



## (12)发明专利

(10)授权公告号 CN 104054347 B

(45)授权公告日 2018.08.28

(21)申请号 201280067184.2

(22)申请日 2012.12.19

(65)同一申请的已公布的文献号

申请公布号 CN 104054347 A

(43)申请公布日 2014.09.17

(30)优先权数据

61/588,096 2012.01.18 US

13/718,883 2012.12.18 US

(85)PCT国际申请进入国家阶段日

2014.07.16

(86)PCT国际申请的申请数据

PCT/US2012/070680 2012.12.19

(87)PCT国际申请的公布数据

W02013/109382 EN 2013.07.25

(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 王益魁 穆哈默德·蔡德·科班

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.

H04N 19/46(2014.01)

H04N 19/463(2014.01)

审查员 肖东

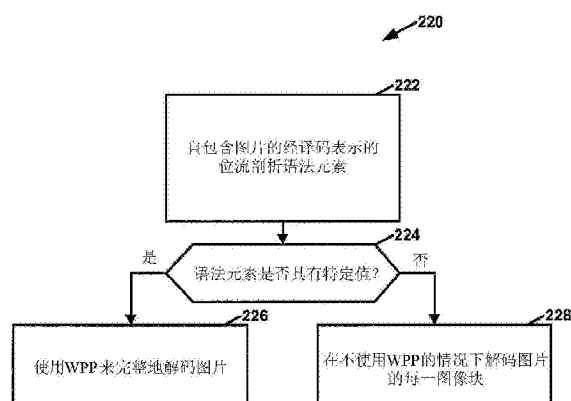
权利要求书4页 说明书31页 附图12页

### (54)发明名称

在视频译码中波前平行处理的使用的指示

### (57)摘要

一种视频编码器产生包含语法元素的位流，所述语法元素指示是根据第一译码模式还是第二译码模式来编码图片。在所述第一译码模式中，使用波前平行处理WPP来完整地编码所述图片。在所述第二译码模式中，在不使用WPP的情况下编码所述图片的每一图像块，且所述图片可具有一或多个图像块。视频解码器可从所述位流剖析所述语法元素。响应于确定所述语法元素具有特定值，所述视频解码器使用WPP来完整地解码所述图片。响应于确定所述语法元素不具有所述特定值，所述视频解码器在不使用WPP的情况下解码所述图片的每一图像块。



1. 一种用于解码视频数据的方法,所述方法包括:

从位流剖析语法元素,所述位流包括所述视频数据中的图片的切片的经译码切片网络抽象层NAL单元,所述图片被分割成译码树块CTB的栅格,每一个CTB与所述图片内的大小相等的不同像素块相关联,所述经译码切片NAL单元包括多个子流,其中,当所述语法元素具有特定值时,其指示CTB的每一行由所述子流中的单一者来表示,而当所述语法元素不具有所述特定值时,其指示具有所述切片中的一个或多个CTB的图片的每一图像块由所述子流中的单一者来表示;

基于所述语法元素,确定所述子流是否包含确保所述子流的字节对准的填补位;

响应于确定所述语法元素具有所述特定值,使用波前平行处理WPP来完整地解码所述图片,其中,使用WPP解码包括自左至右对所述图片的行的CTB进行解码,该解码是在已对所述图片的紧接较高行的两个或多个CTB解码之后进行的;以及

其中,当所述语法元素不具有所述特定值时,所述图片具有一个或多个图像块,且在不使用WPP的情况下解码所述图片的每一图像块。

2. 根据权利要求1所述的方法,其进一步包括:

从所述位流剖析第二语法元素,所述位流包括所述视频数据的第二图片的经译码表示,所述第二图片被分割成CTB的栅格,其中,所述第二图片被分割成至少第一图像块和第二图像块;且

响应于确定所述第二语法元素不具有特定值,并行地解码所述第一图像块的CTB和所述第二图像块的CTB。

3. 根据权利要求1所述的方法,其进一步包括:

确定参数集包含图像块列数语法元素和图像块行数语法元素;

基于所述图像块列数语法元素而确定图像块列的指定数目,其中与所述参数集相关联的每一图片的图像块列的数目等于图像块列的所述指定数目;以及

基于所述图像块行数语法元素而确定图像块行的指定数目,其中与所述参数集相关联的每一图片的图像块行的数目等于图像块行的所述指定数目。

4. 根据权利要求1所述的方法,其中所述语法元素为第一语法元素,所述特定值为第一值,且所述方法进一步包括基于具有第二值的所述第一语法元素而确定所述图片包含仅一个图像块且所述位流包含第二语法元素,所述第二语法元素指示针对参考包含所述第一语法元素和所述第二语法元素的参数集的所述图片的经编码表示是否启用熵切片。

5. 根据权利要求1所述的方法,其中使用WPP解码所述图片包括:

响应于确定第一CTB与所述图片的左边界分离单一CTB,存储与所述第一CTB相关联的上下文变量;以及

至少部分基于与所述第一CTB相关联的所述上下文变量而熵解码第二CTB的一或多个语法元素,所述第二CTB邻近于所述图片的所述左边界且比所述第一CTB低一行CTB。

6. 一种用于编码视频数据的方法,所述方法包括

在位流中产生视频数据的图片的切片的经译码切片网络抽象层NAL单元,所述图片被分割成译码树块CTB的栅格,每一个CTB与所述图片内大小相等的不同像素块相关联,所述经译码切片NAL单元包括多个子流;

在所述位流中产生语法元素;

其中,具有特定值的所述语法元素指示了由所述子流中的单一者来表示所述图片的CTB的每一行且使用波前平行处理WPP来完整地编码所述图片,且其中使用WPP编码包括自左至右对所述图片的行的CTB进行编码,该编码是在已对所述图片的紧接较高行的两个或多个CTB编码之后进行的;

其中,不具有所述特定值的所述语法元素指示了所述图片具有一个或多个图像块且具有所述切片中的一或多个CTB的图片的每一图像块由所述子流的单一者来表示,且在不使用WPP的情况下编码所述图片的每一图像块。

7. 根据权利要求6所述的方法,进一步包括:

在所述位流中包括第二语法元素,所述位流包括所述视频数据的第二图片的经译码表示,所述第二图片被分割成CTB的栅格,所述第二语法元素不具有所述特定值,其中,所述第二图片被分割成至少第一图像块和第二图像块,及

并行地编码所述第一图像块的CTB和所述第二图像块的CTB。

8. 根据权利要求6所述的方法,其中:

产生所述位流包括产生包含图像块列数语法元素和图像块行数语法元素的参数集,

图像块列的数目可基于所述图像块列数语法元素而确定,且与所述参数集相关联的每一图片的图像块列的数目等于图像块列的所述数目,且

图像块行的数目可基于所述图像块行数语法元素而确定,且与所述参数集相关联的每一图片的图像块行的数目等于图像块行的所述数目。

9. 根据权利要求6所述的方法,其中所述语法元素为第一语法元素,所述特定值为第一值,所述第一语法元素具有指示所述图片被分割成单一图像块的第二值,且所述位流包含第二语法元素,所述第二语法元素指示针对参考包含所述第一语法元素和所述第二语法元素的参数集的所述图片的经编码表示是否启用熵切片。

10. 根据权利要求6所述的方法,其进一步包括使用WPP来编码所述图片,其中使用WPP来编码所述图片包括:

响应于确定第一CTB与所述图片的左边界分离单一CTB,存储与所述第一CTB相关联的上下文变量;以及

至少部分基于与所述第一CTB相关联的所述上下文变量而熵编码第二CTB的一或多个语法元素,所述第二CTB邻近于所述图片的所述左边界且比所述第一CTB低一行CTB。

11. 一种视频解码装置,其包括:

用于从位流解析语法元素的装置,所述位流包括视频数据中的图片的切片的经译码切片网络抽象层NAL单元,所述图片被分割成译码树块CTB的栅格,每一个CTB与所述图片内的大小相等的不同像素块相关联,所经译码切片NAL单元包括多个子流,其中,当所述语法元素具有特定值时,其指示由所述子流中的单一者来表示CTB的每一行,而当所述语法元素不具有所述特定值时,其指示由所述子流中的单一者来表示具有所述切片中一个或多个CTB的图片的每一图像块;

用于基于所述语法元素确定所述子流是否包含确保所述子流的字节对准的填补位的装置;

用于响应于确定所述语法元素具有所述特定值,使用波前平行处理WPP来完整地解码所述图片的装置,其中,使用WPP解码包括自左至右对所述图片的行的CTB进行解码,该解码

是在已对所述图片的紧接较高行的两个或多个CTB解码之后进行的;以及

其中,当所述语法元素不具有所述特定值时,所述图片具有一个或更多个图像块,且在不使用WPP的情况下解码所述图片的每一图像块。

12.根据权利要求11所述的视频解码装置,其包括用于执行根据权利要求2-5的方法中任一方法的装置。

13.一种视频编码装置,其包括:

用于在位流中产生所述视频数据中的图片的切片的经译码切片网络抽象层NAL单元的装置,所述图片被分割成译码树块CTB的栅格,每一个CTB与所述图片内的大小相等的不同像素块相关联,所述经译码切片NAL单元包括多个子流;

用于在所述位流中产生语法元素的装置;

其中,具有特定值的所述语法元素指示由所述子流中的单一者来表示CTB的每一行,所述子流包括确保字节对准的填补位且使用波前平行处理WPP来完整地编码所述图片,且其中使用WPP编码包括自左至右对所述图片的行的CTB进行编码,该编码是在已对所述图片的紧接较高行的两个或多个CTB编码之后进行的;

其中,不具有所述特定值的所述语法元素指示了所述图片具有一个或多个图像块且具有所述切片中的一或多个CTB的图片的每一图像块由所述子流的单一者来表示,且在不使用WPP的情况下编码所述图片的每一图像块。

14.根据权利要求13所述的视频编码装置,包括用于执行根据权利要求7-10中任一权利要求所述的方法的装置。

15.一种存储指令的计算机可读存储媒体,所述指令在由视频解码装置的一或多个处理器执行时配置所述视频解码装置以:

从位流解析语法元素,所述位流包括视频数据中的图片的切片的经译码切片网络抽象层NAL单元,所述图片被分割成译码树块CTB的栅格,每一个CTB与所述图片内的大小相等的不同像素块相关联,所述经译码切片NAL单元包括多个子流,其中,当所述语法元素具有特定值时,其指示由所述子流中的单一者来表示CTB的每一行,而当所述语法元素不具有所述特定值时,其指示由所述子流中的单一者来表示具有所述切片中的一个或多个CTB的图片的每一图像块;

基于所述语法元素,确定所述子流是否包含确保所述子流的字节对准的填补位;

响应于确定所述语法元素具有所述特定值,使用波前平行处理WPP来完整地解码所述图片,其中,使用WPP解码包括自左至右对所述图片的行的CTB进行解码,该解码是在已对所述图片的紧接较高行的两个或多个CTB解码之后进行的;以及

其中,当所述语法元素不具有所述特定值时,所述图片具有一个或多个图像块,且在不使用WPP的情况下解码所述图片的每一图像块。

16.一种存储指令的计算机可读存储媒体,所述指令在由视频编码装置的一或多个处理器执行时配置所述视频编码装置以

在位流中产生视频数据中的图片的切片的经译码切片网络抽象层NAL单元,所述图片被分割成译码树块CTB的栅格,每一个CTB与所述图片内的大小相等的不同像素块相关联,所述经译码切片NAL单元包括多个子流;

在所述位流中产生语法元素;

其中,具有特定值的所述语法元素指示由所述子流中的单一者来表示所述图片的CTB的每一行且使用波前平行处理WPP来完整地编码所述图片,且其中使用WPP编码包括自左至右对所述图片的行的CTB进行编码,该编码是在已对所述图片的紧接较高行的两个或多个CTB编码之后进行的;

其中,所述语法元素不具有所述特定值指示了所述图片具有一个或多个图像块且具有所述切片中的一或多个CTB的图片的每一图像块由所述子流的单一者来表示,且在不使用WPP的情况下编码所述图片的每一图像块。

## 在视频译码中波前平行处理的使用的指示

[0001] 本申请案主张于2012年1月18日申请的美国临时专利申请案第61/588,096号的权利,所述申请案的全部内容在此以引用的方式并入。

### 技术领域

[0002] 本发明涉及视频译码(即,视频数据的编码和/或解码)。

### 背景技术

[0003] 可将数字视频能力并入到广泛范围的装置中,所述装置包含数字电视、数字直播系统、无线广播系统、个人数字助理(PDA)、膝上型计算机或桌上型计算机、平板计算机、电子书阅读器、数码相机、数字记录装置、数字媒体播放器、视频游戏装置、视频游戏控制台、蜂窝式或卫星无线电电话、所谓的“智能电话”、视频电话会议装置、视频流式传输装置和类似者。数字视频装置实施视频压缩技术,例如在由MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4第10部分(先进视频译码(AVC))定义的标准、目前在开发中的高效率视频译码(HEVC)标准和这些标准的扩展中所描述的视频压缩技术。视频装置可通过实施这些视频压缩技术来更有效率地发射、接收、编码、解码和/或存储数字视频信息。

[0004] 视频压缩技术执行空间(图片内)预测和/或时间(图片间)预测以减少或去除视频序列中所固有的冗余。对于基于块的视频译码,可将视频切片(即,视频帧或视频帧的一部分)分割成视频块,视频块也可被称作树块、译码单元(CU)和/或译码节点。图片的经帧内译码(I)切片中的视频块是使用相对于相同图片中的相邻块中的参考样本的空间预测来编码。图片的经帧间译码(P或B)切片中的视频块可使用相对于相同图片中的相邻块中的参考样本的空间预测或相对于其它参考图片中的参考样本的时间预测。图片可被称作帧,且参考图片可被称作参考帧。

[0005] 空间或时间预测产生待译码的块的预测性块。残余数据表示待译码的原始块与预测性块之间的像素差。根据指向形成预测性块的参考样本的块的运动向量和指示经译码块与预测性块之间的差的残余数据来编码经帧间译码块。根据帧内译码模式和残余数据来编码经帧内译码块。为了进一步压缩,可将残余数据自像素域变换到变换域,从而产生残余系数,所述残余系数接着可经量化。可按次序扫描最初布置成二维阵列的经量化系数以便产生系数的一维向量,且可应用熵译码以达成甚至更多压缩。

### 发明内容

[0006] 大体来说,本发明描述用于视频译码的技术,其中不允许单一图片内的图像块与波前平行处理(WPP)的组合。更具体来说,视频编码器产生包含语法元素的位流,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用WPP来完整地编码所述图片。在所述第二译码模式中,在不使用WPP的情况下编码所述图片的每一图像块。视频解码器自所述位流剖析所述语法元素,且确定所述语法元素是否具有特定值。响应于确定所述语法元素具有所述特定值,所述视频解码器使用WPP来完整地解码

所述图片。响应于确定所述语法元素不具有所述特定值,所述视频解码器在不使用WPP的情况下解码所述图片的每一图像块。

[0007] 在一方面,本发明描述一种用于解码视频数据的方法。所述方法包括自包含所述视频数据中的图片的经译码表示的位流剖析语法元素。另外,所述方法包括响应于确定所述语法元素具有特定值,使用WPP来完整地解码所述图片。所述方法也包括响应于确定所述语法元素不具有所述特定值,在不使用WPP的情况下解码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0008] 在另一方面,本发明描述一种用于编码视频数据的方法。所述方法包括产生包含语法元素的位流,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用WPP来完整地编码所述图片。在所述第二译码模式中,在不使用WPP的情况下编码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0009] 在另一方面,本发明描述一种包括一或多个处理器的视频解码装置,所述一或多个处理器经配置以自包含视频数据中的图片的经译码表示的位流剖析语法元素。所述一或多个处理器经配置以响应于确定所述语法元素具有特定值,使用WPP来完整地解码所述图片。另外,所述一或多个处理器经配置以响应于确定所述语法元素不具有所述特定值,在不使用WPP的情况下解码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0010] 在另一方面,本发明描述一种包括一或多个处理器的视频编码装置,所述一或多个处理器经配置以产生包含语法元素的位流,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用WPP来完整地编码所述图片。在所述第二译码模式中,在不使用WPP的情况下编码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0011] 在另一方面,本发明描述一种视频解码装置,其包括用于自包含视频数据中的图片的经译码表示的位流剖析语法元素的装置。所述视频解码装置也包括用于响应于确定所述语法元素具有特定值而使用WPP来完整地解码所述图片的装置。另外,所述视频解码装置包括用于响应于确定所述语法元素不具有所述特定值而在不使用WPP的情况下解码所述图片的每一图像块的装置,其中所述图片具有一或多个图像块。

[0012] 在另一方面,本发明描述一种视频编码装置,其包括用于产生包含语法元素的位流的装置,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用WPP来完整地编码所述图片。在所述第二译码模式中,在不使用WPP的情况下编码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0013] 在另一方面,本发明描述一种存储指令的计算机可读存储媒体,所述指令在由视频解码装置的一或多个处理器执行时,配置所述视频解码装置以自视频数据中的图片的经译码表示的位流剖析语法元素。所述指令也使所述视频解码装置响应于确定所述语法元素具有特定值,使用WPP来完整地解码所述图片。另外,所述指令使所述视频解码装置响应于确定所述语法元素不具有所述特定值,在不使用WPP的情况下解码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0014] 在另一方面,一种存储指令的计算机可读存储媒体,所述指令在由视频编码装置的一或多个处理器执行时,配置所述视频编码装置以产生包含语法元素的位流,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用

WPP来完整地编码所述图片。在所述第二译码模式中,在不使用WPP的情况下编码所述图片的每一图像块,其中所述图片具有一或多个图像块。

[0015] 本发明的一或多个实例的细节陈述于随附图式和以下描述中。其它特征、目标和优势将自描述、图式和权利要求书而显而易见。

## 附图说明

[0016] 图1为说明可利用本发明中所描述的技术的实例视频译码系统的框图。

[0017] 图2为说明可实施本发明中所描述的技术的实例视频编码器的框图。

[0018] 图3为说明可实施本发明中所描述的技术的实例视频解码器的框图。

[0019] 图4为说明根据本发明的一或多个方面的用于编码视频数据的视频编码器的实例操作的流程图,其中不允许单一图片内的图像块与波前平行处理(WPP)的组合。

[0020] 图5为说明根据本发明的一或多个方面的用于解码视频数据的视频解码器的实例操作的流程图,其中不允许单一图片内的图像块与WPP的组合。

[0021] 图6为说明根据本发明的一或多个方面的用于解码视频数据的视频解码器的另一实例操作的流程图,其中不允许单一图片内的图像块与WPP的组合。

[0022] 图7为说明根据本发明的一或多个方面的用于编码视频数据的视频编码器的实例操作的流程图,其中图片的译码树块(CTB)的每一行在单独子流中。

[0023] 图8为说明根据本发明的一或多个方面的用于解码视频数据的视频解码器的实例操作的流程图,其中图片的CTB的每一行在单独子流中。

[0024] 图9A为说明根据本发明的一或多个方面的剖析切片数据的实例上下文自适应性二进制算术译码(CABAC)剖析过程的第一部分的流程图。

[0025] 图9B为说明图9A的实例CABAC剖析过程的接续部分的流程图。

[0026] 图10为说明WPP的实例的概念图。

[0027] 图11为说明在将图片分割成多个图像块时的实例译码次序的概念图。

## 具体实施方式

[0028] 在视频译码期间,可将图片分割成多个图像块、波前平行处理(WPP)波和/或熵切片。由穿过图片的水平和/或垂直图像块边界定义图片的图像块。根据光栅扫描次序译码图片的图像块,且也根据光栅扫描次序译码每一图像块内的译码树块(CTB)。在WPP中,图片中的CTB的每一行为“WPP波”。在视频译码器使用WPP来译码图片时,视频译码器可在视频译码器已译码紧接的较高WPP波的两个或两个以上CTB之后,开始自左到右译码WPP波的CTB。熵切片可包含根据光栅扫描次序的一系列连续CTB。来自跨越1切片边界的信息的使用被禁止用于选择熵译码上下文,但可被允许用于达成其它目的。

[0029] 在现有视频译码系统中,图片可具有图像块、WPP波和熵切片的任何组合。例如,可将图片分割成多个图像块。在此实例中,可根据光栅扫描次序译码图像块中的一些图像块中的CTB,而可使用WPP来译码图像块中的其它图像块中的CTB。允许图片包含图像块、WPP波和熵切片的组合可不必要地增加这些视频译码系统的实施方案复杂度和成本。

[0030] 本发明的技术可解决此问题。即,根据本发明的技术,不允许在图片内的两个或两个以上图像块、WPP波和熵切片中的任一者的组合。例如,视频编码器可产生包含语法元素

的位流,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用WPP来完整地编码所述图片。在所述第二译码模式中,所述图片具有一或多个图像块,且在不使用WPP的情况下编码所述图片的每一图像块。

[0031] 此外,在此实例中,视频解码器可自包含图片的经译码表示的位流剖析语法元素。响应于确定所述语法元素具有特定值,所述视频解码器可使用WPP来完整地解码所述图片。响应于确定所述语法元素不具有所述特定值,所述视频解码器可在不使用WPP的情况下解码所述图片的每一图像块。图片可具有一或多个图像块。

[0032] 附图说明实例。在附图中由参考数字指示的元件对应于以下描述中通过相似参考数字指示的元件。在本发明中,具有以序数词(例如,“第一”、“第二”、“第三”等)开始的名称的元件未必暗示元件具有特定次序。相反地,这些序数词仅用以指相同或类似类型的不同元件。

[0033] 图1为说明可利用本发明的技术的实例视频译码系统10的框图。如本文中所描述地使用,术语“视频译码器”一般地指视频编码器与视频解码器两者。在本发明中,术语“视频译码”或“译码”可一般地指视频编码或视频解码。

[0034] 如图1中所展示,视频译码系统10包含源装置12和目的地装置14。源装置12产生经编码视频数据。因此,源装置12可被称作视频编码装置或视频编码设备。目的地装置14可解码由源装置12产生的经编码视频数据。因此,目的地装置14可被称作视频解码装置或视频解码设备。源装置12和目的地装置14可为视频译码装置或视频译码设备的实例。源装置12和目的地装置14可包括广泛范围的装置,包含桌上型计算机、移动计算装置、笔记型(例如,膝上型)计算机、平板计算机、机顶盒、例如所谓的“智能型”电话的电话手持机、电视、相机、显示装置、数字媒体播放器、视频游戏控制台、车载计算机或类似者。

[0035] 目的地装置14可经由信道16自源装置12接收经编码视频数据。信道16可包括能够将经编码视频数据自源装置12移动到目的地装置14的一或多个媒体和/或装置。在一个实例中,信道16可包括使源装置12能够实时将经编码视频数据直接传输到目的地装置14的一或多个通信媒体。在此实例中,源装置12可根据例如无线通信协议的通信标准调制经编码视频数据,且可将经调制视频数据传输到目的地装置14。一或多个通信媒体可包含无线和/或有线通信媒体,例如射频(RF)频谱或一或多个物理传输线。一或多个通信媒体可形成基于包的网路(例如,局域网、广域网或全球网路(例如,因特网))的部分。一或多个通信媒体可包含路由器、交换器、基站或促进自源装置12到目的地装置14的通信的其它装备。

[0036] 在另一实例中,信道16可包含存储由源装置12产生的经编码视频数据的存储媒体。在此实例中,目的地装置14可经由磁盘存取或卡存取来存取存储媒体。存储媒体可包含多种本地存取的数据存储媒体,例如,蓝光光盘、DVD、CD-ROM、快闪存储器或用于存储经编码视频数据的其它合适数字存储媒体。

[0037] 在另一实例中,信道16可包含存储由源装置12产生的经编码视频的文件服务器或另一中间存储装置。在此实例中,目的地装置14可经由流式传输或下载来存取存储于文件服务器或其它中间存储装置处的经编码视频数据。文件服务器可为能够存储经编码视频数据且将经编码视频数据传输到目的地装置14的类型的服务器。实例文件服务器包含页面服务器(例如,用于网站)、文件传送协议(FTP)服务器、网络附接存储(NAS)装置和本地磁盘机。

[0038] 目的地装置14可经由例如因特网连接的标准数据连接存取经编码视频数据。数据连接的实例类型包含适合于存取存储于文件服务器上的经编码视频数据的无线信道(例如,Wi-Fi连接)、有线连接(例如,DSL、缆线调制解调器等)或无线信道与有线连接两者的组合。经编码视频数据自文件服务器的传输可为流式传输、下载传输或流式传输与下载传输两者的组合。

[0039] 本发明的技术不限于无线应用或设定。所述技术可应用于视频译码以支持多种多媒体应用,例如,空中电视广播、有线电视传输、卫星电视传输、流式传输视频传输(例如,经由因特网)、编码视频数据以用于存储于数据存储媒体上、解码存储于数据存储媒体上的视频数据或其它应用。在一些实例中,视频译码系统10可经配置以支持单向或双向视频传输以支持例如视频流式传输、视频播放、视频广播和/或视频电话的应用。

[0040] 在图1的实例中,源装置12包含视频源18、视频编码器20和输出接口22。在一些实例中,输出接口22可包含调制器/解调制器(调制解调器)和/或发射器。视频源18可包含视频俘获装置(例如,摄像机)、含有先前俘获的视频数据的视频存档、自视频内容提供者接收视频数据的视频馈入接口和/或用于产生视频数据的计算机图形系统,或这些视频数据源的组合。

[0041] 视频编码器20可编码来自视频源18的视频数据。在一些实例中,源装置12经由输出接口22将经编码视频数据直接传输到目的地装置14。经编码视频数据也可存储到存储媒体或文件服务器上以供稍后由目的地装置14存取以用于解码和/或播放。

[0042] 在图1的实例中,目的地装置14包含输入接口28、视频解码器30和显示装置32。在一些实例中,输入接口28包含接收器和/或调制解调器。输入接口28可经由信道16接收经编码视频数据。显示装置32可与目的地装置14集成或可在目的地装置14外部。大体来说,显示装置32显示经解码视频数据。显示装置32可包括多种显示装置,例如,液晶显示器(LCD)、等离子显示器、有机发光二极管(OLED)显示器或另一类型的显示装置。

[0043] 视频编码器20和视频解码器30可根据例如目前在开发中的高效率视频译码(HEVC)标准的视频压缩标准操作,且可遵照HEVC测试模型(HM)。被称作“HEVC工作草案5”或“WD5”的即将到来的HEVC标准的草案描述于布罗斯(Bross)等人的“WD5:高效率视频译码的工作草案5(WD5:Working Draft5 of High-Efficiency Video Coding)”(ITU-T SG16WP3和ISO/IEC JTC1/SC29/WG11视频译码联合协作小组(JCT-VC)于2011年11月在瑞士日内瓦的第7次会议)中,所述草案可自2012年10月10日起自[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/7\\_Geneva/wg11/JCTVC-G1103-V3.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1103-V3.zip)下载,所述草案的全部内容以引用的方式并入本文中。被称作“HEVC工作草案9”的即将到来的HEVC标准的另一草案描述于布罗斯等人的“高效率视频译码(HEVC)文本规范草案9(High Efficiency Video Coding(HEVC) text specification draft9)”(ITU-T SG16WP3和ISO/IEC JTC1/SC29/WG11视频译码联合协作小组(JCT-VC)于2012年10月在中国上海的第11次会议)中,所述草案可自2012年11月7日起自[http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/11\\_Shanghai/wg11/JCTVC-K1003-v8.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip)下载,所述草案的全部内容被以引用的方式并入本文中。

[0044] 替代地,视频编码器20和视频解码器30可根据其它专有或工业标准来操作,其它专有或工业标准包含ITU-T H.261、ISO/IEC MPEG-1 Visual、ITU-T H.262或ISO/IEC MPEG-

2Visual、ITU-T H.263、ISO/IEC MPEG-4Visual和ITU-T H.264(也被称为ISO/IEC MPEG-4AVC),包含其可调式视频译码(SVC)和多视图视频译码(MVC)扩展。然而,本发明的技术不限于任何特定译码标准或技术。

[0045] 此外,图1仅为实例且本发明的技术可应用于未必包含编码装置与解码装置之间的任何数据通信的视频译码设定(例如,视频编码或视频解码)。在其它实例中,数据是自本地存储器检索,经由网络而流式传输,或类似者。编码装置可编码数据且将数据存储到存储器,和/或解码装置可自存储器检索数据且解码数据。在许多实例中,由并不彼此通信而是简单地将数据编码到存储器和/或自存储器检索数据且解码数据的装置执行编码和解码。

[0046] 视频编码器20和视频解码器30各自可实施为例如以下各者的多种合适电路中的任一者:一或多个微处理器、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、硬件或其任何组合。如果部分地以软件实施技术,那么装置可将软件的指令存储于合适的非暂时性计算机可读存储媒体中且可使用一或多个处理器以硬件执行指令,以执行本发明的技术。前文中的任一者(包含硬件、软件、硬件与软件的组合等)可被视为一或多个处理器。视频编码器20和视频解码器30中的每一者可包含于一或多个编码器或解码器中,编码器或解码器中的任一者可作为组合式编码器/解码器(CODEC)的部分而集成于相应装置中。

[0047] 本发明可大体上提到视频编码器20将某些信息“用信号发出”到例如视频解码器30的另一装置。术语“用信号发出”大体上可指语法元素和/或表示经编码视频数据的其它数据的传达。此传达可实时地或近实时地发生。交替地,此传达可在时间跨度内发生,例如,可在编码时在将语法元素存储到计算机可读存储媒体时在经编码位流中发生,接着可在将语法元素存储到此媒体之后的任何时间由解码装置检索语法元素。

[0048] 如上文简要地提及,视频编码器20编码视频数据。视频数据可包括一或多个图片。图片中的每一者可为静态图像。在一些例子中,图片可被称作视频“帧”。视频编码器20可产生包含形成视频数据的经译码表示的位序列的位流。位流可包含经译码图片和相关联的数据。经译码图片为图片的经译码表示。相关联的数据可包含序列参数集(SPS)、图片参数集(PPS)和其它语法结构。SPS可含有适用于图片的零个或零个以上序列的参数。PPS可含有适用于零个或零个以上图片的参数。

[0049] 为了产生图片的经编码表示,视频编码器20可将图片分割成译码树块(CTB)的栅格。在一些例子中,CTB可被称作“树块”、“最大译码单元”(LCU)或“译码树单元”。HEVC的CTB可广泛地类似于例如H.264/AVC的先前标准的宏块。然而,CTB未必限于特定大小且可包含一或多个译码单元(CU)。

[0050] CTB中的每一者可与图片内的大小相等的不同像素块相关联。每一像素可包括一个明度(luminance或luma)样本和两个色度(chrominance或chroma)样本。因此,每一CTB可与明度样本的一个块和色度样本的两个块相关联。为了便于解释,本发明可将像素的二维阵列称作像素块且可将样本的二维阵列称作样本块。视频编码器20可使用四分树分割来将与CTB相关联的像素块分割成与CU相关联的像素块(因此名称为“译码树块”)。

[0051] 可将图片的CTB分群成一或多个切片。在一些实例中,切片中的每一者包含整数数目个CTB。作为编码图片的部分,视频编码器20可产生图片的每一切片的经编码表示(即,经译码切片)。为了产生经译码切片,视频编码器20可编码切片的每一CTB以产生切片的CTB中

的每一者的经编码表示(即,经译码CTB)。

[0052] 为了产生经译码CTB,视频编码器20可对与CTB相关联的像素块递归地执行四分树分割以将像素块划分成逐步减小的像素块。较小像素块中的每一者可与CU相关联。经分割CU可为像素块经分割成与其它CU相关联的像素块的CU。未经分割CU可为像素块未经分割成与其它CU相关联的像素块的CU。

[0053] 视频编码器20可针对每一未经分割CU产生一或多个预测单元(PU)。CU的PU中的每一者可与CU的像素块内的一不同像素块相关联。视频编码器20可针对CU的每一PU产生预测性像素块。PU的预测性像素块可为像素的块。

[0054] 视频编码器20可使用帧内预测或帧间预测来产生PU的预测性像素块。如果视频编码器20使用帧内预测来产生PU的预测性像素块,那么视频编码器20可基于与PU相关联的图片的经解码像素而产生PU的预测性像素块。如果视频编码器20使用帧间预测来产生PU的预测性像素块,那么视频编码器20可基于不同于与PU相关联的图片的一或多个图片的经解码像素而产生PU的预测性像素块。

[0055] 视频编码器20可基于CU的PU的预测性像素块而产生CU的残余像素块。CU的残余像素块可指示CU的PU的预测性像素块中的样本与CU的原始像素块中的对应样本之间的差。

[0056] 此外,作为编码未经分割CU的部分,视频编码器20可对CU的残余像素块执行递归四分树分割以将CU的残余像素块分割成与CU的变换单元(TU)相关联的一或多个较小残余像素块。因为与TU相关联的像素块中的像素各自包含一个明度样本和两个色度样本,所以TU中的每一者可与明度样本的一个残余样本块和色度样本的两个残余样本块相关联。

[0057] 视频译码器20可对与TU相关联的残余样本块应用一或多个变换以产生系数块(即,系数的块)。视频编码器20可对系数块中的每一者执行量化过程。量化大体上指对系数进行量化以可能减少用以表示系数的数据的量从而提供进一步压缩的过程。

[0058] 视频编码器20可产生表示经量化系数块中的系数的语法元素的集合。视频编码器20可将例如上下文自适应性二进制算术译码(CABAC)操作的熵编码操作应用于这些语法元素中的至少一些语法元素。作为执行熵编码操作的部分,视频编码器20可选择译码上下文。在CABAC的状况下,译码上下文可指示值0和值1分格的概率。

[0059] 由视频编码器20产生的位流可包含一系列网络抽象层(NAL)单元。NAL单元中的每一者可为含有NAL单元中的数据的类型的指示和含有数据的字节的语法结构。例如,NAL单元可含有表示SPS、PPS、经译码切片、补充增强信息(SEI)、存取单元定界符、填充数据或另一类型的数据的数据。经译码切片NAL单元为包含经译码切片的NAL单元。

[0060] 视频解码器30可接收位流。位流可包含由视频编码器20编码的视频数据的经译码表示。视频解码器30可剖析位流以自位流提取语法元素。作为自位流提取一些语法元素的部分,视频解码器30可熵解码(例如,CABAC解码、指数哥伦布解码等)位流中的数据。视频解码器30可基于自位流提取的语法元素而重建视频数据的图片。

[0061] 基于语法元素而重建视频数据的过程可与由视频编码器20执行以产生语法元素的过程大体上互逆。例如,视频解码器30可基于与CU相关联的语法元素而产生CU的PU的预测性像素块。另外,视频解码器30可对与CU的TU相关联的系数块进行反量化。视频解码器30可对系数块执行反变换以重建与CU的TU相关联的残余像素块。视频解码器30可基于预测性像素块和残余像素块而重建CU的像素块。

[0062] 在一些实例中,视频编码器20可将图片划分成多个熵切片。本发明可使用术语“规则切片”来区分切片与熵切片。熵切片可包含规则切片的CU的子集。在一些实例中,视频编码器20可在熵切片当中分割CU,以使得熵切片中无一者包含比上限多的分格(例如,经熵译码位)。在单独NAL单元中可包含每一熵切片。

[0063] 在本发明中,图片中预测可指使用与图片的第一单元(例如,CTB、CU、PU等)相关联的信息用于译码同一图片的第二单元。允许跨越熵切片边界的图片中预测,但用于达成熵译码的目的除外。例如,如果视频译码器(例如,视频编码器20或视频解码器30)正对特定PU执行帧内预测,那么视频译码器可使用来自相邻PU的样本,即使相邻PU在不同于特定PU的熵切片中也如此。在此实例中,如果相邻PU在不同于特定PU的切片中,那么视频译码器可能不能够使用来自相邻PU的样本。

[0064] 然而,在视频译码器正对与特定PU相关联的数据执行熵译码时,如果特定PU与相邻PU在相同熵切片中,那么仅允许视频译码器基于与相邻PU相关联的信息而选择译码上下文。因为此限制,所以视频译码器可能能够对切片的多个熵切片平行地执行熵译码(即,熵编码或解码)操作。因此,视频解码器30可能能够平行地剖析多个熵切片的语法元素。然而,视频解码器30不能够平行地重建切片的多个熵切片的像素块。

[0065] 如上文所指示,经译码切片NAL单元可含有经译码切片。此切片可为熵切片或规则切片。经译码切片NAL单元中的切片标头可包含指示切片是熵切片还是规则切片的语法元素(例如,entropy\_slice\_flag)。例如,如果语法元素等于1,那么经译码切片NAL单元中的切片可为熵切片。

[0066] 每一经译码切片可包含切片标头和切片数据。熵切片的切片标头可不同于规则切片的切片标头。例如,熵切片的切片标头中的语法元素可包含规则切片的切片标头中的语法元素的子集。因为熵切片的切片标头包含比规则切片的切片标头少的语法元素,所以熵切片也可被称作轻型切片、具有短切片标头的切片或短切片。熵切片可从按解码次序在熵切片之前的规则切片的切片标头继承从熵切片的切片标头中省略的语法元素。

[0067] 按照惯例,视频编码器针对每一熵切片产生单独NAL单元。常常以单独包在网络上输送个别NAL单元。换句话说,在经由网络传输NAL单元期间,每一包可存在一个NAL单元。此情形对于含有熵切片的NAL单元来说可为有问题的。如果含有包含规则切片的NAL单元的包在传输期间丢失,那么视频解码器30可能不能够使用自规则切片的切片标头继承语法元素的熵切片。此外,如果对于图片中预测,第一熵切片的一或多个CTB依赖于第二熵切片的一或多个CTB,且含有包含第二熵切片的NAL单元的包在传输期间丢失,那么视频解码器30可能不能够解码第一熵切片的CTB。

[0068] 在一些实例中,视频译码器可使用波前平行处理(WPP)来译码图片的至少部分。下文中详细描述图9为说明WPP的实例的概念图。如果视频译码器使用WPP来译码图片,那么视频译码器可将图片的CTB划分成多个“WPP波”。WPP波中的每一者可对应于图片中的CTB的不同行。如果视频译码器使用WPP来译码图片,那么视频译码器可开始译码CTB的顶行。在视频译码器已译码顶行的两个或两个以上CTB之后,视频译码器可开始与译码CTB的顶行平行地译码CTB的次顶行。在视频译码器已译码次顶行的两个或两个以上CTB之后,视频译码器可开始与译码CTB的较高行平行地译码CTB的顶部起第三行。此型样可沿图片中的CTB的行往下继续。

[0069] 如果视频译码器正使用WPP,那么只要空间上相邻的CU在当前CTB的左方、左上方、上方或右上方,视频译码器便可使用与在当前CTB之外的空间上相邻的CU相关联的信息来对当前CTB中的特定CU执行图片中预测。如果当前CTB为不同于最顶行的行中的最左CTB,那么视频译码器可使用与紧接的较高行的第二CTB相关联的信息来选择用于CABAC译码当前CTB的一或多个语法元素的上下文。否则,如果当前CTB并非行中的最左CTB,那么视频译码器可使用与当前CTB左方的CTB相关联的信息来选择用于CABAC译码当前CTB的一或多个语法元素的上下文。以此方式,视频译码器可在编码紧接的较高行的两个或两个以上CTB之后,基于紧接的较高行的CABAC状态而初始化行的CABAC状态。

[0070] 因此,响应于确定第一CTB通过单一CTB而与图片的左边界分离,视频译码器可存储与第一CTB相关联的上下文变量。视频译码器可至少部分基于与第一CTB相关联的上下文变量而熵译码(例如,熵编码或熵解码)第二CTB的一或多个语法元素,第二CTB邻近于图片的左边界且比第一CTB低一行CTB。

[0071] 甚至在使用WPP时,仍通常根据光栅扫描次序将切片的经译码CTB布置于经译码切片NAL单元中。此情形可使实施WPP的视频译码器的设计变得复杂。在WPP波的数目大于1且小于图片的CTB行的数目时,CTB的经译码位的位流次序(即,在通过一个解码器核心处理经译码图片而非平行地解码经译码图片的情况下的解码次序)与如下未应用WPP时的情形相比较来说改变了。按位流/解码次序较迟的经译码CTB可能需要由按解码次序较早的另一经译码CTB进行图片中预测。此情形可打破无较早数据取决于按位流/解码次序较迟到来的数据的位流因果关系。位流因果关系已为包含视频译码标准的视频译码设计中的大体遵循的原理。虽然解码过程起作用,但在指示在位流中的当前位置的位流指标可在与经译码切片NAL单元相关联的位流的部分内前后移动时,解码过程可更复杂。

[0072] 在一些实例中,视频编码器20可将图片划分成一或多个图像块。图像块可包括图片的CTB的非重叠集合。视频编码器20可通过界定两个或两个以上垂直图像块边界和两个或两个以上水平图像块边界而将图片划分成图像块。图片的每一垂直侧可为垂直图像块边界。当前图片的每一水平侧可为水平图像块边界。例如,如果视频编码器20界定图片的四个垂直图像块边界和三个水平图像块边界,那么将当前图片划分成六个图像块。

[0073] 例如视频编码器20或视频解码器30的视频译码器可根据图像块扫描次序译码图片的图像块的CTB。为了根据图像块扫描次序译码CTB,视频译码器可根据光栅扫描次序译码图片的图像块。即,视频译码器可按由左到右次序译码一行图像块中的每一图像块,自图像块的顶行开始且接着沿图片向下进行。此外,视频译码器可根据光栅扫描次序译码图像块内的每一CTB。以此方式,视频译码器可在译码图片的另一图像块的任何CTB之前,译码图片的给定图像块的每一CTB。换句话说,图像块扫描次序按图像块内的CTB光栅扫描次序遍历CTB,且按图片内的图像块光栅扫描次序遍历图像块。因此,视频译码器译码图片的CTB的次序在将图片分割成多个图像块的情况下与在不将图片分割成多个图像块的情况下可不同。下文所描述的图10为说明在将图片分割成多个图像块时的实例图像块扫描次序的概念图。

[0074] 在一些例子中,视频译码器可执行跨越图像块边界但不跨越切片边界的图片中预测。在其它例子中,禁止跨越图像块边界和切片边界的图片中预测。在禁止跨越图像块边界和切片边界的图片中预测的例子中,视频译码器可能能够平行地译码多个图像块。

[0075] 在一些实例中,由旗标(例如,“tile\_boundary\_independence\_idc”)控制跨越图像块边界的图片中预测。如果旗标等于1,那么在图片内不允许跨越图像块边界的图片中预测。否则,允许跨越图像块边界的图片中预测,也为图片边界或切片边界的图像块边界除外。如果允许跨越图像块边界的图片中预测,那么图像块的功能性可为与图片不具有图像块或等效地具有仅一个图像块的状况相比较来说改变CTB的扫描次序。如果不允许跨越图像块边界的图片中预测,那么除改变CTB的扫描次序之外,图像块也可提供可用于图像块的平行译码(编码和/或解码)的独立分割。因此,如果将图片分割成至少第一图像块和第二图像块,那么在视频解码器30在不使用WPP的情况下解码图像块时,视频解码器30可平行地解码第一图像块的CTB和第二图像块的CTB。

[0076] 在一些例子中,可将图片分割成图像块、WPP波和熵切片的组合。例如,可将图片分割成图像块和WPP波的集合。在另一实例中,可将图片分割成两个图像块和一个熵切片。允许图片内的图像块、WPP波和熵切片的组合可为有问题的,这是因为允许这些组合可增加视频编码器和/或视频解码器的复杂度和成本。

[0077] 本发明的技术可解决上文所描述的问题。根据本发明的技术,可不将图片分割成图像块、WPP波和熵切片的任何组合。换句话说,可将图片分割成一或多个图像块,可将图片分割成WPP波,或可将图片分割成一或多个熵切片。然而,可不将图片分割成以下组合中的任一者:(a) 图像块、WPP波和熵切片、(b) 图像块和WPP波、(c) 图像块和熵切片或(d) WPP波和熵切片。

[0078] 为了实现此情形,视频编码器20可在位流中包含语法元素,所述语法元素指示根据第一译码模式或是第二译码模式来编码图片。在所述第一译码模式中,使用WPP来完整地编码所述图片。即,可将图片中的每一行CTB编码为WPP波。在第二译码模式中,图片可具有一或多个图像块。此外,在所述第二译码模式中,可在不使用WPP的情况下编码所述图片的每一图像块。例如,在第二译码模式中,视频编码器20可针对图片的每一图像块,按跨越CTB的行自左到右且沿图像块的CTB的行向下的次序顺序地编码图像块内的CTB。为了便于解释,此语法元素可在本文中被称作译码模式语法元素。

[0079] 视频解码器30可自包含视频数据中的图片的经译码表示的位流剖析语法元素。响应于确定所述语法元素具有特定值,视频解码器30可使用WPP来完整地解码所述图片。响应于确定所述语法元素不具有所述特定值,视频解码器30可在不使用WPP的情况下解码所述图片的每一图像块,其中图片具有一或多个图像块。

[0080] 位流的各个部分可包含译码模式语法元素。例如,视频编码器20可产生包含译码模式语法元素的SPS。在此实例中,视频解码器30可自位流剖析包含译码模式语法元素的SPS。在另一实例中,视频编码器20可产生包含译码模式语法元素的PPS。在此实例中,视频解码器30可自位流剖析包含译码模式语法元素的PPS。此外,如果根据第二译码模式编码图片,那么位流可包含指示针对图片是否启用熵切片的一或多个语法元素。位流的各个部分可包含指示针对图片是否启用熵切片的一或多个语法元素。例如,SPS可包含指示针对与SPS相关联的图片启用熵切片的一或多个语法元素。在另一实例中,PPS可包含指示针对与PPS相关联的图片启用熵切片的一或多个语法元素。例如,在此实例中,PPS可包含指示参考PPS的经译码切片是否可由熵切片组成的entropy\_slice\_enabled\_flag语法元素。

[0081] 如果图片包含一或多个熵切片,那么与图片的切片相关联的每一熵切片可包含于

单一经译码切片NAL单元中,而非包含于单独NAL单元中。因此,可将熵切片定义为切片的子集,其中熵切片的熵解码过程独立于相同切片中的其它熵切片。

[0082] 如上文简要地提及,位流可包含包含经译码切片的经译码切片NAL单元。经译码切片可包括切片标头和切片数据。切片数据可包含一或多个子流。根据本发明的技术,如果在第一译码模式中编码图片(即,使用WPP来完整地编码图片),那么由子流中的单一者来表示切片的CTB的每一行。如果在第二译码模式中编码图片(即,在不使用WPP的情况下编码图片的每一图像块),那么由子流中的单一子流表示具有切片中的一或多个CTB的图片的每一图像块。

[0083] 此外,根据本发明的技术,经译码切片的切片标头可包含指示经译码切片NAL单元的切片数据内的图像块、WPP波或熵切片的入口点的语法元素的集合。子流的入口点可为子流的第一位。此外,经译码切片NAL单元的切片数据内的图像块、WPP波或熵切片可包含确保图像块、WPP波或熵切片字节对准的填补位。

[0084] 图2为说明经配置以实施本发明的技术的实例视频编码器20的框图。出于解释的目的而提供图2且不应将图2视为限制如本发明中广泛地例示和描述的技术。出于解释的目的,本发明在HEVC译码的情况下描述视频编码器20。然而,本发明的技术可适用于其它译码标准或方法。

[0085] 在图2的实例中,视频编码器20包含预测处理单元100、残余产生单元102、变换处理单元104、量化单元106、反量化单元108、反变换处理单元110、重构建单元112、滤波器单元113、经解码图片缓冲器114和熵编码单元116。预测处理单元100包含帧间预测处理单元121和帧内预测处理单元126。帧间预测处理单元121包含运动估计单元122和运动补偿单元124。在其它实例中,视频编码器20可包含更多、更少或不同功能组件。

[0086] 视频编码器20可接收视频数据。为了编码视频数据,视频编码器20可编码视频数据的每一图片的每一切片。作为编码切片的部分,视频编码器20可编码切片中的每一CTB。作为编码CTB的部分,预测处理单元100可对与CTB相关联的像素块执行四分树分割,以将像素块划分成逐步减小的像素块。较小像素块可与CU相关联。例如,预测处理单元100可将CTB的像素块分割成四个大小相等的子块,将子块中的一或多者分割成四个大小相等的子子块,等等。

[0087] 视频编码器20可编码CTB的CU以产生CU的经编码表示(即,经译码CU)。视频编码器20可根据z扫描次序编码CTB的CU。换句话说,视频编码器20可按左上方CU、右上方CU、左下方CU和接着右下方CU的彼次序编码所述CU。在视频编码器20编码经分割CU时,视频编码器20可根据z扫描次序编码与经分割CU的像素块的子块相关联的CU。

[0088] 作为编码CU的部分,预测处理单元100可在CU的一或多个PU当中分割CU的像素块。视频编码器20和视频解码器30可支持各种PU大小。假定特定CU的大小为 $2N \times 2N$ ,视频编码器20和视频解码器30可支持 $2N \times 2N$ 或 $N \times N$ 的PU大小以供帧内预测,且支持 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $N \times N$ 或类似大小的对称PU大小以供帧间预测。视频编码器20和视频解码器30也可支持对 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 和 $nR \times 2N$ 的PU大小的非对称分割以供帧间预测。

[0089] 帧间预测处理单元121可通过对CU的每一PU执行帧间预测来产生PU的预测性数据。PU的预测性数据可包含对应于PU的预测性像素块和PU的运动信息。切片可为I切片、P切片或B切片。帧间预测单元121可取决于PU在I切片中、在P切片中或是在B切片中而对CU的PU

执行不同操作。在I切片中,对所有PU进行帧内预测。因此,如果PU在I切片中,那么帧间预测单元121不对PU执行帧间预测。

[0090] 如果PU在P切片中,那么运动估计单元122可搜索参考图片列表(例如,“列表0”)中的参考图片以查找PU的参考块。PU的参考块可为最紧密地对应于PU的像素块的像素块。运动估计单元122可产生指示含有PU的参考块的列表0中的参考图片的参考图片索引,和指示PU的像素块与参考块之间的空间位移的运动向量。运动估计单元122可输出参考图片索引和运动向量作为PU的运动信息。运动补偿单元124可基于通过PU的运动信息指示的参考块而产生PU的预测性像素块。

[0091] 如果PU在B切片中,那么运动估计单元122可对PU执行单向帧间预测或双向帧间预测。为了对PU执行单向帧间预测,运动估计单元122可搜索第一参考图片列表(“列表0”)或第二参考图片列表(“列表1”)的参考图片以查找PU的参考块。运动估计单元122可输出以下各者作为PU的运动信息:指示参考图片的列表0或列表1中含有参考块的位置的参考图片索引、指示PU的像素块与参考块之间的空间位移的运动向量,和指示参考图片在列表0或是列表1中的预测方向指示符。

[0092] 为了对PU执行双向帧间预测,运动估计单元122可搜索列表0中的参考图片以查找PU的参考块,且也可搜索列表1中的参考图片以查找PU的另一参考块。运动估计单元122可产生指示参考图片的列表0和列表1中含有参考块的位置的参考图片索引。另外,运动估计单元122可产生指示参考块与PU的像素块之间的空间位移的运动向量。PU的运动信息可包含PU的参考图片索引和运动向量。运动补偿单元124可基于通过PU的运动信息指示的参考块而产生PU的预测性像素块。

[0093] 帧内预测处理单元126可通过对PU执行帧内预测来产生PU的预测性数据。PU的预测性数据可包含PU的预测性像素块和各种语法元素。帧内预测处理单元126可对I切片、P切片和B切片中的PU执行帧内预测。

[0094] 为了对PU执行帧内预测,帧内预测处理单元126可使用多个帧内预测模式来产生PU的预测性数据的多个集合。为了使用帧内预测模式来产生PU的预测性数据的集合,帧内预测处理单元126可使样本在与帧内预测模式相关联的方向上自相邻PU的样本块跨越PU的样本块延伸。假定PU、CU和CTB的左到右、上到下编码次序,那么相邻PU可在PU的上方、右上方、左上方或左方。帧内预测处理单元126可使用各种数目个帧内预测模式,例如,33个定向帧内预测模式。在一些实例中,帧内预测模式的数目可取决于PU的像素块的大小。

[0095] 预测处理单元100可自CU的PU的由帧间预测处理单元121产生的预测性数据或PU的由帧内预测处理单元126产生的预测性数据当中选择PU的预测性数据。在一些实例中,预测处理单元100基于预测性数据的集合的位率/失真度量而选择CU的PU的预测性数据。所选定预测性数据的预测性像素块可在本文中被称作所选定预测性像素块。

[0096] 残余产生单元102可基于CU的像素块和CU的PU的所选定预测性像素块而产生CU的残余像素块。例如,残余产生单元102可产生CU的残余像素块,以使得残余像素块中的每一样本具有等于CU的像素块中的样本与CU的PU的所选定预测性像素块中的对应样本之间的差的值。

[0097] 预测处理单元100可执行四分树分割以将CU的残余像素块分割成子块。每一未经划分的残余像素块可与CU的不同TU相关联。与CU的TU相关联的残余像素块的大小和位置可

或可不基于CU的PU的像素块的大小和位置。

[0098] 因为TU的残余像素块的像素可包括一个明度样本和两个色度样本,所以TU中的每一者可与明度样本的一个块和色度样本的两个块相关联。变换处理单元104可通过将一或多个变换应用于与TU相关联的残余样本块而产生CU的每一TU的系数块。变换处理单元104可将各种变换应用于与TU相关联的残余样本块。例如,变换处理单元104可将离散余弦变换(DCT)、方向变换或概念上类似的变换应用于残余样本块。

[0099] 量化单元106可对与TU相关联的系数块中的系数进行量化。量化过程可减小与系数中的一些或全部相关联的位深度。例如,可在量化期间将n位系数降值舍位到m位系数,其中n大于m。量化单元106可基于与CU相关联的量化参数(QP)值而对与CU的TU相关联的系数块进行量化。视频编码器20可通过调整与CU相关联的QP值来调整应用于与CU相关联的系数块的量化的程度。

[0100] 反量化单元108和反变换处理单元110可分别将反量化和反变换应用于系数块以自系数块重建残余样本块。重建单元112可将经重建残余样本块与来自预测处理单元100产生的一或多个预测性样本块的对应样本相加以产生与TU相关联的经重建样本块。通过以此方式重建CU的每一TU的样本块,视频编码器20可重建CU的像素块。

[0101] 滤波器单元113可执行解块操作以减少与CU相关联的像素块中的块伪影。经解码图片缓冲器114可在滤波器单元113对经重建像素块执行一或多个解块操作之后,存储经重建像素块。帧间预测单元121可使用含有经重建像素块的参考图片来对其它图片的PU执行帧间预测。另外,帧内预测处理单元126可使用经解码图片缓冲器114中的经重建像素块来对与CU相同的图片中的其它PU执行帧内预测。

[0102] 熵编码单元116可自视频编码器20的其它功能组件接收数据。例如,熵编码单元116可自量化单元106接收系数块且可自预测处理单元100接收语法元素。熵编码单元116可对数据执行一或多个熵编码操作以产生经熵编码数据。例如,熵编码单元116可对数据执行以下操作:上下文自适应性可变长度译码(CAVLC)操作、CABAC操作、变量到变量(V2V)长度译码操作、基于语法的上下文自适应性二进制算术译码(SBAC)操作、概率区间分割熵(PIPE)译码操作、指数哥伦布编码操作或另一类型的熵编码操作。

[0103] 视频编码器20可输出包含由熵编码单元116产生的经熵编码数据的位流。位流可包含一系列NAL单元。NAL单元可包含经译码切片NAL单元、SPS NAL单元、PPS NAL单元等等。为了确保图片不包含图像块、WPP波和熵切片的组合,位流可包含指示是否使用WPP来完整地编码图片或是否在不使用WPP的情况下编码图片的每一图像块的语法元素。

[0104] 图3为说明经配置以实施本发明的技术的实例视频解码器30的框图。出于解释的目的而提供图3且图3并不限制如本发明中广泛地例示和描述的技术。出于解释的目的,本发明在HEVC译码的情况下描述视频解码器30。然而,本发明的技术可适用于其它译码标准或方法。

[0105] 在图3的实例中,视频解码器30包含熵解码单元150、预测处理单元152、反量化单元154、反变换处理单元156、重建单元158、滤波器单元159和经解码图片缓冲器160。预测处理单元152包含运动补偿单元162和帧内预测处理单元164。在其它实例中,视频解码器30可包含更多、更少或不同功能组件。

[0106] 视频解码器30可接收位流。熵解码单元150可剖析位流以自位流提取语法元素。作

为剖析位流的部分,熵解码单元150可熵解码位流中的经熵编码语法元素。预测处理单元152、反量化单元154、反变换处理单元156、重建单元158和滤波器单元159可基于自位流提取的语法元素而产生经解码视频数据。

[0107] 位流可包括一系列NAL单元。位流的NAL单元可包含经译码切片NAL单元。作为剖析位流的部分,熵解码单元150可自经译码切片NAL单元提取语法元素且熵解码语法元素。经译码切片中的每一者可包含切片标头和切片数据。切片标头可含有关于切片的语法元素。切片标头中的语法元素可包含识别与含有切片的图片相关联的PPS的语法元素。

[0108] 另外,视频解码器30可对未经分割CU执行重建操作。为了对未经分割CU执行重建操作,视频解码器30可对CU的每一TU执行重建操作。通过对CU的每一TU执行重建操作,视频解码器30可重建与CU相关联的残余像素块。

[0109] 作为对CU的TU执行重建操作的部分,反量化单元154可对与TU相关联的系数块进行反量化(即,去量化)。反量化单元154可使用与TU的CU相关联的QP值来确定量化的程度,且同样地,确定反量化单元154将应用的反量化的程度。

[0110] 在反量化单元154对系数块进行反量化之后,反变换处理单元156可将一或多个反变换应用于系数块以便产生与TU相关联的残余样本块。例如,反变换处理单元156可将反DCT、反整数变换、反卡忽南-拉维变换(KLT)、反旋转变换、反方向变换或另一反变换应用于系数块。

[0111] 如果使用帧内预测来编码PU,那么帧内预测处理单元164可执行帧内预测以产生PU的预测性样本块。帧内预测处理单元164可使用帧内预测模式来基于空间上相邻的PU的像素块而产生PU的预测性像素块。帧内预测处理单元164可基于自位流剖析的一或多个语法元素而确定PU的帧内预测模式。

[0112] 运动补偿单元162可基于自位流提取的语法元素而构建第一参考图片列表(列表0)和第二参考图片列表(列表1)。此外,如果使用帧间预测来编码PU,那么熵解码单元150可提取PU的运动信息。运动补偿单元162可基于PU的运动信息而确定PU的一或多个参考块。运动补偿单元162可基于PU的一或多个参考块而产生PU的预测性像素块。

[0113] 重建单元158可在可适用时,使用与CU的TU相关联的残余像素块和CU的PU的预测性像素块(即,帧内预测数据或帧间预测数据)来重建CU的像素块。明确地说,重建单元158可将残余像素块的样本与预测性像素块的对应样本相加以重建CU的像素块。

[0114] 滤波器单元159可执行解块操作以减少与CU的像素块相关联的块伪影。视频解码器30可将CU的像素块存储于经解码图片缓冲器160中。经解码图片缓冲器160可提供用于后续运动补偿、帧内预测和呈现于显示装置(例如,图1的显示装置32)上的参考图片。例如,视频解码器30可基于经解码图片缓冲器160中的像素块而对其它CU的PU执行帧内预测或帧间预测操作。

[0115] 如上文所提及,视频解码器30可接收包含译码模式语法元素的位流。如果译码模式语法元素具有特定值,那么译码模式语法元素指示使用WPP来完整地编码图片。在各种实例中,译码模式语法元素可在位流的各个部分中。例如,SPS可包含译码模式语法元素。以下表1提供包含译码模式语法元素(“tile\_mode”)的SPS的实例语法。

[0116] 表1——序列参数集RBSP语法

|        |                                                                |       |
|--------|----------------------------------------------------------------|-------|
|        | seq_parameter_set_rbsp() {                                     | 描述符   |
|        | ...                                                            |       |
|        | <b>inter_4x4_enabled_flag</b>                                  | u(1)  |
|        | <b>tile_mode</b> // 0: 一个图片中的仅一个图像块; 1: 均匀间隔; 2: 不均匀间隔; 3: WPP | u(2)  |
|        | if( tile_mode == 1    tile_mode == 2 ) {                       |       |
|        | <b>num_tile_columns_minus1</b>                                 | ue(v) |
|        | <b>num_tile_rows_minus1</b>                                    | ue(v) |
|        | }                                                              |       |
|        | if( tile_mode == 2 ) {                                         |       |
| [0117] | for ( i = 0; i < num_tile_columns_minus1; i++ )                |       |
|        | <b>column_width[i]</b>                                         | ue(v) |
|        | for ( i = 0; i < num_tile_rows_minus1; i++ )                   |       |
|        | <b>row_height[i]</b>                                           | ue(v) |
|        | }                                                              |       |
|        | if( tile_mode == 1    tile_mode == 2 ) {                       |       |
|        | <b>tile_boundary_independence_flag</b>                         | u(1)  |
|        | if( tile_boundary_independence_flag )                          |       |
|        | <b>loop_filter_across_tile_flag</b>                            | u(1)  |
|        | }                                                              |       |
|        | rbsp_trailing_bits()                                           |       |
|        | }                                                              |       |

[0118] 具有类型描述符ue(v)的语法元素为使用指数哥伦布译码编码的无正负号可变长度值,其中左位居首。具有类型描述符u(1)和u(2)的语法元素为长度分别为1个位或2个位的无正负号值。在表1的实例语法中,inter\_4×4\_enabled\_flag语法元素指定是否可将帧间预测应用于具有4×4明度样本的大小的块。

[0119] 此外,在表1的实例语法中,tile\_mode语法元素指定与SPS相关联的图片的图像块模式。如果tile\_mode语法元素等于0,那么在与SPS相关联的图片中的每一者中存在仅一个图像块。在不使用WPP的情况下根据光栅扫描次序译码每一图片的单一图像块中的CTB。如果tile\_mode语法元素等于1,那么与SPS相关联的图片处于均匀隔开的图像块模式中。在图片处于均匀隔开的图像块模式中时,图像块列边界和图像块行边界均匀地分布于与SPS相关联的每一图片中。因此,在图片处于均匀隔开的图像块模式中时,图片的图像块具有相同大小。可在不使用WPP的情况下根据光栅扫描次序编码均匀分布的图像块中的每一者内的CTB。如果tile\_mode语法元素等于2,那么与SPS相关联的图片处于不均匀隔开的图像块模式中。在图片处于不均匀隔开的图像块模式中时,图像块列边界和图像块行边界跨越图片不均匀地分布,但可使用SPS的column\_width[i]和row\_height[i]语法元素显式地用信号发出。可在不使用WPP的情况下根据光栅扫描次序编码不均匀隔开的图像块中的每一者内的CTB。

[0120] 如果tile\_mode语法元素等于3,那么使用WPP模式来译码与SPS相关联的图片。换句话说,如果tile\_mode语法元素具有特定值(例如,3),那么使用WPP来完整地编码与SPS相关联的图片。如果tile\_mode语法元素具有除了3之外的任一值,那么与SPS相关联的任何图片中无一图像块是使用WPP来编码。此外,在使用WPP来译码图片时,在解码图片的一行CTB中的两个CTB之后调用特定记忆过程。另外,在解码图片的一行CTB中的第一CTB之前调用特定同步过程。另外,在已译码一行中的最右CTB时,调用内部变量的特定CABAC状态重新初始

化过程。

[0121] 在上文所提及的特定记忆过程中,视频译码器可响应于确定第一CTB通过单一CTB而与图片的左边界分离,存储与第一CTB相关联的特定上下文变量。在特定同步过程中,视频译码器可至少部分基于与第一CTB相关联的上下文变量而熵译码(即,熵编码或熵解码)第二CTB的一或多个语法元素,第二CTB定位于邻近于图片的左边界处且定位于比第一CTB低的一行CTB处。

[0122] 此外,在表1的实例语法中,num\_tile\_columns\_minus1语法元素指定分割与SPS相关联的图片中的每一者的图像块列的数目。在tile\_mode语法元素等于0或3时,可推断num\_tile\_columns\_minus1语法元素的值等于0。这是因为在tile\_mode语法元素等于0时,在图片中存在仅一单一图像块,且在tile\_mode语法元素等于3时,图片的每一CTB行为单一图像块。num\_tile\_rows\_minus1语法元素指定分割与SPS相关联的图片中的每一者的图像块行的数目。在tile\_mode语法元素等于0时,可推断num\_tile\_rows\_minus1语法元素的值等于0。在tile\_mode语法元素等于3时,视频解码器30可自动地确定(即,推断)num\_tile\_rows\_minus1语法元素的值等于CTB中的图片的高度减去1所得值。此外,在tile\_mode语法元素等于1或2时,num\_tile\_columns\_minus1语法元素和num\_tile\_rows\_minus1语法元素中的至少一者大于0。

[0123] 视频解码器30可基于column\_width[i]语法元素和row\_height[i]语法元素而确定与SPS相关联的图片的图像块的宽度和高度。column\_width[i]语法元素指示与SPS相关联的图片的图像块列的宽度。视频解码器30可至少部分基于column\_width[i]语法元素而产生指示与SPS相关联的图片中的图像块的列的宽度的columnWidth向量。视频解码器30可使用以下伪码自SPS的column\_width[i]语法元素产生columnWidth向量。

```

for ( i = 0; i <= num_tile_columns_minus1; i++ ) {
    if ( tile_mode != 2 )
        columnWidth[i] =
            ( ( i+1 ) * PicWidthInLCUs ) /
[0124]      ( num_tile_columns_minus1+1 ) -
            ( i * PicWidthInLCUs ) / ( num_tile_columns_minus1+1 )
    else
        columnWidth[i] = column_width[i]
}

```

[0125] 视频解码器30可产生指示与SPS相关联的图片中的图像块的高度的rowHeight向量。在一些实例中,视频解码器30可使用以下伪码来产生rowHeight向量。

```

        for ( i = 0; i <= num_tile_rows_minus1; i++ ) {
            if ( tile_mode != 2 )
                rowHeight[i] =
                    (( i+1 ) * PicHeightInLCUs) /
[0126]    ( num_tile_rows_minus1+1 ) -
                    ( i * PicHeightInLCUs ) / ( num_tile_rows_minus1+1 )
            else
                rowHeight[i] = row_width[i]
        }

```

[0127] 此外,视频解码器30可产生指示图像块的每一列的最左列边界的在与SPS相关联的图片内的位置的colBd向量。在一些实例中,视频解码器30可使用以下伪码来确定colBd向量。

```

[0128]    colBd[0] = 0
[0129]    for ( i = 0; i <= num_tile_columns_minus1; i++ )
[0130]    colBd[i+1] = colBd[i] + columnWidth[i]

```

[0131] 视频解码器30可产生指示图像块的每一行的顶行边界的在与SPS相关联的图片内的位置的rowBd向量。在一些实例中,视频解码器30可使用以下伪码来确定rowBd向量。

```

[0132]    rowBd[0] = 0
[0133]    for ( i = 0; i <= num_tile_rows_minus1; i++ )
[0134]    rowBd[i+1] = rowBd[i] + rowHeight[i]

```

[0135] 在表1的实例语法中,tile\_boundary\_independence\_flag语法元素指示是否可独立地解码图像块。例如,如果tile\_boundary\_independence\_flag等于1,那么可独立地解码图像块。例如,如果tile\_boundary\_independence\_flag等于1且视频解码器30正解码特定CTB,那么确定不在与特定CTB相同的图像块内的与特定CTB相邻的所有CTB不可用于图片中预测。此外,如果tile\_boundary\_independence\_flag等于1,那么视频解码器30在熵解码图像块中的第一CTB之前重新初始化熵译码上下文。

[0136] 如果tile\_boundary\_independence\_flag语法元素等于0,那么用于图片中预测的CTB的可用性不受图像块边界影响。换句话说,如果tile\_boundary\_independent\_flag语法元素等于0,那么视频解码器30可执行跨越图像块边界的图片中预测。此外,如果tile\_boundary\_independence\_flag语法元素等于0,那么熵解码单元150可在解码图像块中的第一CTB时调用同步过程,图片中的第一树块除外。在此同步过程中,熵解码单元150可使用与先前图像块的最后CTB相关联的信息来选择用于熵解码图像块中的第一CTB的一或多个语法元素的译码上下文。另外,熵解码单元150可在解码图像块中的第二CTB行中的第一CTB时执行记忆过程。记忆过程可存储上下文变量,以用于在选择用于CABAC译码CTB之下一较低行中的最左CTB的一或多个语法元素的上下文中使用。

[0137] 如果tile\_mode语法元素等于0(即,每一图片存在仅一个图像块),那么在表1的实例语法中,SPS不包含tile\_boundary\_independence\_flag语法元素。然而,如果tile\_mode语法元素等于0,那么视频解码器30可自动地确定tile\_boundary\_independence\_flag语法

元素的值等于1。类似地,如果tile\_mode语法元素等于3(即,使用WPP来完整地编码图片),那么在表1的实例语法中,SPS不包含tile\_boundary\_independence\_flag语法元素。然而,如果tile\_mode语法元素等于3,那么视频解码器30可自动地确定tile\_boundary\_independence\_flag语法元素的值等于0。

[0138] 在表1的实例语法中,loop\_filter\_across\_tile\_flag语法元素指定视频解码器30是否将执行跨越图像块边界的回路内滤波操作。例如,如果loop\_filter\_across\_tile\_flag语法元素等于1,那么视频解码器30可执行跨越图像块边界的回路内滤波操作。否则,如果loop\_filter\_across\_tile\_flag语法元素等于0,那么视频解码器30可不执行跨越图像块边界的回路内滤波操作。实例回路内滤波操作可包含解块滤波器、样本自适应性偏移和自适应性回路滤波器。

[0139] 如果tile\_mode语法元素等于0(即,每一图片存在仅一个图像块)或等于3(即,使用WPP来完整地编码与SPS相关联的每一图片),那么在表1的实例语法中,SPS不包含loop\_filter\_across\_tile\_flag语法元素。然而,如果tile\_mode语法元素等于0,那么视频解码器30可自动地确定loop\_filter\_across\_tile\_flag语法元素的值等于0。如果tile\_mode语法元素等于3,那么视频解码器30可自动地确定loop\_filter\_across\_tile\_flag语法元素的值等于1。

[0140] 替代地,或除接收包含译码模式语法元素的SPS之外,视频解码器30也可接收包含译码模式语法元素的PPS。在视频解码器30接收应用于相同图片的SPS和PPS且SPS与PPS两者包含译码模式语法元素的一些实例中,视频解码器30可将优先权给予由PPS指定的译码模式语法元素。以下表2呈现包含译码模式语法元素(“tile\_mode”)的PPS的实例语法。

[0141] 表2——图片参数集RBSP语法

|        |                                                                 |  |       |
|--------|-----------------------------------------------------------------|--|-------|
| [0142] | pic_parameter_set_rbsp() {                                      |  | 描述符   |
|        | <b>pic_parameter_set_id</b>                                     |  | ue(v) |
|        | <b>seq_parameter_set_id</b>                                     |  | ue(v) |
|        | <b>num_temporal_layer_switching_point_flags</b>                 |  | ue(v) |
|        | for( i = 0; i < num_temporal_layer_switching_point_flags; i++ ) |  |       |
|        | <b>temporal_layer_switching_point_flag[i]</b>                   |  | u(1)  |
|        | <b>num_ref_idx_l0_default_active_minus1</b>                     |  | ue(v) |
|        | <b>num_ref_idx_l1_default_active_minus1</b>                     |  | ue(v) |
|        | <b>pic_init_qp_minus26</b> /* 相对于 26 */                         |  | se(v) |
|        | <b>constrained_intra_pred_flag</b>                              |  | u(1)  |
|        | <b>slice_granularity</b>                                        |  | u(2)  |
|        | <b>max_cu_qp_delta_depth</b>                                    |  | ue(v) |
|        | <b>weighted_pred_flag</b>                                       |  | u(1)  |
|        | <b>weighted_bipred_idc</b>                                      |  | u(2)  |
|        | <b>tile_partition_info_present_flag</b>                         |  | u(1)  |

[0143]

|                                                                |       |
|----------------------------------------------------------------|-------|
| if( tile_partition_info_present_flag == 1 ) {                  |       |
| <b>tile_mode</b> // 0: 一个图片中的仅一个图像块; 1: 均匀间隔; 2: 不均匀间隔; 3: WPP | u(2)  |
| if( tile_mode == 1    tile_mode == 2 ) {                       |       |
| <b>num_tile_columns_minus1</b>                                 | ue(v) |
| <b>num_tile_rows_minus1</b>                                    | ue(v) |
| }                                                              |       |
| if( tile_mode == 2 ) {                                         |       |
| for( i = 0; i < num_tile_columns_minus1; i++ )                 |       |
| <b>column_width[i]</b>                                         | ue(v) |
| for( i = 0; i < num_tile_rows_minus1; i++ )                    |       |
| <b>row_height[i]</b>                                           | ue(v) |
| }                                                              |       |
| }                                                              |       |
| if( tile_mode == 1    tile_mode == 2 ) {                       |       |
| <b>tile_control_info_present_flag</b>                          | u(1)  |
| if( tile_control_info_present_flag == 1 ) {                    |       |
| <b>tile_boundary_independence_flag</b>                         | u(1)  |
| if( tile_boundary_independence_flag )                          |       |
| <b>loop_filter_across_tile_flag</b>                            | u(1)  |
| }                                                              |       |
| }                                                              |       |
| if( tile_mode == 0 )                                           |       |
| <b>entropy_slice_enabled_flag</b>                              | u(1)  |
| rbsp_trailing_bits()                                           |       |
| }                                                              |       |

[0144] 在表2的实例语法中,如果tile\_partition\_info\_present\_flag语法元素等于1,那么存在tile\_mode语法元素。另外,如果tile\_partition\_info\_present\_flag语法元素等于1,那么num\_tile\_columns\_minus1、num\_tile\_rows\_minus1、column\_width[i]和row\_height[i]语法元素可存在于PPS中。tile\_mode语法元素、num\_tile\_columns\_minus1语法元素、num\_tile\_rows\_minus1语法元素、column\_width语法元素和row\_height的语意可与以上关于表1的实例SPS语法所描述的语意相同。如果tile\_partition\_info\_present\_flag语法元素等于0,那么tile\_mode、num\_tile\_columns\_minus1、num\_tile\_rows\_minus1、column\_width[i]和row\_height[i]语法元素不存在于PPS中。

[0145] 以此方式,视频解码器30可至少部分基于具有指示图片中无CTB是使用WPP来编码的值的译码模式语法元素(例如,tile\_mode),确定参数集(例如,SPS或PPS)包含图像块列数语法元素和图像块行数语法元素。视频解码器30也可基于图像块列数语法元素而确定图像块列的数目。与参数集相关联的每一图片的图像块的列的数目可等于图像块列的数目。视频解码器30也可基于图像块行数语法元素而确定图像块行的数目。与参数集相关联的每一图片的图像块的行的数目可等于图像块行的数目。此外,视频解码器30可确定参数集(例如,SPS或PPS)包含一系列一或多个列宽度语法元素和一系列一或多个图像块高度语法元素。另外,视频解码器30可至少部分基于列宽度语法元素而确定与参数集相关联的每一图片的图像块的列的宽度。此外,视频解码器30可至少部分基于图像块高度语法元素而确定与参数集相关联的每一图片的图像块的高度。

[0146] 类似地,视频编码器20可产生包含图像块列数语法元素和图像块行数语法元素的

参数集。参数集可为图片参数集 (PPS) 或序列参数集 (SPS)。图像块列的数目可基于图像块列数语法元素而确定,且与参数集相关联的每一图片的图像块的列的数目等于图像块列的数目。图像块行的数目可基于图像块行数语法元素而确定,且与参数集相关联的每一图片的图像块的行的数目等于图像块行的数目。在视频编码器20产生参数集时,视频编码器20可产生一系列一或多个列宽度语法元素和一系列一或多个行高度语法元素。与参数集相关联的每一图片的图像块的列的宽度可至少部分基于列宽度语法元素而确定。与参数集相关联的每一图片的图像块的行的高度可至少部分基于行高度语法元素而确定。

[0147] 此外,在表2的实例语法中,如果tile\_control\_info\_present\_flag语法元素等于1,那么tile\_boundary\_independence\_flag和loop\_filter\_across\_tile\_flag语法元素可存在于PPS中。如果tile\_control\_info\_present\_flag语法元素等于0,那么tile\_boundary\_independence\_flag和loop\_filter\_across\_tile\_flag语法元素不存在于PPS中。

[0148] 在表2的实例语法中,如果entropy\_slice\_enabled\_flag等于1,那么参考PPS的经译码切片可包含一或多个熵切片(且可由一或多个熵切片组成)。如果entropy\_slice\_enabled\_flag语法元素等于0,那么参考PPS的经译码切片不含有熵切片。在entropy\_slice\_enabled\_flag语法元素不存在时,视频解码器30可自动地确定(即,推断)entropy\_slice\_enabled\_flag语法元素等于0。PPS的其它语法元素的语意可与HEVCWD5中所定义的语意相同。

[0149] 在表2的实例语法中,如果tile\_mode语法元素等于0,那么PPS仅包含entropy\_slice\_enabled\_flag语法元素。如上文所论述,视频解码器30可基于tile\_mode语法元素而确定是否使用WPP来解码图片的每一图像块的CTB。因此,视频解码器30可基于具有特定值的译码模式语法元素(例如,tile\_mode)而确定位流包含额外语法元素(例如,entropy\_slice\_enabled\_flag),所述额外语法元素指示针对参考包含译码模式语法元素和额外语法元素的参数集(例如,SPS或PPS)的图片的经编码表示是否启用熵切片。

[0150] 如上文所描述,经译码切片NAL单元可包含切片的经译码表示。切片的经译码表示可包含切片标头,继之以切片数据。在一些实例中,视频解码器30可至少部分基于译码模式语法元素(例如,tile\_mode)而确定切片标头是否包含多个入口偏移语法元素,可从所述多个入口偏移语法元素确定切片数据中的子流的入口点。响应于确定切片标头包含入口偏移语法元素,视频解码器30可使用多个入口偏移语法元素来确定切片数据中的子流的入口点。换句话说,视频解码器30可至少部分基于偏移语法元素而确定子流在存储器中的位置。如果译码模式语法元素具有一个值(例如,3),那么由于流中的单一者来表示图片的CTB的每一行。如果译码模式语法元素具有不同值(例如,0、1或2),那么由于流中的单一者来表示具有切片中的一或多个CTB的图片的每一图像块。切片标头可遵照以下表3的实例语法。

[0151] 表3——切片标头语法

[0152]

|                                                                                |       |
|--------------------------------------------------------------------------------|-------|
| slice_header() {                                                               | 描述符   |
| <b>first_slice_in_pic_flag</b>                                                 | u(1)  |
| if( first_slice_in_pic_flag == 0 )                                             |       |
| <b>slice_address</b>                                                           | u(v)  |
| <b>slice_type</b>                                                              | ue(v) |
| <b>pic_parameter_set_id</b>                                                    | ue(v) |
| if( sample_adaptive_offset_enabled_flag    adaptive_loop_filter_enabled_flag ) |       |
| <b>aps_id</b>                                                                  | ue(v) |
| <b>frame_num</b>                                                               | u(v)  |
| if( IdrPicFlag )                                                               |       |
| <b>idr_pic_id</b>                                                              | ue(v) |
| if( pic_order_cnt_type == 0 )                                                  |       |
| <b>pic_order_cnt_lsb</b>                                                       | u(v)  |
| if( slice_type == P    slice_type == B ) {                                     |       |
| <b>num_ref_idx_active_override_flag</b>                                        | u(1)  |
| if( num_ref_idx_active_override_flag ) {                                       |       |
| <b>num_ref_idx_l0_active_minus1</b>                                            | ue(v) |
| if( slice_type == B )                                                          |       |
| <b>num_ref_idx_l1_active_minus1</b>                                            | ue(v) |
| }                                                                              |       |
| }                                                                              |       |
| ref_pic_list_modification()                                                    |       |
| ref_pic_list_combination()                                                     |       |
| if( nal_ref_flag )                                                             |       |
| dec_ref_pic_marking()                                                          |       |
| if( slice_type != I )                                                          |       |
| <b>cabac_init_idc</b>                                                          | ue(v) |
| <b>slice_qp_delta</b>                                                          | se(v) |
| <b>disable_deblocking_filter_flag</b>                                          | u(1)  |
| if( slice_type == B )                                                          |       |
| <b>collocated_from_l0_flag</b>                                                 | u(1)  |
| if( adaptive_loop_filter_enabled_flag && aps_adaptive_loop_filter_flag ) {     |       |
| byte_align()                                                                   |       |
| alf_cu_control_param()                                                         |       |

|        |                                                                                                       |       |
|--------|-------------------------------------------------------------------------------------------------------|-------|
| [0153] | byte_align()                                                                                          |       |
|        | }                                                                                                     |       |
|        | if( ( weighted_pred_flag && slice_type == P )   <br>( weighted_bipred_idc == 1 && slice_type == B ) ) |       |
|        | pred_weight_table()                                                                                   |       |
|        | if( slice_type == P    slice_type == B )                                                              |       |
|        | <b>5 minus max num merge cand</b>                                                                     | ue(v) |
|        | if( tile_mode != 0    entropy_slice_enabled_flag ) {                                                  |       |
|        | <b>num_entry_offsets</b>                                                                              | ue(v) |
|        | if( num_entry_offsets ) {                                                                             |       |
|        | <b>offset_len_minus8</b>                                                                              | ue(v) |
|        | for( i = 0; i < num_entry_offsets; i++ )                                                              |       |
|        | <b>entry_offset[i]</b>                                                                                | u(v)  |
|        | }                                                                                                     |       |
|        | }                                                                                                     |       |
|        | if( entropy_slice_enabled_flag && num_entry_offsets )                                                 |       |
|        | for( i = 0; i < num_entry_offsets; i++ ) {                                                            |       |
|        | <b>entropy_slice_address[i]</b>                                                                       | u(v)  |
|        | if( slice_type != I )                                                                                 |       |
|        | <b>entropy_slice_cabac_init_idc[i]</b>                                                                | ue(v) |
|        | }                                                                                                     |       |
|        | }                                                                                                     |       |

[0154] 在表3的实例语法中,切片标头语法元素“pic\_parameter\_set\_id”、“frame\_num”、“idr\_pic\_id”、“pic\_order\_cnt\_lsb”、“delta\_pic\_order\_cnt[0]”和“delta\_pic\_order\_cnt[1]”的值在经译码图片的所有切片标头中相同。此外,在表3的实例语法中,first\_slice\_in\_pic\_flag语法元素指示切片是否包含涵盖图片的左上方亮度样本的CU。如果first\_slice\_in\_pic\_flag语法元素等于1,那么视频解码器30可将变量SliceAddress与LCUAddress两者设定为0,且视频解码器30可以图片中的第一CTB开始解码。

[0155] 此外,在表3的实例语法中,slice\_address语法元素以切片细微度分辨率指定切片开始的地址。切片细微度分辨率为切片经定义而具有的细微度。slice\_address语法元素的位的数目可等于 $(\text{Ceil}(\text{Log}_2(\text{NumLCUsInPicture})) + \text{SliceGranularity})$ ,其中“NumLCUsInPicture”为图片中的CTB的数目。

[0156] 在表3的实例语法中,视频解码器30将LCUAddress变量设定为 $(\text{slice\_address} > \text{SliceGranularity})$ 。LCUAddress变量指示按光栅扫描次序的切片的切片地址的LCU部分。视频解码器30将GranularityAddress变量设定为 $(\text{slice\_address} - (\text{LCUAddress} < \text{SliceGranularity}))$ 。GranularityAddress变量表示切片地址的子LCU部分。按z扫描次序表达GranularityAddress变量。

[0157] 视频解码器30将SliceAddress变量设定为 $(\text{LCUAddress} < (\log_2\_diff\_max\_min\_coding\_block\_size < 1)) + (\text{GranularityAddress} < ((\log_2\_diff\_max\_min\_coding\_block\_size < 1) - \text{SliceGranularity}))$ 。值 $\log_2\_diff\_max\_min\_coding\_block\_size$ 指定最大CU大小与最小CU大小之间的差。视频解码器30可以切片开始坐标处可能的最大CU开始解码切片。切片开始坐标可为切片的第一CU的左上方像素的坐标。

[0158] 此外,在表3的实例语法中,cabac\_init\_idc语法指定用于确定用于上下文变量的初始化过程中的初始化表的索引。cabac\_init\_idc语法元素的值可在0到2的范围中,包含0和2。

[0159] 在表3的实例语法中,num\_entry\_offsets语法元素指定切片标头中的entry\_offset[i]语法元素的数目。换句话说,多个入口偏移语法元素中的入口偏移语法元素的数目可基于num\_entry\_offsets语法元素而确定。当不存在num\_entry\_offsets语法元素时,视频解码器30可确定num\_entry\_offsets语法元素的值等于0。以此方式,视频解码器30可基于num\_entry\_offsets语法元素而确定在多个入口偏移语法元素中有多少个偏移语法元素。offset\_len\_minus8语法元素加上8指定entry\_offset[i]语法元素的以位为单位的长度。换句话说,入口偏移语法元素中的每一者的以位为单位的长度可基于offset\_len\_minus8语法元素而确定。以此方式,视频解码器30可基于offset\_len\_minus8语法元素而确定偏移语法元素的以位为单位的长度。entry\_offset[i]语法元素指定第i个入口偏移(以字节为单位)。

[0160] 视频解码器30可至少部分基于多个偏移语法元素中有多少个偏移语法元素和偏移语法元素的以位为单位的长度,自位流剖析偏移语法元素。经译码切片NAL单元中的子流的数目可等于num\_entry\_offsets+1。子流的索引值可在0到num\_entry\_offsets的范围中,包含0和num\_entry\_offsets。经译码切片NAL单元的子流0可由以下各者组成:经译码切片NAL单元的切片数据的字节0到entry\_offset[0]-1,包含0和entry\_offset[0]-1。经译码切片NAL单元的子流k(其中k在1到num\_entry\_offsets-1的范围中,包含1和num\_entry\_offsets-1)可由以下各者组成:经译码切片NAL单元的切片数据的字节entry\_offset[k-1]到entry\_offset[k]-1,包含entry\_offset[k-1]和entry\_offset[k]-1。经译码切片NAL单元的最后子流(其中子流索引等于num\_entry\_offsets)可由经译码切片NAL单元的切片数据的字节中的其余部分组成。

[0161] 在表3的实例语法中,如果tile\_mode语法元素大于0,那么具有在1到num\_entry\_offsets-1的范围中的子流索引的每一子流含有一个图像块的每一经译码位,且具有子流索引0的子流含有图像块的每一经译码位或图像块的若干个末尾经译码位。图像块的末尾经译码位为在图像块的末尾处经译码的经译码位。此外,如果tile\_mode语法元素大于0,那么最后子流(即,具有等于num\_entry\_offsets的子流索引的子流)含有图像块的所有经译码位或图像块的若干个开始经译码位。图像块的开始经译码位为在图像块的开始处经译码的经译码位。子流不含有一个以上图像块的经译码位。在表3的实例语法中,子流0中始终包含经译码切片NAL单元的NAL单元标头和切片标头。如果tile\_mode语法元素等于0且entropy\_slice\_enabled\_flag语法元素等于1,那么每一子流含有一个熵切片的每一经译码位且不含有另一熵切片的任何经译码位。

[0162] 在表3的实例语法中,entropy\_slice\_address[i]语法元素以切片细微度分辨率指定经译码切片NAL单元中的第(i+1)个熵切片的开始地址。entropy\_slice\_address[i]语法元素中的每一者的以位为单位的大小可等于(Ceil(Log2(NumLCUsInPicture))+SliceGranularity)。

[0163] 此外,在表3的实例语法中,“entropy\_slice\_cabac\_init\_idc[i]”语法元素指定经译码切片NAL单元中的第(i+1)个熵切片的用于确定用于上下文变量的初始化过程中的初始化表的索引。entropy\_slice\_cabac\_init\_idc[i]的值在0到2的范围中,包含0和2。切片标头的其它语法元素的语意可与HEVC WD5中所定义的语意相同。

[0164] 在一些实例中,entry\_offset[i]语法元素指示就位来说的子流的偏移。此外,在

一些实例中,切片标头可包含指示“entry\_offset[i]”的单位为字节(在等于1时)或是位(在等于0时)的旗标。此旗标可位于切片标头中在offset\_len\_minus8语法元素之后。

[0165] 此外,在一些实例中,切片标头可包含每一子流(包含子流0)的语法元素,所述语法元素用以指示相应子流的子流类型。在此实例中,如果子流的语法元素具有第一值,那么子流为图像块。如果子流的语法元素具有第二值,那么子流为熵切片。

[0166] 如上文所提及,经译码表示可包含切片标头和切片数据。切片数据可包含一或多个子流。如果译码模式语法元素具有第一值(例如,3),那么图片的CTB的每一行由子流中的单一者来表示。如果语法元素具有第二值(例如,0、1或2),那么具有切片中的一或多个CTB的图片的每一图像块由子流中的单一者来表示。为了促进WPP或平行地解码切片的图像块,切片数据中的子流可包含确保子流的字节对准的填补位。然而,在图片中存在仅一个图像块且不启用熵切片的例子中,可能不需要包含这些填补位。因此,视频解码器30可至少部分基于译码模式语法元素(例如,tile\_mode)而确定切片数据中的子流是否包含确保子流的字节对准的填补位。

[0167] 切片数据可遵照以下表4的实例语法。

[0168] 表4——切片数据语法

[0169]

| slice_data() {                                                      | 描述符   |
|---------------------------------------------------------------------|-------|
| CurrTbAddr=LCUAddress                                               |       |
| moreDataFlag=1                                                      |       |
| if (adaptive_loop_filter_flag&&alf_cu_control_flag)                 |       |
| AlfCuFlagIdx=-1                                                     |       |
| subStreamIdx=0                                                      |       |
| do {                                                                |       |
| xCU=HorLumaLocation (CurrTbAddr)                                    |       |
| yCU=VerLumaLocation (CurrTbAddr)                                    |       |
| moreDataFlag=coding_tree (xCU,yCU,Log2TbSize,0)                     |       |
| CurrTbAddr=NextTbAddress (CurrTbAddr)                               |       |
| if (tile_mode!=0    entropy_slice_enabled_flag) {                   |       |
| byteIdx=byte_index ()                                               |       |
| if (byte_aligned ()&&byteIdx==entry_offset[subStreamIdx])           |       |
| subStreamIdx++                                                      |       |
| else if (!byte_aligned ()&&byteIdx==entry_offset[subStreamIdx]-1) { |       |
| while (!byte_aligned ())                                            |       |
| bit_equal_to_one                                                    | f (1) |
| subStreamIdx++                                                      |       |
| }                                                                   |       |
| moreDataFlag=moreDataFlag&& (subStreamIdx==num_entry_offsets)       |       |
| }                                                                   |       |
| } while (moreDataFlag)                                              |       |

|   |  |
|---|--|
| } |  |
|---|--|

[0170] 在表4的实例语法中,切片数据包含coding\_tree()函数。在视频解码器30剖析切片数据时,视频解码器30可执行回圈。在回圈的每一反复期间,视频解码器30调用coding\_tree()函数以剖析切片数据中的经译码CTB。在视频解码器30调用coding\_tree()函数以剖析特定经译码CTB时,视频解码器30可自切片数据剖析end\_of\_slice\_flag语法元素。如果end\_of\_slice\_flag语法元素等于0,那么在切片或熵切片中存在特定经译码CTB之后的另一CTB。如果end\_of\_slice\_flag语法元素等于1,那么特定经译码CTB为切片或熵切片的最后经译码CTB。

[0171] 此外,表4的实例语法包含byte\_index()函数。byte\_index()函数可返回NAL单元的位内的当前位置的字节索引。NAL单元的位内的当前位置可为NAL单元的第一未经剖析位。如果位流中之下一位为NAL单元标头的第一字节的任何位,那么byte\_index()函数返回等于0的值。

[0172] 表4的切片数据语法为实例。在切片数据语法的其它实例中,用条件“if(tile\_mode==1||tile\_mode==2||entropy\_slice\_enabled\_flag)”来替换表4的条件“if(tile\_mode!=0||entropy\_slice\_enabled\_flag)”。

[0173] 图4为说明根据本发明的一或多个方面的用于编码视频数据的视频编码器20的实例操作200的流程图,其中不允许单一图片内的图像块与WPP波的组合。作为实例提供图4。在其它实例中,可使用比图4的实例中所展示的步骤多、比所述步骤少或不同于所述步骤的步骤来实施本发明的技术。

[0174] 在图4的实例中,视频编码器20通过根据第一译码模式编码图片产生第一经译码图片(202)。在视频编码器20根据第一译码模式编码图片时,使用WPP完整地编码图片。另外,视频编码器20可通过根据第二译码模式编码图片产生第二经译码图片(204)。在视频编码器20根据第二译码模式编码图片时,视频编码器20可将图片分割成一或多个图像块。视频编码器20可在不使用WPP的情况下编码图片的每一图像块(即,编码图像块中的每一者中的每一CTB)。例如,视频编码器20可在不使用WPP的情况下根据光栅扫描次序编码图像块中的每一者的CTB。视频编码器20接着可选择第一经译码图片或第二经译码图片(206)。在一些实例中,视频编码器20可基于第一经译码图片和第二经译码图片的位率/失真分析而选择第一经译码图片或第二经译码图片。视频编码器20可产生位流,位流包含所选定经译码图片和指示根据第一译码模式或是第二译码模式编码图片的语法元素(208)。

[0175] 图5为说明根据本发明的一或多个方面的用于解码视频数据的视频解码器30的实例操作220的流程图,其中不允许单一图片内的图像块与WPP的组合。作为实例提供图5。

[0176] 在图5的实例中,视频解码器30可自包含视频数据中的图片的经译码表示的位流剖析语法元素(222)。视频解码器30可确定语法元素是否具有特定值(224)。响应于确定所述语法元素具有特定值(224的“是”),视频解码器30可使用WPP来完整地解码所述图片(226)。响应于确定所述语法元素不具有所述特定值(224的“否”),视频解码器30可在不使用WPP的情况下解码所述图片的每一图像块,其中图片具有一或多个图像块(228)。

[0177] 图6为说明根据本发明的一或多个方面的用于解码视频数据的视频解码器30的实例操作230的流程图,其中不允许单一图片内的图像块与WPP的组合。作为实例提供图6。在其它实例中,可使用比图6的实例中所展示的步骤多、比所述步骤少或不同于所述步骤的步

骤来实施本发明的技术。图6可为图5的操作220的更特定实例。

[0178] 在图6的实例中,视频解码器30接收位流(231)。视频解码器30可自位流剖析语法元素(232)。在一些实例中,位流包含包含语法元素的SPS。在其它实例中,位流包含包含语法元素的PPS。

[0179] 随后,视频解码器30可确定语法元素是否具有第一值(例如,0)(234)。在图6的实例中,如果语法元素具有第一值(234的“是”),那么图片具有单一图像块,且视频解码器30可在不使用WPP的情况下解码图片的单一图像块(236)。

[0180] 然而,如果语法元素不具有第一值(234的“否”),那么视频解码器30可确定语法元素是否具有第二值(例如,1)(238)。响应于确定语法元素具有第二值(238的“是”),视频解码器30可确定图片具有多个均匀隔开的图像块,且视频解码器30可在不使用WPP的情况下解码均匀隔开的图像块中的每一者(240)。

[0181] 另一方面,如果语法元素不具有第二值(238的“否”),那么视频解码器30可确定语法元素是否具有第三值(例如,2)(242)。响应于确定语法元素具有第三值(242的“是”),视频解码器30可确定图片具有多个不均匀隔开的图像块,且视频解码器30可在不使用WPP的情况下解码图片的不均匀隔开的图像块(244)。然而,响应于确定所述语法元素不具有第三值(242的“否”),视频解码器30可使用WPP来完整地解码所述图片(246)。以此方式,如果语法元素具有第一值(例如,3),那么使用WPP完整地编码图片,且如果语法元素具有不同于第一值的第二值(例如,0、1或2),那么将图片分割成一或多个图像块,且在不使用WPP的情况下编码图片。

[0182] 图7为说明根据本发明的一或多个方面的用于编码视频数据的视频编码器20的实例操作270的流程图,其中图片的CTB的每一列在单独子流中。在一些视频译码系统中,存在用信号发出图像块和WPP波的入口点的不同方式。此可增加这些视频译码系统的复杂度。本发明的技术(且如关于图7和图8所解释)可通过提供用于指示图像块、WPP波和(在一些实例中)熵切片的入口点的统一语法来解决这些问题。

[0183] 在图7的实例中,视频编码器20用信号发出:使用WPP来编码视频图片序列中的图片(272)。视频编码器20可以各种方式用信号发出使用WPP来编码图片。例如,视频编码器20可产生包含指示是否使用WPP来完整地解码图片的语法元素(例如,“tile\_mode”)的SPS。在另一实例中,视频编码器20可产生包含指示是否使用WPP来解码图片的语法元素(例如,“tile\_mode”)的PPS。

[0184] 此外,视频编码器20可执行WPP以产生多个子流(274)。子流中的每一者可包含表示图片的切片中的CTB的一个经编码行的位的连续系列。因此,CTB的每一行经编码为一个子流。视频编码器20可产生包含多个子流的经译码切片NAL单元(276)。经译码切片NAL单元可包含遵照上文表3和表4的实例语法的切片标头和切片数据。

[0185] 图8为说明根据本发明的一或多个方面的用于解码视频数据的视频解码器30的实例操作280的流程图,其中图片的CTB的每一行在单独子流中。在图8的实例中,视频解码器30接收包含经译码切片NAL单元的位流(282)。经译码切片NAL单元包含多个子流。子流中的每一者可包含表示图片的切片中的CTB的一个行的位的连续系列。此外,在图8的实例中,视频解码器30基于位流中的一或多个语法元素而确定切片是使用WPP而编码(284)。例如,视频解码器30可基于等于3的tile\_mode语法元素而确定切片是使用WPP而编码。在此实例中,

如果tile\_mode语法元素不等于3,那么视频解码器30可在不使用WPP的情况下解码图片的一或多个图像块中的每一者。

[0186] 接下来,视频解码器30可使用WPP解码切片(286)。在视频解码器30解码切片时,视频解码器30可剖析与切片的CTB相关联的语法元素。视频解码器30可作为剖析与CTB相关联的语法元素的部分对语法元素中的一些语法元素执行CABAC剖析过程。

[0187] 图9A为说明根据本发明的一或多个方面的用以剖析切片数据的实例CABAC剖析过程300的第一部分的流程图。当剖析切片数据中和译码树语法中的具有描述符ae(v)的语法元素时,视频解码器30可执行图9A的过程。图9A的过程可输出语法元素的值。

[0188] 在图9A的实例中,视频解码器30的熵解码单元150执行CABAC剖析过程的初始化(302)。在一些实例中,CABAC剖析过程的初始化与HEVC WD5的子条款9.2.1中所描述的初始化相同。

[0189] 另外,熵解码单元150可确定相邻CTB的地址(304)。相邻CTB可为含有在左方、左上方、上方或右上方相邻于当前CTB(即,视频解码器30当前正解码的CTB)的块的CTB。在一些实例中,熵解码单元150可将相邻CTB的地址确定为:

[0190] 
$$tbAddrT = cuAddress(x0 + 2 * (1 \ll \log2MaxCUsSize) - 1, y0 - 1)$$

[0191] 在以上公式中,tbAddrT表示相邻CTB的地址,x0表示当前CTB的左上方明度样本的x坐标,y0表示当前CTB的左上方明度样本的y坐标,且Log2MaxCUsSize表示CU的最大大小的底数为2的对数。函数cuAddress返回CU的地址,所述地址包含由第一参数指定的x坐标和由第二参数指定的y坐标。

[0192] 接下来,熵解码单元150可使用相邻CTB的地址来确定用于图片中预测的相邻CTB的可用性(306)。换句话说,熵解码单元150可确定与相邻CTB相关联的信息是否可用于在选择CABAC上下文中使用。

[0193] 熵解码单元150可以各种方式确定用于图片中预测的相邻CTB的可用性。例如,熵解码单元150可执行WD5的子条款6.4.3中所描述的过程(其中tbAddrT作为输入),以确定用于图片中预测的相邻CTB的可用性。在另一实例中,除非以下条件中的一者为真,否则熵解码单元150可确定CTB可用于图片中预测。如果以下条件中的一者为真,那么熵解码单元150可确定CTB不可用于图片中预测。第一,如果CTB的地址小于0,那么熵解码单元150可确定CTB不可用于图片中预测。第二,如果CTB的地址大于熵解码单元150当前正剖析的CTB的地址,那么熵解码单元150可确定CTB不可用于图片中预测。第三,如果特定CTB属于不同于熵解码单元150当前正剖析的CTB的切片,那么熵解码单元150可确定特定CTB不可用于图片中预测。例如,如果将特定CTB的地址表示为tbAddr且将熵解码单元150当前正剖析的CTB的地址表示为CurrTbAddr,那么熵解码单元150可确定具有地址tbAddr的CTB属于不同于具有地址CurrTbAddr的CTB的切片。第四,如果位流中的一或多个语法元素指示可独立地解码视频解码器30当前正解码的图片的图像块,且CTB在不同于熵解码单元150当前正剖析的CTB的图像块中,那么熵解码单元150可确定CTB不可用于图片中预测。例如,如果表1的实例语法的tile\_boundary\_independence\_flag语法元素等于1且具有地址tbAddr的CTB含于不同于具有地址CurrTbAddr的CTB的图像块中,那么熵解码单元150可确定CTB不可用于图片中预测。

[0194] 此外,熵解码单元150可确定熵解码单元150当前正剖析的语法元素(即,当前语法

元素)是否在译码树语法结构中(308)。如果当前语法元素不在译码树语法结构中(308的“否”),那么熵解码单元150可执行图9B中所展示的可BAC剖析过程300的部分。另一方面,如果当前语法元素在译码树结构中(308的“是”),那么熵解码单元150可确定是否可独立地解码当前图片(即,包含当前CTB的图片)的图像块(310)。例如,在表1的实例SPS语法中,如果与当前图片相关联的SPS包含等于1的tile\_boundary\_independence\_flag语法元素,那么熵解码单元150可确定可独立地解码当前图片的图像块。响应于确定可独立地解码当前图片的图像块(310的“是”),熵解码单元150可执行图9B中所展示的可BAC剖析过程300的部分。

[0195] 然而,响应于确定不可独立地解码当前图片的图像块(310的“否”),熵解码单元150可确定 $tbAddr \% picWidthInLCUs$ 是否等于0,其中tbAddr为相邻CTB的地址,%表示模数运算符,且picWidthInLCUs指示CTB(即,LCU)中的当前图片的宽度(312)。

[0196] 响应于确定 $tbAddr \% picWidthInLCUs$ 等于0(312的“是”),熵解码单元150可确定相邻CTB是否可用于图片中预测(314)。在一些实例中,熵解码单元150可在动作306中执行过程以确定指示相邻CTB是否可用于图片中预测的变量availableFlagT的值。如果变量availableFlagT等于1,那么相邻CTB可用于图片中预测。在动作314中,熵解码单元150可确定变量availableFlagT是否等于1。

[0197] 响应于确定相邻CTB可用于图片中预测(314的“是”),熵解码单元150可执行可BAC剖析过程的同步过程(316)。在一些实例中,熵解码单元150可执行HEVC WD5的子条款9.2.1.3中所描述的同步过程。在执行同步过程之后或响应于确定相邻CTB不可用于图片中预测(314的“否”),熵解码单元150可执行在终止之前的针对二进制决策的解码过程(318)。大体来说,在终止之前的针对二进制决策的解码过程为用于熵解码end\_of\_slice\_flag和pcm\_flag语法元素的特殊解码过程。视频解码器30可使用end\_of\_slice\_flag和pcm\_flag在剖析切片数据的过程的终止之前作出二进制决策。在一些实例中,熵解码单元150可执行如HEVC WD5的子条款9.2.3.2.4中所指定的在终止之前的针对二进制决策的解码过程。

[0198] 在执行在终止之前的二进制决策的解码过程(318)之后,熵解码单元150可执行算术解码引擎的初始化过程(320)。在一些实例中,熵解码单元150可执行HEVC WD5的子条款9.2.1.4中所定义的初始化过程。在执行算术解码引擎的初始化过程之后,熵解码单元150可执行图9B中所展示的可BAC剖析过程300的部分。

[0199] 如果 $tbAddr \% picWidthInLCUs$ 不等于0(312的“否”),那么熵解码单元150可确定 $tbAddr \% picWidthInLCUs$ 是否等于2(322)。换句话说,熵解码单元150可确定相邻CTB的CTB地址模除CTB中的当前图片的宽度所得值是否等于2。响应于确定 $tbAddr \% picWidthInLCUs$ 不等于2,熵解码单元150可执行图9B中所展示的可BAC剖析过程300的部分。然而,响应于确定 $tbAddr \% picWidthInLCUs$ 等于2(322的“是”),熵解码单元150可执行记忆过程(324)。大体来说,记忆过程输出用于指派给不同于end\_of\_slice\_flag语法元素的语法元素的上下文变量的初始化过程中的变量。在一些实例中,熵解码单元150可执行HEVC WD5的子条款9.2.1.2中所定义的记忆过程。在执行记忆过程之后,熵解码单元150可执行图9B中所展示的可BAC剖析过程300的部分。

[0200] 图9B为说明图9A的实例可BAC剖析过程300的接续部分的流程图。如图9B中所展示,熵解码单元150可使当前语法元素二进制化(330)。换句话说,熵解码单元150可导出当

前语法元素的二进制定化。语法元素的二进制定化可为语法元素的所有可能值的分格字串的集合。分格字串是为来自语法元素的二进制定化的语法元素的值的中间表示的分格的字串。在一些实例中,熵解码单元150可执行HEVC WD5的子条款9.2.2中所定义的过程,以导出当前语法元素的二进制定化。

[0201] 另外,熵解码单元150可确定译码过程流程(332)。熵解码单元150可基于当前语法元素的二进制定化和经剖析分格的序列而确定译码过程流程。在一些实例中,熵解码单元150可确定如HEVC WD5的子条款9.2.2.9中所描述的译码过程流程。

[0202] 此外,熵解码单元150可确定当前语法元素的二进制定化的每一分格的上下文索引(334)。由变量binIdx对当前语法元素的二进制定化的分格中的每一者标以索引,且可将当前语法元素的二进制定化的分格的上下文索引表示为ctxIdx。在一些实例中,熵解码单元150可确定如HEVC WD5的子条款9.2.3.1中所指定的当前语法元素的二进制定化的分格的上下文索引。

[0203] 熵解码单元150可执行每一上下文索引的算术解码过程(336)。在一些实例中,熵解码单元150可执行如HEVC WD5的子条款9.2.3.2中所指定的每一上下文索引的算术解码过程。通过执行每一上下文索引的算术解码过程,熵解码单元150可产生经剖析分格的序列。

[0204] 熵解码单元150可确定经剖析分格的序列是否匹配由当前语法元素的二进制定化产生的分格字串的集合中的分格字串(340)。如果经剖析分格的序列匹配由当前语法元素的二进制定化产生的分格字串的集合中的分格字串(340的“是”),那么熵解码单元150可将对应值指派给当前语法元素(342)。在将对应值指派给当前语法元素之后或响应于确定经剖析分格的序列不匹配通过当前语法元素的二进制定化产生的分格字串的集合中的任何分格字串(340的“否”),熵解码单元150已完成剖析当前语法元素。

[0205] 在一些实例中,如果当前语法元素为mb\_type语法元素,且mb\_type语法元素的经解码值等于I\_PCM,那么可在解码任何pcm\_alignment\_zero\_bit语法元素和所有pcm\_sample\_luma和pcm\_sample\_chroma数据之后初始化熵解码单元150,如HEVC WD5的子条款9.2.1.2中所指定。

[0206] 图10为说明WPP的实例的概念图。如上文所描述,可将图片分割成像素块,所述像素块中的每一者与CTB相关联。图10将与CTB相关联的像素块说明为白色正方形的栅格。图片包含CTB行350A到350E(统称为“CTB行350”)。

[0207] 第一平行处理线程(例如,由多个平行处理核心中的一者执行)可能正译码CTB行350A中的CTB。与此同时,其它线程(例如,由其它平行处理核心执行)可能正译码CTB行350B、350C和350D中的CTB。在图10的实例中,第一线程当前正译码CTB352A,第二线程当前正译码CTB352B,第三线程当前正译码CTB352C,且第四线程当前正译码CTB352D。本发明可将CTB352A、352B、352C和352D通称为“当前CTB352”。因为视频译码器可在已译码紧接的较高行的两个以上CTB之后开始译码CTB行,所以将当前CTB352彼此水平地位移达两个CTB的宽度。

[0208] 在图10的实例中,线程可使用来自粗灰色箭头指示的CTB的数据来对当前CTB352中的CU执行帧内预测或帧间预测。(线程也可使用来自一或多个参考帧的数据来执行CU的帧间预测。)为了译码给定CTB,线程可基于与先前经译码CTB相关联的信息而选择一

或多个CABAC上下文。线程可使用一或多个CABAC上下文来对与给定CTB的第一CU相关联的语法元素执行CABAC译码。如果给定CTB并非行的最左CTB,那么线程可基于与在给定CTB左方的CTB的最后CU相关联的信息而选择一或多个CABAC上下文。如果给定CTB为行的最左CTB,那么线程可基于与在给定CTB上方和右方的两个CTB处的CTB的最后CU相关联的信息而选择一或多个CABAC上下文。线程可使用来自通过细黑色箭头指示的CTB的最后CU的数据来选择当前CTB352的第一CU的CABAC上下文。

[0209] 图11为说明经分割成多个图像块402A、402B、402C、402D、402E和402F(统称为“图像块402”)的图片400的实例CTB译码次序的概念图。图片400中的每一正方形块表示与CTB相关联的像素块。粗虚线指示实例图像块边界。不同类型的交叉影线对应于不同切片。

[0210] 像素块中的编号指示按图片400的图像块译码次序的对应CTB(LCU)的位置。如图11的实例中所说明,首先译码图像块402A中的CTB,接着译码图像块402B中的CTB,接着译码图像块402C中的CTB,接着译码图像块402D中的CTB,接着译码图像块402E中的CTB,接着译码图像块402F中的CTB。在图像块402中的每一者内,根据光栅扫描次序译码CTB。

[0211] 视频编码器可针对图片400产生四个经译码切片NAL单元。第一经译码切片NAL单元可包含CTB1到18的经编码表示。第一经译码切片NAL单元的切片数据可包含两个子流。第一子流可包含CTB1到9的经编码表示。第二子流可包含CTB10到18的经编码表示。因此,第一经译码切片NAL单元可包含含有多个图像块的切片的经编码表示。

[0212] 第二经译码切片NAL单元可包含CTB19到22的经编码表示。第二经译码切片NAL单元的切片数据可包含单一子流。第三经译码切片NAL单元可包含CTB23到27的经编码表示。第三经译码切片NAL单元的切片数据可包含仅单一子流。因此,图像块402C可含有多个切片。

[0213] 第四经译码切片NAL单元可包含CTB28到45的经编码表示。第四经译码切片NAL单元的切片数据可包含三个子流,图像块402D、402E和402F中的每一者一个子流。因此,第四经译码切片NAL单元可包含含有多个图像块的切片的经编码表示。

[0214] 在一或多个实例中,所描述的功能可以硬件、软件、固件或其任何组合来实施。如果以软件来实施,那么所述功能可作为一或多个指令或代码而存储于计算机可读媒体上或经由计算机可读媒体来传输,且由基于硬件的处理单元来执行。计算机可读媒体可包含计算机可读存储媒体或通信媒体,计算机可读存储媒体对应于例如数据存储媒体的有形媒体,通信媒体包含促进计算机程序(例如)根据通信协议自一处传送到另一处的任何媒体。以此方式,计算机可读媒体大体上可对应于:(1)非暂时性的有形计算机可读存储媒体,或(2)例如信号或载波的通信媒体。数据存储媒体可为可由一或多个计算机或一或多个处理器存取以检索用于实施本发明中所描述的技术的指令、代码和/或数据结构的任何可用媒体。计算机程序产品可包含计算机可读媒体。

[0215] 作为实例而非限制,这些计算机可读存储媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置、快闪存储器,或可用于存储呈指令或数据结构的形式的所要程序代码且可由计算机存取的任何其它媒体。又,任何连接可适当地被称作计算机可读媒体。例如,如果使用同轴电缆、光纤缆线、双绞线、数字订户线(DSL)或例如红外线、无线电和微波的无线技术而自网站、服务器或其它远端源传输指令,那么同轴电缆、光纤缆线、双绞线、DSL或例如红外线、无线电和微波的无线技术包含于媒体

的定义中。然而,应理解,计算机可读存储媒体和数据存储媒体不包含连接、载波、信号或其它暂时性媒体,而是针对非暂时性有形存储媒体。如本文中所使用,磁盘和光盘包含压缩光盘(CD)、雷射光盘、光学光盘、数字影音光盘(DVD)、软性磁盘和蓝光光盘,其中磁盘通常以磁性方式再现数据,而光盘通过雷射以光学方式再现数据。以上各者的组合也应包含于计算机可读媒体的范围内。

[0216] 可通过例如一或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程逻辑阵列(FPGA)或其它等效集成或离散逻辑电路的一或多个处理器来执行指令。因此,如本文中所使用,术语“处理器”可指前述结构或适于实施本文中所描述的技术的任何其它结构中的任一者。另外,在一些方面,可将本文所描述的功能性提供于经配置以用于编码和解码的专用硬件和/或软件模块内,或并入于组合式编码解码器中。又,所述技术可完全实施于一或多个电路或逻辑元件中。

[0217] 本发明的技术可实施于广泛多种装置或设备中,所述装置或设备包含无线手持机、集成电路(IC)或IC的集合(例如,芯片组)。本发明中描述各种组件、模块或单元以强调经配置以执行所揭示的技术的装置的功能方面,但未必需要通过不同硬件单元实现。相反地,如上文所描述,各种单元可组合于编码解码器硬件单元中或由互操作性硬件单元的集合(包含如上文所描述的一或多个处理器)结合合适的软件和/或固件来提供。

[0218] 已描述各种实例。这些和其它实例在以下权利要求书的范围内。

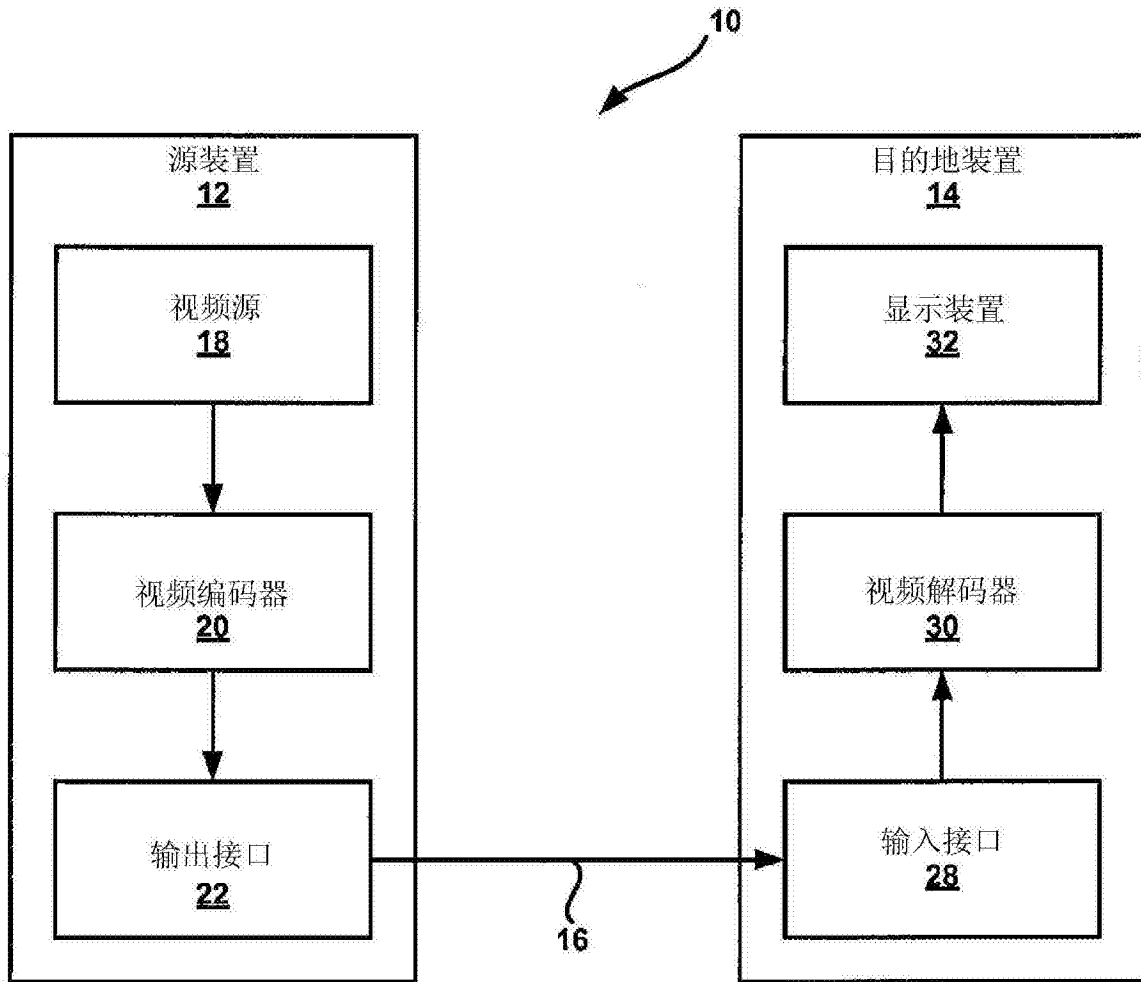


图1

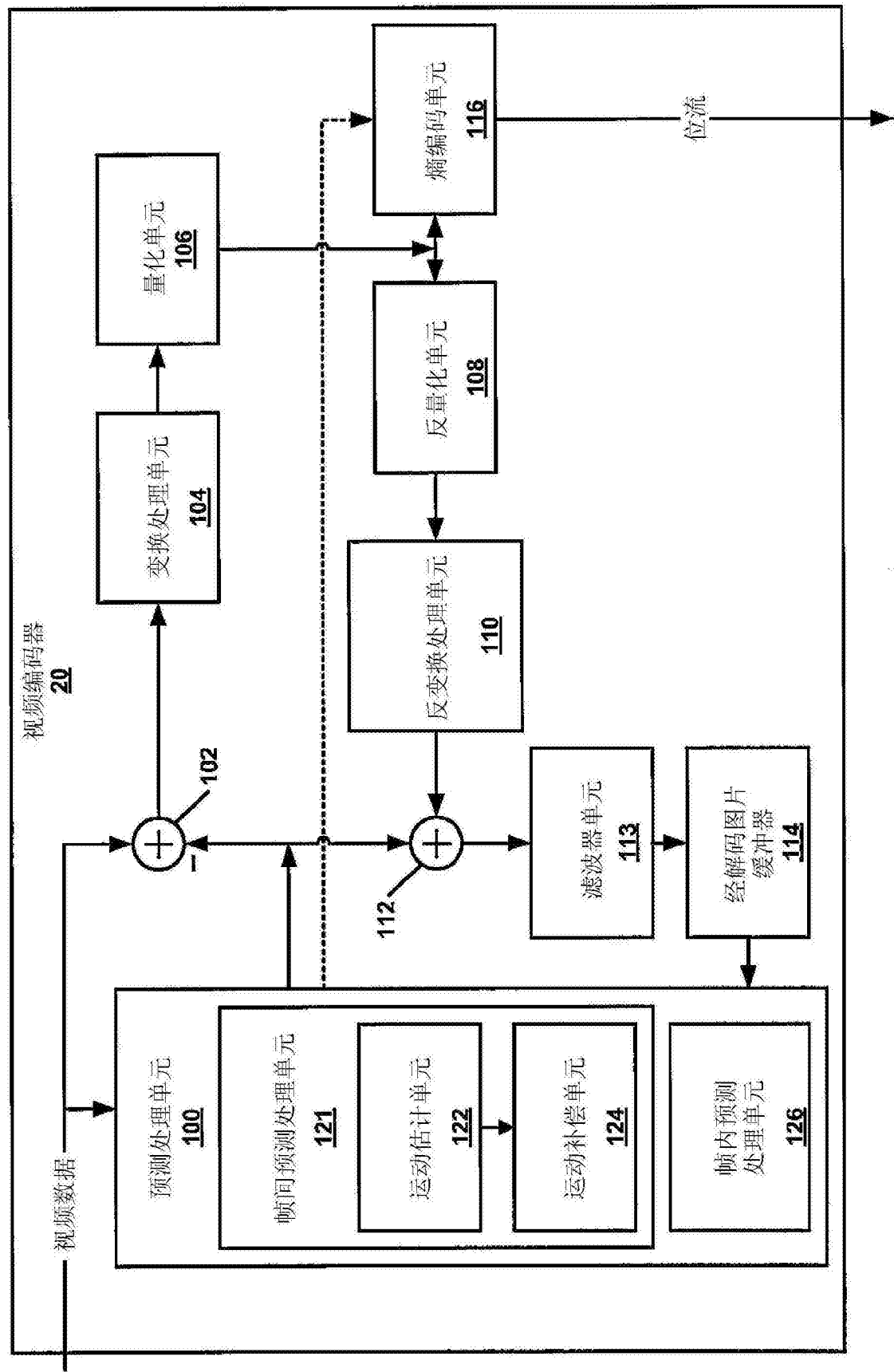


图2

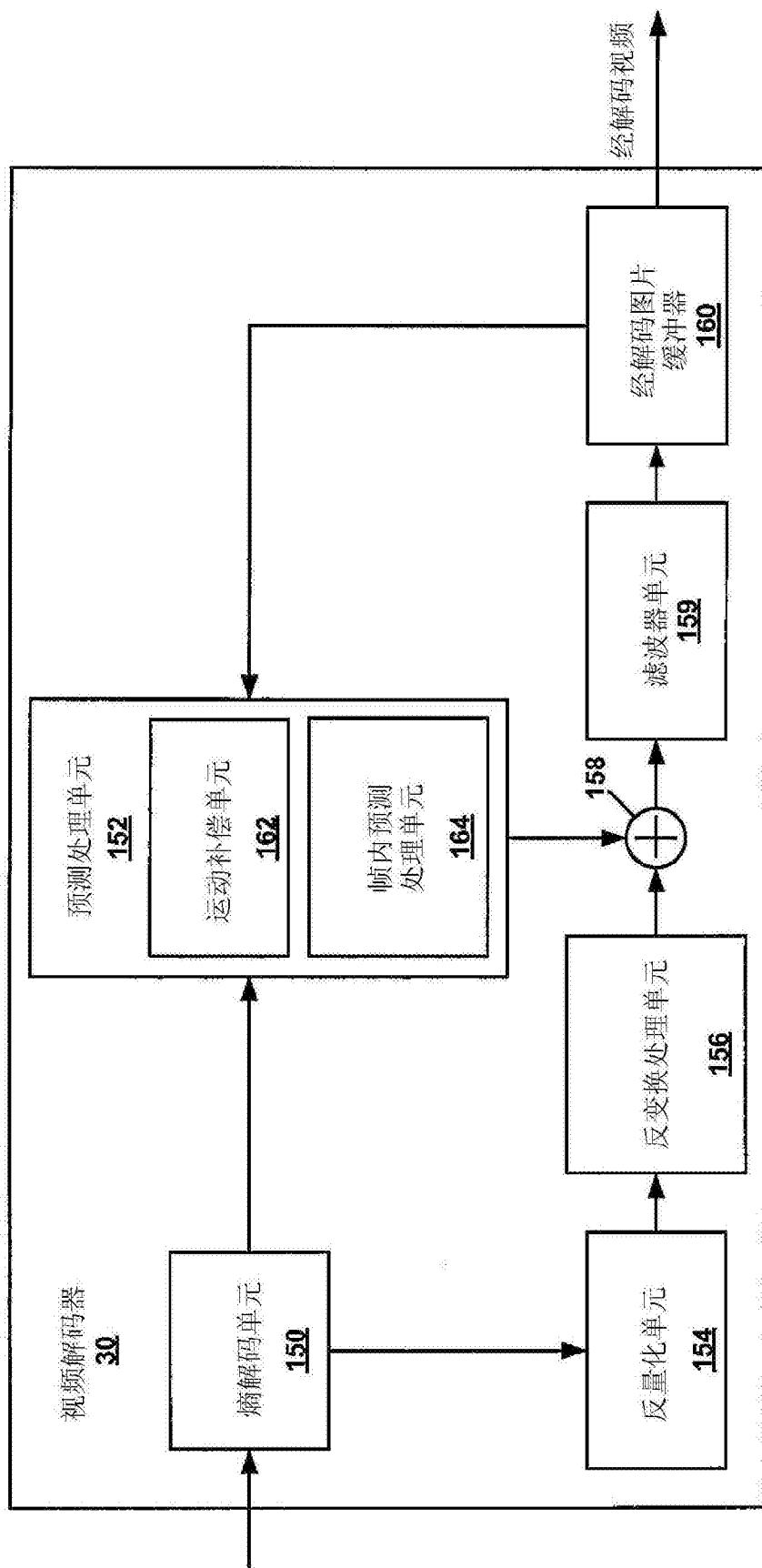


图3

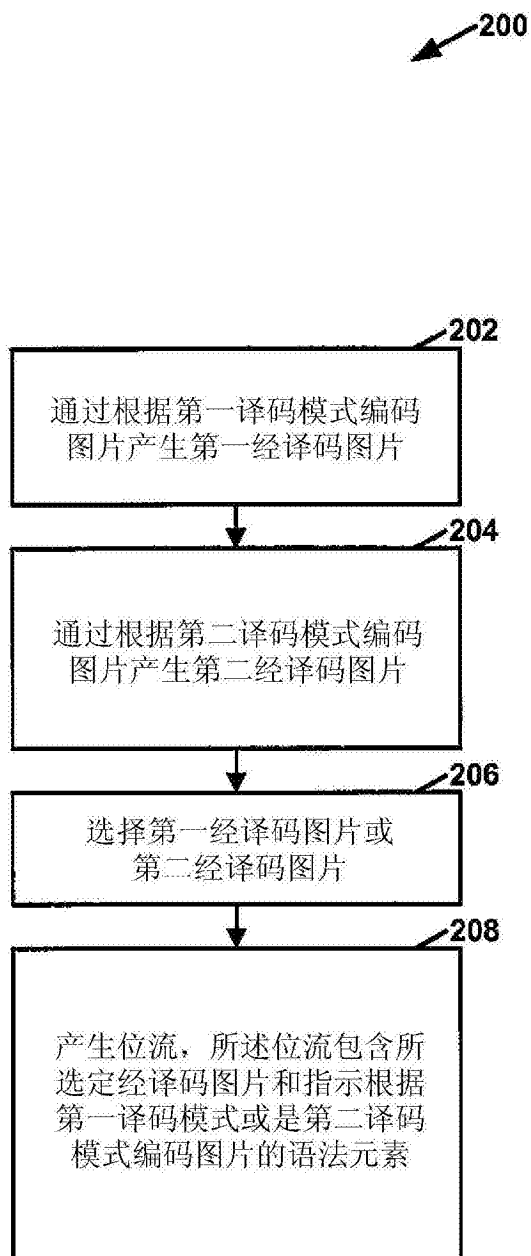


图4

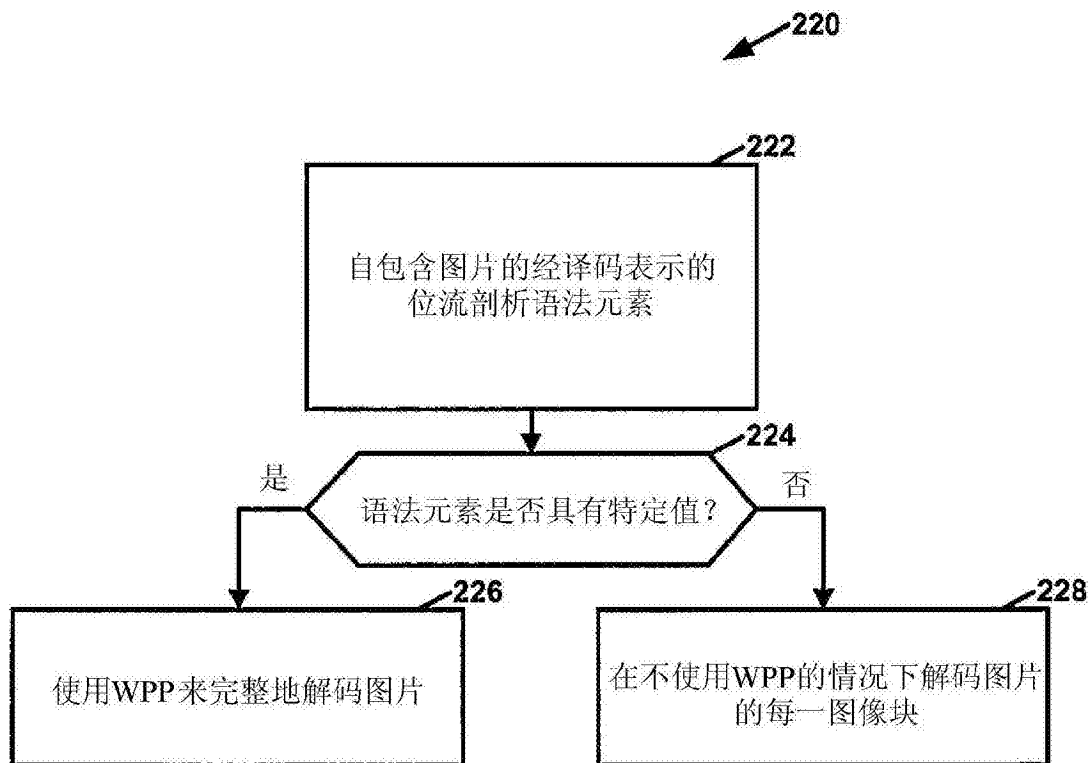


图5

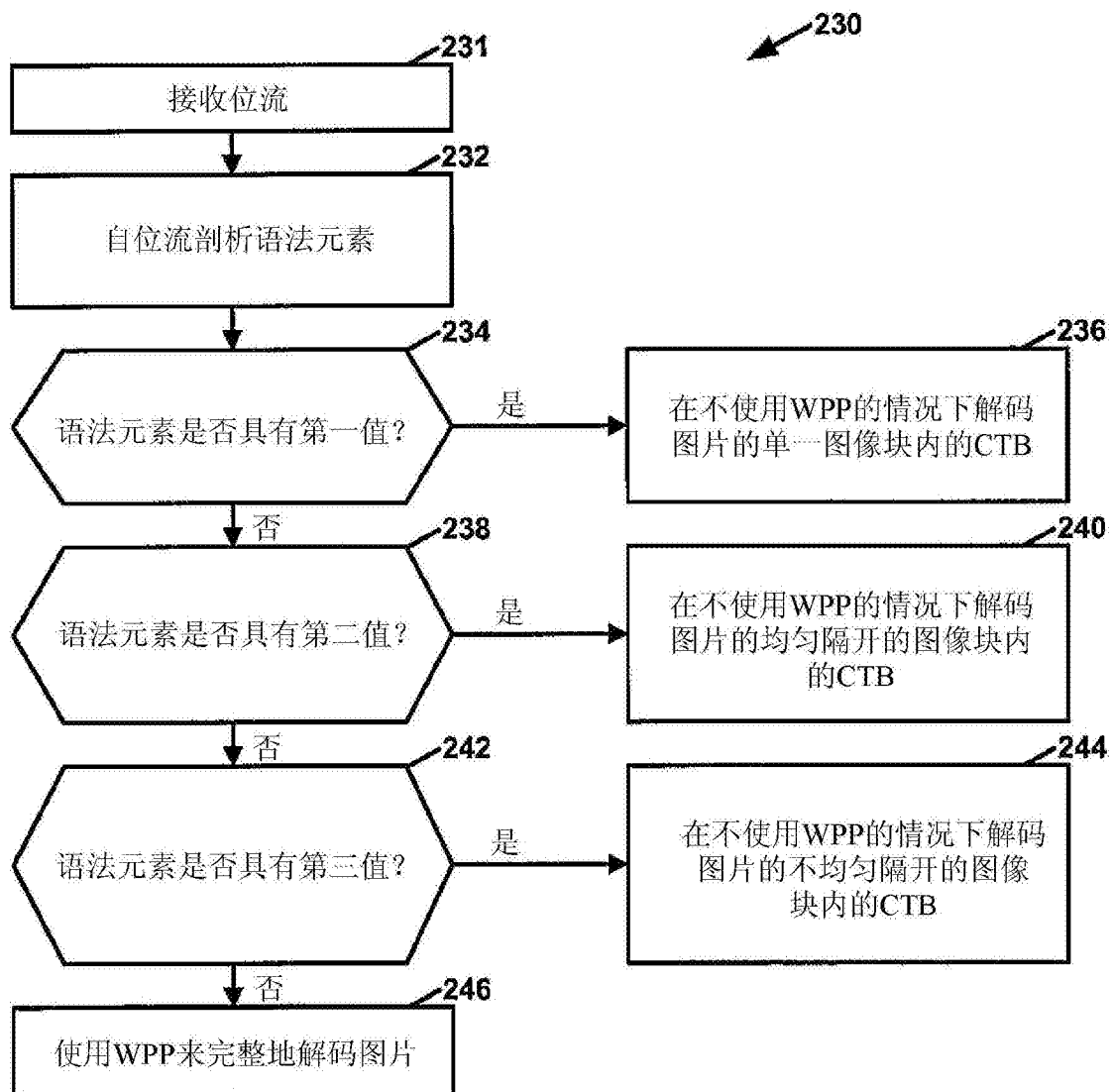


图6

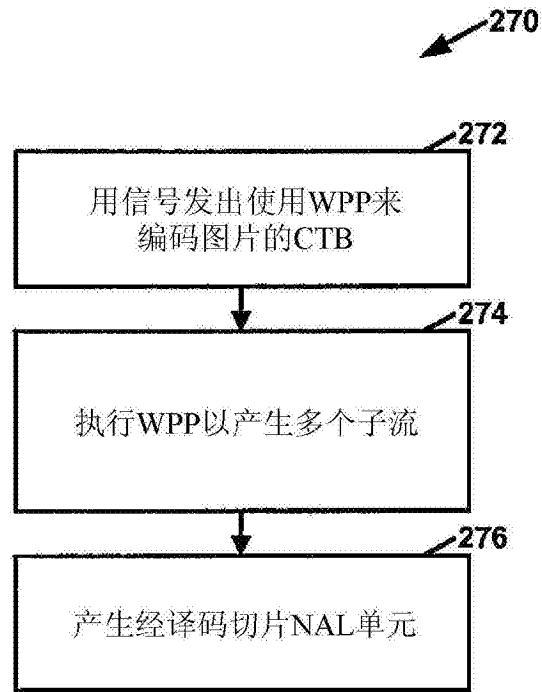


图7

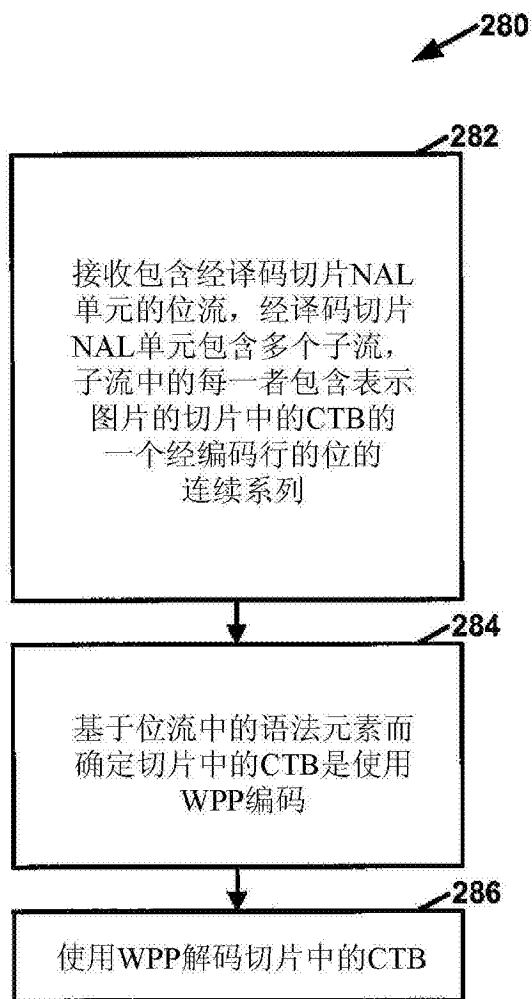


图8

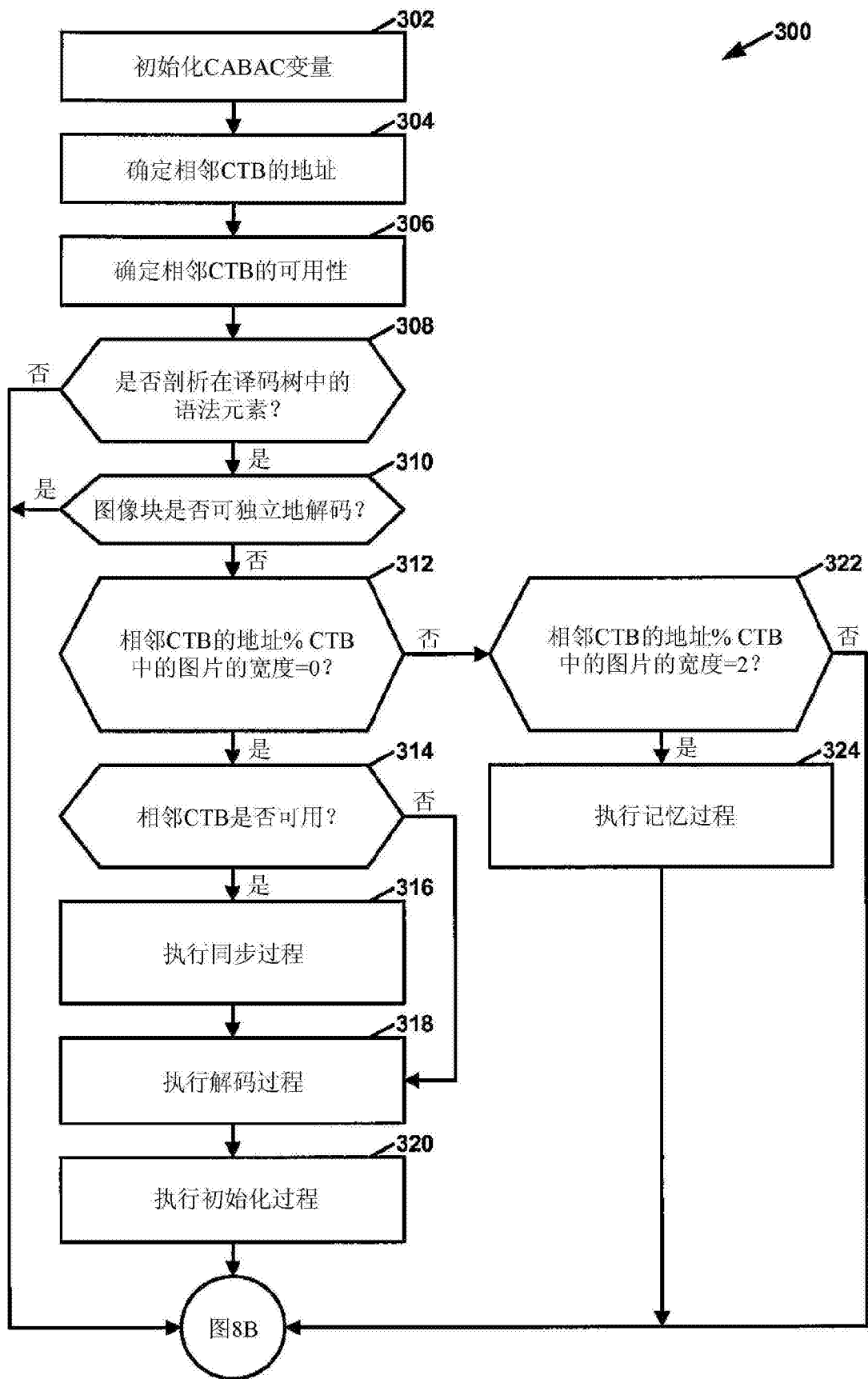


图9A

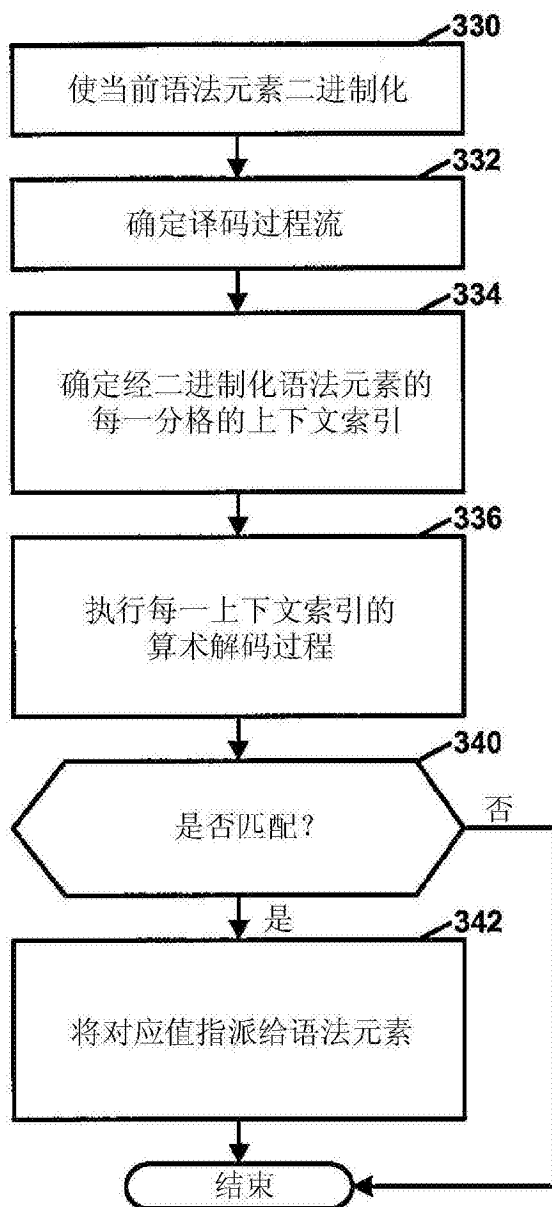


图9B

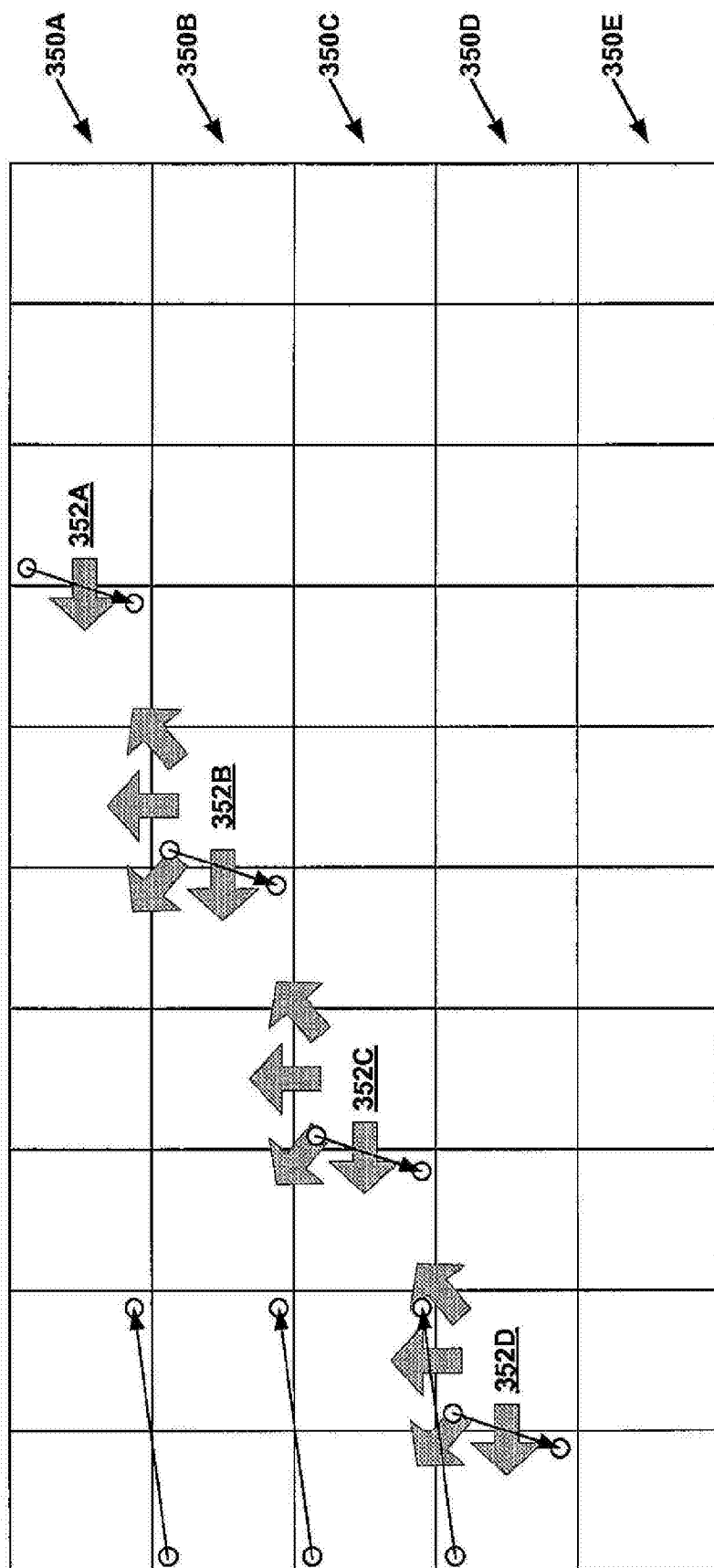


图10

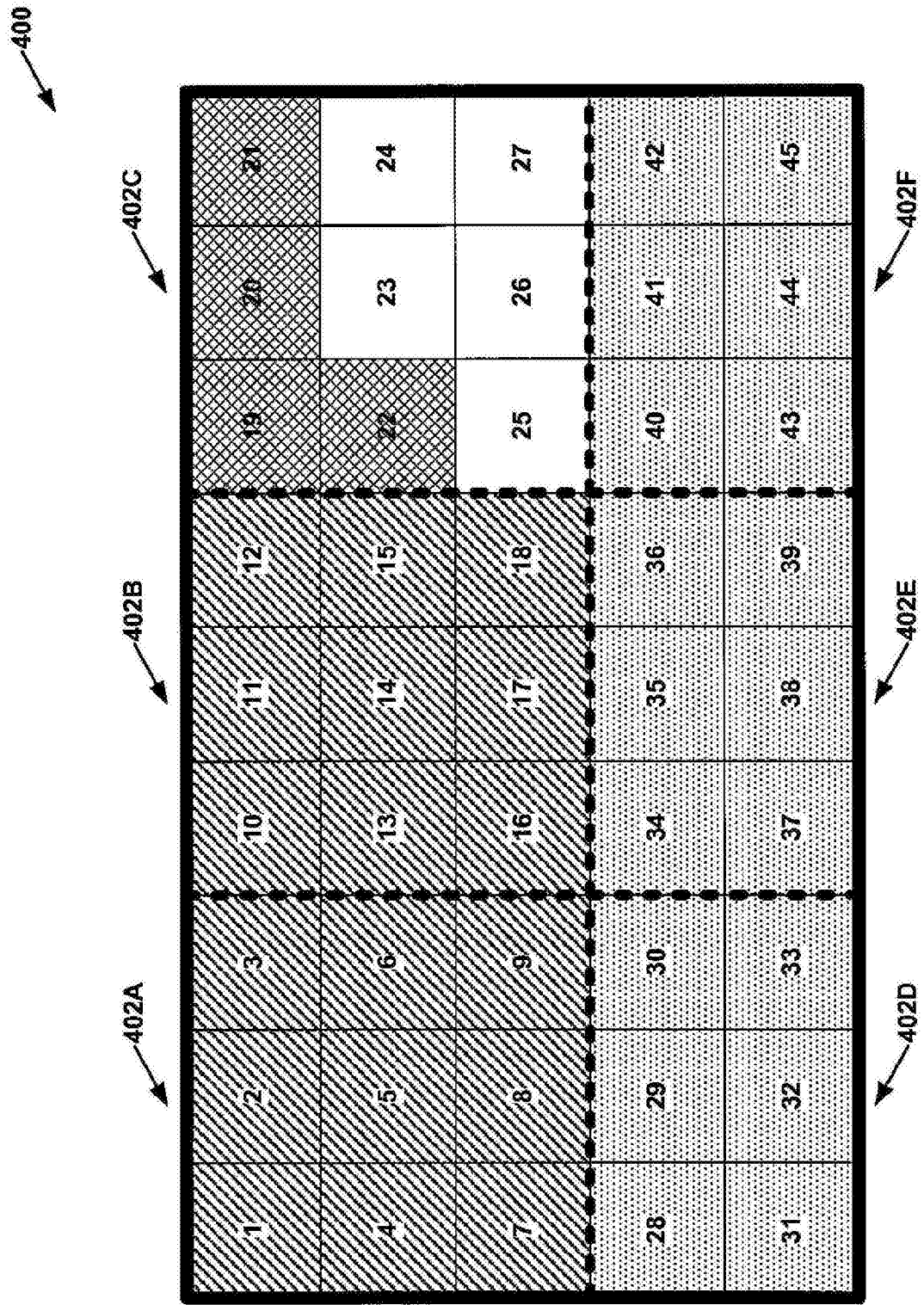


图11