



(12) 发明专利申请

(10) 申请公布号 CN 119653118 A

(43) 申请公布日 2025. 03. 18

(21) 申请号 202411900013.4

H04N 19/119 (2014.01)

(22) 申请日 2020.03.05

H04N 19/13 (2014.01)

(30) 优先权数据

H04N 19/159 (2014.01)

19305279.2 2019.03.11 EP

H04N 19/176 (2014.01)

(62) 分案原申请数据

H04N 19/46 (2014.01)

202080020745.8 2020.03.05

H04N 19/51 (2014.01)

(71) 申请人 交互数字VC控股公司

地址 美国特拉华州

(72) 发明人 陈娅 F·莱林内克 T·波伊里尔

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 赵碧洋

(51) Int. Cl.

H04N 19/91 (2014.01)

H04N 19/70 (2014.01)

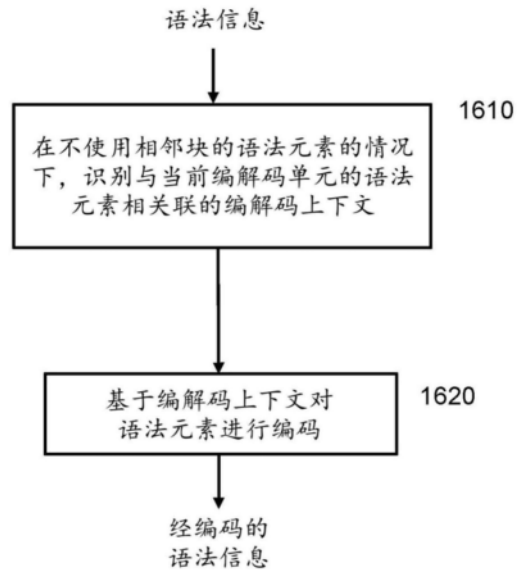
权利要求书2页 说明书21页 附图11页

(54) 发明名称

用于视频编码和解码的熵编解码

(57) 摘要

对与视频信息相关联的语法信息进行编码或解码可涉及识别与视频信息的当前编解码单元的语法元素相关联的编解码上下文,其中识别在不使用相邻块的语法元素的情况下发生,及基于编解码上下文对当前编解码单元的语法元素进行编码或解码。



1. 一种用于对与视频信息相关联的语法信息进行编码的方法,所述方法包括:
推导与所述视频信息的块的语法元素相关联的编解码上下文;以及
基于所述编解码上下文对所述块的所述语法元素进行熵编码。
2. 一种用于对与视频信息相关联的语法信息进行解码的方法,所述方法包括:
推导与所述视频信息的块的语法元素相关联的编解码上下文;以及
基于所述编解码上下文对所述块的所述语法元素进行熵解码。
3. 根据权利要求1或2所述的方法,其中,所述推导在不使用相邻块的对应语法元素的情况下发生。
4. 根据权利要求3所述的方法,其中,所述语法元素是跳过模式标志、或合并三角形预测模式、或当前图片参考模式、或自适应环路滤波器模式、或所述块的分割模式或所述块的四叉树分割模式、或仿射模式、或自适应运动矢量分辨率模式标志中的一个。
5. 根据权利要求1或2所述的方法,其中,所述推导基于所述块的左侧块和上方块的对应语法元素的总和,并且其中,所述对应语法元素的总和基于所述左侧块和所述上方块的所述对应语法元素之间的OR运算符或者基于所述左侧块和所述上方块的所述对应语法元素之间的AND运算符被推导。
6. 根据权利要求4所述的方法,其中,对于为自适应运动矢量分辨率模式标志的所述语法元素,所述推导基于确定所述块的仿射模式而在至少两个编解码上下文中选择一个编解码上下文。
7. 根据权利要求1或2所述的方法,其中,所述推导基于针对同一块尺寸的所述语法元素的不同二进制位索引共享同一上下文模型。
8. 根据权利要求7所述的方法,其中,所述推导包括确定上下文移位变量,其中:
如果所述块的尺寸小于64,则所述上下文移位变量被推导为 $(\log_2 \text{TbSize} + 1) \gg 2$,其中 $\log_2 \text{TbSize}$ 是块尺寸的二进制对数,以及 \gg 是右二进制移位,
否则,所述上下文移位变量被设置为2。
9. 根据权利要求7所述的方法,其中,对于具有至少为64的尺寸的所述块,多于2个二进制位索引共享同一上下文模型。
10. 根据权利要求7所述的方法,其中,对于尺寸至少为64的块,所述语法元素的二值化中的每3个或4个连续二进制位索引共享同一上下文模型。
11. 根据权利要求1或2所述的方法,其中,当所述语法元素是所述块的最后有效系数的坐标时,所述推导基于针对不同的块尺寸共享同一上下文模型。
12. 一种用于对与视频信息相关联的语法信息进行编码的装置,所述装置包括一个或多个处理器,所述一个或多个处理器被配置为:
推导与所述视频信息的块的语法元素相关联的编解码上下文;以及
基于所述编解码上下文对所述块的所述语法元素进行熵编码。
13. 一种用于对与视频信息相关联的语法信息进行解码的装置,所述装置包括一个或多个处理器,所述一个或多个处理器被配置为:
推导与所述视频信息的块的语法元素相关联的编解码上下文;以及
基于所述编解码上下文对所述块的所述语法元素进行熵解码。
14. 根据权利要求12或13所述的装置,其中,所述推导在不使用相邻块的对应语法元素

的情况下发生。

15. 根据权利要求14所述的装置,其中,所述语法元素是跳过模式标志、或合并三角形预测模式、或当前图片参考模式、或自适应环路滤波器模式、或所述块的分割模式或所述块的四叉树分割模式、或仿射模式、或自适应运动矢量分辨率模式标志中的一个。

16. 根据权利要求12或13所述的装置,其中,所述推导基于所述块的左侧块和上方块的对应语法元素的总和,并且其中,所述对应语法元素的总和基于所述左侧块和所述上方块的所述对应语法元素之间的OR运算符或者基于所述左侧块和所述上方块的所述对应语法元素之间的AND运算符被推导。

17. 根据权利要求16所述的装置,其中,对于为自适应运动矢量分辨率模式标志的所述语法元素,所述推导基于确定所述块的仿射模式而在至少两个编解码上下文中选择一个编解码上下文。

18. 根据权利要求12或13所述的装置,其中,所述推导基于针对同一块尺寸的所述语法元素的不同二进制位索引共享同一上下文模型。

19. 根据权利要求18所述的装置,其中,所述推导包括确定上下文移位变量,其中:

如果所述块的尺寸小于64,则所述上下文移位变量被推导为 $(\log_2 \text{TbSize} + 1) \gg 2$,其中 $\log_2 \text{TbSize}$ 是块尺寸的二进制对数,以及 \gg 是右二进制移位,

否则,所述上下文移位变量被设置为2。

20. 根据权利要求18所述的装置,其中,对于具有至少为64的尺寸所述块,多于2个二进制位索引共享同一上下文模型。

21. 根据权利要求18所述的装置,其中,对于尺寸至少为64的块,所述语法元素的二值化中的每3个或4个连续二进制位索引共享同一上下文模型。

22. 根据权利要求12或13所述的装置,其中,当所述语法元素是所述块的最后有效系数的坐标时,所述推导基于针对不同的块尺寸共享同一上下文模型。

23. 一种存储可执行程序指令的非暂时性计算机可读介质,所述可执行程序指令使得执行所述指令的计算机执行根据权利要求1或2所述的方法。

24. 一种设备,包括:

根据权利要求13所述的装置;以及

以下内容中的至少一个:(i) 天线,被配置为接收信号,所述信号包含表示视频信息的数据,(ii) 频带限制器,被配置为将所述信号限制到包含表示所述视频信息的所述数据的频带,以及(iii) 显示器,被配置为显示来自所述视频信息的图像。

25. 根据权利要求24所述的设备,其中,所述设备包括电视、电视信号接收器、机顶盒、网关设备、移动设备、蜂窝式电话、平板或其他电子设备中的一个。

用于视频编码和解码的熵编解码

[0001] 本分案申请是申请日为2020年03月05日、申请号为202080020745.8、发明名称为“用于视频编码和解码的熵编解码”的分案申请。

技术领域

[0002] 本公开涉及视频编码和解码。

背景技术

[0003] 为了实现高压缩效率,图像和视频编解码方案通常采用预测和变换来利用视频内容中的空域和时域冗余。通常,使用帧内或帧间预测来利用帧内或帧间相关性,然后对原始图片块和预测图片块之间的差(通常表示为预测误差或预测残差)进行变换、量化和熵编解码。为了重构视频,通过对应于预测、变换、量化和熵编解码的相反过程来对压缩数据进行解码。

发明内容

[0004] 通常,本公开的一个方面涉及提供对熵编解码的各种方法或修改。

[0005] 提供了实施例的至少一个示例,该实施例涉及编码器和/或解码器,该编码器和/或解码器用于基于由至少一个语法元素提供的信息来应用熵编解码的形式,并且推导用于该至少一个语法元素的上下文的数量,其中,该推导包括减少上下文的数量。

[0006] 提供了涉及用于对与视频信息相关联的语法信息进行编码的方法的实施例的至少一个其他示例,该方法包括:识别与该视频信息的当前编解码单元的语法元素相关联的编解码上下文,其中该识别在不使用相邻块的语法元素的情况下发生;以及基于编解码上下文对当前编解码单元的语法元素进行编码。

[0007] 提供了涉及用于对与视频信息相关联的语法信息进行解码的方法的实施例的至少一个其他示例,该方法包括:识别与该视频信息的当前编解码单元的语法元素相关联的编解码上下文,其中该识别在不使用相邻块的语法元素的情况下发生;以及基于编解码上下文对当前编解码单元的语法元素进行解码。

[0008] 提供了涉及用于对与视频信息相关联的语法信息进行编码的装置的实施例的至少一个其他示例,该装置包括:一个或多个处理器,其被配置为在不使用相邻块的语法元素的情况下识别与当前编解码单元的语法元素相关联的编解码上下文;以及基于编解码上下文对当前编解码单元的语法元素进行编码。

[0009] 提供了涉及用于对与视频信息相关联的语法信息进行解码的装置的实施例的至少一个其他示例,该装置包括:一个或多个处理器,其被配置为在不使用相邻块的语法元素的情况下识别与当前编解码单元的语法元素相关联的编解码上下文;以及基于编解码上下文来对当前编解码单元的语法元素进行解码。

[0010] 实施例的至少一个其他示例可以涉及以下中的一个或多个:包括自适应运动矢量分辨率(AMVR)标志的语法元素、或包括上下文自适应二进制算术编解码(CABAC)的熵编解

码形式、或使用左侧和上方相邻语法元素减少一个或多个语法元素的上下文的数量,或基于针对相同块尺寸的多个不同二进制位(二进制位)索引共享上下文或针对不同块尺寸共享上下文索引集而减少一个或多个语法元素的上下文的数量,例如,当信令通知最后有效系数的坐标时。

[0011] 如以下所解释,可预想可对视频编码和/或解码系统提供改进的各种修改和实施例,该改进和实施例包含但不限于增加的压缩或编解码效率和降低的复杂度中的一个或两个。

[0012] 为了提供对本公开的一些方面的基本理解,上面给出了主题的简要概述。本概述不是本主题的广泛概述。其并非旨在识别各实施例的关键/重要元素或描绘本主题的范围。其唯一目的是以简化形式呈现本主题的一些概念作为以下提供的更详细描述的前言。

附图说明

[0013] 通过考虑以下结合附图的详细描述可以更好地理解本公开,在附图中:

[0014] 图1提供了描绘视频编码器的实施例的示例的框图;

[0015] 图2提供了描绘视频解码器的实施例的示例的框图;

[0016] 图3图示了可以用于表示压缩图片的编解码树单元(CTU)和编解码树概念;

[0017] 图4图示了编解码树单元(CTU)以及将CTU划分为编解码单元(CU)、预测单元(PU)和变换单元(TU);

[0018] 图5图示了四叉树加二叉树(QTBT)表示的示例;

[0019] 图6图示了用于编解码单元(CU)的三叉划分模式的示例,该三叉划分模式包含水平划分模式(图6的左侧)和垂直划分模式(图6的右侧);

[0020] 图7图示了各种CU划分模式的示例;

[0021] 图8提供了图示CABAC编解码过程的示例的流程图:从语法元素到其二值化值、到其二进制位(bin)中的一些二进制位的基于上下文的算术编解码、以及到其二进制位中的一些其他二进制位的旁路编解码;

[0022] 图9图示了在例如用于inter_affine_flag的上下文模型推导的至少一个实施例中使用的相邻块的示例,

[0023] 图10至图14和图16提供了图示根据本公开的一个或多个方面的实施例的各种示例的流程图;以及

[0024] 图15提供了图示根据本文描述的各个方面和实施例的装置的实施例的示例的框图。

[0025] 应当理解,附图是为了图示各个方面和实施例的示例,而不必是唯一可能的配置。在各个附图中,相同的参考标志表示相同或相似的特征。

具体实施方式

[0026] 现转向附图,图1图示了视频编码器100(诸如,HEVC编码器)的示例。HEVC是由视频编解码联合组(JCT-VC)开发的压缩标准(参见,例如,“ITU-T H.265ITU的电信标准化部(2014年10月),H系列:视听和多媒体系统、视听服务的基础设施-移动视频的编解码,高效视频编解码,ITU-T H.265建议书”)。图1还可图示其中对HEVC标准作出改进的编码器或采

用类似于HEVC的技术的编码器,诸如基于或通过联合视频专家组(JVET)开发的,例如,与开发工作指定的通用视频编解码(VVC)相关联的JEM(联合探测模型)而改进的编码器。

[0027] 在本申请中,术语“重构的”和“解码的”可以互换使用,术语“像素”和“样点”可以互换使用,并且术语“图片”和“帧”可以互换使用。通常,但不一定,术语“重构的”在编码器侧使用,而“解码的”在解码器侧使用。

[0028] HEVC规范在“块”与“单元”之间进行区分,其中“块”寻址样点阵列中的特定区域(例如,亮度、Y),且“单元”包含所有经编码色彩分量(Y、Cb、Cr或单色)的共位块、语法元素及与块相关联的预测数据(例如,运动矢量)。

[0029] 对于编解码,图片被分割为具有可配置尺寸的正方形形状的编解码树块(CTB),并且连续的编解码树块集被分组为条带。编解码树单元(CTU)含有编码的颜色分量的CTB。CTB是分割成编解码块(CB)的四叉树的根,并且编解码块可以被分割成一个或多个预测块(PB)并且形成分割成变换块(TB)的四叉树的根。对应于编解码块、预测块和变换块,编解码单元(CU)包含预测单元(PU)和变换单元(TU)的树结构集,PU包含所有颜色分量的预测信息,并且TU包含每个颜色分量的残差编解码语法结构。亮度分量的CB、PB和TB的尺寸适用于相应的CU、PU和TU。在本申请中,术语“块”可用于指CTU、CU、PU、TU、CB、PB和TB中的任一个。另外,“块”还可用于指H.264/AVC或其他视频编解码标准中所指定的宏块和分割,且更一般地指各种尺寸的数据阵列。

[0030] 在图1的编码器100中,如下所述通过编码器元件对图片进行编码。将待编码图片以CU为单位进行处理。使用帧内或帧间模式对每个CU进行编码。当以帧内模式对CU进行编码时,其执行帧内预测(160)。在帧间模式中,执行运动估计(175)和补偿(170)。编码器决定(105)帧内模式或帧间模式中的哪一个用于对CU进行编码,并且通过预测模式标志来指示帧内/帧间决定。通过从原始图像块中减去(110)预测块来计算预测残差。

[0031] 然后对预测残差进行变换(125)和量化(130)。对量化的变换系数以及运动矢量和其他语法元素进行熵编解码(145)以输出比特流。编码器还可跳过变换并在 4×4 TU的基础上将量化直接应用于未经变换的残差信号。编码器还可绕过变换和量化两者,即,在不应用变换或量化过程的情况下直接对残差进行编解码。在直接PCM编解码中,不应用预测并且编解码单元样点被直接编解码到比特流中。

[0032] 编码器对编码块进行解码以提供用于进一步预测的参考。经量化的变换系数被去量化(140)和逆变换(150)以对预测残差进行解码。通过组合(155)解码的预测残差和预测块,重构图像块。例如,将环路滤波器(165)应用于重构图片以执行去块/SAO(样点自适应偏移)滤波以减少编码伪影。将经滤波的图像存储在参考图片缓冲器(180)中。

[0033] 图2图示了视频解码器200(诸如HEVC解码器)的示例的框图。在示例性解码器200中,信号或比特流由如下所述的解码器元件解码。视频解码器200通常执行与图1中所描述的编码通路相反的解码通路,其执行视频解码作为编码视频数据的一部分。图2还可图示其中对HEVC标准作出改进的解码器或采用类似于HEVC的技术的解码器,诸如基于或通过JEM而改进的解码器。

[0034] 具体地,解码器的输入包含可以由诸如图1的视频编码器100的视频编码器生成的视频信号或比特流。首先对信号或比特流进行熵解码(230)以获得变换系数、运动矢量和其他经编解码的信息。变换系数被去量化(240)和逆变换(250)以对预测残差进行解码。通过

组合(255)解码的预测残差和预测块,来重构图像块。可以从帧内预测(260)或运动补偿预测(即,帧间预测)(275)获得预测块(270)。高级运动矢量预测(AMVP)和合并模式技术可用于推导用于运动补偿的运动矢量,其可使用内插滤波器来计算参考块的子整数样点的内插值。将环路滤波器(265)应用于经重构的图像。将滤波后的图像存储在参考图片缓冲器(280)中。

[0035] 在HEVC视频压缩标准中,使用运动补偿时域预测来利用视频的连续图片之间存在的冗余。为此,将运动矢量与每个预测单元(PU)相关联。每个编解码树单元(CTU)由压缩域中的编解码树(CT)表示。这是CTU的四叉树分区,其中每个叶被称为编解码单元(CU),如图3所示。

[0036] 然后给每个CU一些帧内或帧间预测参数或预测信息(Prediction info)。为此,它在空域上被分割成一个或多个预测单元(PU),每个PU被分配一些预测信息。如图4所示,在CU级上分配帧内或帧间编解码模式,图4示出了将编解码树单元分区为编解码单元、预测单元和变换单元的示例。为了对CU进行编解码,从相邻重构样点构建预测块或预测单元(PU)(帧内预测)或从存储在解码图片缓冲器(DPB)中的先前重构图片构建预测块或预测单元(PU)(帧间预测)。接着,对作为原始样点和PU样点之间的差计算的残差样点进行变换和量化。

[0037] 除了HEVC之外的编解码器和视频压缩工具(例如,联合探测模型(JEM)和由通用视频编解码(VVC)参考软件(称为VVC测试模型(VTM))中的JVET(联合视频探测小组)组开发的编解码器和视频压缩工具)可提供压缩域中的CTU表示,其在压缩域中以更灵活的方式表示图片数据。与诸如HEVC标准的CU/PU/TU布置的方法相比,编解码树的更灵活表示可提供增加的压缩效率。更灵活的表示的一个示例是四叉树加二叉树(QTBT)编解码工具。在图5中图示了诸如QTBT的表示的示例,其示出了具有编解码单元的编解码树,该编解码单元可以以四叉树和二叉树的方式来划分。编解码单元的划分可以在编码器侧基于优化过程(例如率失真优化过程)来决定,该优化过程以最小的率失真成本确定CTU的QTBT表示。

[0038] 在QTBT技术中,CU可具有正方形或矩形形状。编解码单元的尺寸可以是2的幂,并且例如具有从4到128的范围。除了用于编解码单元的多种矩形形状之外,诸如QTBT的CTU的表示可以具有与诸如HEVC的方法不同的以下特征:

[0039] • CTU的QTBT分解由两个阶段组成:首先CTU被以四叉树方式划分,然后每个四叉树叶子被以二进制方式进一步划分。这在图5的右侧图示,其中实线表示四叉树分解阶段,虚线表示在空域上嵌入四叉树叶子中的二元分解。

[0040] • 在帧内条带中,亮度(Luma)和色度(Chroma)块分割结构被分离,并且独立地决定。

[0041] • 不采用分割成预测单元或变换单元的CU,即,每一CU系统地由单个预测单元($2N \times 2N$ 预测单元分割类型)和单个变换单元(不分区成变换树)组成。

[0042] 某些系统可以使用一个或多个各种其他CU划分模式。例如,VVC(通用视频编解码)视频压缩标准提供了如图6所示的水平或垂直三叉划分模式。如图6中所示,三叉划分可涉及将CU分区成三个子编解码单元(子CU, sub-CU),其各自的尺寸在所考虑的空域分区的方向上等于父CU尺寸的 $1/4$ 、 $1/2$ 和 $1/4$ 。在图7中图示了各种其他划分模式。

[0043] 在使用划分、帧内或帧间预测来利用帧内或帧间相关性之后,对原始块与预测块

之间的差(通常表示为预测误差或预测残差)进行变换、量化和熵编解码。为了重构视频,通过对应于熵编解码、量化、变换和预测的相反过程来对压缩数据进行解码。

[0044] 通常,本公开的一方面可涉及变换系数的熵编解码、帧间预测标志和分割标志中的一个或多个。在至少一个实施例中,可以降低与熵编解码相关联的至少一些语法元素的信令和解析的复杂度。例如,在至少一个实施例中,降低复杂度可涉及减少解码器过程中的操作的数量和/或用于熵编解码(诸如基于上下文的自适应二进制算术编解码(CABAC))的形式的上下文的数量。

[0045] CABAC可用于将语法元素编码到比特流中。要使用CABAC进行编码,将非二进制语法元素值映射到二进制序列,称为二进制位字符串。对于二进制位,选择上下文模型。该上下文模型存储每个二进制位是“1”或“0”的概率并且可以是自适应的或静态的。静态模型以对于二进制位是“0”和“1”是相等的概率来触发编解码引擎。在自适应编解码引擎中,基于二进制位的实际编解码值更新上下文模型。对应于自适应和静态模型的操作模式分别称为规则模式和旁路模式,如图8所示。

[0046] 对于常规模式,获得用于当前二进制位的解码的上下文。上下文由上下文建模器给出,如图8所示。上下文目标是在给定一些上下文先验信息X的情况下获得当前二进制位具有值“0”的条件概率。先验X可以是在当前二进制位被解码时以同步方式在编码器和解码器侧可用的一些已经解码的语法元素的值。

[0047] 典型地,用于二进制位解码的先验X在标准中被指定,并且被选择,因为它与要解码的当前二进制位统计相关。使用该上下文信息的好处在于其降低了编解码二进制位的率成本。这基于以下事实:因为二进制位和X是相关的,所以给定X的二进制位的条件熵较低。以下关系在信息论中是众所周知的:

[0048] $H(\text{bin}|X) < H(\text{bin})$

[0049] 这意味着,如果bin(二进制位)和X是统计相关的,则已知X的情况下bin的条件熵低于bin的熵。上下文信息X因此用于获得bin为“1”或“0”的概率。给定这些条件概率,图8的常规解码引擎执行二进制值bin的算术解码。在已知当前上下文信息X的情况下,bin的值然后被用于更新与当前bin相关联的条件概率的值,这在图8中被称为上下文模型更新步骤。只要bin被解码(或编解码),更新每个bin的上下文模型允许逐渐细化每个二进制元素的上下文建模。因此,CABAC解码器逐渐学习每个常规编码的bin的统计行为。

[0050] 注意,上下文建模器和上下文模型更新步骤是在编码器和解码器侧的相同操作。这提供了由编码器完成的二值化步骤的倒数。因此,在此执行的逆转换包括基于这些语法元素的相应的经解码的二值化版本来获得这些语法元素的值。

[0051] 多个上下文可用于信令通知一个语法元素。例如,在诸如针对VVC(例如,VTM-4)所提出的系统中,存在用于12个语法元素的387个上下文,这意味着每语法大约32个上下文。如上所述,CABAC解码器需要累积一定量的统计数据,从而逐渐学习每个常规编码的二进制位的统计行为。当存在用于一个编解码二进制位的许多上下文时,每个上下文的统计量可能不足以用于上下文模型的收敛和稳定性,这可能影响CABAC解码器的效率。

[0052] 通常,至少一个实施例可涉及减少用于常规(或基于上下文)经编解码的二进制位的CABAC上下文的数量(变换系数、帧内和帧间预测标志、自适应环路滤波器标志和分割标志等)。结果,可以降低解码过程的复杂度。例如,在至少一个实施例中,可以降低关于与最

后x和最后y坐标两者相关联的48个上下文(语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix)、帧内预测标志之一的3个上下文(语法元素pred_mode_ibc_flag)、用于一些帧间预测标志的18个上下文(语法元素cu_skip_flag,inter_affine_flag,amvr_flag和merge_triangle_flag)、用于自适应环路滤波器标志的9个上下文(语法元素alf_ctb_flag)以及用于一些分割标志的15个上下文(语法元素split_cu_flag和qt_split_cu_flag)。通常,实施例的至少一个示例可涉及基于使用左相邻语法元素及上相邻语法元素来减少某些语法元素的上下文数量来减少用于熵编解码(例如,CABAC)的上下文推导过程的复杂度。通常,实施例的至少一个示例可涉及当信令通知最后有效系数的坐标时,针对相同块尺寸的更多不同二进制位索引共享相同上下文,例如CABAC上下文,或针对不同块尺寸共享相同上下文索引集。

[0053] 更详细地,将基于诸如CABAC的示例来描述用于熵编解码的上下文推导过程的方法,其可涉及将相邻语法元素用于各种语法元素。然后,将描述用于降低这些语法元素的上下文推导的复杂度的实施例的各种示例。接下来,将针对诸如CABAC的熵编解码的示例来描述示例实施例中的最后有效系数的上下文选择的方法。然后,将描述用于降低最后有效系数坐标信令的上下文的复杂度的实施例的示例。

[0054] 在系统的一个或多个示例中,为了指示是否使用一些预测工具或模式,一个标志可以被信令通知到比特流中,然后到解码器。对于诸如CABAC的熵编解码的示例,一些标志用使用相邻语法元素推导的若干上下文来编解码。例如,可以信令通知一个标志,即inter_affine_flag,以指示是否使用基于仿射模型的运动补偿来生成当前CU的预测样点。对于基于CABAC的熵编解码方法的示例,inter_affine_flag是用3个上下文模型进行CABAC编解码的,并且上下文模型是用左块L和上面的块A的inter_affine_flag的和推导的,如图9所描绘。并且CABAC上下文ctxlnc推导过程可以表述如下:

[0055] $ctxlnc = (L \text{ 可用} \&\&L \text{ 是仿射}) + (A \text{ 可用} \&\&A \text{ 是仿射})$

[0056] 图10中所示的流程图图示了提供用于inter_affine_flag的上下文模型ctxlnc推导的实施例的示例。类似的过程也可以应用于推导用于跳过模式(cu-skip-flag)、AMVR模式(amvr_flag)、三角形预测模式(merge_triangle_flag)、当前图片参考模式(pred_mode_ibc_flag)、自适应环路滤波器(alf_ctb_flag)和分割(split_cu_flag和qt_split_cu_flag)的语法元素的CABAC上下文。

[0057] 在实施例的至少一个示例中,所描述CABAC推导过程的输入可为亮度位置(x0,y0),其指定当前亮度块相对于当前图片的左上样点的左上亮度样点,还可以是颜色分量cIdx、当前编码二叉树深度cqDepth及亮度样点中当前编解码块的宽度cbWidth和高度cbHeight。所描述的过程的输出为ctxlnc。

[0058] 其他参数或变量可以包含:

[0059] • 可以设置为等于(x0-1,y0)的左侧块的位置(xNbL,yNbL),

[0060] • 指定直接位于当前块左侧的块的可用性的availableL,

[0061] • 可以设置为等于(x0,y0-1)的上方块的位置(xNbA,yNbA),以及

[0062] • 指定直接位于当前块上方的编解码块的可用性的availableA。

[0063] 对于CABAC的示例,ctxlnc的赋值可以如下用如下表1中指定的condL和condA确定:

[0064] $ctxInc = (condL \&\&availableL) + (condA \&\&availableA) + ctxSetIdx * 3$.

| 语法元素 | condL | condA | ctxSetIdx | ctxInc 数量 |
|-----------------------------|---------------------------------|---------------------------------|------------------------|-----------|
| alf_ctb_flag[x0][y0][cldx] | alf_ctb_flag[xNbl][yNbl][cldx] | alf_ctb_flag[xNbA][yNbA][cldx] | cldx | 9 |
| split_cu_flag | cbHeight[xNbl][yNbl] < cbHeight | cbHeight[xNbA][yNbA] < cbWidth | 3 | 9 |
| qt_split_cu_flag | cqtDepth[xNbl][yNbl] > cqtDepth | cqtDepth[xNbA][yNbA] > cqtDepth | (cqtDepth < 2) ? 0 : 1 | 6 |
| cu_skip_flag[x0][y0] | cu_skip_flag[xNbl][yNbl] | cu_skip_flag[xNbA][yNbA] | 0 | 3 |
| pred_mode_ibc_flag[x0][y0] | pred_mode_ibc_flag[xNbl][yNbl] | pred_mode_ibc_flag[xNbA][yNbA] | 0 | 3 |
| amvr_flag[x0][y0] | amvr_flag[xNbl][yNbl] | amvr_flag[xNbA][yNbA] | 0 | 6 |
| merge_triangle_flag[x0][y0] | merge_triangle_flag[xNbl][yNbl] | merge_triangle_flag[xNbA][yNbA] | 0 | 3 |
| inter_affine_flag[x0][y0] | inter_affine_flag[xNbl][yNbl] | inter_affine_flag[xNbA][yNbA] | 0 | 3 |

[0066] 表1:使用左语法元素和上语法元素指定CABAC上下文

[0067] 在实施例的至少一个示例中,上述语法元素的所述上下文推导过程的复杂度可以通过基于左边和上面的语法元素仅推导两个上下文模型来降低。如上所述,用于CABAC的示例的inter_affine_flag可以用三个上下文模型来编解码,并且上下文模型用左块L和上方块A的inter_affine_flag的和来推导。

[0068] 然而,代替使用相邻块信息来生成三个上下文模型,可以仅推导两个上下文模型以减少冗余上下文。在实施例的第一示例中,可以使用左块L和上方块A的inter_affine_flag的OR值来推导上下文模型,如果左块L的条件(condL)或上方块A的条件(condA)为真,则该OR值指示语法元素的上下文将被设置为1。相应的ctxInc赋值格式被如下指定:

[0069] $ctxInc = (condL \&\&availableL) || (condA \&\&availableA) + ctxSetIdx * 2$.

[0070] 在实施例的第二示例中,可以使用左块L和上方块A的inter-affine-flag的AND值来推导上下文模型,其指示仅当条件condL和condA都为真时语法元素的上下文将被设置为1。相应的ctxInc赋值格式被如下指定:

[0071] $ctxInc = (condL \&\&availableL) \&\& (condA \&\&availableA) + ctxSetIdx * 2$.

[0072] 作为示例,基于所描述的第一示例实施例的inter_affine_flag的上下文模型ctxInc推导过程由图11中所示的流程图图示。将所描述的方法应用于这些语法元素可以减少ctxInc的数量,如表2所示,即,产生14个潜在的上下文减少:

| 语法元素 | ctxInc数量 |
|-----------------------------|----------|
| alf_ctb_flag[x0][y0][cldx] | 9→6 |
| split_cu_flag | 9→6 |
| qt_split_cu_flag | 6→4 |
| cu_skip_flag[x0][y0] | 3→2 |
| pred_mode_ibc_flag[x0][y0] | 3→2 |
| amvr_flag[x0][y0] | 6→4 |
| merge_triangle_flag[x0][y0] | 3→2 |
| inter_affine_flag[x0][y0] | 3→2 |

[0074] 表2:使用左语法元素和上语法元素减少CABAC上下文的数量

[0075] 至少一个其他实施例可提供仅使用一个上下文模型来对这些语法元素进行编解

码以避免使用相邻块。图16图示了本实施例的一个示例。在图16中,在1610处,在不使用相邻块的语法元素(例如,不使用相邻块的amvr_flag)的情况下,识别或确定与当前编解码单元的语法元素(例如,amvr_flag)相关联的编解码上下文。然后,在1620处,基于编解码上下文对语法元素进行编码。上下文推导的该实施例可以减小行缓冲器尺寸以及解析复杂度,因为在上下文推导过程中不使用左边和上面的语法元素。将所描述的实施例应用于这些语法元素可以减少如表3所示的ctxlnc的数量,即,产生28个潜在的上下文减少:

| 语法元素 | ctxlnc数量 |
|-----------------------------|----------|
| alf_ctb_flag[x0][y0][cldx] | 9→3 |
| split_cu_flag | 9→3 |
| qt_split_cu_flag | 6→2 |
| cu_skip_flag[x0][y0] | 3→1 |
| pred_mode_ibc_flag[x0][y0] | 3→1 |
| amvr_flag[x0][y0] | 6→2 |
| merge_triangle_flag[x0][y0] | 3→1 |
| inter_affine_flag[x0][y0] | 3→1 |

[0076] 表3:减少CABAC上下文的数量,而不使用左语法元素和上语法元素基于CABAC的实施例的至少一个示例可提供针对语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix的CABAC上下文的推导。系统的一个示例可包含通过明确地信令通知其(X,Y)坐标而编解码的块中的最后有效系数的位置。坐标X表示列号,且Y表示行号。坐标被二值化为前缀和后缀两部分。第一部分表示区间的索引(语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix)。该前缀具有截断的一元表示,并且二进制位以规则模式编解码。该第二部分或后缀具有固定长度表示并且以旁路模式编解码,其表示该区间内的偏移量。对于块尺寸为4、8、16、32和64,对于一个坐标的截断的一元码的最大长度(其也是规则编解码的二进制位的数量)分别是3、5、7、9和11。作为示例,表4示出了对于等于64的块宽度(高度)的二值化。最后的有效系数坐标x(y)首先被映射到11个二进制位,并且相应的二进制位以规则的方式被编解码。

[0077]

| 坐标 (x or y) | <i>last_sig_coeff_x_prefix</i> (or <i>last_sig_coeff_y_prefix</i>) |
|----------------|--|
| 0 | 0 |
| 1 | 10 |
| 2 | 110 |
| 3 | 1110 |
| 4-5 | 11110 |
| 6-7 | 111110 |
| 8-11 | 1111110 |
| 12-15 | 11111110 |
| 16-23 | 111111110 |
| 24-31 | 1111111110 |
| 32-47 | 11111111110 |
| 48-63 | 11111111111 |

[0078] 表4:等于64的块尺寸的前缀部分的最后位置二值化

[0079] 具有相似统计量的截断的一元部分内的不同二进制位共享上下文,以便减少上下文的总数量。一个坐标前缀的上下文数量为24(亮度为21,色度为3),因此最后位置编解码的上下文总数为48。表5示出了在所有块尺寸T、亮度和色度分量上对于给定坐标的不同二进制位的上下文分配。

[0080]

| 二进制位索引 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--------|----|----|----|----|----|----|----|----|----|----|----|
| T | 亮度 | | | | | | | | | | |
| 4 | 0 | 1 | 2 | | | | | | | | |
| 8 | 3 | 3 | 4 | 4 | 5 | | | | | | |
| 16 | 6 | 6 | 7 | 7 | 8 | 8 | 9 | | | | |
| 32 | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 | 14 | | |
| 64 | 15 | 15 | 16 | 16 | 17 | 17 | 18 | 18 | 19 | 19 | 20 |
| T | 色度 | | | | | | | | | | |
| 4 | 21 | 22 | 23 | | | | | | | | |
| 8 | 21 | 21 | 22 | 22 | 23 | | | | | | |
| 16 | 21 | 21 | 21 | 21 | 22 | 22 | 22 | | | | |
| 32 | 21 | 21 | 21 | 21 | 22 | 22 | 22 | 22 | 23 | | |

[0081] 表5:用于每个截断的一元代码二进制位和块尺寸T的最后位置上下文索引

[0082] 在至少一个实施例中,该过程的输入可以包含变量binIdx、颜色分量索引cIdx、变换块宽度的二进制对数log2TbWidth和变换块高度的二进制对数log2TbHeight。此过程的输出是变量ctxInc。变量log2TbSize推导如下:

[0083] -如果要解析的语法元素是last_sig_coeff_x_prefix,则将log2TbSize设置为等

于log2TbWidth。

[0084] -否则(要解析的语法元素是last_sig_coeff_y_prefix),将log2TbSize设置为等于log2TbHeight。

[0085] 变量ctxOffset和ctxShift推导如下:

[0086] -如果cldx等于0,则将ctxOffset设置为等于(log2TbSize-2)*

[0087] 3+((log2TbSize-1)>>2),且将ctxShift设置为等于(log2TbSize+1)>>2。

[0088] -否则(cldx大于0),将ctxOffset设置为等于21并且将ctxShift设置为等于Clip3(0,2,2log2TbSize>>3)。

[0089] 变量ctxInc推导如下:

[0090] $ctxInc = (binIdx \gg ctxShift) + ctxOffset$

[0091] 图12的流程图图示了提供last_sig_coeff_x_prefix和last_sig_coeff_y_prefix的上下文模型ctxInc推导的实施例的示例。

[0092] 通常,实施例的至少一个示例可提供用于推导亮度组件的语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix的CABAC上下文。如关于上述示例所提及的,具有相似统计量的截断的一元部分内的不同二进制位共享上下文以便减少上下文的总数量。一个或多个实施例可进一步减少关于亮度分量的语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix的上下文数量。

[0093] 在至少一个实施例中,用于相同块尺寸的多个不同的二进制位索引可以共享相同的上下文。可以提供变量ctxShift来决定或指示有多少二进制位索引将共享同一上下文,并且ctxShift的值与块尺寸log2TbSize相关。例如,当块宽度(高度)等于4时,每个二进制位索引将使用一个上下文;当块宽度(高度)大于4时,每2个二进制位索引将共享一个上下文。代替对于大的块尺寸(即,当块尺寸等于64时)的每2个二进制位索引共享一个上下文,每3个或4个二进制位索引可以共享一个上下文。表6示出了针对块尺寸64、亮度分量的修改的上下文分配,其中每四个二进制位索引共享相同的上下文。可通过所描述的实施例减少用于语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix两者的六个上下文。

| 二进制位索引 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|----|----|----|----|----|----|----|----|----|----|----|
| [0094] T | 亮度 | | | | | | | | | | |
| 64 | 15 | 15 | 15 | 15 | 16 | 16 | 16 | 16 | 17 | 17 | 17 |

[0095] 表6:用于块尺寸为64的每个截断的一元码二进制位的最后位置上下文索引

[0096] 并且对于所描述的实施例的变量ctxShift的相应修改可以推导如下:

[0097]
$$ctxshift = \begin{cases} (\log_2 TbSize + 1) \gg 2, & \text{当 } \log_2 TbSize < 64 \\ 2, & \text{其他} \end{cases}$$

[0098] 图13中所示的流程图示出了诸如上文所描述的实施例的示例,其提供了用于last_sig_coeff_x_prefix和last_sig_coeff_y_prefix的上下文模型ctxInc推导,其中对于块尺寸64的更多二进制位索引共享相同的上下文。

[0099] 在诸如上面描述和图13所图示的实施例的至少一个变量中,在每四个二进制位索引之间共享相同上下文也可应用于其他块尺寸。在至少一个其他变体中,在每个x(x>2)二进制位索引之间共享相同的上下文也可以应用于其他块尺寸。在至少一个其他变体中,在

每个 $x(x>2)$ 二进制位索引之间共享相同的上下文也可以应用于色度分量的其他块尺寸。

[0100] 在实施例的至少一个示例中,不同的块尺寸可以共享相同的上下文集。例如,在至少一个系统中,对于亮度分量,3、3、4、5和6个上下文可以分别用于4、8、16、32和64的块尺寸。并且变量ctxOffset可以与块尺寸 $\log_2\text{TbSize}$ 相关($\text{ctxOffset} = [0, 3, 6, 10, 15, 21]$),其指示针对每个块尺寸的上下文集。在本实施例的示例中,可以针对跨越不同块尺寸的不同上下文集共享相同的上下文,而不是将不同的上下文集分配给不同的块尺寸。表7示出了使用相同变量ctxOffset值的用于块尺寸4和8、亮度分量的修改的上下文集分配。并且对于该示例,对应的修改变量ctxOffset值集是 $\text{ctxOffset} = [0, 0, 3, 7, 12, 18]$ 。可以基于所描述的示例的示例来减少语法元素last_sig_coeff_x_prefix和last_sig_coeff_y_prefix两者的六个上下文。

| 二进制索引 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|----|----|----|----|----|----|----|----|----|----|----|
| T | 亮度 | | | | | | | | | | |
| [0101] 4 | 0 | 1 | 2 | | | | | | | | |
| 8 | 0 | 0 | 1 | 1 | 2 | | | | | | |
| 16 | 3 | 3 | 4 | 4 | 5 | 5 | 6 | | | | |
| 32 | 7 | 7 | 8 | 8 | 9 | 9 | 10 | 10 | 11 | | |
| 64 | 12 | 12 | 13 | 13 | 14 | 14 | 15 | 15 | 16 | 16 | 17 |

[0102] 表7:块尺寸4和8共享的最后位置上下文索引集

[0103] 图14中所示的流程图图示了诸如所描述的示例的示例,该示例提供了用于last_sig_coeff_x_prefix和last_sig_coeff_y_prefix的上下文模型ctxInc推导,其中针对块尺寸4和8共享相同上下文集。

[0104] 诸如上面描述以及图14所示的示例的至少一个变体可以涉及在所有块尺寸之间共享上下文索引。至少一个其他变体可涉及在不同块尺寸的各种组合中的任一者之间共享上下文索引。至少一个其他变体可涉及在亮度和色度分量之间共享上下文索引。

[0105] 根据本文描述的一个或多个实施例的涉及视频编码和/或解码的系统可以单独地或以各种布置组合地提供以下非限制性特征示例中的一个或多个:

[0106] • 对于通过左侧和上方语法元素之和推导的各种语法元素的上下文索引可以通过这两个相邻元素的OR值推导;

[0107] • 对于通过左侧和上方语法元素之和推导的各种语法元素的上下文索引可以通过这两个相邻元素的AND值来推导;

[0108] • 对于通过左侧和上方语法元素之和推导的各种语法元素的上下文索引可以通过不使用相邻元素来推导;

[0109] • 用于信令通知最后有效系数的坐标的上下文索引可以对于相同块尺寸的更多不同二进制索引共享相同的上下文;以及

[0110] • 用于信令通知最后有效系数的坐标的上下文索引可以对于不同块尺寸共享相同上下文集。

[0111] 本文件描述了实施例、特征、模型、方法等的各种示例。许多这样的示例是以特异

性描述的,并且至少为了显示个体特征,通常以可能出现限制的方式描述。然而,这是为描述清楚的目的,并且不限制应用或范围。实际上,本文所述的实施例、特征等的各种示例可以各种方式组合和互换以提供实施例的进一步示例。

[0112] 通常,在本文件中描述和设想的实施例的示例可以以许多不同的形式实现。上面描述的图1和图2以及下面描述的图10提供了一些实施例,但是可以设想其他实施例,并且图1、2和15的讨论不限制实现的宽度。至少一个实施例通常提供了涉及视频编码和/或解码的示例,且至少一个其他实施例通常涉及传输所产生或编码的比特流或信号。这些和其他实施例可被实现为方法、装置、其上存储有用于根据所描述的方法中的任一种来编码或解码视频数据的指令的计算机可读存储介质,和/或其上存储有根据所描述的方法中的任一种生成的比特流或信号的计算机可读存储介质。

[0113] 在本申请中,术语“重构的”和“解码的”可以互换使用,术语“像素”和“样点”可以互换使用,术语“图像”、“图片”和“帧”可以互换使用。通常,但不一定,术语“重构的”在编码器侧使用,而“解码的”在解码器侧使用。

[0114] 在本公开中使用术语HDR (高动态范围) 和SDR (标准动态范围)。这些术语通常向本领域的普通技术人员传达动态范围的特定值。然而,还期望另外的实施例,其中对HDR的引用被理解为表示“较高动态范围”,而对SDR的引用被理解为表示“较低动态范围”。这样的附加实施例不受可能经常与术语“高动态范围”和“标准动态范围”相关联的动态范围的任何特定值的约束。

[0115] 本文描述了各种方法,并且每个方法包括用于实现所描述的方法的一个或多个步骤或动作。除非对于方法的适当操作需要步骤或动作的特定顺序,否则可以修改或组合特定步骤和/或动作的顺序和/或使用。

[0116] 本文件中描述的各种方法和其他方面可用于修改视频编码器和/或解码器的模块,诸如图1中所示的编码器100的模块145和图2中所示的解码器200的模块230。此外,本发明的方面不限于VVC或HEVC,且可应用于(例如)现有的或未来开发的其他标准和建议,以及任何此类标准和建议(包含VVC和HEVC)的扩展。除非另外指出或在技术上排除,否则本文所描述的方面可单独或组合使用。

[0117] 例如,在本文件中使用的各种数值。这些具体值是出于示例的目的,并且所描述的方面不限于这些具体值。

[0118] 图15图示了其中可以实现各个方面和实施例的系统的示例的框图。系统1000可以被实现为包含下面描述的各种组件的设备,并且被配置为执行本文件中描述的一个或多个方面。这种设备的示例包含但不限于各种电子设备,诸如个人计算机、膝上型计算机、智能电话、平板计算机、数字多媒体机顶盒、数字电视接收机、个人视频记录系统、连接的家用电器和服务器。系统1000的元件可以单独地或组合地在单个集成电路、多个IC和/或分立组件中实现。例如,在至少一个实施例中,系统1000的处理和编码器/解码器元件分布在多个IC和/或分立组件上。在各种实施例中,系统1000经由例如通信总线或通过专用输入和/或输出端口可通信地耦合到其他类似系统或其他电子设备。在各种实施例中,系统1000被配置为实现本文所描述的方面中的一个或多个。

[0119] 系统1000包含至少一个处理器1010,其被配置为执行加载在其中的指令,以实现例如本文件中描述的各个方面。处理器1010可以包含嵌入式存储器、输入输出接口和本领

域已知的各种其他电路。系统1000包含至少一个存储器1020(例如,易失性存储器设备和/或非易失性存储器设备)。系统1000包含存储设备1040,该存储设备可以包含非易失性存储器 and/或易失性存储器,包含但不限于EEPROM、ROM、PROM、RAM、DRAM、SRAM、闪存、磁盘驱动器和/或光盘驱动器。作为非限制性示例,存储设备1040可以包含内部存储设备、附加存储设备和/或网络可访问存储设备。

[0120] 系统1000包含编码器/解码器模块1030,其被配置为(例如)处理数据以提供经编码视频或经解码视频,且编码器/解码器模块1030可包含其自身的处理器和存储器。编码器/解码器模块1030表示可包含在设备中以执行编码和/或解码功能的模块。如已知的,设备可以包含编码和解码模块中的一个或两个。另外,编码器/解码器模块1030可实现为系统1000的单独元件或可并入处理器1010内作为如所属领域的技术人员已知的硬件与软件的组合。

[0121] 待加载到处理器1010或编码器/解码器1030上以执行本文所描述的各个方面的程序代码可存储在存储设备1040中且随后加载到存储器1020上以由处理器1010执行。根据各种实施例,处理器1010、存储器1020、存储设备1040和编码器/解码器模块1030中的一个或多个可在执行本文所描述的过程期间存储各种项中的一个或多个。这样的存储项目可以包含但不限于输入视频、解码视频或解码视频的部分、比特流或信号、矩阵、变量以及来自等式、公式、操作和操作逻辑的处理的中间或最终结果。

[0122] 在若干实施例中,处理器1010和/或编码器/解码器模块1030内的存储器用于存储指令并提供用于在编码或解码期间所需的处理的工作存储器。然而,在其他实施例中,处理设备外部的存储器(例如,处理设备可以是处理器1010或编码器/解码器模块1030)用于这些功能中的一个或多个。外部存储器可以是存储器1020和/或存储设备1040,例如动态易失性存储器和/或非易失性闪存。在几个实施例中,使用外部非易失性闪存来存储电视的操作系统。在至少一个实施例中,诸如RAM的快速外部动态易失性存储器被用于诸如MPEG-2、HEVC或VVC(通用视频编解码)的视频编码和解码操作的工作存储器。

[0123] 可以通过各种输入设备来提供对系统1000的元件的输入,如框1130所示。这样的输入设备包含但不限于(i)RF部分,其接收例如由广播器通过空中传输的RF信号、(ii)复合输入端子、(iii)USB输入端子,和/或(iv)HDMI输入端子。

[0124] 在各种实施例中,框1130的输入设备具有相关的相应输入处理元件,如本领域已知的。例如,RF部分可以与用于以下内容的元件相关联:(i)选择期望频率(也称为选择信号,或将信号频带限制到频带),(ii)下变频选择的信号,(iii)再次频带限制到较窄频带以选择(例如)在某些实施例中可以称为信道的信号频带,(iv)解调下变频和频带限制的信号,(v)执行纠错,和(vi)解复用以选择期望的数据分组流。各种实施例的RF部分包含一个或多个元件以执行这些功能,例如,频率选择器、信号选择器、限带器、信道选择器、滤波器、下变频器、解调器、纠错器和解复用器。RF部分可以包含执行各种这些功能的调谐器,这些功能包含例如将接收信号下变频到较低频率(例如,中频或近基带频率)或基带。在一个机顶盒实施例中,RF部分及其相关联的输入处理元件接收通过有线(例如,电缆)介质发送的RF信号,并通过滤波、下变频和再次滤波到期望的频带来执行频率选择。各种实施例重新安排上述(和其他)元件的顺序、移除这些元件中的一些,和/或添加执行类似或不同功能的其他元件。添加元件可以包含在现有元件之间插入元件,例如插入放大器和模数转换器。在各

种实施例中,RF部分包含天线。

[0125] 另外,USB和/或HDMI端子可以包含用于通过USB和/或HDMI连接将系统1000连接到其他电子设备的相应接口处理器。应当理解,输入处理的各个方面,例如,里德-所罗门纠错,可以在例如单独的输入处理IC或处理器1010内实现。类似地,USB或HDMI接口处理的各方面可在单独的接口IC内或在处理器1010内实现。解调、纠错和解复用的流被提供给各种处理元件,包含例如处理器1010和编码器/解码器1030,其与存储器和存储元件结合操作以处理数据流以便在输出设备上呈现。

[0126] 系统1000的各种元件可设置在集成壳体内,各种元件可使用合适的连接布置1140(例如本领域已知的内部总线(包含I2C总线、布线和印刷电路板))互连并在其间传输数据。

[0127] 系统1000包含允许经由通信信道1060与其他设备通信的通信接口1050。通信接口1050可以包含但不限于被配置为在通信信道1060上发送和接收数据的收发器。通信接口1050可以包含但不限于调制解调器或网卡,并且通信信道1060可以例如在有线和/或无线介质内实现。

[0128] 在各种实施例中,使用诸如IEEE 802.11的Wi-Fi网络将数据流传输到系统1000。这些实施例的Wi-Fi信号通过适用于Wi-Fi通信的通信信道1060和通信接口1050来接收。这些实施例的通信信道1060典型地连接到访问点或路由器,该访问点或路由器提供到包含因特网的外部网络的访问,以允许串流应用和其他顶层通信。其他实施例使用通过输入块1130的HDMI连接传递数据的机顶盒向系统1000提供流数据。还有其他实施例使用输入块1130的RF连接向系统1000提供流数据。

[0129] 系统1000可以将输出信号提供给各种输出设备,包含显示器1100、扬声器1110和其他外围设备1120。在实施例的各种示例中,其他外围设备1120包含独立DVR、盘播放器、立体声系统、照明系统和基于系统1000的输出提供功能的其他设备中的一个或多个。在各种实施例中,使用诸如AV.Link、CEC的信令或在有或没有用户干预的情况下实现设备到设备控制的其他通信协议,在系统1000和显示器1100、扬声器1110或其他外围设备1120之间传送控制信号。这些输出设备可以经由专用连接通过相应接口1070、1080和1090通信地耦合到系统1000。可替代地,输出设备可经由通信接口1050使用通信信道1060连接到系统1000。显示器1100和扬声器1110可与电子设备(例如,电视)中的系统1000的其他组件集成在单个单元中。在各种实施例中,显示接口1070包含显示驱动器,例如定时控制器(T Con)芯片。

[0130] 例如,如果输入1130的RF部分是单独的机顶盒的一部分,则显示器1100和扬声器1110可以可替代地与一个或多个其他组件分离。在显示器1100和扬声器1110是外部组件的各种实施例中,可以经由专用输出连接(包含例如HDMI端口、USB端口或COMP输出)来提供输出信号。

[0131] 这些实施例可以通过由处理器1010实现的计算机软件或者通过硬件或者通过硬件和软件的组合来实现。作为非限制性示例,实施例可以通过一个或多个集成电路来实现。存储器1020可以是适合于技术环境的任何类型,并且可以使用任何适当的数据存储技术来实现,作为非限制性示例,诸如光学存储器设备、磁存储器设备、基于半导体的存储器设备、固定存储器和可移动存储器。作为非限制性示例,处理器1010可以是适合于技术环境的任何类型,并且可以包含微处理器、通用计算机、专用计算机和基于多核架构的处理器中的一个或多个。

[0132] 在整个本公开的内容中,各种实现方式涉及解码。本申请中使用的“解码”可以包含例如对所接收的编码序列执行的全部或部分处理,以便产生适于显示的最终输出。在各种实施例中,这样的处理包含典型地由解码器执行的一个或多个处理,例如熵解码、逆量化、逆变换和差分解码。在各种实施例中,此类过程还或可替代地包含由本申请案中所描述的各种实施方案的解码器执行的过程,例如,从平铺(titled)(封装)图片提取图片、确定上采样滤波器以使用且对图片进行上采样,以及将图片翻转回其既定定向。

[0133] 作为进一步的示例,在一个实施例中,“解码”仅指熵解码,在另一个实施例中,“解码”仅指差分解码,而在另一个实施例中,“解码”指熵解码和差分解码的组合。基于具体描述的上下文,短语“解码过程”是否旨在具体指操作的子集或一般地指更广泛的解码过程将是清楚的,并且被认为是本领域技术人员充分理解的。

[0134] 此外,各种实现涉及编码。以与以上关于“解码”的讨论类似的方式,如在本申请中使用的“编码”可以包括例如对输入视频序列执行的处理的全部或部分,以便产生编码的比特流或信号。在各种实施例中,这样的处理包含典型地由编码器执行的一个或多个处理,例如分割、差分编码、变换、量化和熵编解码。在各种实施例中,此类过程还或可替代地包含由本申请中描述的各种实施方案的编码器执行的过程。

[0135] 作为进一步的示例,在一个实施例中,“编码”仅指熵编码,在另一个实施例中“编码”仅指差分编码,而在另一个实施例中“编码”指差分编码和熵编码的组合。基于具体描述的上下文,短语“编码过程”是否旨在具体指操作的子集或一般地指更广泛的编码过程将是清楚的,并且被认为是本领域技术人员充分理解的。

[0136] 注意,本文使用的语法元素是描述性术语。同样,它们并不排除使用其他语法元素名称。

[0137] 当图被呈现为流程图时,应当理解,它还提供了对应装置的框图。类似地,当将图表示为框图时,应当理解,它还提供了对应的方法/过程的流程图。

[0138] 各种实施例涉及率失真优化。特别地,在编码过程期间,通常考虑速率和失真之间的平衡或折衷,通常给出计算复杂度的约束。率失真优化通常被公式化为最小化率失真函数,该率失真函数是速率和失真的加权和。存在不同的方法来解决率失真优化问题。例如,这些方法可以基于对所有编码选项的广泛测试,包含所有考虑的模式或编解码参数值,在编解码和解码之后对它们的编解码成本和重构信号的相关失真进行完整评估。还可以使用更快的方法来节省编码复杂度,特别是基于预测或预测残差信号而不是重构信号来计算近似失真。还可以使用这两种方法的混合,诸如通过仅对一些可能的编码选项使用近似失真,而对其他编码选项使用完全失真。其他方法仅评估可能的编码选项的子集。更通常地,许多方法采用各种技术中的任一者来执行优化,但优化未必是对编解码成本和相关失真两者的完全评估。

[0139] 本文描述的实现方式和方面可以在例如方法或过程、装置、软件程序、数据流或信号中实现。即使仅在单一形式的实现的上下文中讨论(例如,仅作为方法讨论),所讨论的特征的实现也可以以其他形式(例如,装置或程序)来实现。装置可以用例如适当的硬件、软件和固件来实现。方法可以在例如处理器中实现,该处理器通常指处理设备,包含例如计算机、微处理器、集成电路或可编程逻辑设备。处理器还包含通信设备,诸如计算机、蜂窝电话、便携式/个人数字助理(“PDA”),以及便于终端用户之间的信息通信的其他设备。

[0140] 对“一个实施例”或“实施例”或“一种实现方式”或“实现方式”以及其他变型的引用意味着结合该实施例描述的特定特征、结构、特性等包含在至少一个实施例中。因此，出现在本文件各处的短语“在一个实施例中”或“在实施例中”或“在一种实现方式中”或“在实现方式中”以及任何其他变化的出现不必都指同一实施例。

[0141] 另外，本文件可以指“获得”各种信息。获得信息可以包含例如确定信息、估计信息、计算信息、预测信息或从存储器中检索信息中的一个或多个。

[0142] 此外，本文件可以指“访问”各种信息。访问信息可以包含例如接收信息、检索信息（例如，从存储器）、存储信息、移动信息、复制信息、计算信息、确定信息、预测信息或估计信息中的一个或多个。

[0143] 另外，本文件可以指“接收”各种信息。如同“访问”一样，接收是广义的术语。接收信息可包含例如访问信息或检索信息（例如从存储器）中的一个或多个。此外，在诸如存储信息、处理信息、发送信息、移动信息、复制信息、擦除信息、计算信息、确定信息、预测信息或估计信息的操作期间，典型地以一种方式或另一种方式涉及“接收”。

[0144] 应当理解，例如，在“A/B”、“A和/或B”和“A和B中的至少一个”的情况下，使用以下“/”、“和/或”和“中的至少一个”中的任何一个旨在包含仅选择第一个列出的选项（A），或仅选择第二个列出的选项（B），或选择两个选项（A和B）。作为进一步示例，在“A、B和/或C”和“A、B和C中的至少一个”的情况下，这样的短语旨在包含仅选择第一个列出的选项（A），或仅选择第二个列出的选项（B），或仅选择第三个列出的选项（C），或仅选择第一个和第二个列出的选项（A和B），或仅选择第一个和第三个列出的选项（A和C），或仅选择第二个和第三个列出的选项（B和C），或选择所有三个选项（A和B和C）。如本领域和相关领域的普通技术人员所清楚的，这可以扩展到所列出的尽可能多的项目。

[0145] 此外，如本文中所使用，词语“信号”尤其是指向对应的解码器指示某物。例如，在某些实施例中，编码器用信号发送多个用于细化的参数中的特定一个。以这种方式，在实施例中，在编码器侧和解码器侧都使用相同的参数。因此，例如，编码器可以将特定参数发送（显式用信号发送）到解码器，使得解码器可以使用相同的特定参数。相反，如果解码器已经具有特定参数以及其他参数，则可以使用用信号发送而无需发送（隐式用信号发送）来简单地允许解码器知道并选择特定参数。通过避免传输任何实际函数，在各种实施例中实现了比特节省。应当理解，可以以各种方式实现用信号发送。例如，在各种实施例中，一个或多个语法元素、标志等用于将信息用信号发送到对应的解码器。虽然前面涉及词语“信号”的动词形式，但是词语“信号”在本文中也可以用作名词。

[0146] 对于本领域普通技术人员显而易见的是，实现方式可以产生被格式化为携带可以例如被存储或发送的信息的各种信号。该信息可以包括例如用于执行方法的指令或由所描述的实现方式之一产生的数据。例如，信号可以被格式化以携带所描述的实施例的比特流或信号。这样的信号可以被格式化为例如电磁波（例如，使用频谱的射频部分）或基带信号。格式化可以包括例如对数据流进行编码和用编码的数据流调制载波。信号携带的信息可以是例如模拟或数字信息。如已知的，信号可以通过各种不同的有线或无线链路发送。信号可以存储在处理器可读介质上。

[0147] 在整个本公开中还支持和考虑了各种概括的以及具体的实施例。根据本公开的实施例的示例包含但不限于以下。

[0148] 通常, 实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行编码的方法, 该方法包括: 识别与该视频信息的当前编解码单元的语法元素相关联的编解码上下文, 其中该识别在不使用相邻块的语法元素的情况下发生; 以及基于编解码上下文对当前编解码单元的语法元素进行编码。

[0149] 通常, 实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行解码的方法, 该方法包括: 识别与该视频信息的当前编解码单元的语法元素相关联的编解码上下文, 其中该识别在不使用相邻块的语法元素的情况下发生; 以及基于编解码上下文对当前编解码单元的语法元素进行解码。

[0150] 通常, 实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行编码的装置, 该装置包括: 一个或多个处理器, 其被配置为在不使用相邻块的语法元素的情况下识别与当前编解码单元的语法元素相关联的编解码上下文; 以及基于编解码上下文对当前编解码单元的语法元素进行编码。

[0151] 通常, 实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行解码的装置, 该装置包括: 一个或多个处理器, 其被配置为在不使用相邻块的语法元素的情况下识别与当前编解码单元的语法元素相关联的编解码上下文; 以及基于编解码上下文来对当前编解码单元的语法元素进行解码。

[0152] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中当前编解码单元的语法元素指示视频编码模式或视频解码模式的使用且包括跳过模式标志、或自适应运动矢量分辨率模式标志、或三角形预测模式、或当前图片参考模式、或自适应环路滤波器模式、或分割模式、或仿射模式中的一个或多个。

[0153] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中当前编解码单元的语法元素指示使用自适应运动矢量分辨率模式来对当前编解码单元进行编码或解码, 且相邻块的语法元素指示用于对相邻块进行编码或解码的自适应运动矢量分辨率模式。

[0154] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中, 编码和/或解码包括熵编码和/或熵解码。

[0155] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中熵编码和/或熵解码包括CABAC。

[0156] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中当前编解码单元的相邻块包括当前编解码单元左侧的相邻块或当前编解码单元上方的相邻块中的至少一个。

[0157] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中当前编解码单元的语法元素包括自适应运动矢量分辨率 (AMVR) 语法元素, 且识别、或一个或多个处理器被配置为识别与当前编解码单元的语法元素相关联的编解码上下文是基于确定当前编解码单元的仿射模式。

[0158] 通常, 实施例的至少一个示例可涉及根据本文所描述的方法或装置, 其中确定当前编解码单元的仿射模式是基于帧间仿射标志。

[0159] 通常, 实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行编码的方法, 该方法包括: 识别与该视频信息的当前编解码单元的语法元素相关联的编解码

上下文,其中该识别避免使用相邻块的语法元素;以及基于编解码上下文对当前编解码单元的语法元素进行编码。

[0160] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行解码的方法,该方法包括:识别与该视频信息的当前编解码单元的语法元素相关联的编解码上下文,其中该识别避免使用相邻块的语法元素;以及基于编解码上下文对当前编解码单元的语法元素进行解码。

[0161] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行编码的装置,该装置包括:一个或多个处理器,其被配置为基于避免相邻块的语法元素的使用来识别与当前编解码单元的语法元素相关联的编解码上下文;以及基于编解码上下文对当前编解码单元的语法元素进行编码。

[0162] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行解码的装置,该装置包括:一个或多个处理器,其被配置为基于避免相邻块的语法元素的使用来识别与当前编解码单元的语法元素相关联的编解码上下文;以及基于编解码上下文来对当前编解码单元的语法元素进行解码。

[0163] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中当前编解码单元的语法元素指示视频编码模式或视频解码模式的使用且包括跳过模式标志、或自适应运动矢量分辨率模式标志、或三角形预测模式、或当前图片参考模式、或自适应环路滤波器模式、或分割模式、或仿射模式中的一个或多个。

[0164] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中当前编解码单元的语法元素指示使用自适应运动矢量分辨率模式来对当前编解码单元进行编码或解码,且相邻块的语法元素指示自适应运动矢量分辨率模式,以对相邻块进行编码或解码。

[0165] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中,编码和/或解码包括熵编码和/或熵解码。

[0166] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中熵编码和/或熵解码包括CABAC。

[0167] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中当前编解码单元的相邻块包括当前编解码单元左侧的相邻块或当前编解码单元上方的相邻块中的至少一个。

[0168] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中当前编解码单元的语法元素包括自适应运动矢量分辨率(AMVR)语法元素,且识别、或一个或多个处理器被配置为识别与当前编解码单元的语法元素相关联的编解码上下文是基于确定当前编解码单元的仿射模式。

[0169] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中确定当前编解码单元的仿射模式是基于帧间仿射标志。

[0170] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行编码的方法,该方法包括:在不使用相邻块的AMVR语法元素的情况下识别与视频信息的当前编解码单元的自适应运动矢量分辨率AMVR语法元素相关联的编解码上下文;以及基于编解码上下文对当前编解码单元的AMVR语法元素进行编码。

[0171] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行解码的方法,该方法包括:在不使用相邻块的AMVR语法元素的情况下识别与视频信息的当前编解码单元的AMVR语法元素相关联的编解码上下文;以及基于编解码上下文对当前编解码单元的AMVR语法元素进行解码。

[0172] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行编码的装置,该装置包括:一个或多个处理器,其被配置为在不使用相邻块的AMVR语法元素的情况下识别与当前编解码单元的AMVR语法元素相关联的编解码上下文;以及基于编解码上下文对当前编解码单元的AMVR语法元素进行编码。

[0173] 通常,实施例的至少一个示例可涉及用于对与视频信息相关联的语法信息进行解码的装置,该装置包括:一个或多个处理器,其被配置为在不使用相邻块的AMVR语法元素的情况下识别与当前编解码单元的语法元素相关联的编解码上下文;以及基于编解码上下文来对当前编解码单元的AMVR语法元素进行解码。

[0174] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中,编码和/或解码包括熵编码和/或熵解码。

[0175] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中熵编码和/或熵解码包括CABAC。

[0176] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中当前编解码单元的相邻块包括当前编解码单元左侧的相邻块或当前编解码单元上方的相邻块中的至少一个。

[0177] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中识别与当前编解码单元的AMVR语法元素相关联的编解码上下文或一个或多个处理器被配置为识别与当前编解码单元的AMVR语法元素相关联的编解码上下文是基于确定当前编解码单元的仿射模式。

[0178] 通常,实施例的至少一个示例可涉及根据本文所描述的方法或装置,其中确定当前编解码单元的仿射模式是基于帧间仿射标志。

[0179] 通常,实施例的至少一个示例可涉及包含指令的计算机程序产品,该指令在由计算机执行时致使计算机执行根据本文所描述的实施例的一个或多个示例的方法。

[0180] 通常,实施例的至少一个示例可涉及存储可执行程序指令的非暂时性计算机可读介质,该可执行程序指令使执行指令的计算机执行根据本文描述的实施例的一个或多个示例的方法。

[0181] 通常,实施例的至少一个示例可涉及包括根据本文所描述的实施例的任何一个或多个示例产生的数据的信号。

[0182] 通常,实施例的至少一个示例可涉及经格式化以包含根据本文所描述的实施例的示例中的任一个或多个示例而产生的语法元素和经编码的图像信息的比特流。

[0183] 通常,实施例的至少一个示例可涉及一种设备,该设备包括:根据在本文所描述的实施例的示例中的任何一个或多个示例的装置;以及以下中的至少一个:(i) 天线,被配置为接收信号,该信号包含表示图像信息的数据,(ii) 频带限制器,被配置为将包含表示图像信息的数据的所接收的信号限制到频带,以及(iii) 显示器,被配置为显示来自图像信息的图像。

[0184] 通常,实施例的至少一个示例可涉及根据本文所描述的设备,其中该设备包括电视、电视信号接收器、机顶盒、网关设备、移动设备、蜂窝式电话、平板或其他电子设备中的一个。

[0185] 已经描述了实施例的各种示例。根据本公开的这些和其他实施例可以包含跨越各种不同权利要求类别和类型、单独地或以任何组合的以下特征或实体中的任何特征或实体:

[0186] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量。

[0187] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量,并且其中减少基于使用左侧和上方语法元素。

[0188] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量,并且其中减少基于使用左侧和上方相邻语法元素。

[0189] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量,并且其中减少基于针对相同块尺寸的不同二进制位索引共享相同上下文。

[0190] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量,并且其中减少是基于当信令通知最后有效系数的坐标时针对不同块尺寸共享相同上下文索引集。

[0191] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量,并且其中减少是基于左侧语法元素和上方语法元素的总和,并且其中左侧和上方语法元素的总和能够基于左侧相邻元素和上方相邻元素的OR值或基于左侧相邻元素和上方相邻元素的AND值推导。

[0192] • 在编码器和/或解码器中提供用于基于由至少一个语法元素提供的信息应用一种形式的熵编解码,以及推导用于至少一个语法元素的上下文的数量,其中该推导包括减少上下文的数量,并且其中减少基于左侧语法元素和上方语法元素的总和,并且其中左侧语法元素和上方语法元素的总和能够基于不使用相邻元素而被推导。

[0193] • 在编码器和/或解码器中,如本文所描述的,提供根据以单独地或以任何组合的任一实施例、特征或实体的熵编解码的形式,其中熵编解码的形式包括CABAC。

[0194] • 在编码器和/或解码器中,如本文所描述的基于提供降低的复杂度和/或提高的压缩效率,提供根据以单独地或以任何组合的任一实施例、特征或实体的熵编解码的形式。

[0195] • 在信令中插入语法元素,该语法元素使得编码器和/或解码器能够根据单独地或以任何组合的实施例、特征或实体中的任一个提供编码和/或解码,如本文所描述。

[0196] • 基于本文所描述的单独或任意组合的这些语法元素、特征或实体选择以应用于

解码器。

[0197] • 包含一个或多个语法元素或其变体的比特流或信号。

[0198] • 在信令中插入语法元素,使解码器能够以与编码器所使用的编码方式相对应的方式提供解码。

[0199] • 创建和/或发送和/或接收和/或解码包含一个或多个语法元素或其变体的比特流或信号。

[0200] • TV、机顶盒、蜂窝电话、平板电脑或其他电子设备,其提供根据如本文所描述的、单独地或以任何组合的实施例、特征或实体中的任一个来应用编码和/或解码。

[0201] • TV、机顶盒、蜂窝电话、平板电脑或其他电子设备,其根据如本文所描述的、单独地或以任何组合的实施例、特征或实体中的任一个来执行编码和/或解码,并且显示(例如,使用监视器、屏幕或其他类型的显示器)所得图像。

[0202] • TV、机顶盒、蜂窝电话、平板电脑或其他电子设备,其调谐(例如,使用调谐器)频道以接收包含编码图像的信号,并且根据如本文所描述的、单独地或以任何组合的实施例、特征或实体中的任一个来执行编码和/或解码。

[0203] • TV、机顶盒、蜂窝电话、平板电脑或其他电子设备,其通过空中接收(例如,使用天线)包含编码图像的信号,并且根据如本文所描述的、单独地或以任何组合的实施例、特征或实体中的任一个来执行编码和/或解码。

[0204] • 存储程序代码的计算机程序产品,该程序代码在由计算机执行时,根据如本文所描述的、单独地或以任何组合的实施例、特征或实体中的任一个来进行编码和/或解码。

[0205] • 包含可执行程序指令的非暂时性计算机可读介质,该可执行程序指令使计算机执行指令以根据如本文所描述的、单独地或以任何组合的实施例、特征或实体中的任一个来实现编码和/或解码。

[0206] 在整个本公开中还支持和考虑了各种其他概括的以及具体的实施例。

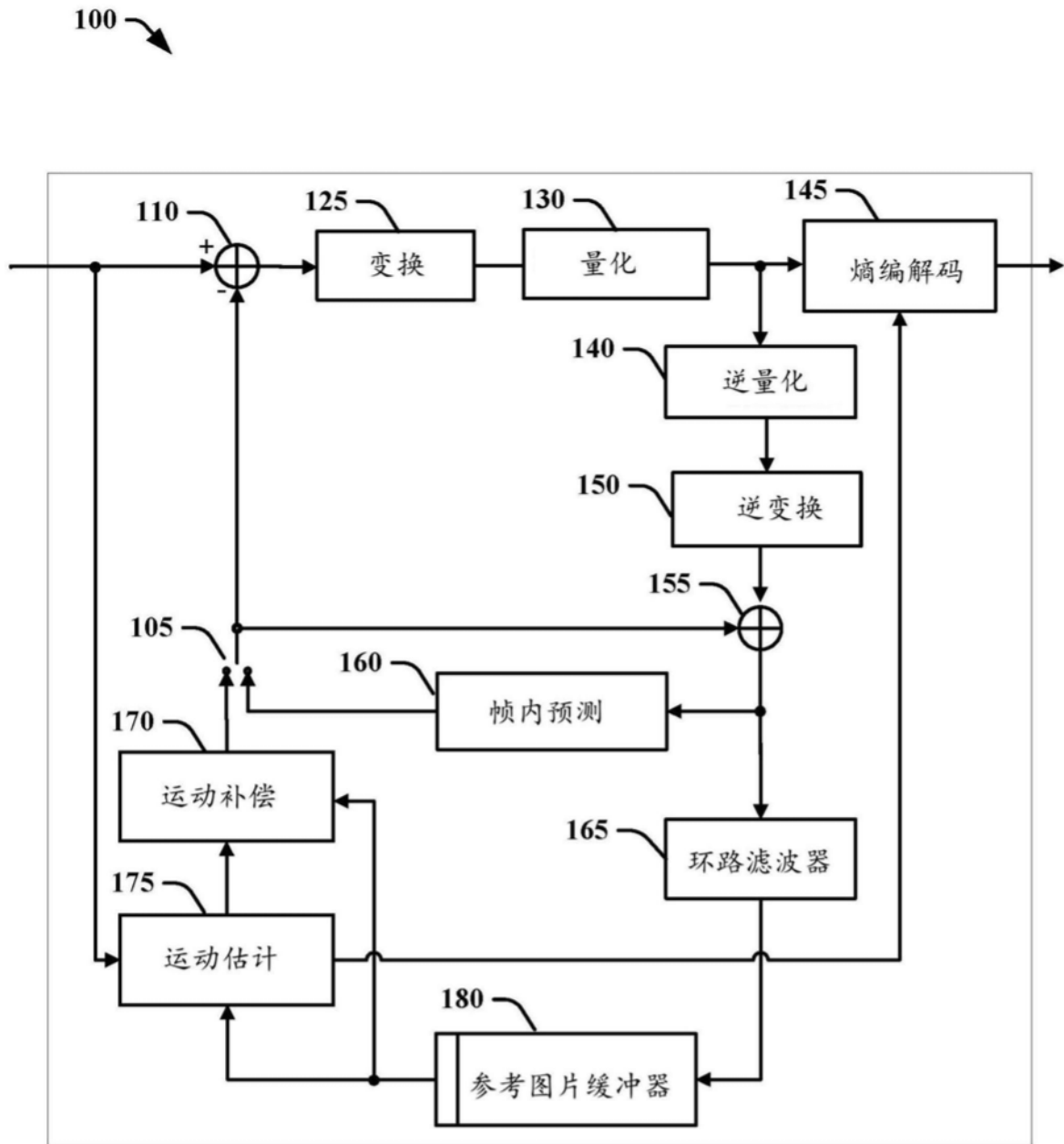


图1

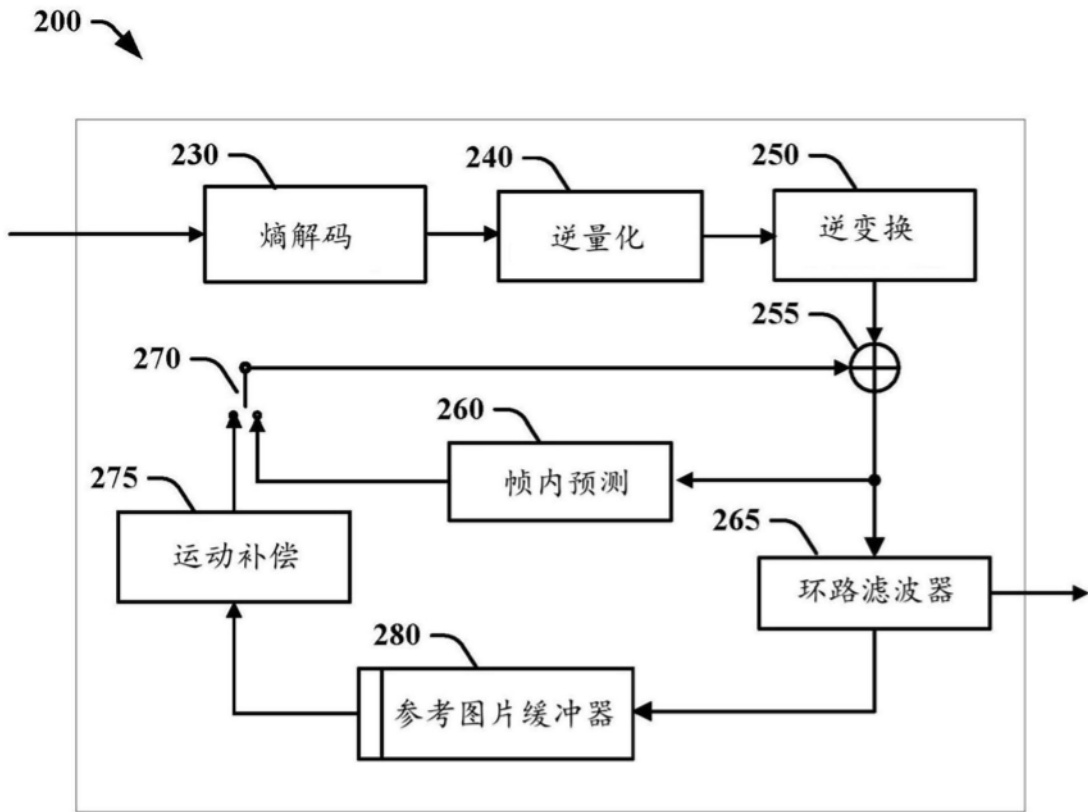


图2

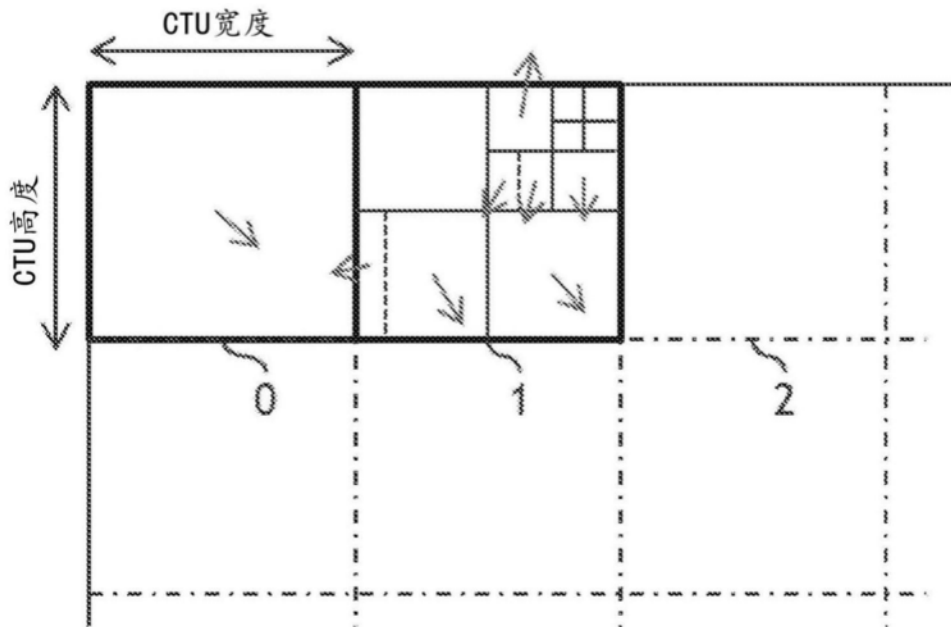


图3

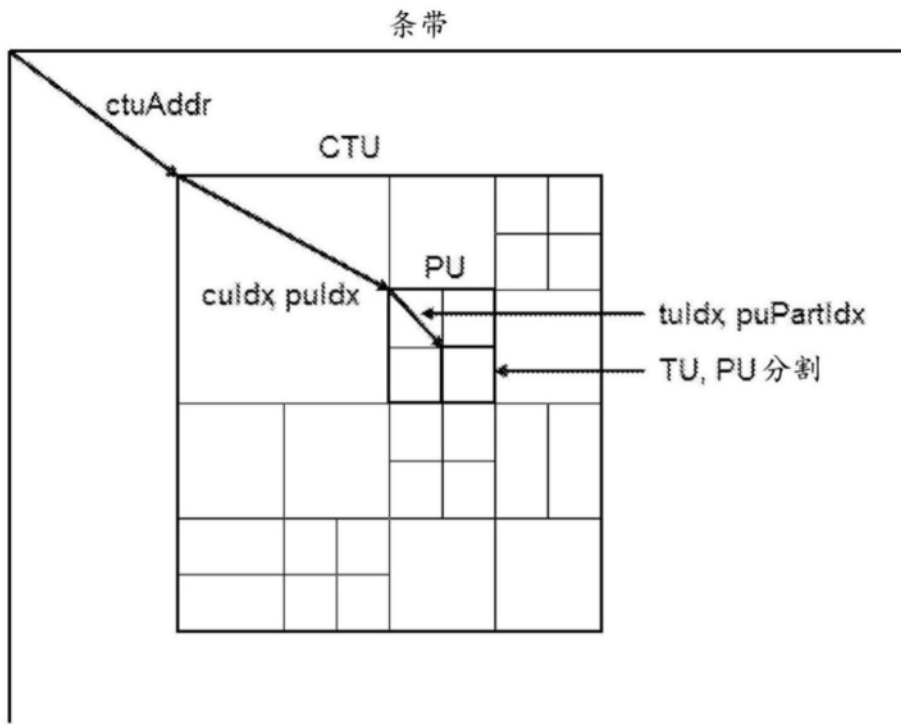


图4

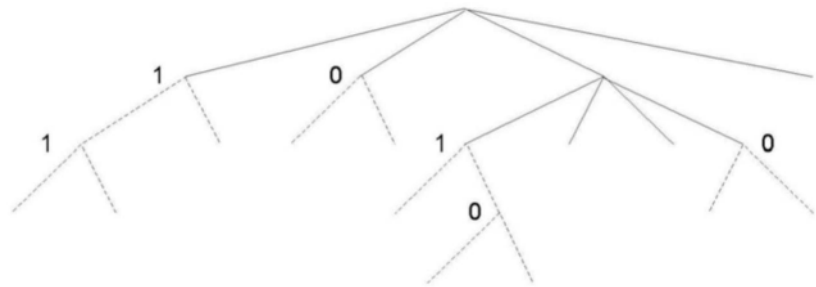
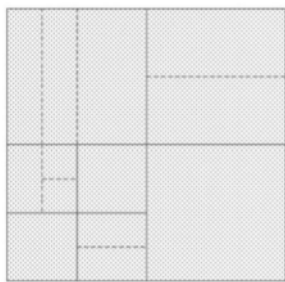


图5

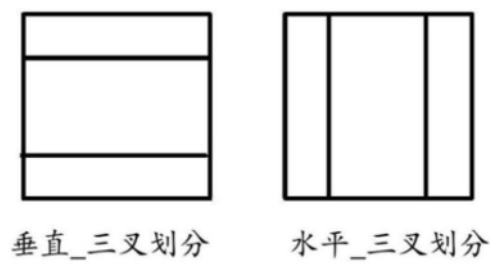


图6

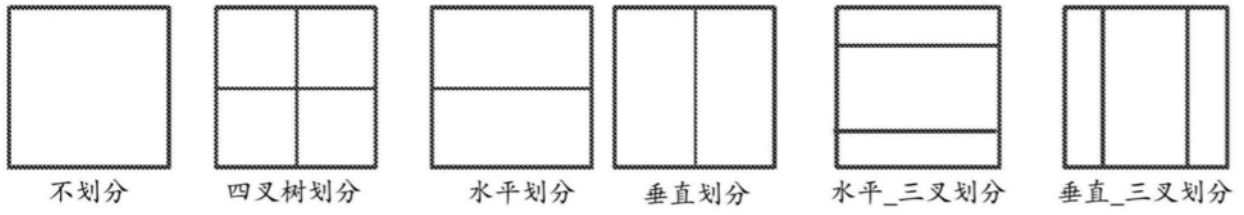


图7

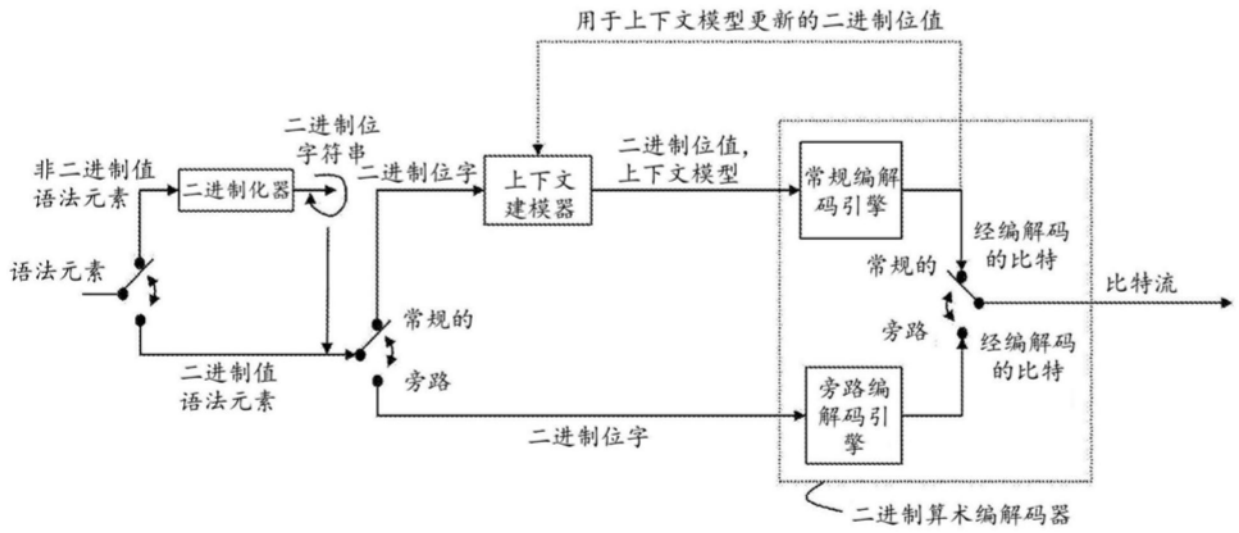


图8

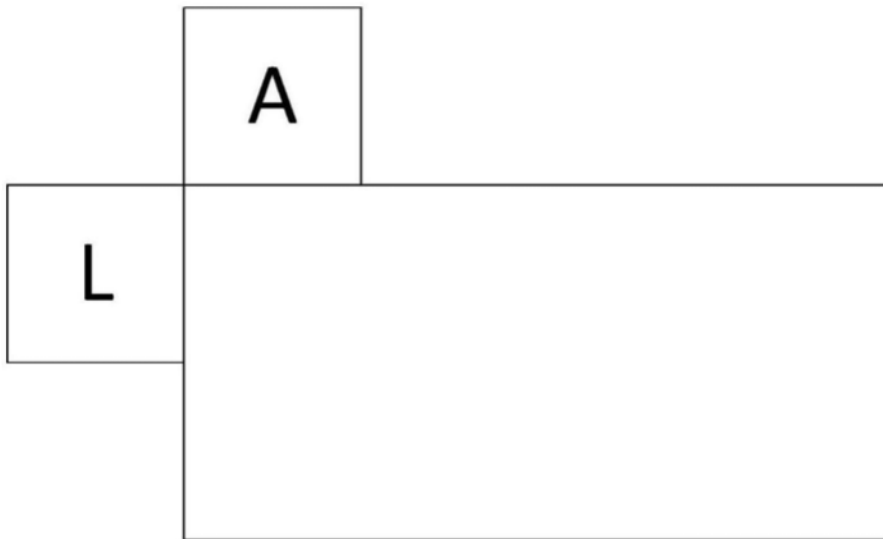


图9

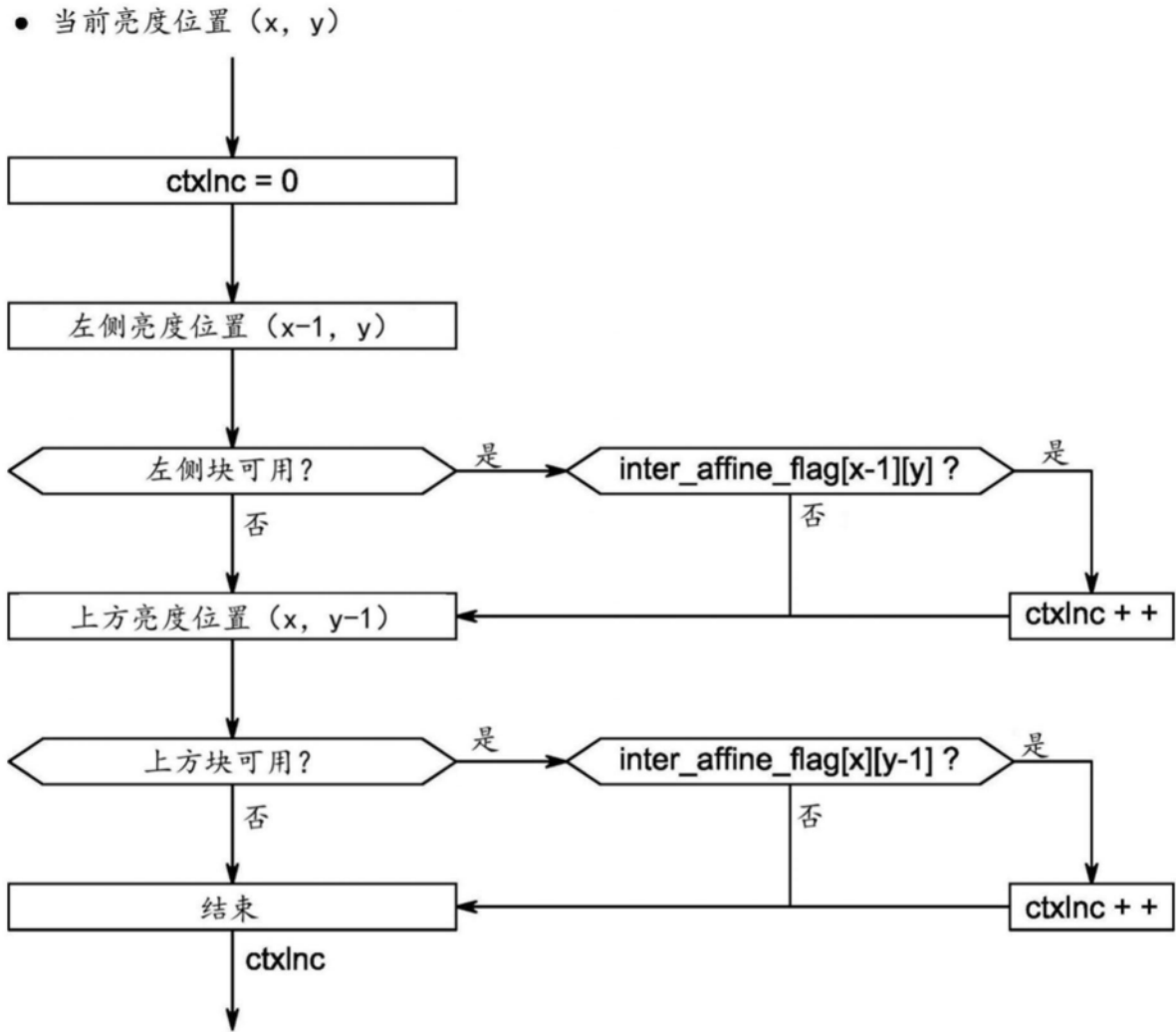


图10

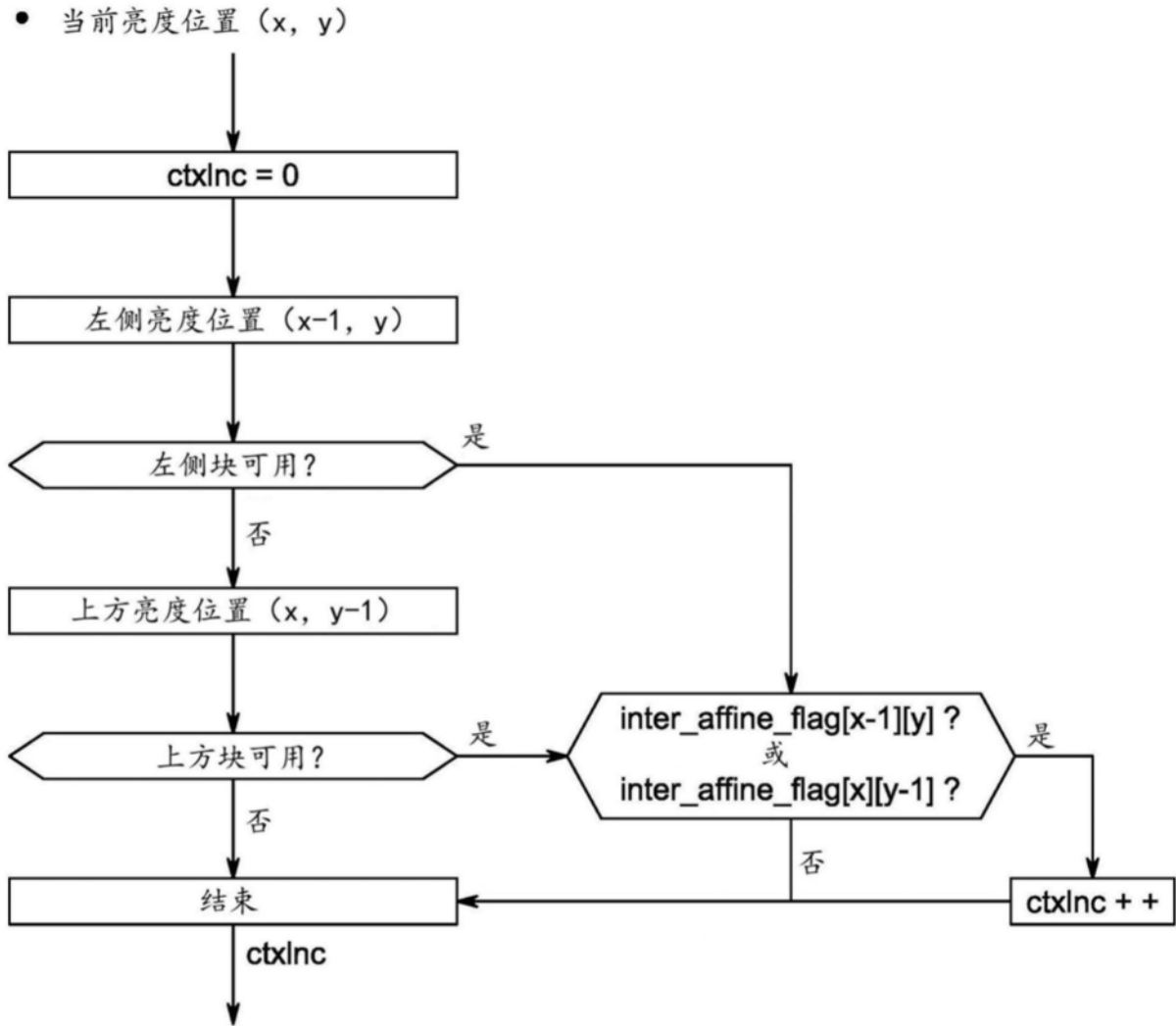


图11

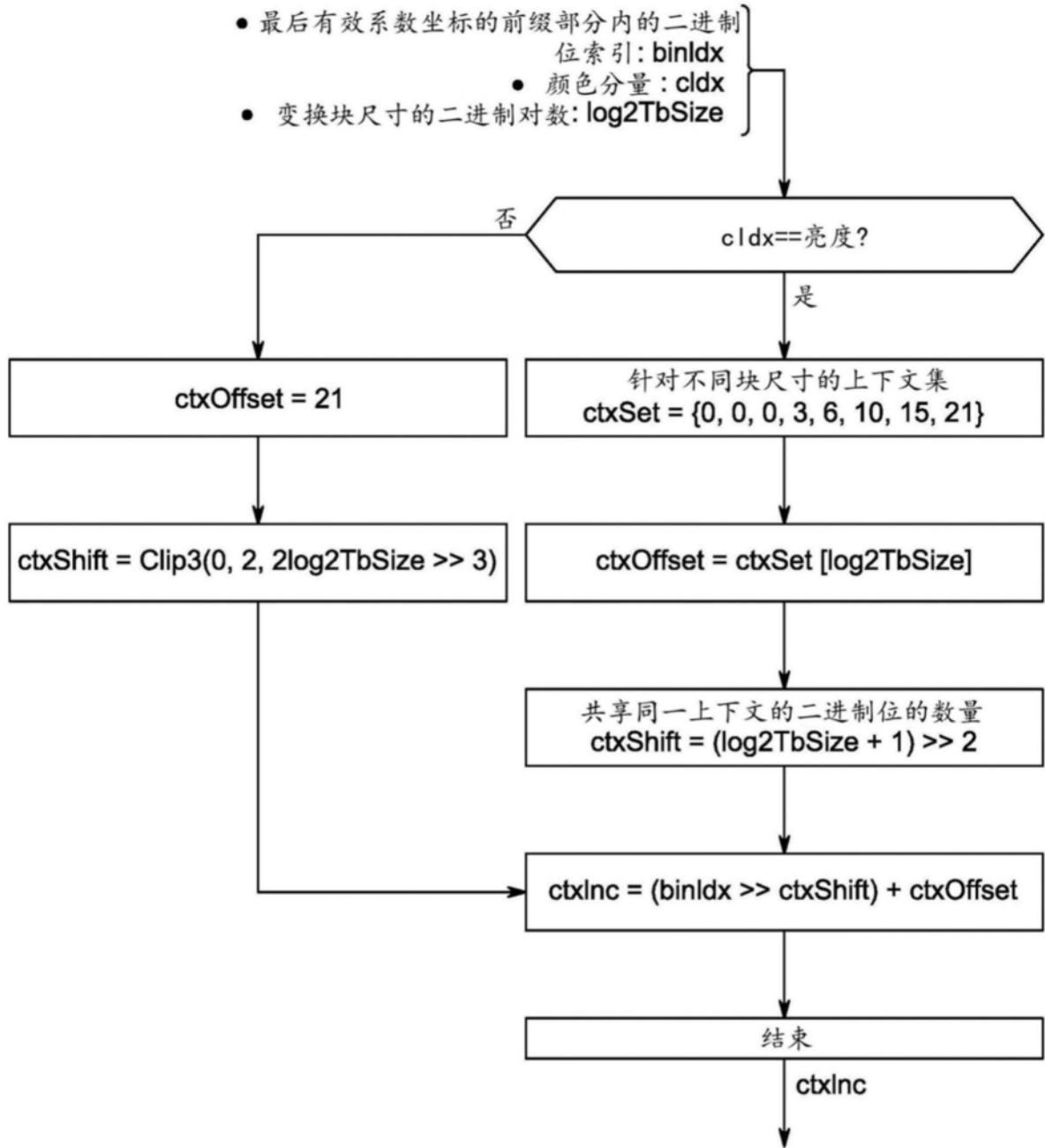


图12

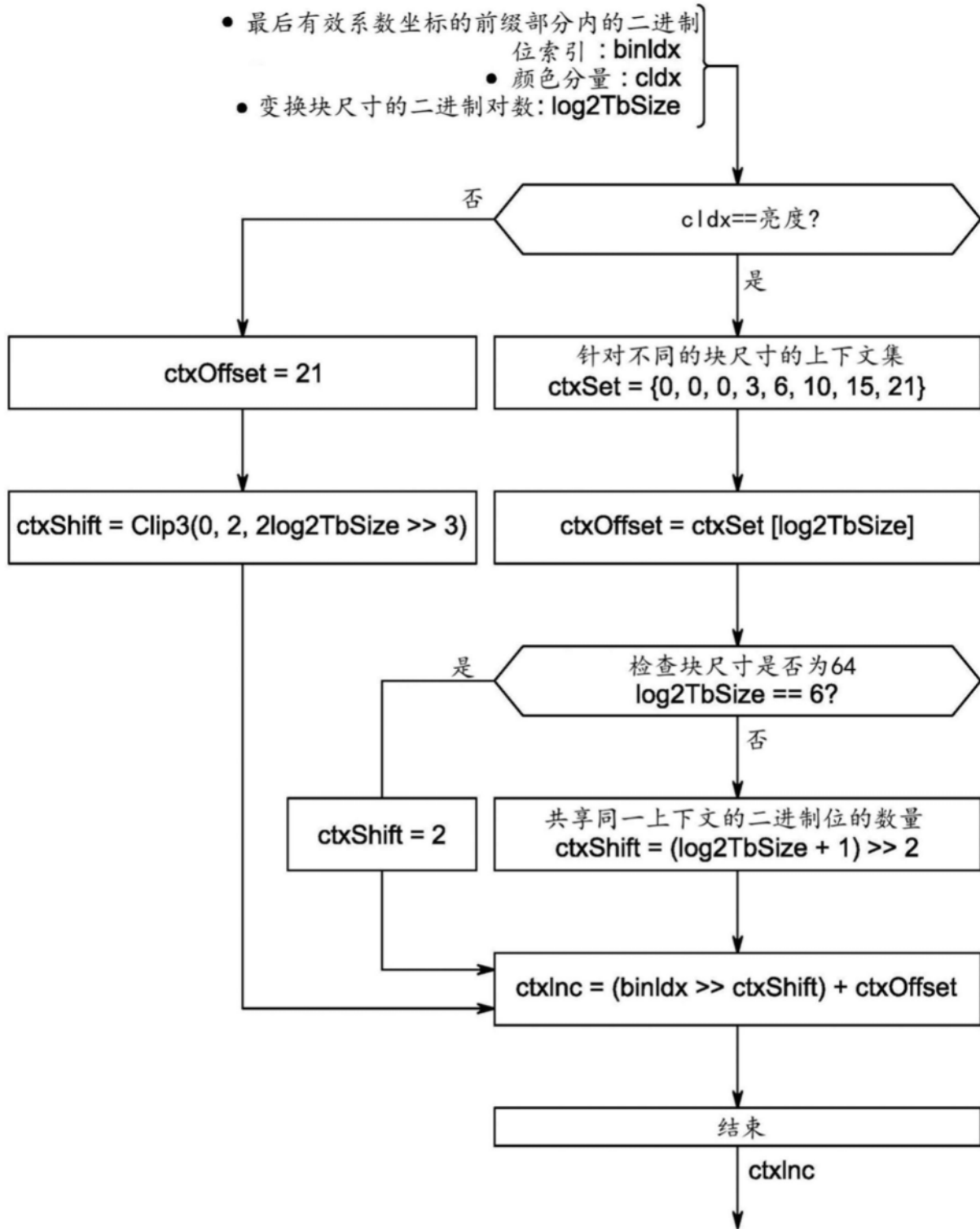


图13

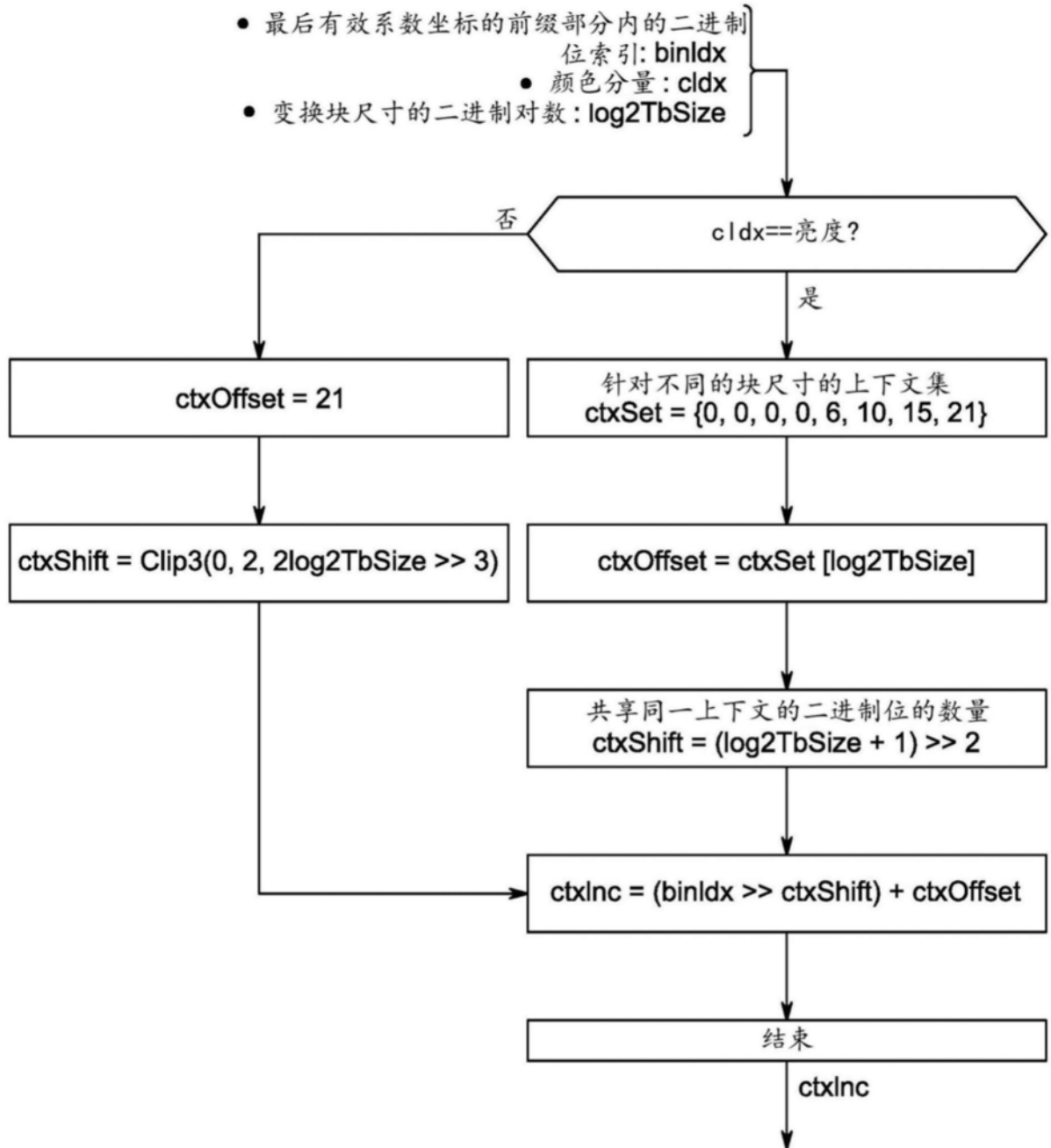


图14

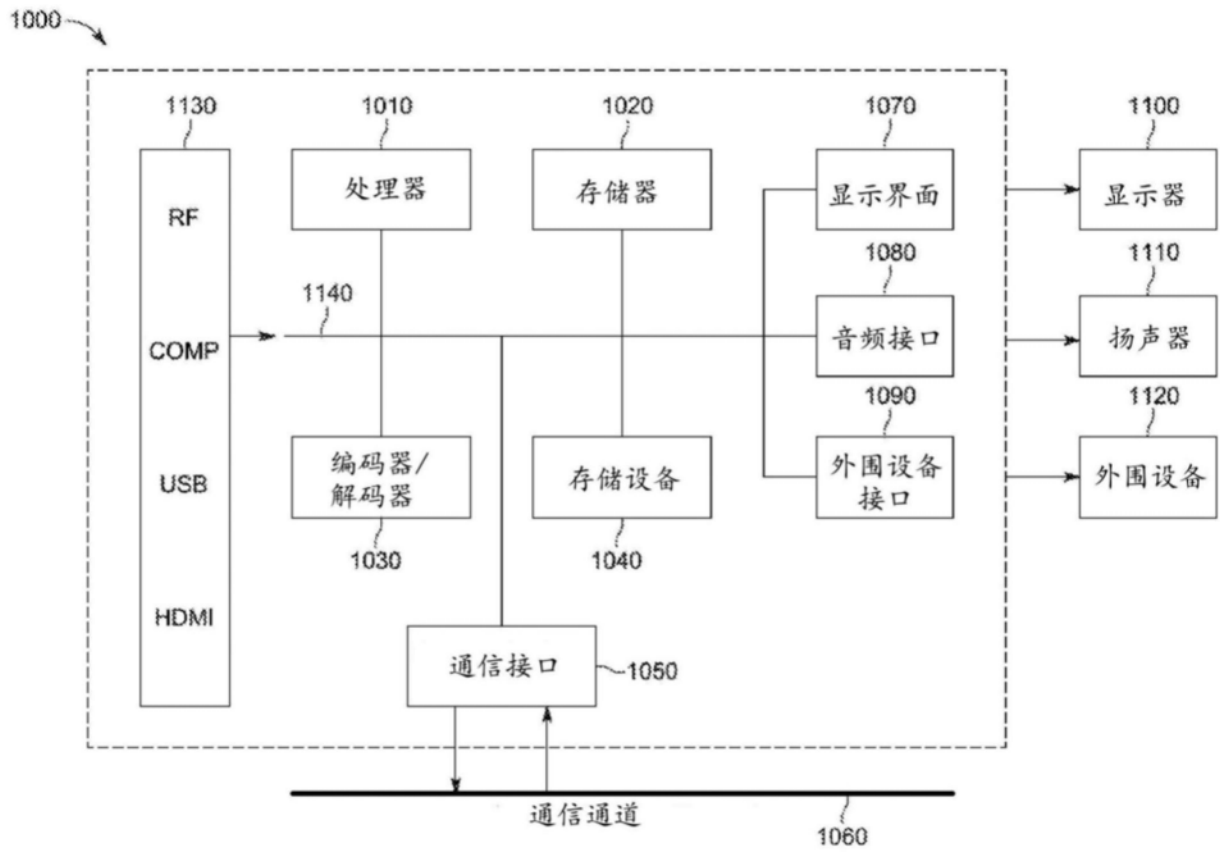


图15

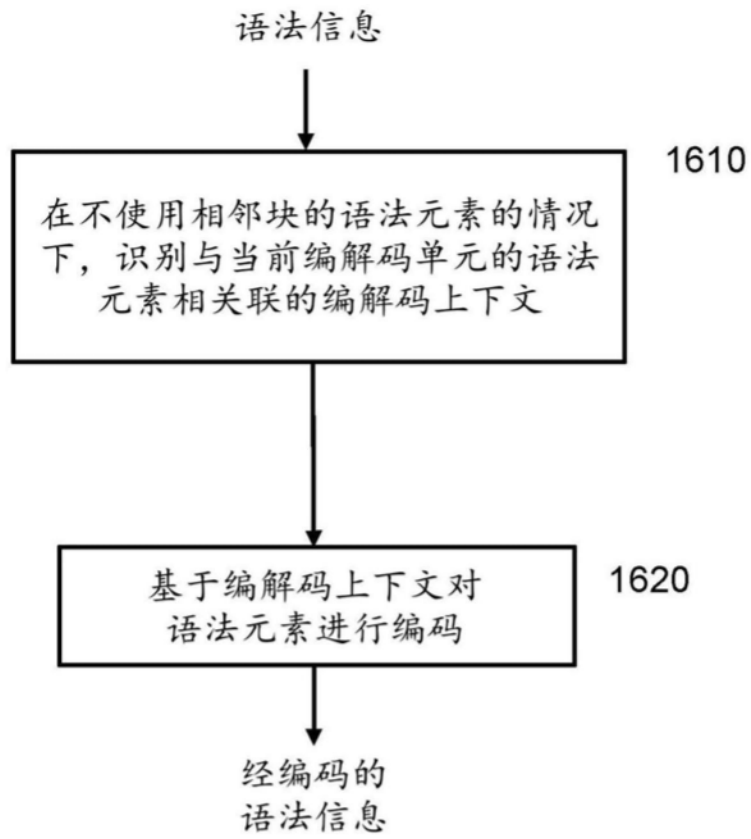


图16