



(12) 发明专利

(10) 授权公告号 CN 112840664 B

(45) 授权公告日 2024. 07. 05

(21) 申请号 201980067187.8

(22) 申请日 2019.10.18

(65) 同一申请的已公布的文献号
申请公布号 CN 112840664 A

(43) 申请公布日 2021.05.25

(30) 优先权数据
62/747,543 2018.10.18 US
62/786,285 2018.12.28 US
62/788,546 2019.01.04 US
62/792,795 2019.01.15 US
16/655,950 2019.10.17 US

(85) PCT国际申请进入国家阶段日
2021.04.12

(86) PCT国际申请的申请数据
PCT/US2019/056966 2019.10.18

(87) PCT国际申请的公布数据
W02020/081951 EN 2020.04.23

(73) 专利权人 高通股份有限公司
地址 美国加利福尼亚州

(72) 发明人 M.卡尔切维茨 M.Z.科班

(74) 专利代理机构 北京市柳沈律师事务所
11105
专利代理师 安之斐

(51) Int.Cl.
H04N 19/70 (2006.01)
H04N 19/122 (2006.01)
H04N 19/132 (2006.01)
H04N 19/176 (2006.01)
H04N 19/18 (2006.01)
H04N 19/91 (2006.01)

(56) 对比文件
Moonmo Koo 等.CE 6-1.1 (c,d): Fast DST-7/DCT-8 based on DFT and 32 point MTS based on skipping high frequency coefficients.Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, 3-12 Oct. 2018.2018,第1.1节.

审查员 黄海云

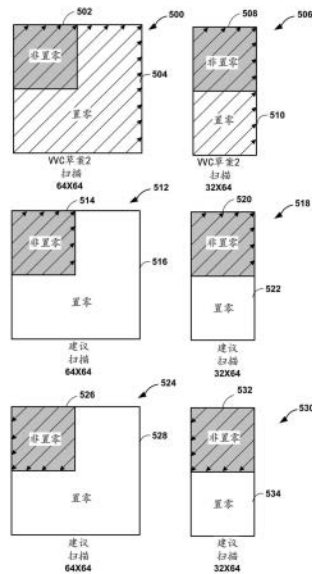
权利要求书3页 说明书21页 附图7页

(54) 发明名称

用于置零变换的扫描和最后系数位置译码

(57) 摘要

一种对视频数据进行解码的方法包括:确定变换单元中变换系数经历置零的第一区域,确定所述变换单元中变换系数未经历置零的第二区域,以及仅扫描所述变换单元的第二区域。



1. 一种对视频数据进行解码的方法,所述方法包括:

基于用于执行置零过程的阈值,对表示X方向上最后非零系数的位置的第一部分的第一前缀码进行解码,其中所述阈值是32,以及其中解码所述第一前缀码包括当基于用于执行所述置零过程的所述阈值来设置所述X方向上最后非零系数的位置时,基于24-31的范围来解码所述第一前缀码;

对表示所述X方向上最后非零系数的位置的第二部分的第一后缀码进行解码;

基于所述用于执行置零过程的阈值,对表示Y方向上最后非零系数的位置的第一部分的第二前缀码进行解码,其中解码所述第二前缀码包括当基于用于执行所述置零过程的所述阈值来设置所述Y方向上最后非零系数的位置时,基于24-31的范围来解码所述第二前缀码;

对表示所述Y方向上最后非零系数的位置的第二部分的第二后缀码进行解码;

基于所述阈值确定变换单元中变换系数经历置零的第一区域;

基于所述阈值确定所述变换单元中变换系数未经历置零的第二区域;以及

仅从所述最后非零系数的位置开始扫描所述变换单元的所述第二区域。

2. 根据权利要求1所述的方法,还包括:

基于所述变换单元的整个尺寸,来确定用于对指示最后非零系数的位置的一个或多个语法元素进行熵解码的上下文。

3. 根据权利要求1所述的方法,其中,基于所述变换系数在所述变换单元中的位置,来确定所述变换单元中变换系数经历置零的第一区域包括:

确定所述第一区域包括所述变换单元中超过所述变换单元的前32行或列的经历置零的变换系数。

4. 根据权利要求1所述的方法,其中,仅扫描所述变换单元的所述第二区域包括:

逐系数组地,仅扫描所述变换单元的所述第二区域。

5. 根据权利要求1所述的方法,其中,所述变换单元具有64个变换系数的宽度或高度,并且其中,所述第一区域是所述变换单元的32×32区域。

6. 根据权利要求1所述的方法,还包括:

对所述变换单元进行逆变换以生成残差数据;

确定预测块;

将所述残差数据加到所述预测块以对视频数据的块进行解码;以及

显示包括视频数据的经解码块的图片。

7. 根据权利要求1所述的方法,其中:

确定所述第一区域包括:基于所述X和Y方向上所述最后非零系数的位置来确定所述变换单元的所述第一区域,其中所述变换单元的最大维度为64。

8. 根据权利要求7所述的方法,其中:

为了解码所述X和Y方向上所述最后非零系数的位置,对应于所述用于执行置零过程的阈值设置所述变换单元的最大维度。

9. 一种配置为对视频数据进行解码的装置,所述装置包括:

存储器,配置为存储视频数据的变换单元;以及

一个或多个处理器,实现在电路中并且与所述存储器通信,所述一个或多个处理器被

配置为：

基于用于执行置零过程的阈值,对表示X方向上最后非零系数的位置的第一部分的第一前缀码进行解码,其中所述阈值是32,以及其中解码所述第一前缀码包括当基于用于执行所述置零过程的所述阈值来设置所述X方向上最后非零系数的位置时,基于24-31的范围来解码所述第一前缀码;

对表示所述X方向上最后非零系数的位置的第二部分的第一后缀码进行解码;

基于所述用于执行置零过程的阈值,对表示Y方向上最后非零系数的位置的第一部分的第二前缀码进行解码,其中解码所述第二前缀码包括当基于用于执行所述置零过程的所述阈值来设置所述Y方向上最后非零系数的位置时,基于24-31的范围来解码所述第二前缀码;

对表示所述Y方向上最后非零系数的位置的第二部分的第二后缀码进行解码;

基于所述阈值确定变换单元中变换系数经历置零的第一区域;

基于所述阈值确定所述变换单元中变换系数未经历置零的第二区域;以及

仅从所述最后非零系数的位置开始扫描所述变换单元的所述第二区域。

10. 根据权利要求9所述的装置,其中,所述一个或多个处理器还被配置为:

基于所述变换单元的整个尺寸,来确定用于对指示最后非零系数的位置的一个或多个语法元素进行熵解码的上下文。

11. 根据权利要求9所述的装置,其中,为了基于所述变换系数在所述变换单元中的位置,来确定所述变换单元中变换系数经历置零的第一区域,所述一个或多个处理器还被配置为:

确定所述第一区域包括所述变换单元中超过所述变换单元的前32行或列的经历置零的变换系数。

12. 根据权利要求9所述的装置,其中,为了仅扫描所述变换单元的所述第二区域,所述一个或多个处理器还被配置为:

逐系数组地,仅扫描所述变换单元的所述第二区域。

13. 根据权利要求9所述的装置,其中,所述变换单元具有64个变换系数的宽度或高度,并且其中,所述第一区域是所述变换单元的 32×32 区域。

14. 根据权利要求9所述的装置,其中,所述一个或多个处理器还被配置为:

对所述变换单元进行逆变换以生成残差数据;

确定预测块;

将所述残差数据加到所述预测块以对视频数据的块进行解码;以及

显示包括视频数据的经解码块的图片。

15. 根据权利要求9所述的装置,其中为了确定所述第一区域,所述一个或多个处理器还被配置为:

基于所述X和Y方向上所述最后非零系数的位置来确定所述变换单元的所述第一区域,其中所述变换单元的最大维度为64。

16. 根据权利要求15所述的装置,其中:

为了解码所述X和Y方向上所述最后非零系数的位置,对应于所述用于执行置零过程的阈值设置所述变换单元的最大维度。

17. 一种配置为对视频数据进行解码的装置,所述装置包括:

用于基于用于执行置零过程的阈值、对表示X方向上最后非零系数的位置的第一部分的第一前缀码进行解码的部件,其中所述阈值是32,以及其中解码所述第一前缀码包括当基于用于执行所述置零过程的所述阈值来设置所述X方向上最后非零系数的位置时,基于24-31的范围来解码所述第一前缀码;

用于对表示所述X方向上最后非零系数的位置的第二部分的第一后缀码进行解码的部件;

用于基于所述用于执行置零过程的阈值、对表示Y方向上最后非零系数的位置的第一部分的第二前缀码进行解码的部件,其中解码所述第二前缀码包括当基于用于执行所述置零过程的所述阈值来设置所述Y方向上最后非零系数的位置时,基于24-31的范围来解码所述第二前缀码;

用于对表示所述Y方向上最后非零系数的位置的第二部分的第二后缀码进行解码的部件;

用于基于所述阈值确定变换单元中变换系数经历置零的第一区域的部件;

用于基于所述阈值确定所述变换单元中变换系数未经历置零的第二区域的部件;以及用于仅从所述最后非零系数的位置开始扫描所述变换单元的所述第二区域的部件。

18. 根据权利要求17所述的装置,还包括:

用于基于所述变换单元的整个尺寸,来确定用于对指示最后非零系数的位置的一个或多个语法元素进行熵解码的上下文的部件。

19. 根据权利要求17所述的装置,其中,所述用于基于所述变换系数在所述变换单元中的位置,来确定所述变换单元中变换系数经历置零的第一区域的部件包括:

用于确定所述第一区域包括所述变换单元中超过所述变换单元的前32行或列的经历置零的变换系数的部件。

20. 根据权利要求17所述的装置,其中,所述用于仅扫描所述变换单元的所述第二区域的部件包括:

用于逐系数组地,仅扫描所述变换单元的所述第二区域的部件。

21. 根据权利要求17所述的装置,其中,所述变换单元具有64个变换系数的宽度或高度,并且其中,所述第一区域是所述变换单元的 32×32 区域。

22. 根据权利要求17所述的装置,还包括:

用于对所述变换单元进行逆变换以生成残差数据的部件;

用于确定预测块的部件;

用于将所述残差数据加到所述预测块以对视频数据的块进行解码的部件;以及

用于显示包括视频数据的经解码块的图片的部件。

23. 根据权利要求17所述的装置,其中所述用于确定第一区域的部件包括:

用于基于所述X和Y方向上所述最后非零系数的位置来确定所述变换单元的所述第一区域的部件,其中所述变换单元的最大维度为64。

24. 根据权利要求23所述的装置,其中:

为了解码所述X和Y方向上所述最后非零系数的位置,对应于所述用于执行置零过程的阈值设置所述变换单元的最大维度。

用于置零变换的扫描和最后系数位置译码

[0001] 本申请要求于2019年10月17日递交的美国专利申请No.16/655,950的优先权,该专利申请要求于2018年10月18日递交的美国临时专利申请No.62/747,543、2018年12月28日递交的美国临时专利申请No.62/786,285、2019年1月4日递交的美国临时专利申请No.62/788,546和2019年1月15日递交的美国临时专利申请No.62/792,795的优先权,每个申请的完整内容以引入的方式并入本文。

技术领域

[0002] 本公开涉及视频编码和视频解码。

背景技术

[0003] 数字视频功能能够集成到广泛的设备中,包括数字电视、数字直接广播系统、无线广播系统、个人数字助理(PDA)、便携式或台式计算机、平板电脑、电子书阅读器、数码相机、数字记录设备、数字媒体播放器、视频游戏设备、视频游戏机、蜂窝式或卫星无线电电话、所谓的“智能电话”、视频电话会议设备、视频流设备等。数字视频设备实施视频译码(coding)技术,诸如那些在由MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4第10部分、高级视频译码(AVC)、ITU-T H.265/高效视频译码(HEVC)所定义的标准以及此类标准的扩展中所描述的技术。通过实施此类视频译码技术,视频设备可以更高效地发送、接收、编码、解码和/或存储数字视频信息。

[0004] 视频译码技术包括空间(图片内)预测和/或时间(图片间)预测来减少或消除视频序列中固有的冗余。对于基于块的视频译码,可以将视频切片(slice)(例如,视频图片或视频图片的一部分)分割为视频块,其也可以称为译码树单元(CTU)、译码单元(CU)和/或译码节点。图片的帧内译码的(I)切片中的视频块可以使用相对于同一图片中相邻块中的参考样本的空间预测而进行编码。图片的帧间译码的(P或B)切片中的视频块可以使用相对于同一图片中相邻块中的参考样本的空间预测或者相对于其它参考图片中的参考样本的时间预测。图片可以称为帧,并且参考图片可以称为参考帧。

发明内容

[0005] 一般地,本公开描述了用于视频译码的变换系数译码的技术。在一些示例中,本公开对于将置零(zero-out)技术用于高频系数的大的变换(large transform),描述了变换系数扫描技术和最后非零系数位置译码技术。当使用置零技术时,视频译码器可被配置为将变换单元的某个区域中的所有变换系数设置为具有零的值。例如,对于 64×64 变换单元,视频译码器可以使变换单元左上角的 32×32 区域中的值(例如,最低频率分量)保持不变。然而,视频译码器将把该区域(即,置零区域)之外的所有变换系数设置为具有零的值。

[0006] 在一个示例中,本公开描述了一种扫描技术,其包括仅扫描变换单元中的在置零区域之外(即,在非置零区域中)的变换系数。以此方式,可以避免对已知具有零值的变换系数的扫描,从而增加译码器吞吐量。

[0007] 另外,本公开描述了对具有置零区域的变换单元中的最后非零变换系数的位置进行译码的技术。例如,本公开描述了其中视频译码器确定用于对指示最后非零变换系数的位置的语法元素进行译码的上下文(context)(例如,概率(probability)模型)的技术。在一个示例中,基于变换单元的整个尺寸而不是基于变换单元的置零区域的尺寸,来确定该上下文。即使最后非零变换系数的位置被保证在非置零区域(例如 64×64 变换单元的左上 32×32 区域)中,但是对最后非零变换系数的位置进行译码的上下文与 64×64 变换单元(而不是 32×32 变换单元)的统计数据(statistics)更密切相关。因此,可以提高对最后非零变换系数的位置的译码效率。

[0008] 在一个示例中,一种方法包括:确定变换单元中变换系数经历置零的第一区域;确定所述变换单元中变换系数未经历置零的第二区域;以及仅扫描所述变换单元的第二区域。

[0009] 在另一示例中,一种设备包括存储器和与存储器通信的一个或多个处理器,该一个或多个处理器被配置为确定变换单元中变换系数经历置零的第一区域,确定所述变换单元中变换系数未经历置零的第二区域,以及仅扫描所述变换单元的第二区域。

[0010] 在另一个示例中,一种设备包括用于确定变换单元中变换系数经历置零的第一区域的部件,用于确定变换单元中变换系数未经历置零的第二区域的部件,以及用于仅扫描变换单元的第二区域的部件。

[0011] 在另一个示例中,一种计算机可读存储介质被编码有置零,该指令被执行时使可编程处理器确定变换单元中变换系数经历置零的第一区域,确定变换单元中变换系数未经历置零的第二区域,以及仅扫描变换单元的第二区域。

[0012] 在附图和以下描述中阐述一个或多个示例的细节。根据说明书、附图和权利要求书,其它特征、目的和优势将显而易见。

附图说明

[0013] 图1是示出可以执行本公开的技术的示例视频编码和解码系统的框图。

[0014] 图2A和图2B是示出示例四叉树二叉树(QTBT)结构和对应的译码树单元(CTU)的概念示意图。

[0015] 图3是示出可以执行本公开的技术的示例视频编码器的框图。

[0016] 图4是示出可以执行本公开的技术的示例视频解码器的框图。

[0017] 图5是示出包括根据本公开的一种或多种技术的示例扫描的示例变换单元扫描的概念示意图。

[0018] 图6是示出根据本公开的技术的示例编码方法的流程图。

[0019] 图7是示出根据本公开的技术的示例解码方法的流程图。

具体实施方式

[0020] 图1是示出可执行本公开的用于扫描和译码置零变换单元的技术的示例视频编码和解码系统100的框图。本公开的技术总体上指向对视频数据进行译码(编码和/或解码)。一般地,视频数据包括用于处理视频的任何数据。从而,视频数据可以包括原始的未译码视频、经编码视频、经解码(例如经重构)视频以及视频元数据(诸如,信令通知(signaling)数

据)。

[0021] 如图1所示,在此示例中,系统100包括提供将由目的地设备116进行解码和显示的经编码视频数据的源设备102。具体地,源设备102经由计算机可读介质110将视频数据提供给目的地设备116。源设备102和目的地设备116可以包括多种设备中的任何一种,包括台式计算机、笔记本(即膝上型计算机)、平板电脑、机顶盒、手持电话(如智能手机)、电视、相机、显示设备、数字媒体播放器、视频游戏机、视频流设备等。在一些情况下,源设备102和目的地设备116可以被配备用于无线通信,并且因此可以称为无线通信设备。

[0022] 在图1的示例中,源设备102包括视频源104、存储器106、视频编码器200和输出接口108。目的地设备116包括输入接口122、视频解码器300、存储器120和显示设备118。根据本公开,源设备102的视频编码器200和目的地设备116的视频解码器300可以配置为应用于变换系数译码的技术。因而,源设备102表示视频编码设备的示例,而目的地设备116表示视频解码设备的示例。在其它示例中,源设备和目的地设备可以包括其它组件或布置。例如,源设备102可以从诸如外部相机的外部视频源接收视频数据。同样,目的地设备116可以与外部显示设备通过接口相连接,而不包括集成的显示设备。

[0023] 如图1所示的系统100仅是一个示例。一般地,任何数字视频编码和/或解码设备都可以执行用于变换系数译码的技术。源设备102和目的地设备116仅仅是此类译码设备的示例,其中,源设备102生成用于传输到目的地设备116的译码视频数据。本公开将“译码”设备称为执行数据译码(编码和/或解码)的设备。从而,视频编码器200和视频解码器300表示译码设备的示例,具体地,分别是视频编码器和视频解码器。在一些示例中,源设备102和目的地设备116可以以基本上对称的方式操作,使得源设备102和目的地设备116中的每一个均包括视频编码和解码组件。因此,系统100可以支持源设备102和目的地设备116之间的单向或双向视频传输,例如用于视频流、视频回放、视频广播或者视频电话。

[0024] 一般地,视频源104表示视频数据源(即原始的未译码视频数据),并将视频数据的连续的图片(也称为“帧”)序列提供给视频编码器200,其对图片的数据进行编码。源设备102的视频源104可以包括视频捕获设备,诸如视频相机、包括先前捕获的原始视频的视频档案和/或从视频内容提供商接收视频的视频馈送接口。作为进一步的替代方案,视频源104可以生成基于计算机图形的数据作为源视频,或者实况视频、存档视频和计算机生成视频的组合。在每种情况下,视频编码器200对捕获的、预捕获的或计算机生成的视频数据进行编码。视频编码器200可以将图片从接收顺序(有时称为“显示顺序”)重新排列为用于译码的译码顺序。视频编码器200可以生成包括经编码视频数据的比特流。然后,源设备102可以经由输出接口108将经编码视频数据输出到计算机可读介质110上,通过例如目的地设备116的输入接口122进行接收和/或取回。

[0025] 源设备102的存储器106和目的地设备116的存储器120表示通用存储器。在一些示例中,存储器106、120可以存储原始视频数据,例如来自视频源104的原始视频和来自视频解码器300的原始经解码视频数据。另外地或可替代地,存储器106、120可以分别存储可由例如视频编码器200和视频解码器300执行的软件指令。尽管在此示例中存储器106和存储器120与视频编码器200和视频解码器300分开示出,但是应当理解的是,视频编码器200和视频解码器300还可以包括实现功能上相似或等效目的的内部存储器。此外,存储器106、120可以存储例如从视频编码器200输出并输入到视频解码器300的经编码视频数据。在一

些示例中,存储器106、120的部分可以被分配为一个或多个视频缓冲区,例如用来存储原始、经解码和/或经编码的视频数据。

[0026] 计算机可读介质110可以表示能够将经编码视频数据从源设备102传输到目的地设备116的任何类型的介质或设备。在一些示例中,计算机可读介质110表示通信介质以使源设备102能够例如经由射频网络或基于计算机的网络将经编码视频数据实时地直接发送到目的地设备116。根据诸如无线通信协议的通信标准,输出接口108可以对包括经编码视频数据的传输信号进行调制,并且输入接口122可以对接收到的传输信号进行解调制。通信介质可以包括任何无线或有线通信介质,诸如射频(RF)频谱或一条或多条物理传输线。通信介质可以形成诸如局域网、广域网或诸如因特网的全球网络的基于分组的网络的一部分。通信介质可以包括路由器、交换机、基站或有助于从源设备102到目的地设备116的通信的任何其它装备。

[0027] 在一些示例中,计算机可读介质110可以包括存储设备112。源设备102可以将经编码数据从输出接口108输出到存储设备112。类似地,目的地设备116可以经由输入接口122访问来自存储设备112的经编码数据。存储设备112可以包括各种分布式或本地访问的数据存储介质中的任何一种,诸如硬盘、蓝光光盘、DVD、CD-ROM、闪存、易失性或非易失性存储器,或者用于存储经编码视频数据的任何其它合适的数字存储介质。

[0028] 在一些示例中,计算机可读介质110可以包括文件服务器114或可存储由源设备102生成的经编码视频数据的另一中间存储设备。源设备102可将经编码视频数据输出到文件服务器114。目的地设备116可以经由流传输或下载来访问来自文件服务器114的已存储视频数据。文件服务器114可以是能够存储经编码视频数据并将经编码视频数据发送到目的地设备116的任何类型的服务器设备。文件服务器114可以表示(例如用于网站的)网络服务器、文件传输协议(FTP)服务器、内容传递网络设备或网络附加存储(NAS)设备。目的地设备116可以通过包括因特网连接的任何标准数据连接来访问来自文件服务器114的经编码视频数据。这可以包括无线信道(例如Wi-Fi连接)、有线连接(例如数字订户线路(DSL)、电缆调制解调器等)或者适合访问存储在文件服务器114上的经编码视频数据的二者的组合。文件服务器114和输入接口122可以配置为根据流传输协议、下载传输协议或其组合来操作。

[0029] 输出接口108和输入接口122可以表示无线发射机/接收机、调制解调器、有线联网组件(例如以太网卡)、根据各种IEEE 802.11标准中的任何一种进行操作的无线通信组件,或者其它物理组件。在输出接口108和输入接口122包括无线组件的示例中,输出接口108和输入接口122可以配置为根据诸如4G、4G-LTE(长期演进)、LTE高级、5G或类似标准的蜂窝通信标准来传输诸如经编码视频数据的数据。在输出接口108和输入接口122包括无线发射机和/或无线接收机的某些示例中,输出接口108和输入接口122可以配置为根据其它无线标准,诸如IEEE 802.11规范、IEEE 802.15规范(例如ZigBee™)、Bluetooth™标准等来传输诸如编码视频数据的数据。在一些示例中,源设备102和/或目的地设备116可以包括各自的片上系统(SoC)设备。例如,源设备102可以包括SoC设备来执行归于视频编码器200和/或输出接口108的功能,并且目的地设备116可以包括SoC设备来执行归于视频解码器300和/或输入接口122的功能。

[0030] 本公开的技术可以应用于支持各种多媒体应用中的任何一种的视频译码,诸如空

中电视广播、有线电视传输、卫星电视传输、诸如基于HTTP的动态自适应流(DASH)的因特网流视频传输、编码到数据存储介质上的数字视频、对存储在数据存储介质上的数字视频进行解码或者其它应用。

[0031] 目的地设备116的输入接口122从计算机可读介质110(例如通信介质、存储设备112、文件服务器114等)接收经编码视频比特流。来自计算机可读介质110的经编码视频比特流可以包括由视频编码器200定义的、也由视频解码器300使用的信令信息,诸如语法元素,该语法元素具有描述视频块或其它译码单元(例如切片、图片、图片组、序列等)的特点和/或处理的值。显示设备118向用户显示经解码视频数据的经解码图片。显示设备118可以表示诸如阴极射线管(CRT)、液晶显示器(LCD)、等离子显示器、有机发光二极管(OLED)显示器或另一类显示设备的各种显示设备中的任何一种。

[0032] 尽管未在图1中示出,但在一些示例中,视频编码器200和视频解码器300中的每一个可以与音频编码器和/或音频解码器集成在一起,并且可以包括适当的MUX-DEMUX单元或其它硬件和/或软件,来处理公共数据流中包括音频和视频的多路复用流。如果适用,MUX-DEMUX单元可以符合ITU H.223多路复用器协议或诸如用户数据报协议(UDP)的其它协议。

[0033] 视频编码器200和视频解码器300中的每一个可以实现为各种合适的编码器和/或解码器电路中的任何一种,诸如一个或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、离散逻辑、软件、硬件、固件或其任何组合。当该技术部分地以软件实现时,设备可以将用于软件的指令存储在合适的非暂时计算机可读介质中,并使用一个或多个处理器在硬件中执行该指令来执行本公开的技术。视频编码器200和视频解码器300中的每一个可以被包括在一个或多个编码器或解码器中,这两者都可以集成为各自设备中组合编码器/解码器(CODEC)的一部分。包括视频编码器200和/或视频解码器300的设备可以包括集成电路、微处理器和/或诸如蜂窝电话的无线通信设备。

[0034] 一般地,本公开描述了用于视频译码的变换系数译码的技术。在一些示例中,本公开对于大的变换(其将置零技术用于高频系数以降低复杂度),描述了变换系数扫描技术和最后非零系数位置译码技术。即,在一些示例中,视频译码器(例如,视频编码器200和/或视频解码器300)可以被配置为将变换单元的某个区域中的所有变换系数设置为具有零的值。例如,对于 64×64 变换单元,视频译码器可以使变换单元左上角的 32×32 区域中的值(例如,最低频率分量)保持原样。然而,视频译码器将把该区域(即,置零区域)之外的所有变换系数设置为具有零的值。

[0035] 在一个示例中,本公开描述了一种扫描技术,其包括仅扫描变换单元中的在置零区域之外的变换系数。以此方式,可以避免对已知具有零值的变换系数的扫描,从而增加译码器吞吐量。

[0036] 另外,本公开描述了对具有置零区域的变换单元中的最后非零变换系数的位置进行译码的技术。例如,本公开描述了其中视频译码器确定用于对指示最后非零变换系数的位置的语法元素进行译码的上下文(例如,概率模型)的技术。在一个示例中,基于变换单元的整个尺寸、而不是基于变换单元的非置零区域的尺寸,来确定该上下文。即使最后非零变换系数的位置被保证在非置零区域(例如 64×64 变换系数的左上 32×32 区域)中,但是对最后非零变换系数的位置进行译码的上下文与 64×64 变换单元(而不是 32×32 变换单元)的统计数据更密切相关。因此,可以提高对最后非零变换系数的位置的译码效率。

[0037] 如以下将更详细解释说明的,视频编码器200和视频解码器300可以被配置为确定变换单元中变换系数经历置零的第一区域,确定变换单元中变换系数未经历置零的第二区域,以及仅扫描变换单元的第二区域。

[0038] 视频编码器200和视频解码器300可以根据视频译码标准(诸如ITU-T H.265,也称为高效视频编码(HEVC))或其扩展(诸如多视图和/或可缩放的视频编码扩展)来操作。作为替代,视频编码器200和视频解码器300可以根据其它专有或工业标准,诸如联合探索测试模型(JEM)和/或H.266/VVC(通用视频编码)。但是,本公开的技术不限于任何特定的编码标准。本公开的技术可适用于对各种类型的视频数据进行编码和解码。具体地,预测视频数据中的数据块、计算差异、使用多个变换之一对该差异或残差进行变换以产生变换系数。然后扫描变换系数。视频译码可以涉及根据各种不同标准的这种技术。

[0039] 在Benjamin Bross的文档JVET-J1001rVersatile Video Coding(Draft 1)”中获得新的视频译码标准的早期草案(被称为H.266/通用视频编码(VVC)标准),并且在Jianle Chen和Elena Alshina的文档JVET-J1002“Algorithm description for Versatile Video Coding and Test Model 1(VTM 1)”中可以获得其算法描述。

[0040] 一般地,视频编码器200和视频解码器300可以执行图片的基于块的译码。术语“块”一般是指包括待处理的(例如待编码、解码或以其它方式在编码和/或解码过程中使用的)数据的结构。例如,块可以包括亮度和/或色度数据的样本的二维矩阵。一般地,视频编码器200和视频解码器300可以对以YUV(例如Y、Cb、Cr)格式表示的视频数据进行译码。也就是说,并非对图片的样本的红色、绿色和蓝色(RGB)数据进行译码,视频编码器200和视频解码器300而是可以对亮度和色度分量进行译码,其中色度分量可以包括红色和蓝色色度分量。在一些示例中,视频编码器200在编码之前将接收到的RGB格式的数据转换成YUV表示,并且视频解码器300将YUV表示转换成RGB格式。可替代地,预处理和后处理单元(未示出)可以执行这些转换。

[0041] 本公开通常涉及图片的译码(例如编码和解码)以包括对图片数据进行编码或解码的过程。类似地,本公开可以涉及对图片的块进行译码以包括对块的数据进行编码或解码的过程,例如预测和/或残差译码。经编码视频比特流一般包括表示译码决策(例如译码模式)和将图片分割为块的语法元素的一系列值。从而,对图片或块进行译码的引用一般应理解为对形成图片或块的语法元素的值进行译码。

[0042] HEVC定义了各种的块,包括译码单元(CU)、预测单元(PU)和变换单元(TU)。根据HEVC,视频译码器(诸如视频编码器200)根据二叉树结构将译码树单元(CTU)分割为CU。也就是说,视频译码器将CTU和CU分割为四个相等的非重叠正方形,并且二叉树的每一个节点具有零个或四个子代节点。没有子代节点的节点可以称为“叶节点”,并且这种叶节点的CU可以包括一个或多个PU和/或一个或多个TU。视频译码器可以进一步分割PU和TU。例如,在HEVC中,残差二叉树(RQT)表示TU的分割。在HEVC中,PU表示帧间预测数据,而TU表示残差数据。被帧内预测的CU包括帧内预测信息,诸如帧内模式指示。

[0043] 作为另一示例,视频编码器200和视频解码器300可以配置为根据JEM或VVC进行操作。根据JEM或VVC,视频译码器(诸如视频编码器200)将图片分割为多个译码树单元(CTU)。视频编码器200可以根据诸如二叉树-二叉树(QTBT)结构或多类型树(MTT)结构的树结构,来对CTU进行分割。QTBT结构消除了多个分割类型的概念,诸如HEVC的CU、PU和TU之间的区

分。QTBT结构包括两个级别：根据二叉树分割而分割出来的第一级，以及根据二叉树分割而分割出来的第二级。QTBT结构的根节点对应于CTU。二叉树的叶节点对应于译码单元(CU)。

[0044] 在MTT分割结构中，可以使用二叉树(QT)分割、二叉树(BT)分割和/或一种或多种类型的三叉树(TT)（也称为三元树(TT)）分割，来对块进行分割。三叉树或三元树分割是将一个块拆分为三个子块的分割。在一些示例中，三叉树或三元树分割将块划分为三个子块，而不通过中心划分原始块。MTT中的分割类型（例如QT、BT和TT）可以是对称的或不对称的。

[0045] 在一些示例中，视频编码器200和视频解码器300可以使用单个QTBT或MTT结构来表示亮度分量和色度分量中的每一个，而在其它示例中，视频编码器200和视频解码器300可以使用两个或更多个QTBT或MTT结构，诸如用于亮度分量的一个QTBT/MTT结构和用于两个色度分量的另一QTBT/MTT结构（或用于各自色度分量的两个QTBT/MTT结构）。

[0046] 视频编码器200和视频解码器300可以配置为使用根据HEVC的二叉树分割、QTBT分割、MTT分割或其它分割结构。出于说明的目的，针对QTBT分割给出了本公开的技术的描述。然而，应该理解的是，本公开的技术还可以应用于配置为使用二叉树分割、MTT分割或其它类型的分割的视频译码器。

[0047] 块（例如，CTU或CU）可以在图片中以各种方式被分组。作为一个示例，砖块(brick)可以指图片中特定图块(tile)内的CTU行的矩形区域。图块可以是图片中特定图块列和特定图块行内的CTU的矩形区域。图块列指的是CTU的矩形区域，其高度等于图片的高度，宽度由语法元素（例如，诸如在图片参数集中）指定。图块行是指CTU的矩形区域，其高度由语法元素（例如，诸如在图片参数集中）指定，宽度等于图片的宽度。

[0048] 在一些示例中，图块可以被分割为多个砖块，每个砖块可以包括图块内的一个或多个CTU行。未分割成多个砖块的图块也可被称为砖块。然而，作为图块的真正子集的砖块可以不被称为图块。

[0049] 图片中的砖块也可以被排列在切片中。切片可以是图片的整数个砖块，其可以排它地包含在单个网络抽象层(NAL)单元中。在一些示例中，切片包括若干个完整图块或仅包括一个图块的完整砖块的连续序列。

[0050] 本公开可以互换地使用“ $N \times N$ ”和“ N 乘 N ”来指代块（诸如CU或其它视频块）在垂直和水平维度方面的样本维度，例如 16×16 样本或 16 乘 16 样本。一般地， 16×16 CU在垂直方向上将有 16 个样本($y=16$)，且在水平方向上将有 16 个样本($x=16$)。同样地， $N \times N$ CU一般地在垂直方向上具有 N 个样本，且在水平方向上具有 N 个样本，其中 N 表示非负整数值。CU中的样本可以按行和列来排列。此外，CU在水平方向上不必具有与垂直方向上相同数量的样本。举例来说，CU可以包含 N 以包个样本，其中 M 不一定等于 N 。

[0051] 视频编码器200对用于表示预测和/或残差信息以及其它信息的CU的视频数据进行编码。预测信息指示将如何预测CU以便形成用于该CU的预测块。残差信息一般表示编码前CU的样本与预测块之间的逐样本差异。

[0052] 为了预测CU，视频编码器200一般地可以通过帧间预测或帧内预测来形成用于CU的预测块。帧间预测一般是指从先前译码的图片的数据中预测CU，而帧内预测一般是指从同一图片的先前译码数据中预测CU。为了执行帧间预测，视频编码器200可以使用一个或多个运动矢量来生成预测块。视频编码器200通常可以执行运动搜索以识别例如在CU和参考块之间的差异的方面与CU紧密地匹配的参考块。视频编码器200可以使用绝对差之和

(SAD)、平方差之和(SSD)、平均绝对差(MAD)、均方差(MSD)或其它此类差值计算,来计算差值度量以确定参考块是否与当前CU紧密地匹配。在某些示例中,视频编码器200可以使用单向预测或双向预测来预测当前CU。

[0053] JEM和VVC的一些示例还提供仿射运动补偿模式,其可以视为帧间预测模式。在仿射运动补偿模式中,视频编码器200可以确定表示非平移运动的两个或更多个运动矢量,上述非平移运动诸如放大或缩小、旋转、透视运动或其它不规则运动类型。

[0054] 为了执行帧内预测,视频编码器200可以选择帧内预测模式来生成预测块。JEM和VVC的一些示例提供了六十七种帧内预测模式,包括各种定向模式以及平面模式和DC模式。一般地,视频编码器200选择帧内预测模式,其描述当前块(例如CU的块)的相邻样本,以从其预测当前块的预测样本。假设视频编码器200以光栅扫描顺序(从左到右、从上到下)对CTU和CU进行译码,则此类样本通常可以在与当前块相同的图片中在当前块的上方、上左侧或左侧。

[0055] 视频编码器200对表示当前块的预测模式的数据进行编码。举例来说,对于帧间预测模式,视频编码器200可以对数据进行编码,该数据表示使用了各种可用帧间预测模式中的哪一种以及对对应模式的运动信息。对于单向或双向帧间预测,例如视频编码器200可以使用高级运动矢量预测(AMVP)模式或合并模式来对运动矢量进行编码。视频编码器200可以使用类似模式来对仿射运动补偿模式的运动矢量进行编码。

[0056] 在预测(诸如块的帧内预测或帧间预测)之后,视频编码器200可以计算该块的残差数据。残差数据(诸如残差块)表示该块与该块的预测块之间的逐样本差异,该预测块是使用对应预测模式而形成的。视频编码器200可以将一个或多个变换应用于残差块以产生变换域而非样本域中的变换数据。举例来说,视频编码器200可以将离散余弦变换(DCT)、整数变换、小波变换或概念上类似的变换应用于残差视频数据。另外,视频编码器200可以在一次变换之后应用二次变换,诸如模式相关的不可分二次变换(MDNSST)、信号相关的变换、Karhunen-Loeve变换(KLT)等。视频编码器200在应用一个或多个变换之后产生变换系数。

[0057] 如上所述,在进行任何变换以产生变换系数之后,视频编码器200可以对变换系数执行量化。量化通常是指对变换系数进行量化来可能地减少用于表示变换系数的数据量,从而提供进一步的压缩的过程。通过执行量化过程,视频编码器200可以减小与变换系数中的一些或全部相关联的比特深度。例如,视频编码器200可以在量化期间将n-位值向下舍入为m-位值,其中n大于m。在一些示例中,为了执行量化,视频编码器200可以执行待量化值的按位右移。

[0058] 在量化之后,视频编码器200可以扫描变换系数,从而从包括经量化变换系数的二维矩阵中产生一维矢量。可以将扫描设计为将较高能量(因此频率较低)的变换系数放在矢量的前面,并将较低能量(因此频率较高)的变换系数放在矢量的后面。在一些示例中,视频编码器200可以利用预定义的扫描顺序来对经量化变换系数进行扫描,以产生序列化的矢量,然后对该矢量的经量化变换系数进行熵编码。在其它示例中,视频编码器200可以执行自适应扫描。在对量化变换系数进行扫描以形成一维矢量之后,视频编码器200可以例如根据上下文自适应二进制算术译码(CABAC)来对一维矢量进行熵编码。视频编码器200还可以对语法元素的值进行熵编码,该语法元素描述与经编码视频数据相关联的元数据,该元数据供视频解码器300在解码视频数据时使用。

[0059] 为了执行CABAC,视频编码器200可以将上下文模型内的上下文分配给待传输的符号。例如,上下文可以涉及该符号的相邻值是否是零值。概率确定可以基于分配给符号的上下文。

[0060] 视频编码器200可以进一步地生成例如在图片标头、块标头、切片标头中发给视频解码器300的语法数据,诸如基于块的语法数据、基于图片的语法数据以及基于序列的语法数据,或者生成其它语法数据,诸如序列参数集(SPS)、图片参数集(PPS)或视频参数集(VPS)。视频解码器300可以类似地对此类语法数据进行解码以确定如何解码对应的视频数据。

[0061] 以此方式,视频编码器200可以生成包括经编码视频数据的比特流,例如描述将图片分割成块(例如CU)的语法元素以及块的预测和/或残差信息。最终,视频解码器300可以接收比特流并且对经编码视频数据进行解码。

[0062] 一般地,视频解码器300执行与由视频编码器200执行的相反的过程,以对比特流的经编码视频数据进行解码。例如,视频解码器300可以使用CABAC,以与视频编码器200的CABAC编码过程基本相似(尽管与之相反)的方式,对比特流的语法元素的值进行解码。语法元素可以定义分割信息,该分割信息关于将图片分割为CTU和根据诸如QTBT结构的对应分割结构对每一个CTU进行分割以定义CTU的CU。语法元素可以进一步定义视频数据的块(例如CU)的预测和残差信息。

[0063] 例如,可以由经量化变换系数表示残差信息。视频解码器300可以对块的经量化变换系数进行逆量化和逆变换,以重现该块的残差块。视频解码器300使用被信令通知的预测模式(帧内或帧间预测)和相关的预测信息(例如用于帧间预测的运动信息)来形成该块的预测块。然后,视频解码器300可以在逐样本的基础上)组合预测块和残差块以重现原始块。视频解码器300可以执行附加过程(诸如执行去方块(deblock)过程)来减少沿着块边界的视觉伪像。

[0064] 一般地,本公开可以涉及“信令通知”某些信息,诸如语法元素。术语“信令通知”通常可以指对于语法元素和/或被用来对经编码视频数据进行解码的其它数据的值的通信。也就是说,视频编码器200可以信令通知比特流中的语法元素的值。一般地,信令通知是指在比特流中生成值。如上所述,源设备102可以基本上实时地(或非实时地,诸如可能在将语法元素存储到存储设备112以供稍后由目的地设备116取回时发生)将比特流传送到目的地设备116。

[0065] 图2A和图2B是示出示例四叉树二叉树(QTBT)结构130和对应的译码树单元(CTU)132的概念示意图。实线表示四叉树划分,虚线指示二叉树划分。在二叉树的每一个划分(即非叶)节点中,一个标志被信令通知以指示使用了哪种划分类型(即水平或垂直),其中在此示例中,0指示水平划分,且1指示垂直划分。对于四叉树划分,由于四叉树节点将块水平地和垂直地划分为具有相等尺寸的4个子块,因此无需指示划分类型。相应地,视频编码器200可以编码QTBT结构130的区域树级(即实线)的语法元素(例如划分信息)和QTBT结构130的预测树级(即虚线)的语法元素(例如划分信息),并且视频解码器300可以对以上进行解码。对于QTBT结构130的终端叶节点表示的CU,视频编码器200可以编码视频数据(诸如预测和变换数据),并且视频解码器300可以对以上进行解码。

[0066] 一般地,图2B的CTU 132可以与参数相关联,这些参数定义与QTBT结构130在第一

和第二级处的节点相对应的块的尺寸。如上所述,第一级可以根据二叉树分割来分割,而第二级可以根据二叉树分割来分割。这些参数可以包括CTU尺寸(表示样本中CTU 132的尺寸)、最小二叉树尺寸(MinQTSize,表示最小允许的二叉树叶节点尺寸)、最大二叉树尺寸(MaxBTSIZE,表示最大允许的二叉树根节点尺寸)、最大二叉树深度(MaxBTDepth,表示最大允许的二叉树深度)和最小二叉树尺寸(MinBTSIZE,表示最小允许的二叉树叶节点尺寸)。

[0067] QTBT结构的与CTU相对应的根节点在QTBT结构的第一级可以具有四个子代节点,每个子代节点可以根据二叉树分割来进行分割。即,第一级的节点是叶节点(没有子代节点)或具有四个子代节点。QTBT结构130的示例将这样的节点表示为包括具有实线分支的子代节点和父节点。如果第一级的节点不大于最大允许的二叉树根节点尺寸(MaxBTSIZE),则能够通过各自的二叉树进一步对节点进行分割。能够迭代一个节点的二叉树划分,直到划分所产生的节点达到最小允许的二叉树叶节点尺寸(MinBTSIZE)或最大允许的二叉树深度(MaxBTDepth)。QTBT结构130的示例将这样的节点表示为具有虚线分支。二叉树叶节点被称为译码单元(CU),其被用于预测(例如图片内或图片间预测)和变换,而无需任何进一步分割。如上所述,CU也可以被称为“视频块”或“块”。

[0068] 在QTBT分割结构的一个示例中,CTU尺寸设置为 128×128 (亮度样本和两个对应的 64×64 色度样本),MinQTSIZE设置为 16×16 ,MaxBTSIZE设置为 64×64 ,MinBTSIZE(用于宽度和高度)设置为4,MaxBTDepth设置为4。首先,将二叉树分割应用于CTU来生成二叉树叶节点。二叉树叶节点可以具有从 16×16 (即MinQTSIZE)到 128×128 (即CTU尺寸)的尺寸。如果叶二叉树节点为 128×128 ,则该叶二叉树节点不会进一步被二叉树划分,因为尺寸超过MaxBTSIZE(在此示例中为 64×64)。否则,该叶二叉树节点将进一步进行二叉树分割。因此,二叉树叶节点也是二叉树的根节点,且具有为0的二叉树深度。当二叉树深度达到MaxBTDepth(在此示例中为4)时,不允许进一步划分。当二叉树节点具有等于MinBTSIZE(在此示例中为4)的宽度时,其意味着不允许进一步的水平划分。类似地,具有等于MinBTSIZE的高度的二叉树节点意味着不允许对该二叉树节点进行进一步的垂直划分。如上所述,二叉树的叶节点被称为CU,并且根据预测和变换对其进行进一步处理而无需进一步分割。

[0069] 图3是示出可以执行本公开的技术的示例视频编码器200的框图。提供图3是为了解释的目的并且不应认为是对本公开中广泛示例和描述的技术的约束。出于说明的目的,本公开在诸如HEVC视频译码标准和开发中的H.266/VVC视频译码标准的视频译码标准的上下文中,描述了视频编码器200。然而,本公开的技术不限于这些视频译码标准并且通常可应用于视频编码和解码。

[0070] 在图3的示例中,视频编码器200包括视频数据存储器230、模式选择单元202、残差生成单元204、变换处理单元206、量化单元208、逆量化单元210、逆变换处理单元212、重构单元214、滤波器单元216、经解码图片缓冲器(DPB)218以及熵编码单元220。

[0071] 视频数据存储器230可以存储将由视频编码器200的组件进行编码的视频数据。视频编码器200可以从例如视频源104(图1)接收存储在视频数据存储器230中的视频数据。DPB 218可以用作参考图片存储器,该参考图片存储器存储参考视频数据,以供视频编码器200用于预测后续视频数据时使用。视频数据存储器230和DPB 218可以由多种存储设备中的任何一种形成,诸如动态随机存取存储器(DRAM),包括同步DRAM(SDRAM)、磁阻RAM(MRAM)、阻变RAM(RRAM)或其它类型的存储设备。视频数据存储器230和DPB 218可以由相同

的存储设备或单独的存储设备提供。在各种示例中,视频数据存储器230可以与视频编码器200的其它组件一起置于片上,如图所示,或者相对于那些组件置于片外。

[0072] 在本公开中,对视频数据存储器230的引用不应解释为限于视频编码器200内部的存储器(除非特别说明如此)或者视频编码器200外部的存储器(除非特别说明如此)。而是,对视频数据存储器230的引用应理解为存储视频编码器200接收的以供编码的视频数据(例如当前块的待编码视频数据)的参考存储器。图1的存储器106还可以对来自视频编码器200各个单元的输出提供临时存储。

[0073] 示出的图3的各个单元用来帮助理解由视频编码器200执行的操作。这些单元可以实现为固定功能电路、可编程电路或其组合。固定功能电路是指提供特定功能并预设了能够执行的操作的电路。可编程电路是指可以被编程来执行各种任务并且在能够执行的操作中提供灵活功能的电路。例如,可编程电路可以执行软件或固件,该软件或固件使可编程电路以软件或固件的指令所定义的方式操作。固定功能电路可以执行软件指令(例如来接收参数或输出参数),但是固定功能电路执行的操作类型通常是不可变的。在一些示例中,这些单元中的一个或多个可以是不同的电路块(固定功能或可编程),并且在一些示例中,一个或多个单元可以是集成电路。

[0074] 视频编码器200可以包括算术逻辑单元(ALU)、基本功能单元(EFU)、数字电路、模拟电路和/或由可编程电路形成的可编程内核。在使用由可编程电路执行的软件来执行视频编码器200的操作的示例中,存储器106(图1)可以存储视频编码器200接收和执行的软件的指令(例如目标代码),或视频编码器200(未示出)内另一存储器可以存储这样的指令。

[0075] 视频数据存储器230被配置为存储接收的视频数据。视频编码器200可以从视频数据存储器230中取回视频数据的图片,并将视频数据提供给残差生成单元204和模式选择单元202。视频数据存储器230中的视频数据可以是待编码的原始视频数据。

[0076] 模式选择单元202包括运动估计单元222、运动补偿单元224和帧内预测单元226。模式选择单元202可以包括附加功能单元,用于根据其它预测模式来执行视频预测。作为示例,模式选择单元202可以包括调色板(palette)单元、帧内块复制单元(其可以是运动估计单元222和/或运动补偿单元224的一部分)、仿射单元、线性模型(LM)单元等。

[0077] 一般地,模式选择单元202协调多个编码遍(pass),来测试编码参数的组合以及对这种组合得出的率失真值。编码参数可以包括CTU到CU的分割、CU的预测模式、CU的残差数据的变换类型、CU的残差数据的量化参数等。模式选择单元202可以最终选择编码参数的组合,该组合具有比其它测试的组合更佳的率失真值。

[0078] 视频编码器200可以将从视频数据存储器230中取回的图片分割为一系列CTU,并将一个或多个CTU封装在切片内。模式选择单元202可以根据树结构(诸如上述HEVC的QTBT结构、MTT结构或二叉树结构)来对图片的CTU进行分割。如上所述,视频编码器200可以从根据树结构对CTU的分割中形成一个或多个CU。这样的CU通常也可以称为“视频块”或“块”。

[0079] 一般地,模式选择单元202还控制其组件(例如运动估计单元222、运动补偿单元224和帧内预测单元226)以生成当前块(例如当前CU,或HEVC中PU和TU的重叠部分)的预测块。对于当前块的帧间预测,运动估计单元222可以执行运动搜索来识别在一个或多个参考图片(例如存储在DPB 218中的一个或多个先前译码的图片)中的一个或多个紧密匹配的参考块。具体地,运动估计单元222可以根据例如绝对差之和(SAD)、差值平方和(SSD)、平均绝

对差 (MAD)、均方差 (MSD) 等, 来计算表示潜在参考块与当前块有多相似的值。运动估计单元 222 通常可以使用当前块与考虑中的参考块之间的逐样本差异, 来执行这些计算。运动估计单元 222 可以识别具有从这些计算生成的最低值的参考块, 其指示与当前块最紧密匹配的参考块。

[0080] 运动估计单元 222 可以形成一个或多个运动矢量 (MV), 运动矢量定义参考图片中参考块的位置相对于当前图片中当前块的位置。然后, 运动估计单元 222 可以将运动矢量提供给运动补偿单元 224。例如, 对于单向帧间预测, 运动估计单元 222 可以提供单个运动矢量, 而对于双向帧间预测, 运动估计单元 222 可以提供两个运动矢量。然后, 运动补偿单元 224 可以使用运动矢量来生成预测块。例如, 运动补偿单元 224 可以使用运动矢量来取回参考块的数据。作为另一示例, 如果运动矢量具有分数样本精度, 则运动补偿单元 224 可以根据一个或多个插值滤波器来对预测块进行插值。此外, 对于双向帧间预测, 运动补偿单元 224 可以取回由各自运动矢量标识的两个参考块的数据, 并且 (例如通过逐样本平均或加权平均) 来组合取回的数据。

[0081] 作为另一示例, 对于帧内预测或帧内预测译码, 帧内预测单元 226 可以根据与当前块相邻的样本来生成预测块。例如, 对于定向模式, 帧内预测单元 226 通常可以数学地组合相邻样本的值, 并且在当前块上沿定义的方向填充这些计算值来产生预测块。作为另一示例, 对于 DC 模式, 帧内预测单元 226 可以计算当前块的相邻样本的平均值, 并且生成预测块以对于预测块的每一个样本包括所得到的平均值。

[0082] 模式选择单元 202 将预测块提供给残差生成单元 204。残差生成单元 204 从视频数据存储单元 230 接收当前块的原始未编码版本, 并从模式选择单元 202 接收预测块。残差生成单元 204 计算当前块和预测块之间的逐样本差异。得到的逐样本差异定义当前块的残差块。在一些示例中, 残差生成单元 204 还可以使用残差差值脉冲编码调制 (RDPCM) 来确定残差块中样本值之间的差以生成残差块。在某些示例中, 可以使用执行二进制减法的一个或多个减法器电路来形成残差生成单元 204。

[0083] 在模式选择单元 202 将 CU 分割为 PU 的示例中, 每个 PU 可以与亮度预测单元和对应的色度预测单元相关联。视频编码器 200 和视频解码器 300 可以支持具有各种尺寸的 PU。如上所述, CU 的尺寸可以指 CU 的亮度译码块的尺寸, 而 PU 的尺寸可以指 PU 的亮度预测单元的尺寸。假设特定 CU 的尺寸为 $2N \times 2N$, 则视频编码器 200 可以支持 $2N \times 2N$ 或 $N \times N$ 的 PU 尺寸以用于帧内预测, 以及 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $N \times N$ 或类似的对称 PU 尺寸以用于帧间预测。视频编码器 20 和视频解码器 30 还可以支持 $2N \times NU$ 、 $2N \times ND$ 、 $nL \times 2N$ 和 $nR \times 2N$ 的 PU 尺寸的非对称分割以用于帧间预测。

[0084] 在模式选择单元 202 不将 CU 进一步分割为 PU 的示例中, 每一个 CU 可以与亮度译码块和对应的色度译码块相关联。如上所述, CU 的尺寸可以指 CU 的亮度译码块的尺寸。视频编码器 200 和视频解码器 300 可以支持 $2N \times 2N$ 、 $2N \times N$ 或 $N \times 2N$ 的 CU 尺寸。

[0085] 对于其它视频译码技术, 诸如作为一些示例的帧内块复制模式译码、仿射模式译码和线性模型 (LM) 模式译码, 模式选择单元 202 经由与译码技术相关联的各自单元来生成正在被编码的当前块的预测块。在一些示例中, 诸如调色板模式译码, 模式选择单元 202 可以不生成预测块, 而是生成语法元素, 该语法元素指示基于所选调色板重构块的方式。在这样的模式中, 模式选择单元 202 可以将这些语法元素提供给熵编码单元 220, 以对其进行编

码。

[0086] 如上所述,残差生成单元204接收当前块和对应预测块的视频数据。然后,残差生成单元204生成用于当前块的残差块。为了生成残差块,残差生成单元204计算预测块和当前块之间的逐样本差异。

[0087] 变换处理单元206将一个或多个变换应用于残差块以生成变换系数的块(在本文中称为“变换系数块”)。变换处理单元206可以将各种变换应用于残差块以形成变换系数块。例如,变换处理单元206可以将离散余弦变换(DCT)、定向变换、Karhunen-Loeve变换(KLT)或概念上类似的变换应用于残差块。在一些示例中,变换处理单元206可以对残差块执行多次变换,例如,初次变换和诸如旋转变换的二次变换。在一些示例中,变换处理单元206不将变换应用于残差块。

[0088] 量化单元208可以对变换系数块中的变换系数进行量化以产生经量化变换系数块。量化单元208可以根据与当前块相关联的量化参数(QP)值来对变换系数块的变换系数进行量化。视频编码器200(例如经由模式选择单元202)可以通过调整与CU相关联的QP值,来调整应用于与当前块相关联的变换系数块的量化程度。量化可能会引入信息损失,从而,经量化变换系数可能比变换处理单元206产生的原始变换系数具有较低的精度。

[0089] 逆量化单元210和逆变换处理单元212可以分别对经量化变换系数块应用逆量化和逆变换,以从变换系数块重构残差块。重构单元214可以基于重构残差块和由模式选择单元202生成的预测块,来产生对应于当前块的重构块(尽管潜在地具有一些程度的失真)。例如,重构单元214可以将重构残差块的样本添加到模式选择单元202生成的预测块的对应样本以产生重构块。

[0090] 滤波器单元216可以对重构块执行一个或多个滤波器操作。例如,滤波器单元216可以执行去方块操作来减少沿着CU的边缘的块状伪影。在一些示例中,可以跳过滤波器单元216的操作。

[0091] 视频编码器200将重构块存储在DPB 218中。例如,在不需要滤波器单元216的操作的示例中,重构单元214可以将重构块存储到DPB 218。在需要滤波器单元216的操作的示例中,滤波器单元216可以将滤波后的重构块存储到DPB 218。运动估计单元222和运动补偿单元224可以从DPB 218中取回参考图片,该参考图片是由重构(并且潜在地经滤波的)块形成的,来对随后经编码图片的块进行帧间预测。另外,帧内预测单元226可以使用当前图片的在DPB 218中的重构块,来对当前图片中的其它块进行帧内预测。

[0092] 一般地,熵编码单元220可以对从视频编码器200的其它功能组件接收的语法元素进行熵编码。例如,熵编码单元220可以对来自量化单元208的经量化变换系数块进行熵编码。作为另一示例,熵编码单元220可以对来自模式选择单元202的预测语法元素(例如用于帧间预测的运动信息或用于帧内预测的帧内模式信息)进行熵编码。熵编码单元220可以对语法元素(其是视频数据的另一示例)执行一个或多个熵编码操作来生成经熵编码数据。例如,熵编码单元220可以对数据执行上下文自适应可变长度译码(CAVLC)操作、CABAC操作、可变到可变(V2V)长度译码操作、基于语法的上下文自适应二进制算术译码(SBAC)操作、概率区间分割熵(PIPE)译码操作、指数-格伦布译码操作或另一类型的熵编码操作。在一些示例中,熵编码单元220可以在语法元素未被熵编码的情况下以旁路模式操作。

[0093] 当熵编码语法元素表示变换系数时,熵编码单元220可被配置为以预定方式扫描

一遍变换系数。根据本公开的示例,如下面将更详细地描述的,熵编码单元220可被配置为针对将置零技术用于高频系数以降低复杂度的大的变换,执行变换系数扫描技术和最后非零系数位置译码技术。即,在一些示例中,熵编码单元220和/或变换处理单元206可以被配置为将变换单元的某个区域中的所有变换系数设置为具有零的值。例如,对于 64×64 变换单元,熵编码单元220和/或变换处理单元206可以使变换单元左上角的 32×32 区域中的值(例如,最低频率分量)保持原样。然而,熵编码单元220和/或变换处理单元206将把该区域(即,置零区域)之外的所有变换系数设置为具有零的值。

[0094] 在一个示例中,本公开描述了一种扫描技术,其中熵编码单元220被配置为仅扫描变换单元中的在置零区域之外的变换系数。以此方式,可以避免对已知具有零值的变换系数的扫描,从而增加译码器吞吐量。

[0095] 另外,本公开描述了对具有置零区域的变换单元中的最后非零变换系数的位置进行编码的技术。例如,本公开描述了其中熵编码单元220确定用于对指示最后非零变换系数的位置的语法元素进行编码的上下文(例如,概率模型)的技术。在一个示例中,熵编码单元220基于变换单元的整个尺寸而不是基于变换单元的非置零区域的尺寸,来确定该上下文。即使最后非零变换系数的位置被保证在非置零区域(例如 64×64 变换系数的左上 32×32 区域)中,但是对最后非零变换系数的位置进行编码的上下文与 64×64 变换单元(而不是 32×32 变换单元)的统计数据更密切相关。因此,可以提高对最后非零变换系数的位置的编码效率。

[0096] 如以下将更详细解释说明的,在本公开的一个示例中,熵编码单元220可以被配置为确定变换单元中变换系数经历置零的第一区域,确定变换单元中变换系数未经历置零的第二区域,以及仅扫描变换单元的第二区域。

[0097] 视频编码器200可以输出比特流,其包括对切片或图片的块进行重构所需的经熵编码语法元素。具体地,熵编码单元220可以输出比特流。

[0098] 上述操作是针对块来描述的。这样的描述应该理解为是用于亮度译码块和/或色度译码块的操作。如上所述,在一些示例中,亮度译码块和色度译码块是CU的亮度和色度分量。在一些示例中,亮度译码块和色度译码块是PU的亮度和色度分量。

[0099] 在一些示例中,不需要针对色度译码块而重复对于亮度译码块执行的操作。作为一个示例,不需要针对识别色度块的运动矢量(MV)和参考图片而重复识别亮度译码块的MV和参考图片的操作。相反,可以缩放亮度译码块的MV来确定色度块的MV,并且参考图片可以是相同的。作为另一示例,对于亮度译码块和色度译码块,帧内预测处理可以是相同的。

[0100] 图4是示出可以执行本公开的技术的示例视频解码器300的框图。提供图4是出于解释的目的,而非限制本公开中广泛示例和描述的技术。为了说明的目的,本公开描述了根据JEM和HEVC的技术描述的视频解码器300。然而,本公开的技术可以由根据其它视频译码标准配置的视频译码设备来执行。

[0101] 在图4的示例中,视频解码器300包括经译码图片缓冲器(CPB)存储器320、熵解码单元302、预测处理单元304、逆量化单元306、逆变换处理单元308、重构单元310、滤波器单元312和经解码图片缓冲器(DPB)314。预测处理单元304包括运动补偿单元316和帧内预测单元318。预测处理单元304可以包括附加单元以按照其它预测模式来执行预测。作为示例,预测处理单元304可以包括调色板单元、帧内块复制单元(其可以形成运动补偿单元316的

一部分)、仿射单元、线性模型(LM)单元等。在其它示例中,视频解码器300可以包括更多、更少或不同的功能组件。

[0102] CPB存储器320可以存储将由视频解码器300的组件进行解码的视频数据,诸如经编码视频比特流。例如,可以从计算机可读介质110(图1)获得存储在CPB存储器320中的视频数据。CPB存储器320可以包括存储来自经编码视频比特流的经编码视频数据(例如语法元素)的CPB。而且,CPB存储器320可以存储除了经译码图片的语法元素之外的视频数据,诸如表示来自视频解码器300各个单元的输出的临时数据。一般地,DPB 314存储经解码图片,该经解码图片是视频解码器300可以输出的和/或在经编码视频比特流的后续数据或图片进行解码时将其用作参考视频数据。CPB存储器320和DPB 314可以由多种存储设备中的任何一种形成,诸如动态随机存取存储器(DRAM),包括同步DRAM(SDRAM)、磁阻RAM(MRAM)、阻变RAM(RRAM)或者其它类型的存储设备。CPB存储器320和DPB 314可以由相同的存储设备或单独的存储设备提供。在各种示例中,CPB存储器320可以与视频解码器300的其它组件置于片上,或者相对于那些组件置于片外。

[0103] 另外地或可替代地,在一些示例中,视频解码器300可以从存储器120(图1)中取回经译码视频数据。即,存储器120可以存储以上关于CPB存储器320所讨论的数据。同样,当视频解码器300的某些或全部功能在将由视频解码器300的处理电路执行的软件中实现时,存储器120可以存储将由视频解码器300执行的指令。

[0104] 图4中示出的各个单元被示出以帮助理解由视频解码器300执行的操作。这些单元可以实现为固定功能电路、可编程电路或其组合。类似于图3,固定功能电路是指提供特定功能并且预设了能够执行的操作的电路。可编程电路是指可以被编程来执行各种任务并且在能够执行的操作中提供灵活功能的电路。例如,可编程电路可以执行软件或固件,该软件或固件使可编程电路以软件或固件的指令所定义的方式操作。固定功能电路可以执行软件指令(例如来接收参数或输出参数),但是固定功能电路执行的操作类型通常是不可变的。在一些示例中,这些单元中的一个或多个可以是不同的电路块(固定功能或可编程的),并且在一些示例中,一个或多个单元可以是集成电路。

[0105] 视频解码器300可包括ALU、EFU、数字电路、模拟电路和/或由可编程电路形成的可编程内核。在视频解码器300的操作由在可编程电路上执行的软件执行的示例中,片上或片外存储器可以存储视频解码器300接收并执行的软件的指令(例如目标代码)。

[0106] 熵解码单元302可从CPB接收经编码视频数据,并且对视频数据进行熵解码以重现语法元素。预测处理单元304、逆量化单元306、逆变换处理单元308、重构单元310和滤波器单元312可以基于从比特流中提取的语法元素来生成经解码视频数据。

[0107] 当对表示变换块的变换系数的经编码语法元素进行熵解码时,熵解码单元302可被配置为以预定方式扫描一遍变换块。根据本公开的示例,如下面将更详细地描述的,熵解码单元302可被配置为针对将置零技术用于高频系数以降低复杂度的大的变换,执行变换系数扫描技术和最后非零系数位置解码技术。即,在一些示例中,熵解码单元302和/或逆变换处理单元308可以被配置为将变换单元的某个区域中的所有变换系数设置为具有零的值。例如,对于 64×64 变换单元,熵解码单元302和/或逆变换处理单元308可以使变换单元左上角的 32×32 区域中的值(例如,最低频率分量)保持原样。然而,熵解码单元302和/或逆变换处理单元308将把该区域(即,置零区域)之外的所有变换系数设置为具有零的值。

[0108] 在一个示例中,本公开描述了一种扫描技术,其中熵解码单元302被配置为仅扫描变换单元中的在置零区域之外的变换系数。以此方式,可以避免对已知具有零值的变换系数的扫描,从而增加译码器吞吐量。

[0109] 另外,本公开描述了对指示具有置零区域的变换单元中的最后非零变换系数的位置的一个或多个语法元素进行解码的技术。例如,本公开描述了其中熵解码单元302确定用于对指示最后非零变换系数的位置的一个或多个语法元素进行解码的上下文(例如,概率模型)的技术。在一个示例中,熵解码单元302基于变换单元的整个尺寸、而不是基于变换单元的置零区域的尺寸,来确定该上下文。即使最后非零变换系数的位置被保证在非置零区域(例如 64×64 变换单元的左上 32×32 区域)中,但是对指示最后非零变换系数的位置的一个或多个语法元素进行解码的上下文与 64×64 变换单元(而不是 32×32 变换单元)的统计数据更密切相关。因此,可以提高对最后非零变换系数的位置的解码效率。

[0110] 如以下将更详细解释说明的,在本公开的一个示例中,熵解码单元302可以被配置为确定变换单元中变换系数经历置零的第一区域,确定变换单元中变换系数未经历置零的第二区域,以及仅扫描变换单元的第二区域。

[0111] 一般地,视频解码器300在逐块的基础上重构图片。视频解码器300可单独地对每一个块执行重构操作(其中当前正在进行重构(即解码)的块可称为“当前块”)。

[0112] 熵解码单元302可以对定义经量化变换系数块的经量化变换系数、以及诸如量化参数(QP)和/或变换模式指示的变换信息的语法元素进行熵解码。逆量化单元306可以使用与经量化变换系数块相关联的QP来确定量化程度,并且同样地,确定供逆量化单元306应用的逆量化程度。逆量化单元306可以(例如执行按位左移操作)对经量化变换系数进行逆量化。逆量化单元306从而可以形成包括变换系数的变换系数块。

[0113] 在逆量化单元306形成变换系数块之后,逆变换处理单元308可以将一个或多个逆变换应用于变换系数块,来生成与当前块相关联的残差块。例如,逆变换处理单元308可以将逆DCT、逆整数变换、逆Karhunen-Loeve变换(KLT)、逆旋转变换、逆定向变换或另一逆变换应用于变换系数块。

[0114] 此外,预测处理单元304根据由熵解码单元302熵解码的预测信息语法元素来生成预测块。例如,如果预测信息语法元素指示当前块是帧间预测的,则运动补偿单元316可以生成预测块。在这种情况下,预测信息语法元素可以指示DPB 314中的参考图片以从中取回参考块,以及指示标识参考图片中参考块的位置相对于当前图片中的当前块的位置的运动矢量。运动补偿单元316通常可以以与针对运动补偿单元224(图3)所描述的方式基本上相似的方式,来执行帧间预测过程。

[0115] 作为另一示例,如果预测信息语法元素指示当前块是帧内预测的,则帧内预测单元318可以根据由预测信息语法元素指示的帧内预测模式来生成预测块。再次,帧内预测单元318通常可以以与针对帧内预测单元226(图3)所描述的方式基本上相似的方式来执行帧内预测过程。帧内预测单元318可以从DPB 314中取回当前块的相邻样本的数据。

[0116] 重构单元310可以使用预测块和残差块来重构当前块。例如,重构单元310可以将残差块的样本添加到预测块的对应样本来重构当前块。

[0117] 滤波器单元312可以对重构块执行一个或多个滤波器操作。例如,滤波器单元312可以执行去方块操作来减少沿着重构块的边缘的块状伪影。不一定在所有示例中都执行滤

波器单元312的操作。

[0118] 视频解码器300可以将重构块存储在DPB 314中。在不需要滤波器单元312的操作的示例中,重构单元310可以将重构块存储到DPB 314。在需要滤波器单元312的操作的示例中,滤波器单元312可以将滤波后的重构块存储到DPB 314。如上所述,DPB 314可以向预测处理单元304提供参考信息,诸如用于帧内预测的当前图片的样本以及用于随后的运动补偿的先前经解码图片。此外,视频解码器300可以从DPB 314输出经解码图片(例如,经解码视频),用于随后呈现在诸如图1的显示设备118的显示设备上。

[0119] 在Bross等人的Versatile Video Coding (Draft 2)中,其可在http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/11_ljubljana/WG11/jvet-k1001-v7.zip处获得,ITU-T SG 16WP 3和ISO/IEC JTC 1/SC 29WG 11的联合视频专家小组(JVET)第11次会议,2018年7月10日至18日,卢布尔雅那,JVET-K1001,最大的变换单元尺寸为 64×64 。为了降低变换的实现复杂度,VVC Draft 2 (VVC草案2)规定超过32行和列(即,从DC系数开始,超过变换块中的变换系数的前32行和列)的高频系数被设置为零(即,被置零),仅保持较低的 $M \times N$ 区域($M \leq 32, N \leq 32$)中的系数,其中M是样本中区域的宽度,N是样本中区域的高度。

[0120] 在VVC草案2中,视频编码器200和视频解码器300的系数译码模块(例如,熵编码单元220和熵解码单元302)忽略了置零发生的事实,并且将大的变换(在使用置零的情况下)当作常规(即,非置零)变换对待。基本上,视频编码器200和视频解码器300可能不必要地处理已知的置零系数(即,由于上述的置零过程,其值被设置为零的系数)的译码。本公开通过修改用于置零变换系数的系数扫描和通过修改最后非零系数位置译码方法来解决该问题。

[0121] 在VVC中,逐系数组(CG)地从最后一个CG到第一个CG来扫描系数。每个系数组可包括预定数量的变换系数。也在系数组内以反向对角线扫描(左下至右上方向)对系数进行扫描。在一些示例中,视频编码器200和视频解码器300还以反向对角线扫描(例如,左下至右上方向)扫描系数组,如图5中的方框500 (64×64 变换单元)和方框506 (32×64 变换单元)所示。图5是示出示例系数组扫描的概念示意图。相同的扫描可用于变换单元。当然,也可以使用其它扫描方向。

[0122] 从方框500和方框506可以看出,所有系数被扫描,而不管它们是否已经历了置零。例如,在方框500中,视频编码器200和视频解码器300会扫描非置零区域502和置零区域504二者。同样,对于方框506,视频编码器200和视频解码器300会扫描非置零区域508和置零区域510二者。即使扫描从最后非零变换系数(其会处于非置零区域中)的位置开始,针对方框500和506所示的扫描在许多情况下仍会进行到置零区域中。在HEVC和VVC中,对于变换系数译码,术语“最后”是指沿着从DC系数(或最低频系数)到较高频系数的扫描路径存在的最后非零系数。术语“第一”是指系数组(CG内的扫描)或变换单元(TU内的CG扫描)内的DC或最低频系数。系数组的尺寸可以是 4×4 或 2×2 。

[0123] 根据本公开的技术,在新的系数组扫描中,扫描区域仅保持在非置零区域中,如方框512、518、524和530中所示。方框512是 64×64 变换单元,其扫描顺序向右上进行。方框512包括 32×32 非置零区域514。方框512的剩余区域为置零区域516。如上所述,将置零区域中的所有变换系数设置为具有0的值。

[0124] 方框524是 64×64 变换单元,其扫描顺序向左下进行。方框524包括 32×32 非置零

区域526。方框524的剩余区域为置零区域528。

[0125] 方框518是 32×64 变换单元,其扫描顺序向右上进行。方框518包括 32×32 非置零区域520。方框518的剩余区域为置零区域522。

[0126] 方框530是 32×64 变换单元,其扫描顺序向左下进行。方框530包括 32×32 非置零区域532。方框530的剩余区域为置零区域534。

[0127] 从图3的方框512、518、524和530可以看出,不管变换单元的维度和扫描方向如何,视频编码器200和视频解码器300被配置为只在每个相应变换单元的非置零区域(例如,区域514、520、526和532)内执行扫描。

[0128] 也就是说,视频编码器200和视频解码器300可以被配置为确定块/CU/TU中的哪些变换系数已经经历了置零过程。如果变换系数已经经历了置零过程,则视频编码器200和视频解码器300将不会扫描该置零区域中的变换系数。如果变换系数未经历过置零过程(例如,在非置零区域中),则视频编码器200和视频解码器300扫描该变换系数。这种扫描方法消除了置零系数的扫描。在一些示例中,不改变CG内的扫描。

[0129] 因此,在本公开的一个示例中,视频编码器200和视频解码器300被配置为确定变换单元中的第一区域(置零区域),在该第一区域中变换系数经历置零。视频编码器200和视频解码器300可以进一步被配置为确定变换单元中的第二区域(非置零区域),在该第二区域中变换系数未经历置零。然后,视频编码器200和视频解码器300可以仅扫描变换单元的第二区域(非置零区域)。

[0130] 对于在X和Y方向上的最后非零系数位置(例如,变换单元中的(X,Y)坐标)的译码,VVC采用与HEVC相似的方案,除了VVC将该方案扩展到更大的变换单元之外。在HEVC中,最大的变换单元为 32×32 。最后非零系数位置可以被译码为截断的一元译码前缀码(例如,CABAC规则模式)后跟固定长度的译码后缀码(旁路译码)。

[0131] 对于置零变换单元(例如,经历置零的较大变换单元)的最后非零系数位置的译码,视频编码器200和视频解码器300可以被配置为将最大变换单元维度设置为置零阈值(例如,32),而不是64,从而将最后系数位置的前缀码长度缩短到范围24-31。基本上,对维度大于32(例如,一个示例性置零阈值)的变换单元中最后非零系数位置进行译码将遵循用于对维度32(例如, 32×32)而不是64(例如, 64×64)的变换单元中的最后系数位置进行译码的规则。当然,用于将置零应用于变换系数的阈值可以是可变的,并且可以采用32以外的值。

[0132] 因此,根据本公开的技术,视频编码器200和视频解码器300可以被配置为基于用于执行置零过程的阈值,来对变换单元中最后非零系数的位置进行编码/解码。在一个示例中,为了基于用于执行置零过程的阈值来对变换单元中最后非零系数的位置进行编码/解码,视频编码器200和视频解码器300可以被配置为基于用于执行置零过程的阈值,对表示最后非零系数的位置的第一部分的前缀码进行编码/解码,并且对表示最后非零系数的位置的第二部分的后缀码进行编码/解码。

[0133] 在对在指定维度(例如,在32宽度或高度)处最大的、已置零变换单元的最后非零系数位置进行译码的另一示例中,视频编码器200和视频解码器300可将用于对最后非零系数位置进行译码的熵译码上下文确定为用于对64维度块(例如, 64×64)的最后非零系数位置进行译码的上下文。也就是,视频编码器200和视频解码器300可以基于变换单元的尺寸

而不是非置零区域的尺寸,来确定用于对最后非零系数的位置进行译码的上下文。即使最后非零变换系数的位置被保证在非置零区域(例如 64×64 变换单元的左上 32×32 区域)中,但是用于对最后非零变换系数的位置进行译码的上下文与 64×64 变换单元(而不是 32×32 变换单元)的统计数据更密切相关。因此,可以提高对最后非零变换系数的位置的译码效率。因此,在本公开的一个示例中,视频编码器200和视频解码器300可以被配置为基于变换单元的整个尺寸来确定用于对最后非零系数的位置进行熵译码的上下文。

[0134] 用于常规(非置零) 32×32 块的最后非零系数位置的上下文和用于已置零的 64×64 块的 32×32 区域最后非零系数位置的上下文在该实例中仍然可以是单独的上下文集合,并且类似地,用于其它大于32维度的变换块(例如,高度或宽度为32的变换块)也可以是单独的上下文集合。在其它示例中,用于对最后非零系数位置进行译码的上下文可以在最大维度32块(例如,高度或宽度最大为32的变换块)和最大维度64块(例如,高度或宽度最大为64的变换块)之间共享。例如,用于常规(非置零) 32×32 块的最后非零系数位置的上下文和用于已置零的 64×64 块的 32×32 区域最后非零系数位置的上下文可以共享相同的上下文集合。

[0135] 在另一示例中,作为仅扫描非置零区域的另一替代方案,大的扫描可以保持原样,但是视频编码器200和视频解码器300可以被配置为强制经编码比特流在大的逆变换的置零区域中仅具有0系数水平。在这种情况下,最后非零系数位置将被强制在左上非置零区域中,并且如上所述仍然被限制为32的最大非零尺寸。

[0136] 上述一种或多种技术可以扩展到 $N \times M$ 、 $M \times N$ 和 $N \times N$ 变换尺寸中的任何一种,其中 N 是变换维度,其中分别针对 x 维度(即,水平维度)或 y 维度(即,垂直维度)或 x 和 y 维度二者,从 $N/2$ 到 $N-1$ 应用置零。仅扫描非置零区域中的系数,并且最后非零系数位置译码将针对非置零区域的尺寸。然而,如上所述,所使用的上下文会是用于全尺寸变换单元的上下文。例如,如果对 32×32 变换应用置零,则会仅保留频率较低的 16×16 区域(例如 32×32 变换单元的左上 16×16 区域),而该区域之外的任何系数将被置零。视频编码器200和视频解码器300将仅扫描 16×16 非置零区域,并且最后非零系数位置译码将对 16×16 变换单元使用二值化。然而,视频编码器200和视频解码器300将基于 32×32 变换单元示例,来确定用于最后非零系数的位置的CABAC上下文。

[0137] 因此,在本公开的其它示例中,为了确定变换单元中变换系数经历置零的第一区域,视频编码器200和视频解码器300可以被配置为基于变换系数在变换单元中的位置来确定变换单元中变换系数经历置零的第一区域。

[0138] 在本公开的另一示例中,为了基于变换系数在变换单元中的位置来确定变换单元中变换系数经历置零的第一区域,视频编码器200和视频解码器300可以被配置为确定第一区域包括变换单元中的超过该变换单元的前32行或列的经历置零的变换系数。

[0139] 在本公开的另一示例中,为了仅扫描变换单元的第二区域(非置零区域),视频编码器200和视频解码器300可以被配置为从最后非零系数的位置开始,仅扫描变换单元的第二区域。

[0140] 在本公开的另一示例中,为了仅扫描变换单元的第二区域,视频编码器200和视频解码器300可被配置为逐系数组地,仅扫描变换单元的第二区域。

[0141] 图6是示出用于对当前块进行编码的示例方法的流程图。当前块可以包括当前CU。

尽管是关于视频编码器200(图1和图3)描述的,但应当理解的是,其它设备可以被配置为执行类似于图6的方法。

[0142] 在此示例中,视频编码器200最初预测当前块(350)。例如,视频编码器200可以形成当前块的预测块。然后,视频编码器200可以计算当前块的残差块(352)。为了计算残差块,视频编码器200可以计算原始的未经译码块与当前块的预测块之间的差异。然后,视频编码器200可以变换和量化残差块的系数(354)。接下来,视频编码器200可以扫描残差块的经量化变换系数(356)。

[0143] 根据本公开的技术,为了扫描变换系数,视频编码器200可以被配置为确定变换单元中变换系数经历置零的第一区域(362),确定变换单元中变换系数未经历置零的第二区域(364),以及仅扫描变换单元的第二区域(366)。在扫描期间或在扫描之后,视频编码器200可以对系数进行熵编码(358)。视频编码器200还可以对预测信息进行熵编码。根据本公开的技术,视频编码器200可以被配置为基于变换单元的整个尺寸,来确定用于对最后非零系数的位置进行熵编码的上下文。视频编码器200可以使用CAVLC或CABAC对系数进行编码。然后,视频编码器200可以输出当前块的经熵译码数据(360)。例如,视频编码器200可以输出对应于当前块的残差块的变换系数的经熵译码数据。

[0144] 图7是示出用于对视频数据的当前块进行解码的示例方法的流程图。当前块可以包括当前CU。尽管是关于视频编码器300(图1和图4)描述的,但应当理解的是,其它设备可以被配置为执行类似于图7的方法。

[0145] 视频解码器300可以接收当前块的经熵译码数据,诸如与当前块相对应的经熵译码的预测信息和残差块的系数的经熵译码数据(370)。视频解码器300可以对经熵译码数据进行解码,以确定当前块的预测信息并重现残差块的系数(372)。根据本公开的技术,为了重现变换系数,视频解码器300可以被配置为确定变换单元中变换系数经历置零的第一区域(382),确定变换单元中变换系数未经历置零的第二区域(384),以及仅扫描变换单元的第二区域(386)。根据本公开的技术,视频解码器300可以进一步被配置为基于变换单元的整个尺寸,来确定用于对指示最后非零系数的位置的一个或多个语法元素进行熵解码的上下文。

[0146] 视频解码器300可以预测当前块(374),例如,通过使用当前块的预测信息所指示的帧内或帧间预测模式,以计算当前块的预测块。然后,视频解码器300可以逆扫描所重现的系数(376),以创建经量化变换系数的块。然后,视频解码器300可以对系数进行逆量化和逆变换以产生残差块(378)。视频解码器300可以通过组合预测块和残差块来最终解码当前块(380)。

[0147] 应该认识到,取决于示例,本文中描述的任何技术的某些动作或事件可以以不同的序列执行,可以被一起添加、合并或省去(例如,不是所有描述的动作或事件是技术实践所必须的)。此外,在某些示例中,动作或事件可以例如通过多线程处理、中断处理或多个处理器并发地处理而不是顺序地执行。

[0148] 在一个或多个示例中,可以以硬件、软件、固件或其任意组合来实现所描述的功能。如果以软件实现,则功能可以作为一个或多个指令或代码存储在计算机可读介质上或通过计算机可读介质传输,并由基于硬件的处理单元执行。计算机可读介质可以包括计算机可读存储介质,其对应于诸如数据存储介质之类的有形介质,或者通信介质,包括例如根

据通信协议来促进将计算机程序从一个地方转移到另一个地方的任何介质。以这种方式,计算机可读介质通常可以对应于(1)非暂时性的有形计算机可读存储介质,或者(2)诸如信号或载波之类的通信介质。数据存储介质可以是由一个或多个计算机或一个或多个处理器访问以取回指令、代码和/或数据结构以实现本公开中描述的技术的任何可用介质。计算机程序产品可以包括计算机可读介质。

[0149] 作为示例而非限制,这种计算机可读存储介质可以包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储、磁盘存储或其它磁性存储设备、闪存或可以用于以指令或数据结构形式存储所需程序代码并且可以由计算机访问的任何其它介质中的一个或多个。而且,任何连接都适当地称为计算机可读介质。例如,如果使用同轴电缆、光纤电缆、双绞线、数字用户线(DSL)或无线技术(诸如红外、无线电和微波)从网站、服务器或其它远程源发送指令,则介质的定义包括同轴电缆、光纤电缆、双绞线、DSL或诸如红外、无线电和微波之类的无线技术。然而,应当理解,计算机可读存储介质和数据存储介质不包括连接、载波、信号或其它暂时性介质,而是针对非暂时性有形存储介质。如本申请中使用的磁盘和光盘包括压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软盘和蓝光光盘,其中,磁盘通常以磁性方式重现数据,而光盘用激光光学地重现数据。上述的组合也应包括在计算机可读介质的范围内。

[0150] 指令可以由一个或多个处理器执行,诸如一个或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或其它等效集成的或分立逻辑电路。因此,如本申请中所使用的术语“处理器”可以是指任何前述结构或适合于实现本申请中描述的技术的任何其它结构。另外,在一些方面,本申请中描述的功能可以在被配置用于编码和解码的专用硬件和/或软件模块内提供,或合并到组合编解码器中。同样,该技术可以全部在一个或多个电路或逻辑元件中实现。

[0151] 本公开的技术可以在包括无线手机、集成电路(IC)或一组IC(例如,芯片组)的多种设备或装置中实现。在本公开中描述各种组件、模块或单元以强调被配置为执行所公开技术的设备的功能方面,但不一定需要由不同硬件单元来实现。而是,如上所述,各种单元可以组合在编解码器硬件单元中,或者由互操作硬件单元的集合来提供,包括与合适的软件和/或固件结合的一个或多个处理器。

[0152] 已经描述了各种示例。这些和其它示例在所附权利要求的范围内。

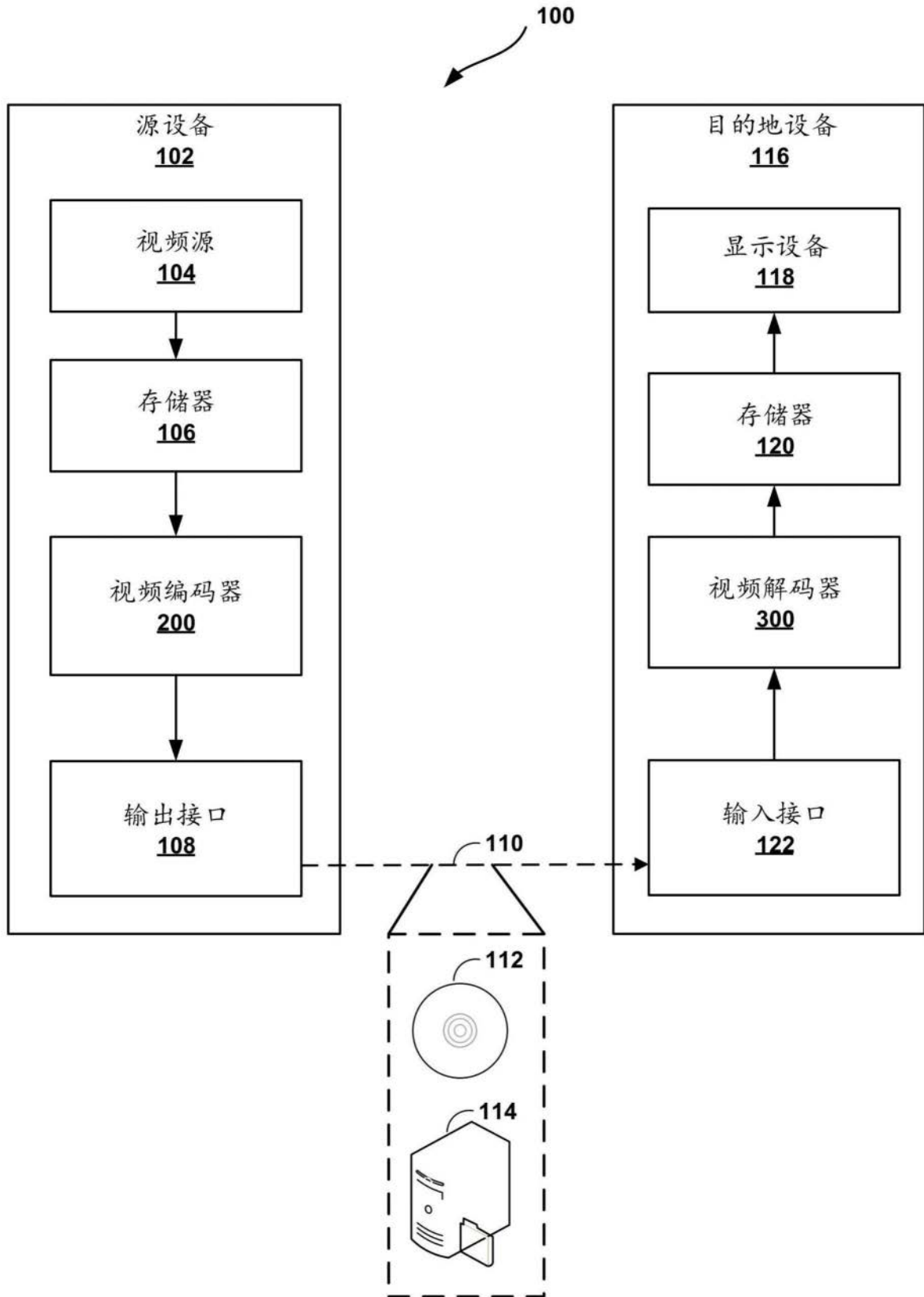


图1

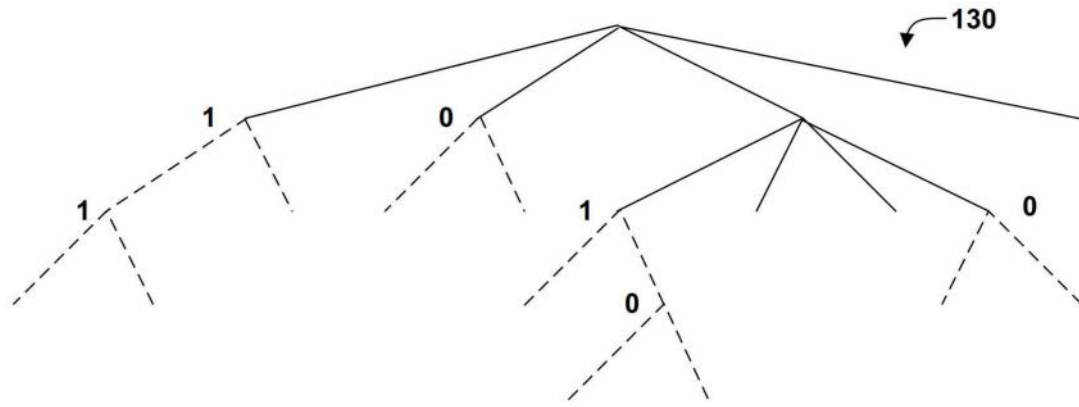


图2A

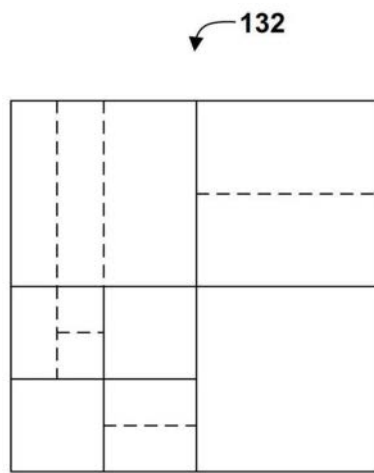


图2B

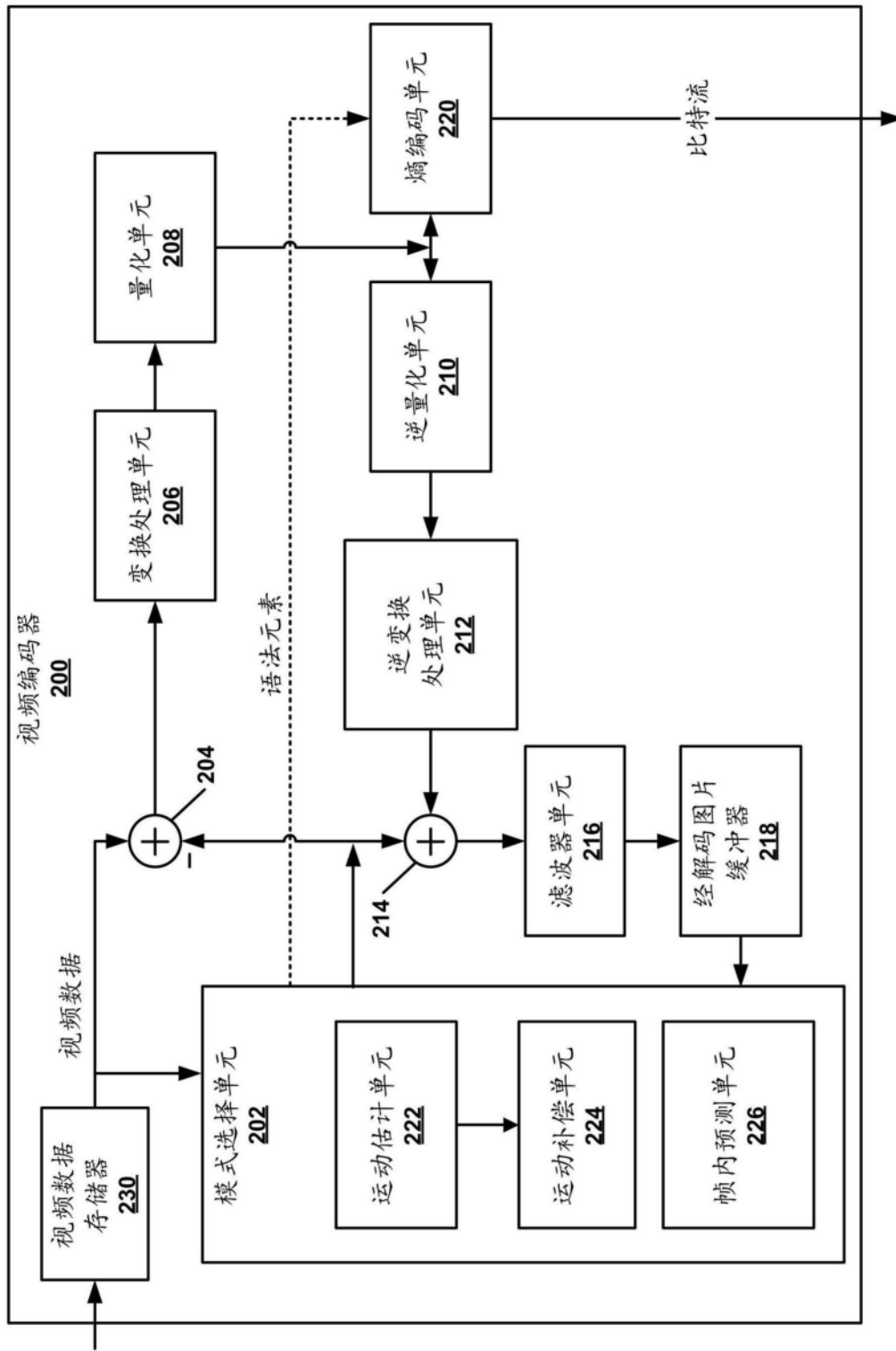


图3

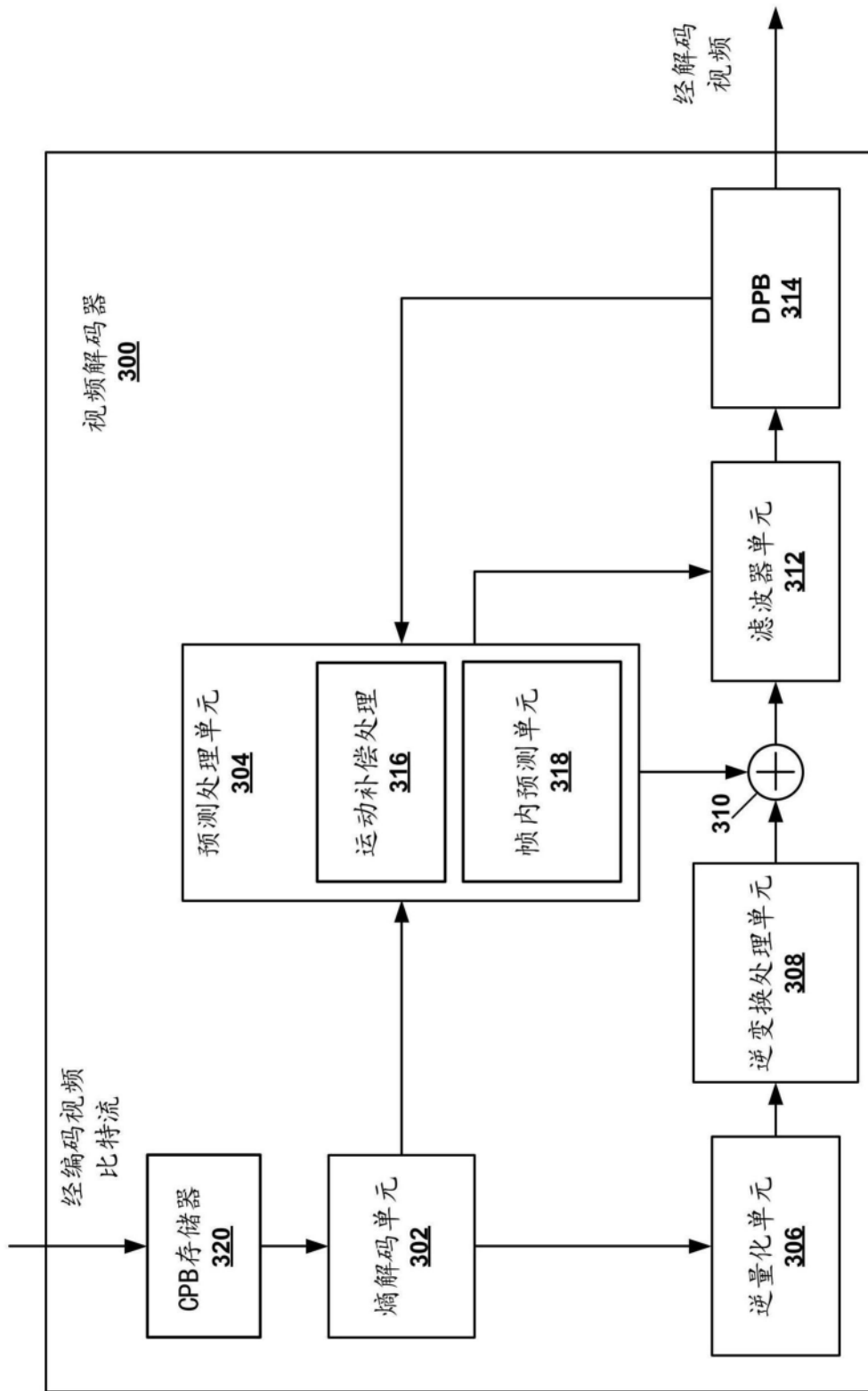


图4

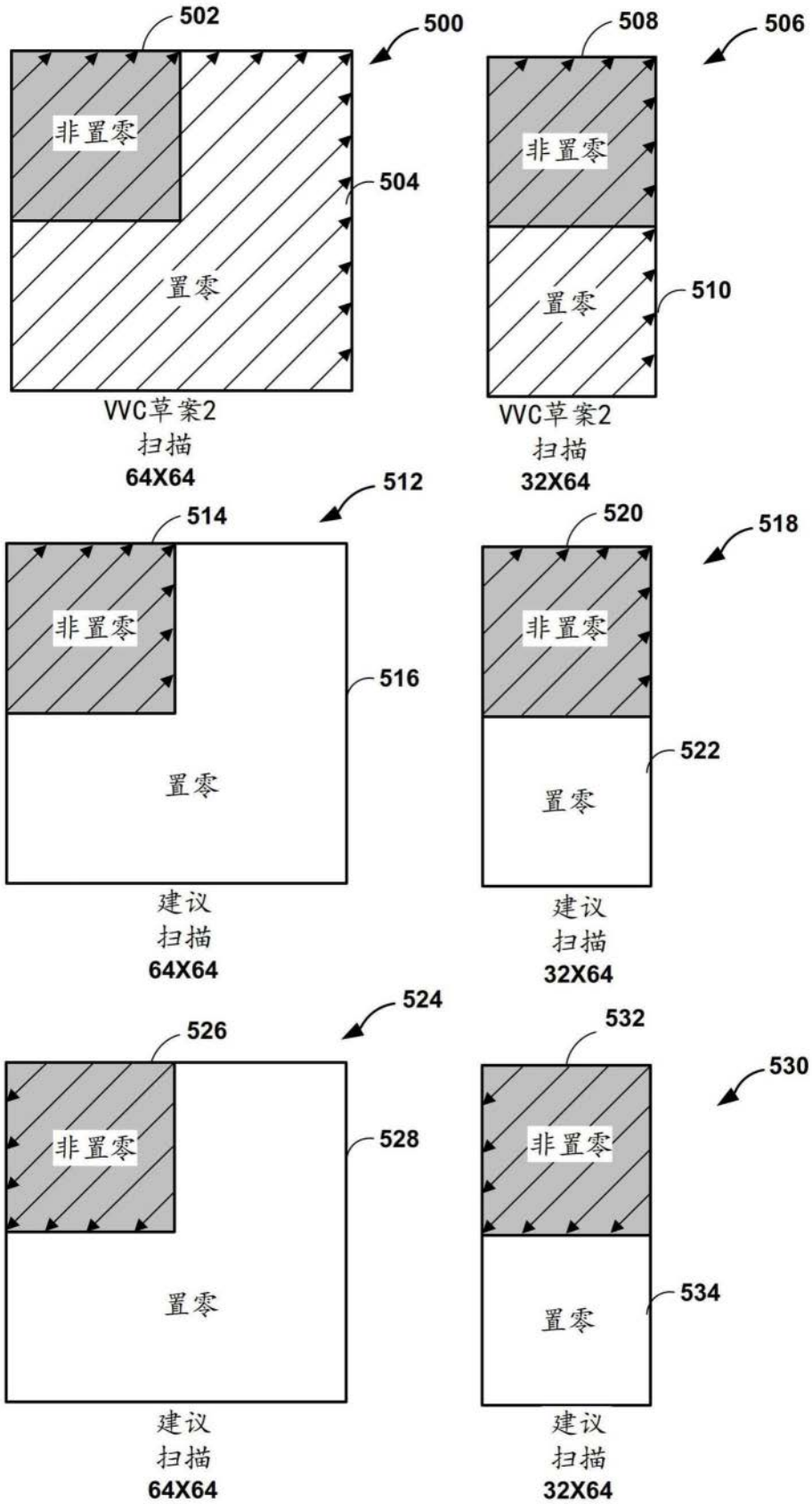


图5

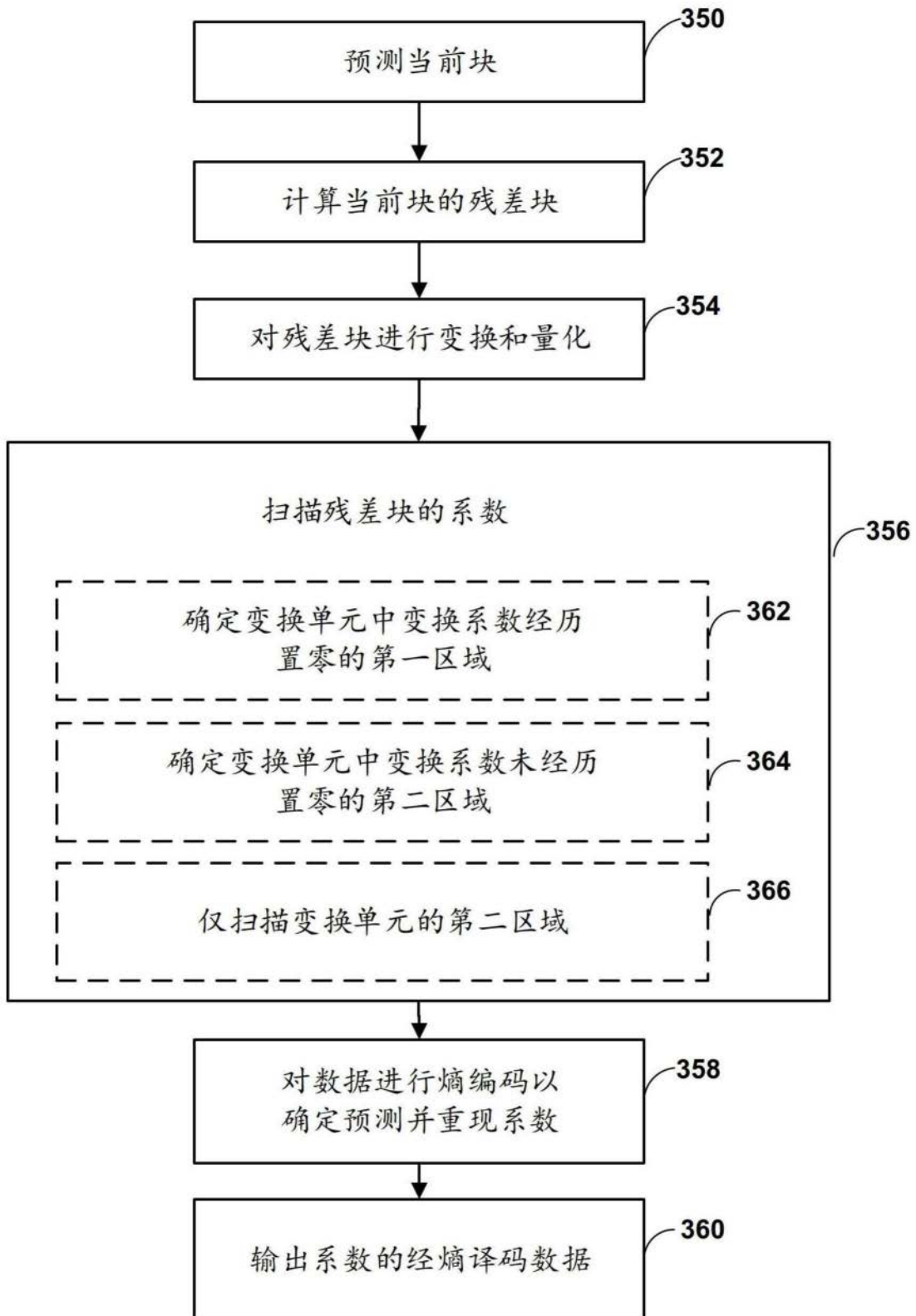


图6

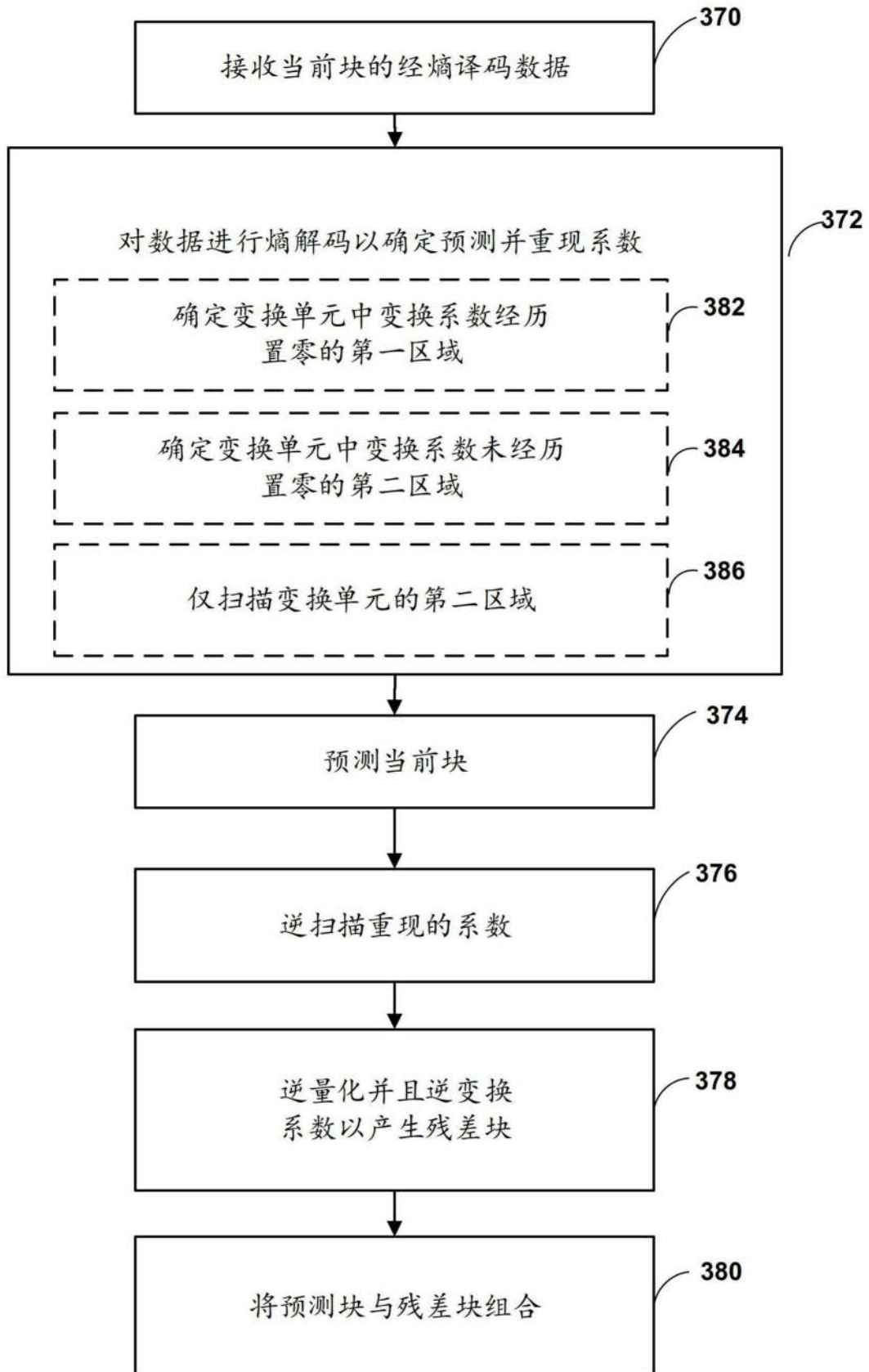


图7