



(12) 发明专利

(10) 授权公告号 CN 116828181 B

(45) 授权公告日 2024. 09. 27

(21) 申请号 202311027653.4

(22) 申请日 2020.02.12

(65) 同一申请的已公布的文献号  
申请公布号 CN 116828181 A

(43) 申请公布日 2023.09.29

(30) 优先权数据  
62/804,666 2019.02.12 US  
16/787,628 2020.02.11 US

(62) 分案原申请数据  
202080013272.9 2020.02.12

(73) 专利权人 腾讯美国有限责任公司  
地址 美国加利福尼亚州帕洛阿尔托公园大道2747号

(72) 发明人 赵欣 刘杉 李翔

(74) 专利代理机构 北京德琦知识产权代理有限公司 11018  
专利代理师 陈美娥 王琦

(51) Int.Cl.

H04N 19/12 (2014.01)

H04N 19/13 (2014.01)

H04N 19/132 (2014.01)

H04N 19/176 (2014.01)

H04N 19/18 (2014.01)

H04N 19/48 (2014.01)

H04N 19/70 (2014.01)

H04N 19/91 (2014.01)

(56) 对比文件

CN 113412622 B, 2023.06.23

WO 2018128323 A1, 2018.07.12

Moonmo Koo.《CE6: Reduced Secondary Transform (RST) (test 6.5.1)》.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11: JVET-M0292\_r5》.2019,正文第1-6页.

审查员 师佳彬

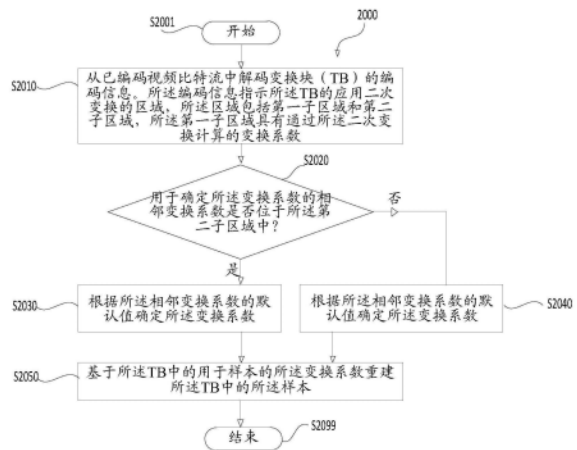
权利要求书3页 说明书45页 附图27页

(54) 发明名称

视频编解码的方法和装置

(57) 摘要

本公开的各方面提供了视频编解码的方法和装置。在一些示例中,视频解码的装置包括处理电路,所述处理电路可以从已编码视频比特流中解码变换块(TB)的编码信息。所述编码信息可以指示所述TB的应用二次变换的区域。所述区域可以包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数。对于所述TB中的变换系数,所述处理电路可以确定用于确定所述变换系数的相邻变换系数是否位于所述第二子区域中。当确定所述相邻变换系数位于所述第二子区域中时,所述处理电路可以根据所述相邻变换系数的默认值确定所述变换系数,并基于所述TB中的用于样本的所述变换系数重建所述样本。



1. 一种视频解码的方法,其特征在于,包括:

从已编码视频比特流中解码变换块TB的编码信息,所述编码信息指示所述TB的应用二次变换的区域,所述应用二次变换的区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数,所述第二子区域为归零区域;

当确定所述TB中的当前变换系数的相邻变换系数位于所述第二子区域中,且所述当前变换系数不在所述第二子区域中时,根据所述相邻变换系数的语法元素的默认值确定所述当前变换系数的语法元素的上下文值,其中,所述相邻变换系数的语法元素的默认值为零;以及

通过使用所述当前变换系数的语法元素的上下文值进行熵解码来确定所述当前变换系数的语法元素,并基于所述当前变换系数重建所述TB中的样本。

2. 根据权利要求1所述的方法,其特征在于,

所述TB中的所述当前变换系数是第一系数群组CG中的多个变换系数中的一个,所述第一CG的第一CG标志指示所述多个变换系数中的至少一个是否为非零变换系数;

包括变换系数的第二CG被预先熵解码,并且与所述第一CG相邻,其中,第二CG标志指示所述第二CG中的多个变换系数中的至少一个是否为非零变换系数;并且

所述方法进一步包括:

确定所述第二CG的位置;以及

当确定所述第二CG位于所述第二子区域中时,基于所述第二CG的第二CG标志的默认值导出所述第一CG标志的上下文值,其中,所述第二CG的第二CG标志的默认值为零。

3. 根据权利要求1所述的方法,其特征在于,

所述TB中的所述当前变换系数是第一CG中的多个变换系数中的一个,所述第一CG的第一CG标志指示所述多个变换系数中的至少一个是否为非零变换系数;

第二CG包括第一变换系数和第二变换系数,所述第二CG被预先熵解码并且与所述第一CG相邻,其中,所述第二CG标志指示所述第二CG中的多个变换系数中的至少一个是否为非零变换系数;并且

所述方法进一步包括:

确定所述第二CG的位置;以及

当所述第二CG的包括所述第二变换系数的部分位于所述第二子区域中并且所述第一变换系数为非零变换系数时,基于所述第二CG中的非零变换系数导出所述第一CG标志的上下文值。

4. 根据权利要求1所述的方法,其特征在于,进一步包括:

确定所述当前变换系数是否位于所述第二子区域中;

当确定所述当前变换系数位于所述第二子区域中时,确定所述当前变换系数的语法元素不用信号通知并且为零;以及

当确定所述当前变换系数不位于所述第二子区域中时,执行所述确定所述相邻变换系数是否位于所述第二子区域中。

5. 根据权利要求1-4任一项所述的方法,其特征在于,所述当前变换系数的语法元素指示以下之一:所述当前变换系数是否是非零变换系数;所述当前变换系数的奇偶性;所述当前变换系数是否大于2;以及所述当前变换系数是否大于4。

6. 一种视频解码的方法,其特征在于,包括:

从已编码视频比特流中解码变换块TB的编码信息;

基于所述编码信息确定是否对所述TB的第一区域执行二次变换,所述第一区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数,所述第二子区域为归零区域;以及

响应于确定对所述TB的第一区域执行所述二次变换,在导出用于熵解码的所述TB的当前变换系数的语法元素的上下文值时,将所述TB中的第二区域中的变换系数视为零,确定所述第二子区域和所述第二区域中的变换系数的语法元素为默认值零,所述第二区域在所述第一区域之外。

7. 根据权利要求6所述的方法,其特征在于,进一步包括:基于所述第一区域,确定所述TB中的包括多个变换系数的系数单元的大小和位置,并且所述系数单元之外的变换系数被视为零。

8. 根据权利要求7所述的方法,其特征在于,所述第一区域是所述TB中的左上 $8 \times 8$ 区域,所述系数单元是所述第一区域,所述第二区域与所述左上 $8 \times 8$ 区域相邻。

9. 根据权利要求8所述的方法,其特征在于,所述第一子区域是所述TB中的左上 $4 \times 4$ 区域,所述系数单元是所述第一区域中的所述第一子区域,并且包括所述第二区域和所述第二子区域的组合区域中的变换系数被视为零。

10. 根据权利要求7所述的方法,其特征在于,所述第一区域是所述TB中的左上 $4 \times 4$ 区域,所述系数单元是所述第一区域,所述第二区域与所述左上 $4 \times 4$ 区域相邻。

11. 一种视频编码的方法,其特征在于,包括:

确定变换块TB的编码信息,所述编码信息指示所述TB应用二次变换的区域,所述应用二次变换的区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数,所述第二子区域为归零区域;

当确定所述TB中的当前变换系数的相邻变换系数位于所述第二子区域中,且所述当前变换系数不在所述第二子区域中时,根据所述相邻变换系数的语法元素的默认值确定所述当前变换系数的语法元素的上下文值,其中,所述相邻变换系数的语法元素的默认值为零;以及

通过使用所述当前变换系数的语法元素的上下文值进行熵编码来确定所述当前变换系数的语法元素。

12. 一种视频编码的方法,其特征在于,包括:

确定是否对变换块TB的第一区域执行二次变换,所述第一区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数,所述第二子区域为归零区域;其中,响应于确定对所述TB的第一区域执行所述二次变换,在导出用于熵编码的所述TB的当前变换系数的语法元素的上下文值时,将所述TB中的第二区域中的变换系数视为零,确定所述第二子区域和所述第二区域中的变换系数的语法元素为默认值零,所述第二区域在所述第一区域之外。

13. 一种视频解码的装置,其特征在于,包括处理电路,被配置为执行如权利要求1-10任一项所述的方法。

14. 一种视频编码的装置,其特征在于,包括处理电路,被配置为执行如权利要求11或

12所述的方法。

15. 一种非易失性计算机可读介质,其特征在于,用于存储有指令,当所述指令由用于视频编码或解码的计算机执行时,使得所述计算机执行权利要求1-12任一项所述的方法。

16. 一种处理视频码流的方法,其特征在于,所述视频码流根据权利要求11或12所述的编码方法产生,或者基于权利要求1至10任一项所述的解码方法进行解码。

## 视频编解码的方法和装置

[0001] 通过引用并入本文

[0002] 本申请要求于2020年2月11日提交的、申请号为16/787,628、名称为“视频编解码的方法和装置”的美国专利申请的优先权,其要求于2019年2月12日提交的、申请号为62/804,666、发明名称为“变换系数归零”的美国临时申请的优先权。在先申请的全部公开内容通过引用整体并入本文。

### 技术领域

[0003] 本公开描述总体上涉及视频编解码的实施例。

### 背景技术

[0004] 可以使用具有运动补偿的图片间预测技术来执行视频编码和解码。未压缩的数字视频可以包括一系列图片,每个图片具有例如 $1920 \times 1080$ 亮度样本及相关色度样本的空间维度。所述系列图片可以具有固定的或可变的图片速率(也非正式地称为帧率),例如每秒60张图片或60Hz。未压缩的视频具有很高的比特率要求。例如,每个样本8比特的1080p60 4:2:0视频(60Hz帧率下 $1920 \times 1080$ 亮度样本分辨率)要求接近1.5Gbit/s带宽。一小时这样的视频就需要超过600GB的存储空间。

[0005] 视频编码和解码的一个目的是通过压缩减少输入视频信号的冗余。压缩可以帮助降低对上述带宽或存储空间的要求,在某些情况下可降低两个或更多数量级。无损压缩和有损压缩、以及两者的组合均可采用。无损压缩是指从压缩的原始信号中重建原始信号精确副本的技术。当使用有损压缩时,重建信号可能与原始信号不完全相同,但是原始信号和重建信号之间的失真足够小,使得重建信号可用于预期应用。有损压缩广泛应用于视频。容许的失真量取决于应用。例如,相比于电视应用的用户,某些消费流媒体应用的用户可以容忍更高的失真。可实现的压缩比反映出:较高的允许/容许失真可产生较高的压缩比。

[0006] 视频编码器和解码器可以利用几大类技术,例如包括:运动补偿、变换、量化和熵编码。

[0007] 视频编解码器技术可以包括已知的帧内编解码技术。在帧内编解码中,在不参考先前重建的参考图片的样本或其它数据的情况下表示样本值。在一些视频编解码器中,图片在空间上被细分为样本块。当所有的样本块都以帧内模式编解码时,该图片可以为帧内图片。帧内图片及其衍生(例如独立解码器刷新图片)可用于复位解码器状态,并且因此可用作已编码视频比特流和视频会话中的第一图片,或用作静止图像。帧内块的样本可用于变换,并且可以在熵编码之前量化变换系数。帧内预测可以是一种使预变换域中的样本值最小化的技术。在某些情况下,变换后的DC值越小,且AC系数越小,则在给定的量化步长尺寸下表示熵编码后的块所需的比特越少。

[0008] 如同从诸如MPEG-2代编解码技术中所获知的,传统帧内编解码不使用帧内预测。然而,一些较新的视频压缩技术包括:试图从例如周围样本数据和/或元数据中得到数据块的技术,其中周围样本数据和/或元数据是在空间相邻块的编码/解码期间、且在解码顺序

之前获得的。这种技术后来被称为“帧内预测”技术。需要注意的是,至少在某些情形下,帧内预测仅使用正在重建的当前图片的参考数据,而不使用参考图片的参考数据。

[0009] 可以存在许多不同形式的帧内预测。当在给定的视频编解码技术中可以使用超过一种这样的技术时,所使用的技术可以按帧内预测模式进行编解码。在某些情形下,模式可以具有子模式和/或参数,且这些模式可以单独编解码或包含在模式码字中。给定模式/子模式/参数组合使用哪个码字会影响通过帧内预测获得的编解码效率增益,因此用于将码字转换成比特流的熵编码技术也会出现这种情况。

[0010] H.264引入了一种帧内预测模式,其在H.265中进行了改进,且在诸如联合开发模型(JEM)、通用视频编解码(VVC)、基准集合(BMS)等更新的编解码技术中进一步被改进。使用属于已经可用的样本的相邻样本值可以形成预测块。将相邻样本的样本值按照某一方向复制到预测块中。对所使用方向的引用可以被编码在比特流中,或者本身可以被预测。

[0011] 参照图1A,右下方描绘了来自H.265的33个可能的预测方向(对应于35个帧内模式的33个角模式)中已知的九个预测方向的子集。箭头会聚的点(101)表示正在被预测的样本。箭头表示样本正在被预测的方向。例如,箭头(102)表示根据右上方与水平方向成45度角的一个或多个样本,预测样本(101)。类似地,箭头(103)表示根据左下方与水平方向成22.5度角的一个或多个样本,预测样本(101)。

[0012] 仍然参考图1A,在左上方示出了一个包括 $4 \times 4$ 个样本的正方形块(104)(由粗虚线表示)。正方形块(104)由16个样本组成,每个样本用“S”、以及其在Y维度(例如行索引)上的位置和在X纬度(例如列索引)上的位置来标记。例如,样本S21是Y维度上的第二个样本(从上方开始)和X维度上的第一个样本(从左侧开始)。类似地,样本S44在Y维度和X维度上都是块(104)中的第四个样本。由于该块为 $4 \times 4$ 大小的样本,因此S44位于右下角。还示出了遵循类似编号方案的参考样本。参考样本用“R”、以及其相对于块(104)的Y位置(例如,行索引)和X位置(例如,列索引)来标记。在H.264与H.265中,预测样本与正在重建的块相邻,因此不需要使用负值。

[0013] 通过从相邻样本复制参考样本值,可以进行图片内预测,其中相邻样本是由信号通知的预测方向所决定的。例如,假设已编码视频比特流包括信令,对于该块,该信令指示与箭头(102)一致的预测方向,即,根据右上方与水平方向成45度角的一个或多个预测样本来预测样本。在这种情况下,根据同一参考样本R05,预测样本S41、S32、S23和S14。然后,根据样本R08,预测样本S44。

[0014] 在某些情况下,例如通过内插,可以合并多个参考样本的值,以便计算参考样本,尤其是当方向不能被45度整除时。

[0015] 随着视频编解码技术的发展,可能的方向的数量已经增加了。在H.264(2003年)中,可以表示九种不同的方向。在H.265(2013年)和JEM/VVC/BMS中增加到了33个,而在此公开时,可以支持多达65个方向。已经进行了实验来识别最可能的方向,并且熵编码中的某些技术被用来使用少量比特来表示那些可能的方向,对于较不可能的方向则接受某些代价。此外,有时可以根据在相邻的、已经解码的块中所使用的相邻方向来预测方向本身。

[0016] 图1B示出了一种示意图(180),其描述了根据JEM的65种帧内预测方向,以说明随着时间的推移预测方向的数量增加。

[0017] 从帧内预测方向到已编码视频比特流中的表示方向的比特的映射可以因视频编

解码技术的不同而不同,例如,其范围可以从对帧内预测模式的预测方向到码字的简单直接映射,到包括最可能模式的复杂自适应方案以及类似技术。然而,在所有情况下,视频内容中可能存在某些方向,其在统计学上比其它方向更不可能出现。由于视频压缩的目的是减少冗余,所以在运行良好的视频编解码技术中,与更可能的方向相比,那些不太可能的方向将使用更多数量的比特来表示。

[0018] 可以使用具有运动补偿的图片间预测来执行视频编码和解码。运动补偿可以是一种有损压缩技术,且可涉及如下技术:来自先前重建的图片或重建图片一部分(参考图片)的样本数据块在空间上按运动矢量(下文称为MV)指示的方向移位后,用于新重建的图片或图片部分的预测。在某些情况下,参考图片可与当前正在重建的图片相同。MV可具有两个维度X和Y,或者三个维度,其中第三个维度表示正在使用的参考图片(后者间接地可以是时间维度)。

[0019] 在一些视频压缩技术中,应用于样本数据的某个区域的MV可根据其它MV来预测,例如根据与正在重建的区域空间相邻的样本数据的另一个区域相关的、且按解码顺序在该MV前面的那些MV。这样做可以大大减少编解码MV所需的数据量,从而消除冗余并增加压缩量。MV预测可以有效地进行,例如,因为在对从相机导出的输入视频信号(称为自然视频)进行编解码时,存在一种统计上的可能性,即面积大于单个MV适用区域的区域,会朝着类似的方向移动,因此,在某些情况下,可以使用从相邻区域的MV导出的相似运动矢量进行预测。这导致针对给定区域发现的MV与根据周围MV预测的MV相似或相同,并且在熵编解码之后,又可以用比直接编解码MV时使用的比特数更少的比特数来表示。在某些情况下,MV预测可以是对从原始信号(即样本流)导出的信号(即MV)进行无损压缩的示例。在其它情况下,例如由于根据几个周围MV计算预测值时产生的取整误差,MV预测本身可能是有损的。

[0020] H.265/HEVC (ITU-T H.265建议书,“高效视频编解码”,2016年12月)描述了各种MV预测机制。在H.265提供的多种MV预测机制中,本公开描述的是下文称为“空间合并”的技术。

[0021] 参照图2,当前块(201)包括编码器在运动搜索过程中发现的样本,所述样本可以根据已在空间上移位的相同大小的先前块进行预测。不直接对MV进行编解码,而是使用与五个周围样本中的任何一个相关联的MV,从与一个或多个参考图片相关联的元数据中导出该MV,例如,从最近的(按解码顺序)参考图片中导出该MV。其中,五个周围样本分别用A0、A1和B0、B1、B2(分别为202到206)表示。在H.265中,MV预测可使用相邻块正在使用的同一参考图片的预测值。

## 发明内容

[0022] 本公开的各方面提供了视频编解码的方法和装置。在一些示例中,视频解码的方法包括:从已编码视频比特流中解码变换块(TB)的编码信息,所述编码信息指示所述TB的应用二次变换的区域,所述区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数;对于所述TB中的当前变换系数,当确定所述当前变换系数的相邻变换系数位于所述第二子区域中时,根据所述相邻变换系数的默认值确定所述当前变换系数;以及基于所述当前变换系数重建所述TB中的样本。

[0023] 本公开的各方面提供了视频编解码的方法和装置。在一些示例中,视频解码的方

法包括：从已编码视频比特流中解码变换块TB的编码信息；基于所述编码信息确定是否对所述TB的第一区域执行二次变换，所述第一区域包括第一子区域和第二子区域，所述第一子区域具有通过所述二次变换计算的变换系数；以及当确定执行所述二次变换时，确定所述TB中的第二区域中的变换系数为零，所述第二区域在所述第一区域之外。

[0024] 本公开的各方面提供了视频编解码的方法和装置。在一些示例中，视频解码的方法包括：从已编码视频比特流中解码变换块TB的编码信息，所述编码信息指示所述TB的应用二次变换的区域，所述区域包括第一子区域和第二子区域，所述第一子区域具有通过所述二次变换计算的变换系数，所述第二子区域为归零区域；对于所述TB中的当前变换系数，确定是否使用所述第二子区域的变换系数之一来确定所述当前变换系数，所述第二子区域的变换系数之一与所述当前变换系数不同；当确定使用所述第二子区域的变换系数之一来确定所述当前变换系数时，根据所述第二子区域的变换系数之一的默认值确定所述当前变换系数，而与所述第二子区域的变换系数之一的实际值无关；基于所述当前变换系数重建所述TB中的样本。

[0025] 本公开的各方面提供了视频编解码的方法和装置。在一些示例中，视频解码的方法包括：从已编码视频比特流中解码变换块TB的编码信息；基于所述编码信息确定是否对所述TB的第一区域执行二次变换，所述第一区域包括第一子区域和第二子区域，所述第一子区域具有通过所述二次变换计算的变换系数，所述第二子区域的变换系数被设置为零；以及当确定对所述第一区域执行所述二次变换时，其中所述第一区域包括所述变换系数设置为零的第二子区域，确定所述TB中的第二区域的变换系数为零，所述第二区域在所述第一区域之外。

[0026] 本公开的各方面提供了视频编解码的方法和装置。在一些示例中，视频解码的装置包括处理电路。所述处理电路可以从已编码视频比特流中解码变换块(TB)的编码信息。所述编码信息可以指示所述TB的应用二次变换的区域。所述区域可以包括第一子区域和第二子区域，所述第一子区域具有通过所述二次变换计算的变换系数。对于所述TB中的变换系数，所述处理电路可以确定用于确定所述变换系数的相邻变换系数是否位于所述第二子区域中。当确定所述相邻变换系数位于所述第二子区域中时，所述处理电路可以根据所述相邻变换系数的默认值确定所述变换系数。所述处理电路可以基于所述TB中的用于样本的所述变换系数重建所述样本。

[0027] 在实施例中，所述TB中的所述变换系数是第一系数群组(CG)中的多个变换系数中的一个。所述第一CG的第一CG标志可以指示所述多个变换系数中的至少一个是否为非零变换系数。包括变换系数的第二CG被预先熵解码，并且与所述第一CG相邻。所述处理电路可以确定所述第二CG的位置。当确定所述第二CG位于所述第二子区域中时，所述处理电路可以基于所述第二CG的第二CG标志的默认值确定所述第一CG标志。

[0028] 在实施例中，所述TB中的所述变换系数是第一CG中的多个变换系数中的一个。所述第一CG的第一CG标志可以指示所述多个变换系数中的至少一个是否为非零变换系数。第二CG包括第一变换系数和第二变换系数。所述第二CG被预先熵解码并且与所述第一CG相邻。所述处理电路可以确定所述第二CG的位置。当所述第二CG的包括所述第二变换系数的部分位于所述第二子区域中并且所述第一变换系数为非零变换系数时，所述处理电路可以基于所述第二CG的第二CG标志确定所述第一CG标志。

[0029] 在实施例中,所述处理电路可以确定所述变换系数是否位于所述第二子区域中。当确定所述变换系数位于所述第二子区域中时,所述处理电路可以确定所述变换系数不用信号通知并且为零。当确定所述变换系数不位于所述第二子区域中时,所述处理电路可以执行所述确定所述相邻变换系数是否位于所述第二子区域中。

[0030] 在实施例中,所述处理电路可以确定所述变换系数的语法元素。所述语法元素可以指示以下之一:所述变换系数是否是非零变换系数;所述变换系数的奇偶性;所述变换系数是否大于2;以及所述变换系数是否大于4。

[0031] 本公开的各方面提供了视频解码的方法和装置。在一些示例中,视频解码的装置包括处理电路。所述处理电路可以从已编码视频比特流中解码变换块(TB)的编码信息。所述处理电路可以基于所述编码信息确定是否对所述TB的第一区域执行二次变换。所述第一区域可以包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数。当确定执行所述二次变换时,所述处理电路可以确定所述TB中的第二区域中的变换系数为零,其中所述第二区域在所述第一区域之外。

[0032] 在示例中,基于所述第一区域,确定所述TB中的包括多个变换系数的系数单元的大小和位置,并且所述系数单元之外的变换系数为零。

[0033] 在示例中,所述第一区域是所述TB中的左上 $8 \times 8$ 区域,所述系数单元是所述第一区域,所述第二区域与所述左上 $8 \times 8$ 区域相邻。

[0034] 在示例中,所述第一子区域是所述TB中的左上 $4 \times 4$ 区域,所述系数单元是所述第一区域中的所述第一子区域,并且包括所述第二区域和所述第二子区域的组合区域中的变换系数为零。

[0035] 在示例中,所述第一区域是所述TB中的左上 $4 \times 4$ 区域,所述系数单元是所述第一区域,所述第二区域与所述左上 $4 \times 4$ 区域相邻。

[0036] 本公开的各方面提供了视频解码的方法和装置。在一些示例中,视频解码的装置包括:解码模块,用于从已编码视频比特流中解码变换块TB的编码信息,所述编码信息指示所述TB的应用二次变换的区域,所述区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数;确定模块,用于对于所述TB中的当前变换系数,当确定所述当前变换系数的相邻变换系数位于所述第二子区域中时,根据所述相邻变换系数的默认值确定所述当前变换系数;以及重建模块,用于基于所述当前变换系数重建所述TB中的样本。

[0037] 本公开的方面还提供了一种非易失性计算机可读介质,用于存储有指令,当所述指令由用于视频解码的计算机执行时,使得所述计算机执行所述视频解码的方法。

## 附图说明

[0038] 根据以下详细描述和附图,所公开的主题的其他特征、性质和各种优点将进一步明确,其中:

[0039] 图1A是帧内预测模式的示例性子集的示意图。

[0040] 图1B是示例性帧内预测方向的示意图。

[0041] 图2是一个示例中的当前块及其周围的空间合并候选的示意图。

[0042] 图3是根据实施例的通信系统的简化框图的示意图。

- [0043] 图4是根据实施例的通信系统简化框图的示意图。
- [0044] 图5是根据实施例的解码器的简化框图的示意图。
- [0045] 图6是根据实施例的编码器的简化框图的示意图。
- [0046] 图7示出了根据另一实施例的编码器的框图。
- [0047] 图8示出了根据另一实施例的解码器的框图。
- [0048] 图9示出了变换单元语法的示例。
- [0049] 图10A-10B示出了残差编码语法的示例。
- [0050] 图11示出了子块扫描过程的示例。
- [0051] 图12示出了用于当前系数的上下文选择的局部模板的示例。
- [0052] 图13示出了残差编码语法的示例。
- [0053] 图14A-14D示出了子块变换(SBT)的示例。
- [0054] 图15A-15F示出了当使用SBT时对视频编解码标准的规范文本的改变。
- [0055] 图16A-16D示出了根据本公开实施例的系数单元的示例。
- [0056] 图17示出了示例性系数块。
- [0057] 图18示出了包括归零方法的示例性二次变换。
- [0058] 图19示出了包括归零方法的示例性二次变换。
- [0059] 图20示出了根据本公开实施例的概述过程的流程图。
- [0060] 图21示出了根据本公开实施例的概述过程的流程图。
- [0061] 图22是根据实施例的计算机系统的示意图。

### 具体实施方式

[0062] 图3示出了根据本公开实施例的通信系统(300)的简化框图。通信系统(300)包括多个终端装置,所述终端装置可通过例如网络(350)彼此通信。举例来说,通信系统(300)包括通过网络(350)互连的第一对终端装置(310)和(320)。在图3的实施例中,第一对终端装置(310)和(320)执行单向数据传输。举例来说,终端装置(310)可对视频数据(例如由终端装置(310)采集的视频图片流)进行编码以通过网络(350)传输到另一终端装置(320)。已编码的视频数据以一个或多个已编码视频码流形式传输。终端装置(320)可从网络(350)接收已编码视频数据,对已编码视频数据进行解码以恢复视频数据,并根据恢复的视频数据显示视频图片。单向数据传输在媒体服务等应用中是较常见的。

[0063] 在另一实施例中,通信系统(300)包括执行已编码视频数据的双向传输的第二对终端装置(330)和(340),所述双向传输可例如在视频会议期间发生。对于双向数据传输,在一个示例中,终端装置(330)和(340)中的每个终端装置可对视频数据(例如由终端装置采集的视频图片流)进行编码,以通过网络(350)传输到终端装置(330)和(340)中的另一终端装置。终端装置(330)和(340)中的每个终端装置还可接收由终端装置(330)和(340)中的另一终端装置传输的已编码视频数据,且可对所述已编码视频数据进行解码以恢复视频数据,且可根据恢复的视频数据在可访问的显示装置上显示视频图片。

[0064] 在图3的实施例中,终端装置(310)、(320)、(330)和(340)可为服务器、个人计算机和智能电话,但本公开的原理可不限于此。本公开的实施例适用于膝上型计算机、平板电脑、媒体播放器和/或专用视频会议设备。网络(350)表示在终端装置(310)、(320)、(330)和

(340)之间传送已编码视频数据的任何数目的网络,包括例如有线(连线的)和/或无线通信网络。通信网络(350)可在电路交换和/或分组交换信道中交换数据。该网络可包括电信网络、局域网、广域网和/或互联网。出于本公开的目的,除非在下文中有所解释,否则网络(350)的架构和拓扑对于本公开的操作来说可能是无关紧要的。

[0065] 作为所公开的主题的实施例,图4示出了视频编码器和视频解码器在流式传输环境中的放置方式。所公开主题可同等地适用于其它支持视频的应用,包括例如视频会议、数字TV、在包括CD、DVD、存储棒等的数字介质上存储压缩视频等等。

[0066] 流式传输系统可包括采集子系统(413),所述采集子系统可包括数码相机等视频源(401),所述视频源创建未压缩的视频图片流(402)。在实施例中,视频图片流(402)包括由数码相机拍摄的样本。相较于已编码的视频数据(404)(或已编码的视频码流),视频图片流(402)被描绘为粗线以强调高数据量的视频图片流,视频图片流(402)可由电子装置(420)处理,所述电子装置(420)包括耦接到视频源(401)的视频编码器(403)。视频编码器(403)可包括硬件、软件或软硬件组合以实现或实施如下文更详细地描述的所公开主题的各方面。相较于视频图片流(402),已编码的视频数据(404)(或已编码的视频码流(404))被描绘为细线以强调较低数据量的已编码的视频数据(404)(或已编码的视频码流(404)),其可存储在流式传输服务器(405)上以供将来使用。一个或多个流式传输客户端子系统,例如图4中的客户端子系统(406)和客户端子系统(408),可访问流式传输服务器(405)以检索已编码的视频数据(404)的副本(407)和副本(409)。客户端子系统(406)可包括例如电子装置(430)中的视频解码器(410)。视频解码器(410)对已编码的视频数据的传入副本(407)进行解码,且产生可在显示器(412)(例如显示屏)或另一呈现装置(未描绘)上呈现的输出视频图片流(411)。在一些流式传输系统中,可根据某些视频编码/压缩标准对已编码的视频数据(404)、视频数据(407)和视频数据(409)(例如视频码流)进行编码。该些标准的实施例包括ITU-TH.265。在实施例中,正在开发的视频编解码标准非正式地称为通用视频编解码(Versatile Video Coding,VVC),所公开的主题可用于VVC标准的上下文中。

[0067] 应注意,电子装置(420)和电子装置(430)可包括其它组件(未示出)。举例来说,电子装置(420)可包括视频解码器(未示出),且电子装置(430)还可包括视频编码器(未示出)。

[0068] 图5示出了根据本公开实施例的视频解码器(510)的框图。视频解码器(510)可设置在电子装置(530)中。电子装置(530)可包括接收器(531)(例如接收电路)。视频解码器(510)可用于代替图4实施例中的视频解码器(410)。

[0069] 接收器(531)可接收将由视频解码器(510)解码的一个或多个已编码视频序列;在同一实施例或另一实施例中,一次接收一个已编码视频序列,其中每个已编码视频序列的解码独立于其它已编码视频序列。可从信道(501)接收已编码视频序列,所述信道可以是通向存储已编码的视频数据的存储装置的硬件/软件链路。接收器(531)可接收已编码的视频数据以及其它数据,例如,可转发到它们各自的使用实体(未标示)的已编码音频数据和/或辅助数据流。接收器(531)可将已编码视频序列与其它数据分开。为了防止网络抖动,缓冲存储器(515)可耦接在接收器(531)与熵解码器/解析器(520)(此后称为“解析器(520)”)之间。在某些应用中,缓冲存储器(515)是视频解码器(510)的一部分。在其它情况下,所述缓冲存储器(515)可设置在视频解码器(510)外部(未标示)。而在其它情况下,视频解码器

(510)的外部设置缓冲存储器(未标示)以例如防止网络抖动,且在视频解码器(510)的内部可配置另一缓冲存储器(515)以例如处理播出定时。而当接收器(531)从具有足够带宽和可控性的存储/转发装置或从等时同步网络接收数据时,也可能不需要配置缓冲存储器(515),或可以将所述缓冲存储器做得较小。当然,为了在互联网等业务分组网络上使用,也可能需要缓冲存储器(515),所述缓冲存储器可相对较大且可具有自适应性大小,且可至少部分地实施于操作系统或视频解码器(510)外部的类似元件(未标示)中。

[0070] 视频解码器(510)可包括解析器(520)以根据已编码视频序列重建符号(521)。这些符号的类别包括用于管理视频解码器(510)的操作的信息,以及用以控制显示装置(512)(例如,显示屏)等显示装置的潜在信息,所述显示装置不是电子装置(530)的组成部分,但可耦接到电子装置(530),如图5中所示。用于显示装置的控制信息可以是辅助增强信息(Supplemental Enhancement Information,SEI消息)或视频可用性信息(Video Usability Information,VUI)的参数集片段(未标示)。解析器(520)可对接收到的已编码视频序列进行解析/熵解码。已编码视频序列的编码可根据视频编码技术或标准进行,且可遵循各种原理,包括可变长度编码、霍夫曼编码(Huffman coding)、具有或不具有上下文灵敏度的算术编码等等。解析器(520)可基于对应于群组的至少一个参数,从已编码视频序列提取用于视频解码器中的像素的子群中的至少一个子群的子群参数集。子群可包括图片群组(Group of Pictures,GOP)、图片、图块、切片、宏块、编码单元(Coding Unit,CU)、块、变换单元(Transform Unit,TU)、预测单元(Prediction Unit,PU)等等。解析器(520)还可从已编码视频序列提取信息,例如变换系数、量化器参数值、运动矢量等等。

[0071] 解析器(520)可对从缓冲存储器(515)接收的视频序列执行熵解码/解析操作,从而创建符号(521)。

[0072] 取决于已编码视频图片或一部分已编码视频图片(例如:帧间图片和帧内图片、帧间块和帧内块)的类型以及其它因素,符号(521)的重建可涉及多个不同单元。涉及哪些单元以及涉及方式可由解析器(520)从已编码视频序列解析的子群控制信息控制。为了简洁起见,未描述解析器(520)与下文的多个单元之间的此类子群控制信息流。

[0073] 除已经提及的功能块以外,视频解码器(510)可在概念上细分成如下文所描述的数个功能单元。在商业约束下运行的实际实施例中,这些单元中的许多单元彼此紧密交互并且可以彼此集成。然而,出于描述所公开主题的目的,概念上细分成下文的功能单元是适当的。

[0074] 第一单元是缩放器/逆变换单元(551)。缩放器/逆变换单元(551)从解析器(520)接收作为符号(521)的量化变换系数以及控制信息,包括使用哪种变换方式、块大小、量化因子、量化缩放矩阵等。缩放器/逆变换单元(551)可输出包括样本值的块,所述样本值可输入到聚合器(555)中。

[0075] 在一些情况下,缩放器/逆变换单元(551)的输出样本可属于帧内编码块;即:不使用来自先前重建的圖片的预测性信息,但可使用来自当前圖片的先前重建部分的预测性信息的块。此类预测性信息可由帧内图片预测单元(552)提供。在一些情况下,帧内图片预测单元(552)采用从当前图片缓冲器(558)提取的周围已重建的信息生成大小和形状与正在重建的块相同的块。举例来说,当前图片缓冲器(558)缓冲部分重建的当前图片和/或完全重建的当前图片。在一些情况下,聚合器(555)基于每个样本,将帧内预测单元(552)生成的

预测信息添加到由缩放器/逆变换单元 (551) 提供的输出样本信息中。

[0076] 在其它情况下,缩放器/逆变换单元 (551) 的输出样本可属于帧间编码和潜在运动补偿块。在此情况下,运动补偿预测单元 (553) 可访问参考图片存储器 (557) 以提取用于预测的样本。在根据符号 (521) 对提取的样本进行运动补偿之后,这些样本可由聚合器 (555) 添加到缩放器/逆变换单元 (551) 的输出(在这种情况下被称作残差样本或残差信号),从而生成输出样本信息。运动补偿预测单元 (553) 从参考图片存储器 (557) 内的地址获取预测样本可受到运动矢量控制,且所述运动矢量以所述符号 (521) 的形式而供运动补偿预测单元 (553) 使用,所述符号 (521) 例如是包括X、Y和参考图片分量。运动补偿还可包括在使用子样本精确运动矢量时,从参考图片存储器 (557) 提取的样本值的内插、运动矢量预测机制等等。

[0077] 聚合器 (555) 的输出样本可在环路滤波器单元 (556) 中被各种环路滤波技术采用。视频压缩技术可包括环路内滤波器技术,所述环路内滤波器技术受控于包括在已编码视频序列(也称作已编码视频码流)中的参数,且所述参数作为来自解析器 (520) 的符号 (521) 可用于环路滤波器单元 (556)。然而,在其他实施例中,视频压缩技术还可响应于在解码已编码图片或已编码视频序列的先前(按解码次序)部分期间获得的元信息,以及响应于先前重建且经过环路滤波的样本值。

[0078] 环路滤波器单元 (556) 的输出可以是样本流,所述样本流可输出到显示装置 (512) 以及存储在参考图片存储器 (557),以用于后续的帧间图片预测。

[0079] 一旦完全重建,某些已编码图片就可用作参考图片以用于将来预测。举例来说,一旦对应于当前图片的已编码图片被完全重建,且已编码图片(通过例如解析器 (520)) 被识别为参考图片,则当前图片缓冲器 (558) 可变为参考图片存储器 (557) 的一部分,且可在开始重建后续已编码图片之前重新分配新的当前图片缓冲器。

[0080] 视频解码器 (510) 可根据例如ITU-T H.265标准中的预定视频压缩技术执行解码操作。在已编码视频序列遵循视频压缩技术或标准的语法以及视频压缩技术或标准中记录的配置文件的意义上,已编码视频序列可符合所使用的视频压缩技术或标准指定的语法。具体地说,配置文件可从视频压缩技术或标准中可用的所有工具中选择某些工具作为在所述配置文件下可供使用的仅有工具。对于合规性,还要求已编码视频序列的复杂度处于视频压缩技术或标准的层级所限定的范围内。在一些情况下,层级限制最大图片大小、最大帧率、最大重建取样率(以例如每秒兆(mega)个样本为单位进行测量)、最大参考图片大小等。在一些情况下,由层级设定的限制可通过假想参考解码器(Hypothetical Reference Decoder,HRD)规范和已编码视频序列中用信号表示的HRD缓冲器管理的元数据来进一步限定。

[0081] 在实施例中,接收器 (531) 可连同已编码视频一起接收附加(冗余)数据。所述附加数据可以是已编码视频序列的一部分。所述附加数据可由视频解码器 (510) 用以对数据进行适当解码和/或较准确地重建原始视频数据。附加数据可呈例如时间、空间或信噪比(signal noise ratio,SNR) 增强层、冗余切片、冗余图片、前向纠错码等形式。

[0082] 图6示出了根据本公开实施例的视频编码器 (603) 的框图。视频编码器 (603) 设置于电子装置 (620) 中。电子装置 (620) 包括传输器 (640) (例如传输电路)。视频编码器 (603) 可用于代替图4实施例中的视频编码器 (403)。

[0083] 视频编码器(603)可从视频源(601)(并非图6实施例中的电子装置(620)的一部分)接收视频样本,所述视频源可采集将由视频编码器(603)编码的视频图像。在另一实施例中,视频源(601)是电子装置(620)的一部分。

[0084] 视频源(601)可提供将由视频编码器(603)编码的呈数字视频样本流形式的源视频序列,所述数字视频样本流可具有任何合适位深度(例如:8位、10位、12位……)、任何色彩空间(例如BT.601Y CrCb、RGB……)和任何合适取样结构(例如Y CrCb 4:2:0、Y CrCb 4:4:4)。在媒体服务系统中,视频源(601)可以是存储先前已准备的视频的存储装置。在视频会议系统中,视频源(601)可以是采集本地图像信息作为视频序列的相机。可将视频数据提供为多个单独的图片,当按顺序观看时,这些图片被赋予运动。图片自身可构建为空间像素阵列,其中取决于所用的取样结构、色彩空间等,每个像素可包括一个或多个样本。所属领域的技术人员可以很容易理解像素与样本之间的关系。下文侧重于描述样本。

[0085] 根据实施例,视频编码器(603)可实时或在由应用所要求的任何其它时间约束下,将源视频序列的图片编码且压缩成已编码视频序列(643)。施行适当的编码速度是控制器(650)的一个功能。在一些实施例中,控制器(650)控制如下文所描述的其它功能单元且在功能上耦接到这些单元。为了简洁起见,图中未标示耦接。由控制器(650)设置的参数可包括速率控制相关参数(图片跳过、量化器、率失真优化技术的 $\lambda$ 值等)、图片大小、图片群组(group of pictures,GOP)布局,最大运动矢量搜索范围等。控制器(650)可用于具有其它合适的功能,这些功能涉及针对某一系统设计优化的视频编码器(603)。

[0086] 在一些实施例中,视频编码器(603)在编码环路中进行操作。作为简单的描述,在实施例中,编码环路可包括源编码器(630)(例如,负责基于待编码的输入图片和参考图片创建符号,例如符号流)和嵌入于视频编码器(603)中的(本地)解码器(633)。解码器(633)以类似于(远程)解码器创建样本数据的方式重建符号以创建样本数据(因为在公开的主题中所考虑的视频压缩技术中,符号与已编码视频码流之间的任何压缩是无损的)。将重建的样本流(样本数据)输入到参考图片存储器(634)。由于符号流的解码产生与解码器位置(本地或远程)无关的位精确结果,因此参考图片存储器(634)中的内容在本地编码器与远程编码器之间也是按比特位精确对应的。换句话说,编码器的预测部分“看到”的参考图片样本与解码器将在解码期间使用预测时所“看到”的样本值完全相同。这种参考图片同步性基本原理(以及在例如因信道误差而无法维持同步性的情况下产生的漂移)也用于一些相关技术。

[0087] “本地”解码器(633)的操作可与例如已在上文结合图5详细描述视频解码器(510)的“远程”解码器相同。然而,另外简要参考图5,当符号可用且熵编码器(645)和解析器(520)能够无损地将符号编码/解码为已编码视频序列时,包括缓冲存储器(515)和解析器(520)在内的视频解码器(510)的熵解码部分,可能无法完全在本地解码器(633)中实施。

[0088] 此时可以观察到,除存在于解码器中的解析/熵解码之外的任何解码器技术,也必定以基本上相同的功能形式存在于对应的编码器中。出于此原因,所公开的主题侧重于解码器操作。可简化编码器技术的描述,因为编码器技术与全面地描述的解码器技术互逆。仅在某些区域中需要更详细的描述,并且在下文提供。

[0089] 在操作期间,在一些实施例中,源编码器(630)可执行运动补偿预测编码。参考来自视频序列中被指定为“参考图片”的一个或多个先前已编码图片,所述运动补偿预测编码

对输入图片进行预测性编码。以此方式,编码引擎(632)对输入图片的像素块与参考图片的像素块之间的差异进行编码,所述参考图片可被选作所述输入图片的预测参考。

[0090] 本地视频解码器(633)可基于源编码器(630)创建的符号,对可指定为参考图片的已编码视频数据进行解码。编码引擎(632)的操作可为有损过程。当已编码视频数据可在视频解码器(图6中未示)处被解码时,重建的视频序列通常可以是带有一些误差的源视频序列的副本。本地视频解码器(633)复制解码过程,所述解码过程可由视频解码器对参考图片执行,且可使重建的参考图片存储在参考图片高速缓存(634)中。以此方式,视频编码器(603)可在本地存储重建的参考图片的副本,所述副本与将由远端视频解码器获得的重建参考图片具有共同内容(不存在传输误差)。

[0091] 预测器(635)可针对编码引擎(632)执行预测搜索。即,对于将要编码的新图片,预测器(635)可在参考图片存储器(634)中搜索可作为所述新图片的适当预测参考的样本数据(作为候选参考像素块)或某些元数据,例如参考图片运动矢量、块形状等。预测器(635)可基于样本块逐像素块操作,以找到合适的预测参考。在一些情况下,根据预测器(635)获得的搜索结果,可确定输入图片可具有从参考图片存储器(634)中存储的多个参考图片取得的预测参考。

[0092] 控制器(650)可管理源编码器(630)的编码操作,包括例如设置用于对视频数据进行编码的参数和子群参数。

[0093] 可在熵编码器(645)中对所有上述功能单元的输出进行熵编码。熵编码器(645)根据诸如霍夫曼编码、可变长度编码、算术编码等技术对各种功能单元生成的符号进行无损压缩,从而将所述符号转换成已编码视频序列。

[0094] 传输器(640)可缓冲由熵编码器(645)创建的已编码视频序列,从而为通过通信信道(660)进行传输做准备,所述通信信道可以是通向将存储已编码的视频数据的存储装置的硬件/软件链路。传输器(640)可将来自视频编码器(603)的已编码视频数据与要传输的其它数据合并,所述其它数据例如是已编码音频数据和/或辅助数据流(未示出来源)。

[0095] 控制器(650)可管理视频编码器(603)的操作。在编码期间,控制器(650)可以为每个已编码图片分配某一已编码图片类型,但这可能影响可应用于相应的图片的编码技术。例如,通常可将图片分配为以下任一种图片类型:

[0096] 帧内图片(I图片),其可以是不将序列中的任何其它图片用作预测源就可被编码和解码的图片。一些视频编解码器容许不同类型的帧内图片,包括例如独立解码器刷新(Independent Decoder Refresh,“IDR”)图片。所属领域的技术人员了解I图片的变体及其相应的应用和特征。

[0097] 预测性图片(P图片),其可以是可使用帧内预测或帧间预测进行编码和解码的图片,所述帧内预测或帧间预测使用至多一个运动矢量和参考索引来预测每个块的样本值。

[0098] 双向预测性图片(B图片),其可以是可使用帧内预测或帧间预测进行编码和解码的图片,所述帧内预测或帧间预测使用至多两个运动矢量和参考索引来预测每个块的样本值。类似地,多个预测性图片可使用多于两个参考图片和相关联元数据以用于重建单个块。

[0099] 源图片通常可在空间上细分成多个样本块(例如,4×4、8×8、4×8或16×16个样本的块),且逐块进行编码。这些块可参考其它(已编码)块进行预测编码,根据应用于块的相应图片的编码分配来确定所述其它块。举例来说,I图片的块可进行非预测编码,或所述

块可参考同一图片的已经编码的块来进行预测编码(空间预测或帧内预测)。P图片的像素块可参考一个先前编码的参考图片通过空间预测或通过时域预测进行预测编码。B图片的块可参考一个或两个先前编码的参考图片通过空间预测或通过时域预测进行预测编码。

[0100] 视频编码器(603)可根据例如ITU-T H.265建议书的预定视频编码技术或标准执行编码操作。在操作中,视频编码器(603)可执行各种压缩操作,包括利用输入视频序列中的时间和空间冗余的预测编码操作。因此,已编码视频数据可符合所用视频编码技术或标准指定的语法。

[0101] 在实施例中,传输器(640)可在传输已编码的视频时传输附加数据。源编码器(630)可将此类数据作为已编码视频序列的一部分。附加数据可包括时间/空间/SNR增强层、冗余图片和切片等其它形式的冗余数据、补充增强信息(SEI)消息、视觉可用性信息(VUI)参数集片段等。

[0102] 采集到的视频可作为呈时间序列的多个源图片(视频图片)。帧内图片预测(常常简化为帧内预测)利用给定图片中的空间相关性,而帧间图片预测则利用图片之间的(时间或其它)相关性。在实施例中,将正在编码/解码的特定图片分割成块,正在编码/解码的特定图片被称作当前图片。在当前图片中的块类似于视频中先前已编码且仍被缓冲的参考图片中的参考块时,可通过称作运动矢量的矢量对当前图片中的块进行编码。所述运动矢量指向参考图片中的参考块,且在使用多个参考图片的情况下,所述运动矢量可具有识别参考图片的第三维度。

[0103] 在一些实施例中,双向预测技术可用于帧间图片预测中。根据双向预测技术,使用两个参考图片,例如按解码次序都在视频中的当前图片之前(但按显示次序可能分别是过去和将来)第一参考图片和第二参考图片。可通过指向第一参考图片中的第一参考块的第一运动矢量和指向第二参考图片中的第二参考块的第二运动矢量对当前图片中的块进行编码。具体来说,可通过第一参考块和第二参考块的组合来预测所述块。

[0104] 此外,合并模式技术可用于帧间图片预测中以改善编码效率。

[0105] 根据本公开的一些实施例,帧间图片预测和帧内图片预测等预测的执行以块为单位。举例来说,根据HEVC标准,将视频图片序列中的图片分割成编码树单元(coding tree unit,CTU)以用于压缩,图片中的CTU具有相同大小,例如 $64 \times 64$ 像素、 $32 \times 32$ 像素或 $16 \times 16$ 像素。一般来说,CTU包括三个编码树块(coding tree block,CTB),所述三个编码树块是一个亮度CTB和两个色度CTB。更进一步的,还可将每个CTU以二叉树拆分为一个或多个编码单元(coding unit,CU)。举例来说,可将 $64 \times 64$ 像素的CTU拆分为一个 $64 \times 64$ 像素的CU,或4个 $32 \times 32$ 像素的CU,或16个 $16 \times 16$ 像素的CU。在实施例中,分析每个CU以确定用于CU的预测类型,例如帧间预测类型或帧内预测类型。此外,取决于时间和/或空间可预测性,将CU拆分为一个或多个预测单元(prediction unit,PU)。通常,每个PU包括亮度预测块(prediction block,PB)和两个色度PB。在实施例中,编码(编码/解码)中的预测操作以预测块为单位来执行。以亮度预测块作为预测块为例,预测块包括像素值(例如,亮度值)的矩阵,例如 $8 \times 8$ 像素、 $16 \times 16$ 像素、 $8 \times 16$ 像素、 $16 \times 8$ 像素等等。

[0106] 图7示出了根据本公开另一实施例的视频编码器(703)的示意图。视频编码器(703)用于接收视频图片序列中的当前视频图片内的样本值的处理块(例如预测块),且将所述处理块编码到作为已编码视频序列的一部分的已编码图片中。在本实施例中,视频编

码器 (703) 用于代替图4实施例中的视频编码器 (403)。

[0107] 在HEVC实施例中,视频编码器 (703) 接收用于处理块的样本值的矩阵,所述处理块为例如 $8 \times 8$ 样本的预测块等。视频编码器 (703) 使用例如率失真 (rate-distortion) 优化来确定是否使用帧内模式、帧间模式或双向预测模式来编码所述处理块。当在帧内模式中编码处理块时,视频编码器 (703) 可使用帧内预测技术以将处理块编码到已编码图片中;且当在帧间模式或双向预测模式中编码处理块时,视频编码器 (703) 可分别使用帧间预测或双向预测技术将处理块编码到已编码图片中。在某些视频编码技术中,合并模式可以是帧间图片预测子模式,其中,在不借助预测值外部的已编码运动矢量分量的情况下,从一个或多个运动矢量预测值导出运动矢量。在某些其它视频编码技术中,可存在适用于主题块的运动矢量分量。在实施例中,视频编码器 (703) 包括其它组件,例如用于确定处理块模式的模式决策模块(未示出)。

[0108] 在图7的实施例中,视频编码器 (703) 包括如图7所示的耦接到一起的帧间编码器 (730)、帧内编码器 (722)、残差计算器 (723)、开关 (726)、残差编码器 (724)、通用控制器 (721) 和熵编码器 (725)。

[0109] 帧间编码器 (730) 用于接收当前块 (例如处理块) 的样本、比较所述块与参考图片中的一个或多个参考块 (例如先前图片和后来图片中的块)、生成帧间预测信息 (例如根据帧间编码技术的冗余信息描述、运动矢量、合并模式信息)、以及基于帧间预测信息使用任何合适的技术计算帧间预测结果 (例如已预测块)。在一些实施例中,参考图片是基于已编码的视频信息解码的已解码参考图片。

[0110] 帧内编码器 (722) 用于接收当前块 (例如处理块) 的样本、在一些情况下比较所述块与同一图片中已编码的块、在变换之后生成量化系数、以及在一些情况下还 (例如根据一个或多个帧内编码技术的帧内预测方向信息) 生成帧内预测信息。在实施例中,帧内编码器 (722) 还基于帧内预测信息和同一图片中的参考块计算帧内预测结果 (例如已预测块)。

[0111] 通用控制器 (721) 用于确定通用控制数据,且基于所述通用控制数据控制视频编码器 (703) 的其它组件。在实施例中,通用控制器 (721) 确定块的模式,且基于所述模式将控制信号提供到开关 (726)。举例来说,当所述模式是帧内模式时,通用控制器 (721) 控制开关 (726) 以选择供残差计算器 (723) 使用的帧内模式结果,且控制熵编码器 (725) 以选择帧内预测信息且将所述帧内预测信息添加在码流中;以及当所述模式是帧间模式时,通用控制器 (721) 控制开关 (726) 以选择供残差计算器 (723) 使用的帧间预测结果,且控制熵编码器 (725) 以选择帧间预测信息且将所述帧间预测信息添加在码流中。

[0112] 残差计算器 (723) 用于计算所接收的块与选自帧内编码器 (722) 或帧间编码器 (730) 的预测结果之间的差 (残差数据)。残差编码器 (724) 用于基于残差数据操作,以对残差数据进行编码以生成变换系数。在实施例中,残差编码器 (724) 用于将残差数据从空间域转换到频域,且生成变换系数。变换系数接着经由量化处理以获得量化的变换系数。在各种实施例中,视频编码器 (703) 还包括残差解码器 (728)。残差解码器 (728) 用于执行逆变换,且生成已解码残差数据。已解码残差数据可适当地由帧内编码器 (722) 和帧间编码器 (730) 使用。举例来说,帧间编码器 (730) 可基于已解码残差数据和帧间预测信息生成已解码块,且帧内编码器 (722) 可基于已解码残差数据和帧内预测信息生成已解码块。适当处理已解码块以生成已解码图片,且在一些实施例中,所述已解码图片可在存储器电路(未示出)中

缓冲并用作参考图片。

[0113] 熵编码器(725)用于将码流格式化以产生已编码的块。熵编码器(725)根据HEVC标准等合适标准产生各种信息。在实施例中,熵编码器(725)用于获得通用控制数据、所选预测信息(例如帧内预测信息或帧间预测信息)、残差信息和码流中的其它合适的信息。应注意,根据所公开的主题,当在帧间模式或双向预测模式的合并子模式中对块进行编码时,不存在残差信息。

[0114] 图8示出了根据本公开另一实施例的视频解码器(810)的示意图。视频解码器(810)用于接收作为已编码视频序列的一部分的已编码图像,且对所述已编码图像进行解码以生成重建的图片。在实施例中,视频解码器(810)用于代替图4实施例中的视频解码器(410)。

[0115] 在图8实施例中,视频解码器(810)包括如图8中所示耦接到一起的熵解码器(871)、帧间解码器(880)、残差解码器(873)、重建模块(874)和帧内解码器(872)。

[0116] 熵解码器(871)可用于根据已编码图片来重建某些符号,这些符号表示构成所述已编码图片的语法元素。此类符号可包括例如用于对所述块进行编码的模式(例如帧内模式、帧间模式、双向预测模式、后两者的合并子模式或另一子模式)、可分别识别供帧内解码器(872)或帧间解码器(880)用以进行预测的某些样本或元数据的预测信息(例如帧内预测信息或帧间预测信息)、呈例如量化的变换系数形式的残差信息等等。在实施例中,当预测模式是帧间或双向预测模式时,将帧间预测信息提供到帧间解码器(880);以及当预测类型是帧内预测类型时,将帧内预测信息提供到帧内解码器(872)。残差信息可经由逆量化并提供到残差解码器(873)。

[0117] 帧间解码器(880)用于接收帧间预测信息,且基于所述帧间预测信息生成帧间预测结果。

[0118] 帧内解码器(872)用于接收帧内预测信息,且基于所述帧内预测信息生成预测结果。

[0119] 残差解码器(873)用于执行逆量化以提取解量化的变换系数,且处理所述解量化的变换系数,以将残差从频域转换到空间域。残差解码器(873)还可能某些控制信息(用以获得量化器参数QP),且所述信息可由熵解码器(871)提供(未标示数据路径,因为这仅仅是低量控制信息)。

[0120] 重建模块(874)用于在空间域中组合由残差解码器(873)输出的残差与预测结果(可由帧间预测模块或帧内预测模块输出)以形成重建的块,所述重建的块可以是重建的图片的一部分,所述重建的图片继而可以是重建的视频的一部分。应注意,可执行解块操作等其它合适的操作来改善视觉质量。

[0121] 应注意,可使用任何合适的技术来实施视频编码器(403)、视频编码器(603)和视频编码器(703)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。在实施例中,可使用一个或多个集成电路来实施视频编码器(403)、视频编码器(603)和视频编码器(703)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。在另一实施例中,可使用执行软件指令的一个或多个处理器来实施视频编码器(403)、视频编码器(603)和视频编码器(703)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。

[0122] 在一些实施例中,例如在HEVC中,主变换可以包括4点、8点、16点和32点离散余弦

变换 (DCT) 类型2 (DCT-2), 并且变换核矩阵可以使用8位整数 (即, 8位变换核) 表示。较小 DCT-2的变换核矩阵是较大DCT-2的变换核矩阵的一部分, 如附录I所示。

[0123] DCT-2核矩阵显示出了对称/反对称特性。因此, 可以支持“部分蝶形”的实施方式以减少操作计数 (例如, 乘法、加法、减法、移位等), 并且可以使用部分蝶形获得矩阵乘法的相同结果。

[0124] 在一些实施例中, 例如在VVC中, 除了上述的4点、8点、16点和32点DCT-2变换之外, 还可以包括额外的2点和64点DCT-2。例如在VVC中使用的64点DCT-2核的示例在附录II中显示为64×64矩阵。

[0125] 除了例如在HEVC中使用的DCT-2和4×4DST-7之外, 还可以 (例如在VVC中) 使用自适应多变换 (AMT) (也称为增强多变换 (EMT) 或多变换选择 (MTS)) 方案对帧间和帧内编码块进行残差编码。AMT方案可以使用从DCT/DST族中选定的、除HEVC中的当前变换之外的多个变换。新引入的变换矩阵是DST-7、DCT-8。表1示出了针对N点输入的所选DST/DCT的基函数的示例。

变换类型	基函数 $T_i(j), i, j=0, 1, \dots, N-1$
DCT-2	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ <p>其中 <math>\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} &amp; i = 0 \\ 1 &amp; i \neq 0 \end{cases}</math></p>
DCT-8	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-7	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

[0127] 表1

[0128] 主变换矩阵 (例如在VVC中使用的) 可以以8位表示来使用。AMT将变换矩阵应用于宽度和高度都小于或等于32的变换块。是否应用AMT可以由标志 (例如, mts\_flag) 来控制。当mts\_flag等于0时, 在一些示例中, 仅应用DCT-2对残差数据进行编码。当mts\_flag等于1时, 可以使用2个二进制数 (bin) 进一步用信号通知索引 (例如, mts\_idx), 以识别例如根据表2要使用的水平和垂直变换, 其中类型值1表示使用DST-7, 类型值2表示使用DCT-8。在表2中, trTypeHor和trTypeVer的规范取决于mts\_idx[x][y][cIdx]。

mts_idx[xTbY][yTbY][cIdx]	trTypeHor	trTypeVer
-1	0	0
0	1	1
1	2	1
2	1	2
3	2	2

[0130] 表2

[0131] 在一些实施例中,当不使用上述基于信令的MTS(即,显式MTS)时,可以应用隐式MTS。利用隐式MTS,根据块的宽度和高度而不是根据信令来进行变换选择。例如,利用隐式MTS,DST-7被选择用于 $M \times N$ 块的较短边(即, $M$ 和 $N$ 中的最小一个),并且DCT-2被选择用于该块的较长边(即, $M$ 和 $N$ 中的最大一个)。

[0132] 在附录III中示出了示例性变换核,每个示例性变换核都是由DST-7和DCT-8的基矢量组成的矩阵。

[0133] 在一些示例中,例如在VVC中,当编码块的高度和宽度都小于或等于64时,TB大小与编码块大小相同。当编码块的高度或宽度大于64时,当进行变换(例如,逆变换、逆主变换等)或帧内预测时,将编码块进一步分割成多个子块,其中每个子块的宽度和高度小于或等于64。可以对每一子块执行一次变换。

[0134] 可以在下面的图9和图10A-10B中描述VVC中一些示例中的MTS的相关语法和语义(例如,相关语法用框901和1001来突出显示)。图9示出了变换单元语法的示例。图10A-10B示出了残差编码语法的示例。变换单元语义的示例如下。`cu_mts_flag[x0][y0]`等于1指定将多变换选择应用于相关亮度变换块的残差样本。`cu_mts_flag[x0][y0]`等于0指定不将多变换选择应用于相关亮度变换块的残差样本。数组索引 $x_0$ 、 $y_0$ 指定所考虑的变换块的左上亮度样本相对于图片的左上亮度样本的位置( $x_0$ , $y_0$ )。当`cu_mts_flag[x0][y0]`不存在时,推断其等于0。

[0135] 残差编码语义的示例如下。`mts_idx[x0][y0]`指定沿当前变换块的水平和垂直方向将哪些变换核应用于亮度残差样本。数组索引 $x_0$ 、 $y_0$ 指定所考虑的变换块的左上亮度样本相对于图片的左上亮度样本的位置( $x_0$ , $y_0$ )。当`mts_idx[x0][y0]`不存在时,推断其等于-1。

[0136] 在一些实施例中,可以将编码块分割成具有例如 $4 \times 4$ 大小的子块。可以根据预定义的扫描顺序对编码块内的子块和每个子块内的变换系数进行编码。对于具有至少一个非零变换系数的子块,可以将变换系数的编码分成多个扫描过程(pass),例如4个扫描过程。在每一个扫描过程期间,可以例如以逆对角线扫描顺序扫描各个子块中的变换系数(也称为系数)。

[0137] 图11示出了子块扫描过程(1100)的示例,从该子块扫描过程可以生成变换系数的不同类型的语法元素。子块可以是具有十六个系数(1110)的 $4 \times 4$ 子块。可以基于例如从0到15的扫描顺序来扫描这16个系数(1110),如图11所示。在第一个扫描过程期间,对系数(1110)进行扫描,并且可以为系数(1110)中的每一个生成三种类型的语法元素(1101)至(1103):

[0138] (i) 第一种类型的语法元素(1101)可以是重要性标志(例如,`sig_coeff_flag`),其指示绝对变换系数级别或相应变换系数的值(例如,`absLevel`)是否大于零。第一种类型的语法元素可以是二进制语法元素。

[0139] (ii) 第二种类型的语法元素(1102)可以是奇偶校验标志(例如,`par_level_flag`),其指示相应变换系数的绝对变换系数级别的奇偶校验。在示例中,仅在相应变换系数的绝对变换系数级别为非零时生成奇偶校验标志。第二种类型的语法元素可以是二进制语法元素。

[0140] (iii) 第三种类型的语法元素 (1103) 可以是大于1标志 (例如, `rem_abs_gt1_flag`), 其指示  $(absLevel-1) \gg 1$  对于相应变换系数是否大于0。在示例中, 仅在相应变换系数的绝对变换系数级别为非零时生成大于1标志。第三种类型的语法元素可以是二进制语法元素。

[0141] 在第二个扫描过程期间, 可能会生成第四种类型的语法元素 (1104)。第四种类型的语法元素 (1104) 可以是大于2标志 (例如, `rem_abs_gt2_flag`)。第四种类型的语法元素 (1104) 指示相应变换系数的绝对变换系数级别是否大于4。在示例中, 仅当  $(absLevel-1) \gg 1$  对于相应变换系数大于0时生成大于2标志。第四种类型的语法元素可以是二进制语法元素。

[0142] 在第三个扫描过程期间, 可能会生成第五种类型的语法元素 (1105)。第五种类型的语法元素 (1105) 由 `abs_remainder` 表示, 并且指示大于4的相应变换系数的绝对变换系数级别的剩余值。在示例中, 仅当相应变换系数的绝对变换系数级别大于4时才生成第五种类型的语法元素 (1105)。第五种类型的语法元素可以是非二进制语法元素。

[0143] 在第四个扫描过程期间, 可以为具有非零系数级别的系数 (1110) 生成第六种类型的语法元素 (1106), 其指示相应变换系数 (1110) 的符号。

[0144] 可以根据这些扫描过程的顺序和每个过程中的扫描顺序, 将上述各种类型的语法元素 (1101至1106) 提供给熵编码器。可以采用不同的熵编码方案来编码不同类型的语法元素。例如, 在实施例中, 可以使用基于CABAC的熵编码器对重要性标志、奇偶校验标志、大于1标志和大于2标志进行编码。相反, 可以使用CABAC旁路熵编码器 (例如, 对于输入二进制数 (bin) 具有固定概率估计的二进制算术编码器) 对第三和第四扫描过程期间生成的语法元素进行编码。

[0145] 可以执行上下文建模以确定用于某些类型的变换系数语法元素的bin的上下文模型。在实施例中, 上下文模型可以根据局部模板和当前系数的对角线位置并可能结合其它因素来确定。

[0146] 图12示出了用于当前系数 (1220) 的上下文选择的局部模板 (1230) 的示例。为了利用变换系数之间的相关性, 可以在当前系数的上下文选择中使用由局部模板 (1230) 覆盖的先前已编码的系数 (1220)。局部模板 (1230) 可以覆盖系数块 (1210) 中的当前系数 (1220) 的一组相邻位置或系数。系数块 (1210) 可以具有  $8 \times 8$  系数的大小。将系数块 (1210) 划分成4个子块, 每个子块具有  $4 \times 4$  系数的大小。在图12的示例中, 局部模板 (1230) 被定义为在当前系数 (1220) 的右下侧覆盖5个系数级别的5位置模板。当逆对角线扫描顺序用于系数块 (1210) 内的系数的多个扫描过程时, 在当前系数 (1220) 之前处理局部模板 (1230) 内的相邻系数。

[0147] 在上下文建模期间, 可以使用局部模板 (1230) 内的系数级别的信息来确定上下文模型。为此, 在一些实施例中定义了称为模板幅度的度量, 以测量或指示局部模板 (1230) 内的变换系数或变换系数级别的幅度。然后, 可以将模板幅度用作选择上下文模型的基础。

[0148] 在一个示例中, 模板幅度被定义为局部模板 (1230) 内的部分重建的绝对变换系数级别的和 (例如, `sumAbs1`)。部分重建的绝对变换系数级别 `absLevel1[x][y]` 可以根据相应变换系数的语法元素 `sig_coeff_flag`、`par_level_flag`、`rem_abs_gt1_flag` 的bin来确定。在熵编码器或熵解码器中执行了子块的变换系数的第一个扫描过程之后, 获得三种类型的语法元素。在实施例中, 在第一个扫描过程之后, 位置 (x, y) 处的部分重建的绝对变换系数

级别absLevel1[x][y]可以根据以下确定:

[0149]  $\text{absLevel1}[x][y] = \text{sig\_coeff\_flag}[x][y] + \text{par\_level\_flag}[x][y] + 2 * \text{rem\_abs\_gt1\_flag}[x][y]$ , 其中x和y是相对于系数块(1210)的左上角的坐标,absLevel1[x][y]表示位置(x,y)处的部分重建的绝对变换系数级别。

[0150] d表示当前系数的对角线位置,其中d是x和y的和,numSig表示局部模板(1230)中非零系数的数量。例如,sumAbs1表示局部模板(1230)覆盖的系数的部分重建的绝对级别absLevel1[x][y]的和。

[0151] 当对当前系数(1220)的sig\_coeff\_flag进行编码时,根据sumAbs1和当前系数(1220)的对角线位置d来选择上下文模型索引。d是x和y的和。sumAbs1表示局部模板(1230)覆盖的系数的部分重建的绝对级别absLevel1[x][y]的和。在实施例中,对于亮度分量,上下文模型索引根据以下确定:

[0152]  $\text{ctxSig} = 18 * \max(0, \text{state} - 1) + \min(\text{sumAbs1}, 5) + (d < 2 ? 12 : (d < 5 ? 6 : 0))$  等式1,其中,ctxSig表示重要性标志语法元素的上下文索引,“state”指定从属量化方案的标量量化器(scalar quantizer)的状态。在示例中,使用状态转换过程导出“state”。等式(1)等同于以下内容:

[0153]  $\text{ctxIdBase} = 18 * \max(0, \text{state} - 1) + (d < 2 ? 12 : (d < 5 ? 6 : 0))$  等式2,

[0154]  $\text{ctxSig} = \text{ctxIdSigTable}[\min(\text{sumAbs1}, 5)] + \text{ctxIdBase}$  等式3,其中,

ctxIdBase表示上下文索引基础。上下文索引基础可以基于状态(state)和对角线位置d来确定。例如,state可以具有0、1、2或3的值,并且因此max(0,state-1)可以具有三个可能值0、1或2中的一个。例如,(d<2?12:(d<5?6:0))可以取值12、6或0,对应于d的不同范围:d<2、2<=d<5或5<=d。ctxIdSigTable[]可以表示数组数据结构,并且存储相对于ctxIdBase的重要性标志的上下文索引偏移。例如,对于不同的sumAbs1值,min(sumAbs1,5)将sumAbs1值裁剪(clip)为小于或等于5。然后,将裁剪后的值映射到上下文索引偏移。例如,在ctxIdSigTable[0~5]={0,1,2,3,4,5}的定义下,裁剪后的值0、1、2、3、4或5分别映射到0、1、2、3、4或5。

[0155] 在实施例中,对于色度分量,上下文索引可以根据以下确定:

[0156]  $\text{ctxSig} = 12 * \max(0, \text{state} - 1) + \min(\text{sumAbs1}, 5) + (d < 2 ? 6 : 0)$ , 等式4,等式4等同于以下的等式5和6。

[0157]  $\text{ctxIdBase} = 12 * \max(0, \text{state} - 1) + (d < 2 ? 6 : 0)$ , 等式5,

[0158]  $\text{ctxSig} = \text{ctxIdSigTable}[\min(\text{sumAbs1}, 5)] + \text{ctxIdBase}$ 。 等式6,其中,state指定在启用从属量化时使用的标量量化器,并且state是使用状态转换过程导出的。表ctxIdSigTable存储上下文模型索引偏移,ctxIdSigTable[0~5]={0,1,2,3,4,5}。

[0159] 当对当前系数(1220)的par\_level\_flag进行编码时,可以根据sumAbs1、numSig和对角线位置d选择上下文索引。numSig表示局部模板(1230)中非零系数的数量。例如,对于亮度分量,上下文索引可以根据以下确定:

[0160]  $\text{ctxPar} = 1 + \min(\text{sumAbs1} - \text{numSig}, 4) + (d == 0 ? 15 : (d < 3 ? 10 : (d < 10 ? 5 : 0)))$  等式7,等式7等同于以下内容:

[0161]  $\text{ctxIdBase} = (d == 0 ? 15 : (d < 3 ? 10 : (d < 10 ? 5 : 0)))$ , 等式

8,

[0162]  $ctxPar = 1 + ctxIdTable[\min(\text{sumAbs1} - \text{numSig}, 4)] + ctxIdBase$ , 等式9,其中,ctxPar表示奇偶校验标志的上下文索引,ctxIdTable[]表示数组数据结构,并存储相对于相应ctxIdBase的上下文索引偏移。例如,ctxIdTable[0~4]={0,1,2,3,4}。

[0163] 对于色度,上下文索引可以根据以下确定:

[0164]  $ctxPar = 1 + \min(\text{sumAbs1} - \text{numSig}, 4) + (d == 0 ? 5 : 0)$ , 等式10,等式10等同于以下内容:

[0165]  $ctxIdBase = (d == 0 ? 5 : 0)$ , 等式11,

[0166]  $ctxPar = 1 + ctxIdTable[\min(\text{sumAbs1} - \text{numSig}, 4)] + ctxIdBase$ 。等式12,其中,表ctxIdTable存储上下文模型索引偏移,并且在示例中,ctxIdTable[0~4]={0,1,2,3,4}。

[0167] 当对当前系数(1220)的rem\_abs\_gt1\_flag和rem\_abs\_gt2\_flag进行编码时,可以用与par\_level\_flag相同的方式确定上下文模型索引:

[0168]  $ctxGt1 = ctxPar$ ,

[0169]  $ctxGt2 = ctxPar$ ,

[0170] 其中,ctxGt1和ctxGt2分别表示大于1标志和大于2标志的上下文索引。

[0171] 注意,不同类型的语法元素可以使用不同组的上下文模型,例如rem\_abs\_gt1\_flag和rem\_abs\_gt2\_flag。因此,即使ctxGt1的值等于ctxGt2的值,rem\_abs\_gt1\_flag使用的上下文模型也不同于rem\_abs\_gt2\_flag使用的上下文模型。

[0172] 主变换(例如正主变换(forward primary transform)或逆主变换(inverse primary transform))可以利用如下所述的归零(zero-out)方法或归零方案。在一些示例中,例如在VVC中,对于64点(或64长度)DCT-2,仅计算前32个系数,其余系数设置为0。因此,对于使用DCT-2变换编码的M×N块,保留或计算左上 $\min(M, 32) \times \min(N, 32)$ 低频系数。其余系数可以设置为0并且不用信号通知。在示例中,不计算其余系数。可以通过将系数块大小设置为 $\min(M, 32) \times \min(N, 32)$ 来执行系数块的熵编码,从而将M×N块的系数编码视为 $\min(M, 32) \times \min(N, 32)$ 系数块。

[0173] 在使用MTS的一些示例中,对于32点DST-7或DCT-8,仅计算前16个系数,其余系数设置为0。因此,对于使用DST-7或DCT-8变换编码的M×N块,保留左上 $\min(M, 16) \times \min(N, 16)$ 低频系数。其余系数可以设置为0并且不用信号通知。然而,与应用64点归零DCT-2时使用的系数编码方案不同,对于32点MTS,即使M或N大于16,仍对整个M×N块执行系数编码。然而,当系数群组(CG)在左上16×16低频区域之外(即,系数群组在归零区域中)时,不发信号通知指示该系数群组是否具有非零系数的标志(例如,coded\_sub\_block\_flag)。归零区域是指系数块中系数为零的区域,因此归零区域中的系数为零。下面描述残差编码语法的示例,其由图13中使用框1301突出显示的文本来指示。

[0174] 在实施例中,可以在编码器侧的正核心变换与量化(例如,系数的量化)之间以及在解码器侧的去量化(例如,量化系数的去量化)与逆核心变换之间使用模式相关的不可分离二次变换(NSST)。例如,为了保持低复杂度,在主变换(或核心变换)之后将NSST应用于低频系数。当变换系数块的宽度(W)和高度(H)均大于或等于8时,将8×8NSST应用于变换系数块的左上8×8区域。否则,当变换系数块的宽度W或高度H为4时,应用4×4NSST,并且在变换

系数块的左上 $\min(8,W) \times \min(8,H)$ 区域执行 $4 \times 4$ NSST。上述变换选择方法适用于亮度分量和色度分量。

[0175] 下面以 $4 \times 4$ 输入块为例描述NSST的矩阵乘法实施方式。 $4 \times 4$ 输入块 $X$ 在等式13中写为

$$[0176] \quad X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix} \quad \text{等式 13.}$$

[0177] 在等式14中,输入块 $X$ 可以表示为矢量 $\vec{X}$ ,其中:

$$[0178] \quad \vec{X} = [X_{00} X_{01} X_{02} X_{03} X_{10} X_{11} X_{12} X_{13} X_{20} X_{21} X_{22} X_{23} X_{30} X_{31} X_{32} X_{33}]^T \quad \text{等式 14,}$$

[0179] 不可分离变换被计算为 $\vec{F} = T \cdot \vec{X}$ ,其中, $\vec{F}$ 表示变换系数矢量, $T$ 为 $16 \times 16$ 变换矩阵。随后,使用用于输入块 $X$ 的扫描顺序(例如,水平扫描顺序、垂直扫描顺序或对角线扫描顺序),将 $16 \times 1$ 变换系数矢量 $\vec{F}$ 重新组织为 $4 \times 4$ 块。具有较小索引的系数可以以较小的扫描索引放置在 $4 \times 4$ 系数块中。

[0180] 可以通过帧内预测模式来选择二次变换的变换核。对于每个帧内预测模式,可以定义一组二次变换核,并且可以由可在比特流中用信号通知的索引来指示选择和/或由其它语法元素(例如,MTS索引)来指示该选择。

[0181] 为了降低二次变换中存储和计算系数的复杂度,可以应用基于归零方案的二次变换。当二次变换包括归零方案时,二次变换仅计算 $M \times N$ 块的前 $K$ 个系数,其中 $K$ 小于 $M \times N$ 。其余 $(M \times N - K)$ 个系数可以设置为0。例如,不计算其余 $(M \times N - K)$ 个系数。

[0182] 在一些实施例中,采用子块变换(SBT),也称为空间变化变换(SVT)。SBT可以应用于帧间预测残差。在一些示例中,残差块包括在编码块中并且小于编码块。因此,SBT中的变换大小小于编码块大小。对于未被残差块覆盖的区域,可以假定为零残差,因此不进行变换处理。

[0183] 图14A-14D示出了在SBT中支持的子块类型(SVT-H、SVT-V)(例如,水平或垂直划分的)、大小和位置(例如,左半部分、左四分之一、右半部分、右四分之一、上半部分、上四分之一、下半部分、下四分之一)。用字母“A”标记的阴影区域是具有变换的残差块,并且假定其它区域为无变换的零残差。

[0184] 作为示例,图15A-15F示出了当使用SBT时对视频编解码标准(例如,VVC)的规范文本的改变。在框(2101)至(2108)中示出了包括添加的文本在内的改变。如图所示,可以用信号通知附加语法元素(例如,附加的开销位`cu_sbt_flag`、`cu_sbt_quad_flag`、`cu_sbt_horizontal_flag`和`cu_sbt_pos_flag`),以分别指示子块类型(水平或垂直)、大小(一半或四分之一)和位置(左、右、上或下)。

[0185] 下面给出序列参数集Rbsp语义的示例。`sps_sbt_enabled_flag`等于0指定对帧间预测CU禁用子块变换。`sps_sbt_enabled_flag`等于1指定对帧间预测CU启用子块变换。

[0186] 下面给出常规切片头语义的示例。`slice_sbt_max_size_64_flag`等于0指定允许子块变换的最大CU宽度和高度为32。`slice_sbt_max_size_64_flag`等于1指定允许子块变换的最大CU宽度和高度为64。`maxSbtSize`可以确定为:`maxSbtSize=slice_sbt_max_size_`

64\_flag?64:32。

[0187] 下面给出编码单元语义的示例。cu\_sbt\_flag[x0][y0]等于1指定对于当前编码单元,使用子块变换。cu\_sbt\_flag[x0][y0]等于0指定对于当前编码单元,不使用子块变换。当cu\_sbt\_flag[x0][y0]不存在时,其值推断为等于0。当使用子块变换时,将编码单元平铺成两个变换单元,一个变换单元具有残差,另一个变换单元没有残差。

[0188] 在一些示例中,cu\_sbt\_quad\_flag[x0][y0]等于1指定对于当前编码单元,子块变换包括当前编码单元的1/4大小的变换单元。在一些示例中,cu\_sbt\_quad\_flag[x0][y0]等于0指定对于当前编码单元,子块变换包括当前编码单元的1/2大小的变换单元。当cu\_sbt\_quad\_flag[x0][y0]不存在时,其值推断为等于0。

[0189] 在一些示例中,cu\_sbt\_horizontal\_flag[x0][y0]等于1指定通过水平分割将当前编码单元平铺成2个变换单元。cu\_sbt\_horizontal\_flag[x0][y0]等于0指定通过垂直分割将当前编码单元平铺成2个变换单元。

[0190] 在一些示例中,当cu\_sbt\_horizontal\_flag[x0][y0]不存在时,其值被导出如下:如果cu\_sbt\_quad\_flag[x0][y0]等于1,则cu\_sbt\_horizontal\_flag[x0][y0]被设置为等于allowSbtHoriQuad。否则(cu\_sbt\_quad\_flag[x0][y0]等于0),则cu\_sbt\_horizontal\_flag[x0][y0]被设置为等于allowSbtHoriHalf。

[0191] 在一些示例中,cu\_sbt\_pos\_flag[x0][y0]等于1指定比特流中不存在当前编码单元中的第一变换单元的tu\_cbf\_luma、tu\_cbf\_cb和tu\_cbf\_cr。cu\_sbt\_pos\_flag[x0][y0]等于0指定比特流中不存在当前编码单元中的第二变换单元的tu\_cbf\_luma、tu\_cbf\_cb和tu\_cbf\_cr。

[0192] 下文描述用于缩放的变换系数的变换过程的示例。

[0193] 该过程的输入是:指定当前亮度变换块的左上样本相对于当前图片的左上亮度样本的亮度位置(xTbY,yTbY)、指定当前变换块的宽度的变量nTbW、指定当前变换块的高度的变量nTbH、指定当前块的颜色分量的变量cIdx、缩放的变换系数的(nTbW)×(nTbH)数组d[x][y],其中x=0..nTbW-1,y=0..nTbH-1。

[0194] 该过程的输出是残差样本的(nTbW)×(nTbH)数组r[x][y],其中x=0..nTbW-1,y=0..nTbH-1。

[0195] 如果cu\_sbt\_flag[xTbY][yTbY]等于1,则根据cu\_sbt\_horizontal\_flag[xTbY][yTbY]和cu\_sbt\_pos\_flag[xTbY][yTbY]在表3中(图15F中示出的)导出指定水平变换核的变量trTypeHor和指定垂直变换核的变量trTypeVer。

[0196] 否则(cu\_sbt\_flag[xTbY][yTbY]等于0),则根据mts\_idx[xTbY][yTbY]和CuPredMode[xTbY][yTbY]在表4中(图15F中示出的)导出指定水平变换核的变量trTypeHor和指定垂直变换核的变量trTypeVer。

[0197] 残差样本的(nTbW)×(nTbH)数组r导出如下:

[0198] 通过将变换块的高度nTbH、y=0..nTbH-1的列表d[x][y]、和设置为等于trTypeVer的变换类型变量trType作为输入,为每一列x=0..nTbW-1调用一维变换过程,从而将缩放的变换系数d[x][y](其中x=0..nTbW-1,y=0..nTbH-1)的每一(垂直)列变换为e[x][y](其中x=0..nTbW-1,y=0..nTbH-1),并且输出为y=0..nTbH-1的列表e[x][y]。

[0199] x=0..nTbW-1,y=0..nTbH-1的中间样本值g[x][y]导出如下:

[0200]  $g[x][y]=\text{Clip3}(\text{CoeffMin},\text{CoeffMax},(e[x][y]+256)\gg 9)$ 。

[0201] 通过将变换块的宽度 $nTbW$ 、 $x=0..nTbW-1$ 的列表 $g[x][y]$ 、和设置为等于 $trTypeHor$ 的变换类型变量 $trType$ 作为输入,为每一行 $y=0..nTbH-1$ 调用一维变换过程,从而将所得数组 $g[x][y]$ (其中 $x=0..nTbW-1,y=0..nTbH-1$ )的每一(水平)行变换为 $r[x][y]$ (其中 $x=0..nTbW-1,y=0..nTbH-1$ ),并且输出为 $x=0..nTbW-1$ 的列表 $r[x][y]$ 。

[0202] 在一些示例中,对于具有归零的64点DCT-2和具有归零的32点MTS,应用两种不同的系数编码方案,因此,系数编码不是统一的设计。系数编码可以指变换系数的熵编码。

[0203] 对于应用于具有归零的32点MTS的系数编码,尽管假定变换系数为零,但是可以扫描归零区域的变换系数。如果编码器的设计不完善但是编码器可以在归零区域计算非零系数,因此,左上非零区域的边界处的CG仍然可以访问非零区域,以导出用于对变换系数进行熵编码的上下文值。编码器和解码器可以获得不同的上下文值(例如,编码器获得非零值,解码器获得零值),因此产生了对于解码器来说不可预测的行为,例如崩溃或失配。

[0204] 在一些示例中,应用了可以使用32点DST-7/DCT-8的SBT,因此,还需要适当地设计应用SBT时的变换归零。

[0205] 二次变换可以包括归零方法,因此需要适当地设计应用二次变换时的变换归零。

[0206] 本文描述的实施例可以单独使用或以任何顺序组合使用。此外,该实施例可以通过编码器、解码器等中的处理电路(例如,一个或多个处理器或者一个或多个集成电路)来实现。在一个示例中,该一个或多个处理器可以执行存储在非易失性计算机可读介质中的程序。

[0207] 在本公开中,适用于MTS候选的DST-7的实施例可以适用于DST-4,反之亦然。类似地,适用于MTS候选的DCT-8的实施例也适用于DCT-4,反之亦然。

[0208] 在本公开中,二次变换可以指NSST或简化二次变换(RST),该简化二次变换是NSST的替代设计。适用于NSST的实施例可以适用于RST。

[0209] 在解码器侧,可以将包括变换系数的系数块解码为具有残差数据的残差块,因此TU可以指系数块或残差块。在本公开中,可以将系数单元(或系数集)定义为一组变换系数(也称为系数),该组变换系数包括TU或系数块内的所有系数或一部分系数。例如,系数单元可以指一个子块,该子块是包括系数块内的所有非零系数的最小子块。例如,对于仅使用DCT-2编码的 $64\times 64$ TU,仅保留左上 $32\times 32$ 区域中的低频系数,并将 $64\times 64$ TU中的其余系数设置为0,因此,系数单元是 $64\times 64$ TU中的左上 $32\times 32$ 区域。在另一示例中,对于使用DST-7或DCT-8编码的 $32\times 32$ TU,仅保留左上 $16\times 16$ 区域中的低频系数,并将 $32\times 32$ TU中的其余系数设置为0,因此,系数单元是 $32\times 32$ TU中的左上 $16\times 16$ 区域。

[0210] TU可以具有多个系数单元,并且该多个系数单元包括TU内的所有非零系数(即,TU内的所有非零系数包括在该多个系数单元中)。因此,TU中的在多个系数单元之外的系数为零。

[0211] 图16A-16D示出了根据本公开实施例的系数单元的示例。参照图16A, $16\times 16$ 系数块(1610)包括阴影区域0、1A、2A和3A,以及白色区域5。阴影区域0、1A、2A和3A可以包括非零变换系数,白色区域5不包括非零变换系数(即,白色区域5中的变换系数为零)。参照图16A,在系数块(1610)中存在4个系数单元(例如,阴影区域0、1A、2A和3A)。

[0212] 参照图16B,系数块(1610)可以被划分为阴影区域0和2A、阴影区域1B和白色区域

5。阴影区域0、1B和2A可以包括非零变换系数,白色区域5不包括非零变换系数。参照图16B,在系数块(1610)中存在3个系数单元(例如,阴影区域0、1B和2A)。参照图16A和图16B,阴影区域1B包括阴影区域1A和3A。

[0213] 参照图16C,系数块(1610)可以被划分为阴影区域0和1A、阴影区域2C和白色区域5。阴影区域0、1A和2C可以包括非零变换系数,白色区域5不包括非零变换系数。参照图16C,在系数块(1610)中存在3个系数单元(例如,阴影区域0、1A和2C)。参照图16A和图16C,阴影区域2C包括阴影区域2A和3A。

[0214] 参照图16D,系数块(1610)可以被划分为阴影区域0、阴影区域1D和白色区域5。阴影区域0和1D可以包括非零变换系数,白色区域5不包括非零变换系数。参照图16D,在系数块(1610)中存在2个系数单元(例如,阴影区域0和1D)。参照图16A和16D,阴影区域1D包括阴影区域1A、2A和3A。

[0215] 通常,系数单元可以具有任何合适的形状或尺寸。参照图16A-16D,系数单元可以具有正方形形状(例如,系数单元1A、2A、3A)、矩形形状(例如,系数单元1B和2C)、不规则形状(例如,系数单元1D的“L”形)。可以相应地修改系数编码,使得在诸如白色区域5的归零区域中的系数不在比特流中用信号通知。

[0216] 可以将系数块划分为不同数量的系数单元。例如,可以将系数块(1610)划分为4、3、3和2个系数单元,分别如图16A-16D所示。

[0217] 根据本公开的实施例,可以以系数单元为单位处理TU的系数编码,其中每个系数单元被独立地编码。因此,TU中一个系数单元的系数编码不需要访问TU中另一个系数单元的编码信息。

[0218] 可以使用各种处理顺序来处理系数块中的多个系数单元。参照图16D,处理电路可以同步开始处理系数单元0和1D。在示例中,处理电路可以在开始处理系数单元1D之前开始处理系数单元0。在示例中,处理电路可以在开始处理系数单元0之前开始处理系数单元1D。

[0219] 在一些示例中,标志(即,系数单元标志)可以用于指示系数单元是否至少包括非零系数。例如,4个标志可以分别用于图16A中的系数单元0和1A-3A。类似地,3个标志可以分别用于图16B中的系数单元0、1B和2A;3个标志可以分别用于图16C中的系数单元0、1A和2C;2个标志可以分别用于图16D中的系数单元0和1D。

[0220] 在一些示例中,对于每个系数单元,可以用信号通知系数单元标志。在示例中,当TU仅具有一个系数单元时,可以从TU的系数块标志(CBF)推断出该一个系数单元的系数单元标志,因此不用信号通知系数单元标志。TU的CBF可以指示TU是否具有至少一个非零系数。

[0221] 在示例中,TU被划分成多个系数单元,并且TU具有非零系数。如果在处理顺序中仅最后一个系数单元具有非零系数,则该最后一个系数单元的系数单元标志指示该最后一个系数单元具有非零系数,并且该系数单元标志被推断出而无需用信号通知。

[0222] 在实施例中,可以以系数群组(CG)为单位处理系数单元的系数编码。CG可以包括系数单元中的16个系数。CG可以具有任何合适的形状,例如正方形、矩形等。例如,CG可以是系数单元中的 $4 \times 4$ 子块、 $2 \times 8$ 子块或 $8 \times 2$ 子块。

[0223] 在示例中,系数单元包括多个CG并且具有非零系数。如果在处理顺序中仅最后一个CG具有非零系数,则可以推断出而无需用信号通知该最后一个CG的标志(例如,coded\_

sub\_block\_flag, 指示该最后一个CG是否具有非零系数), 并且该最后一个CG的标志指示该最后一个CG具有至少一个非零系数。

[0224] 在实施例中, 对于未经二次变换编码的TU, TU仅具有一个系数单元。

[0225] 在示例中, 如果将包括(或使用)归零方法的主变换、SBT、或包括归零方法的主变换和SBT的组合应用于TU, 则可以从包括归零方法的主变换和/或SBT推断出系数单元的大小和位置, 使得在包括归零方法的主变换和/或SBT之后, 系数单元包括非零区域。非零区域包括至少一个非零系数。

[0226] 在示例中, 如果例如由于如上所述的包括归零方法的主变换、SBT等, TU不具有预先已知的归零区域, 则可以推断出系数单元的大小和位置与TU的大小和位置相同。在示例中, 系数单元是TU。

[0227] 在一些实施例中, 将主变换应用于 $M \times N$ 系数块并且主变换使用归零方法, 其中主变换可以是非DCT-2变换或DCT-2变换和非DCT-2变换的组合。主变换可以包括水平M点变换和垂直N点变换。因此, 当X是为水平M点变换保留或计算的系数的数量并且Y是为垂直N点变换保留或计算的系数的数量时, 比率 $X/M$ 小于1和/或比率 $Y/N$ 小于1。根据本公开的方面,  $M \times N$ 系数块的熵编码可以通过将熵编码块的大小设置为 $\min(M, X) \times \min(N, Y)$ 来实现。因此, 当对 $M \times N$ 系数块进行熵编码时, 将 $M \times N$ 系数块视为 $\min(M, X) \times \min(N, Y)$ 区域, 例如, 仅对左上 $\min(M, X) \times \min(N, Y)$ 区域中的变换系数进行熵编码, 不对其余系数进行熵编码(例如, 将其余系数设置为零)。此外, 在示例中, 当对左上 $\min(M, X) \times \min(N, Y)$ 区域进行熵编码时, 不访问左上 $\min(M, X) \times \min(N, Y)$ 区域之外的其余系数。

[0228] 在一些示例中, 非DCT-2变换是MTS中使用的32点DST-7/DCT-8、SBT中使用的32点DST-7/DCT-8、和/或隐式MTS中使用的32点DST-7/DCT-8。当X和Y是16时, 熵编码块的大小是 $16 \times 16$ , 并且仅对左上 $16 \times 16$ 区域进行熵编码。此外, 在示例中, 当对左上 $16 \times 16$ 区域进行熵编码时, 不访问左上 $16 \times 16$ 区域之外的其余系数。

[0229] 如上所述, 可以将主变换(例如, 非DCT-2变换、或DCT-2变换和非DCT-2变换的组合)应用于 $M \times N$ 系数块并且主变换使用归零方法。比率 $X/M$ 小于1和/或比率 $Y/N$ 小于1。可以将 $M \times N$ 系数块中左上 $\min(M, X) \times \min(N, Y)$ 区域之外的区域中的系数视为零, 并且该区域可以称为归零区域。根据本公开的方面, 可以在整个 $M \times N$ 系数块上实现 $M \times N$ 系数块的熵编码。然而, 当访问位于归零区域中的变换系数的语法元素以导出上下文值(用于对例如当前系数进行熵编码)时, 可以使用变换系数的默认值。变换系数的语法元素也可以称为系数相关语法元素。

[0230] 在示例中, 当对当前CG的CG标志(例如, coded\_sub\_block\_flag)进行熵编码时, 可以使用相应的一个或多个相邻CG的一个或多个相邻CG标志来导出用于对当前CG的CG标志进行熵编码的上下文值。然而, 如果该一个或多个相邻CG中的一个位于归零区域中, 则可以将默认值(例如, 0)用于该一个或多个相邻CG中的一个, 以导出上下文值。

[0231] 在示例中, 当对当前变换系数的语法元素(例如, 系数相关标志, 例如sig\_coeff\_flag、par\_level\_flag、rem\_abs\_gt1\_flag、rem\_abs\_gt2\_flag等)进行熵编码时, 可以使用相邻系数的语法元素来导出当前变换系数的上下文值。当相邻系数位于归零区域中时, 可以使用默认值(例如, 0)代替。

[0232] 图17示出了包括第一区域(1720)和第二区域(1730)的 $32 \times 32$ 系数块(1710)。第一

区域(1720)是系数块(1710)中的左上 $16 \times 16$ 区域,第二区域(1730)在第一区域(1720)之外。第一区域(1720)包括CG A-C,第二区域(1730)包括CG AB、AR、BB和CR。在示例中,熵编码顺序是从系数块(1710)的右下角开始到系数块(1710)的左上角结束的反向对角线顺序。因此,CG AB和AR在CG A之前进行熵编码,CG BB在CG B之前进行熵编码,CG CR在CG C之前进行熵编码。

[0233] 在示例中,仅保留第一区域(1720)中的变换系数,并且将第二区域(1730)中的其余系数设置为0,因此第二区域(1730)也称为归零区域。为了对CG A的第一CG标志(例如,coded\_sub\_block\_flag)进行熵编码,可以访问CG AR(右CG)和CG AB(下CG),以导出第一CG标志的上下文值。由于CG AR和AB位于归零区域(1730)中,因此可以将默认值(例如,0)分别分配给CG AR和AB的CG标志。为了对CG B的第二CG标志(例如coded\_sub\_block\_flag)进行熵编码,将访问CG BB以导出上下文值。由于CG BB位于归零区域(1730)中,因此可以将默认值(例如,0)分配给CG BB的CG标志。为了对CG C的第三CG标志(例如,coded\_sub\_block\_flag)进行熵编码,将访问CG CR以导出上下文值。由于CG CR位于归零区域(1730)中,因此可以将默认值(例如,0)分配给CG CR的CG标志。

[0234] 在一些实施例中,在 $X \times Y$ 系数块(1810)的左上 $M \times N$ 区域(1820)上应用包括归零方法的二次变换,如图18所示。例如,保留前16个系数,并且将左上 $M \times N$ 区域(1820)中的其余( $M \times N - 16$ )个系数设置为0。在示例中, $X$ 和 $Y$ 是32, $M$ 和 $N$ 是16,因此保留 $4 \times 4$ 区域0中的前16个系数,并且将区域2中的其余( $16 \times 16 - 16$ )个系数设置为0。因此,区域2被称为归零区域或零区域。参照图18, $X \times Y$ 系数块(1810)包括三个区域,区域0、区域1和区域2。区域2是由于二次变换中的归零方法而产生的归零区域。区域0和/或区域1可以具有非零变换系数。二次变换不适用于区域1,即,区域1未被二次变换处理。

[0235] 可以对整个 $X \times Y$ 系数块(1810)执行 $X \times Y$ 系数块(1810)的熵编码。然而,当访问系数相关语法元素(即,系数的语法元素,其中系数位于归零区域2中)以导出用于熵编码的上下文值时,可以使用该系数的默认值。

[0236] 在实施例中,当对当前CG的CG标志(例如,coded\_sub\_block\_flag)进行熵编码时,可以使用相应的一个或多个相邻CG的一个或多个相邻CG标志来导出CG标志的上下文值。然而,如果该一个或多个相邻CG中的一个完全位于归零区域2(其是由于使用归零方法的二次变换而形成的)中,则可以使用默认值(例如,0)代替。参照图18,区域0包括CG A、B和C,区域2(例如,归零区域)包括CG AB、AR和BB。CG AB和AR完全位于归零区域2中,因此可以使用默认值(例如,0)代替CG AB和AR的CG标志来导出第一CG标志的上下文值。CG BB完全位于归零区域2中,因此可以使用默认值(例如,0)代替CG BB的CG标志来导出第二CG标志的上下文值。

[0237] 在实施例中,当对当前CG的CG标志(例如,coded\_sub\_block\_flag)进行熵编码时,可以使用相应的一个或多个相邻CG的一个或多个相邻CG标志来导出CG标志的上下文值。然而,如果该一个或多个相邻CG中的一个(例如,图18中的CG CR)部分地位于归零区域2中,则可以使用该一个或多个相邻CG中的一个(例如,图18中的CG CR)中的非零系数来导出上下文值。

[0238] 在一个实施例中,当对当前变换系数的语法元素(例如,系数相关标志,例如sig\_coeff\_flag、par\_level\_flag、rem\_abs\_gt1\_flag、rem\_abs\_gt2\_flag等)进行熵编码时,不

用信号通知语法元素。当当前系数位于归零区域(例如归零区域2)中时,当前系数可以推断为指示当前系数为0的默认值。

[0239] 在实施例中,当对当前变换系数的语法元素(例如,系数相关标志,例如sig\_coeff\_flag、par\_level\_flag、rem\_abs\_gt1\_flag、rem\_abs\_gt2\_flag等)进行熵编码时,可以使用一个或多个相邻系数来导出语法元素的上下文值。如果该一个或多个相邻系数中的一个位于零区域2中,则可以使用默认值(例如,0)代替。

[0240] 参照图19,在实施例中,系数块(1910)包括第一区域(1920)和在第一区域(1920)之外的第二区域(1930)。当将二次变换(例如,NSST或SBT)应用于第一区域(1920)并且二次变换使用仅保留部分系数(例如,在左上区域0中)的归零方法时,则也可以将未应用二次变换的第二区域(1930)视为归零区域。在图19所示的示例中,由于在二次变换中使用归零方法,区域0和第二区域(1930)之间的区域1是归零区域。因此,组合的归零区域包括区域1和第二区域(1930)。在示例中,系数块(1910)仅包括一个系数单元(例如,区域0),因此仅对系数块(1910)中的系数单元(例如,区域0)进行熵编码。

[0241] 在实施例中,可以基于包括归零方法的二次变换来推断系数块(1910)中的系数单元的大小和位置。在示例中,当二次变换是 $8 \times 8$ 二次变换时,系数单元是系数块(1910)的左上 $8 \times 8$ 区域(1920)。在示例中,当二次变换是第一区域(1920)上的包括归零方法的 $8 \times 8$ 二次变换时,系数单元是第一区域(1920)的左上 $4 \times 4$ 区域0,其中该归零方法将前16个系数保留在区域0中。

[0242] 在示例中,二次变换是应用于第一区域(1920)(例如,系数块(1910)中的左上 $8 \times 8$ 区域)的 $8 \times 8$ 二次变换,第二区域(1930)在左上 $8 \times 8$ 区域(1920)之外。当在二次变换中仅保留16个系数时,组合的归零区域在左上 $4 \times 4$ 区域0之外。

[0243] 在示例中,系数块包括 $4 \times 4$ 的第一区域和在第一区域之外的第二区域。当将二次变换应用于第一区域并且二次变换使用仅保留部分系数的归零方法时,则可以将未应用二次变换的第二区域视为归零区域。

[0244] 图20示出了根据本公开实施例的概述过程(2000)的流程图。过程(2000)可用于重建以帧内模式编码的块,从而为正在重建的块生成预测块。在一些示例中,过程(2000)可用于重建以帧间模式编码的块。在各种实施例中,过程(2000)由诸如终端装置(310)、(320)、(330)和(340)中的处理电路之类的处理电路、执行视频编码器(403)的功能的处理电路、执行视频解码器(410)的功能的处理电路、执行视频解码器(510)的功能的处理电路、执行视频编码器(603)的功能的处理电路等来执行。在一些实施例中,过程(2000)以软件指令实现,因此,当处理电路执行该软件指令时,处理电路执行过程(2000)。该过程从(S2001)开始,并且进行到(S2010)。

[0245] 在(S2010),可以从已编码视频比特流中解码变换块(TB)(例如系数块)的编码信息。所述编码信息可以指示所述TB的应用二次变换的区域,其中所述区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数。在示例中,所述二次变换使用归零方法,因此,所述第二子区域中的变换系数不通过所述二次变换计算并设置为零。

[0246] 在(S2020),对于所述TB中的变换系数,可以确定用于确定所述变换系数的相邻变换系数是否位于所述第二子区域中。当确定所述相邻变换系数位于所述第二子区域中时,

过程(2000)进行到(S2030)。否则,过程(2000)进行到(S2040)。

[0247] 在(S2030),可以根据所述相邻变换系数的默认值确定所述变换系数。在一些示例中,可以根据所述相邻变换系数的默认值(例如,0)确定所述变换系数的一个或多个语法元素。所述语法元素可以包括系数相关标志,例如sig\_coeff\_flag、a par\_level\_flag、a rem\_abs\_gt1\_flag、a rem\_abs\_gt2\_flag,分别指示所述变换系数是否是非零变换系数、所述变换系数的奇偶性、所述变换系数是否大于2、以及所述变换系数是否大于4。

[0248] 在(S2040),可以根据所述相邻变换系数的默认值确定所述变换系数。在一些示例中,可以确定所述变换系数的一个或多个语法元素。

[0249] 在(S2050),基于所述TB中的用于样本的所述变换系数重建所述样本。

[0250] 过程(2000)可以适当地修改。例如,可以修改,省略或组合一个或多个步骤。还可以修改执行过程(2000)的顺序。

[0251] 还可以增加其它步骤。例如,所述TB中的变换系数可以是第一CG中的多个变换系数中的一个,并且所述第一CG的第一CG标志指示所述多个变换系数中的至少一个是否是非零变换系数。包括变换系数的第二CG被预先熵解码,并且与所述第一CG相邻。过程(2000)可以包括确定所述第二CG的位置的步骤。当确定所述第二CG位于所述第二子区域中时,过程(2000)可以包括基于所述第二CG的默认值确定所述第一CG标志的步骤。可选地,当所述第二CG的一部分位于所述第二子区域中并且所述第二CG的另一部分包括至少一个非零系数时,过程(2000)可以包括基于所述至少一个非零系数确定所述第一CG标志的步骤。

[0252] 在一些示例中,其它步骤可以包括确定所述变换系数是否位于所述第二子区域中。当确定所述变换系数位于所述第二子区域中时,确定所述变换系数不用信号通知并且为零。

[0253] 图21示出了根据本公开实施例的概述过程(2100)的流程图。过程(2100)可用于重建以帧内模式编码的块,从而为正在重建的块生成预测块。在一些示例中,过程(2100)可用于重建以帧间模式编码的块。在各种实施例中,过程(2100)由诸如终端装置(310)、(320)、(330)和(340)中的处理电路之类的处理电路、执行视频编码器(403)的功能的处理电路、执行视频解码器(410)的功能的处理电路、执行视频解码器(510)的功能的处理电路、执行视频编码器(603)的功能的处理电路等来执行。在一些实施例中,过程(2100)以软件指令实现,因此,当处理电路执行该软件指令时,处理电路执行过程(2100)。该过程从(S2101)开始,并且进行到(S2110)。

[0254] 在(S2110),可以从已编码视频比特流中解码变换块(TB)的编码信息。

[0255] 在(S2120),可以基于所述编码信息确定是否对所述TB的第一区域执行二次变换,其中所述第一区域包括第一子区域和第二子区域,所述第一子区域具有通过所述二次变换计算的变换系数。在示例中,所述二次变换使用归零方法,因此,所述第二子区域中的变换系数不通过所述二次变换计算并设置为零。当确定执行所述二次变换时,过程(2100)进行到(S2130)。否则,过程(2100)进行到(S2199)。

[0256] 在(S2130),确定所述TB中的第二区域中的变换系数为零,其中所述第二区域在所述第一区域之外。

[0257] 过程(2100)可以适当地修改。例如,可以修改,省略或组合一个或多个步骤。还可以添加其它步骤。例如,基于所述第一区域确定所述TB中的系数单元的大小和位置,并且所

述系数单元之外的变换系数为零。还可以修改执行过程 (2100) 的顺序。

[0258] 上述技术可以通过计算机可读指令实现为计算机软件,并且物理地存储在一个或多个计算机可读介质中。例如,图22示出了计算机系统 (2200),其适于实现所公开主题的某些实施例。

[0259] 所述计算机软件可通过任何合适的机器代码或计算机语言进行编码,通过汇编、编译、链接等机制创建包括指令的代码,所述指令可由一个或多个计算机中央处理单元 (CPU),图形处理单元 (GPU) 等直接执行或通过译码、微代码等方式执行。

[0260] 所述指令可以在各种类型的计算机或其组件上执行,包括例如个人计算机、平板电脑、服务器、智能手机、游戏设备、物联网设备等。

[0261] 图22所示的用于计算机系统 (2200) 的组件本质上是示例性的,并不用于对实现本公开实施例的计算机软件的使用范围或功能进行任何限制。也不应将组件的配置解释为与计算机系统 (2200) 的示例性实施例中所示的任一组件或其组合具有任何依赖性 or 要求。

[0262] 计算机系统 (2200) 可以包括某些人机界面输入设备。这种人机界面输入设备可以通过触觉输入 (如:键盘输入、滑动、数据手套移动)、音频输入 (如:声音、掌声)、视觉输入 (如:手势)、嗅觉输入 (未示出),对一个或多个人类用户的输入做出响应。所述人机界面设备还可用于捕获某些媒体,气与人类有意识的输入不必直接相关,如音频 (例如:语音、音乐、环境声音)、图像 (例如:扫描图像、从静止影像相机获得的摄影图像)、视频 (例如二维视频、包括立体视频的三维视频)。

[0263] 人机界面输入设备可包括以下中的一个或多个 (仅绘出其中一个):键盘 (2201)、鼠标 (2202)、触控板 (2203)、触摸屏 (2210)、数据手套 (未示出)、操纵杆 (2205)、麦克风 (2206)、扫描仪 (2207)、照相机 (2208)。

[0264] 计算机系统 (2200) 还可以包括某些人机界面输出设备。这种人机界面输出设备可以通过例如触觉输出、声音、光和嗅觉/味觉来刺激一个或多个人类用户的感受。这样的人机界面输出设备可包括触觉输出设备 (例如通过触摸屏 (2210)、数据手套 (未示出) 或操纵杆 (2205) 的触觉反馈,但也可以有不用作输入设备的触觉反馈设备)、音频输出设备 (例如,扬声器 (2209)、耳机 (未示出))、视觉输出设备 (例如,包括阴极射线管屏幕、液晶屏幕、等离子屏幕、有机发光二极管屏的屏幕 (2210),其中每一个都具有或没有触摸屏输入功能、每一个都具有或没有触觉反馈功能——其中一些可通过诸如立体画面输出的手段输出二维视觉输出或三维以上的输出;虚拟现实眼镜 (未示出)、全息显示器和放烟箱 (未示出)) 以及打印机 (未示出)。

[0265] 计算机系统 (2200) 还可以包括人可访问的存储设备及其相关介质,如包括具有 CD/DVD 的高密度只读/可重写式光盘 (CD/DVD ROM/RW) (2220) 或类似介质 (2221) 的光学介质、拇指驱动器 (2222)、可移动硬盘驱动器或固体状态驱动器 (2223),诸如磁带和软盘 (未示出) 的传统磁介质,诸如安全软件保护器 (未示出) 等的基于 ROM/ASIC/PLD 的专用设备,等等。

[0266] 本领域技术人员还应当理解,结合所公开的主题使用的术语“计算机可读介质”不包括传输介质、载波或其它瞬时信号。

[0267] 计算机系统 (2200) 还可以包括通往一个或多个通信网络的接口。例如,网络可以是无线的、有线的、光学的。网络还可为局域网、广域网、城域网、车载网络和工业网络、实时

网络、延迟容忍网络等等。网络还包括以太网、无线局域网、蜂窝网络 (GSM、3G、4G、5G、LTE 等) 等局域网、电视有线或无线广域数字网络 (包括有线电视、卫星电视、和地面广播电视)、车载和工业网络 (包括CANBus) 等等。某些网络通常需要外部网络接口适配器,用于连接到某些通用数据端口或外围总线 (2249) (例如,计算机系统 (2200) 的USB端口);其它系统通常通过连接到如下所述的系统总线集成到计算机系统 (2200) 的核心 (例如,以太网接口集成到PC计算机系统或蜂窝网络接口集成到智能电话计算机系统)。通过使用这些网络中的任何一个,计算机系统 (2200) 可以与其它实体进行通信。所述通信可以是单向的,仅用于接收 (例如,无线电视),单向的仅用于发送 (例如CAN总线到某些CAN总线设备),或双向的,例如通过局域或广域数字网络到其它计算机系统。上述的每个网络和网络接口可使用某些协议和协议栈。

[0268] 上述的人机界面设备、人可访问的存储设备以及网络接口可以连接到计算机系统 (2200) 的核心 (2240)。

[0269] 核心 (2240) 可包括一个或多个中央处理单元 (CPU) (2241)、图形处理单元 (GPU) (2242)、以现场可编程门阵列 (FPGA) (2243) 形式的专用可编程处理单元、用于特定任务的硬件加速器 (2244) 等。这些设备以及只读存储器 (ROM) (2245)、随机存取存储器 (2246)、内部大容量存储器 (例如内部非用户可存取硬盘驱动器、固态硬盘等) (2247) 等可通过系统总线 (2248) 进行连接。在某些计算机系统中,可以以一个或多个物理插头的形式访问系统总线 (2248),以便可通过额外的中央处理单元、图形处理单元等进行扩展。外围装置可直接附接到核心的系统总线 (2248),或通过外围总线 (2249) 进行连接。外围总线的体系结构包括外部控制器接口PCI、通用串行总线USB等。

[0270] CPU (2241)、GPU (2242)、FPGA (2243) 和加速器 (2244) 可以执行某些指令,这些指令组合起来可以构成上述计算机代码。该计算机代码可以存储在ROM (2245) 或RAM (2246) 中。过渡数据也可以存储在RAM (2246) 中,而永久数据可以存储在例如内部大容量存储器 (2247) 中。通过使用高速缓冲存储器可实现对任何存储器设备的快速存储和检索,高速缓冲存储器可与一个或多个CPU (2241)、GPU (2242)、大容量存储器 (2247)、ROM (2245)、RAM (2246) 等紧密关联。

[0271] 所述计算机可读介质上可具有计算机代码,用于执行各种计算机实现的操作。介质和计算机代码可以是为本公开的目的而特别设计和构造的,也可以是计算机软件领域的技术人员所熟知和可用的介质和代码。

[0272] 作为实施例而非限制,具有体系结构 (2200) 的计算机系统,特别是核心 (2240),可以作为处理器 (包括CPU、GPU、FPGA、加速器等) 提供执行包含在一个或多个有形的计算机可读介质中的软件的功能。这种计算机可读介质可以是与上述的用户可访问的大容量存储器相关联的介质,以及具有非易失性的核心 (2240) 的特定存储器,例如核心内部大容量存储器 (2247) 或ROM (2245)。实现本公开的各种实施例的软件可以存储在这种设备中并且由核心 (2240) 执行。根据特定需要,计算机可读介质可包括一个或一个以上存储设备或芯片。该软件可以使得核心 (2240) 特别是其中的处理器 (包括CPU、GPU、FPGA等) 执行本文所述的特定过程或特定过程的特定部分,包括定义存储在RAM (2246) 中的数据结构以及根据软件定义的过程来修改这种数据结构。另外或作为替代,计算机系统可以提供逻辑硬连线或以其它方式包含在电路 (例如,加速器 (2244)) 中的功能,该电路可以代替软件或与软件一起运

行以执行本文所述的特定过程或特定过程的特定部分。在适当的情况下,对软件的引用可以包括逻辑,反之亦然。在适当的情况下,对计算机可读介质的引用可包括存储执行软件的电路(如集成电路(IC)),包含执行逻辑的电路,或两者兼备。本公开包括任何合适的硬件和软件组合。

- [0273] 附录A:首字母缩略词
- [0274] JEM:联合开发模型(joint exploration model)
- [0275] VVC:通用视频编解码(versatile video coding)
- [0276] BMS:基准集合(benchmark set)
- [0277] MV:运动矢量(Motion Vector)
- [0278] HEVC:高效视频编解码(High Efficiency Video Coding)
- [0279] SEI:辅助增强信息(Supplementary Enhancement Information)
- [0280] VUI:视频可用性信息(Video Usability Information)
- [0281] GOPs:图片群组(Groups of Pictures)
- [0282] TUs:变换单元(Transform Units)
- [0283] PUs:预测单元(Prediction Units)
- [0284] CTUs:编码树单元(Coding Tree Units)
- [0285] CTBs:编码树块(Coding Tree Blocks)
- [0286] PBs:预测块(Prediction Blocks)
- [0287] HRD:假想参考解码器(Hypothetical Reference Decoder)
- [0288] SNR:信噪比(Signal Noise Ratio)
- [0289] CPUs:中央处理单元(Central Processing Units)
- [0290] GPUs:图形处理单元(Graphics Processing Units)
- [0291] CRT:阴极射线管(Cathode Ray Tube)
- [0292] LCD:液晶显示器(Liquid-Crystal Display)
- [0293] OLED:有机发光二极管(Organic Light-Emitting Diode)
- [0294] CD:光盘(Compact Disc)
- [0295] DVD:数字化视频光盘(Digital Video Disc)
- [0296] ROM:只读存储器(Read-Only Memory)
- [0297] RAM:随机存取存储器(Random Access Memory)
- [0298] ASIC:专用集成电路(Application-Specific Integrated Circuit)
- [0299] PLD:可编程逻辑设备(Programmable Logic Device)
- [0300] LAN:局域网(Local Area Network)
- [0301] GSM:全球移动通信系统(Global System for Mobile communications)
- [0302] LTE:长期演进(Long-Term Evolution)
- [0303] CANBus:控制器局域网络总线(Controller Area Network Bus)
- [0304] USB:通用串行总线(Universal Serial Bus)
- [0305] PCI:外围组件互连(Peripheral Component Interconnect)
- [0306] FPGA:现场可编程门阵列(Field Programmable Gate Areas)
- [0307] SSD:固态驱动器(Solid-state Drive)

- [0308] IC:集成电路(Integrated Circuit)
- [0309] CU:编码单元(Coding Unit)
- [0310] 虽然本公开已对多个示例性实施例进行了描述,但实施例的各种变更、排列和各种等同替换均属于本公开的范围之内。因此应理解,本领域技术人员能够设计多种系统和方法,所述系统和方法虽然未在本文中明确示出或描述,但其体现了本公开的原则,因此属于本公开的精神和范围之内。
- [0311] 附录I
- [0312] 4x4变换
- [0313] {64,64,64,64}
- [0314] {83,36,-36,-83}
- [0315] {64,-64,-64,64}
- [0316] {36,-83,83,-36}
- [0317] 8x8变换
- [0318] {64,64,64,64,64,64,64,64}
- [0319] {89,75,50,18,-18,-50,-75,-89}
- [0320] {83,36,-36,-83,-83,-36,36,83}
- [0321] {75,-18,-89,-50,50,89,18,-75}
- [0322] {64,-64,-64,64,64,-64,-64,64}
- [0323] {50,-89,18,75,-75,-18,89,-50}
- [0324] {36,-83,83,-36,-36,83,-83,36}
- [0325] {18,-50,75,-89,89,-75,50,-18}
- [0326] 16x16变换
- [0327] {64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
- [0328] {90 87 80 70 57 43 25 9 -9-25-43-57-70-80-87-90}
- [0329] {89 75 50 18-18-50-75-89-89-75-50-18 18 50 75 89}
- [0330] {87 57 9-43-80-90-70-25 25 70 90 80 43 -9-57-87}
- [0331] {83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83}
- [0332] {80 9-70-87-25 57 90 43-43-90-57 25 87 70 -9-80}
- [0333] {75-18-89-50 50 89 18-75-75 18 89 50-50-89-18 75}
- [0334] {70-43-87 9 90 25-80-57 57 80-25-90 -9 87 43-70}
- [0335] {64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64}
- [0336] {57-80-25 90 -9-87 43 70-70-43 87 9-90 25 80-57}
- [0337] {50-89 18 75-75-18 89-50-50 89-18-75 75 18-89 50}
- [0338] {43-90 57 25-87 70 9-80 80 -9-70 87-25-57 90-43}
- [0339] {36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36}
- [0340] {25-70 90-80 43 9-57 87-87 57 -9-43 80-90 70-25}
- [0341] {18-50 75-89 89-75 50-18-18 50-75 89-89 75-50 18}
- [0342] {9-25 43-57 70-80 87-90 90-87 80-70 57-43 25 -9}
- [0343] 32x32变换





[0382] {bg,bj,bm,bp,bs,bv,by,cb,ce,ch,ck,-ci,-cf,-cc,-bz,-bw,-bt,-bq,-bn,-bk,-bh,-bf,-bi,-bl,-bo,-br,-bu,-bx,-ca,-cd,-cg,-cj,cj,cg,cd,ca,bx,bu,br,bo,bl,bi,bf,bh,bk,bn,bq,bt,bw,bz,cc,cf,ci,-ck,-ch,-ce,-cb,-by,-bv,-bs,-bp,-bm,-bj,-bg}

[0383] {ah,ai,aj,ak,al,am,an,ao,-ao,-an,-am,-al,-ak,-aj,-ai,-ah,-ah,-ai,-aj,-ak,-al,-am,-an,-ao,ao,an,am,al,ak,aj,ai,ah,ah,ai,aj,ak,al,am,an,ao,-ao,-an,-am,-al,-ak,-aj,-ai,-ah,-ah,-ai,-aj,-ak,-al,-am,-an,-ao,ao,an,am,al,ak,aj,ai,ah}

[0384] {bh,bm,br,bw,cb,cg,-ck,-cf,-ca,-bv,-bq,-bl,-bg,-bi,-bn,-bs,-bx,-cc,-ch,cj,ce,bz,bu,bp,bk,bf,bj,bo,bt,by,cd,ci,-ci,-cd,-by,-bt,-bo,-bj,-bf,-bk,-bp,-bu,-bz,-ce,-cj,ch,cc,bx,bs,bn,bi,bg,bl,bq,bv,ca,cf,ck,-cg,-cb,-bw,-br,-bm,-bh}

[0385] {aq,at,aw,az,bc,-be,-bb,-ay,-av,-as,-ap,-ar,-au,-ax,-ba,-bd,bd,ba,ax,au,ar,ap,as,av,ay,bb,be,-bc,-az,-aw,-at,-aq,-aq,-at,-aw,-az,-bc,be,bb,ay,av,as,ap,ar,au,ax,ba,bd,-bd,-ba,-ax,-au,-ar,-ap,-as,-av,-ay,-bb,-be,bc,az,aw,at,aq}

[0386] {bi,bp,bw,cd,ck,-ce,-bx,-bq,-bj,-bh,-bo,-bv,-cc,-cj,cf,by,br,bk,bg,bn,bu,cb,ci,-cg,-bz,-bs,-bl,-bf,-bm,-bt,-ca,-ch,ch,ca,bt,bm,bf,bl,bs,bz,cg,-ci,-cb,-bu,-bn,-bg,-bk,-br,-by,-cf,cj,cc,bv,bo,bh,bj,bq,bx,ce,-ck,-cd,-bw,-bp,-bi}

[0387] {ad,ae,af,ag,-ag,-af,-ae,-ad,-ad,-ae,-af,-ag,ag,af,ae,ad,ad,ae,af,ag,-ag,-af,-ae,-ad,-ad,-ae,-af,-ag,ag,af,ae,ad,ad,ae,af,ag,-ag,-af,-ae,-ad,-ad,-ae,-af,-ag,ag,af,ae,ad}

[0388] {bj,bs,cb,ck,-cc,-bt,-bk,-bi,-br,-ca,-cj,cd,bu,bl,bh,bq,bz,ci,-ce,-bv,-bm,-bg,-bp,-by,-ch,cf,bw,bn,bf,bo,bx,cg,-cg,-bx,-bo,-bf,-bn,-bw,-cf,ch,by,bp,bg,bm,bv,ce,-ci,-bz,-bq,-bh,-bl,-bu,-cd,cj,ca,br,bi,bk,bt,cc,-ck,-cb,-bs,-bj}

[0389] {ar,aw,bb,-bd,-ay,-at,-ap,-au,-az,-be,ba,av,aq,as,ax,bc,-bc,-ax,-as,-aq,-av,-ba,be,az,au,ap,at,ay,bd,-bb,-aw,-ar,-ar,-aw,-bb,bd,ay,at,ap,au,az,be,-ba,-av,-aq,-as,-ax,-bc,bc,ax,as,aq,av,ba,-be,-az,-au,-ap,-at,-ay,-bd,bb,aw,ar}

[0390] {bk,bv,cg,-ce,-bt,-bi,-bm,-bx,-ci,cc,br,bg,bo,bz,ck,-ca,-bp,-bf,-bq,-cb,cj,by,bn,bh,bs,cd,-ch,-bw,-bl,-bj,-bu,-cf,cf,bu,bj,bl,bw,ch,-cd,-bs,-bh,-bn,-by,-cj,cb,bq,bf,bp,ca,-ck,-bz,-bo,-bg,-br,-cc,ci,bx,bm,bi,bt,ce,-cg,-bv,-bk}

[0391] {ai,al,ao,-am,-aj,-ah,-ak,-an,an,ak,ah,aj,am,-ao,-al,-ai,-ai,-al,-ao,am,aj,ah,ak,an,-an,-ak,-ah,-aj,-am,ao,al,ai,ai,al,ao,-am,-aj,-ah,-ak,-an,an,ak,ah,aj,am,-ao,-al,-ai,-ai,-al,-ao,am,aj,ah,ak,an,-an,-ak,-ah,-aj,-am,ao,al,

ai}

[0392] {bl,by,-ck,-bx,-bk,-bm,-bz,cj,bw,bj,bn,ca,-ci,-bv,-bi,-bo,-cb,ch,bu,bh,bp,cc,-cg,-bt,-bg,-bq,-cd,cf,bs,bf,br,ce,-ce,-br,-bf,-bs,-cf,cd,bq,bg,bt,cg,-cc,-bp,-bh,-bu,-ch,cb,bo,bi,bv,ci,-ca,-bn,-bj,-bw,-cj,bz,bm,bk,bx,ck,-by,-bl}

[0393] {as,az,-bd,-aw,-ap,-av,-bc,ba,at,ar,ay,-be,-ax,-aq,-au,-bb,bb,au,aq,ax,be,-ay,-ar,-at,-ba,bc,av,ap,aw,bd,-az,-as,-as,-az,bd,aw,ap,av,bc,-ba,-at,-ar,-ay,be,ax,aq,au,bb,-bb,-au,-aq,-ax,-be,ay,ar,at,ba,-bc,-av,-ap,-aw,-bd,az,as}

[0394] {bm,cb,-cf,-bq,-bi,-bx,cj,bu,bf,bt,ci,-by,-bj,-bp,-ce,cc,bn,bl,ca,-cg,-br,-bh,-bw,ck,bv,bg,bs,ch,-bz,-bk,-bo,-cd,cd,bo,bk,bz,-ch,-bs,-bg,-bv,-ck,bw,bh,br,cg,-ca,-bl,-bn,-cc,ce,bp,bj,by,-ci,-bt,-bf,-bu,-cj,bx,bi,bq,cf,-cb,-bm}

[0395] {ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab,ab,ac,-ac,-ab,-ab,-ac,ac,ab}

[0396] {bn,ce,-ca,-bj,-br,-ci,bw,bf,bv,-cj,-bs,-bi,-bz,cf,bo,bm,cd,-cb,-bk,-bq,-ch,bx,bg,bu,-ck,-bt,-bh,-by,cg,bp,bl,cc,-cc,-bl,-bp,-cg,by,bh,bt,ck,-bu,-bg,-bx,ch,bq,bk,cb,-cd,-bm,-bo,-cf,bz,bi,bs,cj,-bv,-bf,-bw,ci,br,bj,ca,-ce,-bn}

[0397] {at,bc,-ay,-ap,-ax,bd,au,as,bb,-az,-aq,-aw,be,av,ar,ba,-ba,-ar,-av,-be,aw,aq,az,-bb,-as,-au,-bd,ax,ap,ay,-bc,-at,-at,-bc,ay,ap,ax,-bd,-au,-as,-bb,az,aq,aw,-be,-av,-ar,-ba,ba,ar,av,be,-aw,-aq,-az,bb,as,au,bd,-ax,-ap,-ay,bc,at}

[0398] {bo,ch,-bv,-bh,-ca,cc,bj,bt,-cj,-bq,-bm,-cf,bx,bf,by,-ce,-bl,-br,-ck,bs,bk,cd,-bz,-bg,-bw,cg,bn,bp,ci,-bu,-bi,-cb,cb,bi,bu,-ci,-bp,-bn,-cg,bw,bg,bz,-cd,-bk,-bs,ck,br,bl,ce,-by,-bf,-bx,cf,bm,bq,cj,-bt,-bj,-cc,ca,bh,bv,-ch,-bo}

[0399] {aj,ao,-ak,-ai,-an,al,ah,am,-am,-ah,-al,an,ai,ak,-ao,-aj,-aj,-ao,ak,ai,an,-al,-ah,-am,am,ah,al,-an,-ai,-ak,ao,aj,aj,ao,-ak,-ai,-an,al,ah,am,-am,-ah,-al,an,ai,ak,-ao,-aj,-aj,-ao,ak,ai,an,-al,-ah,-am,am,ah,al,-an,-ai,-ak,ao,aj}

[0400] {bp,ck,-bq,-bo,-cj,br,bn,ci,-bs,-bm,-ch,bt,bl,cg,-bu,-bk,-cf,bv,bj,ce,-bw,-bi,-cd,bx,bh,cc,-by,-bg,-cb,bz,bf,ca,-ca,-bf,-bz,cb,bg,by,-cc,-bh,-bx,cd,bi,bw,-ce,-bj,-bv,cf,bk,bu,-cg,-bl,-bt,ch,bm,bs,-ci,-bn,-br,cj,bo,bq,-ck,-bp}

[0401] {au,-be,-at,-av,bd,as,aw,-bc,-ar,-ax,bb,aq,ay,-ba,-ap,-az,az,ap,ba,-ay,-aq,-bb,ax,ar,bc,-aw,-as,-bd,av,at,be,-au,-au,be,at,av,-bd,-as,-aw,bc,ar,

ax, -bb, -aq, -ay, ba, ap, az, -az, -ap, -ba, ay, aq, bb, -ax, -ar, -bc, aw, as, bd, -av, -at, -be, au}

[0402] {bq, -ci, -bl, -bv, cd, bg, ca, -by, -bi, -cf, bt, bn, ck, -bo, -bs, cg, bj, bx, -cb, -bf, -cc, bw, bk, ch, -br, -bp, cj, bm, bu, -ce, -bh, -bz, bz, bh, ce, -bu, -bm, -cj, bp, br, -ch, -bk, -bw, cc, bf, cb, -bx, -bj, -cg, bs, bo, -ck, -bn, -bt, cf, bi, by, -ca, -bg, -cd, bv, bl, ci, -bq}

[0403] {ae, -ag, -ad, -af, af, ad, ag, -ae, -ae, ag, ad, af, -af, -ad, -ag, ae, ae, -ag, -ad, -af, af, ad, ag, -ae, -ae, ag, ad, af, -af, -ad, -ag, ae, ae, -ag, -ad, -af, af, ad, ag, -ae, -ae, ag, ad, af, -af, -ad, -ag, ae}

[0404] {br, -cf, -bg, -cc, bu, bo, -ci, -bj, -bz, bx, bl, ck, -bm, -bw, ca, bi, ch, -bp, -bt, cd, bf, ce, -bs, -bq, cg, bh, cb, -bv, -bn, cj, bk, by, -by, -bk, -cj, bn, bv, -cb, -bh, -cg, bq, bs, -ce, -bf, -cd, bt, bp, -ch, -bi, -ca, bw, bm, -ck, -bl, -bx, bz, bj, ci, -bo, -bu, cc, bg, cf, -br}

[0405] {av, -bb, -ap, -bc, au, aw, -ba, -aq, -bd, at, ax, -az, -ar, -be, as, ay, -ay, -as, be, ar, az, -ax, -at, bd, aq, ba, -aw, -au, bc, ap, bb, -av, -av, bb, ap, bc, -au, -aw, ba, aq, bd, -at, -ax, az, ar, be, -as, -ay, ay, as, -be, -ar, -az, ax, at, -bd, -aq, -ba, aw, au, -bc, -ap, -bb, av}

[0406] {bs, -cc, -bi, -cj, bl, bz, -bv, -bp, cf, bf, cg, -bo, -bw, by, bm, -ci, -bh, -cd, br, bt, -cb, -bj, -ck, bk, ca, -bu, -bq, ce, bg, ch, -bn, -bx, bx, bn, -ch, -bg, -ce, bq, bu, -ca, -bk, ck, bj, cb, -bt, -br, cd, bh, ci, -bm, -by, bw, bo, -cg, -bf, -cf, bp, bv, -bz, -bl, cj, bi, cc, -bs}

[0407] {ak, -am, -ai, ao, ah, an, -aj, -al, al, aj, -an, -ah, -ao, ai, am, -ak, -ak, am, ai, -ao, -ah, -an, aj, al, -al, -aj, an, ah, ao, -ai, -am, ak, ak, -am, -ai, ao, ah, an, -aj, -al, al, aj, -an, -ah, -ao, ai, am, -ak, -ak, am, ai, -ao, -ah, -an, aj, al, -al, -aj, an, ah, ao, -ai, -am, ak}

[0408] {bt, -bz, -bn, cf, bh, ck, -bi, -ce, bo, by, -bu, -bs, ca, bm, -cg, -bg, -cj, bj, cd, -bp, -bx, bv, br, -cb, -bl, ch, bf, ci, -bk, -cc, bq, bw, -bw, -bq, cc, bk, -ci, -bf, -ch, bl, cb, -br, -bv, bx, bp, -cd, -bj, cj, bg, cg, -bm, -ca, bs, bu, -by, -bo, ce, bi, -ck, -bh, -cf, bn, bz, -bt}

[0409] {aw, -ay, -au, ba, as, -bc, -aq, be, ap, bd, -ar, -bb, at, az, -av, -ax, ax, av, -az, -at, bb, ar, -bd, -ap, -be, aq, bc, -as, -ba, au, ay, -aw, -aw, ay, au, -ba, -as, bc, aq, -be, -ap, -bd, ar, bb, -at, -az, av, ax, -ax, -av, az, at, -bb, -ar, bd, ap, be, -aq, -bc, as, ba, -au, -ay, aw}

[0410] {bu, -bw, -bs, by, bq, -ca, -bo, cc, bm, -ce, -bk, cg, bi, -ci, -bg, ck, bf, cj, -bh, -ch, bj, cf, -bl, -cd, bn, cb, -bp, -bz, br, bx, -bt, -bv, bv, bt, -bx, -br, bz, bp, -cb, -bn, cd, bl, -cf, -bj, ch, bh, -cj, -bf, -ck, bg, ci, -bi, -cg, bk, ce, -bm, -cc, bo, ca, -bq, -by, bs, bw, -bu}

[0411] {aa, -aa, -aa, aa, aa, -aa, -aa, aa, aa, -aa, -aa, aa, aa, -aa, -aa, aa, aa, -aa, -aa,



[0421] {az, -ap, ba, ay, -aq, bb, ax, -ar, bc, aw, -as, bd, av, -at, be, au, -au, -be, at, -av, -bd, as, -aw, -bc, ar, -ax, -bb, aq, -ay, -ba, ap, -az, -az, ap, -ba, -ay, aq, -bb, -ax, ar, -bc, -aw, as, -bd, -av, at, -be, -au, au, be, -at, av, bd, -as, aw, bc, -ar, ax, bb, -aq, ay, ba, -ap, az}

[0422] {ca, -bf, bz, cb, -bg, by, cc, -bh, bx, cd, -bi, bw, ce, -bj, bv, cf, -bk, bu, cg, -bl, bt, ch, -bm, bs, ci, -bn, br, cj, -bo, bq, ck, -bp, bp, -ck, -bq, bo, -cj, -br, bn, -ci, -bs, bm, -ch, -bt, bl, -cg, -bu, bk, -cf, -bv, bj, -ce, -bw, bi, -cd, -bx, bh, -cc, -by, bg, -cb, -bz, bf, -ca}

[0423] {am, -ah, al, an, -ai, ak, ao, -aj, aj, -ao, -ak, ai, -an, -al, ah, -am, -am, ah, -al, -an, ai, -ak, -ao, aj, -aj, ao, ak, -ai, an, al, -ah, am, am, -ah, al, an, -ai, ak, ao, -aj, aj, -ao, -ak, ai, -an, -al, ah, -am, -am, ah, -al, -an, ai, -ak, -ao, aj, -aj, ao, ak, -ai, an, al, -ah, am}

[0424] {cb, -bi, bu, ci, -bp, bn, -cg, -bw, bg, -bz, -cd, bk, -bs, -ck, br, -bl, ce, by, -bf, bx, cf, -bm, bq, -cj, -bt, bj, -cc, -ca, bh, -bv, -ch, bo, -bo, ch, bv, -bh, ca, cc, -bj, bt, cj, -bq, bm, -cf, -bx, bf, -by, -ce, bl, -br, ck, bs, -bk, cd, bz, -bg, bw, cg, -bn, bp, -ci, -bu, bi, -cb}

[0425] {ba, -ar, av, -be, -aw, aq, -az, -bb, as, -au, bd, ax, -ap, ay, bc, -at, at, -bc, -ay, ap, -ax, -bd, au, -as, bb, az, -aq, aw, be, -av, ar, -ba, -ba, ar, -av, be, aw, -aq, az, bb, -as, au, -bd, -ax, ap, -ay, -bc, at, -at, bc, ay, -ap, ax, bd, -au, as, -bb, -az, aq, -aw, -be, av, -ar, ba}

[0426] {cc, -bl, bp, -cg, -by, bh, -bt, ck, bu, -bg, bx, ch, -bq, bk, -cb, -cd, bm, -bo, cf, bz, -bi, bs, -cj, -bv, bf, -bw, -ci, br, -bj, ca, ce, -bn, bn, -ce, -ca, bj, -br, ci, bw, -bf, bv, cj, -bs, bi, -bz, -cf, bo, -bm, cd, cb, -bk, bq, -ch, -bx, bg, -bu, -ck, bt, -bh, by, cg, -bp, bl, -cc}

[0427] {ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac, -ab, ab, -ac, -ac, ab, -ab, ac, ac}

[0428] {cd, -bo, bk, -bz, -ch, bs, -bg, bv, -ck, -bw, bh, -br, cg, ca, -bl, bn, -cc, -ce, bp, -bj, by, ci, -bt, bf, -bu, cj, bx, -bi, bq, -cf, -cb, bm, -bm, cb, cf, -bq, bi, -bx, -cj, bu, -bf, bt, -ci, -by, bj, -bp, ce, cc, -bn, bl, -ca, -cg, br, -bh, bw, ck, -bv, bg, -bs, ch, bz, -bk, bo, -cd}

[0429] {bb, -au, aq, -ax, be, ay, -ar, at, -ba, -bc, av, -ap, aw, -bd, -az, as, -as, az, bd, -aw, ap, -av, bc, ba, -at, ar, -ay, -be, ax, -aq, au, -bb, -bb, au, -aq, ax, -be, -ay, ar, -at, ba, bc, -av, ap, -aw, bd, az, -as, as, -az, -bd, aw, -ap, av, -bc, -ba, at, -ar, ay, be, -ax, aq, -au, bb}

[0430] {ce, -br, bf, -bs, cf, cd, -bq, bg, -bt, cg, cc, -bp, bh, -bu, ch, cb, -bo, bi, -bv, ci, ca, -bn, bj, -bw, cj, bz, -bm, bk, -bx, ck, by, -bl, bl, -by, -ck, bx, -bk, bm, -bz, -cj, bw, -bj, bn, -ca, -ci, bv, -bi, bo, -cb, -ch, bu, -bh, bp, -cc, -cg, bt, -bg, bq, -cd, -cf, bs, -bf, br, -

ce}

[0431] {an,-ak,ah,-aj,am,ao,-al,ai,-ai,al,-ao,-am,aj,-ah,ak,-an,-an,ak,-ah,aj,-am,-ao,al,-ai,ai,-al,ao,am,-aj,ah,-ak,an,an,-ak,ah,-aj,am,ao,-al,ai,-ai,al,-ao,-am,aj,-ah,ak,-an,-an,ak,-ah,aj,-am,-ao,al,-ai,ai,-al,ao,am,-aj,ah,-ak,an}

[0432] {cf,-bu,bj,-bl,bw,-ch,-cd,bs,-bh,bn,-by,cj,cb,-bq,bf,-bp,ca,ck,-bz,bo,-bg,br,-cc,-ci,bx,-bm,bi,-bt,ce,cg,-bv,bk,-bk,bv,-cg,-ce,bt,-bi,bm,-bx,ci,cc,-br,bg,-bo,bz,-ck,-ca,bp,-bf,bq,-cb,-cj,by,-bn,bh,-bs,cd,ch,-bw,bl,-bj, bu,-cf}

[0433] {bc,-ax,as,-aq,av,-ba,-be,az,-au,ap,-at,ay,-bd,-bb,aw,-ar,ar,-aw,bb, bd,-ay,at,-ap,au,-az,be,ba,-av,aq,-as,ax,-bc,-bc,ax,-as,aq,-av,ba,be,-az,au,- ap,at,-ay,bd,bb,-aw,ar,-ar,aw,-bb,-bd,ay,-at,ap,-au,az,-be,-ba,av,-aq,as,-ax, bc}

[0434] {cg,-bx,bo,-bf,bn,-bw,cf,ch,-by,bp,-bg,bm,-bv,ce,ci,-bz,bq,-bh,bl,- bu,cd,cj,-ca,br,-bi,bk,-bt,cc,ck,-cb,bs,-bj,bj,-bs,cb,-ck,-cc,bt,-bk,bi,-br, ca,-cj,-cd,bu,-bl,bh,-bq,bz,-ci,-ce,bv,-bm,bg,-bp,by,-ch,-cf,bw,-bn,bf,-bo, bx,-cg}

[0435] {ag,-af,ae,-ad,ad,-ae,af,-ag,-ag,af,-ae,ad,-ad,ae,-af,ag,ag,-af,ae,- ad,ad,-ae,af,-ag,-ag,af,-ae,ad,-ad,ae,-af,ag,ag,-af,ae,-ad,ad,-ae,af,-ag,-ag, af,-ae,ad,-ad,ae,-af,ag,ag,-af,ae,-ad,ad,-ae,af,-ag,-ag,af,-ae,ad,-ad,ae,-af, ag}

[0436] {ch,-ca,bt,-bm,bf,-bl,bs,-bz,cg,ci,-cb,bu,-bn,bg,-bk,br,-by,cf,cj,- cc,bv,-bo,bh,-bj,bq,-bx,ce,ck,-cd,bw,-bp,bi,-bi,bp,-bw,cd,-ck,-ce,bx,-bq,bj,- bh,bo,-bv,cc,-cj,-cf,by,-br,bk,-bg,bn,-bu,cb,-ci,-cg,bz,-bs,bl,-bf,bm,-bt, ca,-ch}

[0437] {bd,-ba,ax,-au,ar,-ap,as,-av,ay,-bb,be,bc,-az,aw,-at,aq,-aq,at,-aw, az,-bc,-be,bb,-ay,av,-as,ap,-ar,au,-ax,ba,-bd,-bd,ba,-ax,au,-ar,ap,-as,av,- ay,bb,-be,-bc,az,-aw,at,-aq,aq,-at,aw,-az,bc,be,-bb,ay,-av,as,-ap,ar,-au,ax,- ba,bd}

[0438] {ci,-cd,by,-bt,bo,-bj,bf,-bk,bp,-bu,bz,-ce,cj,ch,-cc,bx,-bs,bn,-bi, bg,-bl,bq,-bv,ca,-cf,ck,cg,-cb,bw,-br,bm,-bh,bh,-bm,br,-bw,cb,-cg,-ck,cf,-ca, bv,-bq,bl,-bg,bi,-bn,bs,-bx,cc,-ch,-cj,ce,-bz,bu,-bp,bk,-bf,bj,-bo,bt,-by, cd,-ci}

[0439] {ao,-an,am,-al,ak,-aj,ai,-ah,ah,-ai,aj,-ak,al,-am,an,-ao,-ao,an,-am, al,-ak,aj,-ai,ah,-ah,ai,-aj,ak,-al,am,-an,ao,ao,-an,am,-al,ak,-aj,ai,-ah,ah,- ai,aj,-ak,al,-am,an,-ao,-ao,an,-am,al,-ak,aj,-ai,ah,-ah,ai,-aj,ak,-al,am,-an, ao}

[0440] {cj,-cg,cd,-ca,bx,-bu,br,-bo,bl,-bi,bf,-bh,bk,-bn,bq,-bt,bw,-bz,cc,- cf,ci,ck,-ch,ce,-cb,by,-bv,bs,-bp,bm,-bj,bg,-bg,bj,-bm,bp,-bs,bv,-by,cb,-ce,

ch, -ck, -ci, cf, -cc, bz, -bw, bt, -bq, bn, -bk, bh, -bf, bi, -bl, bo, -br, bu, -bx, ca, -cd, cg, -cj}

[0441] {be, -bd, bc, -bb, ba, -az, ay, -ax, aw, -av, au, -at, as, -ar, aq, -ap, ap, -aq, ar, -as, at, -au, av, -aw, ax, -ay, az, -ba, bb, -bc, bd, -be, -be, bd, -bc, bb, -ba, az, -ay, ax, -aw, av, -au, at, -as, ar, -aq, ap, -ap, aq, -ar, as, -at, au, -av, aw, -ax, ay, -az, ba, -bb, bc, -bd, be}

[0442] {ck, -cj, ci, -ch, cg, -cf, ce, -cd, cc, -cb, ca, -bz, by, -bx, bw, -bv, bu, -bt, bs, -br, bq, -bp, bo, -bn, bm, -bl, bk, -bj, bi, -bh, bg, -bf, bf, -bg, bh, -bi, bj, -bk, bl, -bm, bn, -bo, bp, -bq, br, -bs, bt, -bu, bv, -bw, bx, -by, bz, -ca, cb, -cc, cd, -ce, cf, -cg, ch, -ci, cj, -ck}

[0443] }

[0444] 其中,

[0445] {aa, ab, ac, ad, ae, af, ag, ah, ai, aj, ak, al, am, an, ao, ap, aq, ar, as, at, au, av, aw, ax, ay, az, ba, bb, bc, bd, be, bf, bg, bh, bi, bj, bk, bl, bm, bn, bo, bp, bq, br, bs, bt, bu, bv, bw, bx, by, bz, ca, cb, cc, cd, ce, cf, cg, ch, ci, cj, ck} = {64, 83, 36, 89, 75, 50, 18, 90, 87, 80, 70, 57, 43, 25, 9, 90, 90, 88, 85, 82, 78, 73, 67, 61, 54, 46, 38, 31, 22, 13, 4, 91, 90, 90, 90, 88, 87, 86, 84, 83, 81, 79, 77, 73, 71, 69, 65, 62, 59, 56, 52, 48, 44, 41, 37, 33, 28, 24, 20, 15, 11, 7, 2}

[0446] 附录III

[0447] 4点DST-7

[0448] {a, b, c, d}

[0449] {c, c, 0, -c}

[0450] {d, -a, -c, b}

[0451] {b, -d, c, -a}

[0452] 其中, {a, b, c, d} = {29, 55, 74, 84}

[0453] 8点DST-7:

[0454] {a, b, c, d, e, f, g, h,}

[0455] {c, f, h, e, b, -a, -d, -g,}

[0456] {e, g, b, -c, -h, -d, a, f,}

[0457] {g, c, -d, -f, a, h, b, -e,}

[0458] {h, -a, -g, b, f, -c, -e, d,}

[0459] {f, -e, -a, g, -d, -b, h, -c,}

[0460] {d, -h, e, -a, -c, g, -f, b,}

[0461] {b, -d, f, -h, g, -e, c, -a,}

[0462] 其中, {a, b, c, d, e, f, g, h} = {17, 32, 46, 60, 71, 78, 85, 86}

[0463] 16点DST-7

[0464] {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,}

[0465] {c, f, i, l, o, o, l, i, f, c, 0, -c, -f, -i, -l, -o,}

[0466] {e, j, o, m, h, c, -b, -g, -l, -p, -k, -f, -a, d, i, n,}

- [0467] {g,n,l,e,-b,-i,-p,-j,-c,d,k,o,h,a,-f,-m,}
- [0468] {i,o,f,-c,-l,-l,-c,f,o,i,0,-i,-o,-f,c,l,}
- [0469] {k,k,0,-k,-k,0,k,k,0,-k,-k,0,k,k,0,-k,}
- [0470] {m,g,-f,-n,-a,l,h,-e,-o,-b,k,i,-d,-p,-c,j,}
- [0471] {o,c,-l,-f,i,i,-f,-l,c,o,0,-o,-c,l,f,-i,}
- [0472] {p,-a,-o,b,n,-c,-m,d,l,-e,-k,f,j,-g,-i,h,}
- [0473] {n,-e,-i,j,d,-o,a,m,-f,-h,k,c,-p,b,l,-g,}
- [0474] {l,-i,-c,o,-f,-f,o,-c,-i,l,0,-l,i,c,-o,f,}
- [0475] {j,-m,c,g,-p,f,d,-n,i,a,-k,l,-b,-h,o,-e,}
- [0476] {h,-p,i,-a,-g,o,-j,b,f,-n,k,-c,-e,m,-l,d,}
- [0477] {f,-l,o,-i,c,c,-i,o,-l,f,0,-f,l,-o,i,-c,}
- [0478] {d,-h,l,-p,m,-i,e,-a,-c,g,-k,o,-n,j,-f,b,}
- [0479] {b,-d,f,-h,j,-l,n,-p,o,-m,k,-i,g,-e,c,-a,}
- [0480] 其中, {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p} = {9,17,25,33,41,49,56,62,66,72,77,81,83,87,89,90}
- [0481] 32点DST-7
- [0482] {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F,}
- [0483] {c,f,i,l,o,r,u,x,A,D,F,C,z,w,t,q,n,k,h,e,b,-a,-d,-g,-j,-m,-p,-s,-v,-y,-B,-E,}
- [0484] {e,j,o,t,y,D,D,y,t,o,j,e,0,-e,-j,-o,-t,-y,-D,-D,-y,-t,-o,-j,-e,0,e,j,o,t,y,D,}
- [0485] {g,n,u,B,D,w,p,i,b,-e,-l,-s,-z,-F,-y,-r,-k,-d,c,j,q,x,E,A,t,m,f,-a,-h,-o,-v,-C,}
- [0486] {i,r,A,C,t,k,b,-g,-p,-y,-E,-v,-m,-d,e,n,w,F,x,o,f,-c,-l,-u,-D,-z,-q,-h,a,j,s,B,}
- [0487] {k,v,F,u,j,-a,-l,-w,-E,-t,-i,b,m,x,D,s,h,-c,-n,-y,-C,-r,-g,d,o,z,B,q,f,-e,-p,-A,}
- [0488] {m,z,z,m,0,-m,-z,-z,-m,0,m,z,z,m,0,-m,-z,-z,-m,0,m,z,z,m,0,-m,-z,-z,-m,0,m,z,}
- [0489] {o,D,t,e,-j,-y,-y,-j,e,t,D,o,0,-o,-D,-t,-e,j,y,y,j,-e,-t,-D,-o,0,o,D,t,e,-j,-y,}
- [0490] {q,E,n,-c,-t,-B,-k,f,w,y,h,-i,-z,-v,-e,l,C,s,b,-o,-F,-p,a,r,D,m,-d,-u,-A,-j,g,x,}
- [0491] {s,A,h,-k,-D,-p,c,v,x,e,-n,-F,-m,f,y,u,b,-q,-C,-j,i,B,r,-a,-t,-z,-g,l,E,o,-d,-w,}
- [0492] {u,w,b,-s,-y,-d,q,A,f,-o,-C,-h,m,E,j,-k,-F,-l,i,D,n,-g,-B,-p,e,z,r,-c,-x,-t,a,v,}
- [0493] {w,s,-d,-A,-o,h,E,k,-l,-D,-g,p,z,c,-t,-v,a,x,r,-e,-B,-n,i,F,j,-m,-C,-f,q,y,b,-u,}

- [0494] {y,o,-j,-D,-e,t,t,-e,-D,-j,o,y,0,-y,-o,j,D,e,-t,-t,e,D,j,-o,-y,0,y,o,-j,-D,-e,t,}
- [0495] {A,k,-p,-v,e,F,f,-u,-q,j,B,a,-z,-l,o,w,-d,-E,-g,t,r,-i,-C,-b,y,m,-n,-x,c,D,h,-s,}
- [0496] {C,g,-v,-n,o,u,-h,-B,a,D,f,-w,-m,p,t,-i,-A,b,E,e,-x,-l,q,s,-j,-z,c,F,d,-y,-k,r,}
- [0497] {E,c,-B,-f,y,i,-v,-l,s,o,-p,-r,m,u,-j,-x,g,A,-d,-D,a,F,b,-C,-e,z,h,-w,-k,t,n,-q,}
- [0498] {F,-a,-E,b,D,-c,-C,d,B,-e,-A,f,z,-g,-y,h,x,-i,-w,j,v,-k,-u,l,t,-m,-s,n,r,-o,-q,p,}
- [0499] {D,-e,-y,j,t,-o,-o,t,j,-y,-e,D,0,-D,e,y,-j,-t,o,o,-t,-j,y,e,-D,0,D,-e,-y,j,t,-o,}
- [0500] {B,-i,-s,r,j,-A,-a,C,-h,-t,q,k,-z,-b,D,-g,-u,p,l,-y,-c,E,-f,-v,o,m,-x,-d,F,-e,-w,n,}
- [0501] {z,-m,-m,z,0,-z,m,m,-z,0,z,-m,-m,z,0,-z,m,m,-z,0,z,-m,-m,z,0,-z,m,m,-z,0,z,-m,}
- [0502] {x,-q,-g,E,-j,-n,A,-c,-u,t,d,-B,m,k,-D,f,r,-w,-a,y,-p,-h,F,-i,-o,z,-b,-v,s,e,-C,l,}
- [0503] {v,-u,-a,w,-t,-b,x,-s,-c,y,-r,-d,z,-q,-e,A,-p,-f,B,-o,-g,C,-n,-h,D,-m,-i,E,-l,-j,F,-k,}
- [0504] {t,-y,e,o,-D,j,j,-D,o,e,-y,t,0,-t,y,-e,-o,D,-j,-j,D,-o,-e,y,-t,0,t,-y,e,o,-D,j,}
- [0505] {r,-C,k,g,-y,v,-d,-n,F,-o,-c,u,-z,h,j,-B,s,-a,-q,D,-l,-f,x,-w,e,m,-E,p,b,-t,A,-i,}
- [0506] {p,-F,q,-a,-o,E,-r,b,n,-D,s,-c,-m,C,-t,d,l,-B,u,-e,-k,A,-v,f,j,-z,w,-g,-i,y,-x,h,}
- [0507] {n,-B,w,-i,-e,s,-F,r,-d,-j,x,-A,m,a,-o,C,-v,h,f,-t,E,-q,c,k,-y,z,-l,-b,p,-D,u,-g,}
- [0508] {l,-x,C,-q,e,g,-s,E,-v,j,b,-n,z,-A,o,-c,-i,u,-F,t,-h,-d,p,-B,y,-m,a,k,-w,D,-r,f,}
- [0509] {j,-t,D,-y,o,-e,-e,o,-y,D,-t,j,0,-j,t,-D,y,-o,e,e,-o,y,-D,t,-j,0,j,-t,D,-y,o,-e,}
- [0510] {h,-p,x,-F,y,-q,i,-a,-g,o,-w,E,-z,r,-j,b,f,-n,v,-D,A,-s,k,-c,-e,m,-u,C,-B,t,-l,d,}
- [0511] {f,-l,r,-x,D,-C,w,-q,k,-e,-a,g,-m,s,-y,E,-B,v,-p,j,-d,-b,h,-n,t,-z,F,-A,u,-o,i,-c,}
- [0512] {d,-h,l,-p,t,-x,B,-F,C,-y,u,-q,m,-i,e,-a,-c,g,-k,o,-s,w,-A,E,-D,z,-v,r,-n,j,-f,b,}
- [0513] {b,-d,f,-h,j,-l,n,-p,r,-t,v,-x,z,-B,D,-F,E,-C,A,-y,w,-u,s,-q,o,-m,k,-

i, g, -e, c, -a,}

[0514] 其中, {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F} = {4, 9, 13, 17, 21, 26, 30, 34, 38, 42, 45, 50, 53, 56, 60, 63, 66, 68, 72, 74, 77, 78, 80, 82, 84, 85, 86, 88, 88, 89, 90}

[0515] 4点DCT-8

[0516] {a, b, c, d,}

[0517] {b, 0, -b, -b,}

[0518] {c, -b, -d, a,}

[0519] {d, -b, a, -c,}

[0520] 其中, {a, b, c, d} = {84, 74, 55, 29}

[0521] 8点DCT-8:

[0522] {a, b, c, d, e, f, g, h,}

[0523] {b, e, h, -g, -d, -a, -c, -f,}

[0524] {c, h, -e, -a, -f, g, b, d,}

[0525] {d, -g, -a, -h, c, e, -f, -b,}

[0526] {e, -d, -f, c, g, -b, -h, a,}

[0527] {f, -a, g, e, -b, h, d, -c,}

[0528] {g, -c, b, -f, -h, d, -a, e,}

[0529] {h, -f, d, -b, a, -c, e, -g,}

[0530] 其中, {a, b, c, d, e, f, g, h} = {86, 85, 78, 71, 60, 46, 32, 17}

[0531] 16点DCT-8

[0532] {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,}

[0533] {b, e, h, k, n, 0, -n, -k, -h, -e, -b, -b, -e, -h, -k, -n,}

[0534] {c, h, m, -p, -k, -f, -a, -e, -j, -o, n, i, d, b, g, l,}

[0535] {d, k, -p, -i, -b, -f, -m, n, g, a, h, o, -l, -e, -c, -j,}

[0536] {e, n, -k, -b, -h, 0, h, b, k, -n, -e, -e, -n, k, b, h,}

[0537] {f, 0, -f, -f, 0, f, f, 0, -f, -f, 0, f, f, 0, -f, -f,}

[0538] {g, -n, -a, -m, h, f, -o, -b, -l, i, e, -p, -c, -k, j, d,}

[0539] {h, -k, -e, n, b, 0, -b, -n, e, k, -h, -h, k, e, -n, -b,}

[0540] {i, -h, -j, g, k, -f, -l, e, m, -d, -n, c, o, -b, -p, a,}

[0541] {j, -e, -o, a, -n, -f, i, k, -d, -p, b, -m, -g, h, l, -c,}

[0542] {k, -b, n, h, -e, 0, e, -h, -n, b, -k, -k, b, -n, -h, e,}

[0543] {l, -b, i, o, -e, f, -p, -h, c, -m, -k, a, -j, -n, d, -g,}

[0544] {m, -e, d, -l, -n, f, -c, k, o, -g, b, -j, -p, h, -a, i,}

[0545] {n, -h, b, -e, k, 0, -k, e, -b, h, -n, -n, h, -b, e, -k,}

[0546] {o, -k, g, -c, b, -f, j, -n, -p, l, -h, d, -a, e, -i, m,}

[0547] {p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o,}

[0548] 其中, {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p} = {90, 89, 87, 83, 81, 77, 72, 66, 62, 56, 49, 41, 33, 25, 17, 9} 32点DCT-8

- [0549] {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F,}
- [0550] {b,e,h,k,n,q,t,w,z,C,F,-E,-B,-y,-v,-s,-p,-m,-j,-g,-d,-a,-c,-f,-i,-l,-o,-r,-u,-x,-A,-D,}
- [0551] {c,h,m,r,w,B,0,-B,-w,-r,-m,-h,-c,-c,-h,-m,-r,-w,-B,0,B,w,r,m,h,c,c,h,m,r,w,B,}
- [0552] {d,k,r,y,F,-A,-t,-m,-f,-b,-i,-p,-w,-D,C,v,o,h,a,g,n,u,B,-E,-x,-q,-j,-c,-e,-l,-s,-z,}
- [0553] {e,n,w,F,-y,-p,-g,-c,-l,-u,-D,A,r,i,a,j,s,B,-C,-t,-k,-b,-h,-q,-z,E,v,m,d,f,o,x,}
- [0554] {f,q,B,-A,-p,-e,-g,-r,-C,z,o,d,h,s,D,-y,-n,-c,-i,-t,-E,x,m,b,j,u,F,-w,-l,-a,-k,-v,}
- [0555] {g,t,0,-t,-g,-g,-t,0,t,g,g,t,0,-t,-g,-g,-t,0,t,g,g,t,0,-t,-g,-g,-t,0,t,g,g,t,}
- [0556] {h,w,-B,-m,-c,-r,0,r,c,m,B,-w,-h,-h,-w,B,m,c,r,0,-r,-c,-m,-B,w,h,h,w,-B,-m,-c,-r,}
- [0557] {i,z,-w,-f,-l,-C,t,c,o,F,-q,-a,-r,E,n,d,u,-B,-k,-g,-x,y,h,j,A,-v,-e,-m,-D,s,b,p,}
- [0558] {j,C,-r,-b,-u,z,g,m,F,-o,-e,-x,w,d,p,-E,-l,-h,-A,t,a,s,-B,-i,-k,-D,q,c,v,-y,-f,-n,}
- [0559] {k,F,-m,-i,-D,o,g,B,-q,-e,-z,s,c,x,-u,-a,-v,w,b,t,-y,-d,-r,A,f,p,-C,-h,-n,E,j,l,}
- [0560] {l,-E,-h,-p,A,d,t,-w,-a,-x,s,e,B,-o,-i,-F,k,m,-D,-g,-q,z,c,u,-v,-b,-y,r,f,C,-n,-j,}
- [0561] {m,-B,-c,-w,r,h,0,-h,-r,w,c,B,-m,-m,B,c,w,-r,-h,0,h,r,-w,-c,-B,m,m,-B,-c,-w,r,h,}
- [0562] {n,-y,-c,-D,i,s,-t,-h,E,d,x,-o,-m,z,b,C,-j,-r,u,g,-F,-e,-w,p,l,-A,-a,-B,k,q,-v,-f,}
- [0563] {o,-v,-h,C,a,D,-g,-w,n,p,-u,-i,B,b,E,-f,-x,m,q,-t,-j,A,c,F,-e,-y,l,r,-s,-k,z,d,}
- [0564] {p,-s,-m,v,j,-y,-g,B,d,-E,-a,-F,c,C,-f,-z,i,w,-l,-t,o,q,-r,-n,u,k,-x,-h,A,e,-D,-b,}
- [0565] {q,-p,-r,o,s,-n,-t,m,u,-l,-v,k,w,-j,-x,i,y,-h,-z,g,A,-f,-B,e,C,-d,-D,c,E,-b,-F,a,}
- [0566] {r,-m,-w,h,B,-c,0,c,-B,-h,w,m,-r,-r,m,w,-h,-B,c,0,-c,B,h,-w,-m,r,r,-m,-w,h,B,-c,}
- [0567] {s,-j,-B,a,-C,-i,t,r,-k,-A,b,-D,-h,u,q,-l,-z,c,-E,-g,v,p,-m,-y,d,-F,-f,w,o,-n,-x,e,}
- [0568] {t,-g,0,g,-t,-t,g,0,-g,t,t,-g,0,g,-t,-t,g,0,-g,t,t,-g,0,g,-t,-t,g,0,-g,t,t,-g,}

- [0569] {u,-d,B,n,-k,-E,g,-r,-x,a,-y,-q,h,-F,-j,o,A,-c,v,t,-e,C,m,-l,-D,f,-s,-w,b,-z,-p,i,}
- [0570] {v,-a,w,u,-b,x,t,-c,y,s,-d,z,r,-e,A,q,-f,B,p,-g,C,o,-h,D,n,-i,E,m,-j,F,l,-k,}
- [0571] {w,-c,r,B,-h,m,0,-m,h,-B,-r,c,-w,-w,c,-r,-B,h,-m,0,m,-h,B,r,-c,w,w,-c,r,B,-h,m,}
- [0572] {x,-f,m,-E,-q,b,-t,-B,j,-i,A,u,-c,p,F,-n,e,-w,-y,g,-l,D,r,-a,s,C,-k,h,-z,-v,d,-o,}
- [0573] {y,-i,h,-x,-z,j,-g,w,A,-k,f,-v,-B,l,-e,u,C,-m,d,-t,-D,n,-c,s,E,-o,b,-r,-F,p,-a,q,}
- [0574] {z,-l,c,-q,E,u,-g,h,-v,-D,p,-b,m,-A,-y,k,-d,r,-F,-t,f,-i,w,C,-o,a,-n,B,x,-j,e,-s,}
- [0575] {A,-o,c,-j,v,F,-t,h,-e,q,-C,-y,m,-a,l,-x,-D,r,-f,g,-s,E,w,-k,b,-n,z,B,-p,d,-i,u,}
- [0576] {B,-r,h,-c,m,-w,0,w,-m,c,-h,r,-B,-B,r,-h,c,-m,w,0,-w,m,-c,h,-r,B,B,-r,h,-c,m,-w,}
- [0577] {C,-u,m,-e,d,-l,t,-B,-D,v,-n,f,-c,k,-s,A,E,-w,o,-g,b,-j,r,-z,-F,x,-p,h,-a,i,-q,y,}
- [0578] {D,-x,r,-l,f,-a,g,-m,s,-y,E,C,-w,q,-k,e,-b,h,-n,t,-z,F,B,-v,p,-j,d,-c,i,-o,u,-A,}
- [0579] {E,-A,w,-s,o,-k,g,-c,b,-f,j,-n,r,-v,z,-D,-F,B,-x,t,-p,l,-h,d,-a,e,-i,m,-q,u,-y,C,}
- [0580] {F,-D,B,-z,x,-v,t,-r,p,-n,l,-j,h,-f,d,-b,a,-c,e,-g,i,-k,m,-o,q,-s,u,-w,y,-A,C,-E,}
- [0581] 其中, {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F} = {90,90,89,88,88,86,85,84,82,80,78,77,74,72,68,66,63,60,56,53,50,45,42,38,34,30,26,21,17,13,9,4}

	R01	R02	R03	R04	R05	R06	R07	R08	R09
R10	<u>S11</u>	S12	S13	S14					
<u>R20</u>	S21	S22	S23	<u>S24</u>					
R30	S31	<u>S32</u>	S33	S34					
R40	<u>S41</u>	S42	S43	<u>S44</u>					
R50									
<u>R60</u>									
R70									

104

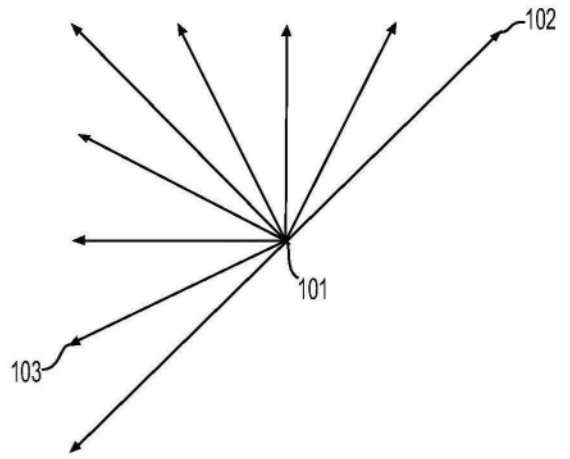
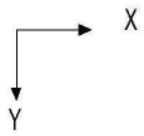


图1A

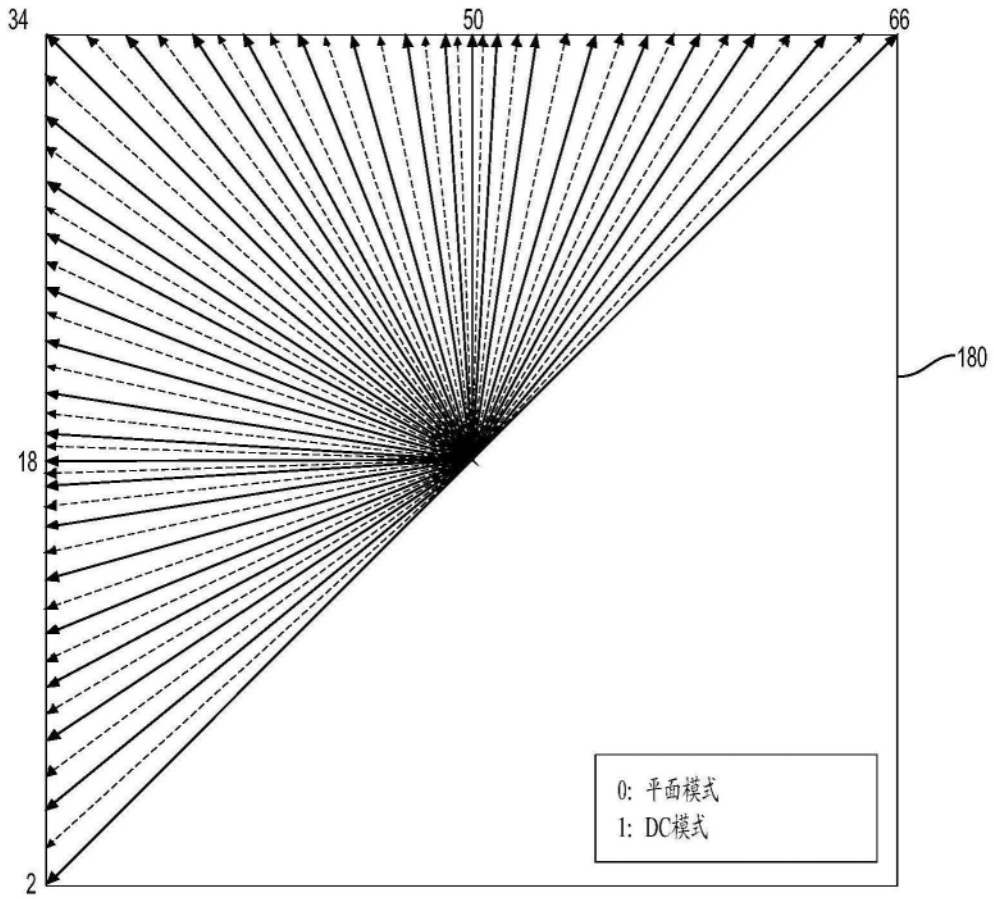
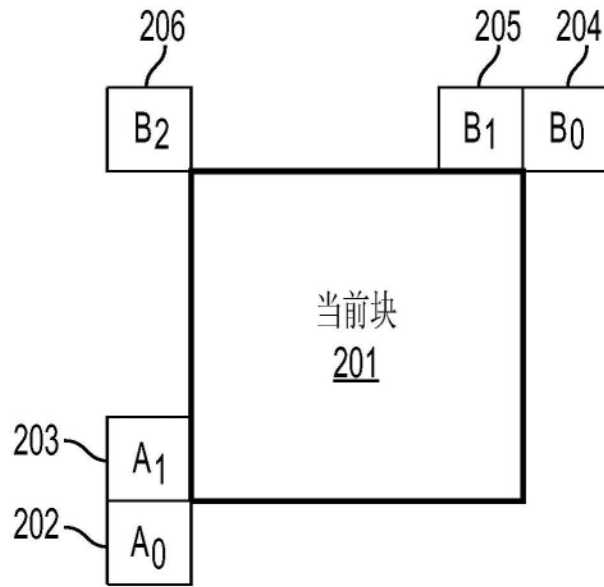


图1B



(相关技术)

图2

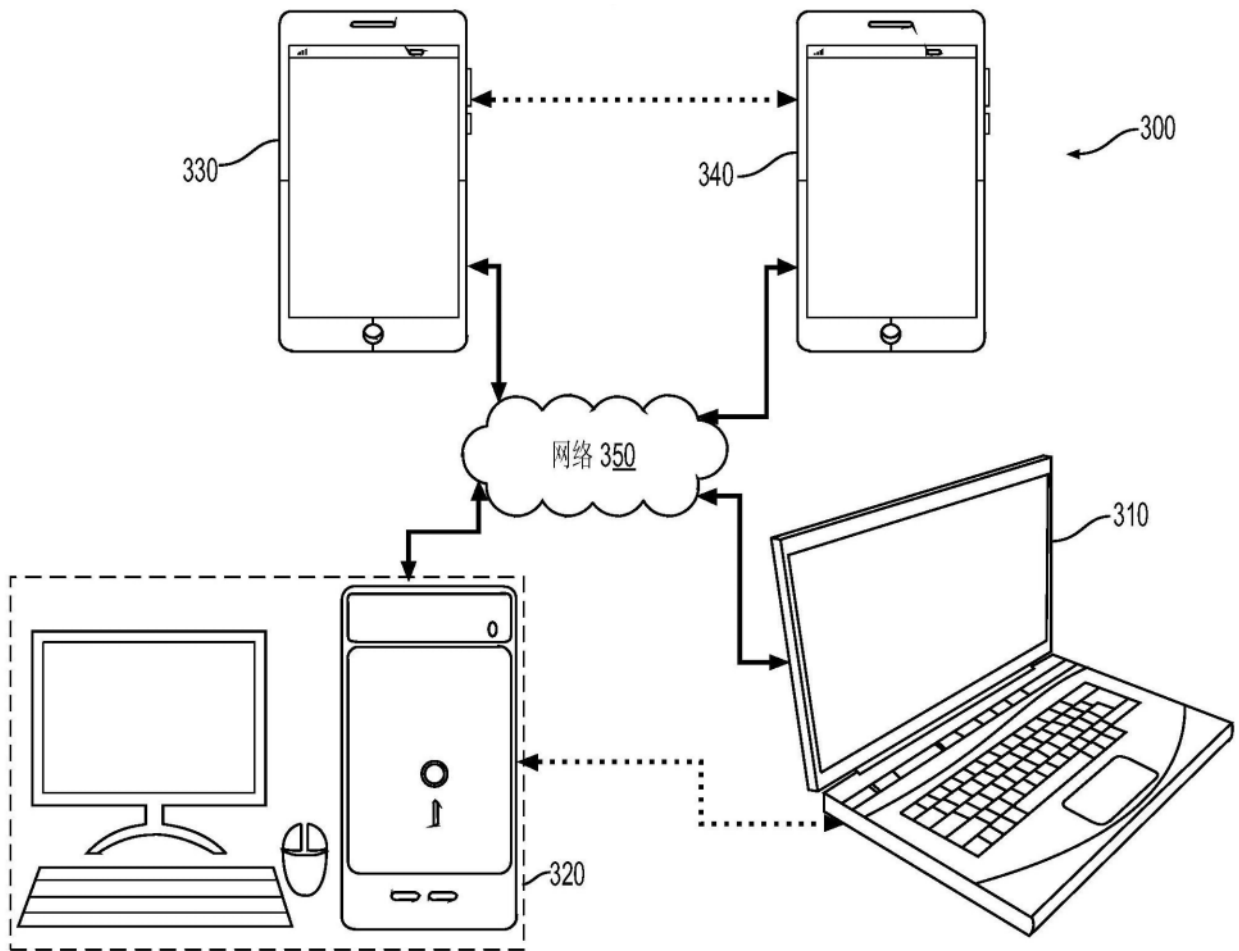


图3

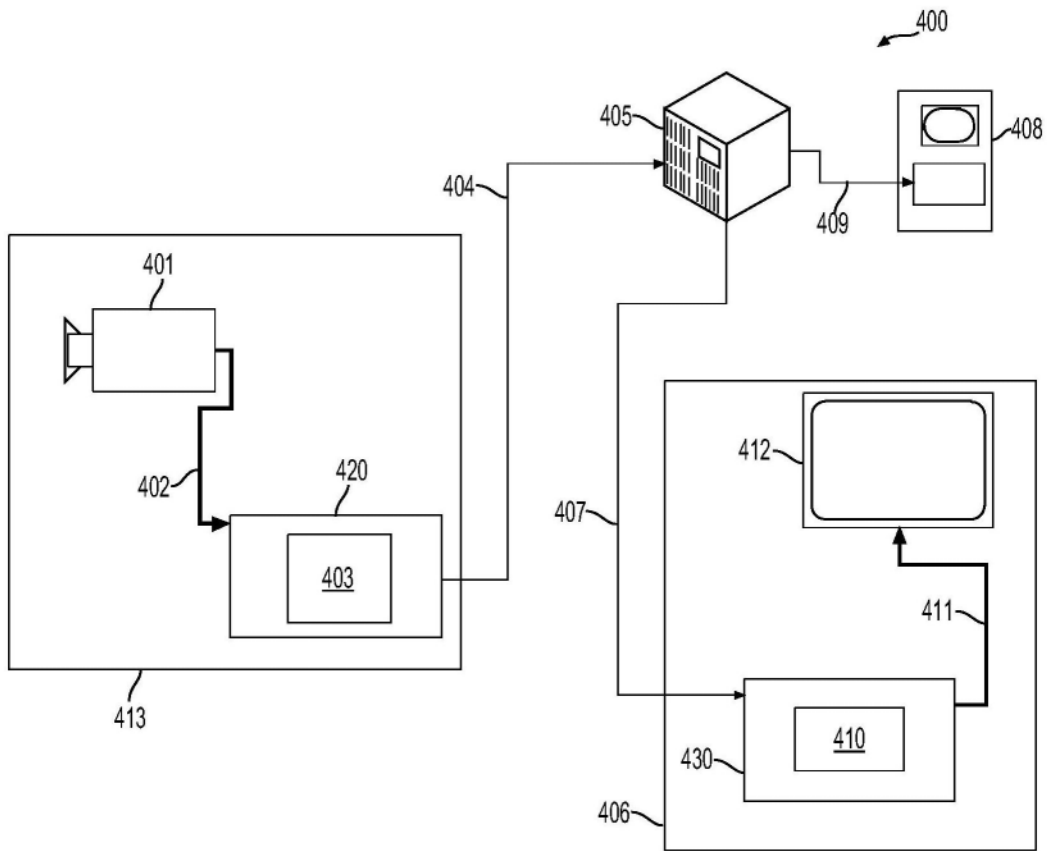


图4

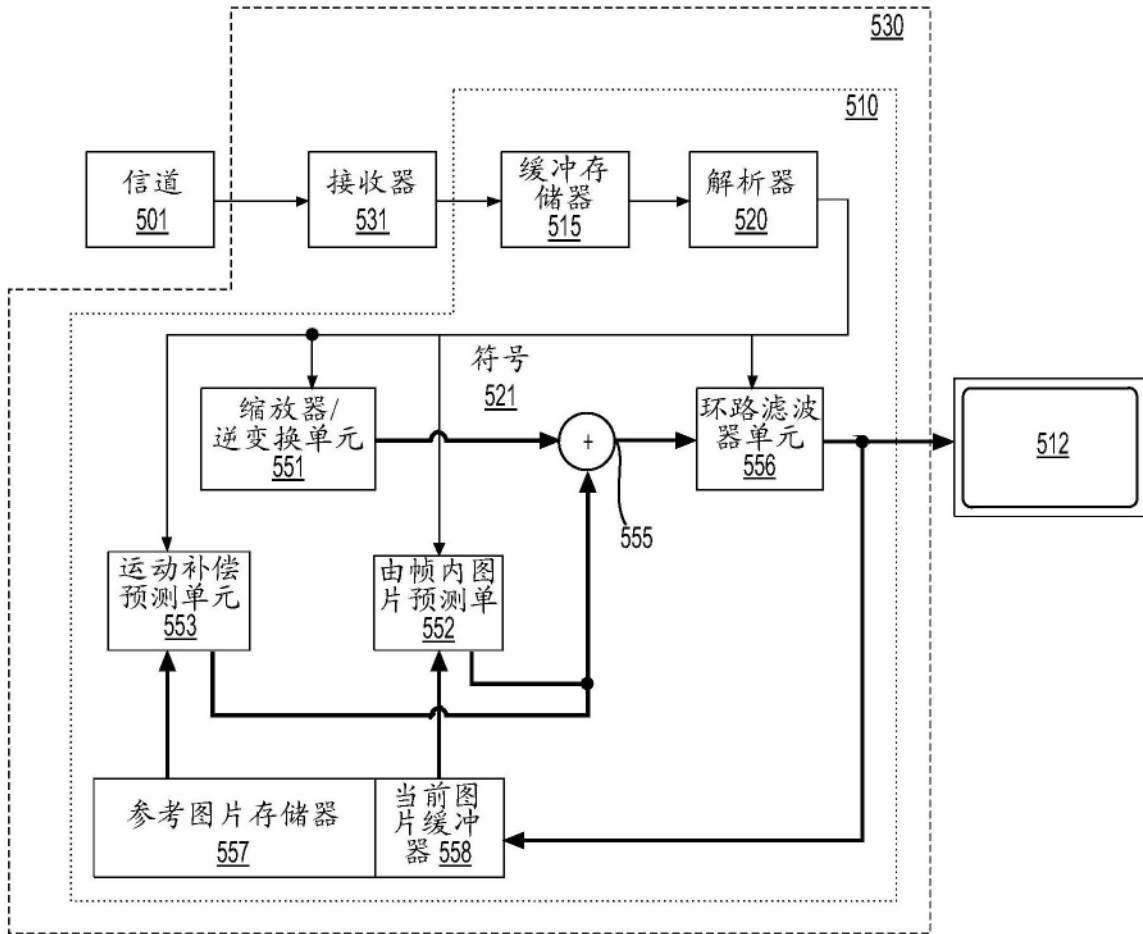


图5

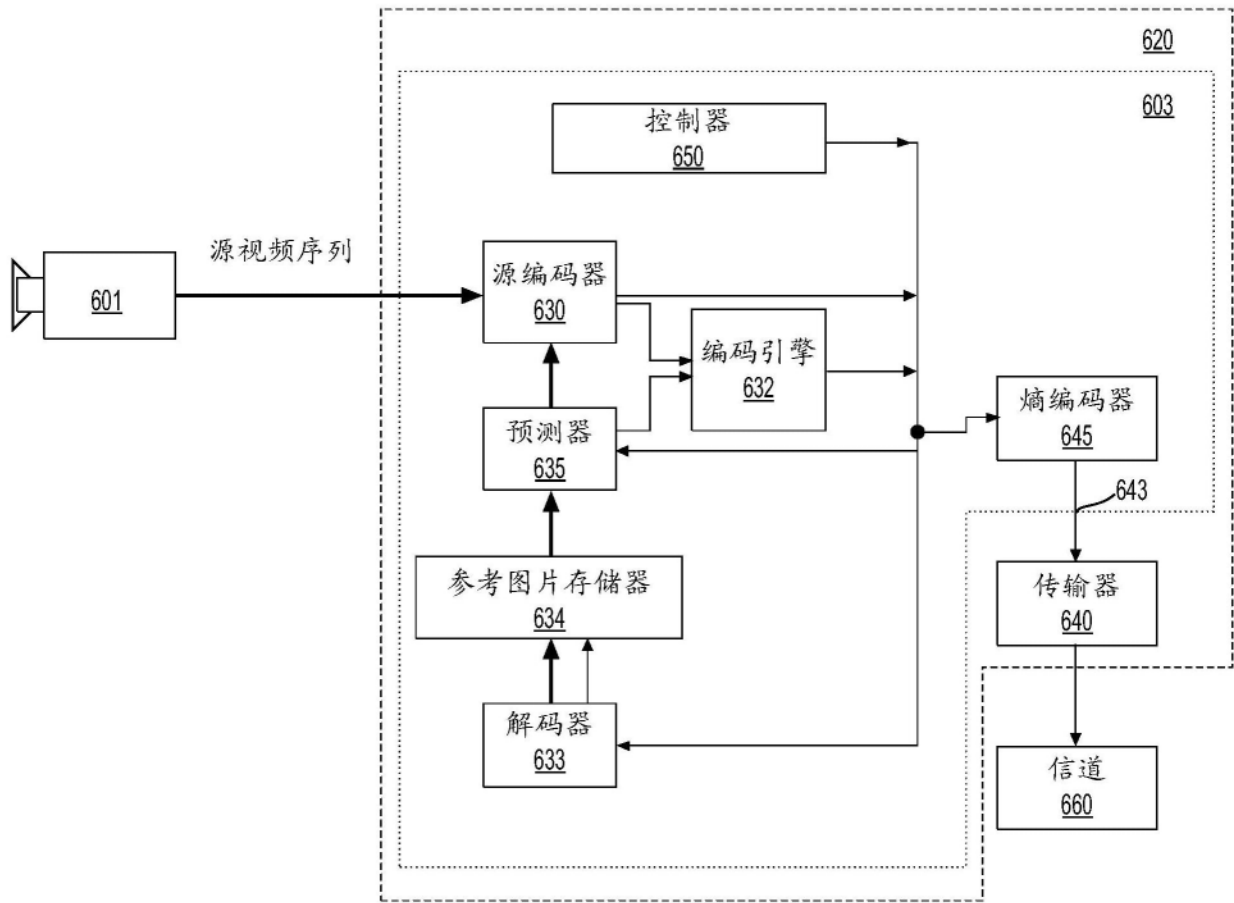


图6

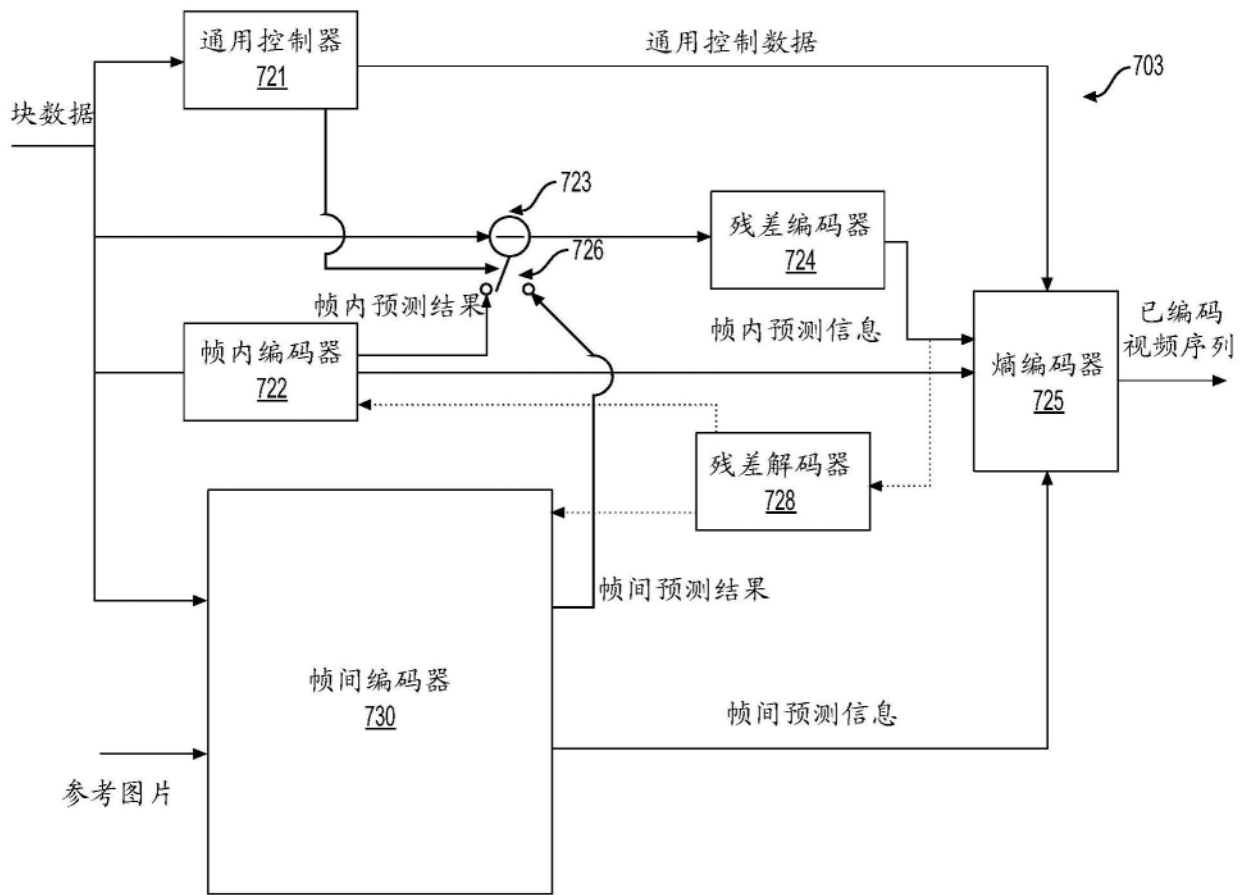


图7

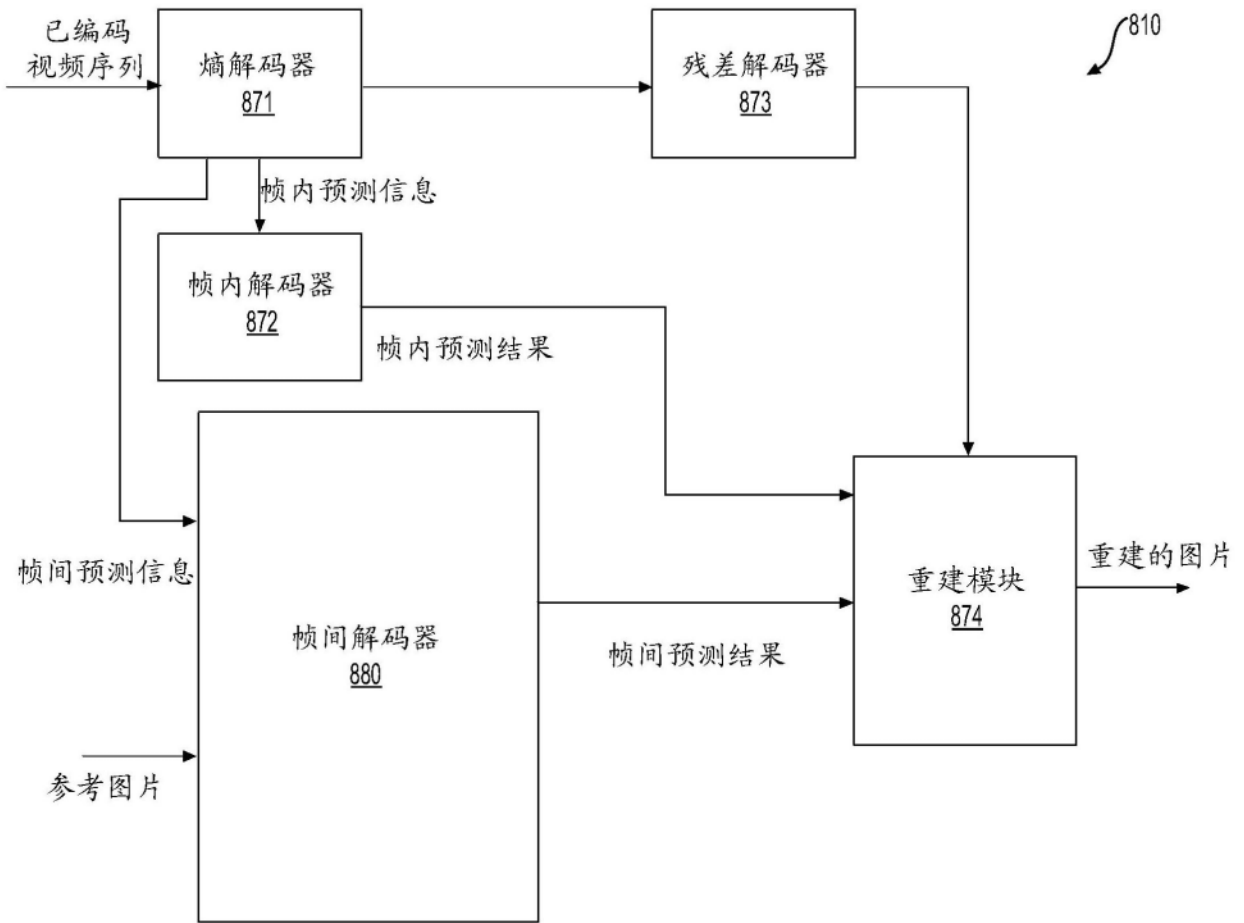


图8

	描述符
transform_unit(x0, y0, tbWidth, tbHeight, treeType) {	
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_LUMA )	
<b>tu_cbf_luma</b> [x0][y0]	ae(v)
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_CHROMA ) {	
<b>tu_cbf_cb</b> [x0][y0]	ae(v)
<b>tu_cbf_cr</b> [x0][y0]	ae(v)
}	
if( ( ( CuPredMode[x0][y0] == MODE_INTRA) && sps_mts_intra_enabled_flag   ( CuPredMode[x0][y0] == MODE_INTER) && sps_mts_inter_enabled_flag ) && tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && ( tbWidth <= 32 ) && ( tbHeight <= 32 ) )	
<b>cu_mts_flag</b> [x0][y0]	ae(v)
if( tu_cbf_luma[x0][y0] )	
residual_coding(x0, y0, log2( tbWidth ), log2( tbHeight ), 0 )	
if( tu_cbf_cb[x0][y0] )	
residual_coding(x0, y0, log2( tbWidth/ 2 ), log2( tbHeight/ 2 ), 1 )	
if( tu_cbf_cr[x0][y0] )	
residual_coding(x0, y0, log2( tbWidth/ 2 ), log2( tbHeight/ 2 ), 2 )	
}	

901

图9

if( transform_skip_enabled_flag && ( cIdx != 0    cu_mts_flag[ x0 ][ y0 ] == 0 ) && ( log2TbWidth <= 2 ) && ( log2TbHeight <= 2 ) )	
<b>transform_skip_flag</b> [ x0 ][ y0 ][ cIdx ]	ae(v)
<b>last_sig_coeff_x_prefix</b>	ae(v)
<b>last_sig_coeff_y_prefix</b>	ae(v)
if( last_sig_coeff_x_prefix > 3 )	
<b>last_sig_coeff_x_suffix</b>	ae(v)
if( last_sig_coeff_y_prefix > 3 )	
<b>last_sig_coeff_y_suffix</b>	ae(v)
.....	
if( dep_quant_enabled_flag ) {	
QState = startQStateSb	
for( n = numSbCoeff - 1; n >= 0; n-- ) {	
xC = ( xS << log2SbSize ) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 0 ]	
yC = ( yS << log2SbSize ) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 1 ]	
if( sig_coeff_flag[ xC ][ yC ] )	
TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] = ( 2 * AbsLevel[ xC ][ yC ] - ( QState > 1 ? 1 : 0 ) ) * ( 1 - 2 * coeff_sign_flag[ n ] )	
QState = QStateTransTable[ QState ][ par_level_flag[ n ] ]	

图10A

继续图10A

<pre> } else {     sumAbsLevel = 0     for( n = numSbCoeff- 1; n &gt;= 0; n - ) {         xC = ( xS &lt;&lt; log2SbSize ) +             DiagScanOrder[ log2SbSize][ log2SbSize][ n ][ 0 ]         yC = ( yS &lt;&lt; log2SbSize ) +             DiagScanOrder[ log2SbSize][ log2SbSize][ n ][ 1 ]         if( sig_coeff_flag[xC ][ yC ] ) {             TransCoeffLevel[x0 ][ y0 ][ cIdx ][ xC ][ yC ] =                 AbsLevel[ xC ][ yC ] * ( 1 - 2 * coeff_sign_flag[ n ] )             if( signHidden ) {                 sumAbsLevel += AbsLevel[xC ][ yC ]             }             if( ( n == firstSigScanPosSb ) &amp;&amp; ( sumAbsLevel%2 ) == 1 )                 TransCoeffLevel[x0 ][ y0 ][ cIdx ][ xC ][ yC ] =                     - TransCoeffLevel[x0 ][ y0 ][ cIdx ][ xC ][ yC ]         }     } } } } } } </pre>	
<pre> if( cu_mts_flag[ x0 ][ y0 ] &amp;&amp; ( cIdx == 0 ) &amp;&amp;     ! transform_skip_flag[ x0 ][ y0 ][ cIdx ] &amp;&amp;     ( ( CuPredMod[ x0 ][ y0 ] == MODE_INTRA &amp;&amp; numSigCoeff &gt; 2 )          ( CuPredMod[ x0 ][ y0 ] == MODE_INTER ) ) ) {     mts_idx[ x0 ][ y0 ] </pre>	ac(v)
<pre> } </pre>	

1001

图10B

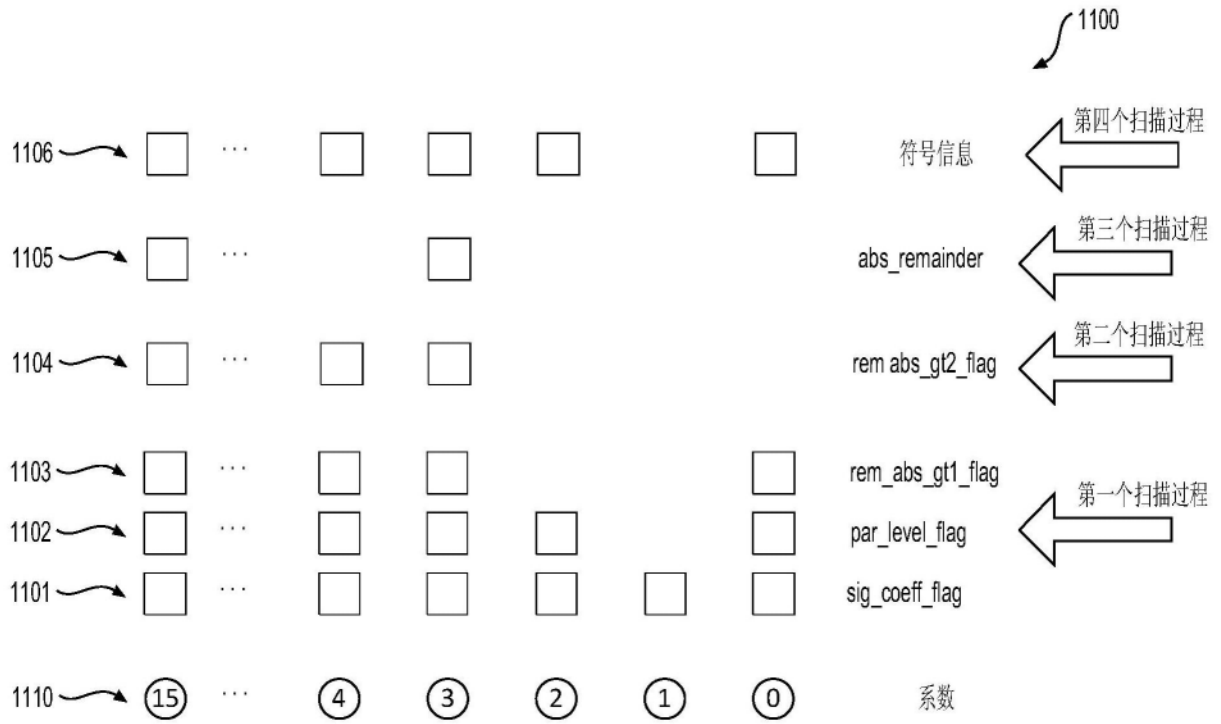


图11

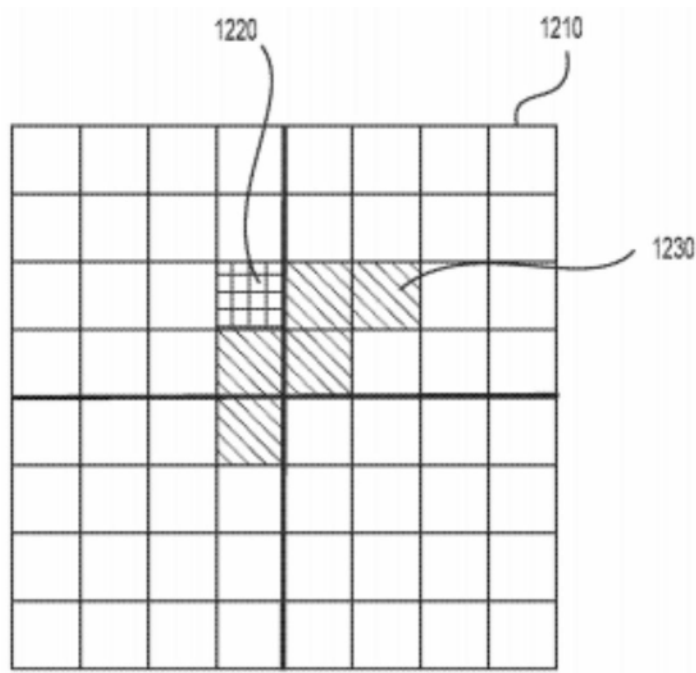


图12

	描述符
<code>residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {</code>	
<code>if( transform_skip_enabled_flag &amp;&amp; (cIdx!= 0    tu_mts_flag[x0][y0] == 0) &amp;&amp; ( log2TbWidth &lt;= 2) &amp;&amp; ( log2TbHeight &lt;= 2) )</code>	
<code>transform_skip_flag[x0][y0][cIdx]</code>	ae(v)
<code>last_sig_coeff_x_prefix</code>	ae(v)
<code>last_sig_coeff_y_prefix</code>	ae(v)
<code>if( last_sig_coeff_x_prefix &gt; 3 )</code>	
<code>last_sig_coeff_x_suffix</code>	ae(v)
<code>if( last_sig_coeff_y_prefix &gt; 3 )</code>	
<code>last_sig_coeff_y_suffix</code>	ae(v)
.....	
<code>for( i = lastSubBlock; i &gt;= 0?i? ) {</code>	
.....	
<code>if( ( i &lt; lastSubBlock ) &amp;&amp; ( i &gt; 0 ) ) {</code>	
<code>if( ( ( tu_mts_flag[ x0 ][ y0 ] == 0    cIdx != 0 ) &amp;&amp; ( xS &lt;&lt; log2SbSize ) &lt; 32 &amp;&amp; ( yS &lt;&lt; log2SbSize ) &lt; 32 )    ( tu_mts_flag[ x0 ][ y0 ] == 1 &amp;&amp; ( xS &lt;&lt; log2SbSize ) &lt; 16 &amp;&amp; ( yS &lt;&lt; log2SbSize ) &lt; 16 ) )</code>	
<code>coded_sub_block_flag[xS][yS]</code>	ae(v)
<code>inferSbDcSigCoeffFlag = 1</code>	
<code>}</code>	
.....	
<code>}</code>	
<code>if( tu_mts_flag[x0][y0] &amp;&amp; ( cIdx == 0 ) )</code>	
<code>mts_idx[x0][y0][cIdx]</code>	ae(v)
<code>}</code>	

1301

图13

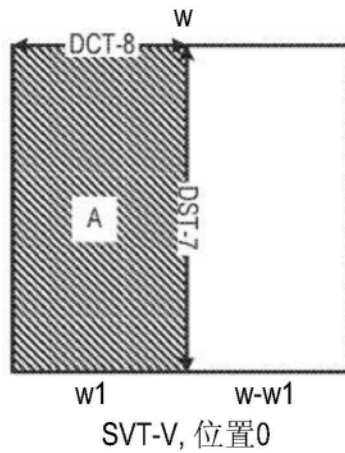


图14A

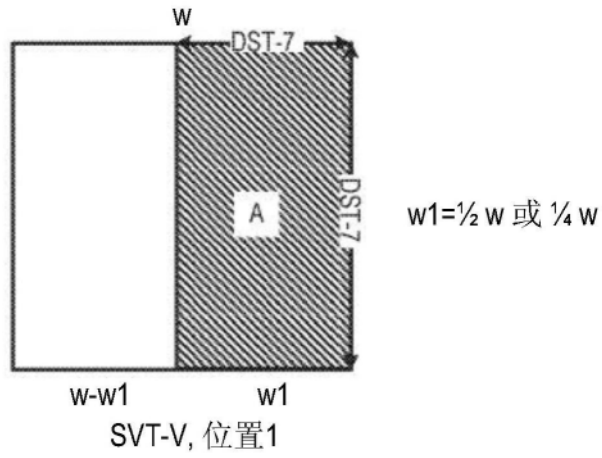


图14B

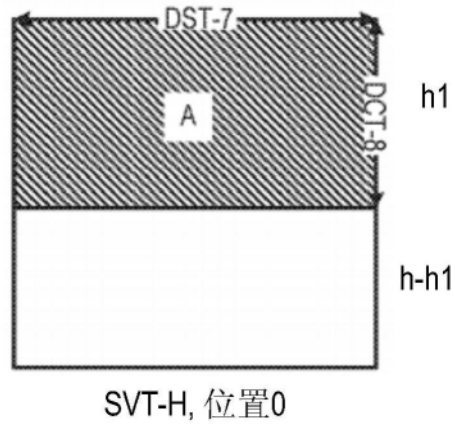


图14C

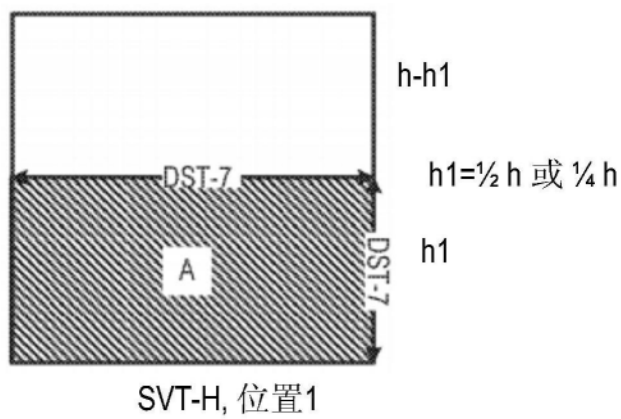


图14D

序列参数集Rbsp语法

seq_parameter_set_rbsp() {	描述符
<b>sps_seq_parameter_set_id</b>	ue(v)
...	
<b>sps_mts_intra_enabled_flag</b>	u(1)
<b>sps_mts_inter_enabled_flag</b>	u(1)
<b>sps_sbt_enable_flag</b>	u(1)
rbsp_trailing_bits()	
}	

2101

图15A

常规切片头语法

slice_header() {	描述符
<b>slice_pic_parameter_set_id</b>	ue(v)
<b>slice_address</b>	u(v)
<b>slice_type</b>	ue(v)
if ( slice_type != I ) {	
<b>log2_diff_ctu_max_bt_size</b>	ue(v)
if( sps_sbtmvp_enabled_flag ) {	
<b>sbtmvp_size_override_flag</b>	u(1)
if( sbtmvp_size_override_flag )	
<b>log2_sbtmvp_active_size_minus2</b>	u(3)
}	
if( sps_temporal_mvp_enabled_flag )	
<b>slice_temporal_mvp_enabled_flag</b>	u(1)
if( slice_type == B )	
<b>mvd_l1_zero_flag</b>	u(1)
if( slice_temporal_mvp_enabled_flag ) {	
if( slice_type == B )	
<b>collocated_from_10_flag</b>	u(1)
}	
<b>six_minus_max_num_merge_cand</b>	ue(v)
if( sps_sbt_enable_flag )	
<b>slice_max_sbt_size_64_flag</b>	u(1)
}	
if ( sps_alf_enabled_flag ) {	
<b>slice_alf_enabled_flag</b>	u(1)
if( slice_alf_enabled_flag )	
alf_data()	
}	
<b>dep_quant_enabled_flag</b>	u(1)
if( !dep_quant_enabled_flag )	
<b>sign_data_hiding_enabled_flag</b>	u(1)
byte_alignment()	
}	

2102

图15B

编码单元语法

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
...	
if( CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0 )	
<b>cu_cbf</b>	ae(v)
if( cu_cbf ) {	
if( CuPredMode[x0][y0] != MODE_INTRA && sps_sbt_enable_flag ) {	
if( cbWidth <= maxSbtSize && cbHeight <= maxSbtSize ) {	
allowSbtVerHalf = cbWidth >= 8	
allowSbtVerQuad = cbWidth >= 16	
allowSbtHorHalf = cbHeight >= 8	
allowSbtHorQuad = cbHeight >= 16	
if( allowSbtVerHalf    allowSbtHorHalf    allowSbtVerQuad    allowSbtHorQuad )	
<b>cu_sbt_flag[x0][y0]</b>	ae(v)
}	
if( cu_sbt_flag[x0][y0] ) {	
if( ( allowSbtVerHalf    allowSbtHorHalf ) && ( allowSbtVerQuad    allowSbtHorQuad ) )	
<b>cu_sbt_quad_flag[x0][y0]</b>	ae(v)
if( ( cu_sbt_quad_flag[x0][y0] && allowSbtVerQuad && allowSbtHorQuad )    ( !cu_sbt_quad_flag[x0][y0] && allowSbtVerHalf && allowSbtHorHalf ) )	
<b>cu_sbt_horizontal_flag[x0][y0]</b>	ae(v)
<b>cu_sbt_pos_flag[x0][y0]</b>	ae(v)
}	
}	
transform_tree( x0, y0, cbWidth, cbHeight, treeType )	
}	
}	

2103

图15C

变换树语法

	描述符
transform_tree(x0, y0, tbWidth, tbHeight, treeType) {	
if( tbWidth > MaxTbSizeY    tbHeight > MaxTbSizeY) {	
trafoWidth = ( tbWidth > MaxTbSizeY ) ? (tbWidth / 2) : tbWidth	
trafoHeight = ( tbHeight > MaxTbSizeY ) ? (tbHeight / 2) : tbHeight	
transform_tree(x0, y0, trafoWidth, trafoHeight)	
if( tbWidth > MaxTbSizeY)	
transform_tree( x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType )	
if( tbHeight > MaxTbSizeY)	
transform_tree( x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType )	
if( tbWidth > MaxTbSizeY && tbHeight > MaxTbSizeY)	
transform_tree( x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType )	
} else if( cu_sbt_flag[ x0 ][ y0 ] )	
factorTb0 = cu_sbt_quad_flag[ x0 ][ y0 ] ? 1 : 2	
factorTb0 = cu_sbt_pos_flag[ x0 ][ y0 ] ? ( 4 - factorTb0 ) : factorTb0	
noResiTb0 = cu_sbt_pos_flag[ x0 ][ y0 ] ? 1 : 0	
if( !cu_sbt_horizontal_flag[ x0 ][ y0 ] ) {	
trafoWidth = tbWidth * factorTb0 / 4	
transform_tree( x0, y0, trafoWidth, tbHeight, treeType, noResiTb0)	
transform_tree( x0 + trafoWidth, y0, tbWidth - trafoWidth, tbHeight, treeType, !noResiTb0)	
}	
else {	
trafoHeight = tbHeight * factorTb0 / 4	
transform_tree( x0, y0, tbWidth, trafoHeight, treeType, noResiTb0)	
transform_tree( x0, y0 + trafoHeight, tbWidth, tbHeight - trafoHeight, treeType, !noResiTb0)	
}	
} else {	
transform_unit(x0, y0, tbWidth, tbHeight, treeType, 0)	
}	
}	

2104

图15D

变换单元语法

	描述符
<code>transform_unit( x0, y0, tbWidth, tbHeight, treeType, noResi ) {</code>	
<code>if( ( treeType == SINGLE_TREE    treeType == DUAL_TREE_LUMA ) &amp;&amp; !noResi )</code>	
<code>tu_cbf_luma[ x0 ][ y0 ]</code>	ae(v)
<code>if( ( treeType == SINGLE_TREE    treeType == DUAL_TREE_CHROMA ) &amp;&amp; !noResi ) {</code>	
<code>tu_cbf_cb[ x0 ][ y0 ]</code>	ae(v)
<code>tu_cbf_cr[ x0 ][ y0 ]</code>	ae(v)
<code>}</code>	
<code>if( ( ( ( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) &amp;&amp; sps_mts_intra_enabled_flag )   </code>	
<code>( ( CuPredMode[ x0 ][ y0 ] == MODE_INTER ) &amp;&amp; sps_mts_inter_enabled_flag )</code>	
<code>&amp;&amp; tu_cbf_luma[ x0 ][ y0 ] &amp;&amp; treeType != DUAL_TREE_CHROMA</code>	
<code>&amp;&amp; ( tbWidth &lt;= 32 ) &amp;&amp; ( tbHeight &lt;= 32 ) &amp;&amp; !cu_sbt_flag[ x0 ][ y0 ] )</code>	
<code>cu_mts_flag[ x0 ][ y0 ]</code>	ae(v)
<code>if( tu_cbf_luma[ x0 ][ y0 ] )</code>	
<code>residual_coding( x0, y0, log2( tbWidth ), log2( tbHeight ), 0 )</code>	
<code>if( tu_cbf_cb[ x0 ][ y0 ] )</code>	
<code>residual_coding( x0, y0, log2( tbWidth / 2 ), log2( tbHeight / 2 ), 1 )</code>	
<code>if( tu_cbf_cr[ x0 ][ y0 ] )</code>	
<code>residual_coding( x0, y0, log2( tbWidth / 2 ), log2( tbHeight / 2 ), 2 )</code>	
<code>}</code>	

2107

图15E

表 3

	cu_sbt_horizontal_flag [ xTbY ][ yTbY ] == 0		cu_sbt_horizontal_flag [ xTbY ][ yTbY ] == 1	
	cu_sbt_pos_flag [ xTbY ][ yTbY ] == 0	cu_sbt_pos_flag [ xTbY ][ yTbY ] == 1	cu_sbt_pos_flag [ xTbY ][ yTbY ] == 0	cu_sbt_pos_flag [ xTbY ][ yTbY ] == 1
trTypeHor	2	1	nTbW > 32 ? 0 : 1	nTbW > 32 ? 0 : 1
trTypeVer	nTbH > 32 ? 0 : 1	nTbH > 32 ? 0 : 1	2	1

2108

表 4

mts_idx[ xTbY ][ yTbY ]	CuPredMode[ xTbY ][ yTbY ] == MODE_INTRA		CuPredMode[ xTbY ][ xTbY ] == MODE_INTER	
	trTypeHor	trTypeVer	trTypeHor	trTypeVer
-1 ( 推断 )	0	0	0	0
0 (00)	1	1	2	2
1 (01)	2	1	1	2
2 (10)	1	2	2	1
3 (11)	2	2	1	1

图15F

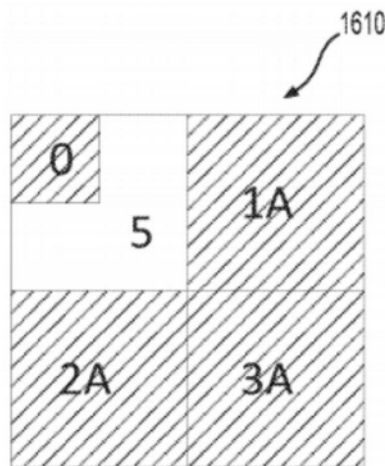


图16A

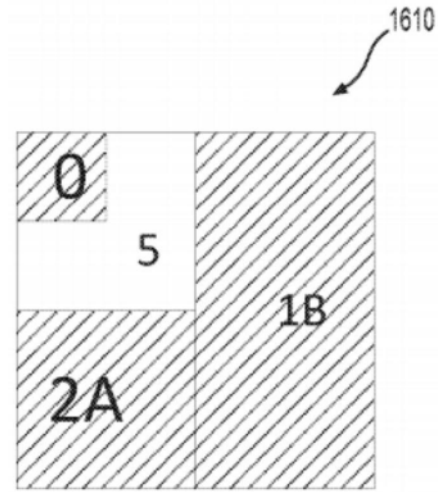


图16B

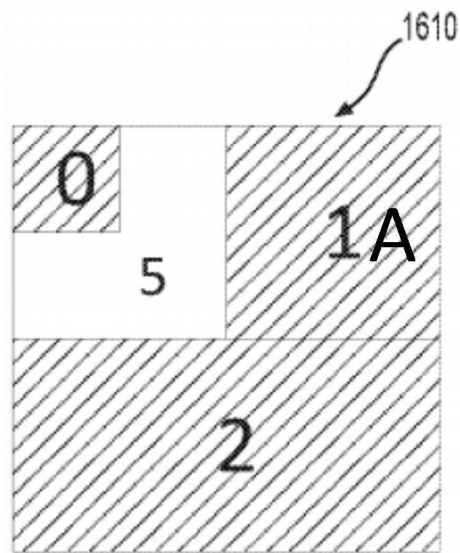


图16C

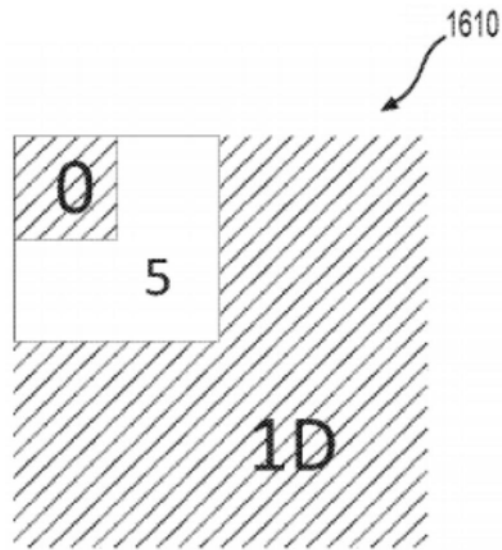


图16D

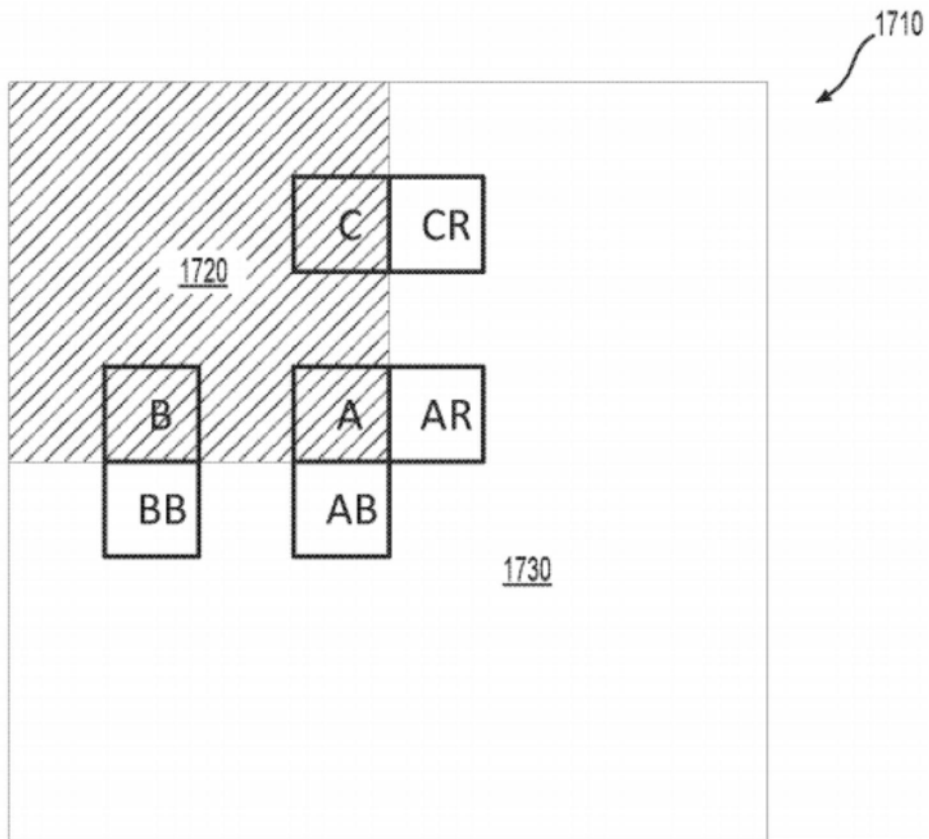


图17

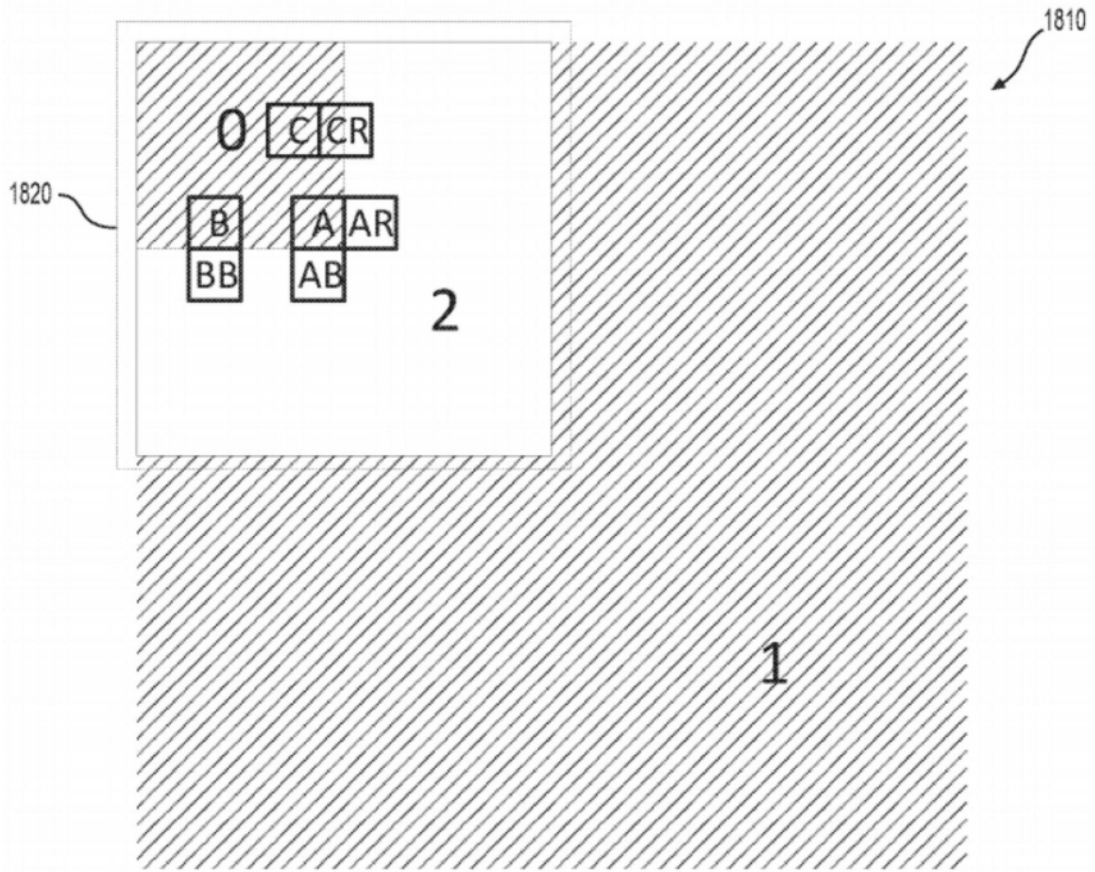


图18

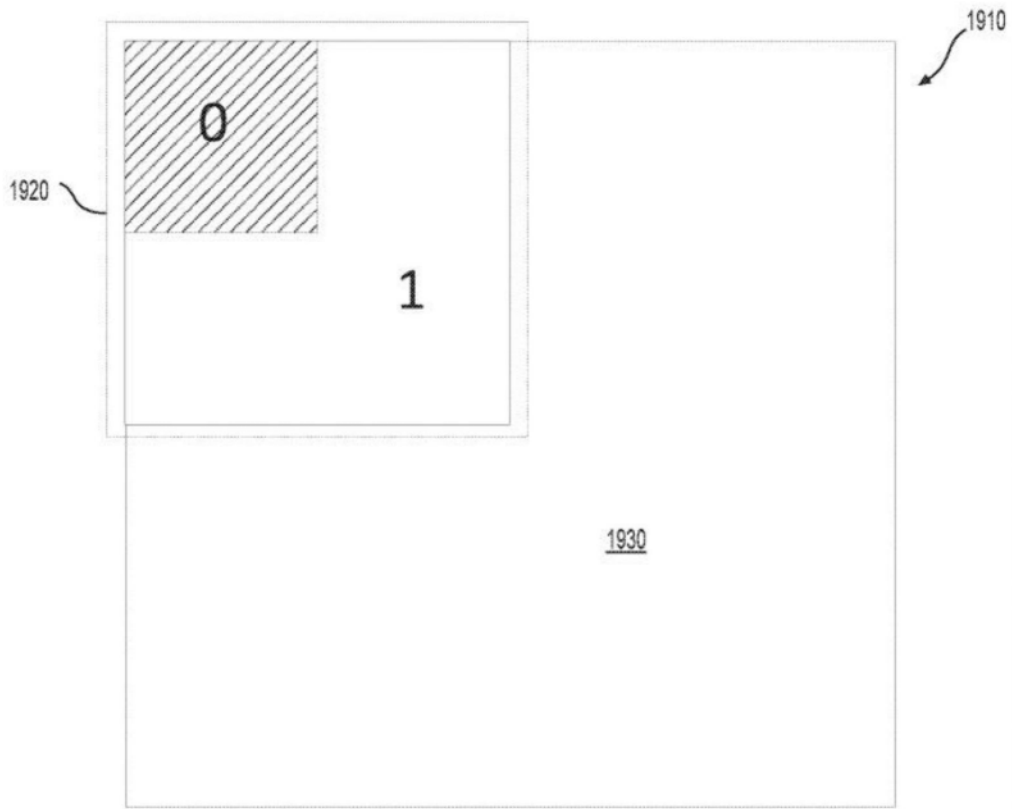


图19

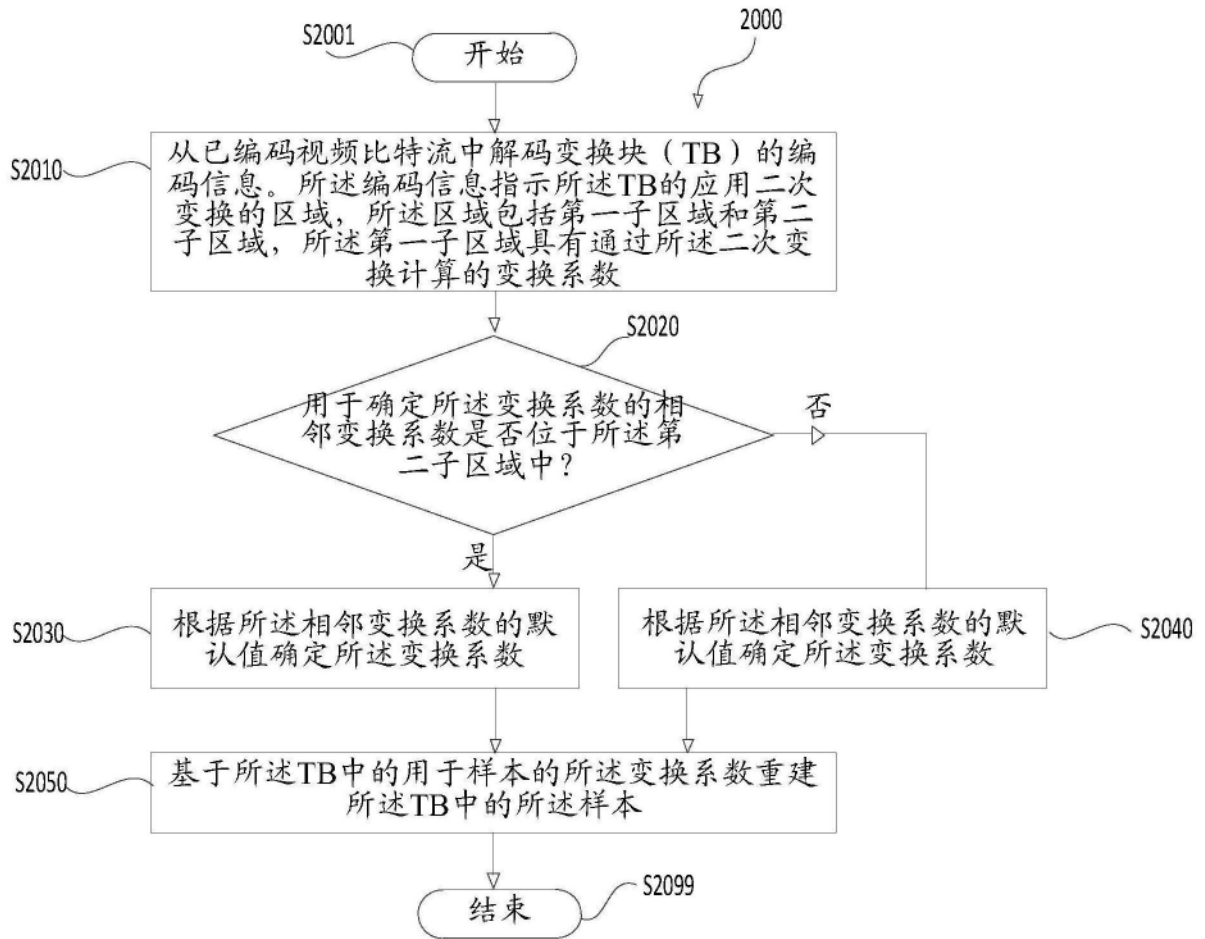


图20

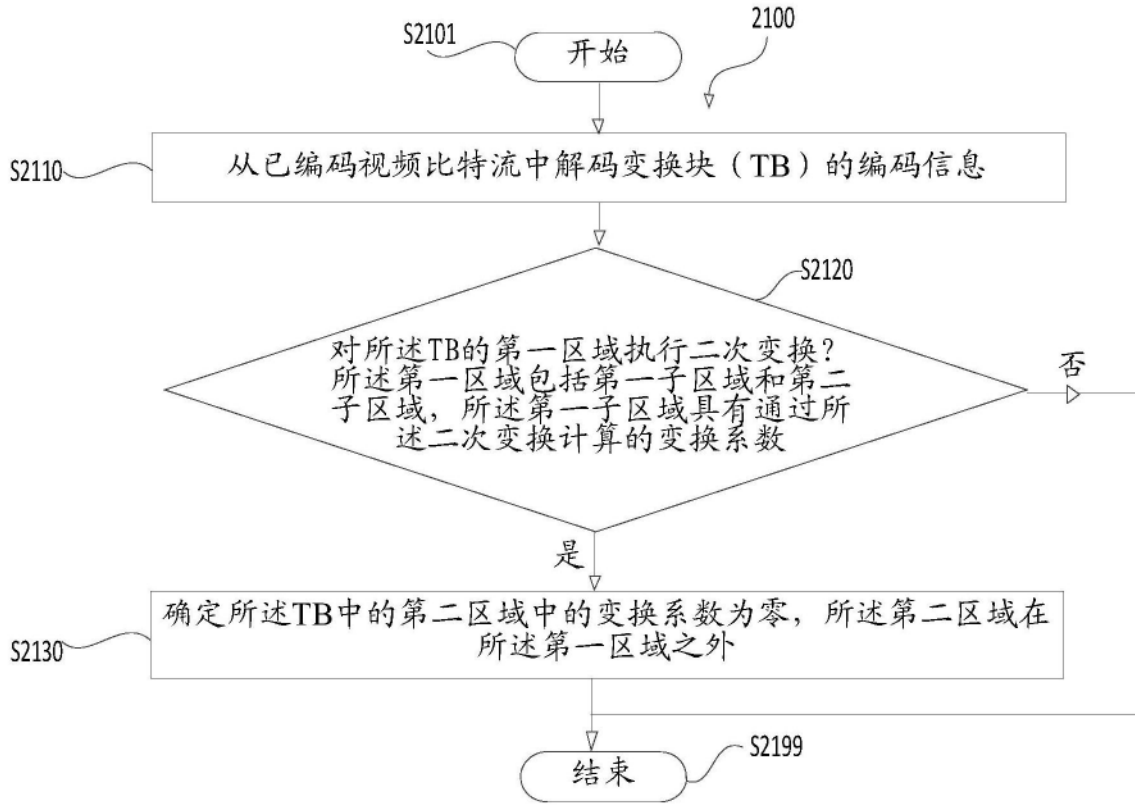


图21

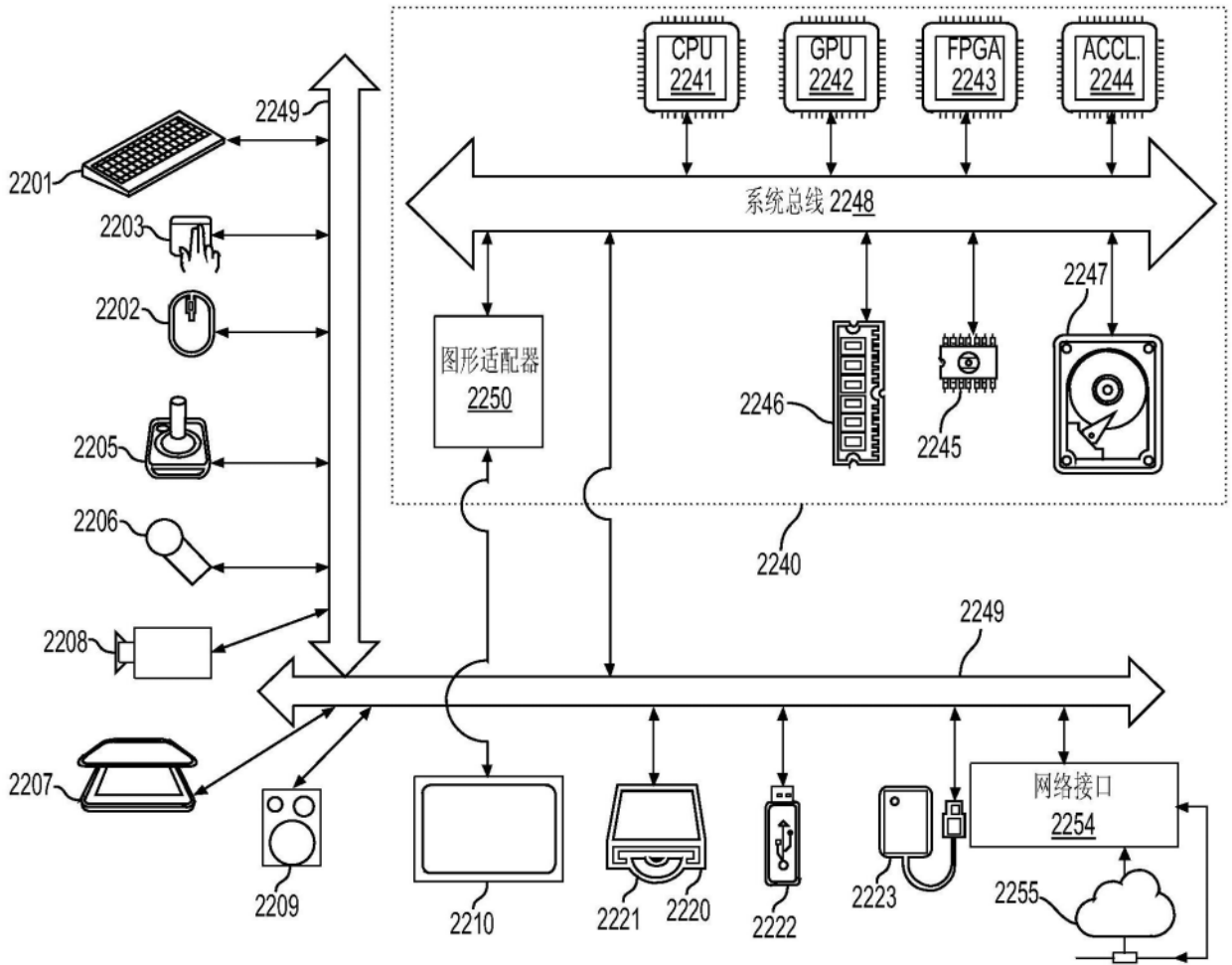


图22