



(12) 发明专利

(10) 授权公告号 CN 113614736 B

(45) 授权公告日 2024. 07. 19

(21) 申请号 202080023049.2  
 (22) 申请日 2020.05.22  
 (65) 同一申请的已公布的文献号  
 申请公布号 CN 113614736 A  
 (43) 申请公布日 2021.11.05  
 (30) 优先权数据  
 62/860,149 2019.06.11 US  
 16/878,390 2020.05.19 US  
 (85) PCT国际申请进入国家阶段日  
 2021.09.18  
 (86) PCT国际申请的申请数据  
 PCT/US2020/034271 2020.05.22  
 (87) PCT国际申请的公布数据  
 W02020/251743 EN 2020.12.17

(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 刘杉 赵欣 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 19/122 (2014.01)  
 H04N 19/159 (2014.01)  
 H04N 19/176 (2014.01)  
 H04N 19/196 (2014.01)  
 H04N 19/61 (2014.01)  
 H04N 19/70 (2014.01)  
 审查员 曹俊杰

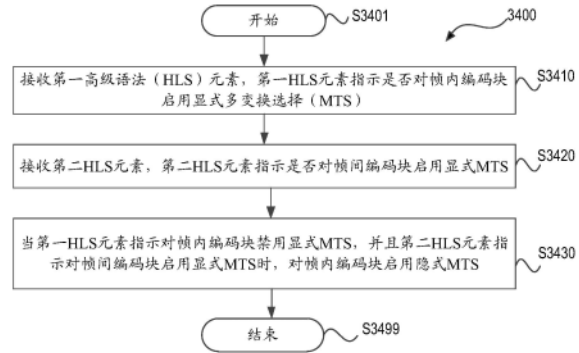
权利要求书3页 说明书28页 附图35页

(54) 发明名称

视频编解码的方法和装置

(57) 摘要

本公开的实施例提供了视频编解码的方法和装置。在视频解码器处进行视频解码的方法中,接收第一高级语法 (HLS) 元素和第二HLS元素。所述第一HLS元素指示对帧内编码块启用还是禁用显式多变换选择 (MTS)。所述第二HLS元素指示对帧间编码块启用还是禁用所述显式MTS。所述第一HLS元素和所述第二HLS元素控制同一组编码块,所述同一组编码块包括所述帧内编码块和所述帧间编码块。当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧间编码块启用所述显式MTS时,对所述帧内编码块启用隐式MTS。



1. 一种视频解码的方法,其特征在于,包括:

接收第一高级语法HLS元素,所述第一HLS元素指示对帧内编码块启用还是禁用显式多变换选择MTS;

接收第二HLS元素,所述第二HLS元素指示对帧间编码块启用还是禁用所述显式MTS,其中所述第一HLS元素和所述第二HLS元素控制同一组编码块,所述同一组编码块包括所述帧内编码块和所述帧间编码块;以及

当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧间编码块启用所述显式MTS时,对所述帧内编码块启用隐式MTS。

2. 根据权利要求1所述的方法,其特征在于,进一步包括:

将所述隐式MTS应用于所述帧内编码块,其中用于处理所述帧内编码块的变换类型根据所述帧内编码块的大小来确定。

3. 根据权利要求1所述的方法,其特征在于,所述第一HLS元素或所述第二HLS元素为以下之一:

视频参数集VPS语法元素,  
序列参数集SPS语法元素,  
图片参数集PPS语法元素,  
切片头语法元素,  
图块头语法元素,或  
图块组头语法元素。

4. 根据权利要求1所述的方法,其特征在于,所述帧内编码块不使用帧内子分区ISP模式进行编码。

5. 根据权利要求1-4任一项所述的方法,其特征在于,进一步包括:

接收第三HLS元素,所述第三HLS元素指示对所述帧间编码块和所述帧内编码块中的每一个启用MTS。

6. 一种视频解码的方法,其特征在于,包括:

接收第一高级语法HLS元素,所述第一HLS元素指示对帧内编码块启用还是禁用显式多变换选择MTS;

接收第二HLS元素,所述第二HLS元素指示对所述帧内编码块禁用还是启用不可分离二次变换NSST或基于矩阵的帧内预测MIP;以及

当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧内编码块禁用所述NSST或所述MIP时,对所述帧内编码块启用隐式MTS,所述帧内编码块使用帧内子分区ISP模式进行编码。

7. 根据权利要求6所述的方法,其特征在于,进一步包括:

当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧内编码块启用所述NSST或所述MIP时,对所述帧内编码块禁用所述隐式MTS。

8. 根据权利要求6所述的方法,其特征在于,所述启用包括:

当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧内编码块禁用所述NSST和所述MIP时,对所述帧内编码块启用所述隐式MTS。

9. 根据权利要求6所述的方法,其特征在于,进一步包括:

将所述隐式MTS应用于所述帧内编码块。

10. 根据权利要求6所述的方法, 其特征在于, 所述第一HLS元素或所述第二HLS元素为以下之一:

视频参数集VPS语法元素,  
序列参数集SPS语法元素,  
图片参数集PPS语法元素,  
切片头语法元素,  
图块头语法元素, 或  
图块组头语法元素。

11. 根据权利要求6-10任一项所述的方法, 其特征在于, 进一步包括:

接收第三HLS元素, 所述第三HLS元素指示对所述帧内编码块启用MTS。

12. 一种视频解码的方法, 其特征在于, 包括:

接收与第一块级别语法元素和第二块级别语法元素相关联的帧内编码块, 所述第一块级别语法元素指示是否应用多变换选择MTS, 所述第二块级别语法元素指示是否应用不可分离二次变换NSST; 以及

当所述第一块级别语法元素指示不应用所述MTS, 并且所述第二块级别语法元素指示不应用所述NSST时, 对所述帧内编码块启用隐式MTS, 所述帧内编码块使用帧内子分区ISP模式进行编码。

13. 根据权利要求12所述的方法, 其特征在于, 进一步包括:

当所述第一块级别语法元素指示不应用所述MTS, 并且所述第二块级别语法元素指示应用所述NSST时, 对所述帧内编码块禁用所述隐式MTS。

14. 根据权利要求12所述的方法, 其特征在于, 所述帧内编码块与第三块级别语法元素相关联, 所述第三块级别语法元素指示是否应用基于矩阵的帧内预测MIP; 以及

所述启用进一步包括: 当所述第一块级别语法元素指示不应用所述MTS, 所述第二块级别语法元素指示不应用所述NSST, 并且所述第三块级别语法元素指示不应用所述MIP时, 对所述帧内编码块启用所述隐式MTS。

15. 根据权利要求14所述的方法, 其特征在于, 进一步包括:

当所述第一块级别语法元素指示不应用所述MTS, 并且所述第二块级别语法元素指示应用所述NSST或所述第三块级别语法元素指示应用所述MIP时, 对所述帧内编码块禁用所述隐式MTS。

16. 根据权利要求12所述的方法, 其特征在于, 所述第一块级别语法元素通过指示将DCT-2变换用于处理所述帧内编码块来指示不应用所述MTS。

17. 根据权利要求12所述的方法, 其特征在于, 进一步包括:

将所述隐式MTS应用于所述帧内编码块。

18. 根据权利要求12-17任一项所述的方法, 其特征在于, 进一步包括:

接收高级语法HLS元素, 所述HLS元素指示对所述帧内编码块启用MTS。

19. 一种用于生成视频码流的编码方法, 其特征在于, 包括:

发信号通知第一高级语法HLS元素, 所述第一HLS元素指示对帧内编码块启用还是禁用显式多变换选择MTS;

发信号通知第二HLS元素,所述第二HLS元素指示对帧间编码块启用还是禁用所述显式MTS,其中所述第一HLS元素和所述第二HLS元素控制同一组编码块,所述同一组编码块包括所述帧内编码块和所述帧间编码块;以及

当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧间编码块启用所述显式MTS时,对所述帧内编码块启用隐式MTS。

20. 一种用于生成视频码流的编码方法,其特征在于,包括:

发信号通知第一高级语法HLS元素,所述第一HLS元素指示对帧内编码块启用还是禁用显式多变换选择MTS;

发信号通知第二HLS元素,所述第二HLS元素指示对所述帧内编码块禁用还是启用不可分离二次变换NSST或基于矩阵的帧内预测MIP;以及

当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧内编码块禁用所述NSST或所述MIP时,对所述帧内编码块启用隐式MTS,所述帧内编码块使用帧内子分区ISP模式进行编码。

21. 一种视频解码的装置,其特征在于,包括:

第一接收模块,用于接收第一高级语法HLS元素,所述第一HLS元素指示对帧内编码块启用还是禁用显式多变换选择MTS;

第二接收模块,用于接收第二HLS元素,所述第二HLS元素指示对帧间编码块启用还是禁用所述显式MTS,其中所述第一HLS元素和所述第二HLS元素控制同一组编码块,所述同一组编码块包括所述帧内编码块和所述帧间编码块;以及

启用模块,用于当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧间编码块启用所述显式MTS时,对所述帧内编码块启用隐式MTS。

22. 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1-18任一项所述的方法。

23. 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19或20所述的编码方法。

24. 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解码的计算机执行时,使得所述计算机执行如权利要求1-18任一项所述的方法。

## 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5月19日提交的、申请号为16/878,390、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6月11日提交的、申请号为62/860,149、名称为“启用隐式变换选择的高级语法控制”的美国临时申请的优先权。在先申请的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] 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)。

[0012] 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。

[0013] 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。

[0014] 参照图1,当前块(101)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102到106)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

[0015] 本公开的各方面提供了一种在视频解码器处进行视频解码的第一方法。所述方法可以包括:接收第一高级语法(HLS)元素,所述第一HLS元素指示对帧内编码块启用还是禁

用显式多变换选择 (MTS) ;接收第二HLS元素,所述第二HLS元素指示对帧间编码块启用还是禁用所述显式MTS。所述第一HLS元素和所述第二HLS元素控制同一组编码块,所述同一组编码块包括所述帧内编码块和所述帧间编码块。当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧间编码块启用所述显式MTS时,对所述帧内编码块启用隐式MTS。

[0016] 所述方法的实施例可以进一步包括将所述隐式MTS应用于所述帧内编码块。用于处理所述帧内编码块的变换类型根据所述帧内编码块的大小来确定。在各种示例中,所述第一HLS元素或所述第二HLS元素为以下之一:视频参数集 (VPS) 语法元素,序列参数集 (SPS) 语法元素,图片参数集 (PPS) 语法元素,切片头语法元素,图块头语法元素,或图块组头语法元素。在实施例中,所述帧内编码块不使用帧内子分区 (ISP) 模式进行编码。在实施例中,所述方法可以进一步包括:接收第三HLS元素,所述第三HLS元素指示对所述帧间编码块和所述帧内编码块中的每一个启用MTS。

[0017] 本公开的各方面提供了一种在视频解码器处进行视频解码的第二方法。所述第二方法可以包括:接收第一HLS元素,所述第一HLS元素指示对帧内编码块启用还是禁用显式MTS;接收第二HLS元素,所述第二HLS元素指示对所述帧内编码块禁用还是启用不可分离二次变换 (NSST) 或基于矩阵的帧内预测 (MIP)。当所述第一HLS元素指示对所述帧内编码块禁用所述显式MTS,并且所述第二HLS元素指示对所述帧内编码块禁用所述NSST或所述MIP时,对所述帧内编码块启用隐式MTS。

[0018] 本公开还提供了一种在视频解码器处进行视频解码的第三方法。所述第三方法可以包括:接收与第一块级别语法元素和第二块级别语法元素相关联的帧内编码块,所述第一块级别语法元素指示是否应用MTS,所述第二块级别语法元素指示是否应用NSST。当所述第一块级别语法元素指示不应用所述MTS,并且所述第二块级别语法元素指示不应用所述NSST时,对所述帧内编码块启用隐式MTS。

[0019] 注意,虽然本申请涉及NSST,但所公开的方法和系统可以应用于NSST的变体,例如缩减大小变换 (RST) 和低频不可分离二次变换 (LFNST)。因此,NSST、RST和/或LFNST可以在整个本申请中互换使用。

## 附图说明

[0020] 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,

[0021] 其中:

[0022] 图1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。

[0023] 图2是根据实施例的通信系统的简化框图的示意图。

[0024] 图3是根据实施例的通信系统的简化框图的示意图。

[0025] 图4是根据实施例的解码器的简化框图的示意图。

[0026] 图5是根据实施例的编码器的简化框图的示意图。

[0027] 图6示出了根据另一实施例的编码器的框图。

[0028] 图7示出了根据另一实施例的解码器的框图。

[0029] 图8A-8D分别示出了根据实施例的4点、8点、16点和32点DCT-2的示例变换核矩阵。

- [0030] 图9A-9D示出了根据实施例的子块变换 (SBT) 中支持的子块类型、大小和位置。
- [0031] 图10示出了根据实施例的帧内子分区 (ISP) 编码模式中取决于块大小的子分区的数量。
- [0032] 图11示出了在ISP编码模式中将块划分为两个子分区的示例。
- [0033] 图12示出了在ISP编码模式中将块划分为四个子分区的示例。
- [0034] 图13A-13E示出了根据实施例的64点DCT-2变换的64x64变换核矩阵。
- [0035] 图14示出了根据实施例的DST/DCT变换的变换基函数。
- [0036] 图15示出了根据实施例的图示mts\_idx值与相应的水平或垂直变换之间的映射关系的表。
- [0037] 图16A-16D示出了根据实施例的DST-7变换类型的变换核矩阵。
- [0038] 图17A-17D示出了根据实施例的DCT-8变换类型的变换核矩阵。
- [0039] 图18示出了使用序列参数集 (SPS) 语法元素控制多变换选择 (MTS) 的使用的示例。
- [0040] 图19示出了根据实施例的帧内预测模式和变换集之间的映射表。
- [0041] 图20-21示出了根据实施例的分别使用16x64变换核和16x48变换核的RST8x8的两种可选变换编码过程。
- [0042] 图22示出了示例的CU级别语法表,其中在CU级别语法的末尾发信号通知语法元素 lfnst\_idx,其指示低频不可分离二次变换 (LFNST) 内核的选择。
- [0043] 图23示出了根据实施例的缩减变换的过程和缩减逆变换的过程。
- [0044] 图24A示出了残差块的整个左上8x8系数 (阴影子块),其用作计算RST8x8中的二次变换的输入。
- [0045] 图24B示出了残差块的左上三个4x4子块系数 (阴影子块),其用作计算RST8x8中的二次变换的输入。
- [0046] 图25示出了根据实施例的基于帧内预测模式的变换集选择的表。
- [0047] 图26示出了基于矩阵的帧内预测 (MIP) 模式的示例过程。
- [0048] 图27示出了根据实施例的CU级别语法表,其中发信号通知基于矩阵的帧内预测 (MIP) 模式的标志在帧中示出。
- [0049] 图28A-28B组合示出了文本,该文本基于从比特流接收的相关语法元素指定对当前块执行显式或隐式变换选择的变换编码过程。
- [0050] 图29示出了对文本的修改,这些修改对应于隐式变换启用方案,其中用于帧内残差块的隐式变换和用于帧间残差块的显式变换可以共存。
- [0051] 图30示出了对文本的修改,该修改对应于当不可分离二次变换 (NSST) 被禁用时启用隐式变换的情形。
- [0052] 图31示出了对文本的修改,该修改对应于当MIP被禁用时启用隐式变换的情形。
- [0053] 图32示出了对文本的修改,该修改对应于MTS和NSST都不应用于当前块的情形。
- [0054] 图33示出了对文本的修改,该修改对应于MTS、NSST或MIP中的任何一个都不应用于当前块的情形。
- [0055] 图34-36示出了根据本公开的一些实施例的变换编码过程的流程图。
- [0056] 图37是根据实施例的计算机系统的示意图。

## 具体实施方式

[0057] I. 用于视频编解码的编码器和解码器

[0058] 图2示出了根据本公开实施例的通信系统(200)的简化框图。通信系统(200)包括多个终端装置,所述终端装置可通过例如网络(250)彼此通信。举例来说,通信系统(200)包括通过网络(250)互连的第一对终端装置(210)和(220)。在图2的实施例中,第一对终端装置(210)和(220)执行单向数据传输。举例来说,终端装置(210)可对视频数据(例如由终端装置(210)采集的视频图片流)进行编码以通过网络(250)传输到另一终端装置(220)。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(220)可从网络(250)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。

[0059] 在另一实施例中,通信系统(200)包括执行已编码视频数据的双向传输的第二对终端装置(230)和(240),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(230)和(240)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(250)传输到终端装置(230)和(240)中的另一终端装置。终端装置(230)和(240)中的每个终端装置还可接收由终端装置(230)和(240)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。

[0060] 在图2的实施例中,终端装置(210)、(220)、(230)和(240)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(250)表示在终端装置(210)、(220)、(230)和(240)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(250)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(250)的架构和拓扑对于本公开的操作来说可能是无关紧要的。

[0061] 作为所公开的主题的实施例,图3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。

[0062] 流式传输系统可包括采集子系统(313),所述采集子系统可包括数码相机等视频源(301),所述视频源创建未压缩的视频图片流(302)。在实施例中,视频图片流(302)包括由数码相机拍摄的样本。相较于已编码的视频数据(304)(或已编码的视频码流),视频图片流(302)被描绘为粗线以强调高数据量的视频图片流,视频图片流(302)可由电子装置(320)处理,所述电子装置(320)包括耦接到视频源(301)的视频编码器(303)。视频编码器(303)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(302),已编码的视频数据(304)(或已编码的视频码流(304))被描绘为细线以强调较低数据量的已编码的视频数据(304)(或已编码的视频码流(304)),其可存储在流式传输服务器(305)上以供将来使用。一个或多个流式传输客户端子系统,例如如图3中的客户端子系统(306)和客户端子系统(308),可访问流式传输服务器(305)以检索已编码的视频数据(304)的副本(307)和副本(309)。客户端子系统(306)可包括例如电子装置(330)中的视频解码器(310)。视频解码器(310)对已编码的视频数据的传入副本(307)进行

解码,且产生可在显示器(312)(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(311)。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(304)、视频数据(307)和视频数据(309)(例如视频码流)进行编码。这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。

[0063] 应注意,电子装置(320)和电子装置(330)可包括其它组件(未示出)。举例来说,电子装置(320)可包括视频解码器(未示出),且电子装置(330)还可包括视频编码器(未示出)。

[0064] 图4示出了根据本公开实施例的视频解码器(410)的框图。视频解码器(410)可设置在电子装置(430)中。电子装置(430)可包括接收器(431)(例如接收电路)。视频解码器(410)可用于代替图3实施例中的视频解码器(310)。

[0065] 接收器(431)可接收将由视频解码器(410)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(401)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(431)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(431)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(415)可耦接在接收器(431)与熵解码器/解析器(420)(此后称为“解析器(420)”)之间。在某些应用中,缓冲存储器(415)是视频解码器(410)的一部分。在其它情况下,所述缓冲存储器(415)可设置在视频解码器(410)外部(未标示)。而在其它情况下,视频解码器(410)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(410)的内部可配置另一缓冲存储器(415)以例如处理播出定时。而当接收器(431)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(415),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(415),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(410)外部的类似元件(未标示)中。

[0066] 视频解码器(410)可包括解析器(420)以根据已编码视频序列重建符号(421)。这些符号的类别包括用于管理视频解码器(410)的操作的信息,以及用以控制显示装置(412)(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(430)的组成部分,但可耦接到电子装置(430),如图4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(420)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(420)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(420)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。

[0067] 解析器(420)可对从缓冲存储器(415)接收的视频序列执行熵解码/解析操作,从而创建符号(421)。

[0068] 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(421)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(420)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(420)与下文的多个单元之间的此类子群控制信息流。

[0069] 除已经提及的功能块以外,视频解码器(410)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。

[0070] 第一单元是缩放器/逆变换单元(451)。缩放器/逆变换单元(451)从解析器(420)接收作为符号(421)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(451)可输出包括样本值的块,所述样本值可输入到聚合器(455)中。

[0071] 在一些情况下,缩放器/逆变换单元(451)的输出样本可属于帧内编码块;即:不使用来自先前重建的图像的预测性信息,但可使用来自当前图像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(452)提供。在一些情况下,帧内图片预测单元(452)采用从当前图片缓冲器(458)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(458)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(455)基于每个样本,将帧内预测单元(452)生成的预测信息添加到由缩放器/逆变换单元(451)提供的输出样本信息中。

[0072] 在其它情况下,缩放器/逆变换单元(451)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(453)可访问参考图片存储器(457)以提取用于预测的样本。在根据符号(421)对提取的样本进行运动补偿之后,这些样本可由聚合器(455)添加到缩放器/逆变换单元(451)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(453)从参考图片存储器(457)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(421)的形式而供运动补偿预测单元(453)使用,所述符号(421)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(457)提取的样本值的内插、运动矢量预测机制等等。

[0073] 聚合器(455)的输出样本可在环路滤波器单元(456)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(420)的符号(421)可用于环路滤波器单元(456)。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。

[0074] 环路滤波器单元(456)的输出可以是样本流,所述样本流可输出到显示装置(412)以及存储在参考图片存储器(457),以用于后续的帧间图片预测。

[0075] 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

且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(420))被识别为参考图片,则当前图片缓冲器(458)可变为参考图片存储器(457)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。

[0076] 视频解码器(410)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。

[0077] 在实施例中,接收器(431)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(410)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。

[0078] 图5示出了根据本公开实施例的视频编码器(503)的框图。视频编码器(503)设置于电子装置(520)中。电子装置(520)包括传输器(540)(例如传输电路)。视频编码器(503)可用于代替图3实施例中的视频编码器(303)。

[0079] 视频编码器(503)可从视频源(501)(并非图5实施例中的电子装置(520)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(503)编码的视频图像。在另一实施例中,视频源(501)是电子装置(520)的一部分。

[0080] 视频源(501)可提供将由视频编码器(503)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(501)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(501)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。

[0081] 根据实施例,视频编码器(503)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(543)。施行适当的编码速度是控制器(550)的一个功能。在一些实施例中,控制器(550)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(550)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(550)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(503)。

[0082] 在一些实施例中,视频编码器(503)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(530)(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(503)中的(本地)解码器(533)。解码器(533)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(534)。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(534)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。

[0083] “本地”解码器(533)的操作可与例如已在上文结合图4详细描述视频解码器(410)的“远程”解码器相同。然而,另外简要参考图4,当符号可用且熵编码器(545)和解析器(420)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(415)和解析器(420)在内的视频解码器(410)的熵解码部分,可能无法完全在本地解码器(533)中实施。

[0084] 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。

[0085] 在操作期间,在一些实施例中,源编码器(530)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(532)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。

[0086] 本地视频解码器(533)可基于源编码器(530)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(532)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5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(533)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(534)中。以此方式,视频编码器(503)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)。

[0087] 预测器(535)可针对编码引擎(532)执行预测搜索。即,对于将要编码的新图片,预测器(535)可在参考图片存储器(534)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(535)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(535)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(534)中存储的多个参考图片取得的预测参考。

[0088] 控制器(550)可管理源编码器(530)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。

[0089] 可在熵编码器(545)中对所有上述功能单元的输出进行熵编码。熵编码器(545)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。

[0090] 传输器(540)可缓冲由熵编码器(545)创建的已编码视频序列,从而为通过通信信道(560)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(540)可将来自视频编码器(503)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)。

[0091] 控制器(550)可管理视频编码器(503)的操作。在编码期间,控制器(550)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:

[0092] 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。

[0093] 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。

[0094] 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。

[0095] 源图片通常可在空间上细分成多个样本块(例如, $4 \times 4$ 、 $8 \times 8$ 、 $4 \times 8$ 或 $16 \times 16$ 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。

[0096] 视频编码器(503)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(503)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。

[0097] 在实施例中,传输器(540)可在传输已编码的视频时传输附加数据。源编码器(530)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。

[0098] 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。

[0099] 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。

[0100] 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。

[0101] 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。

[0102] 图6示出了根据本公开另一实施例的视频编码器(603)的示意图。视频编码器(603)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(603)用于代替图3实施例中的视频编码器(303)。

[0103] 在HEVC实施例中,视频编码器(603)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(603)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(603)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(603)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(603)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)。

[0104] 在图6的实施例中,视频编码器(603)包括如图6所示的耦接到一起的帧间编码器(630)、帧内编码器(622)、残差计算器(623)、开关(626)、残差编码器(624)、通用控制器(621)和熵编码器(625)。

[0105] 帧间编码器(630)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。

[0106] 帧内编码器 (622) 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 (622) 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)。

[0107] 通用控制器 (621) 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 (603) 的其它组件。在实施例中, 通用控制器 (621) 确定块的模式, 且基于所述模式将控制信号提供到开关 (626)。举例来说, 当所述模式是帧内模式时, 通用控制器 (621) 控制开关 (626) 以选择供残差计算器 (623) 使用的帧内模式结果, 且控制熵编码器 (625) 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 (621) 控制开关 (626) 以选择供残差计算器 (623) 使用的帧间预测结果, 且控制熵编码器 (625) 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。

[0108] 残差计算器 (623) 用于计算所接收的块与选自帧内编码器 (622) 或帧间编码器 (630) 的预测结果之间的差 (残差数据)。残差编码器 (624) 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 (624) 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 (603) 还包括残差解码器 (628)。残差解码器 (628) 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 (622) 和帧间编码器 (630) 使用。举例来说, 帧间编码器 (630) 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 (622) 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。

[0109] 熵编码器 (625) 用于将码流格式化以产生已编码的块。熵编码器 (625) 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 (625) 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。

[0110] 图7示出了根据本公开另一实施例的视频解码器 (710) 的示意图。视频解码器 (710) 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 (710) 用于代替图3实施例中的视频解码器 (310)。

[0111] 在图7实施例中, 视频解码器 (710) 包括如图7中所示耦接到一起的熵解码器 (771)、帧间解码器 (780)、残差解码器 (773)、重建模块 (774) 和帧内解码器 (772)。

[0112] 熵解码器 (771) 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 (772) 或帧间解码器 (780) 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 (780); 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 (772)。残差信息可经由逆量化并提

供到残差解码器(773)。

[0113] 帧间解码器(780)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。

[0114] 帧内解码器(772)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。

[0115] 残差解码器(773)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(773)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(771)提供(未标示数据路径,因为这仅仅是低量控制信息)。

[0116] 重建模块(774)用于在空间域中组合由残差解码器(773)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。

[0117] 应注意,可使用任何合适的技术来实施视频编码器(303)、视频编码器(503)和视频编码器(603)以及视频解码器(310)、视频解码器(410)和视频解码器(710)。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(303)、视频编码器(503)和视频编码器(603)以及视频解码器(310)、视频解码器(410)和视频解码器(710)。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(303)、视频编码器(503)和视频编码器(603)以及视频解码器(310)、视频解码器(410)和视频解码器(710)。

[0118] II. 变换编码技术和相关技术

[0119] 1. DCT-2主变换示例

[0120] 在一些实施例中,4点、8点、16点和32点DCT-2变换被用作主变换。图8A-8D分别示出了4点、8点、16点和32点DCT-2的变换核矩阵。这些变换核矩阵的元素可以用8位整数表示,因此这些变换核矩阵被称为8位变换核。如图所示,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。

[0121] DCT-2核矩阵显示出了对称/反对称特性。因此,可以支持所谓的“部分蝶形”的实施方式以减少运算次数(乘法、加法/减法、移位)。与不使用部分蝶形的实施方式相比,使用部分蝶形的实施方式可以获得相同的矩阵乘法结果。

[0122] 2. 子块变换编码示例

[0123] 2.1 子块变换(SBT)

[0124] 在一些实施例中,采用子块变换(SBT),也称为空间变化变换(SVT)。在一些实施例中,SBT应用于帧间预测残差。例如,可以将编码块划分为多个子块,在残差块处仅处理这些子块的一部分。这些子块的剩余部分假设为零残差。因此,残差块小于编码块,并且SBT中的变换大小小于编码块大小。对于未被残余块覆盖的区域,不执行变换处理。

[0125] 图9A-9D示出了SBT中支持的子块类型(SVT-H、SVT-V)(例如,垂直或水平分区的)、大小和位置(例如,左半部分、左四分之一部分、右半部分、右四分之一部分、上半部分、上四分之一部分、下半部分、下四分之一部分)。用字母“A”标记的阴影区域是要进行变换编码的残差块,其它区域假设为零残差而不进行变换。

[0126] 2.2. 帧内子分区(ISP)编码模式

[0127] 在一些实施例中,采用帧内子分区 (ISP) 编码模式。在ISP编码模式中,可将亮度帧内预测块垂直或水平地划分成2个或4个子分区。子分区的数量可以取决于块的大小。图10示出了取决于块大小的子分区的数量。图11示出了将块划分为两个子分区的情形。图12示出了将块划分为四个子分区的情形。在示例中,所有子分区均满足具有至少16个样本的条件。在示例中,ISP不应用于色度分量。

[0128] 在示例中,对于从编码块划分出的每个子分区,通过对编码器发送的各个系数进行熵解码然后对它们进行逆量化和逆变换,来生成残差信号。然后,对这些子分区中的第一个子分区进行帧内预测,以生成预测信号。将预测信号与第一个子分区的各个残差信号相加得到相应的重建样本。此后,第一个子分区重建样本值可用于生成这些子分区中的第二个子分区的预测。该过程可以逐个子分区重复,直到编码块中的所有子分区都被重建。在示例中,所有子分区共享相同的帧内模式。

[0129] 在实施例中,ISP编码模式仅使用作为最可能模式 (MPM) 列表的一部分的帧内模式进行测试。因此,如果一个块使用ISP,那么MPM标志可以推断为1。此外,当ISP用于某个块时,则修改相应的MPM列表,以排除DC模式,并对用于ISP水平分割的水平帧内模式和用于垂直分割的垂直帧内模式进行优先级排序。

[0130] 在ISP编码模式中,因为对每个子分区单独执行变换和重建,因此每个子分区都可以看作一个子TU。

[0131] 3. 使用扩展的DCT-2变换和多变换选择 (MTS) 进行变换编码

[0132] 在一些实施例中,当编码块的高度和宽度都小于或等于64个样本时,变换大小始终与编码块大小相同。当编码块的高度或宽度大于64个样本时,在进行变换或帧内预测时,将编码块进一步分割为多个子块,其中每个子块的宽度和高度都小于或等于64,并且对每个子块执行变换处理。

[0133] 3.1使用扩展的DCT-2变换进行变换编码

[0134] 在一些实施例中,除了上述的4点、8点、16点和32点DCT-2变换之外,还可以使用2点和64点DCT-2变换。图13A-13E示出了64点DCT-2变换的64x64变换核矩阵。

[0135] 3.2显式地发信号通知的变换

[0136] 在一些实施例中,除了DCT-2和 $4 \times 4$ DST-7变换编码之外,多变换选择 (MTS) (也称为增强多变换 (EMT) 或自适应多变换 (AMT)) 也可以用于帧间和帧内编码块的残差编码。MTS使用从离散余弦变换 (DCT) /离散正弦变换 (DST) 族中选定的、除DCT-2和 $4 \times 4$ DST-7变换之外的多个变换。该选择可以在编码器处执行,并且从编码器显式地发信号通知给解码器。例如,选定的变换可以包括DST-7或DCT-8变换。图14示出了DST/DCT变换的变换基函数。在一些实施例中,MTS中使用的DST/DCT变换核矩阵用8位整数表示。

[0137] 在一些实施例中,MTS可以应用于宽度和高度都小于或等于32个样本的CU。是否应用MTS可以由`mts_flag`表示的标志控制。例如,当`mts_flag`等于0时,仅应用DCT-2对残差块进行编码。当`mts_flag`等于1时,其指示应用了MTS,可以使用选定的变换。例如,由`mts_idx`表示的索引可以进一步使用2个二进制数 (bin) 发信号通知,以指定要使用的水平变换和垂直变换。

[0138] 图15示出了图示`mts_idx`值与相应的水平变换或垂直变换之间的映射关系的表(1500)。`mts_idx`的值为-1的行(1301)对应于`mts_flag`等于0(其指示未应用MTS)并且使用

DCT-2变换的场景。`mts_idx`的值为0、1、2或3的行(1302) - (1305)对应于`mts_flag`等于1(其指示应用MTS)的场景。在表(1500)的右两列中,0表示DCT-2的变换类型,1表示DST-7的变换类型,2表示DCT-8的变换类型。

[0139] 图16A-16D示出了DST-7变换类型的变换核矩阵。图17A-17D示出了DCT-8变换类型的变换核矩阵。

[0140] 在一些实施例中,可以使用高级语法(HLS)元素来启用或禁用MTS。每个HLS元素可以是视频参数集(VPS)语法元素、序列参数集(SPS)语法元素、图片参数集(PPS)语法元素、切片头语法元素、图块头语法元素或图块组头语法元素等。图18示出了使用SPS语法元素控制MTS的使用的示例。如图所示,可以发信号通知SPS语法元素`sps_mts_enabled_flag`,以指示是否对视频序列启用MTS。当启用MTS时,可以发信号通知两个语法元素`sps_explicit_mts_intra_enabled_flag`和`sps_explicit_mts_inter_enabled_flag`,以分别指示是否启用MTS以对帧间预测块或帧内预测块进行编码。

[0141] 在实施例中,如果不使用上述基于信令的MTS(称为显式MTS),则应用隐式MTS。使用隐式MTS,可以根据块宽度和高度而不是基于信令来进行变换选择。例如,使用隐式MTS,可以为变换块的较短边选择DST-7变换,并且可以为变换块的较长边选择DCT-2变换。

[0142] 3.3应用隐式变换选择的场景

[0143] 在各种实施例中,对于某些场景,无需显式地发信号通知就可以使用DST-7和/或DCT-8。例如,可以基于编码器和相应的解码器都可用的信息来隐式地使用DST-7和/或DCT-8。这些场景包括:

[0144] 3.3.1帧内子分区(ISP)

[0145] 对于以ISP模式编码的残差块,只要残差块的块宽度大于或等于4且小于或等于16,就将水平变换选择为DST-7,只要残差块的块高度大于或等于4且小于或等于16,就将垂直变换选择为DST-7。

[0146] 3.3.2子块变换(SBT)

[0147] 对于SBT模式,对于位于当前CU的左半部分(或四分之一部分)或右半部分(或四分之一部分)的子TU,水平变换可以分别为DCT-8或DST-7。否则(子TU具有与当前CU相同的宽度),可以使用DCT-2。对于位于当前CU的上半部分(或四分之一部分)或下半部分(或四分之一部分)的子TU,垂直变换可以分别为DCT-8或DST-7。否则(子TU具有与当前CU相同的高度),可以使用DCT-2。

[0148] 3.3.3通过HLS元素禁用MTS

[0149] 例如,当发信号通知`sps_mts_enabled_flag`为真,但发信号通知`sps_explicit_mts_intra_enabled_flag`和`sps_explicit_mts_inter_enabled_flag`均为假时,对于帧内预测残差,只要相应的块宽度大于或等于4且小于或等于16,就可以将水平变换选择为DST-7,只要相应的块高度大于或等于4且小于或等于16,就可以将垂直变换选择为DST-7。

[0150] 4.不可分离二次变换(NSST)

[0151] 4.1 NSST的初步设计

[0152] 在一些实施例中,在正向核变换与量化之间(在编码器处)以及在去量化与逆核变换之间(在相应的解码器处),可以应用模式相关的不可分离二次变换(NSST)。为了保持低复杂度,在一些实施例中,NSST只能应用于主变换后的低频系数。如果变换系数块的宽度

(W)和高度(H)均大于或等于8,则可以将 $8 \times 8$ 不可分离二次变换应用于变换系数块的左上 $8 \times 8$ 区域。否则,如果变换系数块的W或H等于4,则可以应用 $4 \times 4$ 不可分离二次变换,并且可以对变换系数块的左上 $\min(8,W) \times \min(8,H)$ 区域执行 $4 \times 4$ 不可分离变换。上述变换选择的规则可以应用于亮度分量和色度分量。

[0153] 下面以 $4 \times 4$ 输入块为例描述不可分离变换的矩阵乘法的实现。为了应用不可分离变换, $4 \times 4$ 输入块X

$$[0154] \quad X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix} \quad (\text{等式 1})$$

[0155] 表示为一个矢量 $\vec{X}$ :

$$[0156] \quad \vec{X} = [X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T \quad (\text{等式 2})$$

[0157] 不可分离变换计算为:

$$[0158] \quad \vec{F} = T \cdot \vec{X}, \quad (\text{等式 3})$$

[0159] 其中 $\vec{F}$ 指示变换系数矢量,T为 $16 \times 16$ 变换矩阵。 $16 \times 1$ 系数矢量 $\vec{F}$ 随后使用该块的扫描顺序(水平、垂直或对角线)被重新组织为 $4 \times 4$ 块。具有较小索引的系数将与较小的扫描索引一起放置在 $4 \times 4$ 系数块中。在示例中,使用具有蝶形实施方式的超立方体-吉文斯(Hypercube-Givens)变换(HyGT)来代替矩阵乘法,以降低不可分离变换的复杂度。

[0160] 在NSST的示例中,总共可以有 $35 \times 3$ 个不可分离二次变换用于 $4 \times 4$ 块大小和 $8 \times 8$ 块大小,其中35是变换集的数量,每个变换集对应于一个帧内预测模式,3是每个帧内预测模式的NSST候选的数量。在图19中所示的表(1900)中定义了从帧内预测模式到变换集的映射。例如,根据表(1900),应用于亮度/色度变换系数的变换集可以由相应的亮度/色度帧内预测模式指定。对于大于34(例如,对角线预测方向)的帧内预测模式,变换系数块在二次变换之前/之后在编码器/解码器处进行转置。

[0161] 对于每个变换集,所选择的不可分离二次变换候选由显式地发信号通知的CU级别NSST索引进一步指定。在使用变换系数和截断一元二值化(truncated unary binarization)之后,每个帧内CU在比特流中发信号通知索引一次。在平面或DC模式的情况下,截断值为2,在角度帧内预测模式的情况下,截断值为3。仅当CU中存在多于一个非零系数时,发信号通知该NSST索引。当未发信号通知时,默认值为零。该语法元素的零值指示二次变换不应用于当前CU,值1-3指示应该应用集中的哪个二次变换。

[0162] NSST可能不适用于以变换跳过模式编码的块。当为CU发信号通知NSST索引且NSST索引不等于零时,NSST不用于在CU中以变换跳过模式编码的分量的块。当具有包括所有分量的块的CU以变换跳过模式进行编码或者非变换跳过模式CB的非零系数的数量小于2时,不为CU发信号通知NSST索引。

[0163] 4.2缩减大小变换(RST)

[0164] 在一些实施例中,采用了NSST的变体,称为缩减大小变换(RST,reduced size transform),或低频不可分离二次变换(LFNST,low-frequency non-separable secondary transform)。RST使用变换归零方案。检查帧内预测模式是平面还是DC,以对NSST的变换索

引进行熵编码。

[0165] 在示例中,应用四个变换集,并且每个变换集包括三个RST变换核。RST变换核的大小可以是 $16 \times 48$  (或 $16 \times 64$ ) (适用于高度和宽度都大于或等于8的变换系数块) 或 $16 \times 16$  (适用于高度或宽度等于4的变换系数块)。为了便于表示, $16 \times 48$  (或 $16 \times 64$ ) 变换表示为RST $8 \times 8$ , $16 \times 16$ 变换表示为RST $4 \times 4$ 。

[0166] 对于RST $8 \times 8$ ,使用 $16 \times 64$ 变换核和 $16 \times 48$ 变换核的两种可选变换编码过程(2000)和(2100)分别在图20和图21中示出。VVC草案5中采用了使用 $16 \times 48$ 变换核的变换编码过程。

[0167] 在图20示例的过程(2000)中,在编码器侧,可以首先对残差块执行正向主变换(2010),然后对从正向主变换(2010)生成的系数执行正向二次变换(2012)。在正向二次变换(2012)中,系数块(2013)左上角处的 $4 \times 4$ 子块A/B/C/D的64个系数表示为64长度矢量,并根据等式(等式3)与大小为 $16 \times 64$ 的变换矩阵相乘,得到16长度矢量。16长度矢量中的元素被填充回系数块(2013)的左上 $4 \times 4$ 子块A。子块B/C/D中的系数可以取零值。然后,在步骤(2014)处,对正向二次变换(2012)之后得到的系数进行量化,并对其进行熵编码,以生成比特流(2016)中的已编码比特。

[0168] 可以在解码器侧接收已编码比特,并进行熵解码,然后进行去量化(2024),以生成系数块(2023)。可以对左上 $4 \times 4$ 子块E处的16个系数执行逆二次变换(2022),以获得64个系数,这些系数被填充回 $4 \times 4$ 子块E/F/G/H。此后,可以利用逆主变换(2020)来处理逆二次变换(2022)后的块(2023)中的系数,以获得恢复的残差块。

[0169] 图21示例的过程(2100)与过程(2000)类似,只是在正向二次变换(2012)期间处理的系数较少(48个)。具体地,子块A/B/C中的48个系数使用大小为 $16 \times 48$ 的较小变换矩阵进行处理。使用较小的变换矩阵可以减少用于存储变换矩阵的存储器大小以及相应的计算复杂度。

[0170] 图22示出了一个示例的CU级别语法表(2200),其中在CU级别语法的末尾发信号通知语法元素`lfnst_idx`,其指示LFNST内核的选择。

[0171] 4.3 RST计算的示例

[0172] 缩减变换(RT)的主要思想是将N维矢量映射到不同空间中的R维矢量,其中 $R/N$  ( $R < N$ )是缩减因子。

[0173] RST矩阵是一个 $R \times N$ 矩阵,如下:

$$[0174] \quad T_{R \times N} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \dots & t_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \dots & t_{RN} \end{bmatrix} \quad (\text{等式 4})$$

[0175] 其中,变换的R行是N维空间的R个基(base)。RT的逆变换矩阵是其正向变换的转置。

[0176] 图23示出了缩减变换的过程(2301)和缩减逆变换的过程(2302)。T表示 $R \times N$ 维的RST变换矩阵, $T^T$ 表示 $N \times R$ 维的T的转置矩阵。

[0177] 在RST $8 \times 8$ 中,实现了 $4(1/4)$ 大小的缩减因子。例如,使用 $16 \times 64$ 直接矩阵而不是 $64 \times 64$  (其为传统的 $8 \times 8$ 不可分离变换矩阵大小)。在解码器侧使用 $64 \times 16$ 逆RST矩阵,以在 $8 \times 8$ 左上区域中生成核(主)变换系数。正向RST $8 \times 8$ 使用 $16 \times 64$  (或对于 $8 \times 8$ 块为 $8 \times 64$ ) 矩

阵,使得正向RST $8 \times 8$ 仅在给定 $8 \times 8$ 区域内的左上 $4 \times 4$ 区域中产生非零系数。换句话说,如果应用RST,则除左上 $4 \times 4$ 区域之外的 $8 \times 8$ 区域将只有零系数。对于RST $4 \times 4$ ,可以应用 $16 \times 16$ (或对于 $4 \times 4$ 块为 $8 \times 16$ )直接矩阵乘法。

[0178] 另外,对于RST $8 \times 8$ ,为了进一步减小变换矩阵大小,不使用残差块(2410)的整个左上 $8 \times 8$ 系数(图24A中的阴影子块)作为计算二次变换的输入,而是使用残差块(2410)的左上三个 $4 \times 4$ 子块系数(图24B中的阴影子块)作为计算二次变换的输入。

[0179] 在示例中,当满足以下两个条件时,有条件地应用逆RST:(i)相应块大小大于或等于给定阈值( $W \geq 4$ 且 $H \geq 4$ );以及(ii)变换跳过模式标志等于零。例如,如果变换系数块的宽度(W)和高度(H)都大于4,则RST $8 \times 8$ 应用于变换系数块的左上 $8 \times 8$ 区域。否则,RST $4 \times 4$ 应用于变换系数块的左上 $\min(8, W) \times \min(8, H)$ 区域。

[0180] 在示例中,当RST索引等于0时,不应用RST。否则,应用RST,并选择具有RST索引的内核。例如,RST应用于帧内切片和帧间切片中的帧内CU,以及亮度和色度。如果启用双树,则分别发信号通知亮度和色度的RST索引。对于帧间切片(禁用双树),发信号通知单个RST索引并将其用于亮度和色度。当选择ISP模式时,禁用RST,并且不发信号通知RST索引。

[0181] 4.4 RST变换矩阵选择的示例

[0182] 在示例中,可以从四个变换集中选择RST矩阵,并且每个变换集由两个变换组成。应用哪个变换集可以根据应用的帧内预测模式确定如下。当指示三个跨分量线性模型(CCLM)模式中的一个时,可以选择变换集0。否则,可以根据图25中所示的表(2500)执行变换集选择。访问表(2500)的索引(由IntraPredMode表示)可以在 $[-14, 83]$ 的范围内,例如,其是用于宽角度(wide angle)帧内预测的变换模式索引。

[0183] 5. 基于矩阵的帧内预测(MIP)模式

[0184] 在一些实施例中,采用基于矩阵的帧内预测(MIP)模式。图26示出了MIP模式的示例过程(2600)。为了预测宽度为W和高度为H的矩形块(2610)的样本,MIP取块(2610)左侧的一列H个重建的相邻边界样本、以及块(2610)上方的一行W个重建的相邻边界样本作为输入。如果重建样本不可用,则可以采用与传统帧内预测类似的方式生成重建样本。

[0185] 预测信号可以基于以下(2601)到(2603)的三个步骤来生成。在步骤(2601),在边界样本中, $W=H=4$ 的情况下的四个样本和所有其它情况下的八个样本通过平均提取。

[0186] 在步骤(2602),以平均样本 $\text{bdry}_{\text{red}}$ 作为输入,执行矩阵矢量乘法 $A_k \cdot \text{bdry}_{\text{red}}$ ,然后加上偏移 $b_k$ 。结果是原始块中的子采样样本集(2621)上的预测信号减少。矩阵 $A_k$ 和偏移 $b_k$ 可以基于MIP模式索引k来选择。

[0187] 在步骤(2603),其余位置(2622)处的预测信号通过线性插值根据子采样集(2621)上的预测信号生成,该线性插值是每个方向上的单步线性插值(single step linear interpolation)。

[0188] 生成预测信号所需的矩阵 $A_k$ 和偏移矢量 $b_k$ 可以取自矩阵的三个集合 $S_0$ 、 $S_1$ 、 $S_2$ 。集合 $S_0$ 由18个矩阵 $A_0^i, i \in \{0, \dots, 17\}$ 组成,每个矩阵具有16行、4列和18个偏移矢量 $b_0^i, i \in \{0, \dots, 17\}$ ,每个偏移矢量的大小为16。该集合的矩阵和偏移矢量用于大小为 $4 \times 4$ 的块。集合 $S_1$ 由10个矩阵 $A_1^i, i \in \{0, \dots, 9\}$ 组成,每个矩阵具有16行、8列和10个偏移矢量 $b_1^i, i \in \{0, \dots, 9\}$ ,每个偏移矢量的大小为16。该集合的矩阵和偏移矢量用于大小为 $4 \times 8 \times 8$

4和 $8 \times 8$ 的块。最后,集合 $S_2$ 由6个矩阵 $A_2^i, i \in \{0, \dots, 5\}$ 组成,每个矩阵具有64行、8列和6个大小为64的偏移矢量 $b_2^i, i \in \{0, \dots, 5\}$ 。该集合的矩阵和偏移矢量或这些矩阵和偏移矢量的一部分用于所有其它块形状。

[0189] 如图所示,给定一个 $8 \times 8$ 块(2610),MIP沿边界的每个轴取四个平均值。得到的八个输入样本进入矩阵矢量乘法。矩阵取自集合 $S_1$ 。这在预测块的奇数位置上产生16个样本(2621)。因此,每个样本总共执行 $(8 \cdot 16) / (8 \cdot 8) = 2$ 次乘法。在添加偏移之后,这些样本通过使用缩减的上边界样本进行垂直插值。随后通过使用原始的左边界样本进行水平插值。在这种情况下,插值过程不需要任何乘法。

[0190] 5.1 MIP模式的信令

[0191] 在一些实施例中,对于帧内模式中的每个编码单元(CU),可以在比特流中发送一个标志,其指示是否对相应的预测单元(PU)应用MIP模式。如果应用MIP模式,则使用包括3个MPM的MPM列表发信号通知MIP模式的索引predmode。

[0192] MPM的推导可以使用上方和左侧PU的帧内模式来执行如下。有三个固定的映射表 $\text{map\_angular\_to\_mip}_{idx}, idx \in \{0, 1, 2\}$ ,并且每个表将传统帧内预测模式 $\text{predmode}_{\text{Angular}}$ 与特定MIP模式相关联,如下式所述。

[0193]  $\text{predmode}_{\text{MIP}} = \text{map\_angular\_to\_mip}[\text{predmode}_{\text{Angular}}]$ . (等式5)

[0194] 其中, $\text{map\_angular\_to\_mip}$ 是固定的查找表。映射表的索引基于PU的宽度W和高度H来确定,并且总共三个索引可用,如下所述。

[0195]  $\text{idx}(\text{PU}) = \text{idx}(W, H) \in \{0, 1, 2\}$  (等式6)

[0196] 其指示从三个集合中的哪个集合获取上面的MIP参数。

[0197] 在示例中,为了生成使用MIP模式编码的当前块的MPM列表,首先导出上MIP模式(即 $\text{mode}_{\text{MIP}}^{\text{above}}$ )和左MIP模式(即 $\text{mode}_{\text{MIP}}^{\text{left}}$ )。

[0198]  $\text{mode}_{\text{MIP}}^{\text{above}}$ 的值可以导出如下:

[0199] 如果上PUPU<sub>above</sub>可用,并且其属于当前PU所在的同一CTU,并且PU<sub>above</sub>通过MIP使用MIP模式 $\text{predmode}_{\text{MIP}}^{\text{above}}$ 进行编码,并且 $\text{idx}(\text{PU}) = \text{idx}(\text{PU}_{\text{above}})$ (等式7),则

[0200]  $\text{mode}_{\text{MIP}}^{\text{above}} = \text{predmode}_{\text{MIP}}^{\text{above}}$ . (等式8)

[0201] 如果上PUPU<sub>above</sub>可用,并且其属于当前PU所在的同一CTU,并且PU<sub>above</sub>使用传统帧内预测模式 $\text{predmode}_{\text{Angular}}^{\text{above}}$ 进行编码,则

[0202]  $\text{mode}_{\text{MIP}}^{\text{above}} = \text{map\_angular\_to\_mip}[\text{predmode}_{\text{Angular}}^{\text{above}}]$ . (等式9)

[0203] 否则,

[0204]  $\text{mode}_{\text{MIP}}^{\text{above}} = -1$  (等式10)

[0205] 这意味着该模式不可用。

[0206]  $\text{mode}_{\text{MIP}}^{\text{left}}$ 的值使用与导出 $\text{mode}_{\text{MIP}}^{\text{above}}$ 相同的方式导出,但不检查左PU是否属于当前PU所在的同一CTU。

[0207] 最后,给定导出的 $\text{mode}_{\text{MIP}}^{\text{above}}$ 和 $\text{mode}_{\text{MIP}}^{\text{left}}$ 以及三个预定义的固定默认MPM列表 $\text{list}_{idx}, idx \in \{0, 1, 2\}$ ,其中每个MPM列表包含三种不同的MIP模式,构建MPM列表。

基于给定的默认列表 $\text{list}_{\text{idx(PU)}}$ 和 $\text{mode}_{\text{MIP}}^{\text{above}}$ 和 $\text{mode}_{\text{MIP}}^{\text{left}}$ 、通过用默认值替换-1以及移除重复的MIP模式来构建。

[0208] 作为示例,图27示出了CU级别语法表,其中发信号通知MIP模式的标志在帧(2701)中示出。

[0209] 5.2传统帧内预测模式的MPM列表推导

[0210] 在一些实施例中,MIP模式与传统帧内预测模式的基于MPM的编码协调如下。用于传统帧内预测模式的亮度和色度MPM列表的推导过程使用单独的固定表 $\text{map\_mip\_to\_angular}_{\text{idx}}$ , $\text{idx} \in \{0, 1, 2\}$ ,该固定表将MIP模式 $\text{predmode}_{\text{MIP}}$ 映射到传统帧内预测模式中的一个

[0211]  $\text{predmode}_{\text{Angular}} = \text{map\_mip\_to\_angular}[\text{predmode}_{\text{MIP}}]$ . (等式11)

[0212] 其中, $\text{map\_mip\_to\_angular}$ 是固定的查找表。对于亮度MPM列表的推导,每当通过MIP模式 $\text{predmode}_{\text{MIP}}$ 对相邻亮度块进行编码时,该块就可以被视为使用传统帧内预测模式 $\text{predmode}_{\text{Angular}}$ 。对于色度MPM列表的推导,每当当前亮度块使用MIP模式时,使用相同的映射将MIP模式转换为传统帧内预测模式。

[0213] III. 基于高级语法元素或块级别语法元素启用隐式变换选择

[0214] 在一些实施例中,可以采用两种变换编码方案:隐式变换(或称为隐式变换选择)和显式变换(或称为显式变换选择)。

[0215] 在隐式变换中,可以在没有变换索引信令的情况下选择一组非DCT-2变换(例如,DST-1、DCT-5、DST-7、DCT-8、DST-4、DCT-4)。例如,可以使用编码器和相应解码器都可用的已编码信息来选择一组非DCT-2变换。已编码信息可以包括但不限于,帧内预测模式(例如,平面模式、DC模式、角度模式)、块大小、块宽度、块高度、块宽高比、块面积大小、帧内编码模式(例如,是否使用多参考行(MRL)、ISP、MIP)、所选择的空合并候选的位置(例如,上合并候选、左合并候选)、帧间预测模式(例如,帧间位置相关预测组合(inter-PDPC)模式、组合的帧间帧内预测(CIIP)模式)。

[0216] 相反,在显式变换中,可以从一组变换类型候选(例如DCT-2、DST-1、DCT-5、DST-7、DCT-8、DST-4、DCT-4)中选择一个变换,该变换具有发信号通知的索引以指示选择了哪种变换类型。

[0217] 1. 当显式MTS被禁用时启用隐式变换

[0218] 在一些实施例中,如第II.3.2节中所述,对于未由ISP编码的帧内预测残差块,当对由高级语法(HLS)元素指示的帧内和帧间预测残差块禁用显式MTS时,可以启用隐式变换选择。例如,当SPS语法元素 $\text{sps\_explicit\_mts\_intra\_enabled\_flag}$ 和 $\text{sps\_explicit\_mts\_inter\_enabled\_flag}$ 均为0时,解码器可以确定对与SPS语法元素相关联的图片或块启用隐式变换选择。

[0219] 例如,图28A-28B组合示出了文本(2800),该文本(2800)基于从比特流接收的相关语法元素指定对当前块执行显式或隐式变换选择的变换编码过程。该文本(2800)可用作视频编码标准的一部分。图28A中示出了两个部分(2801)和(2802),图28B中示出了一个部分(2803)和两个表(2804)和(2805)。

[0220] 在部分(2801)中,描述了变换编码过程的输入和输出。具体地,当前块的位置、大小、颜色分量、逆量化变换系数是输入,使用所选择的变换进行逆变换处理后的残差样本是

输出。

[0221] 在部分(2802)中,描述了由implicitMtsEnabled表示的变量的推导。该变量指示是否启用隐式选择。如上所述,如果SPS语法元素sps\_mts\_enabled\_flag等于1(这表示对与该SPS语法元素相关联的图片或块启用MTS),则当以下三个条件中的一个为真时,implicitMtsEnabled等于1:(i)ISP用于对当前块进行编码;(ii)启用SBT,并且当前块的两个边都小于或等于32,这表示使用SBT;或(iii)sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0(这表示对帧内和帧间编码块都禁用显式MTS),并且当前块是帧内编码的。否则,implicitMtsEnabled被设置为0,这表示对当前块禁用隐式变换选择。

[0222] 如部分(2802)中所述,根据条件(i),对于使用ISP模式编码的帧内预测块,可以启用隐式变换来编码该帧内预测块。根据条件(iii),对于未使用ISP模式编码的帧内预测块,当对帧内和帧间预测块禁用显式MTS时,可以启用隐式变换来编码未使用ISP模式编码的帧内预测块。

[0223] 在部分(2803)中,根据变量implicitMtsEnabled、该过程的输入和相关语法元素确定垂直和水平变换。例如,当sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0时,可以根据表达式(3-1)和(3-2)分别为当前块(该当前块是亮度分量的并且是帧内预测的)提供水平变换核(由trTypeHor表示)和垂直变换核(由trTypeVer表示)。

[0224] 在表(2804)和(2805)中,表示水平或垂直变换核类型的数字1和2分别指示DST-7变换和DCT-8变换,而数字0指示DCT-2变换。在一些示例中,可以使用DST-4变换来代替DST-7变换。

[0225] 在图28A-28B的示例中,对于由HLS元素sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag控制的块,用于帧间编码块的显式MTS和用于帧内编码块(非ISP编码)的隐式变换不能共存。然而,帧内预测残差块上的隐式变换选择不必依赖于HLS元素(例如,sps\_explicit\_mts\_inter\_enabled\_flag),其中该HLS元素控制对帧间预测残差启用MTS。因此,在一些实施例中,对于不是由ISP模式预测的帧内编码块,是否可以应用隐式变换取决于MTS是否可以应用于帧内预测残差(例如,sps\_explicit\_mts\_intra\_enabled\_flag的值),但不取决于MTS是否可以应用于帧间预测残差(sps\_explicit\_mts\_inter\_enabled\_flag的值)。通过这种控制机制,允许同时启用帧间MTS(应用于帧间预测残差块的显式MTS)和隐式变换(用于帧内预测残差)。

[0226] 例如,可以在解码器处接收第一HLS元素和第二HLS元素。第一HLS元素和第二HLS元素控制同一组图片或区域(例如,视频序列、图片、切片、图块等),该同一组图片或区域可以包括帧内编码残差块和帧间编码残差块。第一HLS元素指示对各个帧内编码残差块启用还是禁用显式MTS,而第二HLS元素指示对各个帧间编码残差块启用还是禁用显式MTS。关于帧间MTS的术语,应用于帧内编码残差块的显式MTS可以称为帧内MTS。

[0227] 当第一HLS元素指示对帧内编码残差块禁用显式MTS时,解码器可以相应地确定对帧内编码块启用隐式变换(或隐式MTS),而不考虑第二HLS元素的值。例如,第二HLS元素可以是0或1,这不影响对帧内编码残差块启用隐式变换。

[0228] 作为示例,图29示出了对文本(2800)的修改(2900),这些修改对应于隐式变换启

用方案,其中用于帧内残差块的隐式变换和用于帧间残差块的显式变换可以共存。在修改(2900)中,移除的文本用删除线标记,而添加的文本用下划线标记。如图所示,sps\_explicit\_mts\_inter\_enabled\_flag为0的条件已被删除。

[0229] 2.当启用其它编码工具时禁用隐式变换

[0230] 在一些实施例中,隐式变换是否可以应用于帧内编码块(其不是由ISP模式预测的)取决于是否如HLS元素所指示的那样启用了特定的编码工具。

[0231] 2.1当启用NSST时

[0232] 在实施例中,当HLS元素指示启用NSST时,可以对帧内编码块(其不是由ISP模式预测的)禁用隐式变换。例如,如第II.4.1节所述,在等式(等式3)中使用变换矩阵T进行二次变换处理。变换矩阵T可以包括常数元素,并且在假设目标系数块(例如由显式变换选择产生的系数块)的特定统计数据的情况下进行设计。因此,由隐式变换选择产生的系数块在各个系数的统计数据方面可能与NSST不匹配。例如,对隐式变换的结果应用NSST可能不会提高编解码性能。由于上述原因,在启用NSST时禁用隐式变换可能是可取的。

[0233] 在本公开中,术语NSST可用于指代一系列不可分离二次变换编码方案,例如NSST、RST、LFNST等的初始设计。

[0234] 例如,解码器可以接收两个HLS元素:一个HLS元素(例如,sps\_explicit\_mts\_intra\_enabled\_flag)指示禁用显式MTS,而另一个HLS元素(例如,sps\_lfnst\_enabled\_flag)指示启用NSST。因此,解码器可以确定不对这两个HLS元素控制的帧内编码块启用隐式变换。相反,如果一个HLS元素指示禁用显式MTS,另一个HLS元素指示也禁用NSST,则解码器可以确定对各个帧内编码块启用隐式变换。

[0235] 作为示例,图30示出了对文本(2800)的修改(3000),该修改对应于当NSST被禁用时启用隐式变换的情形。如图所示,将附加限制“sps\_lfnst\_enabled\_flag等于0”添加到条件(iii)以导出变量implicitNtsEnabled。

[0236] 2.2当启用MIP时

[0237] 类似于启用NSST的场景,在一些实施例中,当HLS元素指示启用MIP时,对帧内编码块(其不是由ISP模式预测的)禁用隐式变换。例如,对块应用MIP帧内编码模式可能导致残差块具有与使用常规帧内模式编码的残差块不同的统计数据。因此,使用MIP编码的残差块可能与隐式变换不匹配,隐式变换可能假定由常规帧内模式产生的残差统计数据。因此,当启用MIP时禁用隐式变换可能是可取的。

[0238] 例如,解码器可以接收两个HLS元素:一个HLS元素(例如,sps\_explicit\_mts\_intra\_enabled\_flag)指示禁用显式MTS,而另一个HLS元素(例如,sps\_mip\_enabled\_flag)指示启用MIP。因此,解码器可以确定不对这两个HLS元素控制的帧内编码块启用隐式变换。相反,如果如这两个HLS元素所指示的,显式MTS和MIP都被禁用,则解码器可以确定对各个帧内编码块启用隐式变换。

[0239] 作为示例,图31示出了对文本(2800)的修改(3100),该修改对应于当MIP被禁用时启用隐式变换的情形。如图所示,将附加限制“sps\_mip\_enabled\_flag等于0”添加到条件(iii)以导出变量implicitNtsEnabled。

[0240] 3、基于块级别语法元素的指示启用隐式变换

[0241] 在一些实施例中,隐式变换是否可以应用于帧内编码块(其不是由ISP模式预测

的)取决于MTS、NSST或MIP是否如块级别语法元素所指示的那样应用于帧内编码块。

[0242] 3.1当不应用MTS和NSST时

[0243] 在实施例中,隐式变换是否可以应用于帧内编码块取决于MTS和NSST是否如块级别语法元素所指示的那样不应用于帧内编码块。

[0244] 例如,解码器可以接收与当前块相关联的第一和第二CU级别(或块级别)语法元素,其中,当前块是帧内编码的且不是由ISP模式预测的。当前块可以位于图片内的坐标[x0][y0]位置处。

[0245] 第一CU级别语法元素(例如,tu\_mts\_idx[x0][y0])可以指示对当前块使用DCT-2变换,而不是MTS中使用的变换(例如,DST-7、DCT-8、DCT-4等)。作为示例,在表(2804)中,当tu\_mts\_idx[x0][y0]的值为0时,变量trTypeHor和trTypeVer的值均为0,这表示水平变换和垂直变换都是DCT-2变换,并且不应用DST-7或DCT-8变换。第二CU级别语法元素(例如,1fnst\_idx[x0][y0])可以指示对当前块不应用NSST。基于上述第一CU级别语法元素和第二CU级别语法元素,解码器可以确定对当前块启用隐式MTS。

[0246] 相反,如果第一CU级别语法元素和第二CU级别语法元素指示应用MTS(tu\_mts\_idx[x0][y0]的值为1、2、3或4),或者应用NSST(1fnst\_idx[x0][y0]具有非零值),则解码器可以确定禁用隐式MTS。

[0247] 作为示例,图32示出了对文本(2800)的修改(3200),该修改对应于MTS和NSST都不应用于当前块的情形。如图所示,文本(2800)中的原始条件(iii)被替换为两个CU级别语法元素“tu\_mts\_idx[x0][y0]和1fnst\_idx[x0][y0]均等于0”的条件。

[0248] 3.2当不应用MTS、NSST和MIP时

[0249] 在实施例中,隐式变换是否可以应用于帧内编码块取决于MTS、NSST和MIP是否如块级别语法元素所指示的那样不应用于帧内编码块。与第III.3.1节相比,另外考虑了一种编码工具MIP。

[0250] 例如,解码器可以接收与当前块相关联的第一、第二和第三CU级别(或块级别)语法元素,其中当前块是帧内编码的且不是由ISP模式预测的。当前块可以位于图片内的坐标[x0][y0]位置处。

[0251] 第一CU级别语法元素(例如,tu\_mts\_idx[x0][y0])可以指示对当前块使用DCT-2变换,而不是MTS中使用的变换。第二CU级别语法元素(例如,1fnst\_idx[x0][y0])可以指示NSST不应用于当前块。第三CU级别语法元素(例如,tu\_mip\_flag)可以指示MIP不应用于当前块。基于上述三个CU级别语法元素,解码器可以确定对当前块启用隐式MTS。

[0252] 相反,如果上述三个CU级别语法元素指示应用MTS(tu\_mts\_idx[x0][y0]的值为1、2、3或4)、应用NSST(1fnst\_idx[x0][y0]具有非零值)、或应用MIP(tu\_mip\_flag的值为1),则解码器可以确定禁用隐式MTS。

[0253] 作为示例,图33示出了对文本(2800)的修改(3300),该修改对应于MTS、NSST或MIP都不应用于当前块的情形。如图所示,文本(2800)中的原始条件(iii)被替换为三个CU级别语法元素“tu\_mts\_idx[x0][y0]、intra\_mip\_flag[x0][y0]和1fnst\_idx[x0][y0]都等于0”的条件。

[0254] 4.变换编码过程的示例

[0255] 图34-36示出了根据本公开的一些实施例的变换编码过程(3400)、(3500)和

(3600)的流程图。过程(3400)、(3500)和(3600)可用于在解码器处的逆变换处理,以便为正在重建的块生成残差块。在各种实施例中,过程(3400)、(3500)和(3600)可以由处理电路执行,例如终端装置(210)、(220)、(230)和(240)中的处理电路、执行视频解码器(310)的功能的处理电路、执行视频解码器(410)的功能的处理电路等。在一些实施例中,过程(3400)、(3500)和(3600)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(3400)、(3500)和(3600)。

[0256] 4.1过程(3400):当隐式MTS被禁用时启用隐式变换

[0257] 过程(3400)从(S3401)开始,并且进行到(S3410)。

[0258] 在(S3410),在解码器处接收第一HLS元素。第一HLS元素可以指示是否对帧内编码块禁用显式MTS。例如,帧内编码块正在重建,因此可以称为当前块。第一HLS元素可以与包括当前块的一组编码块相关联。

[0259] 在(S3420),在解码器处接收第二HLS元素。第二HLS元素可以指示是否对帧间编码块启用显式MTS。例如,第二HLS元素可以与与第一HLS元素相关联的同一组编码块相关联。当前块和帧间编码块都包括在该组编码块中。

[0260] 在(S3430),当第一HLS元素指示对帧内编码块禁用显式MTS,并且第二HLS元素指示对帧间编码块启用显式MTS时,解码器可以确定对当前块启用隐式MTS。过程(3400)可以进行到(S3499),并且在(S3499)结束。

[0261] 4.2过程(3500):当启用其它编码工具时禁用隐式变换

[0262] 过程(3500)从(S3501)开始,并且进行到(S3510)。

[0263] 在(S3510),在解码器处接收第一HLS元素。第一HLS元素可以指示是否对帧内编码块启用显式MTS。帧内编码块可以是正在重建的块。第一HLS元素控制包括该帧内编码块的一组编码块。

[0264] 在(S3520),在解码器处接收第二HLS元素。第二个HLS元素可以指示是否对帧内编码块禁用NSST或MIP。例如,第二HLS元素可以控制与第一HLS元素控制的同一组编码块。

[0265] 在(S3530),解码器可以基于第一HLS元素和第二HLS元素确定是否启用隐式MTS来处理帧内编码块。例如,当第一HLS元素指示对帧内编码块禁用显式MTS,并且第二HLS元素指示对帧内编码块禁用NSST或MIP时,解码器可以确定对帧内编码块启用隐式MTS。或者,当第一HLS元素指示对帧内编码块禁用显式MTS,但第二HLS元素指示对帧内编码块启用NSST或MIP时,解码器可以确定对帧内编码块禁用隐式MTS。过程(3500)可以进行到(S3599),并且在(S3599)结束。

[0266] 4.3过程(3600):基于块级别的语法元素指示启用隐式变换

[0267] 过程(3600)从(S3601)开始,并且进行到(S3610)。

[0268] 在(S3610),在解码器处接收帧内编码块。帧内编码块可以与第一块级别语法元素和第二块级别语法元素相关联,第一块级别语法元素指示是否应用MTS,第二块级别语法元素指示是否应用NSST。例如,帧内编码块属于包括CU级别语法元素的CU,该CU级别语法元素包括第一块级别语法元素和第二块级别语法元素。

[0269] 在(S3620),解码器可以基于第一块级别语法元素和第二块级别语法元素确定是否对帧内编码块启用隐式MTS。例如,当第一块级别语法元素指示不应用MTS,并且第二块级别语法元素指示不应用NSST时,解码器可以确定对帧内编码块启用隐式MTS。或者,当第一

块级别语法元素指示不应用MTS,但第二块级别语法元素指示应用NSST时,解码器可以确定对帧内编码块禁用隐式MTS。过程(3600)可以进行到(S3699),并且在(S3699)结束。

#### [0270] IV. 计算机系统

[0271] 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37示出了计算机系统(3700),其适于实现所公开主题的某些实施例。

[0272] 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。

[0273] 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。

[0274] 图37所示的用于计算机系统(3700)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(3700)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要求。

[0275] 计算机系统(3700)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)。

[0276] 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(3701)、鼠标(3702)、触控板(3703)、触摸屏(3710)、数据手套(未示出)、操纵杆(3705)、麦克风(3706)、扫描仪(3707)、照相机(3708)。

[0277] 计算机系统(3700)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(3710)、数据手套(未示出)或操纵杆(3705)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(3709)、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(3710),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)。

[0278] 计算机系统(3700)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)(3720)或类似介质(3721)的光学介质、拇指驱动器(3722)、可移动硬盘驱动器或固体状态驱动器(3723),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。

[0279] 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。

[0280] 计算机系统(3700)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、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(3749)(例如,计算机系统(3700)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(3700)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(3700)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。

[0281] 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(3700)的核心(3740)。

[0282] 核心(3740)可包括一个或多个中央处理单元(CPU)(3741)、图形处理单元(GPU)(3742)、以现场可编程门阵列(FPGA)(3743)形式的专用可编程处理单元、用于特定任务的硬件加速器(3744)等。这些设备以及只读存储器(ROM)(3745)、随机存取存储器(3746)、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)(3747)等可通过系统总线(3748)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(3748),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(3748),或通过外围总线(3749)进行连接。外围总线的体系结构包括外部外围组件互联PCI、通用串行总线USB等。

[0283] CPU(3741)、GPU(3742)、FPGA(3743)和加速器(3744)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(3745)或RAM(3746)中。过渡数据也可以存储在RAM(3746)中,而永久数据可以存储在例如内部大容量存储器(3747)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(3741)、GPU(3742)、大容量存储器(3747)、ROM(3745)、RAM(3746)等紧密关联。

[0284] 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。

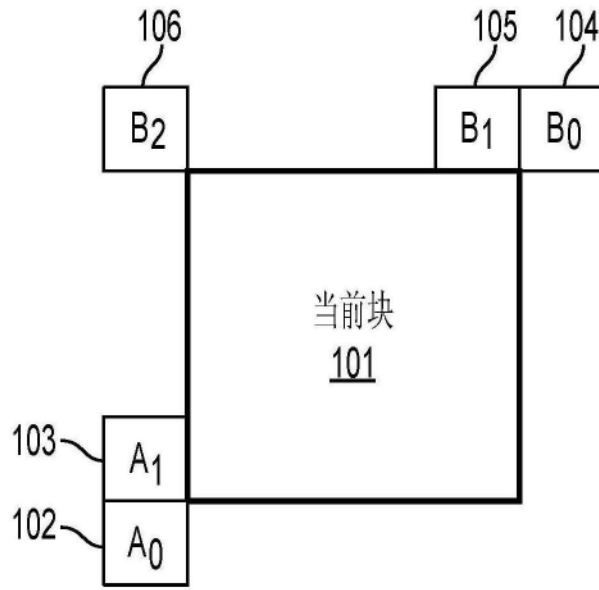
[0285] 作为实施例而非限制,具有体系结构(3700)的计算机系统,特别是核心(3740),可以作为处理器(包括CPU、GPU、FPGA、加速器等)提供执行包含在一个或多个有形的计算机可读介质中的软件的功能。这种计算机可读介质可以是与上述的用户可访问的大容量存储器相关联的介质,以及具有非易失性的核心(3740)的特定存储器,例如核心内部大容量存储器(3747)或ROM(3745)。实现本公开的各种实施例的软件可以存储在这种设备中并且由核心(3740)执行。根据特定需要,计算机可读介质可包括一个或一个以上存储设备或芯片。该软件可以使得核心(3740)特别是其中的处理器(包括CPU、GPU、FPGA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(3746)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(3744))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。

- [0286] 附录A:首字母缩略词
- [0287] AMT:自适应多变换(Adaptive Multiple Transform)
- [0288] ASIC:专用集成电路(Application-Specific Integrated Circuit)
- [0289] BMS:基准集合(benchmark set)
- [0290] CANBus:控制器局域网络总线(Controller Area Network Bus)
- [0291] CCLM:跨分量线性模式/模型(Cross-Component Linear Model)
- [0292] CD:光盘(Compact Disc)
- [0293] COT:复合正交变换(Compound Orthonormal Transform)
- [0294] CPUs:中央处理单元(Central Processing Units)
- [0295] CRT:阴极射线管(Cathode Ray Tube)
- [0296] CTBs:编码树块(Coding Tree Blocks)
- [0297] CTUs:编码树单元(Coding Tree Units)
- [0298] CU:编码单元(Coding Unit)
- [0299] DVD:数字化视频光盘(Digital Video Disc)
- [0300] EMT:增强多变换(Enhanced Multiple Transform)
- [0301] FPGA:现场可编程门阵列(Field Programmable Gate Areas)
- [0302] GOPs:图片群组(Groups of Pictures)
- [0303] GPUs:图形处理单元(Graphics Processing Units)
- [0304] GSM:全球移动通信系统(Global System for Mobile communications)
- [0305] HDR:高动态范围(high dynamic range)
- [0306] HEVC:高效视频编解码(High Efficiency Video Coding)
- [0307] HLS:高级语法(High-Level Syntax)
- [0308] HRD:假想参考解码器(Hypothetical Reference Decoder)
- [0309] IBC:帧内块复制(Intra Block Copy)
- [0310] IC:集成电路(Integrated Circuit)
- [0311] IDT:恒等变换(Identity transform)
- [0312] ISP:帧内子分区(Intra Sub-Partitioning)
- [0313] JEM:联合开发模型(joint exploration model)
- [0314] JVET:联合视频开发组(Joint Video Exploration Team)
- [0315] KLT:K-L变换(Karhunen-Loève Transform)
- [0316] LAN:局域网(Local Area Network)
- [0317] LCD:液晶显示器(Liquid-Crystal Display)
- [0318] LFNST:低频不可分离二次变换(Low-Frequency Non-Separable Secondary

Transform)

- [0319] LTE:长期演进(Long-Term Evolution)
- [0320] MIP:基于矩阵的帧内预测模式(Matrix-based Intra Prediction Mode)
- [0321] MRL(or MRLP):多参考行预测(Multiple reference line prediction)
- [0322] MTS:多变换选择(Multiple Transform Selection)
- [0323] MV:运动矢量(Motion Vector)
- [0324] NSST:不可分离二次变换(Non-Separable Secondary Transform)
- [0325] OLED:有机发光二极管(Organic Light-Emitting Diode)
- [0326] PBs:预测块(Prediction Blocks)
- [0327] PCI:外围组件互连(Peripheral Component Interconnect)
- [0328] PLD:可编程逻辑设备(Programmable Logic Device)
- [0329] PPS:图片参数集(Picture Parameter Set)
- [0330] PU:预测单元(Prediction Unit)
- [0331] RAM:随机存取存储器(Random Access Memory)
- [0332] ROM:只读存储器(Read-Only Memory)
- [0333] RST:缩减大小变换(Reduced-Size Transform)
- [0334] SBT:子块变换(Sub-block Transform)
- [0335] SDR:标准动态范围(standard dynamic range)
- [0336] SEI:辅助增强信息(Supplementary Enhancement Information)
- [0337] SNR:信噪比(Signal Noise Ratio)
- [0338] SPS:序列参数集(Sequence Parameter Set)
- [0339] SSD:固态驱动器(solid-state drive)
- [0340] SVT:空间变化变换(Spatially Varying Transform)
- [0341] TSM:变换跳过模式(Transform Skip Mode)
- [0342] TUs:变换单元(Transform Units)
- [0343] USB:通用串行总线(Universal Serial Bus)
- [0344] VPS:视频参数集(Video Parameter Set)
- [0345] VUI:视频可用性信息(Video Usability Information)
- [0346] VVC:通用视频编解码(versatile video coding)
- [0347] 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。



(相关技术)

图1

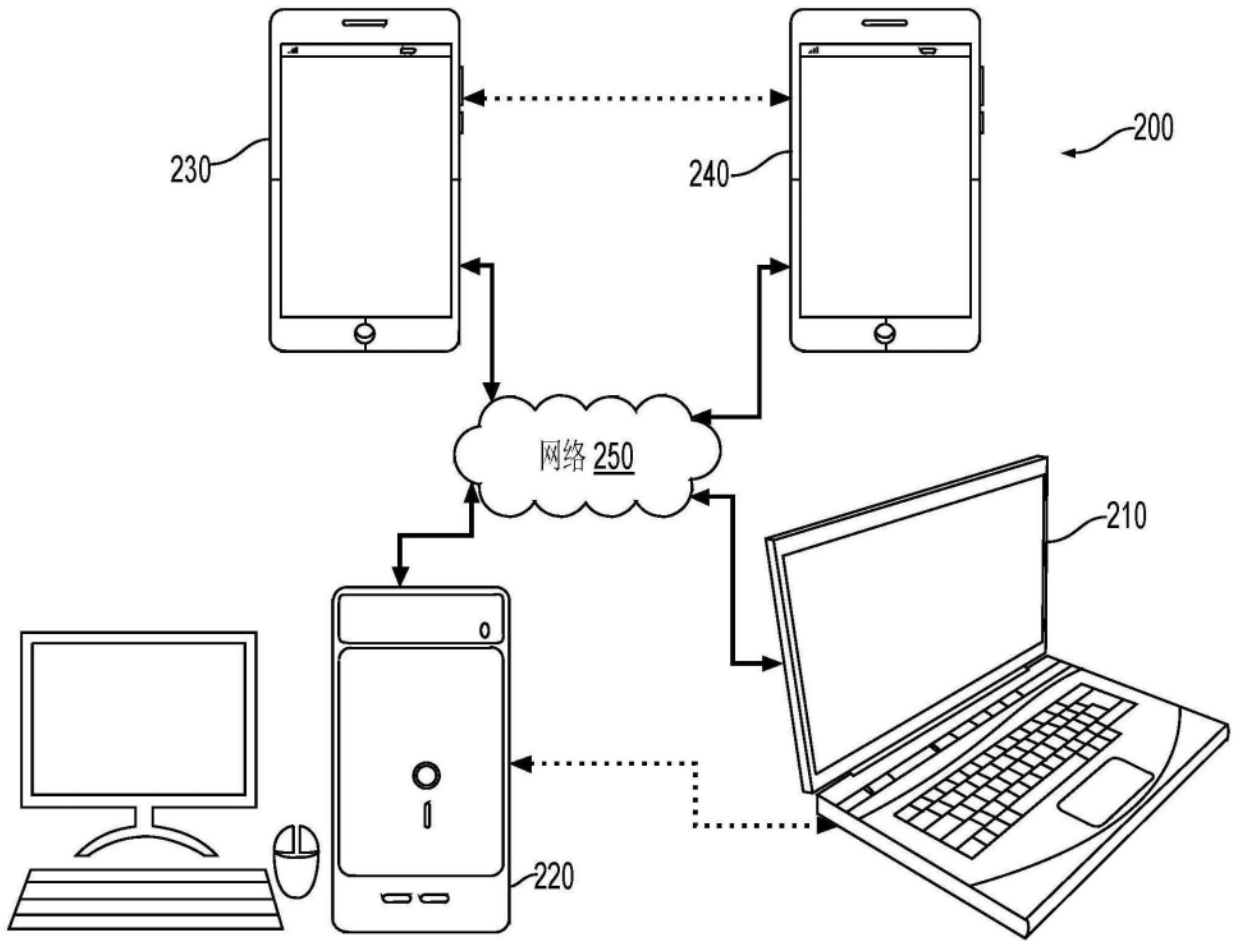


图2

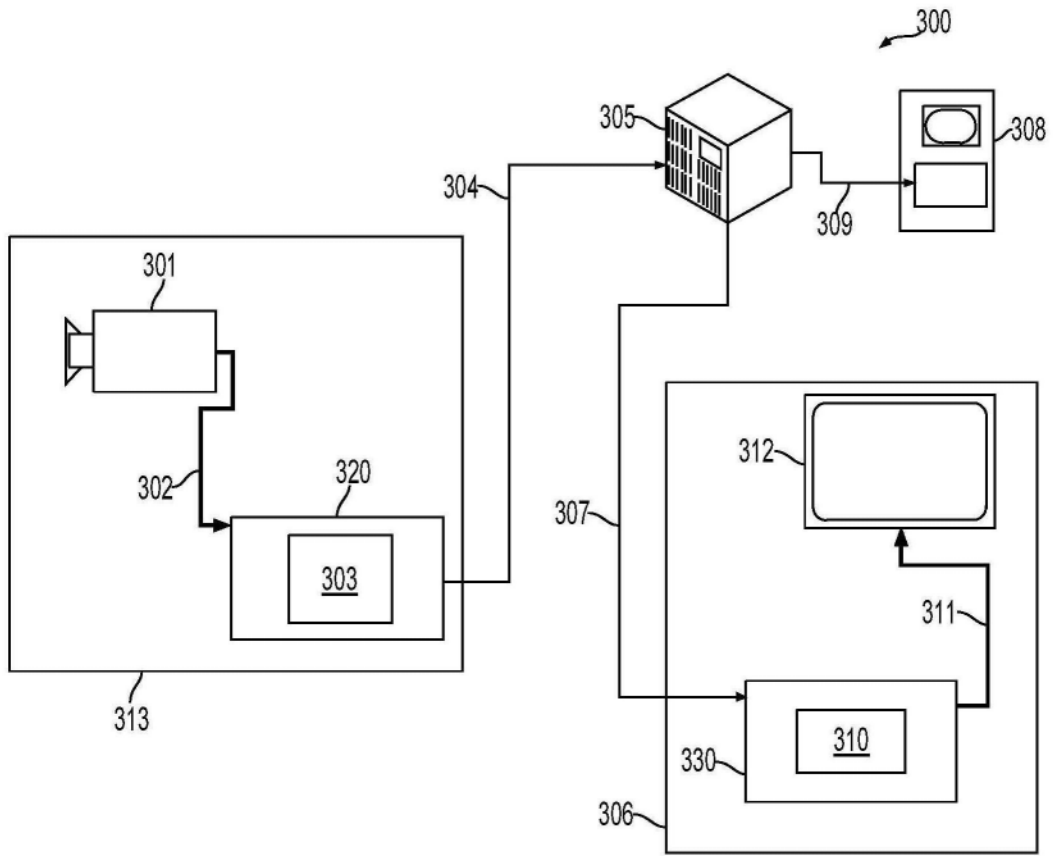


图3

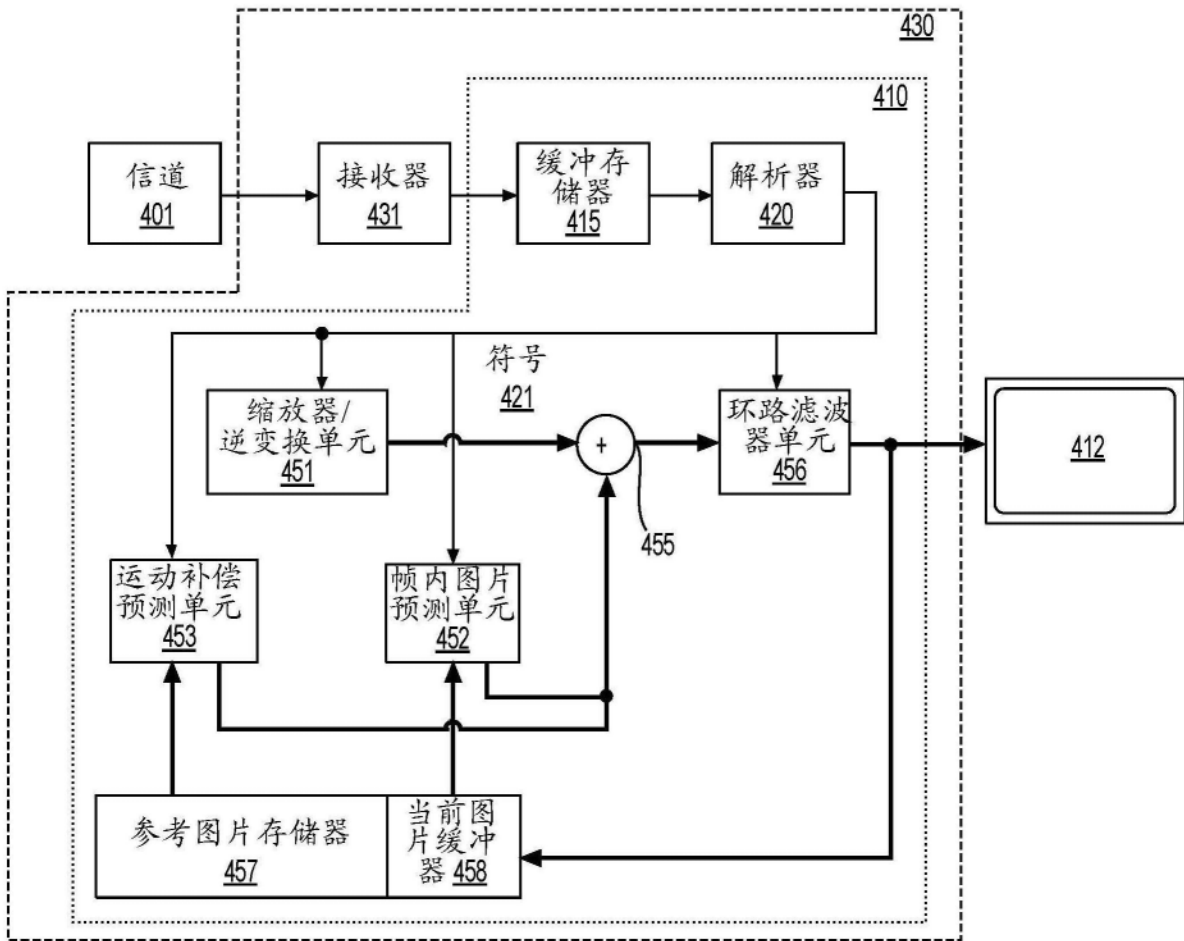


图4

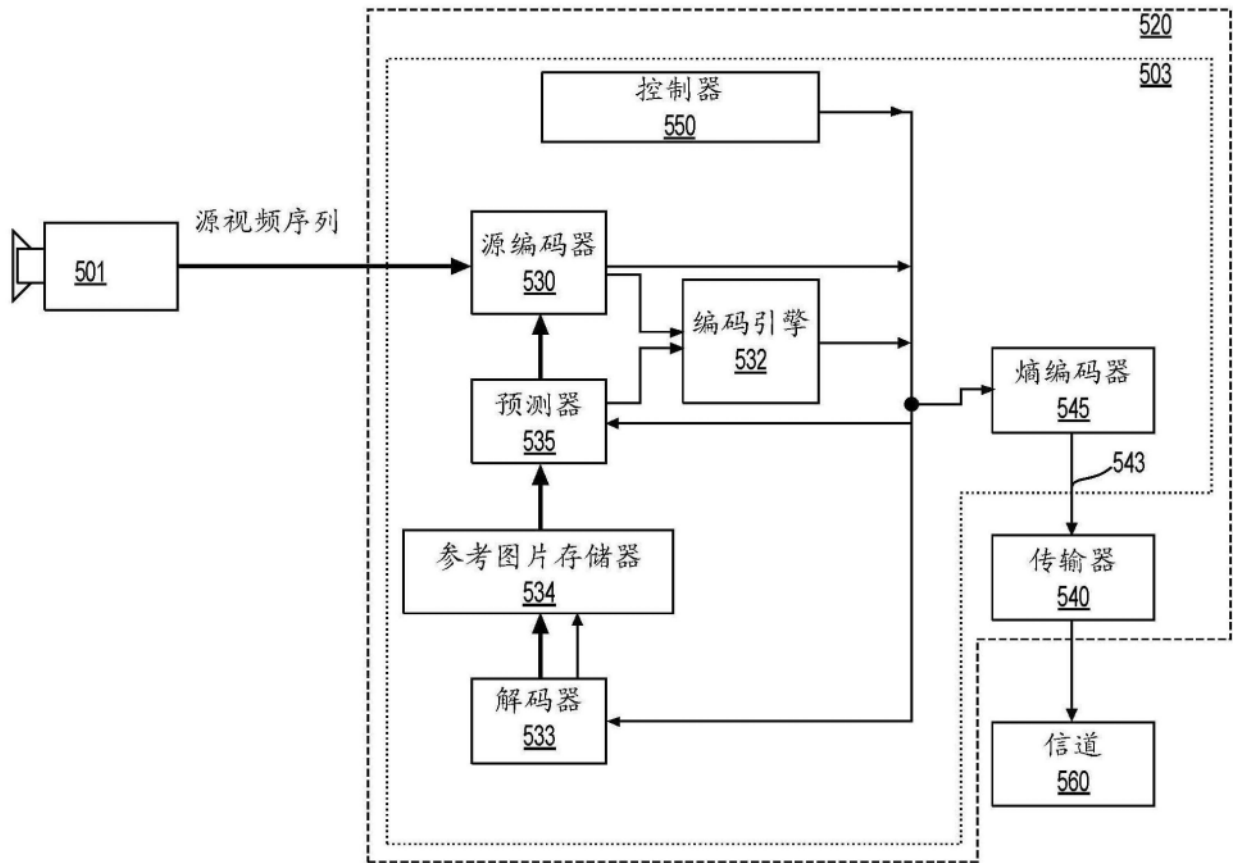


图5

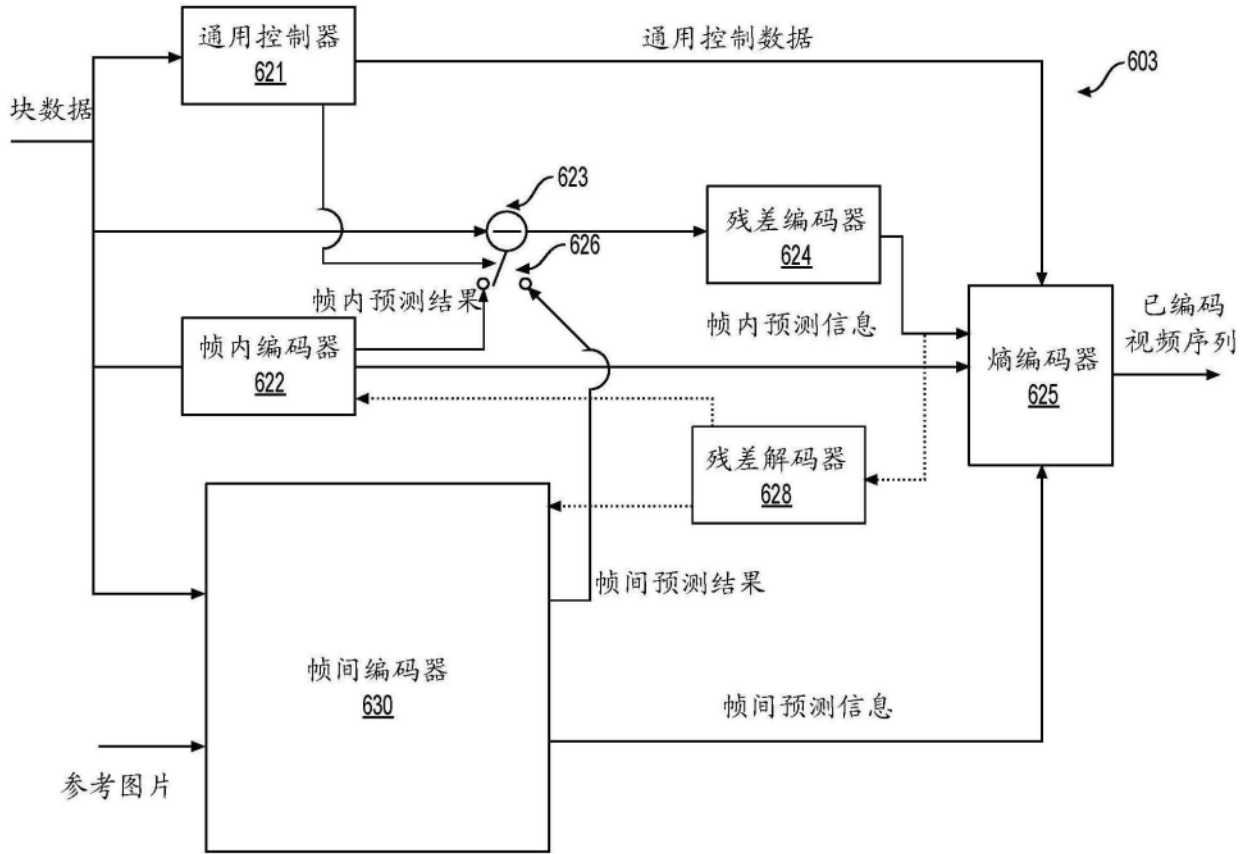


图6

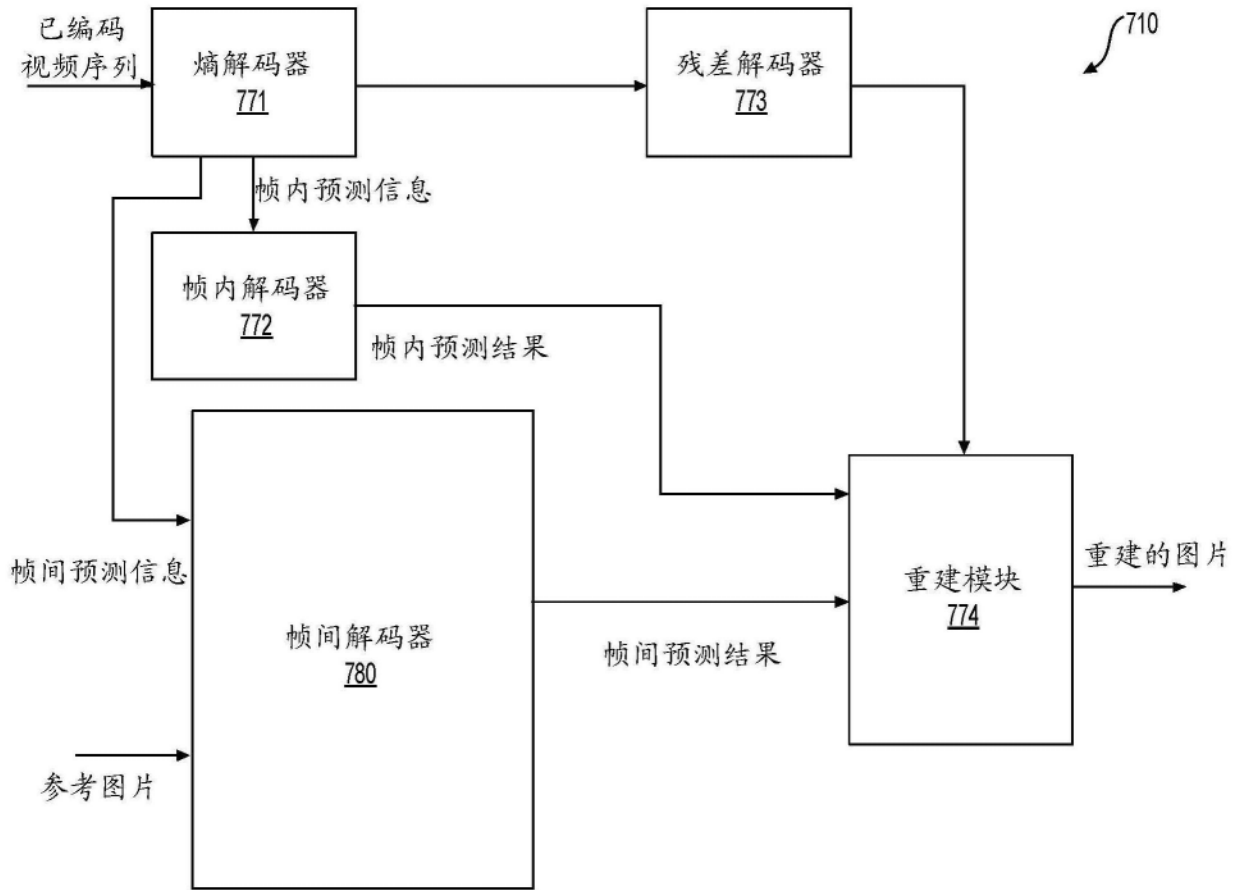


图7

4×4变换

{ 64, 64, 64, 64 }  
{ 83, 36, -36, -83 }  
{ 64, -64, -64, 64 }  
{ 36, -83, 83, -36 }

图8A

**8×8变换**

```

{64, 64, 64, 64, 64, 64, 64, 64}
{89, 75, 50, 18, -18, -50, -75, -89}
{83, 36, -36, -83, -83, -36, 36, 83}
{75, -18, -89, -50, 50, 89, 18, -75}
{64, -64, -64, 64, 64, -64, -64, 64}
{50, -89, 18, 75, -75, -18, 89, -50}
{36, -83, 83, -36, -36, 83, -83, 36}
{18, -50, 75, -89, 89, -75, 50, -18}

```

图8B

**16×16变换**

```

{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 87 80 70 57 43 25 9 -9 -25 -43 -57 -70 -80 -87 -90}
{89 75 50 18 -18 -50 -75 -89 -89 -75 -50 -18 18 50 75 89}
{87 57 9 -43 -80 -90 -70 -25 25 70 90 80 43 -9 -57 -87}
{83 36 -36 -83 -83 -36 36 83 83 36 -36 -83 -83 -36 36 83}
{80 9 -70 -87 -25 57 90 43 -43 -90 -57 25 87 70 -9 -80}
{75 -18 -89 -50 50 89 18 -75 -75 18 89 50 -50 -89 -18 75}
{70 -43 -87 9 90 25 -80 -57 57 80 -25 -90 -9 87 43 -70}
{64 -64 -64 64 64 -64 -64 64 64 -64 -64 64 64 -64 -64 64}
{57 -80 -25 90 -9 -87 43 70 -70 -43 87 9 -90 25 80 -57}
{50 -89 18 75 -75 -18 89 -50 -50 89 -18 -75 75 18 -89 50}
{43 -90 57 25 -87 70 9 -80 80 -9 -70 87 -25 -57 90 -43}
{36 -83 83 -36 -36 83 -83 36 36 -83 83 -36 -36 83 -83 36}
{25 -70 90 -80 43 9 -57 87 -87 57 -9 -43 80 -90 70 -25}
{18 -50 75 -89 89 -75 50 -18 -18 50 -75 89 -89 75 -50 18}
{ 9 -25 43 -57 70 -80 87 -90 90 -87 80 -70 57 -43 25 -9}

```

图8C



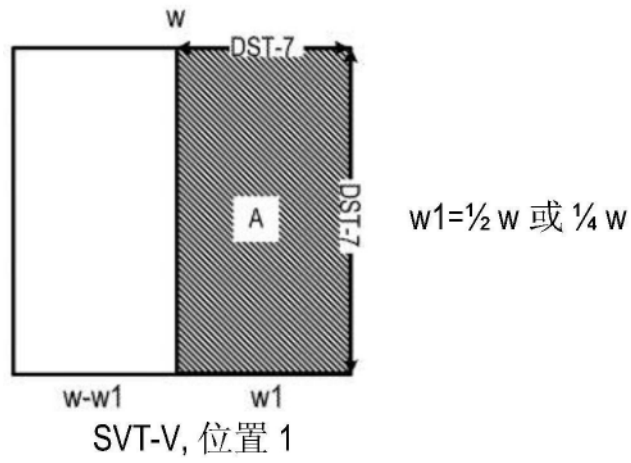


图9B

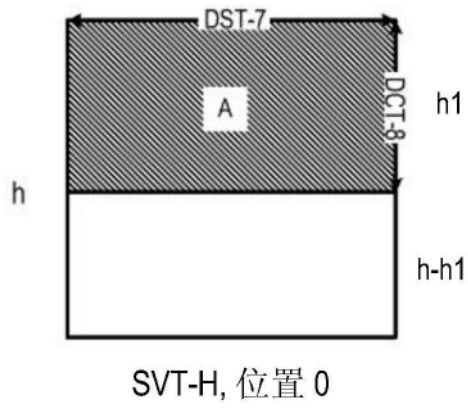


图9C

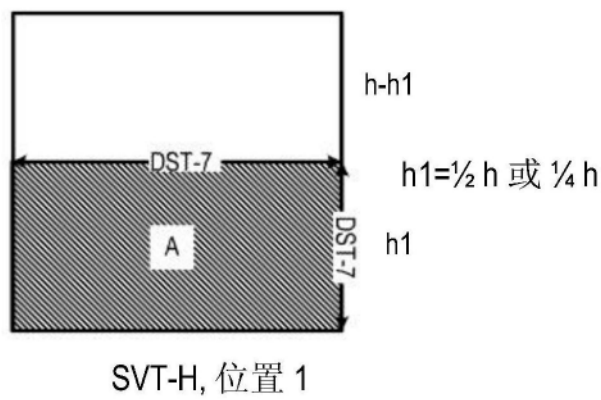


图9D

取决于块大小的子分区的数量

块大小	子分区的数量
4 × 4	不划分
4 × 8和8 × 4	2
所有其它情况	4

图10

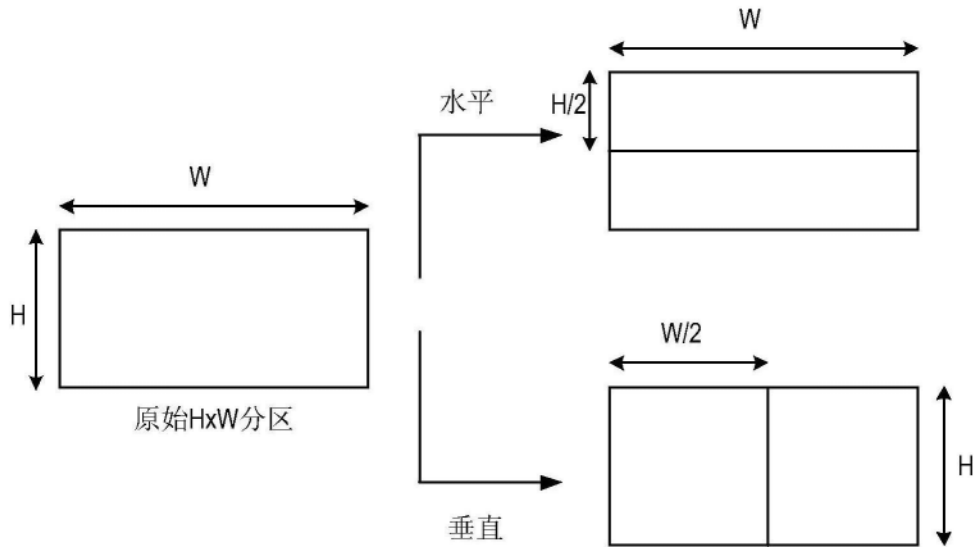


图11

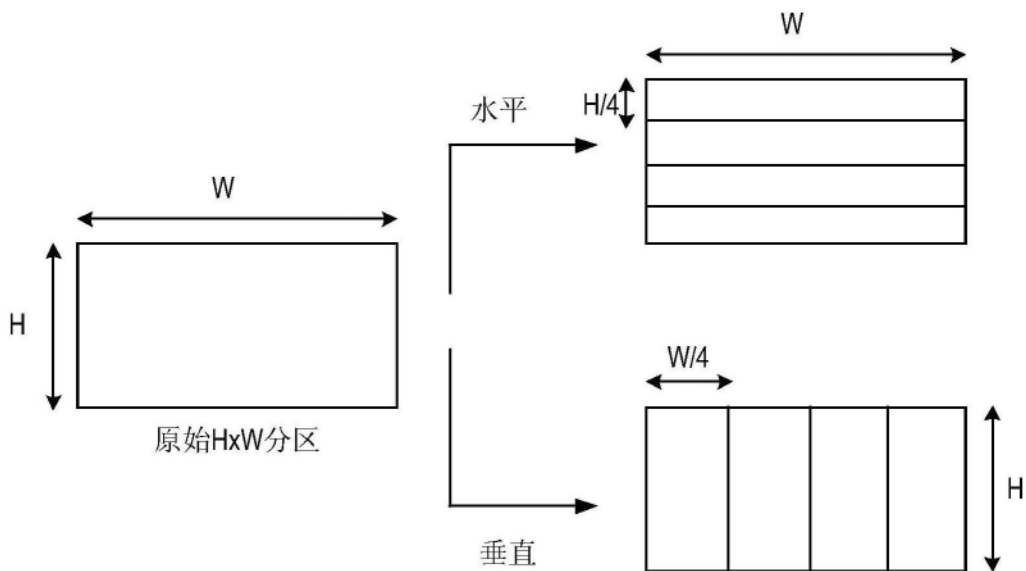


图12









## 继续图13D

{ 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, -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 }

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

{ 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 }

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

{ 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 }

其中

{ 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 }

图13E

针对N点输入的DCT-2、DST-7和DCT-8的变换基函数

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>where <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	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

图14

mts_idx	变换类型 水平	变换类型 垂直	
-1	0	0	
0	1	1	
1	2	1	
2	1	2	
3	2	2	

变换类型  
 0-DCT2  
 1-DST7  
 2-DCT8

图15

4点DST-7

```

{ a, b, c, d }
{ c, c, 0, -c }
{ d, -a, -c, b }
{ b, -d, c, -a }

```

其中 {a, b, c, d} = { 29, 55, 74, 84}

图16A

8点DST-7

```

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

```

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

图16B

16点DST-7

```

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, }
{ c, f, i, l, o, o, l, i, f, c, 0, -c, -f, -i, -l, -o, }
{ e, j, o, m, h, c, -b, -g, -l, -p, -k, -f, -a, d, i, n, }
{ g, n, l, e, -b, -i, -p, -j, -c, d, k, o, h, a, -f, -m, }
{ i, o, f, -c, -l, -l, -c, f, o, i, 0, -i, -o, -f, c, l, }
{ k, k, 0, -k, -k, 0, k, k, 0, -k, -k, 0, k, k, 0, -k, }
{ m, g, -f, -n, -a, l, h, -e, -o, -b, k, i, -d, -p, -c, j, }
{ o, c, -l, -f, i, i, -f, -l, c, o, 0, -o, -c, l, f, -i, }
{ p, -a, -o, b, n, -c, -m, d, l, -e, -k, f, j, -g, -i, h, }
{ n, -e, -i, j, d, -o, a, m, -f, -h, k, c, -p, b, l, -g, }
{ l, -i, -c, o, -f, -f, o, -c, -i, l, 0, -l, i, c, -o, f, }
{ j, -m, c, g, -p, f, d, -n, i, a, -k, l, -b, -h, o, -e, }
{ h, -p, i, -a, -g, o, -j, b, f, -n, k, -c, -e, m, -l, d, }
{ f, -l, o, -i, c, c, -i, o, -l, f, 0, -f, l, -o, i, -c, }
{ d, -h, l, -p, m, -i, e, -a, -c, g, -k, o, -n, j, -f, b, }
{ b, -d, f, -h, j, -l, n, -p, o, -m, k, -i, g, -e, c, -a, }

```

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}

图16C



4点DCT-8 :
$$\begin{aligned} & \{ a, b, c, d, \} \\ & \{ b, 0, -b, -b, \} \\ & \{ c, -b, -d, a, \} \\ & \{ d, -b, a, -c, \} \end{aligned}$$

其中  $\{a, b, c, d\} = \{84, 74, 55, 29\}$

图17A

8点DCT-8

$$\begin{aligned} & \{ a, b, c, d, e, f, g, h, \} \\ & \{ b, e, h, -g, -d, -a, -c, -f, \} \\ & \{ c, h, -e, -a, -f, g, b, d, \} \\ & \{ d, -g, -a, -h, c, e, -f, -b, \} \\ & \{ e, -d, -f, c, g, -b, -h, a, \} \\ & \{ f, -a, g, e, -b, h, d, -c, \} \\ & \{ g, -c, b, -f, -h, d, -a, e, \} \\ & \{ h, -f, d, -b, a, -c, e, -g, \} \end{aligned}$$

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

图17B

16点DCT-8

$$\begin{aligned} & \{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, \} \\ & \{ b, e, h, k, n, 0, -n, -k, -h, -e, -b, -b, -e, -h, -k, -n, \} \\ & \{ c, h, m, -p, -k, -f, -a, -e, -j, -o, n, i, d, b, g, l, \} \\ & \{ d, k, -p, -i, -b, -f, -m, n, g, a, h, o, -l, -e, -c, -j, \} \\ & \{ e, n, -k, -b, -h, 0, h, b, k, -n, -e, -e, -n, k, b, h, \} \\ & \{ f, 0, -f, -f, 0, f, f, 0, -f, -f, 0, f, f, 0, -f, -f, \} \\ & \{ g, -n, -a, -m, h, f, -o, -b, -l, i, e, -p, -c, -k, j, d, \} \\ & \{ h, -k, -e, n, b, 0, -b, -n, e, k, -h, -h, k, e, -n, -b, \} \\ & \{ i, -h, -j, g, k, -f, -l, e, m, -d, -n, c, o, -b, -p, a, \} \\ & \{ j, -e, -o, a, -n, -f, i, k, -d, -p, b, -m, -g, h, l, -c, \} \\ & \{ k, -b, n, h, -e, 0, e, -h, -n, b, -k, -k, b, -n, -h, e, \} \\ & \{ l, -b, i, o, -e, f, -p, -h, c, -m, -k, a, -j, -n, d, -g, \} \\ & \{ m, -e, d, -l, -n, f, -c, k, o, -g, b, -j, -p, h, -a, i, \} \\ & \{ n, -h, b, -e, k, 0, -k, e, -b, h, -n, -n, h, -b, e, -k, \} \\ & \{ o, -k, g, -c, b, -f, j, -n, -p, l, -h, d, -a, e, -i, m, \} \\ & \{ p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o, \} \end{aligned}$$

其中  $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p\} = \{90, 89, 87, 83, 81, 77, 72, 66, 62, 56, 49, 41, 33, 25, 17, 9\}$

图17C

32点DCT-8

{ 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, } =

{ { b, e, h, m, r, y, F, -A, -p, -e, -g, -t, 0, -t, -g, -t, 0, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { c, h, k, r, y, F, -Y, -p, -e, -g, -t, 0, t, g, t, 0, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { d, k, r, y, F, -Y, -p, -e, -g, -t, 0, t, g, t, 0, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { e, n, w, F, -A, -p, -e, -g, -t, 0, t, g, t, 0, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { f, q, B, -A, -p, -e, -g, -t, 0, t, g, t, 0, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { g, t, 0, -t, -g, -t, 0, t, g, t, 0, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { h, w, -B, -m, -c, -l, -C, t, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { i, z, -w, -f, -l, -C, t, r, c, o, F, -q, -a, -i, -v, k, F, -E, -B, -Y, -C, -h, -m, -r, -w, -i, -p, -D, C, v, o, h, a, g, n, u, B, -E, -x, -g, -j, -o, -x, } }

{ { 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, } }

{ { k, F, -m, -l, -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, } }

{ { 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, } }

{ { 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, -B, -C, -w, r, h, } }

{ { 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, } }

{ { o, -v, -h, C, a, D, -g, -w, n, P, -u, -i, B, C, E, -f, -z, i, w, -l, t, o, g, -r, -n, u, k, -x, h, A, e, -D, -b, } }

{ { p, -S, -m, V, j, -y, -g, B, d, E, -a, -a, -F, C, B, E, -f, -z, i, w, -l, t, o, g, -r, -n, u, k, -x, h, A, e, -D, -b, } }

{ { q, -S, -r, o, s, -n, -t, m, u, -l, -v, k, w, -j, -x, i, Y, -h, -z, o, g, -r, -n, u, k, -x, h, A, e, -D, -b, } }

{ { 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, } }

{ { 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, } }

{ { t, -g, 0, g, -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, } }

{ { 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, } }

{ { v, -a, w, u, -b, x, t, -c, Y, s, -d, z, r, -e, A, g, -f, B, p, -g, C, o, -h, D, n, -i, E, m, -j, F, l, -k, } }

{ { w, -C, r, B, -h, m, 0, -m, h, -B, -r, c, u, -c, p, F, -n, e, -w, y, g, -l, D, r, -a, s, C, k, h, -z, -v, d, -o, } }

{ { x, -f, m, -E, -g, b, -t, -B, j, -i, A, -k, f, -v, -B, l, -e, u, C, -m, d, -t, -D, n, -c, s, E, -o, a, -n, B, x, -j, e, -s, } }

{ { 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, a, -n, B, x, -j, e, -s, } }

{ { z, -l, c, -g, 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, } }

{ { 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, } }

{ { B, -r, h, -c, m, -w, 0, w, -m, c, -h, r, -B, r, -h, c, -m, w, 0, -w, m, -c, h, -r, B, -r, h, -c, m, -w, } }

{ { 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, } }

{ { 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, } }

{ { E, -A, w, -s, o, -k, g, -c, b, -f, j, -n, r, -v, z, -D, -F, B, -x, t, -p, l, -h, d, e, -i, m, -q, u, -y, C, } }

{ { 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, } }

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 }

图17D

seq_parameter_set_rbsp( ) {	描述符
.....	
sps_mts_enabled_flag	u(1)
if( sps_mts_enabled_flag ) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
.....	

图18

从帧内预测模式到变换集索引的映射

1900

帧内模式	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
集	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
帧内模式	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67 (LM)
集	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	NULL

图19

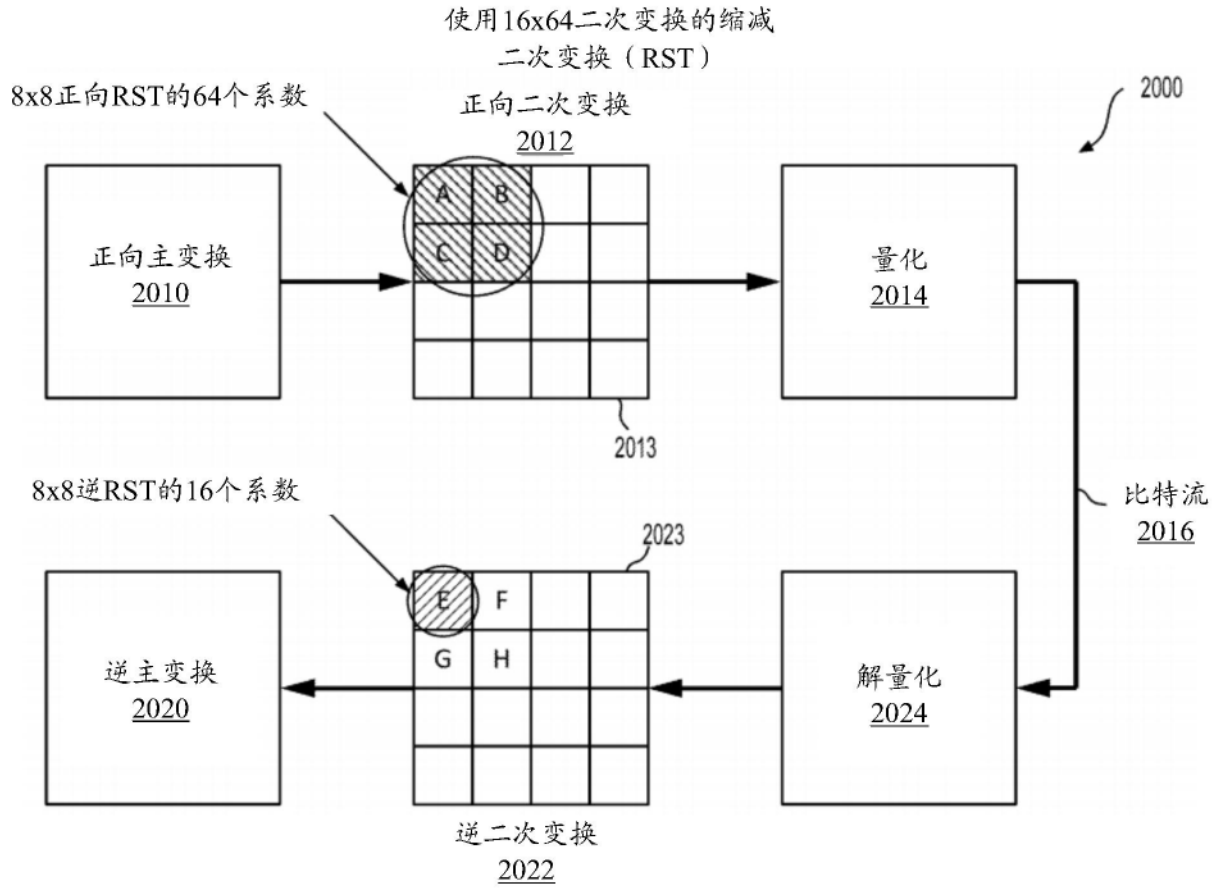


图20

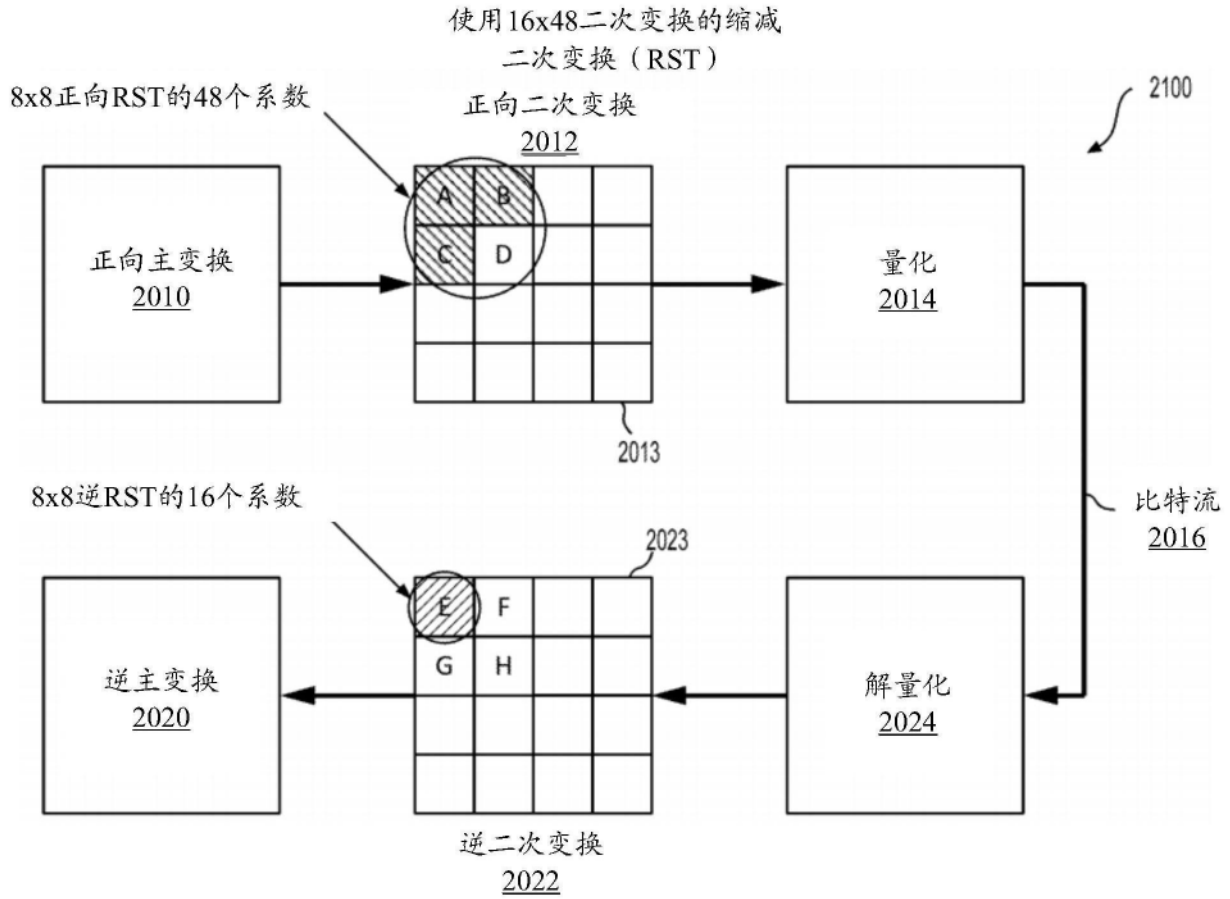


图21

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
.....	
numSigCoeff = 0	
numZeroOutSigCoeff = 0	
transform_tree( x0, y0, cbWidth, cbHeight, treeType )	
lfstWidth = ( treeType == DUAL_TREE_CHROMA ) ? cbWidth / SubWidthC : cbWidth	
lfstHeight = ( treeType == DUAL_TREE_CHROMA ) ? cbHeight / SubHeightC : cbHeight	
if( Min( lfstWidth, lfstHeight ) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[ x0 ][ y0 ] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && !intra_mip_flag[ x0 ][ y0 ] ) {	
if( ( numSigCoeff > ( ( treeType == SINGLE_TREE ) ? 2 : 1 ) ) && numZeroOutSigCoeff == 0 )	
lfst_idx[ x0 ][ y0 ]	ae(v)
}	
}	
}	
}	

图22



正向缩减变换和逆缩减变换

图23

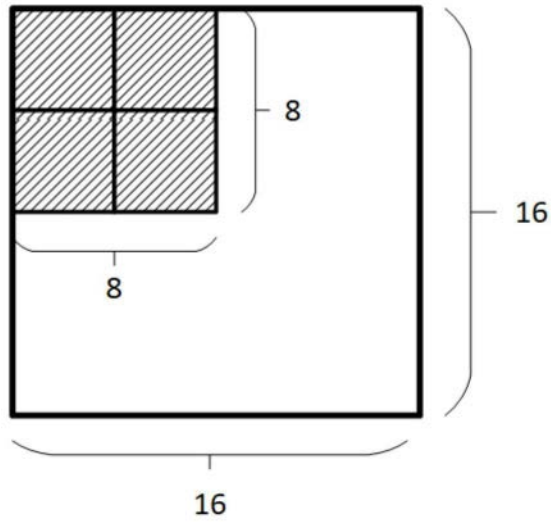


图24A

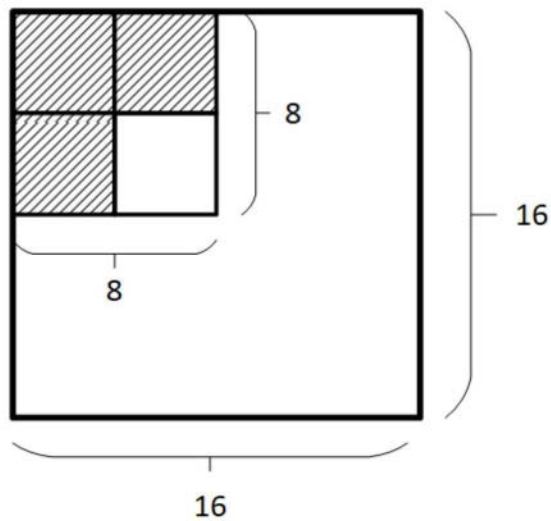


图24B

变换集选择表

帧内预测模式 ( IntraPredMode )	变换集索引
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode	1

图25

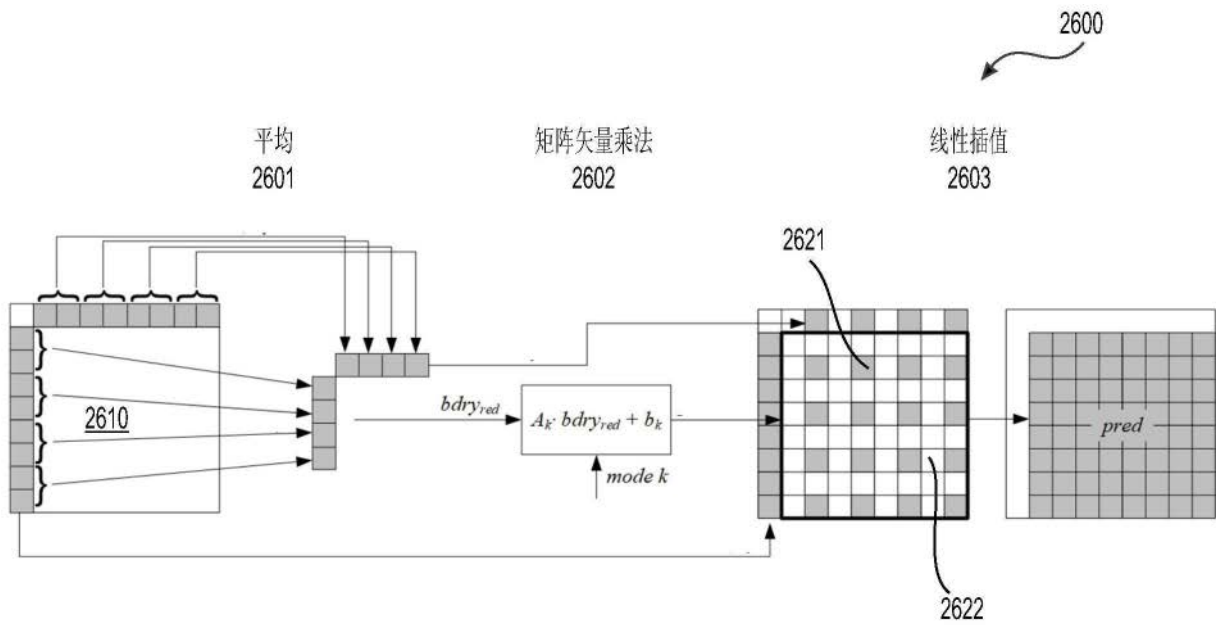


图26

2701

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
.....	
if( sps_mip_enabled_flag && ( Abs(Log2( cbWidth ) - Log2( cbHeight )) <= 2 ) && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY )	
<b>intra_mip_flag[x0][y0]</b>	ae(v)
if( intra_mip_flag[x0][y0] ) {	
<b>intra_mip_mpm_flag[x0][y0]</b>	ae(v)
if( intra_mip_mpm_flag[x0][y0] )	
<b>intra_mip_mpm_idx[x0][y0]</b>	ae(v)
else	
<b>intra_mip_mpm_remainder[x0][y0]</b>	ae(v)
} else {	
if( sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0) )	
<b>intra_luma_ref_idx[x0][y0]</b>	ae(v)
if( sps_isp_enabled_flag && intra_luma_ref_idx[x0][y0] == 0 && ( cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY ) && ( cbWidth * cbHeight > MinTbSizeY * MinTbSizeY ) )	
<b>intra_subpartitions_mode_flag[x0][y0]</b>	ae(v)
if( intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY )	
<b>intra_subpartitions_split_flag[x0][y0]</b>	ae(v)
if( intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0 )	
<b>intra_luma_mpm_flag[x0][y0]</b>	ae(v)
if( intra_luma_mpm_flag[x0][y0] ) {	
if( intra_luma_ref_idx[x0][y0] == 0 )	
<b>intra_luma_not_planar_flag[x0][y0]</b>	ae(v)
if( intra_luma_not_planar_flag[x0][y0] )	
<b>intra_luma_mpm_idx[x0][y0]</b>	ae(v)
} else	
<b>intra_luma_mpm_remainder[x0][y0]</b>	ae(v)
}	
}	
}	
}	
.....	

图27

常规 (General)

2801

2800

该过程的输入为:

- 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)x(nTbH)数组d[ x ][ y ], 其中  $x = 0..nTbW - 1$ ,  $y = 0..nTbH - 1$ 。

.....

变量implicitMtsEnabled导出如下:

- 如果sps\_mts\_enabled\_flag等于1并且以下条件之一为真, 则 implicitMtsEnabled设置为等于1:
  - IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT
  - cu\_sbt\_flag等于1且Max ( nTbW, nTbH ) 小于或等于32
  - sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0并且CuPredMode[ xTbY ][ yTbY ]等于MODE\_INTRA。
- 否则, implicitMtsEnabled设置为等于0。

2802

图28A

2803

继续图28A

2800

指定水平变换核的变量trTypeHor和指定垂直变换核的变量trTypeVer导出如下:

- 如果cIdx大于0, 则trTypeHor和trTypeVer设置为等于0。
- 否则, 如果implicitMtsEnabled等于1, 则以下适用:
- 如果IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT, 或者sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0并且CuPredMode[ xTbY ][ yTbY ]等于

MODE\_INTRA, 则trTypeHor和trTypeVer导出如下:

$$\text{trTypeHor} = (\text{nTbW} \geq 4 \ \&\& \ \text{nTbW} \leq 16) ? 1 : 0 \quad (3-1)$$

$$\text{trTypeVer} = (\text{nTbH} \geq 4 \ \&\& \ \text{nTbH} \leq 16) ? 1 : 0 \quad (3-2)$$

- 否则 ( cu\_sbt\_flag等于1 ), trTypeHor和trTypeVer在表II中根据cu\_sbt\_horizontal\_flag和cu\_sbt\_pos\_flag指定。

- 否则, trTypeHor和trTypeVer在表I中根据tu\_mts\_idx[xTbY][yTbY]指定。

.....

表I—取决于tu\_mts\_idx[x][y]的trTypeHor和trTypeVer的规范

tu_mts_idx[ x0 ][ y0 ]	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

表II—取决于cu\_sbt\_horizontal\_flag和cu\_sbt\_pos\_flag的trTypeHor和trTypeVer的规范

cu_sbt_horizontal_flag	cu_sbt_pos_flag	trTypeHor	trTypeVer
0	0	2	1
0	1	1	1
1	0	1	2
1	1	1	1

图28B

变量implicitMtsEnabled导出如下:

- 如果sps\_mts\_enabled\_flag等于1并且以下条件之一为真, 则implicitMtsEnabled设置为等于1:
  - IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT
  - cu\_sbt\_flag等于1且Max ( nTbW, nTbH ) 小于或等于32
  - ~~sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都~~等于0并且CuPredMode[ xTbY ][ yTbY ]等于MODE\_INTRA。
- 否则, implicitMtsEnabled设置为等于0。

图29

变量implicitMtsEnabled导出如下:

- 如果sps\_mts\_enabled\_flag等于1并且以下条件之一为真, 则implicitMtsEnabled设置为等于1:
  - IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT
  - cu\_sbt\_flag等于1且Max ( nTbW, nTbH ) 小于或等于32
  - sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0并且CuPredMode[ xTbY ][ yTbY ]等于MODE\_INTRA并且sps\_lfnst\_enabled\_flag等于0。
- 否则, implicitMtsEnabled设置为等于0。

图30

变量implicitMtsEnabled导出如下：

- 如果sps\_mts\_enabled\_flag等于1并且以下条件之一为真，则implicitMtsEnabled设置为等于1：
  - IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT
  - cu\_sbt\_flag等于1且Max ( nTbW, nTbH ) 小于或等于32
  - sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0并且CuPredMode[ xTbY ][ yTbY ]等于MODE\_INTRA并且sps\_mip\_enabled\_flag等于0。
- 否则， implicitMtsEnabled设置为等于0。

图31

变量implicitMtsEnabled导出如下：

- 如果sps\_mts\_enabled\_flag等于1并且以下条件之一为真，则implicitMtsEnabled设置为等于1：
  - IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT
  - cu\_sbt\_flag等于1且Max ( nTbW, nTbH ) 小于或等于32
  - sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0并且CuPredMode[ xTbY ][ yTbY ]等于MODE\_INTRA
  - tu\_mts\_idx[ x0 ][ y0 ]和lfnst\_idx[ x0 ][ y0 ]都等于0。
- 否则， implicitMtsEnabled设置为等于0。

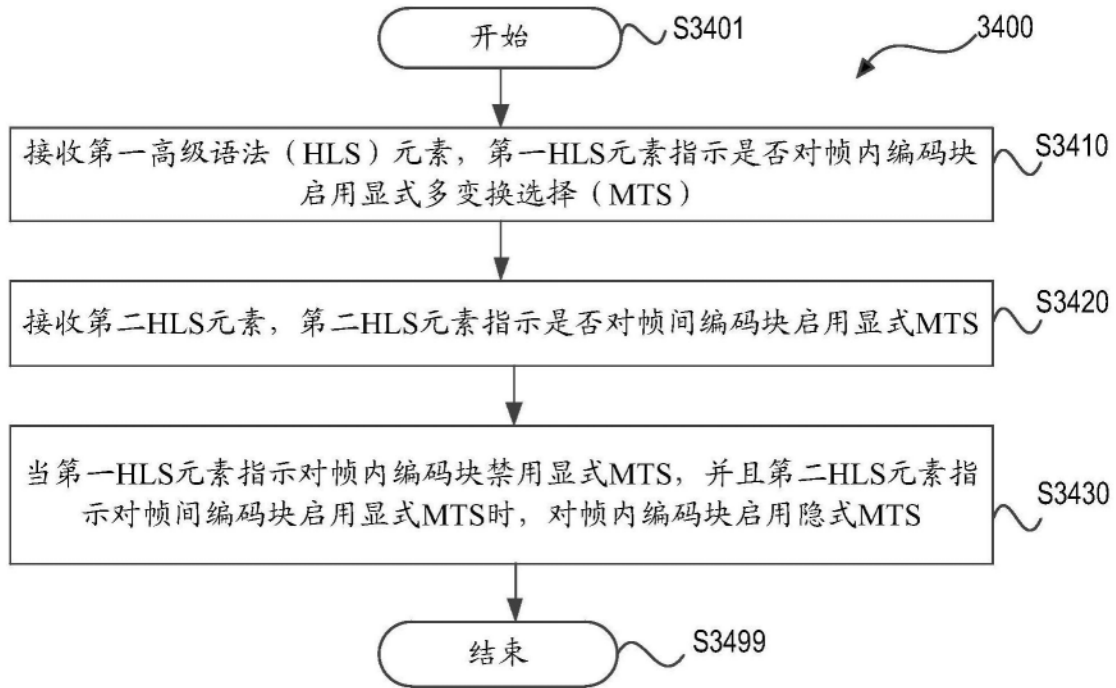
图32

变量implicitMtsEnabled导出如下：

- 如果sps\_mts\_enabled\_flag等于1并且以下条件之一为真，则implicitMtsEnabled设置为等于1：
  - IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT
  - cu\_sbt\_flag等于1且Max (nTbW, nTbH) 小于或等于32
  - ~~sps\_explicit\_mts\_intra\_enabled\_flag和sps\_explicit\_mts\_inter\_enabled\_flag都等于0并且CuPredMode[xTbY][yTbY]等于MODE\_INTRA~~
  - tu\_mts\_idx[x0][y0]、intra\_mip\_flag[x0][y0]和lfnst\_idx[x0][y0]都等于0。
- 否则，implicitMtsEnabled设置为等于0。

3300

图33



3400

图34

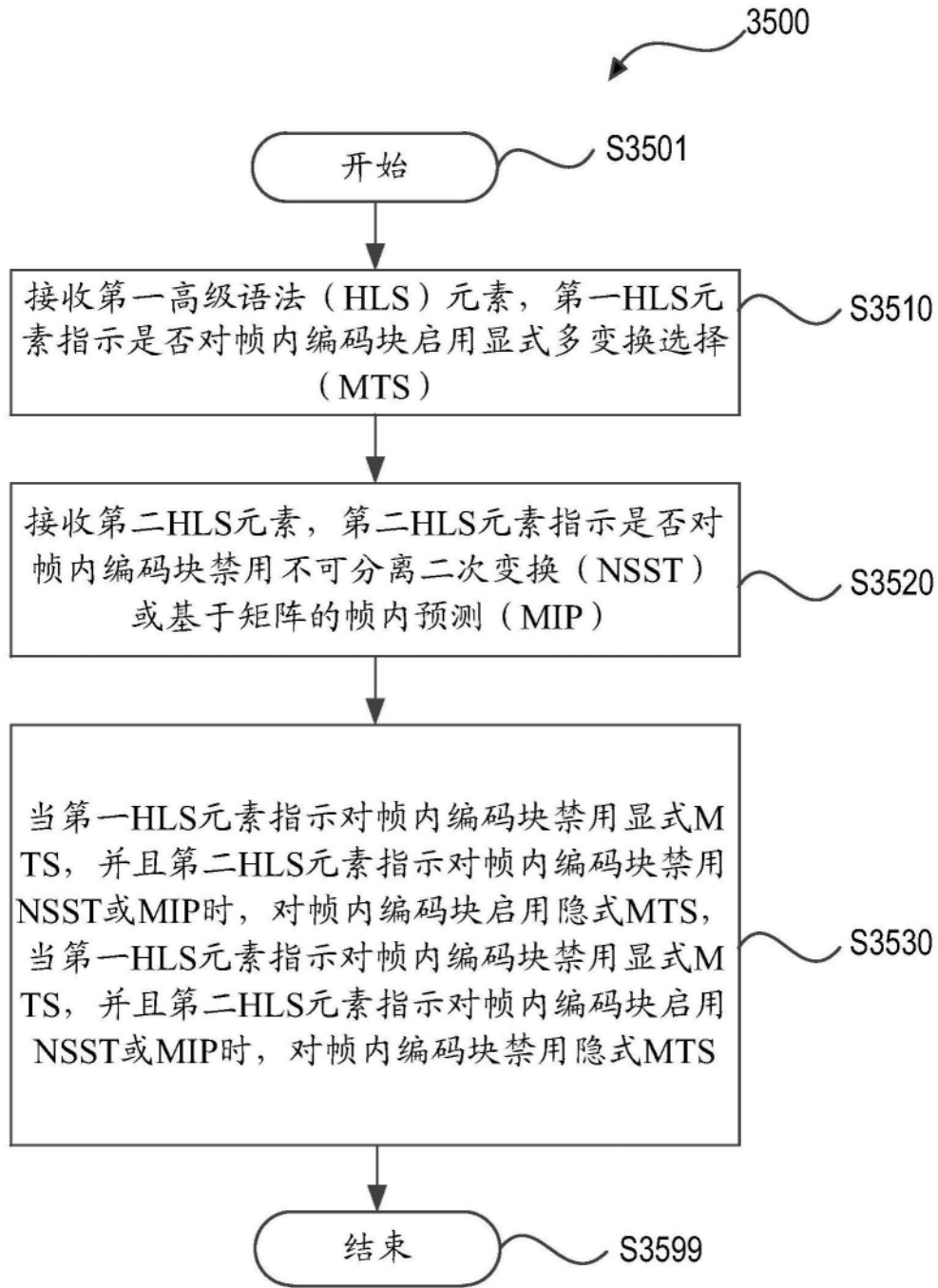


图35

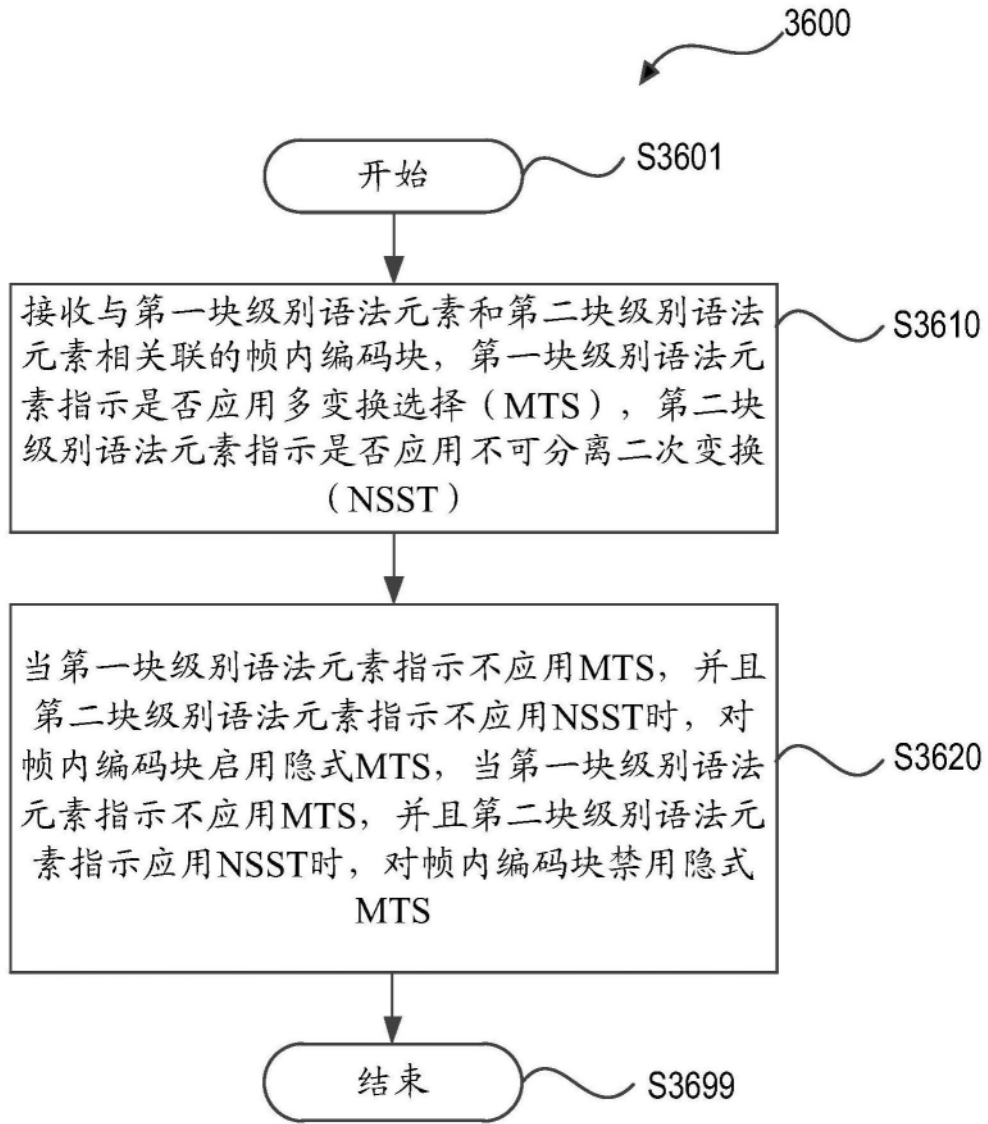


图36

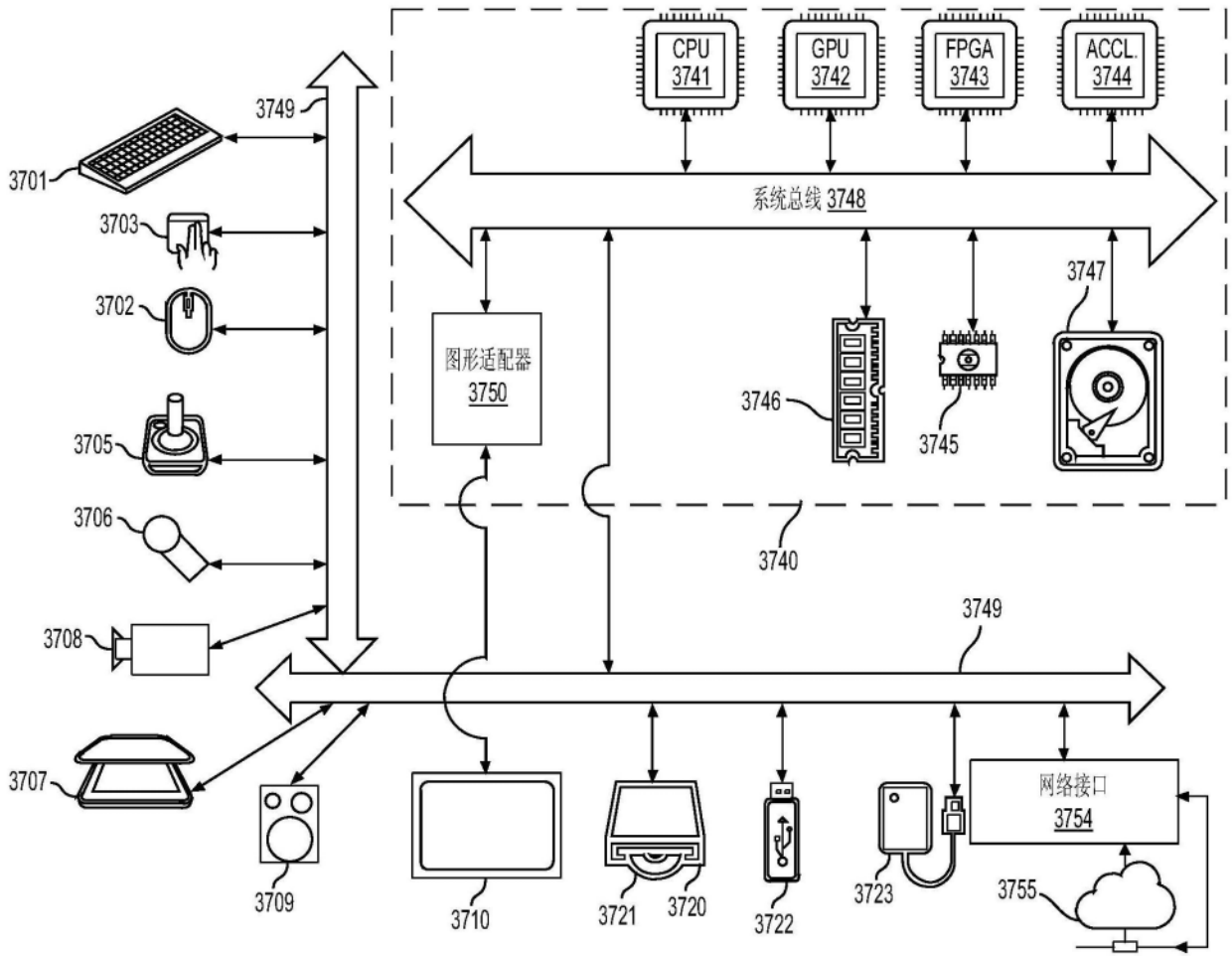


图37