,图5中所示的反向对角线扫描系数的编码结果可以是如表3所表示的。在表3中,scan_pos表示根据反向对角线扫描的系数的位置。scan_pos 15表示在4×4块中首先被扫描(即,右下角)的系数,而scan_pos 0表示最后被扫描(即,左上角)的系数。此外,在实施方式中,scan_pos也可以称为扫描位置。例如,scan_pos 0可以称为扫描位置0。

[0125] 表3

[0126]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----------	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	0	1
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	1	1
rem_abs_gt2_flag										0		0	0	1	1	1
abs_remainder														0	1	2
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0127] 如表1所示,在实施方式中,以 4×4 子块为单位的主要语法元素可以包括sig_coeff_flag、par_level_flag、rem_abs_gt1_flag、rem_abs_gt2_flag、abs_remainder、coeff_sign_flag等。其中,sig_coeff_flag、par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag可以表示使用常规编码引擎来编码的上下文编码的bin,而abs_remainder和coeff_sign_flag可以表示通过使用旁路编码引擎来编码的旁路bin。

[0128] 上下文编码的bin可以表示较高的数据依赖性,因为它使用在处理先前的bin时更新的概率状态和范围。即,由于对于上下文编码的bin,可以在当前bin全部被编码/解码之后对下一个bin进行编码/解码,因此可能难以进行并行处理。此外,读取概率部分并确定当前状态也可能花费长时间。因此,实施方式可以提出一种用于通过减少上下文编码的bin的数量并增加旁路bin的数量来提高CABAC吞吐量的方法。

[0129] 在实施方式中,可以以反向扫描顺序对系数级别信息进行编码。即,可以在从单位块的右下侧的系数开始朝左上侧的系数进行扫描之后对系数级别信息进行编码。在示例中,以反向扫描顺序首先扫描的系数级别可以表示较小的值。发信号通知用于这样首先扫描的系数的par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag可以减少用于表示系数级别的二值化bin的长度,并且可以通过使用预定的上下文,基于先前编码的上下文,通过算术编码来有效地对各个语法元素进行编码。

[0130] 但是,在具有较大值的一些系数级别(即,位于单位块的左上侧的系数级别)的情况下,发信号通知par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag可能无助于改善压缩性能。par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag的使用也可能降低编码效率。

[0131] 在实施方式中,通过将编码成上下文编码的bin的语法元素(par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag)快速切换到通过使用旁路编码引擎编码的(即,编码成旁路bin的)abs_remainder语法元素,可以减少上下文编码的bin的数量。

[0132] 在实施方式中,可以限制用于对rem_abs_gt2_flag进行编码的系数的数量。在 4×4 块内显式编码的rem_abs_gt2_flag的最大数量可以为16。也就是说,也可以针对绝对值大于2的所有系数对rem_abs_gt2_flag进行编码,并且在实施方式中,可以仅针对根据扫描顺序的前N个绝对值大于2的系数(即,rem_abs_gt1_flag为1的系数),对rem_abs_gt2_flag进行编码。N也可以由编码器选择,并且也可以设置为0到16的任意值。表4示出了当N为1时实施方式的应用示例。根据实施方式,在 4×4 块中,rem_abs_gt2_flag的编码次数可以减少多达由表4中的X所指示的次数,从而减少上下文编码的bin的数量。当与表3进行比较时,针对rem_abs_gt2_flag不被编码的扫描位置,可以如以下表4所表示地改变系数的abs_remainder值。

[0133] 表4

[0134]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	0	1
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	1	1
rem_abs_gt2_flag										0		X	X	X	X	X
abs_remainder												0	0	1	2	3
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0135] 在实施方式中,可以限制用rem_abs_gt1_flag编码的系数的数量。在4×4块内显式编码的rem_abs_gt2_flag的最大数量可以为16。即,可以针对绝对值大于零的所有系数,对rem_abs_gt1_flag进行编码,并且在实施方式中,可以仅针对根据扫描顺序的前M个绝对值大于零的系数(即,sig_coeff_flag为1的系数),对rem_abs_gt1_flag进行编码。M也可以由编码器选择,并且也可以设置为0到16中的任意值。表5示出了当M为4时实施方式的应用示例。根据本实施方式,如果rem_abs_gt1_flag不被编码,则rem_abs_gt2_flag也不被编码,使得在4×4块中rem_abs_gt1_flag和rem_abs_gt2_flag的编码次数可以减少多达由X所指示的次数,从而减少上下文编码的bin的数量。当与表3比较时针对rem_abs_gt1_flag不被编码的扫描位置,系数的rem_abs_gt2_flag和abs_remainder的值可以如下表6所表示地改变。表6示出了当M为8时实施方式的应用示例。

[0136] 表5

[0137]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	0	1
rem_abs_gt1_flag					0	0		0		1	X	X	X	X	X	X
rem_abs_gt2_flag										0		X	X	X	X	X
abs_remainder											0	1	1	2	3	4
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0138] 表6

[0139]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	0	1
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	X	X
rem_abs_gt2_flag										0		0	0	1	X	X
abs_remainder														0	3	4
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0140] 在实施方式中,可以组合分别限制rem_abs_gt1_flag的数量和rem_abs_gt2_flag的数量的前述实施方式。表示rem_abs_gt1_flag的数量限制的M和表示rem_abs_gt2_flag的数量限制的N全部可以具有0到16的值,但是N可以等于M或小于M。表7示出了当M为8并且N为1时本实施方式的应用示例。由于在用X标记的位置处没有对相应语法元素进行编码,因此可以减少上下文编码的bin的数量。

[0141] 表7

[0142]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	0	1
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	X	X
rem_abs_gt2_flag										0		X	X	X	X	X
abs_remainder												0	0	1	3	4
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0143] 在实施方式中,可以限制对par_level_flag进行编码的系数的数量。在4×4块内显式编码的par_level_flag的最大数量可以是16。也就是说,可以针对绝对值大于零的所有系数,对par_level_flag进行编码,并且在实施方式中,可以仅针对根据扫描顺序的前L个绝对值大于零的系数(即,sig_coeff_flag为1的系数),对par_level_flag进行编码。L也可以由编码器选择,并且也可以设置为0至16中的任意值。表8示出了当L为8时实施方式的应用示例。根据实施方式,4×4块中par_level_flag的编码次数可以减少由X指示的数量,从而减少上下文编码的bin的数量。当与表3比较时,针对par_level_flag不被编码的扫描位置,系数的rem_abs_gt1_flag、rem_abs_gt2_flag和abs_remainder的值可以如以下表8所表示地改变。

[0144] 表8

[0145]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	X	X
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	1	1
rem_abs_gt2_flag										0		0	0	1	1	1
abs_remainder														0	4	7
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0146] 在实施方式中,可以组合限制par_level_flag的数量和rem_abs_gt2_flag的数量前述实施方式。表示par_level_flag的数量限制的L和表示rem_abs_gt2_flag的数量限制的N全部可以具有从0到16的值。表9示出了当L为8并且N为1时本实施方式的应用示例。由于在用X标记的位置处没有对相应语法元素进行编码,因此可以减少上下文编码的bin的数量。

[0147] 表9

[0148]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	X	X
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	1	1
rem_abs_gt2_flag										0		X	X	X	X	X
abs_remainder												0	0	1	5	8
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0149] 在实施方式中,可以组合限制par_level_flag和rem_abs_gt1_flag的数量的上述实施方式。表示par_level_flag的数量限制的L和表示rem_abs_gt1_flag的数量限制的M全

部可以具有0至16的值。表10示出了当L为8且M为8时的本实施方式的应用示例。由于在用X标记的位置处没有对相应语法元素进行编码,因此可以减少上下文编码的bin的数量。

[0150] 表10

[0151]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	X	X
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	X	X
rem_abs_gt2_flag										0		0	0	1	X	X
abs_remainder														0	6	9
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0152] 在实施方式中,可以组合分别限制par_level_flag的数量、rem_abs_gt1_flag的数量和rem_abs_gt2_flag的数量的前述实施方式。表示par_level_flag的数量限制的L、表示rem_abs_gt1_flag的数量限制的M和表示rem_abs_gt2_flag的数量限制的N全部可以具有从0到16的值,但是N可以等于M或小于M。表11示出了当L为8,M为8且N为1时本实施方式的应用示例。由于在用X标记的位置处相应语法元素不被编码,因此可以减少上下文编码的bin的数量。

[0153] 表11

[0154]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
par_level_flag					0	0		1		0	1	0	1	1	X	X
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	X	X
rem_abs_gt2_flag										0		X	X	X	X	X
abs_remainder												0	0	1	6	9
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0155] 实施方式可以提出一种用于限制sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和的方法。假设sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和被限制为K,则K可以具有0至48的值。在本实施方式中,如果sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和超过K并且sig_coeff_flag、par_level_flag和rem_abs_gt1_flag不被编码,则rem_abs_gt2_flag也可以不被编码。表12示出了将K限制为30的情况。

[0156] 表12

[0157]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	X	X
par_level_flag					0	0		1		0	1	0	1	1	X	X
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	X	X
rem_abs_gt2_flag										0		0	0	1	X	X
abs_remainder														0	7	10
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0158] 在实施方式中,可以组合用于限制sig_coeff_flag的数量、par_level_flag的数

量和rem_abs_gt1_flag的数量之和的方法和用于限制rem_abs_gt2_flag的数量的上述方法。假设sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和被限制为K并且rem_abs_gt2_flag的数量被限制为N,则K可以具有0到48的值,并且N可以具有0到16的值。表13示出了K被限制为30而N被限制为2的情况。

[0159] 表13

[0160]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	X	X
par_level_flag					0	0		1		0	1	0	1	1	X	X
rem_abs_gt1_flag					0	0		0		1	0	1	1	1	X	X
rem_abs_gt2_flag										0		0	X	X	X	X
abs_remainder													0	1	7	10
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0161] 在实施方式中,可以按照par_level_flag和rem_abs_gt1_flag的顺序执行编码,但是在根据本公开的实施方式中,可以通过改变编码顺序按照rem_abs_gt1_flag和par_level_flag的顺序执行编码。如上所述,如果改变了par_level_flag和rem_abs_gt1_flag的编码顺序,则在对sig_coeff_flag进行编码之后对rem_abs_gt1_flag进行编码,并且仅当rem_abs_gt1_flag为1时才可以对par_level_flag进行编码。因此,实际变换系数值(coeff)和各个语法元素之间的关系可以改变,如下式5所表示。下表14示出改变了par_level_flag和rem_abs_gt1_flag的编码顺序的情况的示例。当与表2进行比较时,在下表14中,如果|coeff|为1,则不对par_level_flag进行编码,因此就吞吐量和编码而言是有利的。当然,在表14中,如果|coeff|为2,则需要对rem_abs_gt2_flag进行编码,这与表2不同;并且如果|coeff|为4,则需要对abs_remainder进行编码,这与表2不同,但是|coeff|为1的情况可能比|coeff|为2或4的情况出现得更多,使得根据表14的方法可以比根据表2的方法表现出更高的吞吐量和编码性能。可以由下表15表示如图5所示的4×4子块的编码结果。

[0162] 式5

[0163] $|coeff| = sig_coeff_flag + rem_abs_gt1_flag + par_level_flag + 2 * (rem_abs_gt2_flag + abs_remainder)$ 表14

[0164]

coeff	sig_coeff_flag	rem_abs_gt1_flag	par_level_flag	rem_abs_gt2_flag	abs_remainder
0	0				
1	1	0			
2	1	1	0	0	
3	1	1	1	0	
4	1	1	0	1	0
5	1	1	1	1	0
6	1	1	0	1	1
7	1	1	1	1	1
8	1	1	0	1	2
9	1	1	1	1	2
10	1	1	0	1	3
11	1	1	1	1	3
...	...	...	...	...	...

[0165] 表15

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
rem_abs_gt1_flag					0	0		1		1	1	1	1	1	1	1
par_level_flag								0		1	0	1	0	0	1	0
rem_abs_gt2_flag								0		0	0	0	1	1	1	1
abs_remainder													0	1	1	3
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0167] 在实施方式中,当按照sig_coeff_flag、rem_abs_gt1_flag、par_level_flag、rem_abs_gt2_flag、abs_remainder和coeff_sign_flag的顺序执行编码时,本公开可以提出一种用于限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和的方法。如果sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和被限制为K,则K可以具有0至48的值。在本实施方式中,当sig_coeff_flag、rem_abs_gt1_flag和par_level_flag不再被编码时,rem_abs_gt2_flag也可以不被编码。表16示出了K被限制为25的情况。

[0168] 表16

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	X	X	X
rem_abs_gt1_flag					0	0		1		1	1	1	1	X	X	X
par_level_flag								0		1	0	1	0	X	X	X
rem_abs_gt2_flag								0		0	0	0	1	X	X	X
abs_remainder													0	6	7	10
ceoff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0170] 在实施方式中,可以在(残差)语法内的一个for循环中对sig_coeff_flag、rem_abs_gt1_flag和par_level_flag进行编码。尽管三个语法元素(sig_coeff_flag、rem_abs_gt1_flag和par_level_flag)的数量之和未超过K并且三个语法元素的数量之和与K不完全匹配,但是可以在相同的扫描位置停止编码。下表17示出了K被限制为27的情况。当执行编码直到扫描位置3时,sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和为25。该和是不超过K的值,但是此时,由于编码设备不知道作为下一个扫描位置的scan_pos 2的系数级别的值,所以编码设备不知道在scan_pos 2中生成的上下文编码的bin的数量具有从1到3中的哪个值。因此,编码设备最多仅可以编码到scan_pos 3并终止编码。K值不同,但是编码结果可以相同,如表16和下表17表示。

[0171] 表17

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	X	X	X
rem_abs_gt1_flag					0	0		1		1	1	1	1	X	X	X
par_level_flag								0		1	0	1	0	X	X	X
rem_abs_gt2_flag								0		0	0	0	1	X	X	X
abs_remainder													0	6	7	10

coeff_sign_flag					0	1		0		0	1	1	0	0	1	0
-----------------	--	--	--	--	---	---	--	---	--	---	---	---	---	---	---	---

[0173] 实施方式可以提出一种用于改变par_level_flag和rem_abs_gt1_flag的编码顺序,并限制rem_abs_gt2_flag的数量的方法。也就是说,可以按照sig_coeff_flag、rem_abs_gt1_flag、par_level_flag、rem_abs_gt2_flag、abs_remainder和coeff_sign_flag的顺序执行编码,并且可以限制用rem_abs_gt2_flag编码的系数的数量。

[0174] 在 4×4 块内显式编码的rem_abs_gt2_flag的最大数量为16。即,可以针对绝对值大于2的所有系数,对rem_abs_gt2_flag进行编码。相反,在本实施方式中,可以仅针对按照扫描顺序的前N个绝对值大于2的系数(即,rem_abs_gt1_flag为1的系数),对rem_abs_gt2_flag进行编码。N也可以由编码器选择,并且也可以设置为0到16中的任意值。下表18示出了当N为1时本实施方式的应用示例。在 4×4 块中,rem_abs_gt2_flag的编码次数可以减少多达由X指示的数量,从而减少上下文编码的bin的数量。当与表15比较时,针对rem_abs_gt2_flag不被编码的扫描位置,系数的abs_remainder值可以改变,如表18所表示。

[0175] 表18

[0176]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	1	1	1
rem_abs_gt1_flag					0	0		1		1	1	1	1	1	1	1
par_level_flag								0		1	0	1	0	0	1	0
rem_abs_gt2_flag								0		0	X	X	X	X	X	X
abs_remainder											0	0	1	2	2	4
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0177] 实施方式可以提供一种用于改变par_level_flag和rem_abs_gt1_flag的编码顺序并分别限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和以及rem_abs_gt2_flag的数量的方法。在示例中,当按照sig_coeff_flag、rem_abs_gt1_flag、par_level_flag、rem_abs_gt2_flag、abs_remainder和coeff_sign_flag的顺序执行编码时,可以组合用于限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和的方法以及上述用于限制rem_abs_gt2_flag的数量的方法。假设sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和被限制为K,并且rem_abs_gt2_flag的数量被限制为N,则K可以具有0到48的值,并且N可以具有0到16的值。表19示出了K被限制为25并且N被限制为2的情况。

[0178] 表19

[0179]

scan_pos	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
系数	0	0	0	0	1	-1	0	2	0	3	-2	-3	4	6	-7	10
sig_coeff_flag	0	0	0	0	1	1	0	1	0	1	1	1	1	X	X	X
rem_abs_gt1_flag					0	0		1		1	1	1	1	X	X	X
par_level_flag								0		1	0	1	0	X	X	X
rem_abs_gt2_flag								0		0	X	X	X	X	X	X
abs_remainder											0	0	1	6	7	10
coeff_sign_flag					0	1		0		0	1	1	0	0	1	0

[0180] 图6是例示了 2×2 块内的变换系数的示例的图。

[0181] 图6所示的 2×2 块表示量化系数的示例。图6中所示的方框可以是 2×2 变换块、或

者 4×4 、 8×8 、 16×16 、 32×32 或 64×64 变换块的 2×2 子块。图6中所示的 2×2 块可以表示亮度块或色度块。例如,图6中所示的反向对角线扫描系数的编码结果可以由表20表示。在表20中,scan_pos表示根据反向角线扫描的系数的位置。scan_pos 3表示 2×2 块中首先被扫描(即,右下角)的系数,并且scan_pos 0表示最后被扫描(即,左上角)的系数。

[0182] 表20

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	1	1
par_level_flag	1	1	0	1
rem_abs_gt1_flag	1	1	1	1
rem_abs_gt2_flag	0	1	1	1
abs_remainder		0	1	2
coeff_sign_flag	0	0	1	0

[0184] 如表1所表示,在实施方式中,以 2×2 子块为单位的主要语法元素可以包括sig_coeff_flag、par_level_flag、rem_abs_gt1_flag、rem_abs_gt2_flag、abs_remainder、coeff_sign_flag等。其中,sig_coeff_flag、par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag可以表示通过使用常规编码引擎编码的上下文编码的bin,而abs_remainder和coeff_sign_flag可以表示通过使用旁路编码引擎编码的旁路bin。

[0185] 上下文编码的bin可以展现高数据依赖性,这是因为它使用在处理先前bin时更新的概率状态和范围。即,由于对于上下文编码的bin,可以在当前bin全部被编码/解码之后对下一个bin进行编码/解码,因此可能难以进行并行处理。此外,读取概率部分并确定当前状态可能花费长时间。因此,实施方式可以提出一种用于通过减少上下文编码的bin的数量并增加旁路bin的数量来提高CABAC吞吐量的方法。

[0186] 在实施方式中,可以以反向扫描顺序对系数级别信息进行编码。即,可以在从单位块的右下侧的系数开始朝左上侧的系数扫描之后对系数级别信息进行编码。在示例中,按反向扫描顺序首先扫描的系数级别可以表示较小的值。发信号通知用于这样首先扫描的系数的par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag可以减少用于表示系数级别的二值化bin的长度,并且可以通过使用预定的上下文,基于先前编码的上下文,通过算术编码来有效地对各个语法元素进行编码。

[0187] 但是,在具有较大值的一些系数级别(即,位于单位块的左上侧的系数级别)的情况下,发信号通知par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag可能无助于改善压缩性能。par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag的使用也可能降低编码效率。

[0188] 在实施方式中,通过将编码成上下文编码的bin的语法元素(par_level_flag、rem_abs_gt1_flag和rem_abs_gt2_flag)快速切换到通过使用旁路编码引擎编码的(即,编码成旁路bin的)abs_remainder语法元素,可以减少上下文编码的bin的数量。

[0189] 在实施方式中,可以限制用rem_abs_gt2_flag编码的系数的数量。在 2×2 块内显式编码的rem_abs_gt2_flag的最大数量可以为4。也就是说,也可以针对绝对值大于2的所有系数对rem_abs_gt2_flag进行编码,并且在实施方式中,可以仅针对根据扫描顺序的前N

个绝对值大于2的系数(即,rem_abs_gt1_flag为1的系数),对rem_abs_gt2_flag进行编码。N也可以由编码器选择,并且也可以设置为0到4的任意值。假设通过与本实施方式中类似的方法在编码器中限制用于亮度或色度4×4子块的上下文编码的bin,则也可以通过使用所使用的限制值来计算N。作为计算N的方法,用于亮度或色度4×4子块的上下文编码的bin的限制值($N_{4 \times 4}$)可以如式6中所表示的被原样使用,或者2×2子块中的像素数是4,使得可以通过式7来计算N。这里,a和b表示常数,并且不限于本公开中的特定值。

[0190] 类似地,也可以通过使用子块的水平/垂直尺寸值来计算N。由于子块具有正方形形状,因此水平尺寸值和垂直尺寸值相同。由于2×2子块的水平或垂直尺寸值为2,因此可以通过式8计算N。

[0191] 式6

[0192] $N = N_{4 \times 4}$

[0193] 式7

[0194] $N = \{N_{4 \times 4} \gg (4-a)\} + b$

[0195] 式8

[0196] $N = \{N_{4 \times 4} \gg (a-2)\} + b$

[0197] 表21示出了当N为1时本实施方式的应用示例。在2×2块中,rem_abs_gt2_flag的编码次数可以减少多达由X所指示的次数,从而减少上下文编码的bin的数量。当与表20比较时,rem_abs_gt2_flag不被编码的扫描位置的系数的abs_remainder值改变,如下表21所表示。

[0198] 表21

[0199]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	1	1
par_level_flag	1	1	0	1
rem_abs_gt1_flag	1	1	1	1
rem_abs_gt2_flag	0	X	X	X
abs_remainder		1	2	3
coeff_sign_flag	0	0	1	0

[0200] 在根据实施方式的色度块的2×2子块的编码中,可以限制sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和。假设sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和被限制为K,则K可以具有0至12的值。在本实施方式中,当sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和超过K并且sig_coeff_flag、par_level_flag和rem_abs_gt1_flag不被编码时,rem_abs_gt2_flag也可以不被编码。

[0201] K也可以由编码器选择,并且K可以设置为0到12的任意值。假设通过与本实施方式类似的方法在编码器中限制用于亮度或色度4×4子块的上下文编码的bin,则还可以通过使用所使用的限制值来计算K。作为计算K的方法,用于亮度或色度4×4子块的上下文编码的bin的限制值($K_{4 \times 4}$)可以如式9所表示的被原样使用,或者2×2子块中的像素数是4,使得可以通过式10来计算K。这里,a和b是常数,并且不限于本公开中的特定值。

[0202] 类似地,还可以通过使用子块的水平/垂直尺寸值来计算K。由于子块具有正方形形状,因此水平尺寸值和垂直尺寸值相同。由于 2×2 子块的水平或垂直尺寸值为2,因此可以通过式11计算K。

[0203] 式9

[0204] $K = K_{4 \times 4}$

[0205] 式10

[0206] $K = \{K_{4 \times 4} \gg (4-a)\} + b$

[0207] 式11

[0208] $K = \{K_{4 \times 4} \gg (a-2)\} + b$

[0209] 表22示出了K被限制为6的情况。

[0210] 表22

[0211]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	X	X
par_level_flag	1	1	X	X
rem_abs_gt1_flag	1	1	X	X
rem_abs_gt2_flag	0	1	X	X
abs_remainder		0	7	10
coeff_sign_flag	0	0	1	0

[0212] 在实施方式中,可以组合用于限制sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和的方法和上述用于限制rem_abs_gt2_flag的数量的方法。假设sig_coeff_flag的数量、par_level_flag的数量和rem_abs_gt1_flag的数量之和被限制为K并且rem_abs_gt2_flag的数量被限制为N,则K可以具有0到12的值,N可以具有为0到4的值。

[0213] K和N也可以由编码器确定,并且可以基于关于式6至式11的前述方法来计算。

[0214] 表23示出了K限制为6并且N限制为1的情况。

[0215] 表23

[0216]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	X	X
par_level_flag	1	1	X	X
rem_abs_gt1_flag	1	1	X	X
rem_abs_gt2_flag	0	X	X	X
abs_remainder		1	7	10
coeff_sign_flag	0	0	1	0

[0217] 实施方式可以提出一种用于改变par_level_flag和rem_abs_gt1_flag的编码顺序的方法。更具体地,在本实施方式中,当对色度块的具有 2×2 尺寸的子块进行编码时,可以按照rem_abs_gt1_flag和par_level_flag的顺序执行编码,而不是按照par_level_flag和rem_abs_gt1_flag的顺序进行编码。当par_level_flag和rem_abs_gt1_flag的编码顺序

更改时,在sig_coeff_flag之后对rem_abs_gt1_flag进行编码,并且可以只有在rem_abs_gt1_flag为1时,才对par_level_flag进行编码。因此,实际变换系数值(coeff)与各个语法元素之间的关系可以改变,如下式12所表示。

[0218] 式12

[0219] $|coeff| = sig_coeff_flag + rem_abs_gt1_flag + par_level_flag + 2 * (rem_abs_gt2_flag + abs_remainder)$

[0220] 下表24示出了与式12有关的一些示例。当与表2进行比较时,根据表24,如果|coeff|为1,则不对par_level_flag进行编码,因此就吞吐量和编码而言可以是有利的。当然,如果|coeff|为2,则需要对rem_abs_gt2_flag进行编码,这与表2不同,并且如果|coeff|为4,则需要对abs_remainder进行编码,这与表2不同,但是由于|coeff|为1的情况比|coeff|为2或4的情况发生更多,所以根据表24的方法可能比根据表2的方法展现出更高的吞吐量和编码性能。表25示出了如图6所示的对4×4子块进行编码的结果。

[0221] 表24

[0222]

coeff	sig_coeff_flag	rem_abs_gt1_flag	par_level_flag	rem_abs_gt2_flag	abs_remainder
0	0				
1	1	0			
2	1	1	0	0	
3	1	1	1	0	
4	1	1	0	1	0
5	1	1	1	1	0
6	1	1	0	1	1
7	1	1	1	1	1
8	1	1	0	1	2
9	1	1	1	1	2
10	1	1	0	1	3
11	1	1	1	1	3
...	...	...	...	...	...

[0223] 表25

[0224]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	1	1
rem_abs_gt1_flag	1	1	1	1
par_level_flag	0	0	1	0
rem_abs_gt2_flag	1	1	1	1
abs_remainder	0	1	1	3
coeff_sign_flag	0	0	1	0

[0225] 实施方式可以提出一种改变par_level_flag和rem_abs_gt1_flag的编码顺序并限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和的方法。例如,当以sig_coeff_flag、rem_abs_gt1_flag、par_level_flag、rem_abs_gt2_flag、abs_remainder和coeff_sign_flag的顺序执行编码时,可以限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和。假设sig_coeff_flag的数量、

rem_abs_gt1_flag的数量和par_level_flag的数量之和被限制为K,则K可以具有0至12的值。K可以由编码器选择,并且可以设置为0到12的任意值。此外,可以基于上述关于式9至式11的描述来计算K。

[0226] 在实施方式中,当不再对sig_coeff_flag、rem_abs_gt1_flag和par_level_flag进行编码时,也可以不对rem_abs_gt2_flag进行编码。表26示出了K被限制为6的情况。

[0227] 表26

[0228]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	X	X
rem_abs_gt1_flag	1	1	X	X
par_level_flag	0	0	X	X
rem_abs_gt2_flag	1	1	X	X
abs_remainder	0	1	7	10
coeff_sign_flag	0	0	1	0

[0229] 在实施方式中,可以在一个for循环中对sig_coeff_flag、rem_abs_gt1_flag和par_level_flag进行编码。尽管三个语法元素(sig_coeff_flag、rem_abs_gt1_flag和par_level_flag)的数量之和不超过K并且该和与K不完全匹配,但是可以在相同的扫描位置停止编码。表27示出了K被限制为8的情况。当执行编码直到扫描位置2(scan_pos 2)时,sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和为6。该和是不超过K的值,但是此时,由于编码器不知道下一扫描位置1(scan_pos 1)的系数级别的值,因此编码器不知道在scan_pos 1中生成的上下文编码的bin的数量是1到3中的哪个值。因此,编码器最多仅可以编码到scan_pos 2并终止编码。因此,K值不同,但是在表26和表27中编码结果可以相同。

[0230] 表27

[0231]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	X	X
rem_abs_gt1_flag	1	1	X	X
par_level_flag	0	0	X	X
rem_abs_gt2_flag	1	1	X	X
abs_remainder	0	1	7	10
coeff_sign_flag	0	0	1	0

[0232] 实施方式可以提出一种用于改变par_level_flag和rem_abs_gt1_flag的编码顺序并限制rem_abs_gt2_flag的数量的方法。例如,当以sig_coeff_flag、rem_abs_gt1_flag、par_level_flag、rem_abs_gt2_flag、abs_remainder和coeff_sign_flag的顺序执行编码时,可以限制用rem_abs_gt2_flag编码的系数的数量。

[0233] 在实施方式中,在 2×2 块内被编码的rem_abs_gt2_flag的最大数量是4。即,可以针对绝对值大于2的所有系数,对rem_abs_gt2_flag进行编码。此外,另一实施方式可以提出用于仅针对根据扫描顺序的前N个绝对值大于2的系数(即,rem_abs_gt1_flag为1的系

数)对rem_abs_gt2_flag进行编码的方法。

[0234] N也可以由编码器来选择,并且也可以设置为0至4中的任意值。此外,可以通过上述关于式6至式8的方法来计算N。

[0235] 表28示出了当N为1时本实施方式的应用示例。在 4×4 块中,rem_abs_gt2_flag的编码次数可以减少多达由X所指示的次数,从而减少上下文编码的bin的数量。当与表25比较时,关于rem_abs_gt2_flag不被编码的扫描位置,系数的abs_remainder值可以更改,如下表28所表示的。

[0236] 表28

[0237]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	1	1
rem_abs_gt1_flag	1	1	1	1
par_level_flag	0	0	1	0
rem_abs_gt2_flag	1	X	X	X
abs_remainder	0	2	2	4
coeff_sign_flag	0	0	1	0

[0238] 实施方式可以提供一种用于改变par_level_flag和rem_abs_gt1_flag的编码顺序并分别限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和以及rem_abs_gt2_flag的数量的方法。例如,当以sig_coeff_flag、rem_abs_gt1_flag、par_level_flag、rem_abs_gt2_flag、abs_remainder和coeff_sign_flag的顺序执行编码时,也可以组合用于限制sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和的方法以及用于限制rem_abs_gt2_flag的数量的方法。假设sig_coeff_flag的数量、rem_abs_gt1_flag的数量和par_level_flag的数量之和被限制为K并且rem_abs_gt2_flag的数量被限制为N,则K可以具有0到12的值,并且N可以具有0到4的值。K和N也可以由编码器选择,并且K可以设置为0到12的任意值,N可以设置为0到4的任意值。此外,可以通过上述关于式6至式11的方法来计算K和N。

[0239] 下表29示出了K被限制为6并且N被限制为1的情况。

[0240] 表29

[0241]

scan_pos	3	2	1	0
系数	4	6	-7	10
sig_coeff_flag	1	1	X	X
rem_abs_gt1_flag	1	1	X	X
par_level_flag	0	0	X	X
rem_abs_gt2_flag	1	X	X	X
abs_remainder	0	2	7	10
coeff_sign_flag	0	0	1	0

[0242] 图7是例示了根据实施方式的编码设备的操作的流程图,并且图8是例示了根据实施方式的编码设备的配置的框图。

[0243] 根据图7和图8的编码设备可以执行与根据图9和图10的解码设备相对应的操作。

因此,稍后将参照图9和图10描述的解码设备的操作也可以以相同方式应用于根据图7和图8的编码设备。

[0244] 图7所示的各步骤可以由图2所示的编码设备200来执行。更具体地说,可以由图2所示的减法器231执行S700,可以由图2所示的量化器233执行S710,并且可以由图2所示的熵编码器240执行S720。此外,根据S700至S720的操作是基于参照图4至图6的前述描述中的一些操作的。因此,将省略或简化与以上参照图2和图4至图6所描述的内容交叠的详细描述。

[0245] 如图8所示,根据实施方式的编码设备可以包括减法器231、变换器232、量化器233和熵编码器240。然而,在一些情况下,图8所示的全部组件可以不是编码设备的必要组件,并且编码设备可以由比图8所示的组件更多或更少的组件来实现。

[0246] 在根据实施方式的编码设备中,减法器231、变换器232、量化器233和熵编码器240可以分别实现为单独的码片,或者也可以通过单个芯片实现至少两个组件。

[0247] 根据实施方式的编码设备可以推导当前块的残差样本(S700)。更具体地,编码设备的减法器231可以推导当前块的残差样本。

[0248] 根据实施方式的编码设备可以基于当前块的残差样本来推导量化变换系数(S710)。更具体地,编码设备的量化器233可以基于当前块的残差样本来推导量化变换系数。

[0249] 根据实施方式的编码设备可以对包括关于量化变换系数的信息的残差信息进行编码(S720)。更具体地,编码设备的熵编码器240可以对包括关于量化变换系数的信息的残差信息进行编码。

[0250] 在实施方式中,残差信息可以包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志。在示例中,奇偶校验级别标志可以表示`par_level_flag`,第一变换系数级别标志可以表示`rem_abs_gt1_flag`或`abs_level_gtx_flag[n][0]`,并且第二变换系数级别标志可以表示`rem_abs_gt2_flag`或`abs_level_gtx_flag[n][1]`。

[0251] 在实施方式中,对残差信息进行编码可以包括:基于量化变换系数推导奇偶校验级别标志的值和第一变换系数级别标志的值;以及对第一变换系数级别标志进行编码并且对奇偶校验级别标志进行编码。

[0252] 在实施方式中,可以在对奇偶校验级别标志进行编码之前执行对第一变换系数级别标志的编码。例如,编码设备可以在对`par_level_flag`进行编码之前,对`rem_abs_gt1_flag`或`abs_level_gtx_flag[n][0]`执行编码。

[0253] 在实施方式中,残差信息还可以包括表示量化变换系数是否为非零有效系数的有效系数标志和关于量化变换系数的变换系数级别是否大于第二阈值的第二变换系数级别标志。在示例中,有效系数标志可以表示`sig_coeff_flag`。

[0254] 在实施方式中,残差信息中所包括的当前块内的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和可以是预定阈值或更小。在示例中,用于与当前块内的当前子块有关的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和可以被限制为预定阈值或更小。

[0255] 在实施方式中,可以基于当前块(或当前块内的当前子块)的尺寸来确定预定阈值。

[0256] 在实施方式中,残差信息中所包括的有效系数标志的数量、第一变换系数级别标志的数量和奇偶校验级别标志的数量之和为第三阈值或更小,残差信息中包括的第二变换系数级别标志的数量是第四阈值或更小,并且预定阈值可以表示第三阈值和第四阈值之和。

[0257] 在示例中,如果当前块或当前块内的当前子块的尺寸为 4×4 ,则第三阈值可以表示K,并且此时,K可以表示0至48中的一个值。此外,第四阈值可以表示N,并且此时,N可以表示0至16中的一个值。

[0258] 在另一示例中,如果当前块或当前块内的当前子块的尺寸为 2×2 ,则第三阈值可以表示K,并且此时,K可以表示0至12中的一个值。此外,第四阈值可以表示N,并且此时,N可以表示0至4中的一个值。

[0259] 在实施方式中,如果基于按照系数扫描顺序确定的第0量化变换系数到第n量化变换系数而推导的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和达到预定阈值,则针对按照系数扫描顺序确定的第(n+1)量化变换系数,省略有效系数标志、第一变换系数级别标志、奇偶校验级别标志和第二变换系数级别标志的显式信令,并且可以基于残差信息中所包括的系数级别信息的值来推导第(n+1)量化变换系数的值。

[0260] 例如,如果基于按照系数扫描顺序确定的第0量化变换系数(或第一量化变换系数)至第n量化变换系数(或第n量化变换系数)而推导的sig_coeff_flag的数量、rem_abs_gt1_flag(或abs_level_gtx_flag[n][0])的数量、par_level_flag的数量和rem_abs_gt2_flag(或abs_level_gtx_flag[n][1])的数量之和达到预定阈值,则针对按照系数扫描顺序确定的第(n+1)量化变换系数,省略sig_coeff_flag、rem_abs_gt1_flag(或abs_level_gtx_flag[n][0])、par_level_flag、abs_level_gtx_flag[n][1]和rem_abs_gt2_flag(或abs_level_gtx_flag[n][1])的显式信令,并且可以基于残差信息中所包括的abs_remainder或dec_abs_level的值来推导第(n+1)量化变换系数的值。

[0261] 在实施方式中,残差信息中所包括的有效系数标志、第一变换系数级别标志、奇偶校验级别标志和第二变换系数级别标志是基于上下文被编码的,并且系数级别信息可以以旁路为基础被编码。

[0262] 根据图7和图8所示的编码设备和编码设备的操作方法,编码设备推导当前块的残差样本(S700),基于当前块的残差样本来推导量化变换系数(S710),并且对包括关于量化变换系数的信息的残差信息进行编码(S720);并且残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志,并且残差信息的编码包括基于量化变换系数来推导奇偶校验级别标志的值和第一变换系数级别标志的值,以及对第一变换系数级别标志进行编码并且对奇偶校验级别标志进行编码,并且在奇偶校验级别标志进行编码之前执行对第一变换系数级别标志的编码。即,根据本公开,通过确定(或改变)用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志的解码顺序,可以提高编码效率。

[0263] 图9是例示了根据实施方式的解码设备的操作的流程图,并且图10是例示了根据实施方式的解码设备的配置的框图。

[0264] 图9所示的各步骤可以由图3所示的解码设备300来执行。更具体地,可以由图3所示的熵解码器310执行S900和S910,可以由图3所示的反量化器321和/或逆变换器322执行S920,并且可以由图3中所示的加法器340执行S930。此外,根据S900至S930的操作是基于参照图4至图6的前述描述的一部分。因此,将省略或简化与以上参照图3至图6描述的内容交叠的详细描述。

[0265] 如图10所示,根据实施方式的解码设备可以包括熵解码器310、反量化器321、逆变换器322和加法器340。然而,在一些情况下,图10所示的所有组件可以不是解码设备的必要组件,并且解码设备可以由比图10所示的组件更多或更少的组件来实现。

[0266] 在根据实施方式的解码设备中,熵解码器310、反量化器321、逆变换器322和加法器340分别实现为单独芯片,或者也可以通过单个芯片实现至少两个组件。

[0267] 根据实施方式的解码设备可以接收包括残差信息的比特流(S900)。更具体地,解码设备的熵解码器310可以接收包括残差信息的比特流。

[0268] 根据实施方式的解码设备可以基于比特流中包括的残差信息来推导针对当前块的量化变换系数(S910)。更具体地,解码设备的熵解码器310可以基于比特流中所包括的残差信息来推导当前块的量化变换系数。

[0269] 根据实施方式的解码设备可以基于量化变换系数来推导当前块的残差样本(S920)。更具体地,解码设备的反量化器321可以基于反量化处理从量化变换系数中推导变换系数,并且解码设备的逆变换器322可以通过对变换系数进行逆变换来推导当前块的残差样本。

[0270] 根据实施方式的解码设备可以基于当前块的残差样本来生成重构图片(S930)。更具体地,解码设备的加法器340可以基于当前块的残差样本来生成重构图片。

[0271] 在实施方式中,残差信息可以包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志。在示例中,奇偶校验级别标志可以表示`par_level_flag`,第一变换系数级别标志可以表示`rem_abs_gt1_flag`或`abs_level_gtx_flag[n][0]`,并且第二变换系数级别标志可以表示`rem_abs_gt2_flag`或`abs_level_gtx_flag[n][1]`。

[0272] 在实施方式中,量化变换系数的推导可以包括:对变换系数级别标志进行解码和对奇偶校验级别标志进行解码;以及基于经解码的奇偶校验级别标志的值和经解码的第一变换系数级别标志的值来推导量化变换系数。

[0273] 在实施方式中,可以在对奇偶校验级别标志进行解码之前执行对第一变换系数级别标志的解码。例如,解码设备可以在对`par_level_flag`进行解码之前,执行对`rem_abs_gt1_flag`或`abs_level_gtx_flag[n][0]`的解码。

[0274] 在实施方式中,残差信息还可以包括表示量化变换系数是否非零有效系数的有效系数标志和关于量化变换系数的变换系数级别是否大于第二阈值的第二变换系数级别标志。在示例中,有效系数标志可以表示`sig_coeff_flag`。

[0275] 在实施方式中,残差信息中所包括的当前块内的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志

的数量之和可以为预定阈值或更小。在示例中,与当前块内的当前子块有关的量化变换系数的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和可以被限制为预定阈值或更小。

[0276] 在实施方式中,可以基于当前块(或者当前块内的当前子块)的尺寸来确定预定阈值。

[0277] 在实施方式中,残差信息中所包括的有效系数标志的数量、第一变换系数级别标志的数量和奇偶校验级别标志的数量之和为第三阈值或更小,残差信息中所包括的第二变换系数级别标志的数量为第四阈值或更小,并且预定阈值可以表示第三阈值和第四阈值之和。

[0278] 在示例中,如果当前块或当前块内的当前子块的尺寸为 4×4 ,则第三阈值可以表示K,并且此时,K可以表示0至48中的一个值。此外,第四阈值可以表示N,并且此时,N可以表示0至16中的一个值。

[0279] 在另一示例中,如果当前块或当前块内的当前子块的尺寸为 2×2 ,则第三阈值可以表示K,并且此时,K可以表示0至12中的一个值。此外,第四阈值可以表示N,并且此时,N可以表示0至4中的一个值。

[0280] 在实施方式中,如果基于按照系数扫描顺序确定的第0量化变换系数到第n量化变换系数而推导的有效系数标志的数量、第一变换系数级别标志的数量、奇偶校验级别标志的数量和第二变换系数级别标志的数量之和达到预定阈值,则针对按照系数扫描顺序确定的第(n+1)量化变换系数,省略有效系数标志、第一变换系数级别标志、奇偶校验级别标志和第二变换系数级别标志的显式信令,并且可以基于残差信息中所包括的系数级别信息的值来推导第(n+1)量化变换系数的值。

[0281] 例如,如果按照系数扫描顺序确定的第0量化变换系数(或第一量化变换系数)至第n量化变换系数(或第n量化变换系数)而推导的sig_coeff_flag的数量、rem_abs_gt1_flag(或abs_level_gtx_flag[n][0])的数量、par_level_flag的数量和rem_abs_gt2_flag(或abs_level_gtx_flag[n][1])的数量之和达到预定阈值,则针对按照系数扫描顺序确定的第(n+1)量化变换系数,省略sig_coeff_flag、rem_abs_gt1_flag(或abs_level_gtx_flag[n][0])、par_level_flag、abs_level_gtx_flag[n][1]和rem_abs_gt2_flag(或abs_level_gtx_flag[n][1])的显式信令,并且可以基于残差信息中所包括的abs_remainder或dec_abs_level的值来推导第(n+1)量化变换系数的值。

[0282] 在实施方式中,残差信息中所包括的有效系数标志、第一变换系数级别标志、奇偶校验级别标志和第二变换系数级别标志是基于上下文被编码的,并且系数级别信息可以以旁路为基础被编码。

[0283] 根据图9和图10中公开的解码设备和解码设备的操作方法,特征可以在于,解码设备接收包括残差信息的比特流(S900),基于比特流中包括的残差信息来推导当前块的量化变换系数(S910),基于量化变换系数推导当前块的残差样本(S920),并基于当前块的残差样本生成重构图片(S930);并且残差信息包括用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志,量化变换系数的推导包括对变换系数级别标志进行解码并对奇偶校验级别标志进行解码,以及基于经解码的奇偶校验级别标志的值和经解码的第一变换系数级别标志的值推导量

化变换系数,并且在奇偶校验级别标志进行解码之前执行对第一变换系数级别标志的解码。即,根据本公开,通过确定(或改变)用于量化变换系数的变换系数级别的奇偶校验的奇偶校验级别标志和关于变换系数级别是否大于第一阈值的第一变换系数级别标志的解码顺序,可以提高编码效率。

[0284] 在前述实施方式中,尽管基于被示出为一系列步骤或方框的流程图描述了方法,但是本公开对步骤的顺序没有限制,并且一些步骤可以以与以上描述的顺序不同的顺序发生或与以上描述的另一步骤同时发生。此外,本领域技术人员将理解,流程图中所示的步骤不是排他的,并且可以包括其他步骤,或者可以在不影响本公开范围的情况下删除流程图中的一个或更多个步骤。

[0285] 根据本公开的前述方法可以以软件的形式实现,并且根据本公开的编码设备和/或解码设备可以被包括在例如TV、计算机、智能手机、机顶盒、显示装置等中的用于执行图像处理设备中。

[0286] 当本公开中的实施方式以软件实现时,上述方法可以被实现为用于执行上述功能的模块(处理、功能等)。该模块可以存储在存储器中,并由处理器执行。存储器可以位于处理器内部或外部,并且可以通过各种已知方式与处理器联接。处理器可以包括专用集成电路(ASIC)、其他芯片组、逻辑电路和/或数据处理装置。存储器可以包括只读存储器(ROM)、随机存取存储器(RAM)、闪存、存储卡、存储介质和/或其他存储装置。即,本公开中描述的实施方式可以通过在处理器、微处理器、控制器或芯片上实现来执行。例如,每个附图中所示出的功能单元可以通过在计算机、处理器、微处理器、控制器或芯片上实现来执行。在这种情况下,用于实现的信息(例如,关于指令的信息)或算法可以存储在数字存储介质中。

[0287] 此外,应用了本公开的解码设备和编码设备可以被包括在多媒体广播收发器、移动通信终端、家庭影院视频装置、数字影院视频装置、监视相机、视频通信装置、诸如视频通信之类的实时通信装置、移动流传输装置、存储介质、便携式摄像机、视频点播(VoD)服务提供者、顶置视频(OTT video)装置、互联网流服务提供者、三维(3D)视频装置、虚拟现实(VR)装置、增强现实(AR)装置、视频电话视频装置、运输终端(例如,车辆(包括自动驾驶车辆)、飞机终端、轮船终端等)以及医疗视频装置等中,并且可以用于处理视频信号或数据信号。例如,OTT视频装置可以包括游戏机、蓝光播放器、连接互联网的电视、家庭影院系统、智能电话、平板PC、数字视频记录仪(DVR)等。

[0288] 此外,应用本公开的处理方法可以以计算机执行的程序的形式产生,并且可以存储在计算机可读记录介质中。根据本公开的具有数据结构的多媒体数据也可以存储在计算机可读记录介质中。计算机可读记录介质包括其中存储计算机可读数据的所有类型的存储装置和分布式存储装置。计算机可读记录介质可以包括例如蓝光盘(BD)、通用串行总线(USB)、ROM、PROM、EPROM、EEPROM、RAM、CD-ROM、磁带、软盘和光学数据存储装置。此外,计算机可读记录介质包括以载波(例如,经由互联网的传输)形式实现的介质。此外,通过编码方法生成的比特流可以存储在计算机可读记录介质中或通过有线/无线通信网络来传输。

[0289] 此外,本公开的实施方式可以通过程序代码实现为计算机程序产品,并且根据本公开实施方式可以在计算机上执行程序代码。程序代码可以存储在计算机可读载体上。

[0290] 图11例示了可以应用本文件中公开的公开内容的上下文流系统的示例。

[0291] 参照图11,应用了本公开的内容流系统可以主要包括编码服务器、流服务器、网络

服务器、媒体存储器、用户装置和多媒体输入装置。

[0292] 编码服务器用于将从诸如智能手机、相机和便携式摄像机之类的多媒体输入装置输入的内容压缩为数字数据,以生成比特流并向流服务器发送比特流。作为另一示例,当诸如智能电话、相机和便携式摄像机之类的多媒体输入装置直接生成比特流时,可以省略编码服务器。

[0293] 可以通过应用本公开的编码方法或比特流生成方法来生成比特流,并且流服务器可以在发送或接收比特流的处理中临时存储比特流。

[0294] 流服务器用作通过网络服务器基于用户请求向用户装置发送多媒体数据的媒介,并且网络服务器用作向用户通知哪些服务可用的媒介。如果用户从网络服务器请求所需的服务,则网络服务器向流服务器传递请求,并且流服务器向用户发送多媒体数据。此时,内容流系统可以包括单独的控制服务器,并且在这种情况下,控制服务器起到控制内容流系统内的装置之间的命令/响应的作用。

[0295] 流服务器可以从媒体存储器和/或编码服务器接收内容。例如,如果从编码服务器接收内容,则可以实时接收内容。在这种情况下,为了提供平稳的流服务,流服务器可以将比特流存储一定时间。

[0296] 用户装置的示例可以包括移动电话、智能手机、膝上型计算机、数字广播终端、个人数字助理(PDA)、便携式多媒体播放器(PMP)、导航终端、触屏PC、平板PC、超级本、可穿戴装置(例如,智能手表、智能眼镜或头戴式显示器(HMD))、数字TV、台式计算机、数字标牌等。

[0297] 内容流系统中的每个服务器可以作为分布式服务器来操作,并且在这种情况下,可以以分布式方式处理每个服务器接收的数据。

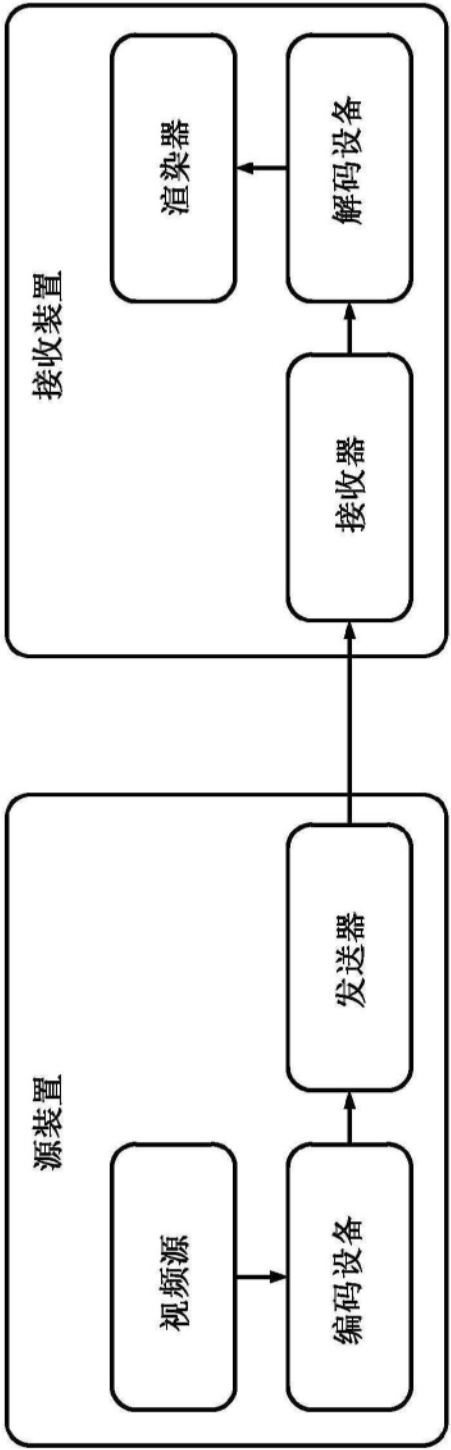


图1

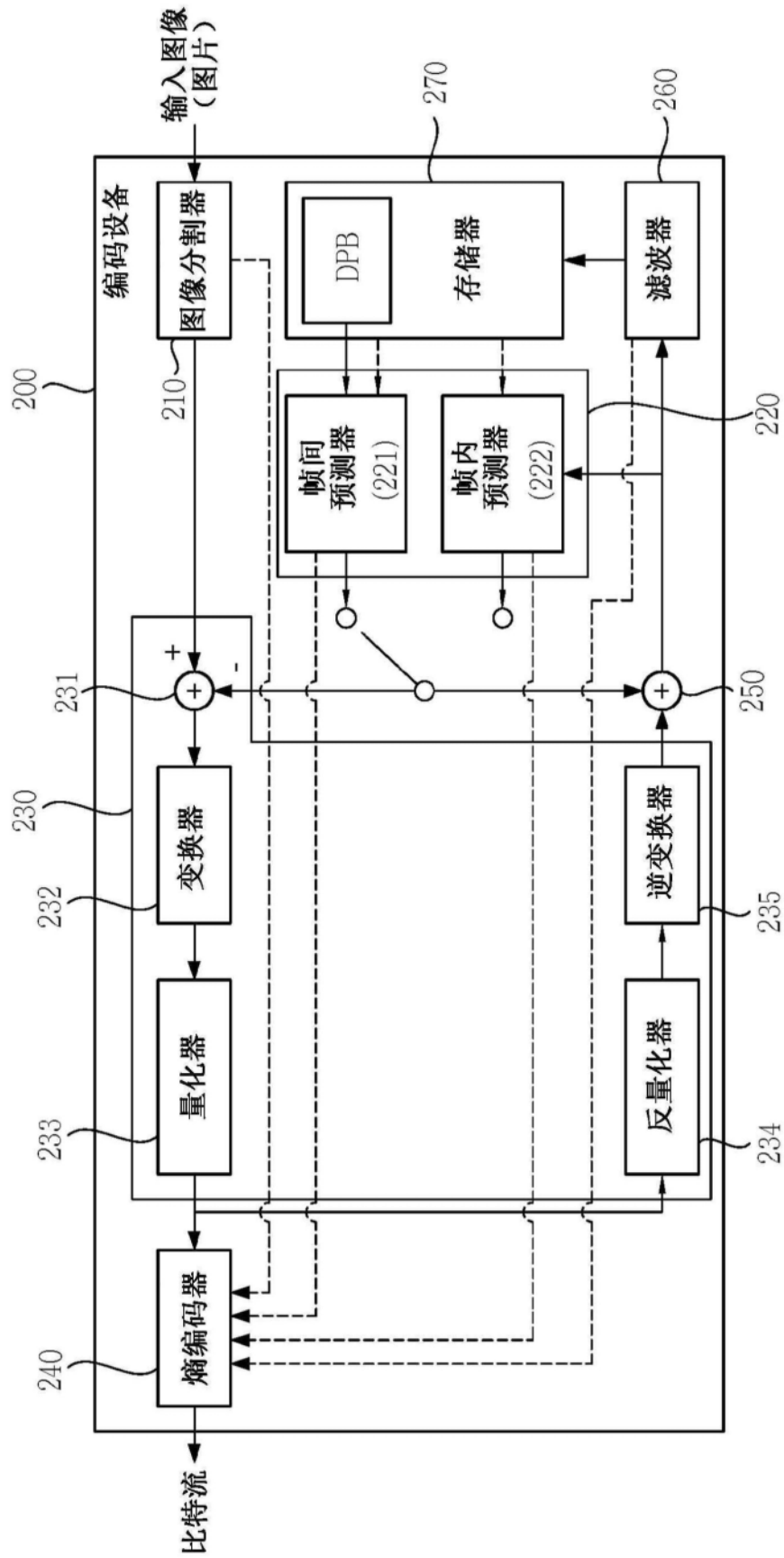


图2

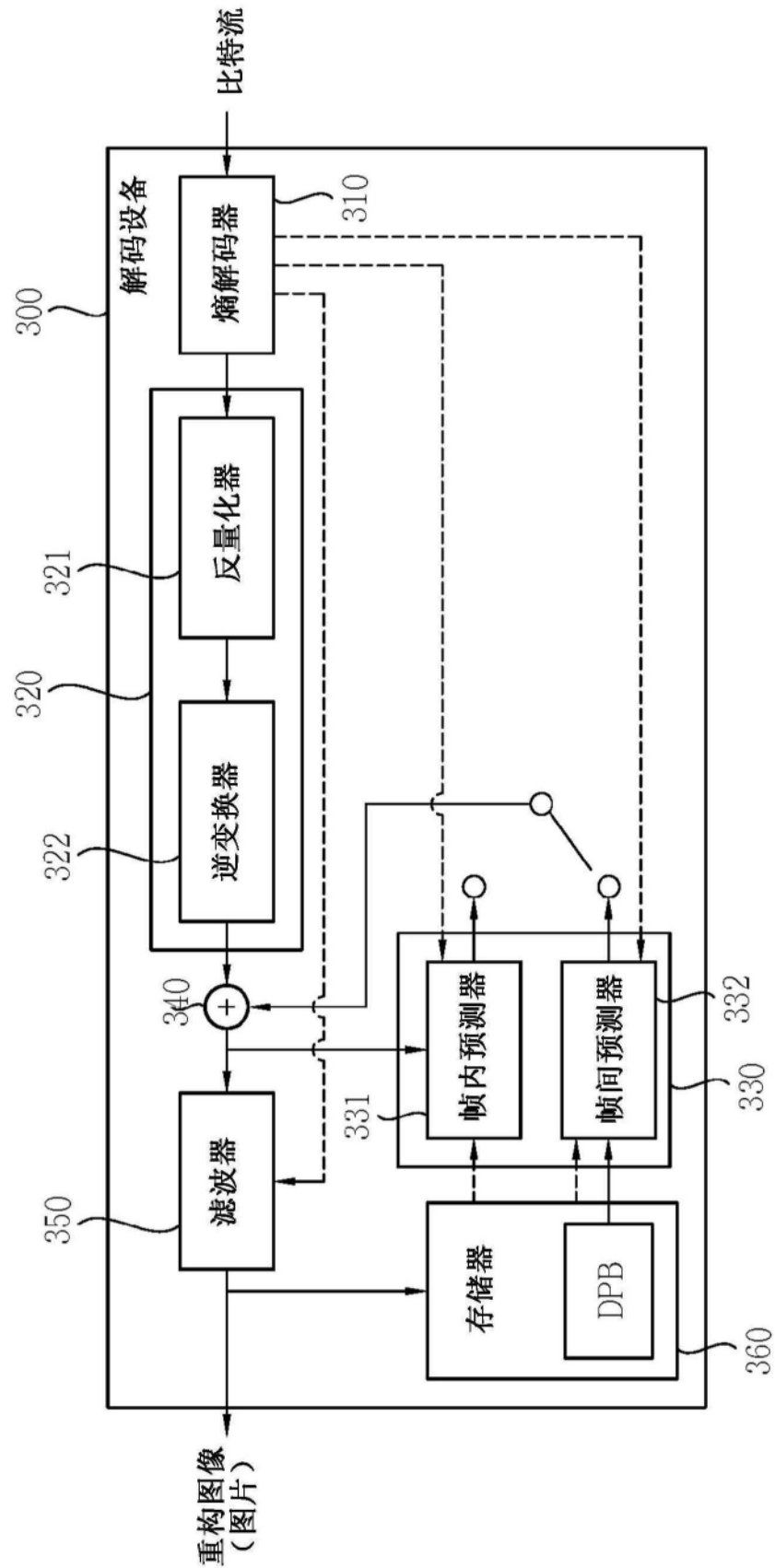


图3

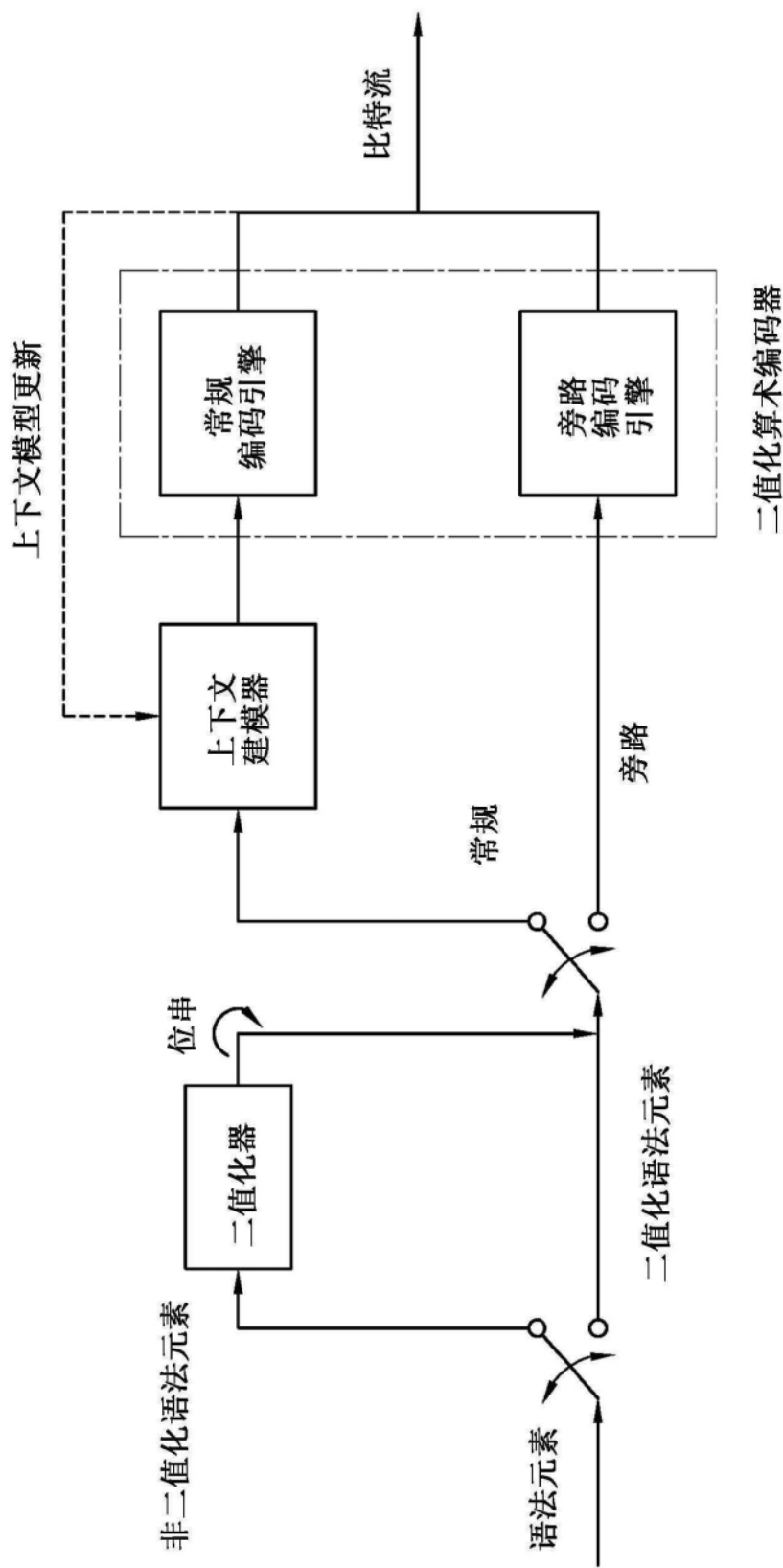


图4

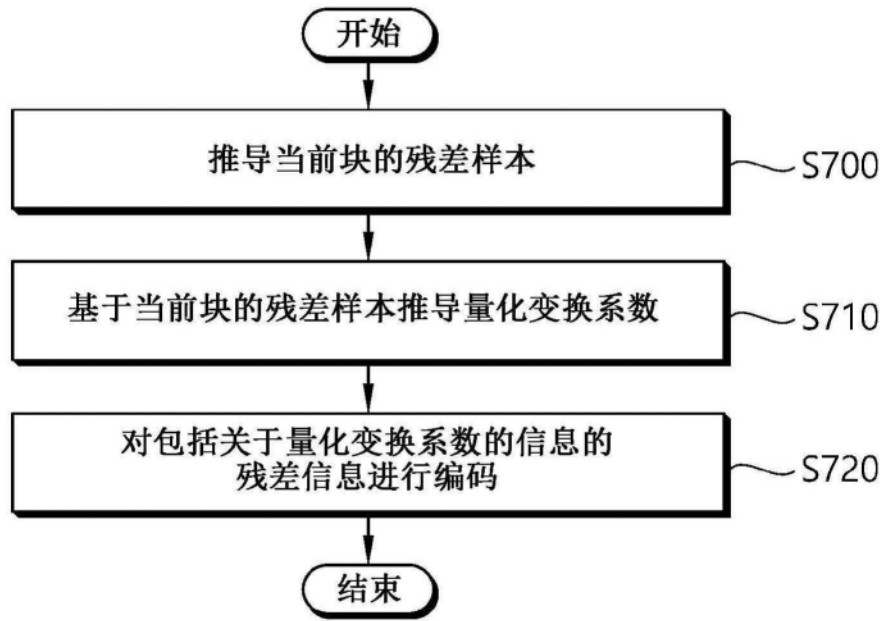


图7

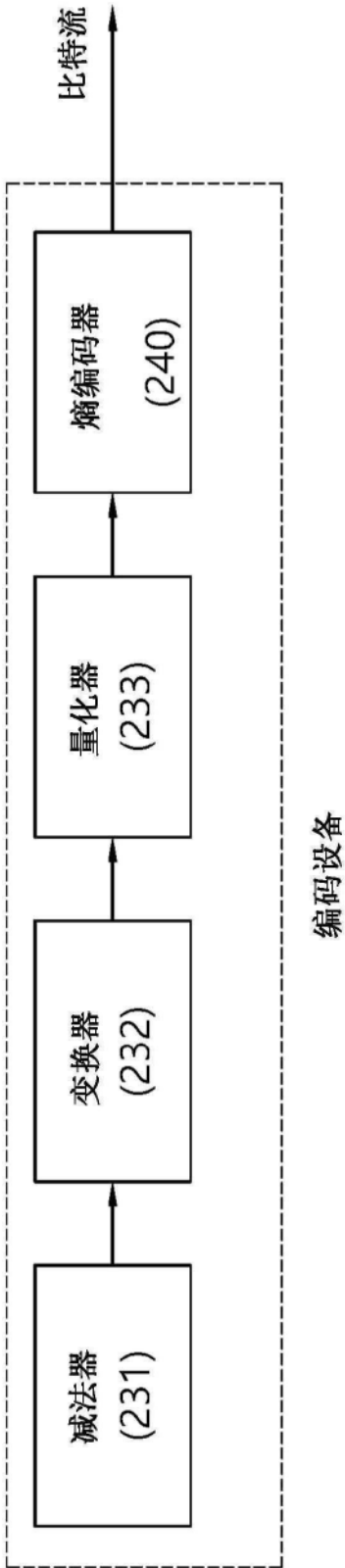


图8

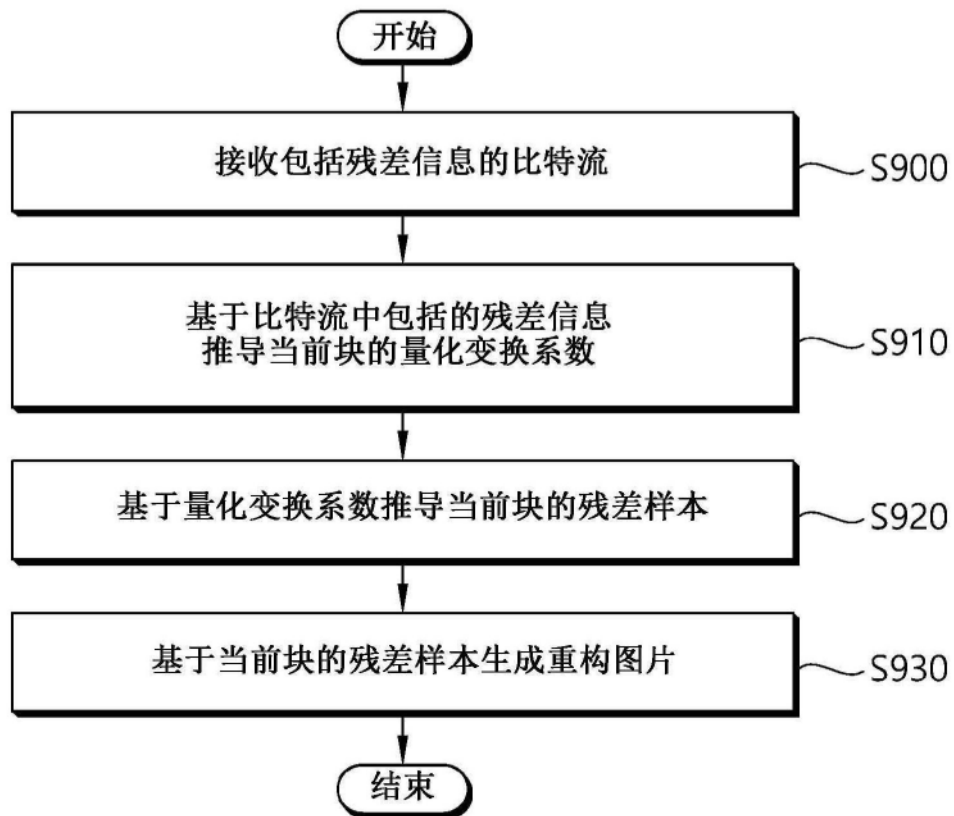


图9

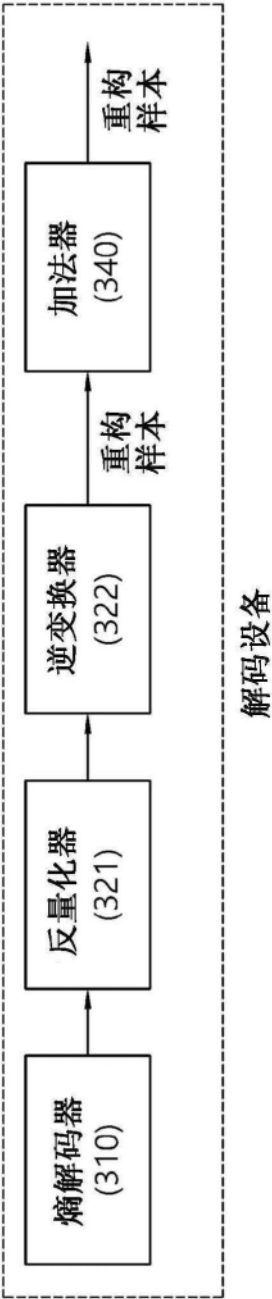


图10

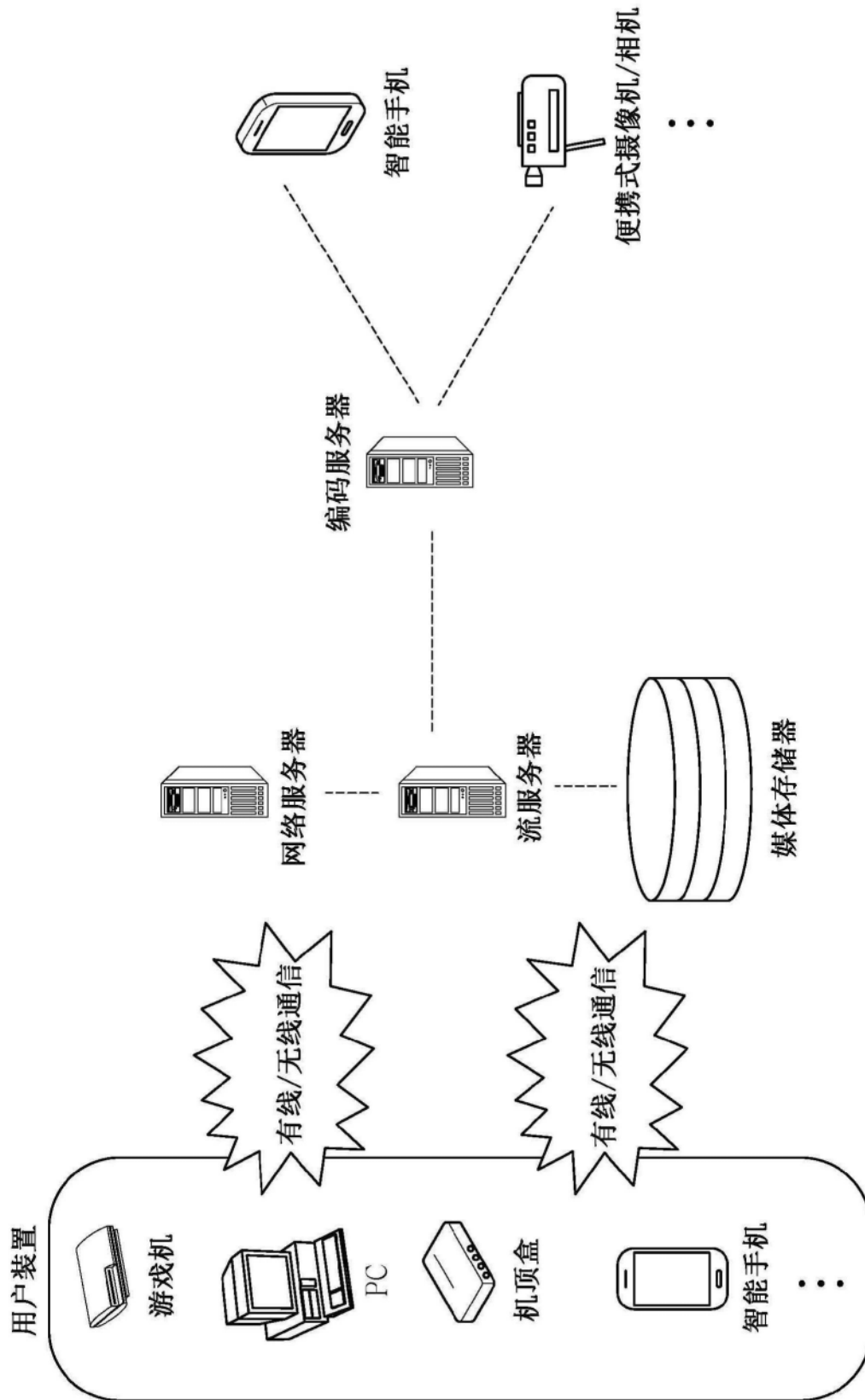


图11