



(10) 授权公告号 CN 112335241 B

(45) 授权公告日 2023. 10. 20

(21) 申请号 201980041724.1

(22) 申请日 2019.05.29

(65) 同一申请的已公布的文献号
申请公布号 CN 112335241 A

(43) 申请公布日 2021.02.05

(30) 优先权数据
2018204786 2018.06.29 AU

(85) PCT国际申请进入国家阶段日
2020.12.21

(86) PCT国际申请的申请数据
PCT/AU2019/050534 2019.05.29

(87) PCT国际申请的公布数据
W02020/000019 EN 2020.01.02

(73) 专利权人 佳能株式会社
地址 日本东京都大田区下丸子3丁目30番2号

(72) 发明人 克里斯托弗·詹姆斯·罗斯沃恩

(74) 专利代理机构 北京魏启学律师事务所
11398

专利代理师 魏启学

(51) Int.Cl.
H04N 19/12 (2006.01)
H04N 19/91 (2006.01)
H04N 19/60 (2006.01)
H04N 19/13 (2006.01)
H04N 19/70 (2006.01)

(56) 对比文件
CN 107835432 A, 2018.03.23
US 2004126021 A1, 2004.07.01
WO 2018011345 A1, 2018.01.18
CN 104221383 A, 2014.12.17
US 2014056361 A1, 2014.02.27
PURI S. et al..CNN-Based Transform Index Prediction in Multiple Transforms Framework to Assist Entropy Coding.《25th European Signal Processing Conference (EUSIPCO)》.2017,

审查员 陈瑞宁

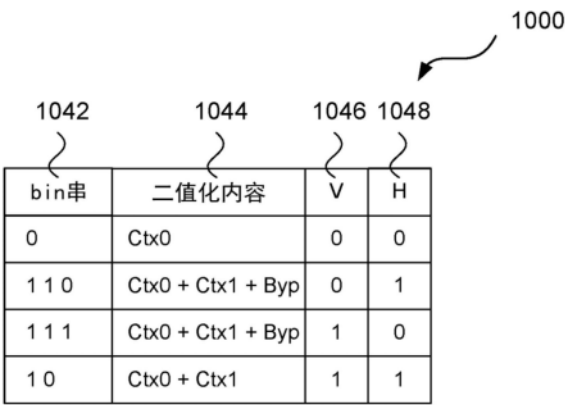
权利要求书2页 说明书21页 附图15页

(54) 发明名称

对视频样本的变换块编码和解码的方法、设备和系统

(57) 摘要

本发明提供一种用于从位流中解码图像帧中的变换块的方法(1200)。该方法包括：从位流中解码单个截断一元二值化内容，该单个截断一元二值化内容用于图像帧的变换块的水平变换和垂直变换。然后，该方法基于解码得到的单个截断一元二值化内容来确定水平变换和垂直变换的类型，并且通过将所确定的水平变换和垂直变换的类型应用于图像的变换块来对图像帧中的变换块进行解码。



1. 一种用于从位流中解码图像中的变换块的方法,所述方法包括:
通过使用算术解码,从所述位流中至少解码单个截断一元码字;
通过使用所述单个截断一元码字,针对所述变换块确定水平变换类型和垂直变换类型,其中,通过使用所述单个截断一元码字,能够针对所述水平变换类型和所述垂直变换类型确定不同的变换类型;以及
通过将所确定的水平变换类型和所确定的垂直变换类型应用于所述变换块,来对所述变换块进行解码,
其中,在所述单个截断一元码字对应于由多个bin组成的bin串的情况下,分别与所述多个bin中的不同bin相对应的上下文值是相互不同的。
2. 根据权利要求1所述的方法,其中,所述水平变换类型和所述垂直变换类型的选项包括DST-VII和DCT-VIII。
3. 一种用于将图像中的变换块编码到流中的方法,所述方法包括:
针对所述变换块确定水平变换类型;
针对所述变换块确定垂直变换类型;
通过将所确定的水平变换类型和所确定的垂直变换类型应用于所述变换块,来对所述变换块进行编码;以及
通过使用算术编码,至少对单个截断一元码字进行编码,
其中,在解码所述变换块的方法中,所述水平变换类型和所述垂直变换类型是通过使用所述单个截断一元码字针对所述变换块确定的,
其中,通过使用所述单个截断一元码字,在解码所述变换块的方法中能够针对所述水平变换类型和所述垂直变换类型确定不同的变换类型,以及
其中,在所述单个截断一元码字对应于由多个bin组成的bin串的情况下,分别与所述多个bin中的不同bin相对应的上下文值是相互不同的。
4. 根据权利要求3所述的方法,其中,所述水平变换类型和所述垂直变换类型的选项包括DST-VII和DCT-VIII。
5. 一种用于从位流中解码图像中的变换块的设备,所述设备包括:
解码单元,其被配置为通过使用算术解码,从所述位流中至少解码单个截断一元码字;
以及
确定单元,其被配置为通过使用所述单个截断一元码字,针对所述变换块确定水平变换类型和垂直变换类型,其中,通过使用所述单个截断一元码字,能够针对所述水平变换类型和所述垂直变换类型确定不同的变换类型,
其中,所述解码单元被配置为通过将所确定的水平变换类型和所确定的垂直变换类型应用于所述变换块,来对所述变换块进行解码,以及
其中,在所述单个截断一元码字对应于由多个bin组成的bin串的情况下,分别与所述多个bin中的不同bin相对应的上下文值是相互不同的。
6. 一种用于将图像中的变换块编码到流中的设备,所述设备包括:
确定单元,其被配置为针对所述变换块确定水平变换类型,并且针对所述变换块确定垂直变换类型;以及
编码单元,其被配置为通过将所确定的水平变换类型和所确定的垂直变换类型应用于

所述变换块,来对所述变换块进行编码,

其中,所述编码单元被配置为通过使用算术编码,至少对单个截断一元码字进行编码,

其中,在用于解码所述变换块的设备中,所述水平变换类型和所述垂直变换类型是通过使用所述单个截断一元码字针对所述变换块确定的,

其中,通过使用所述单个截断一元码字,在用于解码所述变换块的设备中能够针对所述水平变换类型和所述垂直变换类型确定不同的变换类型,以及

其中,在所述单个截断一元码字对应于由多个bin组成的bin串的情况下,分别与所述多个bin中的不同bin相对应的上下文值是相互不同的。

7.一种包括计算机程序的非暂时性计算机可读存储介质,所述计算机程序包括指令,在由计算机执行所述计算机程序时,所述指令使得所述计算机执行用于从位流中解码图像中的变换块的方法,所述方法包括:

通过使用算术解码,从所述位流中至少解码单个截断一元码字;

通过使用所述单个截断一元码字,针对所述变换块确定水平变换类型和垂直变换类型,其中,通过使用所述单个截断一元码字,能够针对所述水平变换类型和所述垂直变换类型确定不同的变换类型;以及

通过将所确定的水平变换类型和所确定的垂直变换类型应用于所述变换块,来对所述变换块进行解码,

其中,在所述单个截断一元码字对应于由多个bin组成的bin串的情况下,分别与所述多个bin中的不同bin相对应的上下文值是相互不同的。

8.一种包括计算机程序的非暂时性计算机可读存储介质,所述计算机程序包括指令,在由计算机执行所述计算机程序时,所述指令使得所述计算机执行用于将图像中的变换块编码到位流中的方法,所述方法包括:

针对所述变换块确定水平变换类型;

针对所述变换块确定垂直变换类型;

通过将所确定的水平变换类型和所确定的垂直变换类型应用于所述变换块,来对所述变换块进行编码;以及

通过使用算术编码,至少对单个截断一元码字进行编码,

其中,在解码所述变换块的方法中,所述水平变换类型和所述垂直变换类型是通过使用所述单个截断一元码字针对所述变换块确定的,

其中,通过使用所述单个截断一元码字,在解码所述变换块的方法中能够针对所述水平变换类型和所述垂直变换类型确定不同的变换类型,以及

其中,在所述单个截断一元码字对应于由多个bin组成的bin串的情况下,分别与所述多个bin中的不同bin相对应的上下文值是相互不同的。

对视频样本的变换块编码和解码的方法、设备和系统

技术领域

[0001] 本发明通常涉及数字视频信号处理,尤其涉及用于对视频样本的变换块进行编码和解码的方法、设备和系统。本发明还涉及包括记录有用于对视频样本的变换块进行编码和解码的计算机程序的计算机可读介质的计算机程序产品。

背景技术

[0002] 当前存在包括用于传输和存储视频数据的应用的许多视频编码用的应用。还开发了许多视频编码标准并且其它视频编码标准当前正在开发中。视频编码标准化的最新进展已导致形成被称为“联合视频专家组”(JVET)的组。该联合视频专家组(JVET)包括:已知为“视频编码专家组”(VCEG)的国际电信联盟(ITU)的电信标准化部门(ITU-T)的研究组16、问题6(SG16/Q6)的成员;以及还已知为“运动图片专家组”(MPEG)的国际标准化组织/国际电工委员会联合技术委员会1/小组委员会29/工作组11(ISO/IEC JTC1/SC29/WG11)的成员。

[0003] 联合视频专家组(JVET)发布了提案征集(CfP),并在美国圣地亚哥市举行的第10次会议上对答复进行了分析。所提交的答复表明,视频压缩能力明显优于当前最先进的视频压缩标准(即,“高效率视频编码”(HEVC))的视频压缩能力。基于该优异表现,决定开始用以开发命名为“通用视频编码”(VVC)的新视频压缩标准的项目。预计VVC将特别是随着视频格式的能力的增加(例如,具有更高的分辨率和更高的帧频)解决针对甚至更高的压缩性能的持续需求、以及解决针对通过WAN的服务提供(其中,带宽成本相对较高)的日益增长的市场需求。同时,VVC必须可在当代硅工艺中实现,并且在所实现的性能与实现成本之间(例如,在硅面积、CPU处理器负荷、内存利用率和带宽方面)提供可接受的折衷。

[0004] 视频数据包括各自包括一个或多个颜色通道的图像数据的帧序列。通常,存在一个主要颜色通道和两个次要颜色通道。主要颜色通道通常被称为“亮度”通道,并且(一个或多个)次要颜色通道通常被称为“色度”通道。尽管视频数据通常在RGB(红-绿-蓝)颜色空间中显示,但该颜色空间在三个相应分量之间具有高度相关性。编码器或解码器所看到的视频数据表示通常使用诸如YCbCr等的颜色空间。YCbCr将亮度集中在Y(主要)通道中,并且将色度集中在Cb和Cr(次要)通道中。此外,可以以与亮度通道相比更低的速率(例如,在水平方向上为一半且在垂直方向上为一半(被称为“4:2:0色度格式”))对Cb和Cr通道进行采样。

[0005] VVC是“基于块”的编解码,其中将帧分割成块,并且按特定顺序处理块。针对各块,生成该块的内容的预测,并且形成该预测与被视为向编码器的输入的实际块内容之间的差(或空间域中的“残差”)的表示。该差可被编码为残差系数的序列,该残差系数的序列是通过将诸如离散余弦变换(DCT)或其它变换等的正向主要变换应用于残差值的块而得到的。将该主要变换分别应用于残差样本的块,即分两遍进行二维变换,其中首先通过对各行应用一维变换来对行进行变换,然后通过对各列应用一维变换来对该结果进行变换,以产生基本上对残差样本进行去相关的残差系数的块。此外,多个变换可供用在该行和列处理中,这需要指示要使用哪个变换的信令。尽管用以将不同变换用在块中的选项提供了压缩优势,但该压缩优势因对索引进行编码以选择在水平方向和垂直方向上要使用哪个变换的需

求而略微减少。该信令的高效编码对该信令开销的影响是有益的。

发明内容

[0006] 本发明的目的是大致克服或至少改进现有配置的一个或多个缺点。

[0007] 本发明的一方面提供一种用于从位流中解码图像帧中的变换块的方法，所述方法包括：从所述位流中解码单个截断一元二值化内容，所述单个截断一元二值化内容用于所述图像帧的变换块的水平变换和垂直变换；基于解码得到的单个截断一元二值化内容来确定所述水平变换和所述垂直变换的类型；以及通过将所确定的所述水平变换和所述垂直变换的类型应用于图像的所述变换块，来对所述图像帧中的所述变换块进行解码。

[0008] 本发明的另一方面提供一种包括计算机应用程序的非暂时性计算机可读介质，所述程序能够由计算机设备执行，以使计算机进行用于从位流中解码图像帧中的变换块的方法，所述方法包括：从所述位流中解码单个截断一元二值化内容，所述单个截断一元二值化内容用于所述图像帧的变换块的水平变换和垂直变换；基于解码得到的单个截断一元二值化内容来确定所述水平变换和所述垂直变换的类型；以及通过将所确定的所述水平变换和所述垂直变换的类型应用于图像的所述变换块，来对所述图像帧中的所述变换块进行解码。

[0009] 还公开了其它方面。

附图说明

[0010] 现在将参考以下的附图和附录来说明本发明的至少一个实施例，其中：

[0011] 图1是示出视频编码和解码系统的示意框图；

[0012] 图2A和2B构成可以实践图1的视频编码和解码系统的其中一个或这两者的通用计算机系统的示意框图；

[0013] 图3是示出视频编码器的功能模块的示意框图；

[0014] 图4是示出视频解码器的功能模块的示意框图；

[0015] 图5是示出通用视频编码的树结构中的块向一个或多个块的可用分割的示意框图；

[0016] 图6是用以在通用视频编码的树结构中实现块向一个或多个块的许可分割的数据流的示意图；

[0017] 图7是编码树单位(CTU)向多个编码单位(CU)的示例分割；

[0018] 图8A是示出帧内预测模式的图；

[0019] 图8B是示出根据变换索引从变换集到特定变换的映射的表；

[0020] 图8C是示出在水平方向和垂直方向上针对变换块从帧内预测模式到变换集的映射的表；

[0021] 图9是示出图3的视频编码器或图4的视频解码器的逆变换模块的示意框图；

[0022] 图10A是示出截断一元句法元素与在水平方向和垂直方向上编索引的变换所用的变换索引之间的关系图；

[0023] 图10B是示出截断一元句法元素与在水平方向和垂直方向上编索引的变换所用的变换索引之间的替代关系的图；

[0024] 图11是根据显式变换选择来进行主要变换以将残差系数的变换块编码到位流中的方法的流程图;以及

[0025] 图12是根据显式变换选择通过主要逆变换来从位流中解码残差系数的变换块的方法的流程图。

具体实施方式

[0026] 在任一个或多个附图中参考具有相同附图标记的步骤和/或特征的情况下,除非出现相反意图,否则这些步骤和/或特征为了本说明书的目的而具有相同的(一个或多个)功能或(一个或多个)操作。

[0027] 图1是示出视频编码和解码系统100的功能模块的示意框图。系统100可以利用系数扫描方法来提高压缩效率和/或实现降低的实现成本。系统100包括源装置110和目的地装置130。通信通道120用于从源装置110向目的地装置130通信编码视频信息。在一些配置中,源装置110和目的地装置130中的一个或两个分别可以包括移动电话手机或“智能电话”,其中在这种情况下,通信通道120是无线通道。在其它配置中,源装置110和目的地装置130可以包括视频会议设备,其中在这种情况下,通信通道120通常是诸如因特网连接等的有线通道。此外,源装置110和目的地装置130可以包括范围广泛的任意装置,其中这些装置包括支持空中电视广播、有线电视应用、因特网视频应用(包括流传输)、以及在某些计算机可读存储介质(诸如文件服务器中的硬盘驱动器等)上捕获编码视频数据的应用的装置。

[0028] 如图1所示,源装置110包括视频源112、视频编码器114和发送器116。视频源112通常包括所捕获视频帧数据(表示为113)的源,诸如摄像传感器、存储在非暂时性记录介质上的先前捕获到的视频序列、或者来自远程摄像传感器的视频馈送。视频源112也可以是计算机显卡的输出(例如,显示操作系统和在计算装置(例如,平板计算机)上执行的各种应用的视频输出)。可以包括摄像传感器作为视频源112的源装置110的示例包括智能电话、视频摄录机、专业摄像机和网络视频照相机。如进一步参考图3所述,视频编码器114将来自视频源112的(由箭头113指示的)所捕获帧数据转换(或“编码”)成(由箭头115指示的)位流。位流115由发送器116经由通信通道120作为编码视频数据(或“编码视频信息”)进行发送。位流115也可以存储在诸如“闪存”存储器或硬盘驱动器等非暂时性存储装置122中,直到随后通过通信通道120发送或者作为通过通信通道120的发送的代替为止。

[0029] 目的地装置130包括接收器132、视频解码器134和显示装置136。接收器132从通信通道120接收编码视频数据并将所接收到的视频数据作为(由箭头133指示的)位流传递至视频解码器134。然后,视频解码器134将(由箭头135指示的)解码后的帧数据输出至显示装置136。显示装置136的示例包括阴极射线管、液晶显示器(诸如在智能电话、平板计算机、计算机监视器、或者单机型电视机中等)。也可以将源装置110和目的地装置130各自的功能体现在单个装置中,该单个装置的示例包括移动电话手机和平板计算机。

[0030] 尽管以上说明了示例装置,但源装置110和目的地装置130各自通常经由硬件组件和软件组件的组合可以配置在通用计算机系统内。图2A示出这种计算机系统200,该计算机系统200包括:计算机模块201;诸如键盘202、鼠标指示器装置203、扫描器226、可被配置为视频源112的照相机227、以及麦克风280等的输入装置;以及包括打印机215、可被配置为显示装置136的显示装置214、以及扬声器217的输出装置。计算机模块201可以使用外部调制

器-解调器(调制解调器)收发器装置216来经由接线221与通信网络220进行通信。可以表示通信通道120的通信网络220可以是广域网(WAN),诸如因特网、蜂窝电信网络或私有WAN等。在接线221是电话线的情况下,调制解调器216可以是传统的“拨号上网”调制解调器。可替代地,在接线221是高容量(例如,线缆或光学的)接线的情况下,调制解调器216可以是宽带调制解调器。还可以使用无线调制解调器来进行向通信网络220的无线连接。收发器装置216可以提供发送器116和接收器132的功能,并且通信通道120可以体现在接线221中。

[0031] 计算机模块201通常包括至少一个处理器单元205和存储器单元206。例如,存储器单元206可以具有半导体随机存取存储器(RAM)和半导体只读存储器(ROM)。计算机模块201还包括多个输入/输出(I/O)接口,其中这多个输入/输出(I/O)接口包括:音频-视频接口207,其连接至视频显示器214、扬声器217和麦克风280;I/O接口213,其连接至键盘202、鼠标203、扫描器226、照相机227以及可选的操纵杆或其它人机接口装置(未示出);以及外部调制解调器216和打印机215所用的接口208。从音频-视频接口207向计算机监视器214的信号通常是计算机显卡的输出。在一些实现中,调制解调器216可以内置于计算机模块201内,例如内置于接口208内。计算机模块201还具有本地网络接口211,其中该本地网络接口211允许计算机系统200经由接线223连接至已知为局域网(LAN)的局域通信网络222。如图2A所示,局域通信网络222还可以经由接线224连接至广域网220,其中该局域通信网络222通常包括所谓的“防火墙”装置或具有相似功能的装置。本地网络接口211可以包括以太网(EthernetTM)电路卡、蓝牙(BluetoothTM)无线配置或IEEE 802.11无线配置;然而,对于接口211,可以实践多种其它类型的接口。本地网络接口211还可以提供发送器116和接收器132的功能,并且通信通道120也可以体现在局域通信网络222中。

[0032] I/O接口208和213可以提供串行连接和并行连接中的任一个或这两者,其中前者通常根据通用串行总线(USB)标准来实现并且具有相应的USB连接器(未示出)。设置有存储装置209,并且存储装置209通常包括硬盘驱动器(HDD)210。还可以使用诸如软盘驱动器和磁带驱动器等的其它存储装置(未示出)。通常设置有光盘驱动器212以用作数据的非易失性源。可以使用例如光盘(例如,CD-ROM、DVD、蓝光盘(Blu-ray DiscTM))、USB-RAM、便携式外部硬盘驱动器和软盘等的便携式存储器装置作为针对计算机系统200的数据的适当源。通常,HDD 210、光盘驱动器212、网络220和222中的任意还可被配置成作为视频源112进行工作、或者作为为了经由显示器214进行再现所要存储的解码视频数据的目的地进行工作。系统100的源装置110和目的地装置130、或者系统100的源装置110和目的地装置130可以体现在计算机系统200中。

[0033] 计算机模块201的组件205-213通常经由互连总线204并且以得到相关领域技术人员已知的计算机系统200的传统操作模式的方式进行通信。例如,处理器205使用接线218连接至系统总线204。同样,存储器206和光盘驱动器212通过接线219连接至系统总线204。可以实践所述配置的计算机的示例包括IBM-PC和兼容机、Sun SPARCstation、Apple MacTM或相似的计算机系统。

[0034] 在适当或期望的情况下,可以使用计算机系统200来实现视频编码器114和视频解码器134以及以下所述的方法,其中可以将视频编码器114、视频解码器134和要说明的方法作为在计算机系统200内可执行的一个或多个软件应用程序233来实现。特别地,利用软件233中的在计算机系统200内执行的指令231(参见图2B)来实现视频编码器114、视频解码器

134和所述方法的步骤。可以将软件指令231形成各自用于进行一个或多个特定任务的一个或多个代码模块。还可以将软件分割成两个单独部分,其中第一部分和相应的代码模块进行所述方法,并且第二部分和相应的代码模块管理第一部分和用户之间的用户界面。

[0035] 例如,可以将软件存储在包括以下所述的存储装置的计算机可读介质中。将软件从计算机可读介质载入计算机系统200,然后由计算机系统200来执行。具有这样的软件的计算机可读介质或者该计算机可读介质上所记录的计算机程序是计算机程序产品。在计算机系统200中使用该计算机程序产品优选地实现了用于实施视频编码器114、视频解码器134和所述方法的有利设备。

[0036] 通常将软件233存储在HDD 210或存储器206中。将该软件从计算机可读介质载入计算机系统200,并且由计算机系统200来执行。因而,例如,可以将软件233存储在光盘驱动器212所读取的光学可读盘存储介质(例如,CD-ROM)225上。

[0037] 在一些实例中,将应用程序233以编码在一个或多个CD-ROM 225上并且经由相应的驱动器212进行读取的方式供给至用户,或者可替代地,可以由用户从网络220或222读取应用程序233。更进一步地,还可以将软件从其它计算机可读介质载入计算机系统200。计算机可读存储介质是指将所记录的指令和/或数据提供至计算机系统200以供执行和/或处理的任何非暂时性有形存储介质。这种存储介质的示例包括软盘、磁带、CD-ROM、DVD、蓝光盘(Blu-ray Disc™)、硬盘驱动器、ROM或集成电路、USB存储器、磁光盘、或者诸如PCMCIA卡等的计算机可读卡等,而与这些装置在计算机模块201的内部还是外部无关。还可以参与将软件、应用程序、指令和/或视频数据或编码视频数据提供至计算机模块401的暂时性或非有形计算机可读传输介质的示例包括:无线电或红外线传输通道及向着其它计算机或联网装置的网络接线、以及包括电子邮件发送和网站上所记录的信息等的因特网或内联网。

[0038] 可以执行上述的应用程序233的第二部分和相应的代码模块来实现要绘制或以其它方式呈现在显示器214上的一个或多个图形用户界面(GUI)。通过典型地对键盘202和鼠标203进行操作,计算机系统200的用户和应用可以以在功能上可适用的方式对界面进行操作,以将控制命令和/或输入提供至与这些(一个或多个)GUI相关联的应用。还可以实现在功能上可适用的其它形式的用户界面,诸如利用经由扬声器217所输出的语音提示和经由麦克风280所输入的用户声音命令的音频界面等。

[0039] 图2B是处理器205和“存储器”234的详细示意框图。存储器234表示图2A中的计算机模块201可以访问的(包括HDD 209和半导体存储器206的)所有存储器模块的逻辑聚合。

[0040] 在初始对计算机模块201通电的情况下,执行上电自检(power-on self-test, POST)程序250。通常将POST程序250存储在图2A的半导体存储器206的ROM249中。有时将诸如存储有软件的ROM 249等的硬件装置称为固件。POST程序250检查计算机模块201内的硬件以确保适当工作,并且通常检查处理器205、存储器234(209,206)和通常还存储在ROM 249中的基本输入-输出系统软件(BIOS)模块251,以进行正确操作。一旦POST程序250成功运行,BIOS251启动图2A的硬盘驱动器210。启动硬盘驱动器210使得经由处理器205执行驻留在硬盘驱动器210上的引导装入程序252。这样将操作系统253载入RAM存储器206,其中在该RAM存储器206上,操作系统253开始工作。操作系统253是处理器205可执行的系统级应用,以实现包括处理器管理、存储器管理、装置管理、存储管理、软件应用接口和通用用户界面等的各种高级功能。

[0041] 操作系统253管理存储器234(209,206),以确保计算机模块201上运行的各处理或应用具有在不会与分配至其它处理的内存冲突的情况下执行的充足内存。此外,必须适当使用图2A的计算机系统200中可用的不同类型的存储器,以使得各处理可以高效地运行。因此,聚合存储器234并不意图例示如何分配存储器的特定分段(除非另外说明),而是提供计算机系统200可访问的存储器的概述图以及如何使用该存储器。

[0042] 如图2B所示,处理器205包括多个功能模块,其中这多个功能模块包括控制单元239、算术逻辑单元(ALU)240和有时称为高速缓冲存储器的本地或内部存储器248。高速缓冲存储器248在寄存器区段中通常包括多个存储寄存器244-246。一个或多个内部总线241从功能上使这些功能模块相互连接。处理器205通常还具有用于使用接线218经由系统总线204与外部装置进行通信的一个或多个接口242。存储器234使用接线219连接至总线204。

[0043] 应用程序233包括可以包含条件分支指令和循环指令的指令序列231。程序233还可以包括执行程序233时所使用的数据232。将指令231和数据232分别存储在存储器位置228、229、230和235、236、237中。根据指令231和存储器位置228-230的相对大小,如存储器位置230中示出的指令所描述的,可以将特定指令存储在单个存储器位置中。可选地,如存储器位置228和229中示出的指令段所描述的,可以将指令分割成各自被存储在单独的存储器位置的多个部分。

[0044] 通常,向处理器205赋予一组指令,其中在该处理器205内执行该组指令。处理器205等待随后输入,其中处理器205通过执行另一组指令来对该随后输入作出反应。可以从多个源中的一个或多个源提供各输入,其中该输入包括输入装置202、203中的一个或多个所生成的数据、从外部源经由网络220、202其中之一所接收到的数据、从存储装置206、209其中之一所检索到的数据或者从插入相应的读取器212内的存储介质225所检索到的数据(所有这些均在图2A中示出)。执行一组指令在一些情况下可能会导致输出数据。执行还可能涉及将数据或变量存储至存储器234。

[0045] 视频编码器114、视频解码器134和所述方法可以使用存储器234内的相应存储器位置255、256、257中所存储的输入变量254。视频编码器114、视频解码器134和所述方法产生存储器234内的相应存储器位置262、263、264中所存储的输出变量261。可以将中间变量258存储在存储器位置259、260、266和267中。

[0046] 参考图2B的处理器205,寄存器244、245、246、算术逻辑单元(ALU)240和控制单元239一起工作以进行微操作序列,其中这些微操作序列是针对构成程序233的指令集中的各指令进行“提取、解码和执行”周期所需的。各提取、解码和执行周期包括:

[0047] (a)提取操作,用于从存储器位置228、229、230提取或读取指令231;

[0048] (b)解码操作,其中在该解码操作中,控制单元239判断提取了哪个指令;以及

[0049] (c)执行操作,其中在该执行操作中,控制单元239和/或ALU 240执行该指令。

[0050] 之后,可以执行针对下一指令的进一步提取、解码和执行周期。同样,可以进行存储周期,其中通过该存储周期,控制单元239将值存储至或写入存储器位置232。

[0051] 要说明的图12和图13的方法中的各步骤或子处理与程序233的一个或多个区段相关联,并且通常通过处理器205中的寄存器部244、245、247、ALU240和控制单元239一起工作以针对程序233的所述分段的指令集中的各指令进行提取、解码和执行周期,来进行该步骤或子处理。

[0052] 图3是示出视频编码器114的功能模块的示意框图。图4是示出视频解码器134的功能模块的示意框图。通常,数据以样本或系数的组(诸如块向固定大小的子块的分割等)或者作为阵列在视频编码器114和视频解码器134内的功能模块之间传递。如图2A和2B所示,可以使用通用计算机系统200来实现视频编码器114和视频解码器134,其中可以利用计算机系统200内的专用硬件、利用计算机系统200内可执行的软件(诸如驻留在硬盘驱动器205上并且由处理器205控制其执行的软件应用程序233的一个或多个软件代码模块等),来实现各种功能模块。可替代地,可以利用在计算机系统200内可执行的专用硬件和软件的组合来实现视频编码器114和视频解码器134。可以可替代地在诸如进行所述方法的功能或子功能的一个或多个集成电路等的专用硬件中实现视频编码器114、视频解码器134和所述方法。这种专用硬件可以包括图形处理单元(GPU)、数字信号处理器(DSP)、专用标准产品(ASSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或者一个或多个微处理器和关联存储器。特别地,视频编码器114包括模块322-386,并且视频解码器134包括模块420-496,其中这些模块各自可被实现为软件应用程序233的一个或多个软件代码模块。

[0053] 尽管图3的视频编码器114是通用视频编码(VVC)视频编码流水线的示例,但也可以使用其它视频编解码器来进行这里所述的处理阶段。视频编码器114接收诸如一系列帧(各帧包括一个或多个颜色通道)等的所捕获帧数据113。块分区器310首先将帧数据113分割成通常称为“编码树单位”(CTU)的区域,这些区域的形状通常为正方形,并且被配置成使得使用CTU的特定大小。例如,编码树单位的大小可以是 64×64 、 128×128 或 256×256 个亮度样本。块分区器310进一步将各CTU分割成一个或多个编码单位(CU),其中CU具有可以包括正方形和非正方形长宽比这两者的各种大小。因而,从块分区器310输出当前块312(“编码单位”(CU)),从而根据对CTU的一个或多个块的迭代而前进。然而,CU的概念不限于从块分区器310得到的块分区。视频解码器134也可被认为产生CU,这些CU由于使用有损压缩技术因而通常是来自块分区器310的块的近似。此外,视频编码器114产生具有与在视频解码器134中看到的相同的近似的CU,从而能够准确地了解可用于视频解码器134中的块预测方法的样本数据。以下参考图5和6来进一步说明用于将CTU分区为CU的选项。

[0054] 从帧数据113的第一次分割得到的编码树单位(CTU)可以是按光栅扫描顺序扫描的,并被分组成一个或多个“条带(slice)”。由于帧数据113通常包括多个颜色通道,CTU和CU与来自同从块分区器310的操作定义的块区域重叠的所有颜色通道的样本相关联。CU可被认为包括一个或多个编码块(CB),其中各CB占据与CU相同的块区域,但与帧数据113的颜色通道中的每一个颜色通道相关联。由于色度通道与亮度通道的采样速率潜在地不同,因此色度通道的CB的尺寸可能不同于亮度通道的CB的尺寸。当使用4:2:0色度格式时,CU的色度通道的CB的尺寸是CU的亮度通道的CB的宽度和高度的一半。

[0055] 在对从块分区器310得到的所有CU进行迭代时,视频编码器114针对各块(例如,块312)产生由箭头320指示的“预测单位”(PU)。PU 320是关联的CU312的内容的预测。减法器模块322产生PU 320和CU 312之间的表示为324的差(或“残差”,其是指差在空间域中)。差324是PU 320和CU 312中的相应样本之间的块大小差。差324被变换、量化和表示为由箭头336指示的变换单位(TU)。PU 320通常被选择为得到多个可能的候选PU其中之一的“最佳”。候选PU是从视频编码器114可用的预测模式其中之一得到的PU。各候选PU得到相应的变换单位。变换单位336是差324的量化和变换表示。当与解码器114中的预测PU组合时,变换单

位336以位流中的附加信令为代价来减少解码CU和原始块312之间的差。

[0056] 因而,各候选PU均具有关联的编码成本(速率)和关联的差(或“失真”)。编码速率(成本)通常是以位为单位测量的。块的编码失真通常被估计为样本值的差,诸如绝对差和(SAD)或平方差和(SSD)等。模式选择器386使用差324来确定从各候选PU得到的估计,以确定帧内预测模式(由箭头388表示)。可以以与残差的熵编码相比明显更低的成本来进行与各候选预测模式相关联的编码成本和相应的残差编码的估计。因此,可以评价多个候选模式,以确定率失真意义上的最佳模式。

[0057] 确定最佳模式通常是使用拉格朗日优化的变形来实现的。选择帧内预测模式388通常涉及确定从应用特定帧内预测模式得到的残差数据的编码成本。可以通过使用“变换差和”来近似编码成本,由此使用诸如Hadamard变换等的相对简单的变换来获得估计变换残差成本。对于使用相对简单的变换的实现,只要从简化估计方法得到的成本与否则将从完整评价确定的实际成本单调相关,就可以使用该简化估计方法来作出相同的决定(即,帧内预测模式),其中视频编码器114的复杂度降低。为了允许可能从可供残差数据的编码用的更多模式决策得到的、估计成本和实际成本之间的关系的可能非单调性,可以使用该简化估计方法来生成最佳候选的列表。最佳候选的列表可以具有任意数量。可以使用最佳候选来进行更完整的搜索,以建立用于对各个候选的残差数据进行编码的最佳模式选择,从而允许最终选择帧内预测模式以及其它模式决策。

[0058] 其它模式决策包括跳过主要变换和次要变换的能力(被称为“变换跳过”)。跳过变换适合于缺乏足够的相关性来经由作为变换基础函数的表达式降低编码成本的残差数据。诸如相对简单的计算机生成图形等的某些类型的内容可以表现出类似的行为。

[0059] 与模式选择器模块386相关联的另一模式决策是选择显式多变换索引(由箭头390表示)。显式多变换索引390具有四个可能值。这四个可能值是(在图10A的列1046和1048中示出的)两个索引的组合,其中各索引在0到1的范围内。第一个索引与水平变换索引H有关,并且第二个索引与垂直变换索引V有关。水平变换索引H也称为水平变换选择,并且垂直变换索引V也称为垂直变换选择。参考图10来说明显式多变换索引390的选择和编码。

[0060] 可以采用拉格朗日或类似的优化处理来进行(利用块分区器310的)CTU向CU的最佳分区的选择以及从多个可能性中的最佳预测模式的选择这两者。通过将候选模式的拉格朗日优化处理应用在模式选择器模块386中,选择成本测量最低的帧内预测模式作为最佳模式。最佳模式是所选择的帧内预测模式388,并且也由熵编码器338编码在位流115中。通过模式选择器模块386的操作对帧内预测模式388的选择扩展到块分区器310的操作。例如,帧内预测模式388的选择的候选可以包括可应用于给定块的模式和可应用于整体与给定块同位置的多个更小块的模式。在这样的情况下,候选的选择的处理隐含地也是用于确定CTU向CU的最佳层级分解的处理。

[0061] 熵编码器338支持句法元素的可变长度编码和句法元素的算术编码这两者。使用上下文自适应二进制算术编码处理来支持算术编码。算术编码的句法元素由一个或多个“bin(二进制文件)”的序列组成。与位一样,bin的值为“0”或“1”。然而,bin未作为离散位编码在位流115中。bin具有关联可能值和关联概率(被称为“上下文”)。当要编码的实际bin与可能值匹配时,对“最大概率符号”(MPS)进行编码。对最大概率符号进行编码在消耗位方面相对便宜。当要编码的实际bin与可能值不匹配时,对“最小概率符号”(LPS)进行编码。对最

小概率符号进行编码在消耗位方面具有相对高的成本。bin编码技术使得能够使“0”vs“1”的概率歪斜的bin进行高效编码。对于具有两个可能值的句法元素(即,“flag(标志)”),单个bin就足够了。对于具有许多可能值的句法元素,需要bin的序列。然后,可以基于该序列中的较前bin的值来确定该序列中的较后bin的存在。另外,各bin可以与多于一个的上下文相关联,其中根据句法元素中的较前bin和相邻句法元素的bin值(即,来自相邻块的bin值)等来选择特定上下文。每次对bin进行编码时,更新上下文以适应于新的bin值。如此,二进制算术编码方案被认为是自适应的。

[0062] 编码器114也支持缺少上下文的bin(“旁路bin”)。假定“0”和“1”之间的等概率分布来对旁路bin进行编码。不存在上下文节省了内存并降低了复杂度,因而使用特定bin的值的分布未歪斜的旁路bin。采用上下文和自适应的熵编码器的一个示例在本领域中被称为CABAC(上下文自适应二进制算术编码器),并且在视频编码中采用了该编码器的许多变体。

[0063] 熵编码器338使用上下文编码的和(可选地)旁路编码的bin的组合来对帧内预测模式388进行编码。通常,在视频编码器114中生成“最大概率模式”的列表。最大概率模式的列表通常是诸如三个或六个模式等的固定长度,并且可以包括在早期块中遇到的模式。上下文编码bin对指示帧内预测模式是否是最大概率模式其中之一的flag进行编码。如果帧内预测模式388是最大概率模式其中之一,则例如使用截断一元bin串来对指示哪个最大概率模式与帧内预测模式388相对应的进一步信令进行编码。否则,使用诸如固定长度代码等的替代句法来将帧内预测模式388编码为“剩余模式”,以表示除存在于最大概率模式列表中的帧内预测模式以外的帧内预测模式。

[0064] 如参考图10所述,熵编码器338还针对特定编码单位或变换块对显式多变换索引390进行编码。特别地,显式多变换索引390包括垂直变换索引V和水平变换索引H。垂直变换索引V和水平变换索引H这两者都是由模式选择器386选择的(如上所述)。基于垂直变换索引V和水平变换索引H的组合(参见图10A),熵编码器338将在(在图10A的列1042中示出的)相应bin串编码到位流115中。

[0065] 多路复用器模块384根据选自所测试的候选预测模式中的所确定的最佳帧内预测模式388来输出PU 320。候选预测模式无需包括视频编码器114所支持的每个可想到的预测模式。预测模式大致分为两个类别。第一个类别是“帧内预测(intra-frame prediction)”(或“帧内预测(intra prediction)”)。在帧内预测中,使用从当前帧中提取的其它样本来产生块的预测。第二个类别是“帧间预测(inter-frame prediction)”(或“帧间预测(inter prediction)”)。在帧内预测中,使用来自按位流中的编码帧的顺序(其可能不同于捕获或显示时的帧的顺序)在当前帧之前的帧的样本来产生块的预测。在各类别(即,帧内和帧间预测)内,可以应用不同的技术以生成PU。例如,帧内预测可以使用来自先前重建样本的相邻行和列的值组合方向来根据规定的滤波处理生成PU。可替代地,可以使用少量参数来描述PU。帧间预测方法在运动参数的数量及其精度上可能变化。运动参数通常包括参考帧偏移+一个或两个参考帧的平移,但可以包括更多帧、特殊帧、或者诸如缩放和旋转等的复杂仿射参数。另外,可以应用预先确定的运动细化处理以基于参考样本块生成密集运动估计。

[0066] 在确定并选择了最佳PU 320、并在减法器322处从原始样本块中减去了PU 320的情况下,获得编码成本最低的残差324并对该残差324进行进行有损压缩。有损压缩处理包

括变换、量化和熵编码的步骤。变换模块326对差324应用第一变换,从而将差324转换到频率域并且产生由箭头328表示的中间变换系数。第一变换通常是可分离的,从而对各块的一组行然后对一组列进行变换。通过在块的各行和列中应用重复的一维变换来进行各组行和列的变换。变换跨越整个块。

[0067] 特别地,使用帧内预测模式388和显式多变换索引390来选择整体构成由变换模块326进行的主要变换的水平一维变换和垂直一维变换。水平一维变换和垂直一维变换各自是可分离变换。水平一维变换和垂直一维变换是在整个变换块上进行的,从而基本上去相关残差样本。

[0068] 变换模块326可用的变换类型包括DST-VII、DST-VIII、DCT-VIII、DST-I和DCT-V。如参考图8B所述,根据该组变换,定义了三个“变换集”842。对于水平或垂直方向中的各方向,如参考图8C所述,根据帧内预测模式388来选择变换集842。如参考图11所述,从各变换组842中,选择并应用一个变换类型。中间变换系数328被传递到次要变换模块330。

[0069] 次要变换模块330对中间变换系数328的子集(诸如占据整个块的左上方的 4×4 或 8×8 区域的中间变换系数等)进行操作。中间变换系数328中的其它变换系数不变地通过模块330。次要变换模块330对中间变换系数328的子集应用各种变换其中之一,以产生由箭头332表示的变换系数。次要变换模块330应用以类似于逆次要变换模块344的方式所选择的正向次要变换。可供次要变换模块330用的变换通常是不可分离的,因此不能如变换模块326的情况那样在两个阶段(即,行和列)中进行。由于次要变换的复杂度较高并且主要变换的去相关较早,因此仅对变换块的左上方区域进行次要变换就足够了。次要变换模块330所进行的变换可以是至少部分取决于预测模式所选择的。另外,视频编码器114可以将模块330处应用的变换的选择视为基于率/失真成本评价的选择的进一步候选的测试。

[0070] 变换系数332被传递到量化器模块334。在模块334处,进行根据“量化参数”的量化以产生由箭头336表示的残差系数。量化参数对于给定变换块是恒定的,因而得到针对变换块的残差系数的产生的均匀缩放。通过应用“量化矩阵”也可以实现非均匀缩放,由此对各残差系数应用的缩放因子是从量化参数和大小通常等于变换块的大小的缩放矩阵中的相应条目的组合导出的。因而,要表示的昂贵的量化矩阵(如果有的话)仅不频繁地编码在位流115中。量化矩阵的编码要求将缩放因子的二维矩阵转换成要熵编码到位流115中的缩放因子的列表。现有的Z顺序扫描可重复用于该目的,从而避免与支持不频繁地进行的对量化矩阵进行编码的操作所用的附加扫描模式相关联的开销。残差系数336被供给至熵编码器338以编码在位流115中。通常,根据扫描模式对变换块的残差系数进行扫描以生成值的有序列表。扫描模式通常将变换块扫描为 4×4 个“子块”的序列,从而按 4×4 的残差系数集合的粒度提供常规扫描操作,其中子块的配置取决于变换块的大小。另外,预测模式和相应块分区也编码在位流115中。如上所述,视频编码器114需要访问与在视频解码器134中看到的帧表示相对应的帧表示。因而,残差系数336也由去量化器模块340进行逆量化以产生由箭头342表示的逆变换系数。逆变换系数342通过逆次要变换模块344。逆次要变换模块344应用所选择的次要变换以产生如由箭头346表示的中间逆变换系数。中间逆变换系数346被供给至逆变换模块348以产生变换单位的由箭头350表示的残差样本。求和模块352将残差样本350和PU 320相加以产生CU的重建样本(由箭头354指示)。重建样本354被传递到参考样本高速缓冲存储器356和环内滤波器模块368。通常使用ASIC上的静态RAM实现(因此避免了

昂贵的片外存储器访问)的参考样本高速缓冲存储器356提供了满足用于为帧中的后续CU生成帧内预测块的依赖关系所需的最小样本存储。最小依赖关系通常包括沿着一行CTU的底部的样本的“线缓冲器”,以供下一行CTU以及范围由CTU的高度设置的列缓冲使用。参考样本高速缓冲存储器356将参考样本(由箭头358表示)供给至参考样本滤波器360。样本滤波器360应用平滑操作以产生滤波参考样本(由箭头362指示)。滤波参考样本362由帧内预测模块364使用以产生由箭头366表示的样本的帧内预测块。对于各候选帧内预测模式,帧内预测模块364产生样本块即366。

[0071] 环内滤波器模块368对重建样本354应用数个滤波阶段。滤波阶段包括“去块滤波器”(DBF),该DBF应用与CU边界对齐的平滑化,以减少由不连续而产生的伪影。环内滤波器模块368中存在的另一滤波阶段是“自适应环路滤波器”(ALF),该ALF应用基于Wiener的自适应滤波器以进一步降低失真。环内滤波器模块368中的另一可用滤波阶段是“样本自适应偏移”(SAO)滤波器。SAO滤波器通过首先将重建样本分类为一个或多个类别、并且根据所分配的类别在样本级别应用偏移来工作。从环内滤波器模块368输出滤波样本370。滤波样本370被存储在帧缓冲器372中。帧缓冲器372通常具有存储数个(例如,多达16个)图片的容量,因而存储在存储器206中。如此,对帧缓冲器372的访问在内存带宽方面是昂贵的。帧缓冲器372将参考帧(由箭头374表示)提供至运动估计模块376和运动补偿模块380。

[0072] 运动估计模块376估计多个“运动矢量”(表示为378),其各自是相对于当前CU的位置的笛卡尔空间偏移,从而参考帧缓冲器372中的参考帧其中之一中的块。针对各运动矢量产生参考样本的滤波块(表示为382)。滤波参考样本382形成可供模式选择器386的潜在选择用的进一步候选模式。此外,对于给定CU,PU 320可以使用一个参考块(“单预测”)形成,或者可以使用两个参考块(“双预测”)形成。对于所选择的运动矢量,运动补偿模块380根据支持运动矢量中的子像素精度的滤波处理来产生PU 320。如此,运动估计模块376(其对许多候选运动矢量进行操作)与运动补偿模块380(其仅对所选择的候选进行操作)相比可以想象地进行简化滤波处理,以实现降低的计算复杂度。

[0073] 尽管参考通用视频编码(VVC)说明了图3的视频编码器114,但其它视频编码标准或实现也可以采用模块310-386的处理阶段。帧数据113(和位流115)也可以从存储器206、硬盘驱动器210、CD-ROM、蓝光盘(Blue-ray disk™)或其它计算机可读存储介质中读取(或者被写入存储器206、硬盘驱动器210、CD-ROM、蓝光盘或其它计算机可读存储介质)。另外,帧数据113(和位流115)可以从外部源(诸如连接至通信网络220的服务器或者射频接收器等)接收(或者被发送至该外部源)。

[0074] 在图4中示出视频解码器134。尽管图4的视频解码器134是通用视频编码(VVC)视频编位流水线的示例,但其它视频编解码器也可用于进行本文所述的处理阶段。如图4所示,位流133被输入到视频解码器134。位流133可以从存储器206、硬盘驱动器210、CD-ROM、蓝光盘或其它非暂时性计算机可读存储介质中读取。可替代地,位流133可以从外部源(诸如连接至通信网络220的服务器或者射频接收器等)接收。位流133包含表示要解码的所捕获帧数据的编码句法元素。

[0075] 位流133被输入到熵解码器模块420。熵解码器模块420从位流133中提取句法元素并将句法元素的值传递到视频解码器134中的其它模块。熵解码器模块420应用CABAC算法以从位流133解码句法元素。解码得到的句法元素用于重建视频解码器134内的参数。参数

包括残差系数(由箭头424表示)以及诸如帧内预测模式458和显式多变换索引454等的模式选择信息。模式选择信息还包括诸如运动矢量等的信息、以及各CTU向一个或多个CU的分区。参数用于通常与来自先前解码的CU的样本数据组合生成PU。

[0076] 位流133还包含来自bin串列1042(参见图10A)的bin串。熵解码器模块420提取bin串并产生(来自垂直变换集索引列1046的)垂直变换集索引和(来自水平变换集索引列1048的)水平变换集索引。

[0077] 残差系数424被输入到去量化器模块428。去量化器模块428对残差系数424进行逆缩放,以根据量化参数生成重建中间变换系数(由箭头432表示)。如果在位流133中指示使用非均匀逆量化矩阵,则视频解码器134从位流133读取量化矩阵作为缩放因子序列,并且根据用于对残差系数进行编码的Z顺序扫描将缩放因子排列成矩阵。然后,逆缩放将量化矩阵与量化参数组合使用以创建重建中间变换系数。将Z顺序扫描用于残差系数以同样扫描量化矩阵缩放因子,这避免了附加扫描模式的存在、以及针对不频繁地进行的扫描的关联内存和复杂度负担。重建中间变换系数432被传递到逆次要变换模块436。逆次要模块436进行“次要逆变换”以产生由箭头440表示的重建变换系数。次要变换是根据所确定的变换块大小来进行的。如参考图9所述,重建变换系数440被传递到逆变换模块444。模块444将系数从频率域变换回到空间域。变换块有效地基于有效残差系数和非有效残差系数值。模块444的操作的结果是由箭头448表示的残差样本的块。残差样本448在大小上等于相应的CU。残差样本448被供给至求和模块450。在求和模块450处,将残差样本448加到解码PU 452,以产生由箭头456表示的重建样本的块。重建样本456被供给至重建样本高速缓冲存储器460和环内滤波模块488。环内滤波模块488产生表示为492的帧样本的重建块。帧样本492被写入帧缓冲器496。

[0078] 重建样本高速缓冲存储器460以类似于视频编码器114的重建样本高速缓冲存储器356的方式操作。重建样本高速缓冲存储器460在无存储器206的情况下(例如,通过作为代替使用通常是片上存储器的数据232)为对后续CU进行帧内预测所需的重建样本提供存储。由箭头464表示的参考样本是从重建样本高速缓冲存储器460获得的,并被供给至参考样本滤波器468以产生由箭头472表示的滤波参考样本。滤波参考样本472被供给至帧内预测模块476。模块476根据在位流133中表示的并由熵解码器420解码的帧内预测模式参数458,产生由箭头480表示的帧内预测样本的块。

[0079] 当针对当前CU在位流133中指示帧内预测时,帧内预测样本480经由多路复用器模块484形成解码PU 452。

[0080] 当针对当前CU在位流133中指示帧间预测时,运动补偿模块434使用运动矢量和参考帧索引从帧缓冲器496中选择和过滤样本块,来产生帧间预测样本438的块。样本块498是从帧缓冲器496中所存储的先前解码帧获得的。为了进行双预测,产生两个样本块并将这两个样本块混合在一起以产生解码PU 452的样本。帧缓冲器496由来自环内滤波模块488的滤波块数据492填充。与视频编码器114的环内滤波模块368一样,环内滤波模块488应用任何、至少或全部的DBF、ALF和SAO滤波操作。环内滤波模块368从重建样本456产生滤波块数据492。

[0081] 图5是示出通用视频编码的树结构中的块向一个或多个块的可用分割或拆分的集合500的示意框图。如参考图3所述,集合500中示出的分割可供编码器114的块分区器310利

用以根据拉格朗日优化将各CTU分割成一个或多个CU。

[0082] 尽管集合500仅示出将正方形块分割成其它可能非正方形块,但应当理解,图500正示出潜在分割、而并未将包含块约束为正方形。如果包含块为非正方形,则根据包含块的长宽比来对从分割得到的块的尺寸进行缩放。利用块分区器310的CTU向一个或多个CU的特定子分割被称为CTU的“编码树”。在本发明的上下文中,叶节点是子分割的处理终止的节点。当与叶节点相对应的区域等于最小编码单位大小时,子分割的处理必须终止。得到最小大小的编码单位的叶节点存在于编码树的分解的最深级别处。子分割的处理也可以在分解的最深级别之前终止,这样得到大于最小编码单位大小的编码单位。

[0083] 在不存在进一步子分割的情况下,在编码树的叶节点处存在CU。例如,叶节点510包含一个CU。在编码树的非叶节点处,存在向两个或更多个其它节点的拆分,其中各节点可以包含一个CU、或者包含向更小区域的进一步拆分。

[0084] 如图5所示,四叉树拆分512将包含区域分割成四个相等大小的区域。与HEVC相比,通用视频编码通过添加水平二元拆分514和垂直二元拆分516实现了附加的灵活性。拆分514和516各自将包含区域分割成两个相等大小的区域。分割沿着包含块内的水平边界(514)或垂直边界(516)。

[0085] 在通用视频编码中通过添加三元水平拆分518和三元垂直拆分520实现了进一步的灵活性。三元拆分518和520将块分割成沿着包含区域宽度或高度的1/4和3/4在水平方向(518)或垂直方向(520)上形成界限的三个区域。四叉树、二叉树和三叉树的组合被称为“QTBTTT”或可替代地被称为多叉树(MT)。

[0086] 与仅支持四叉树、因而仅支持正方形块的HEVC相比,QTBTTT特别是考虑到二叉树和/或三叉树拆分的可能递归应用而得到更多可能的CU大小。可以通过约束拆分选项以消除将得到小于四个样本的块宽度或高度或者将得到不是四个样本的倍数的拆分来降低异常(例如,非正方形)块大小的可能性。通常,约束将在考虑亮度样本时适用。然而,约束也可单独应用于色度通道的块,从而可能针对亮度vs色度(例如,在帧数据采用4:2:0色度格式时)得到不同的最小块大小。

[0087] 图6是示出在通用视频编码中使用的QTBTTT(或“编码树”)结构的数据流600的示意图。将QTBTTT结构用于各CTU以定义CTU向一个或多个CU的分割。各CTU的QTBTTT结构由视频编码器114中的块分区器310确定,并被编码到流115中或者由视频解码器134中的熵解码器420从流133解码。根据图5所示的分割,数据流600进一步表现可供块分区器310将CTU分割成一个或多个CU用的许可组合的特征。

[0088] 从层级结构的顶层开始、即在CTU处,首先进行零个或更多个四叉树分割。具体地,由块分区器310作出四叉树(QT)拆分决策610。610处的决策返回“1”符号,这表明决定根据四叉树拆分512将当前节点拆分成四个子节点。结果是诸如在620处等生成四个新节点,并且针对各新节点,递归回到QT拆分决策610。各新节点均是按光栅(或Z扫描)顺序考虑的。可替代地,如果QT拆分决策610指示不进行进一步拆分(返回“0”符号),则四叉树分区停止,并且随后考虑多树(MT)拆分。

[0089] 首先,由块分区器310作出MT拆分决策612。在612处,指示进行MT拆分的决策。在决策612处返回“0”符号,这表明将不进行节点向子节点的进一步拆分。如果将不进行节点的进一步拆分,则节点是编码树的叶节点并且对应于编码单位(CU)。在622处输出叶节点。可

替代地,如果MT拆分612指示决定进行MT拆分(返回“1”符号),则块分区器310进入方向决策614。

[0090] 方向决策614将MT拆分的方向指示为水平(“H”或“0”)或垂直(“V”或“1”)。如果决策614返回指示水平方向的“0”,则块分区器310进入决策616。如果决策614返回指示垂直方向的“1”,则块分区器310进入决策618。

[0091] 在决策616和618各自中,在BT/TT拆分时将MT拆分的分区数量指示为两个(二元拆分或“BT”节点)或三个(三元拆分或“TT”)。也就是说,当从614指示的方向为水平时,由块分区器310作出BT/TT拆分决策616,并且当从614指示的方向为垂直时,由块分区器310作出BT/TT拆分决策618。

[0092] BT/TT拆分决策616指示水平拆分是通过返回“0”所指示的二元拆分514、还是通过返回“1”所指示的三元拆分518。当BT/TT拆分决策616指示二元拆分时,在生成HBT CTU节点的步骤625处,块分区器310根据二元水平拆分514生成两个节点。当BT/TT拆分616指示三元拆分时,在生成HTT CTU节点的步骤626处,块分区器310根据三元水平拆分518生成三个节点。

[0093] BT/TT拆分决策618指示垂直拆分是通过返回“0”所指示的二元拆分516、还是通过返回“1”所指示的三元拆分520。当BT/TT拆分618指示二元拆分时,在生成VBT CTU节点的步骤627处,块分区器310根据垂直二元拆分516生成两个节点。当BT/TT拆分618指示三元拆分时,在生成VTT CTU节点的步骤628处,块分区器310根据垂直三元拆分520生成三个节点。对于从步骤625-628得到的各节点,根据方向614按从左到右或从上到下的顺序来应用数据流600返回到MT拆分决策612的递归。结果,可以应用二叉树和三叉树拆分以生成具有各种大小的CU。

[0094] 图7A和7B提供CTU 710向多个编码单位(CU)的示例分割700。在图7A中示出示例CU 712。图7A示出CTU 710中的CU的空间排列。示例分割700在图7B中也被示出为编码树720。

[0095] 在图7A的CTU 710中的各非叶节点(例如,节点714、716和718)处,按“Z顺序”扫描或遍历所包含的节点(其可以是进一步分割的或者可以是CU),以创建在编码树720中表示为列的节点列表。对于四叉树拆分,Z顺序扫描得到从左上方到右方之后是从左下方到右方的顺序。对于水平和垂直拆分,Z顺序扫描(遍历)分别简化为从上方到下方和从左方到右方的扫描。图7B的编码树720根据所应用的扫描顺序列出所有的节点和CU。各拆分在树的下一级别生成2、3或4个新节点的列表,直到到达叶节点(CU)为止。

[0096] 在如参考图3所述利用块分区器310将图像分解为CTU并进一步分解为CU、并且使用CU生成各残差块(324)的情况下,利用编码器114对残差块进行正向变换和量化。随后扫描如此得到的变换块(TB) 336以形成残差系数的顺序列表,作为熵编码模块338的操作的一部分。在视频解码器134中进行等效处理以从位流133获得变换块。

[0097] 图8A示出使用帧内预测模式388和458可以指示的变换块的帧内预测模式的集合800。在图8A中,定义了67个帧内预测模式。模式0是“平面”帧内预测模式,模式1是“DC”帧内预测模式,并且模式2-66是“角度”帧内预测模式。平面帧内预测模式(模式0)根据平面用样本填充预测块(即,在水平方向和垂直方向上具有偏移和梯度)。平面参数是从相邻的参考样本(如果有的话)获得的。同样,DC帧内预测模式(模式1)也使用相邻的参考样本(如果有的话)填充具有偏移的预测块。

[0098] 角度帧内预测模式(模式2-66)通过产生与65个方向或“角度”其中之一对齐的纹理来填充块。为清楚起见,在图8A中示出65个角度的子集(即,模式2、18、34、50和66)。对于各模式,使用相邻的参考样本来产生针对角度帧内预测模式在由箭头指示的方向上填充预测块的纹理。在图8A中未明确示出的附加角度在中间位置(即,模式3-17、19-33、35-49和51-65)处。从图8A沿着角度模式34并且在预测块的从左上方到右下方的对角方向上可看出第一对称性。从第一对称性来看,通过沿着该对角轴的转置,模式2到34被示出为对应于模式66向下直到模式34。

[0099] 图8B示出根据变换索引844从各个变换集842到特定变换类型的映射840。三个变换集842由0、1和2的相应变换集索引来定义和标识。各变换集842包括可由变换索引844选择的两个变换类型。结果,映射840允许基于变换集842的变换集索引以及变换索引844来选择给定的变换类型。所选择的变换是一维变换。对水平和垂直方向进行变换类型的单独选择。变换的大小(点数)对应于变换块的尺寸。也就是说,当选择垂直变换时,变换的大小对应于变换块的高度,而当选择水平变换时,变换的大小对应于变换块的宽度。

[0100] 图8C是示出针对变换块基于该变换块是在水平方向(“H”)还是垂直方向(“V”)上的从帧内预测模式到变换集842的变换集索引的映射880的表。对于各帧内预测模式,针对水平方向H选择与三个变换集842相关联的变换集索引其中之一,并且针对垂直方向V选择与三个变换集842相关联的变换集索引其中之一。

[0101] 例如,对于垂直方向上的变换块,将帧内预测模式14映射到变换集索引0,而对于水平方向上的变换块,将帧内预测模式14映射到变换集索引2。如图8B中的映射840所示,将变换集索引0映射到根据变换索引844可选择的DST-VII或DCT-VIII。如图8B的映射840所示,将变换集索引2映射到根据变换索引844可选择的DST-VII或DCT-V。因此,对于垂直方向上的变换块,帧内预测模式14根据变换索引844使用DST-VII或DCT-VIII变换。因此,对于水平方向上的变换块,帧内预测模式14根据变换索引844使用DST-VII或DCT-V变换。变换索引844是基于显式多变换索引390或454。如上所述,显式多变换索引390是由模式选择器306选择的,并且显式多变换索引454是由熵编码器420从位流133导出的。

[0102] 图9是示出视频编码器114的逆变换模块348或视频解码器134的逆变换模块444的示意框图。两个模块348和444的操作是相同的,从而反映视频编码器114对在视频解码器134中进行的残差系数的逆量化和逆变换进行建模的需求。

[0103] 逆变换模块348和444各自包括变换集表模块910、变换索引导出器模块920、水平变换选择器模块940、垂直变换选择器模块945、水平变换模块960和垂直变换模块980。

[0104] 变换集表模块910接收帧内预测模式388或458。当被实现为逆变换模块348时,变换集表模块910从模式选择器386接收帧内预测模式388(参见图3)。当被实现为逆变换模块444时,变换集表模块910从熵编码器420接收帧内预测模式458(参见图4)。

[0105] 变换集表模块910从帧内预测模式388或458确定水平变换集索引(表示为箭头915)和垂直变换集索引(表示为箭头917)。水平变换集索引915和垂直变换集索引917是根据映射880(参见图8C)确定的。

[0106] 变换索引导出器920接收显式多变换索引390或454。当被实现为逆变换模块348时,变换索引导出器920从模式选择器386接收显式多变换索引390(参见图3)。当被实现为逆变换模块444时,变换索引导出器920从熵编码器420接收显式多变换索引454(参见图4)。

[0107] 变换索引导出器920从显式多变换索引390或454中提取水平变换索引920和垂直变换索引925。水平变换索引920和垂直变换索引925各自在0到1的范围内。

[0108] 水平变换选择器模块940接收(来自变换集表模块910的)水平变换集索引915和(来自变换索引导出器920的)水平变换索引920。水平变换选择器940根据映射840(参见图8B)从水平变换集索引915和水平变换索引920产生所选择的水平变换(表示为箭头950)。水平变换集索引915用于选择变换集842其中之一,并且水平变换索引920用于选择变换索引844其中之一。

[0109] 垂直变换选择器模块945接收(来自变换集表模块910的)垂直变换集索引917和(来自变换索引导出器920的)垂直变换索引925。垂直变换选择器模块945根据映射840(参见图8B)从垂直变换集索引917和垂直变换索引925产生所选择的垂直变换955。垂直变换集索引917用于选择变换集842其中之一,并且垂直变换索引925用于选择变换索引844其中之一。

[0110] 水平变换模块960将所选择的水平变换950应用于中间逆变换系数346或重建变换系数440。中间逆变换系数346来自逆次要变换模块344(参见图3)。重建变换系数440来自逆次要变换模块436(参见图4)。

[0111] 变换系数346或440是根据所选择的水平变换950进行变换的。也就是说,将具有一组一维变换的所选择的水平变换950应用于变换系数346或440的阵列的各行。各一维变换跨越变换块的宽度。水平变换模块960输出水平变换的变换系数970作为具有与当前变换块相等的大小的阵列。变换系数970被供给至垂直变换模块980。为便于描述,变换系数970参考水平变换的变换系数346和440这两者。

[0112] 垂直变换模块980将具有一组一维变换的所选择的垂直变换955应用于水平变换的变换系数970的各列。垂直变换模块980使用垂直选择变换955来从相应的水平变换的变换系数346或440产生残差样本350或448。

[0113] 图10A示出表1000,该表1000示出截断一元句法元素与变换选择所用的变换索引之间的关系。变换索引是(在列1048中示出的)水平变换索引H和(在列1046中示出的)垂直变换索引V。表1000包括bin串列1042、二值化内容(binariesation)列1044、垂直变换索引列1046和水平变换索引列1048。

[0114] bin串列1042具有四个可能的bin串,这些bin串来自于最大值为3(即,范围为0到3)的截断一元二值化内容。对于编码器114,在列1042中示出的bin串是熵编码器338编码到位流115中的bin串。基于模式选择器386所选择的水平变换索引H和垂直变换索引V来选择要编码到位流115中的bin串。

[0115] 对于解码器134,在列1042中示出的bin串是由熵解码器420从位流133解码得到的bin串。然后,熵解码器420基于解码得到的bin串输出水平变换索引H和垂直变换索引V。

[0116] 二值化内容列1044包括bin串列1042中的各bin串的组成。在该命名中,“Ctx0”和“Ctx1”是指用于对bin串列1042中的相应bin进行编码的两个不同的CABAC内容。术语“Byp”是指用于对bin串列1042中的相应bin进行编码的旁路编码bin。

[0117] 垂直变换索引列1046包含用于从垂直1D变换所用的变换集中选择变换的索引值(其为0或1)。水平变换索引列1048包含用于从水平1D变换所用的变换集中选择变换的索引值(其为0或1)。在表1000中,将最短的bin串指派至垂直索引和水平索引为“0”和“0”的情

况。最短的bin串对应于利用模式选择器386的最常选择的情况。然后,将次最短的bin串指派至垂直索引和水平索引为“1”和“1”的情况。这对应于利用模式选择器386的次最常选择的情况。然后,针对垂直变换索引和水平变换索引的组别的索引值组合“01”或“10”的情况同样是不可能的,并且被指派列1044中的最长的bin串。此外,使用旁路编码bin来进行这两个情况之间的区别,从而反映利用模式选择器386的这两个选择之间的大致相等的发生概率。所描述的统计性质被发现在变换大小方面大体上不变,因而根据这些上下文在截断一元bin串中的位置所选择的这些上下文的存在足以捕获bin值的不对称概率分布。

[0118] 图10B是示出截断一元句法元素与水平和垂直变换索引所用的变换索引之间的替代关系的表1050,其中截断一元二值化内容的第三bin使用上下文编码而不是旁路编码。

[0119] 图11是用于根据显式变换选择进行主要变换以将残差系数的变换块编码到位流115中的方法1100的流程图。该方法1100可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,该方法1100可以在处理器205的执行下由视频编码器114进行。如此,方法1100可以存储在计算机可读存储介质上和/或存储器206中。方法1100利用处理器205在确定帧内预测模式的步骤1110处开始。

[0120] 方法1100从帧内预测模式的步骤1110开始。在帧内预测模式的步骤1110处,模式选择器386在处理器205的执行下针对与当前编码单位相关联的预测单位选择帧内预测模式388。该选择通常分两遍进行。在第一遍中,测试亮度预测块的所有帧内预测模式。对于各模式,例如使用诸如Hadamard变换等的“绝对变换差和”方法来近似残差成本。从该测试导出“最佳”(最低失真)候选预测模式的列表。然后,对候选预测模式的列表进行残差编码的完整测试。作为完整测试的结果,针对当前编码单位选择一个帧内预测模式388。熵编码器338将帧内预测模式388编码到位流115中。然后,处理器205中的控制从帧内预测模式的步骤1110进入确定变换集的步骤1120。

[0121] 在确定变换集的步骤1120处,逆变换模块348在处理器205的执行下使用所确定的帧内预测模式388来根据映射880(参见图8C)确定水平变换集索引915(如以上关于图9所述)和垂直变换集索引917(如以上关于图9所述)。从水平变换集索引915,根据映射840来针对水平变换确定一组变换。从垂直变换集索引917中,根据映射840来针对垂直变换确定一组变换。

[0122] 如在以上的示例中所述,如果帧内预测模式388是模式14,则垂直变换的可能变换是DST-VII或DCT-VIII,并且水平变换的可能变换是DST-VII或DCT-V。

[0123] 然后,处理器205中的控制从确定变换集的步骤1120进入测试变换的步骤1130。

[0124] 在测试变换的步骤1130处,变换模块326和逆变换模块348在处理器205的执行下,测试来自步骤1120的水平变换所用的变换集和垂直变换所用的变换集中的变换的各组合。在帧内预测模式388是模式14的示例中,针对垂直变换来测试DST-VII和DCT-VIII的变换,并且针对水平变换来测试DST-VII或DCT-V的变换。

[0125] 如上所述,测试由变换模块326和逆变换模块348进行。使用(水平变换所用的)水平变换索引920和(垂直变换所用的)垂直变换索引925来参考各集合中的特定变换。测试垂直变换和水平变换的各组合,以确定用于对残差进行编码的失真。

[0126] 在测试期间,变换模块326和逆变换模块348进行模块的正常操作。因此,对于逆变换模块348所进行的所有逆变换操作,变换模块326进行相应的正向变换操作。测试处理可

以使用近似方法来进行测试,其中与进行完整变换相比,复杂度降低。

[0127] 然后,处理器205中的控制从测试变换集的步骤1130进入选择变换1140。

[0128] 在选择变换的步骤1140处,选择得到对残差进行编码的最低失真的垂直变换和水平变换的组合。在帧内预测模式388是模式14的示例中,提供最低失真的组合可以是垂直变换所用的DCT-VIII和水平变换所用的DST-VII。然后,处理器205中的控制从选择变换的步骤1140进入应用水平变换1150。

[0129] 在应用水平变换的步骤1150处,逆变换模块348进行水平变换。特别地,水平变换模块960进行所选择的水平变换950。在帧内预测模式388是模式14的示例中,将DST-VII用于水平变换。然后,处理器205中的控制从应用水平变换1150进入应用垂直变换1160。

[0130] 在应用垂直变换的步骤1160处,逆变换模块348进行垂直变换。特别地,垂直变换模块980进行所选择的垂直变换955。在帧内预测模式388是模式14的示例中,将DCT-VIII用于垂直变换。作为应用水平变换的步骤1150和应用垂直变换的步骤1160的结果,并且在次要变换的可选应用(即,模块330的操作)之后,产生变换系数332。然后,处理器205中的控制从应用垂直变换的步骤1160进入对残差系数进行量化的步骤1170。

[0131] 在对残差系数进行量化的步骤1170处,量化模块334在处理器205的执行下对变换系数332进行量化,以根据量化参数产生残差系数336。然后,处理器205中的控制从对残差系数进行量化的步骤1170进入对残差系数进行编码的步骤1180。

[0132] 在对残差系数进行编码的步骤1180处,熵编码器338在处理器205的执行下将残差系数336编码到位流115中。通过将系数的二维阵列扫描到列表中来对给定变换块的残差系数336进行编码。扫描通常将变换块分割成各自大小为 4×4 (尽管其它大小也是可以的)的一个或多个子块。用“编码子块标志”表示各子块中的有效系数的存在,并且如果存在至少一个有效系数,则针对子块中的各残差系数存在一个有效性标志。编码子块标志和有效性标记使用上下文编码bin。诸如“大于1个标志”和“大于2个标志”等的大小的附加指示也是使用上下文编码bin来编码的。最后,大小的进一步指示通常是使用利用旁路编码bin的Golomb Rice编码来编码的。在对残差系数336进行编码时的位的成本在很长程度上受其大小的影响。作为选择最佳EMT索引的结果,这些大小略微减小,这导致压缩效率提高。在步骤1190处进行的这些索引的二值化内容进一步提高了压缩效率。然后,处理器205中的控制从对残差系数进行编码的步骤1180进入对EMT bin串进行编码的步骤1190。

[0133] 在对EMT bin串进行编码的步骤1190处,熵编码器338在处理器205的执行下使用截断一元二值化内容(最大值3)、上下文建模、以及如参考图10A所述的bin串向索引值的映射,来将水平变换索引920和垂直变换索引925编码到位流115中。由于使用最短的bin串(“0”)来对最常选择的索引对(在水平方向上为“0”且在垂直方向上为“0”)进行编码,因此编码效率与将一个bin用于各索引的简单情况相比变高。使用次最短的bin串(即,“1 0”)来对次最常选择的索引对(在水平方向上为“1”且在垂直方向上为“1”)进行编码,并且使用最长的bin串(“1 1 0”和“1 1 1”)来对两个近似等同的最不可能的组合(“1”和“0”、“0”和“1”)进行编码。由于不存在可供使用上下文编码的开发用的统计对称性,因此第三bin使用旁路编码。然后,方法1100在步骤1190的结束时终止。

[0134] 图12是用于从位流133解码残差系数的变换块的方法1200的流程图。方法1200的解码通过根据显式变换选择所选择的主要逆变换来进行。方法1200可以由诸如配置的

FPGA、ASIC或ASSP等的设备来体现。另外,方法1200可以在处理器205的执行下由视频解码器134进行。如此,方法1200可以存储在计算机可读存储介质上和/或存储器206中。方法1200利用处理器205在对帧内预测模式进行解码的步骤1210处开始。

[0135] 在对帧内预测模式进行解码的步骤1210处,熵解码器420在处理器205的执行下从位流133解码帧内预测模式458。熵解码器420产生包括被视为最有可能由模式选择器386选择的模式的最高概率模式的列表。对上下文编码bin进行解码,以指示使用最高概率模式中的任一模式或者未在最高概率模式列表中的其它模式(“剩余模式”)。然后,对进一步的bin进行解码,以识别哪个最高概率模式在使用中或者哪个剩余模式在使用中。然后,处理器205中的控制从对帧内预测模式进行解码的步骤1210进入对残差系数进行解码的步骤1220。

[0136] 在对残差系数进行解码的步骤1220处,熵解码器420在处理器205的执行下对残差系数的列表进行解码,并将残差系数组装成用于变换块的阵列。该组装遵循通常将残差系数分组为 4×4 个子块的扫描模式。通常沿反方向扫描残差系数的列表,该反方向是从“最后一个”有效残差系数返回朝向左上方的残差系数(“DC”残差系数,尽管这可能不对应于不基于余弦函数的变换的块偏移)。对上下文编码bin进行解码,以指示在各子块中存在有效残差系数,并且在指示的情况下,对附加的上下文编码bin进行解码,以指示子块中的各残差系数的有效性。可以通过对“大于1”或“大于2”标志进行解码来进一步表现各残差系数的大小的特征。最后,对于大小不是由前一上下文编码bin确定的残差系数,使用具有旁路编码bin的Golomb-Rice编码来对“剩余大小”进行解码。作为该解码的结果,从位流133解码变换块的残差系数424。然后,处理器205中的控制从对残差系数进行解码的步骤1220进入对残差系数进行逆量化的步骤1230。

[0137] 在对残差系数进行逆量化的步骤1230处,去量化器模块428在处理器205的执行下通过根据量化参数缩放残差系数424来产生中间变换系数432。可以通过应用通常仅应用于变换块的左上方区域的不可分离的次要变换来对中间变换系数432进行变换,以产生中间变换系数432。作为该缩放操作的结果,中间变换系数432具有与视频编码器中的量化之前的系数(即,变换系数332)大致相同的大小。通过可选地应用逆不可分离的次要变换(即,逆次要变换模块436),从中间变换系数432获得重建变换系数440。然后,处理器205中的控制从对残差系数进行逆量化的步骤1230进入对EMT bin串进行解码的步骤1240。

[0138] 在对EMT bin串进行解码的步骤1240处,熵解码器420在处理器205的执行下从位流133中解码截断一元bin串。截断一元bin串是在图10A的列1042中列出的bin串其中之一。如图10A所示和以上所述,一元bin串在长度上局限于三个bin(其在0到3的范围内)。截断一元bin串在长度上为从1到3个bin,其中对第一个bin进行上下文编码,利用不同的上下文对第二个bin(在存在的情况下)进行上下文编码,并且对第三个bin(在存在的情况下)进行旁路编码。该二值化内容如参考图10所述。然后,处理器205中的控制从对EMT bin串进行解码的步骤1240进入确定EMT索引的步骤1250。

[0139] 在确定EMT索引的步骤1250处,视频解码器134中的变换索引导出器920在处理器205的执行下,根据图10A中的表1000来确定水平变换索引920和垂直变换索引925。然后,处理器205中的控制从确定EMT索引的步骤1250进入确定变换集的步骤1260。

[0140] 在确定变换集的步骤1260处,逆变换模块444在处理器205的执行下确定映射840

中的哪个变换集将用于主要逆变换的水平和垂直变换。特别地,使用帧内预测模式458,变换集表910用于确定水平变换集915和垂直变换集917。然后,处理器205中的控制从确定变换集的步骤1260进入选择变换的步骤1270。

[0141] 在选择变换的步骤1270处,逆变换模块444在处理器205的执行下确定水平变换950和垂直变换955。水平变换950和垂直变换955的确定分别由水平变换选择器940和垂直变换选择器945进行。这些模块如图9所示采用水平变换集索引915、垂直变换集索引917、水平变换索引920和垂直变换索引925作为输入,并且根据图8B来产生所选择的水平变换950和所选择的垂直变换960。然后,处理器205中的控制从选择变换的步骤1270进入应用水平变换的步骤1280。

[0142] 在应用水平变换的步骤1280处,逆变换模块444在处理器205的执行下,使用水平变换模块960来应用所选择的水平变换950。将所选择的水平变换950应用于重建变换系数440的各行,以产生水平变换的变换系数970。水平变换的变换系数970被供给至垂直变换模块980。然后,垂直变换模块980通过应用所选择的垂直变换955来产生残差样本448。如参考图4所述,残差样本950进一步用于与帧内预测PU的相加以产生重建样本。然后,方法1200终止。

[0143] 在方法1100和1200的另一配置中,修改截断一元bin串的二值化内容,使得仅使用一个上下文编码bin。特别地,参考图10A,二值化内容1044的第一个bin保持被上下文编码,而二值化内容1044的第二个bin使用旁路编码bin来代替上下文编码。仅使用一个上下文编码bin的配置以压缩效率略微下降为代价节省了实现成本。然而,仅使用一个上下文编码bin的配置能够实现略高的吞吐量。较高的吞吐量是因为:至少由于缺少上下文状态更新,因此熵编码器338和熵解码器420中的CABAC处理能够以与上下文编码bin相比更高的速率来处理旁路编码bin。

[0144] 如参考图10A所述,选择分别具有值“0”和“1”或者分别具有值“1”和“0”的水平变换索引H和垂直变换索引V的概率大致相等。如此,图10A中所示的二值化内容的替代二值化内容对第三个旁路编码bin的使用取反。对第三个bin的使用取反的配置将bin串“1 1 1”映射到用于各个垂直和水平变换索引的“0”、“1”,并且将bin串“1 1 0”映射到用于各个垂直和水平变换索引的“1”、“0”。由于这两个选择之间的概率大致相等,因而实现了等效的压缩优势。

[0145] 作为水平方向和垂直方向上的显式多变换选择的二值化内容(特别是根据选择各索引值的预期概率来对这两者进行联合编码的方面)的结果,与将独立的句法元素各自用于水平和垂直索引(例如,各自为一个上下文编码bin)的情况相比,提高了编码效率。

[0146] 产业上的可利用性

[0147] 所述的配置可应用于计算机和数据处理行业,并且特别可用于对诸如视频和图像信号等的信号进行编码或解码的数字信号处理,从而在不会由于残差系数的非局部扫描而造成内存带宽方面的过高成本的情况下实现高压缩效率。

[0148] 前述仅说明本发明的一些实施例,并且可以在没有背离本发明的范围和精神的情况下对本发明进行修改和/或改变,其中这些实施例仅是示例性而非限制性的。

[0149] 在本说明书的上下文中,词语“包括”意味着“主要但未必仅包括”或“具有”或“包含”,而不是“仅由…组成”。词语“包括(comprising)”的诸如“comprise”和“comprises”等

的词尾变化具有相应的变化含义。

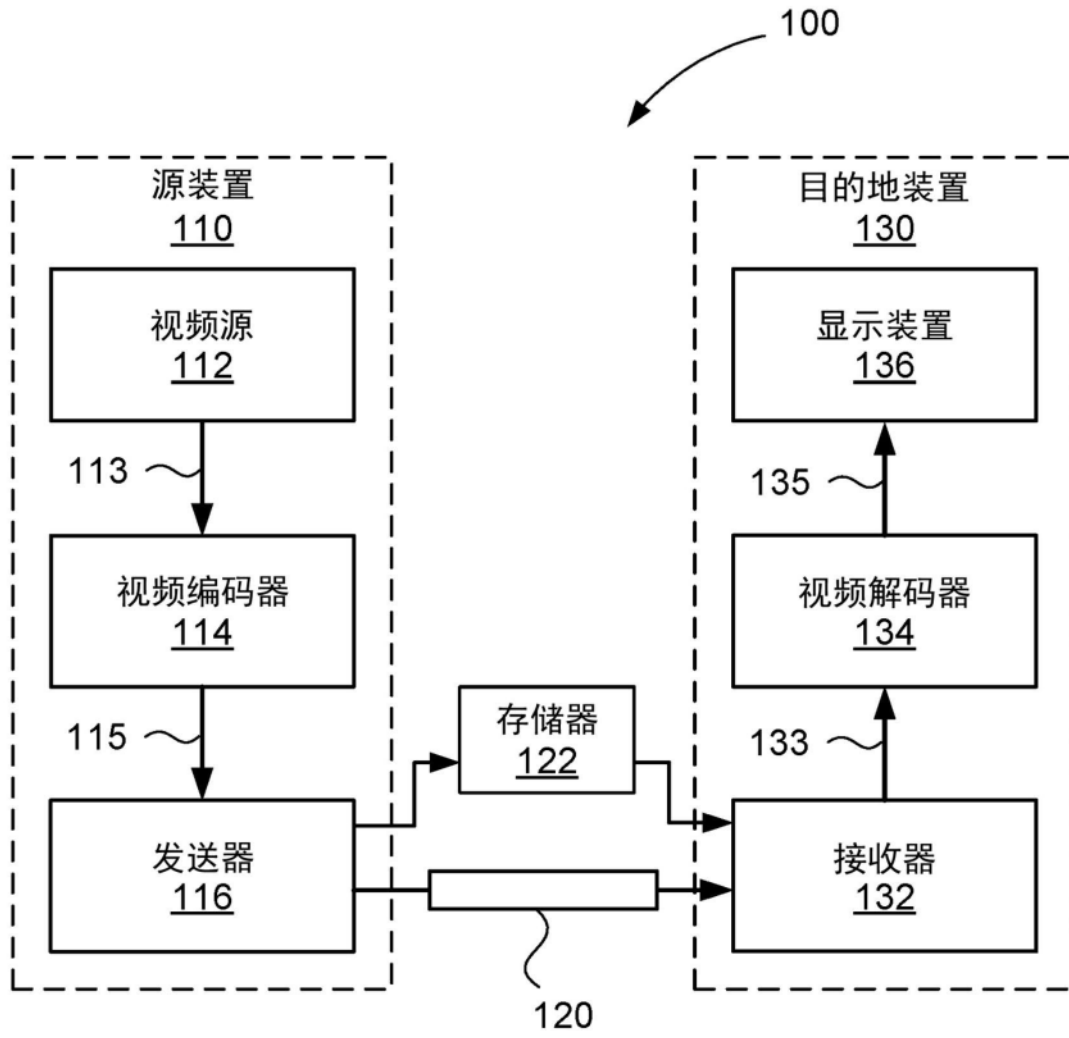


图1

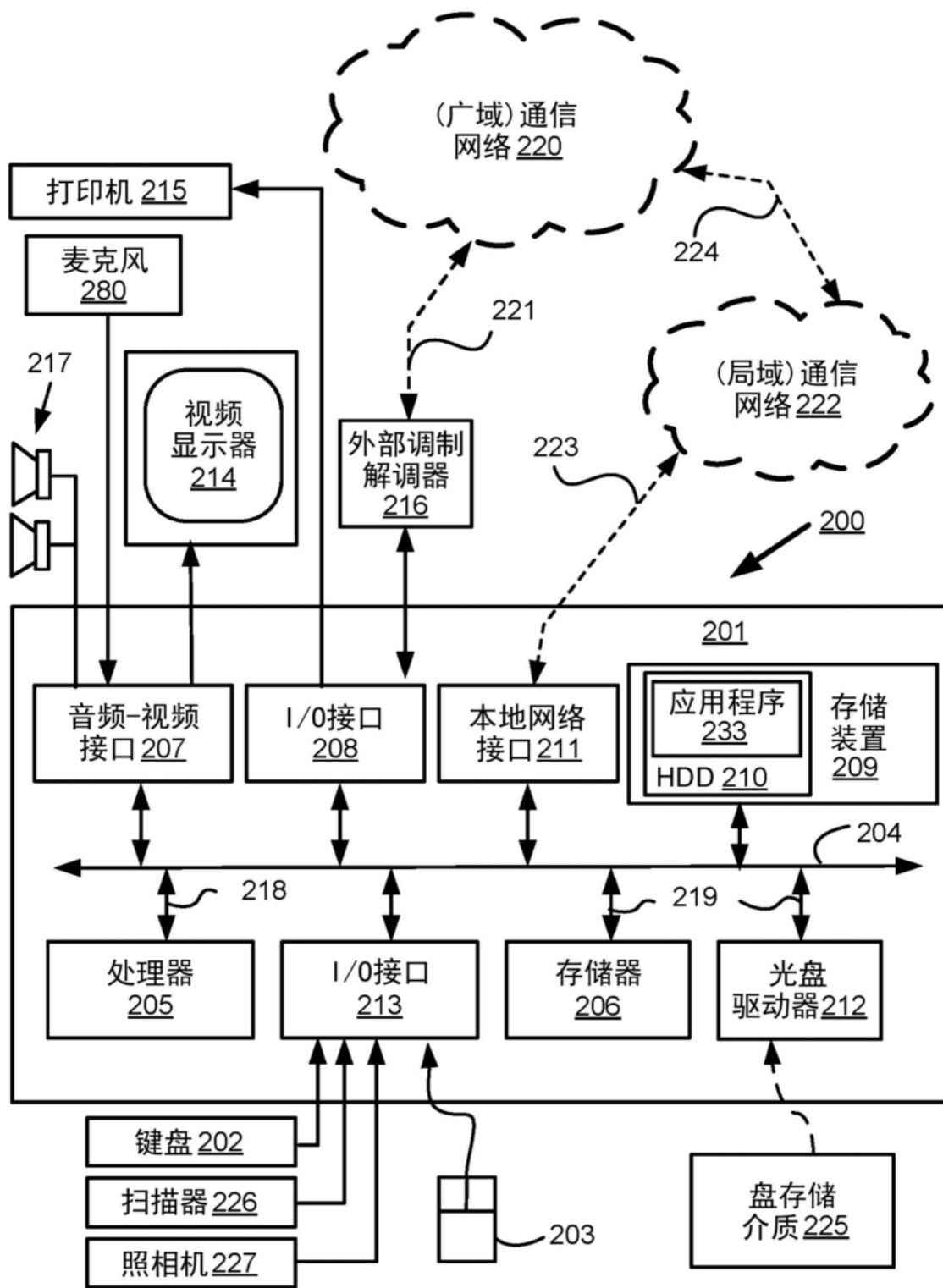


图2A

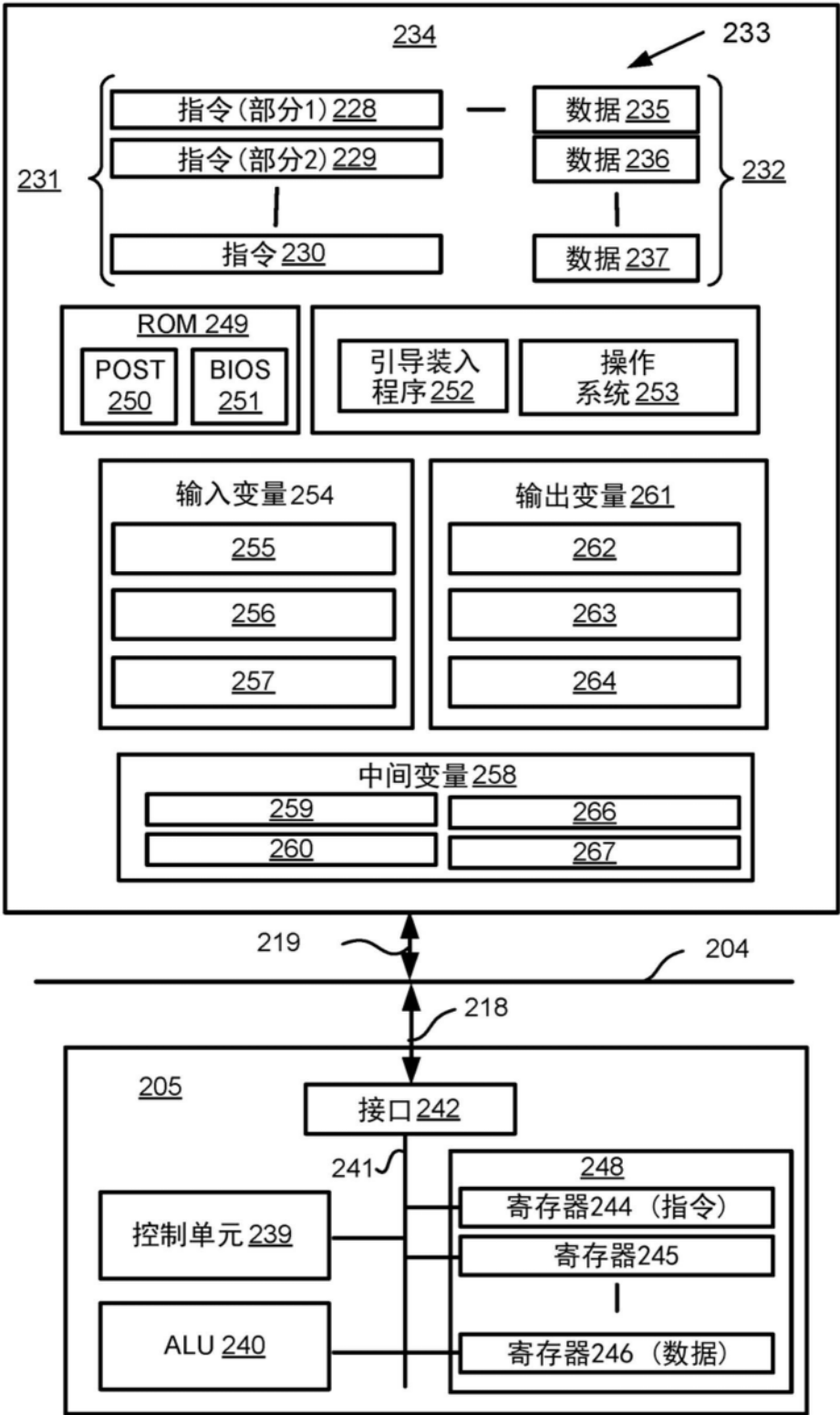


图2B

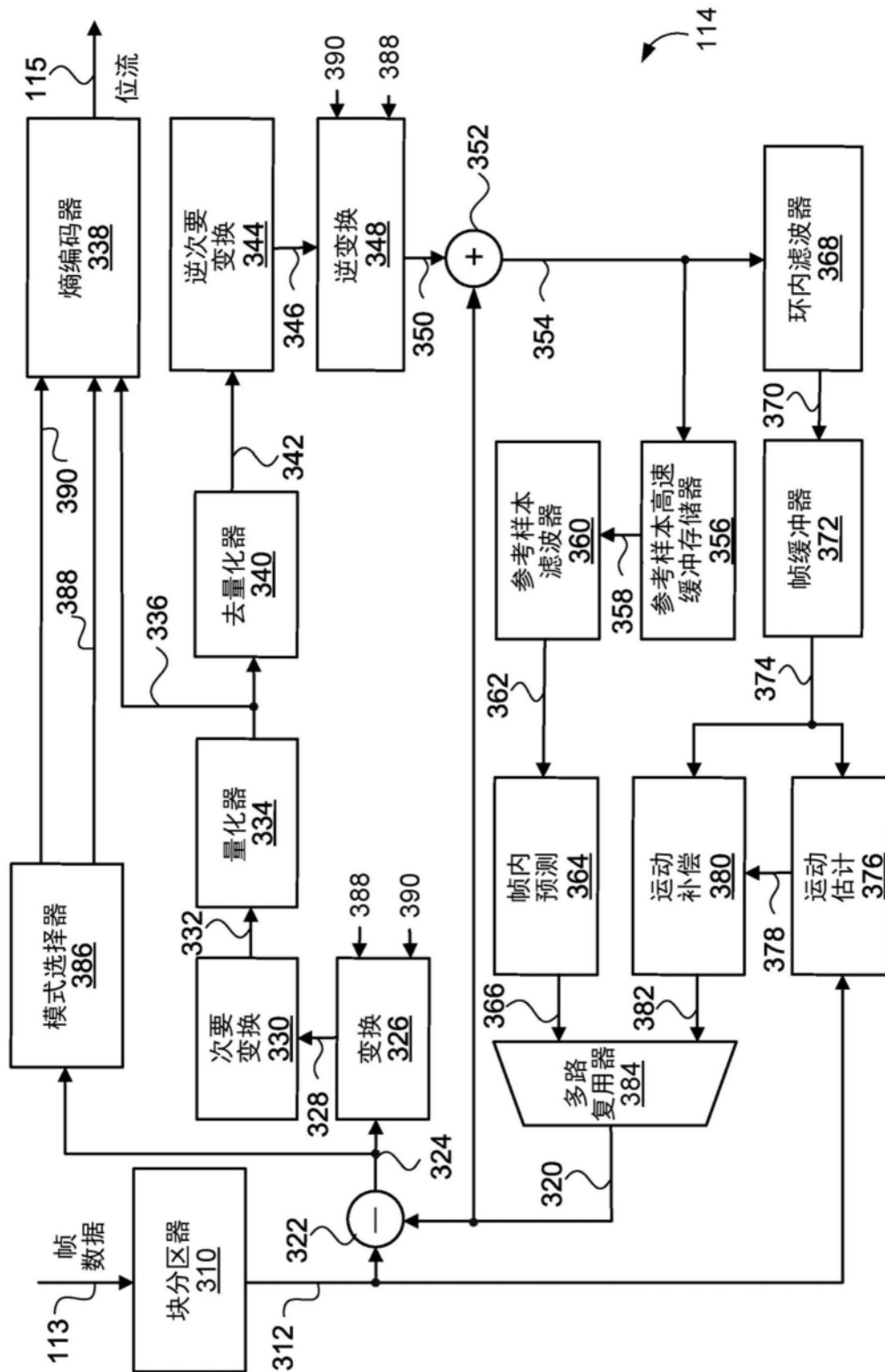


图3

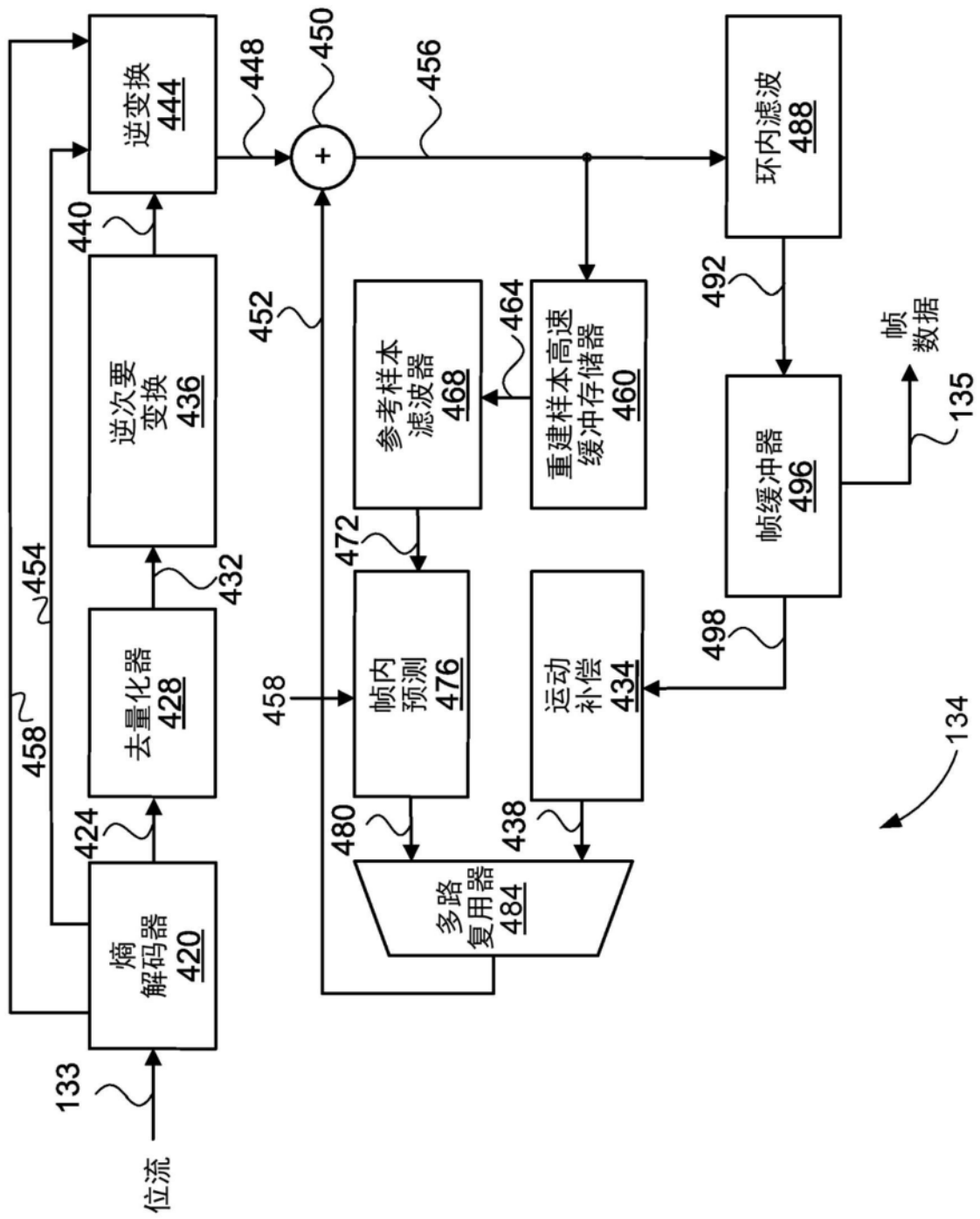


图4

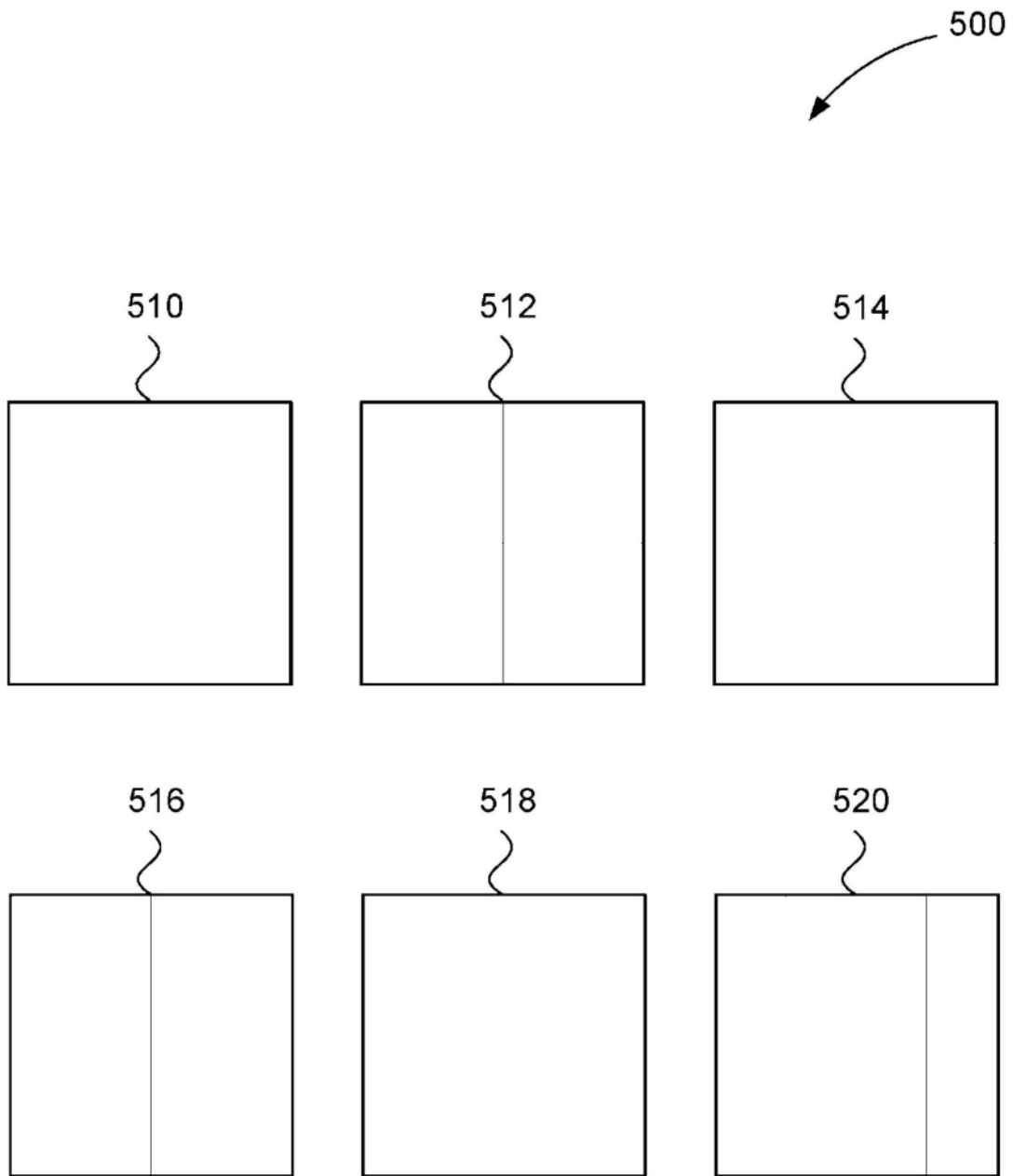


图5

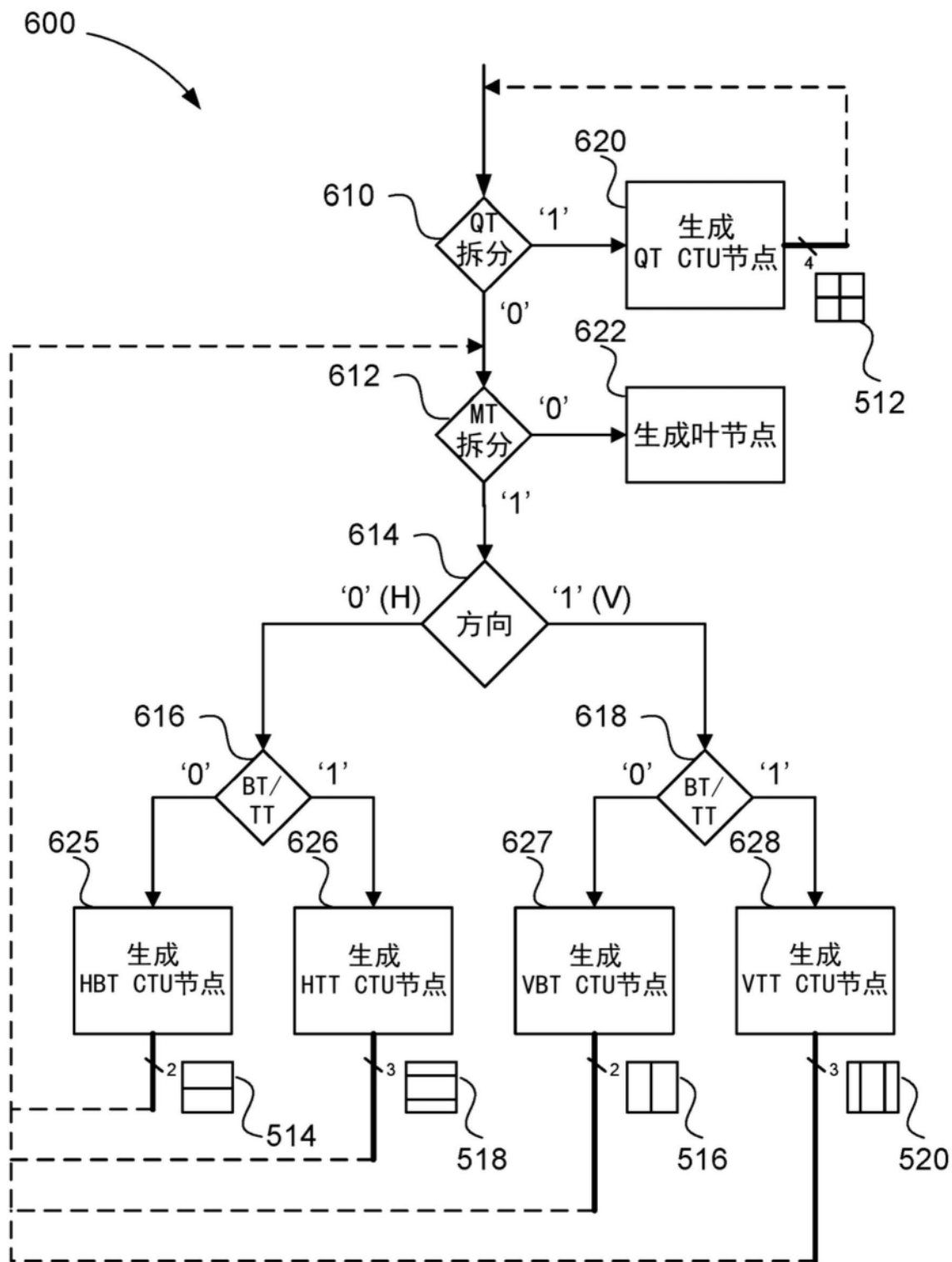


图6

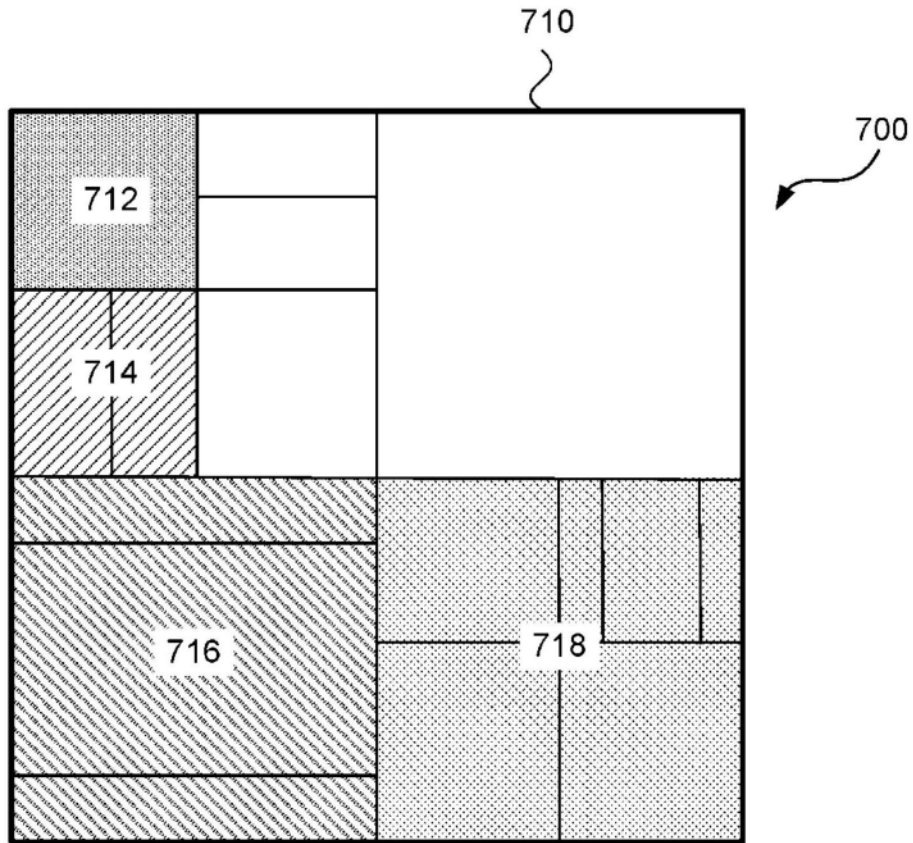


图7A

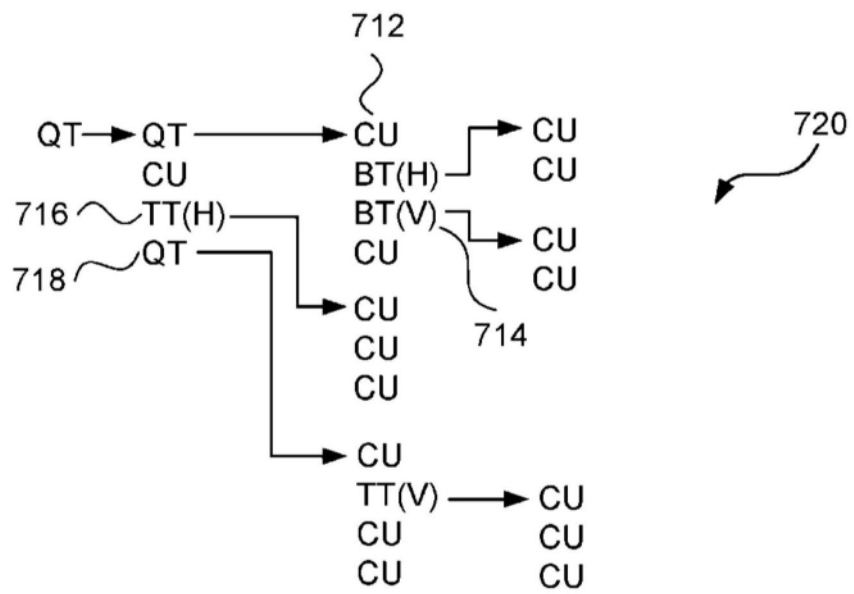


图7B

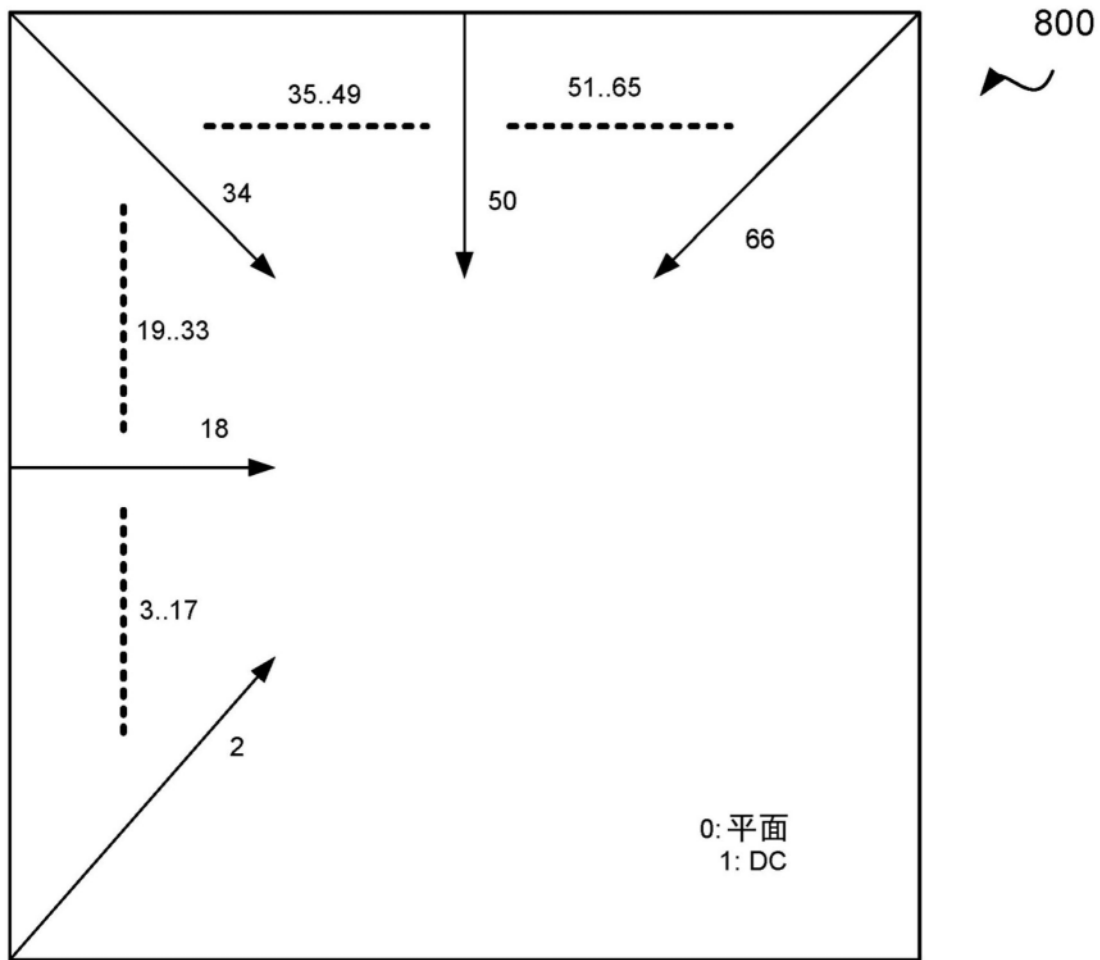


图8A

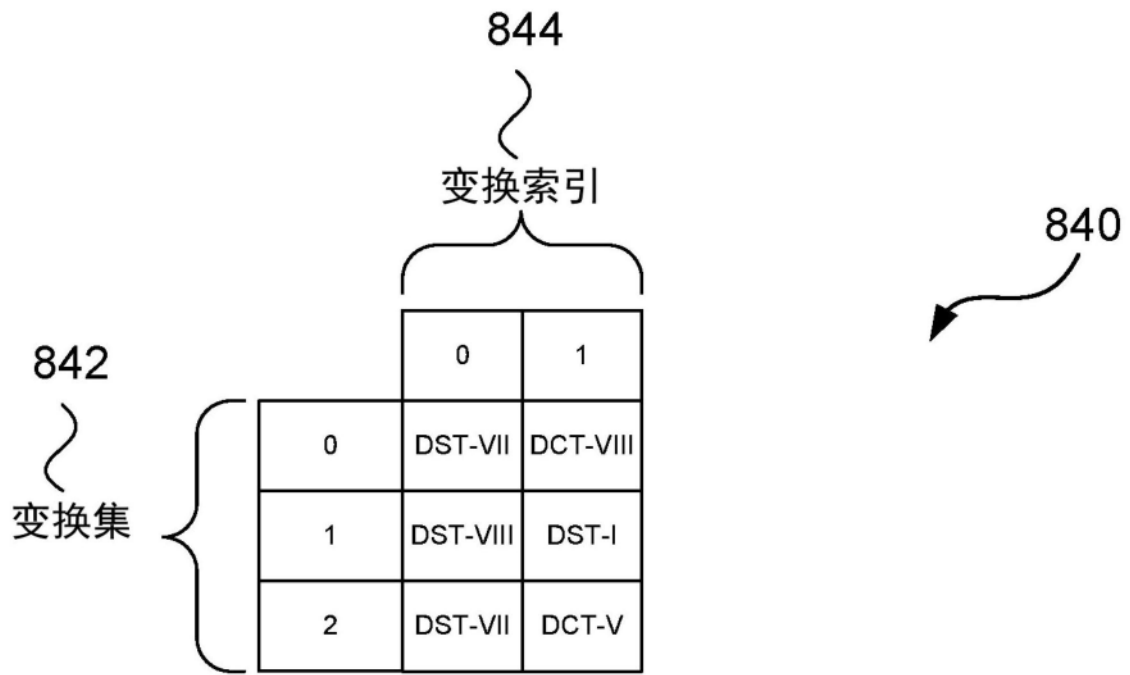


图8B

880



帧内模式	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V	2	1	0	1	0	1	0	1	0	1	0	1	0	1	0
H	2	1	0	1	0	1	0	1	0	1	0	1	0	1	2

帧内模式	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
V	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
H	2	2	2	2	2	2	2	2	1	0	1	0	1	0	1

帧内模式	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
V	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
H	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

帧内模式	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
V	1	2	2	2	2	2	2	2	2	2	1	0	1	0	1
H	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1

帧内模式	60	61	62	63	64	65	66
V	0	1	0	1	0	1	0
H	0	1	0	1	0	1	0

图8C

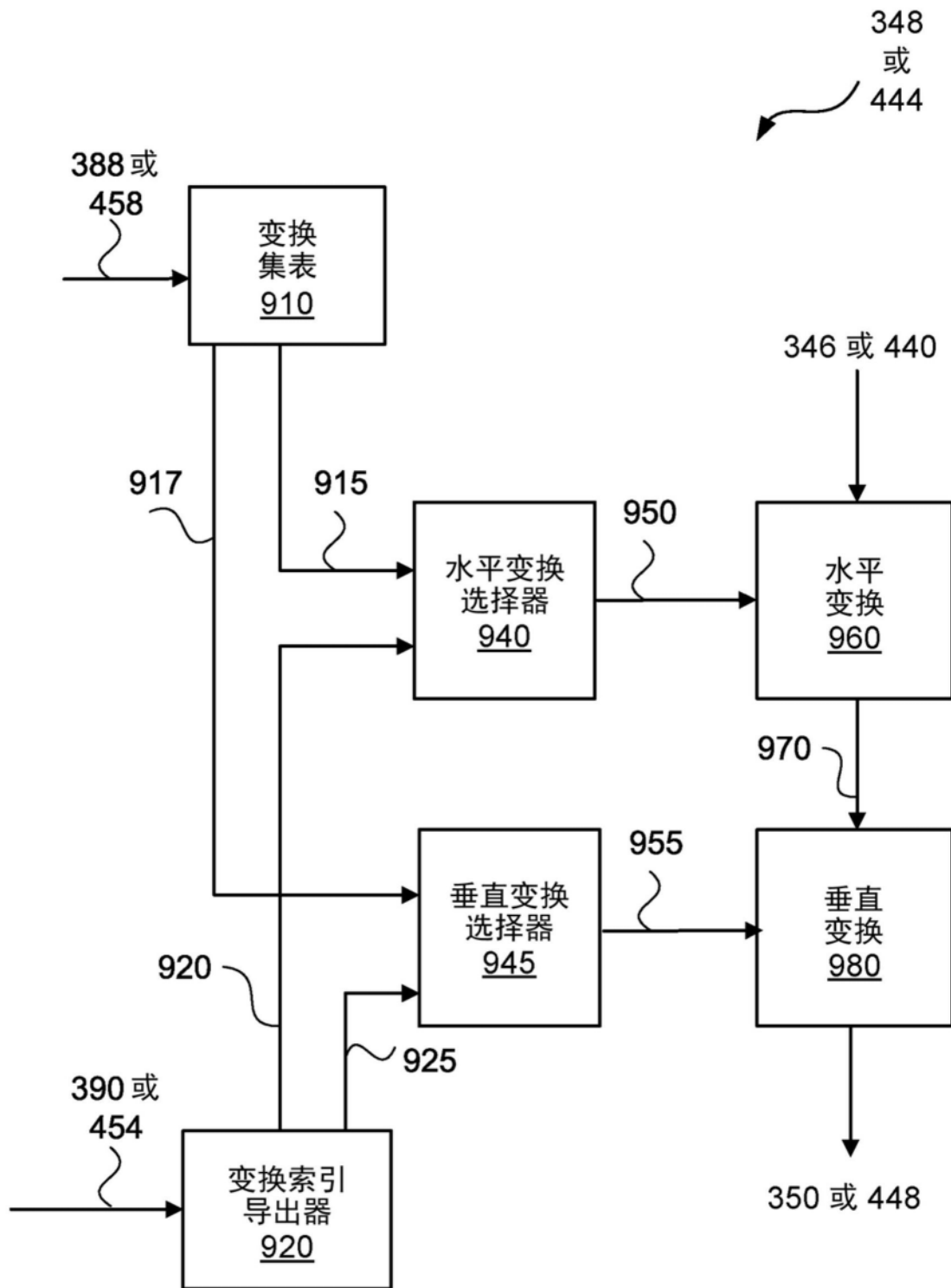


图9

1000

1042	1044	1046	1048
bin串	二值化内容	V	H
0	Ctx0	0	0
1 1 0	Ctx0 + Ctx1 + Byp	0	1
1 1 1	Ctx0 + Ctx1 + Byp	1	0
1 0	Ctx0 + Ctx1	1	1

图10A

1050

bin串	二值化内容	V	H
0	Ctx0	0	0
1 0	Ctx0 + Ctx1	0	1
1 1 0	Ctx0 + Ctx1 + ctx2	1	0
1 1 1	Ctx0 + Ctx1 + Ctx2	1	1

图10B

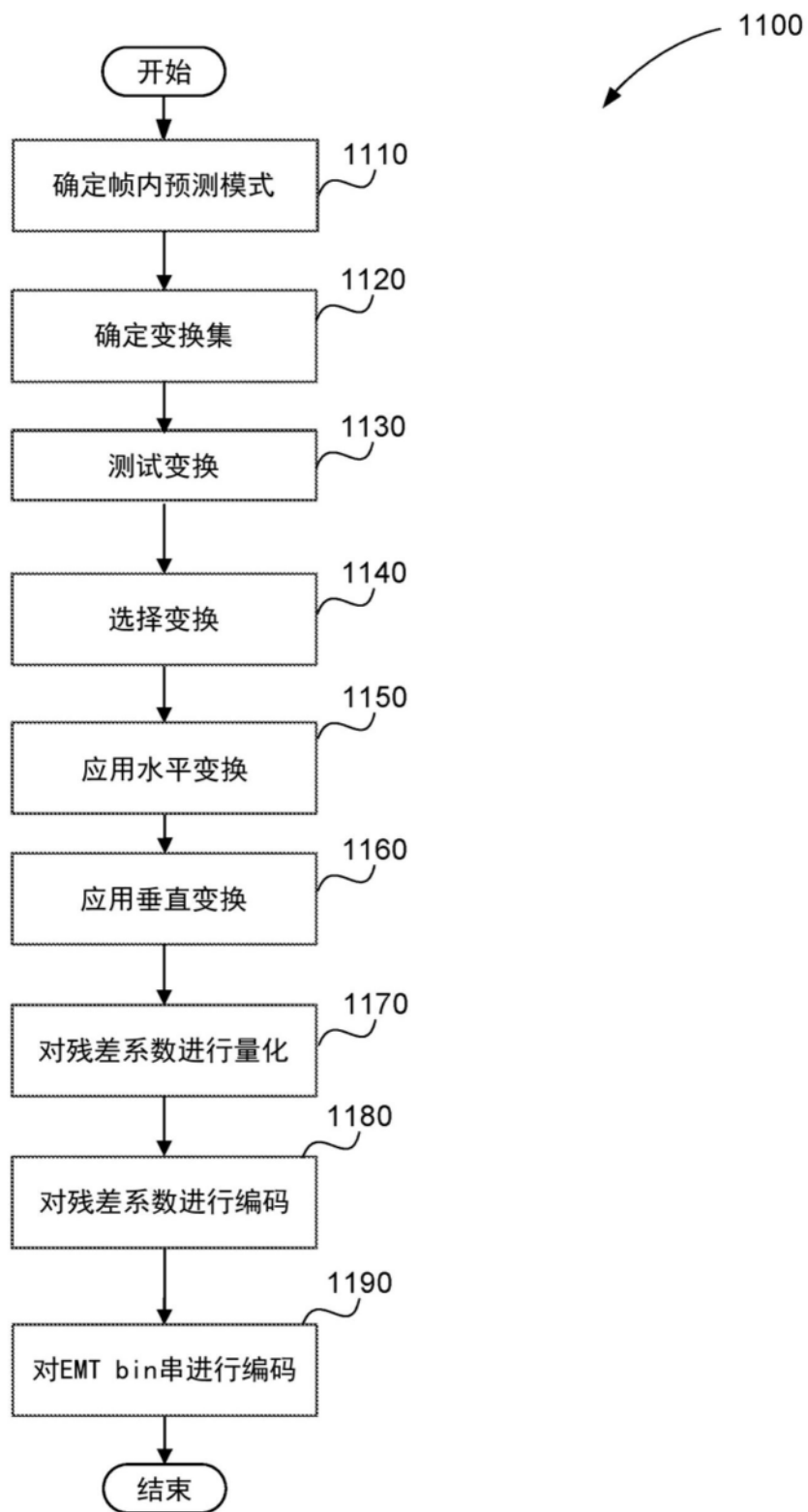


图11

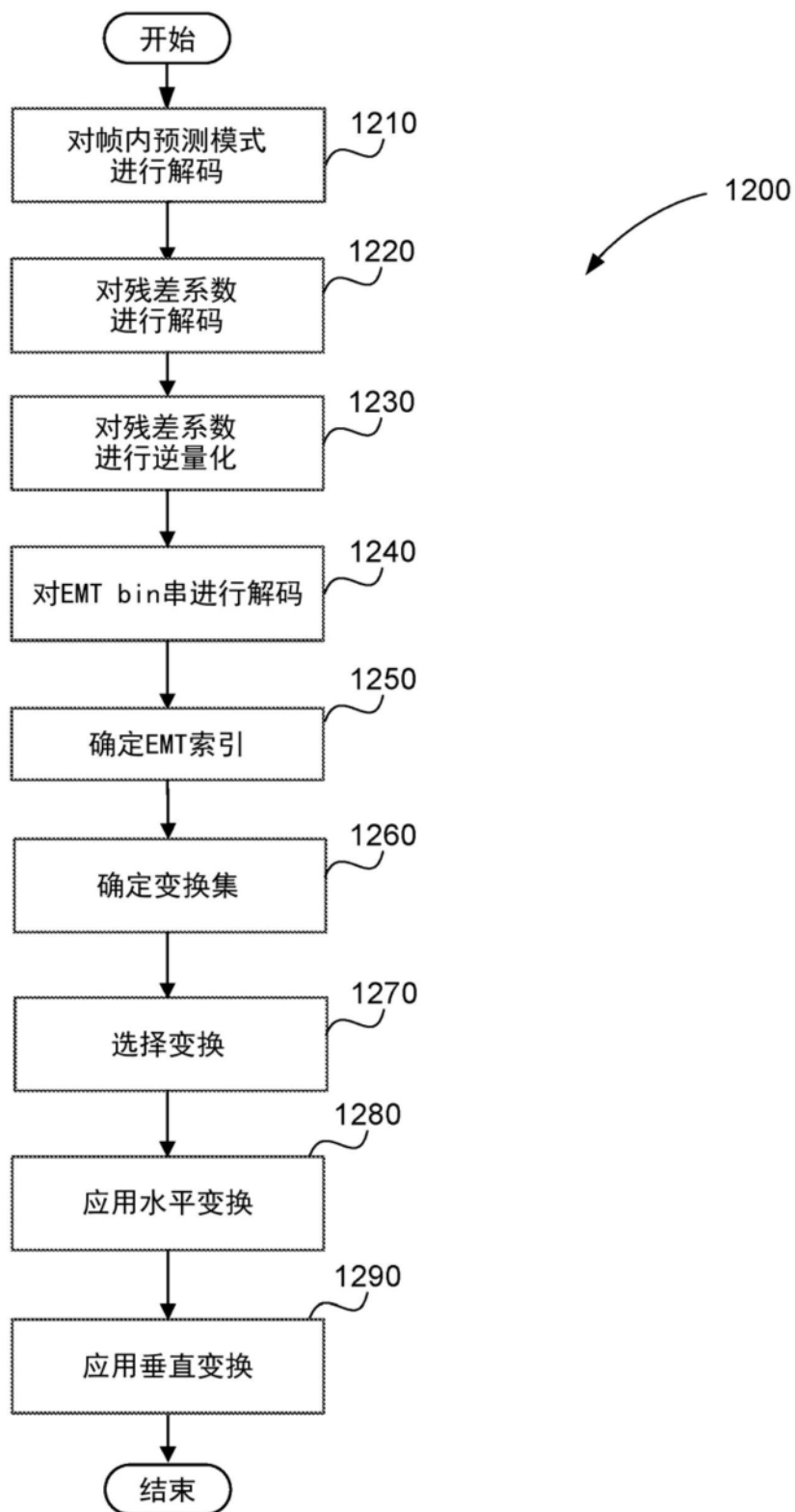


图12