



(12) 发明专利申请

(10) 申请公布号 CN 11978852 A

(43) 申请公布日 2025. 04. 08

(21) 申请号 202510115710.7

H04N 19/593 (2014.01)

(22) 申请日 2020.04.15

H04N 19/70 (2014.01)

(30) 优先权数据

H04N 19/13 (2014.01)

2019204437 2019.06.24 AU

H04N 19/91 (2014.01)

(62) 分案原申请数据

202080042249.2 2020.04.15

(71) 申请人 佳能株式会社

地址 日本

(72) 发明人 克里斯托弗·詹姆斯·罗斯沃恩

伊夫特哈尔·艾哈迈德

(74) 专利代理机构 北京魏启学律师事务所

11398

专利代理师 陈涛

(51) Int. Cl.

H04N 19/11 (2014.01)

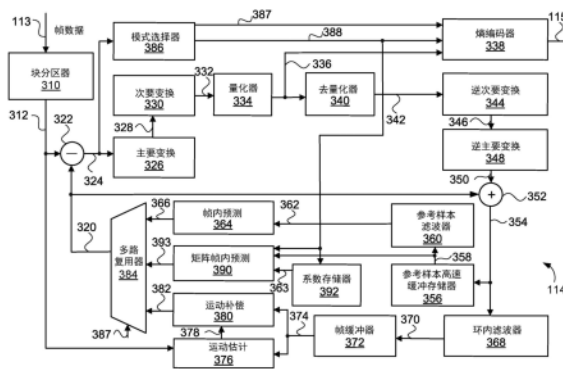
权利要求书3页 说明书33页 附图20页

(54) 发明名称

用于生成预测样本的方法和设备、存储介质和计算机程序产品

(57) 摘要

本发明公开了用于生成预测样本的方法和设备、存储介质和计算机程序产品。还公开了从视频位流中解码图像帧的编码树的编码单元的方法。该方法包括：将编码树的区域拆分成多个编码块，各编码块包括预测块；以及确定各编码块的预测块的矩阵帧内预测标志，各矩阵帧内预测标志指示是否针对编码块之一的预测块使用矩阵帧内预测，该确定在区域满足阈值的情况下基于区域的面积，或者在区域的面积不满足阈值的情况下基于区域的预算。该方法还包括：针对根据所确定的标志被确定为使用矩阵帧内预测的各预测块，从存储器读取矩阵系数；以及使用预测块对编码单元进行解码，所述预测块是使用各预测块的参考样本和矩阵系数针对区域中的各编码单元生成的。



1. 一种生成图像帧的编码树单元中的目标块的预测样本的方法,所述方法包括:

对所述目标块的矩阵帧内预测标志进行解码,所述矩阵帧内预测标志指示针对所述目标块是否使用矩阵帧内预测;

在所述矩阵帧内预测标志指示针对所述目标块使用矩阵帧内预测的情况下,对所述目标块的矩阵帧内预测模式进行解码,其中,截断二代码能够用于所述矩阵帧内预测模式;

根据所述矩阵帧内预测模式,选择用于所述目标块的矩阵帧内预测的矩阵;以及

通过应用基于与所述目标块相邻的样本的输入样本和根据所述矩阵帧内预测模式所选择的矩阵的相乘来生成所述预测样本,

其中,与所述目标块相邻的样本包括所述目标块的左侧和上方的样本,以及

其中,在所述编码树单元中的给定区域被拆分为各自具有 16×8 的大小的四个块并且所述目标块是所述四个块其中之一的情況下,对所述目标块的所述矩阵帧内预测标志的解码取决于与所述目标块相邻的两个块的矩阵帧内预测的使用状态。

2. 一种生成图像帧的编码树单元中的目标块的预测样本的方法,所述方法包括:

选择用于所述目标块的矩阵帧内预测的矩阵;

通过应用基于与所述目标块相邻的样本的输入样本和所选择的矩阵的相乘来生成所述预测样本,其中,与所述目标块相邻的样本包括所述目标块的左侧和上方的样本;

对所述目标块的矩阵帧内预测标志进行编码,所述矩阵帧内预测标志指示针对所述目标块是否使用矩阵帧内预测;以及

在针对所述目标块使用矩阵帧内预测的情况下,对指示所选择的矩阵的矩阵帧内预测模式进行编码,其中,截断二代码能够用于所述矩阵帧内预测模式,

其中,在所述编码树单元中的给定区域被拆分为各自具有 16×8 的大小的四个块并且所述目标块是所述四个块其中之一的情況下,对所述目标块的所述矩阵帧内预测标志的编码取决于与所述目标块相邻的两个块的矩阵帧内预测的使用状态。

3. 一种生成图像帧的编码树单元中的目标块的预测样本的设备,所述设备包括:

第一解码单元,其被配置为对所述目标块的矩阵帧内预测标志进行解码,所述矩阵帧内预测标志指示针对所述目标块是否使用矩阵帧内预测;

第二解码单元,其被配置为在所述矩阵帧内预测标志指示针对所述目标块使用矩阵帧内预测的情况下,对所述目标块的矩阵帧内预测模式进行解码,其中,截断二代码能够用于所述矩阵帧内预测模式;

选择单元,其被配置为根据所述矩阵帧内预测模式,选择用于所述目标块的矩阵帧内预测的矩阵;以及

生成单元,其被配置为通过应用基于与所述目标块相邻的样本的输入样本和根据所述矩阵帧内预测模式所选择的矩阵的相乘来生成所述预测样本,

其中,与所述目标块相邻的样本包括所述目标块的左侧和上方的样本,以及

其中,在所述编码树单元中的给定区域被拆分为各自具有 16×8 的大小的四个块并且所述目标块是所述四个块其中之一的情況下,对所述目标块的所述矩阵帧内预测标志的解码取决于与所述目标块相邻的两个块的矩阵帧内预测的使用状态。

4. 一种生成图像帧的编码树单元中的目标块的预测样本的设备,所述设备包括:

选择单元,其被配置为选择用于所述目标块的矩阵帧内预测的矩阵;

生成单元,其被配置为通过应用基于与目标块相邻的样本的输入样本和所选择的矩阵的相乘来生成所述预测样本,其中,与目标块相邻的样本包括目标块的左侧和上方的样本;

第一编码单元,其被配置为对目标块的矩阵帧内预测标志进行编码,所述矩阵帧内预测标志指示针对目标块是否使用矩阵帧内预测;以及

第二编码单元,其被配置为在针对目标块使用矩阵帧内预测的情况下,对指示所选择的矩阵的矩阵帧内预测模式进行编码,其中,截断二源码能够用于所述矩阵帧内预测模式,

其中,在所述编码树单元中的给定区域被拆分为各自具有 16×8 的大小的四个块并且所述目标块是所述四个块其中之一,对目标块的所述矩阵帧内预测标志的编码取决于与目标块相邻的两个块的矩阵帧内预测的使用状态。

5. 一种非暂时性计算机可读存储介质,其存储有程序,所述程序使计算机执行生成图像帧的编码树单元中的目标块的预测样本的方法,所述方法包括:

对目标块的矩阵帧内预测标志进行解码,所述矩阵帧内预测标志指示针对目标块是否使用矩阵帧内预测;

在所述矩阵帧内预测标志指示针对目标块使用矩阵帧内预测的情况下,对目标块的矩阵帧内预测模式进行解码,其中,截断二源码能够用于所述矩阵帧内预测模式;

根据所述矩阵帧内预测模式,选择用于目标块的矩阵帧内预测的矩阵;以及

通过应用基于与目标块相邻的样本的输入样本和根据所述矩阵帧内预测模式所选择的矩阵的相乘来生成所述预测样本,

其中,与目标块相邻的样本包括目标块的左侧和上方的样本,以及

其中,在所述编码树单元中的给定区域被拆分为各自具有 16×8 的大小的四个块并且所述目标块是所述四个块其中之一,对目标块的所述矩阵帧内预测标志的解码取决于与目标块相邻的两个块的矩阵帧内预测的使用状态。

6. 一种非暂时性计算机可读存储介质,其存储有程序,所述程序使计算机执行生成图像帧的编码树单元中的目标块的预测样本的方法,所述方法包括:

选择用于目标块的矩阵帧内预测的矩阵;

通过应用基于与目标块相邻的样本的输入样本和所选择的矩阵的相乘来生成所述预测样本,其中,与目标块相邻的样本包括目标块的左侧和上方的样本;

对目标块的矩阵帧内预测标志进行编码,所述矩阵帧内预测标志指示针对目标块是否使用矩阵帧内预测;以及

在针对目标块使用矩阵帧内预测的情况下,对指示所选择的矩阵的矩阵帧内预测模式进行编码,其中,截断二源码能够用于所述矩阵帧内预测模式,

其中,在所述编码树单元中的给定区域被拆分为各自具有 16×8 的大小的四个块并且所述目标块是所述四个块其中之一,对目标块的所述矩阵帧内预测标志的编码取决于与目标块相邻的两个块的矩阵帧内预测的使用状态。

7. 一种计算机程序产品,其包括程序,所述程序在被计算机执行时使所述计算机进行根据权利要求1所述的方法。

8. 一种计算机程序产品,其包括程序,所述程序在被计算机执行时使所述计算机进行

根据权利要求2所述的方法。

用于生成预测样本的方法和设备、存储介质和计算机程序 产品

[0001] (本申请是申请日为2020年4月15日、申请号为2020800422492、发明名称为“用于编码和解码视频样本的块的方法、设备和系统”的申请的分案申请。)

技术领域

[0002] 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

[0003] 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:还已知为“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/国际电工协会联合技术协会1/小组协会29/工作组11(ISO/IEC JTC1/SC29/WG11)的成员。

[0004] 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、存储器利用率和带宽方面)提供可接受的折衷。

[0005] 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,需要一个主颜色通道和两个二次颜色通道。主颜色通道通常被称为“亮度”通道,并且(一个或多个)二次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将发光度(根据变换方程映射到“亮度”)集中在Y(主)通道中,并且将色度集中在Cb和Cr(二次)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通道进行空间采样。4:2:0色度格式通常用于“消费者”应用中,诸如互联网视频流式传输、广播电视以及蓝光TM盘上的存储。在水平方向上以半速率对Cb和Cr通道进行子采样而不是垂直地进行子采样被称为“4:2:2色度格式”。4:2:2色度格式通常用于专业应用中,包括用于电影制作等的镜头的捕获。4:2:2色度格式的较高采样速率使得所得视频对编辑操作(诸如颜色分级等)更有弹性。在分发到消费者之前,4:2:2色度格式材料经常被

转换为4:2:0色度格式,然后被编码以分发到消费者。除了色度格式之外,视频还由分辨率和帧频来表征。示例分辨率是具有 3840×2160 的分辨率的超高清清晰度(UD)或具有 7680×4320 的分辨率的“8K”,并且示例帧频是60Hz或120Hz。亮度样本速率的范围可以从约500兆样本/秒到数千兆样本/秒。对于4:2:0色度格式,各个色度通道的采样率是亮度采样率的四分之一,并且对于4:2:2色度格式,各个色度通道的采样率是亮度采样率的一半。

[0006] VVC标准是一种“基于块”的编解码器,其中,帧首先被分割成称为“编码树单元”(CTU)的正方形区域阵列。CTU通常占据相对大的区域,诸如 128×128 个亮度样本等。然而,各帧的右和底边缘的CTU的区域可能较小。与各CTU相关联的是针对亮度通道的“编码树”和针对色度通道的附加编码树。编码树定义了将CTU的区域分解为一组区域,也称为“编码块”(CB)。还可以针对单个编码树指定亮度通道和色度通道这两者的块,在这种情况下,并置的编码块的集合被称为“编码单元”(CU),即,各个CU具有用于各个颜色通道的编码块。按特定顺序处理CB以进行编码或解码。作为使用4:2:0色度格式的结果,包括 128×128 亮度样本区域的亮度编码树的CTU具有与 128×128 亮度样本区域并置的 64×64 色度样本区域的相应色度编码树。当单个编码树用于亮度通道和色度通道时,给定区域的并置块的集合通常称为“单元”,例如上述CU以及“预测单元”(PU)和“变换单元”(TU)。当针对给定区域使用单独的编码树时,将使用上述CB以及“预测块”(PB)和“变换块”(TB)。

[0007] 尽管在“单元”和“块”之间有上述区别,但是术语“块”可以用作针对将操作应用于所有颜色通道的帧的区域(area)或区(region)的通用术语。

[0008] 对于各CU,生成帧数据的相应区域的内容(样本值)的预测单元(PU) (“预测单位”)。此外,形成了在编码器的输入处看到的预测与区域内容之间的差(或在空间域中的“残差”)的表示。各颜色通道的差可以被变换并编码为残差系数的序列,从而形成给定CU的一个或多个TU。所应用的变换可以是应用于残差值的各个块的离散余弦变换(DCT)或其它变换。该变换是分开应用的,即分两遍进行二维变换。首先通过对块中的各行样本应用一维变换来对块进行变换。然后,通过对部分结果的各列应用一维变换来对部分结果进行变换,以产生基本上对残差样本进行去相关的变换系数的最终块。VVC标准支持各种大小的变换,包括矩形块(各边尺寸为2的幂)的变换。量化变换系数以用于将熵编码在位流中。

[0009] VVC的特征是帧内预测和帧间预测。帧内预测涉及使用正被使用的帧中的先前处理的样本来生成对帧中的当前样本块的预测。帧间预测涉及使用从先前解码帧获得的样本块来生成对帧中的样本的当前块的预测。先前解码帧根据运动矢量从当前块的空间位置偏移,该运动矢量通常已应用滤波。帧内预测块可以是均匀样本值(“DC帧内预测”)、具有偏移及水平和垂直梯度的平面(“平面帧内预测”)、或者具有在特定方向上应用的相邻样本的块的群体(“角度帧内预测”)或使用相邻样本和所选择的矩阵系数的矩阵乘法的结果。由于相邻样本包括来自先前处理块的样本,因此用于帧内预测的反馈环路是相当限制性的,并且计算复杂度需要被保持得低于满足最高支持的分辨率和帧频所需要的水平。

发明内容

[0010] 本发明的目的是基本上克服或至少改善现有布置的一个或多个缺点。

[0011] 本公开的一个方面提供一种从视频位流中解码图像帧的编码树的编码单元的方法,所述方法包括:将所述编码树的区域拆分成多个编码块,所述编码块中的各个编码块包

括预测块；确定所述编码块中的各个编码块的预测块的矩阵帧内预测标志，各个矩阵帧内预测标志指示针对所述编码块之一的预测块是否使用矩阵帧内预测，所述确定 (i) 在所述区域满足阈值的情况下基于所述区域的面积，或者 (ii) 在所述区域的面积不满足所述阈值的情况下基于所述区域的预算；针对根据所确定的标志被确定为使用矩阵帧内预测的各个预测块，从存储器读取矩阵系数；以及使用预测块对所述编码单元进行解码，所述预测块是使用各个预测块的参考样本和所述矩阵系数针对所述区域中的各个编码单元生成的。

[0012] 在另一方面，所述阈值是大于512个亮度样本的大小。

[0013] 在另一方面，所述阈值是大于64个亮度样本的大小。

[0014] 在另一方面，所述预算使得能够针对所述区域读取 4×4 块的40个字。

[0015] 在另一方面，仅在使用矩阵帧内预测的情况下，才针对所述CU解码矩阵帧内预测标志。

[0016] 在另一方面，无论是否使用矩阵帧内预测，都针对所述CU解码矩阵帧内预测标志。

[0017] 本公开的另一面提供一种从视频位流中解码图像帧的编码树的编码单元的方法，所述方法包括：将所述编码树的区域拆分成多个编码块，各个编码块包括预测块；基于各个编码块的大小来确定该编码块的预测块的矩阵帧内预测标志，各个矩阵帧内预测标志指示针对相应编码块的预测块是否使用矩阵帧内预测；针对根据所确定的标志被确定为使用矩阵帧内预测的各个预测块，从存储器读取矩阵系数；以及从预测块对所述编码单元进行解码，所述预测块是使用各个预测块的参考样本和所述矩阵系数所生成的所述区域中的各个编码单元的预测块。

[0018] 在另一方面，在所述编码单元的大小不是 4×4 的情况下，对矩阵帧内预测标志进行解码。

[0019] 在另一方面，在所述编码单元的大小不是 4×4 、 8×4 和 4×8 其中之一的情況下，对矩阵帧内预测标志进行解码。

[0020] 在另一方面，在所述编码单元的大小不是 4×4 、 8×4 、 4×8 和 8×8 其中之一的情況下，对矩阵帧内预测标志进行解码。

[0021] 在另一方面，在所述编码单元的大小不是 4×4 、 8×4 、 4×8 、 8×8 、 8×16 和 16×8 其中之一的情況下，对矩阵帧内预测标志进行解码。

[0022] 本公开的另一面提供一种从视频位流生成图像帧的编码树的预测块的方法，所述方法包括：通过从所述视频位流中解码矩阵帧内预测模式标志来确定编码单元的预测模式；在预测模式指示使用矩阵帧内预测模式的情况下，解码截断二数码字以确定矩阵帧内预测模式；以及通过将矩阵相乘应用于与所述预测块相邻的参考样本和根据解码的矩阵帧内预测模式所选择的矩阵来生成所述预测块。

[0023] 本公开的另一面提供一种非暂时性计算机可读介质，其上存储有计算机程序以实现从视频位流中解码图像帧的编码树的编码单元的方法，所述方法包括：将所述编码树的区域拆分成多个编码块，所述编码块中的各个编码块包括预测块；确定所述编码块中的各个编码块的预测块的矩阵帧内预测标志，各个矩阵帧内预测标志指示针对所述编码块之一的预测块是否使用矩阵帧内预测，所述确定 (i) 在所述区域满足阈值的情况下基于所述区域的面积，或者 (ii) 在所述区域的面积不满足所述阈值的情况下基于所述区域的预算；针对根据所确定的标志被确定为使用矩阵帧内预测的各个预测块，从存储器读取矩阵系

数;以及使用预测块对所述编码单元进行解码,所述预测块是使用各个预测块的参考样本和所述矩阵系数针对所述区域中的各个编码单元生成的。

[0024] 本公开的另一方面提供一种视频解码器,其被配置为:从视频位流接收图像帧的编码树的编码单元;将所述编码树的区域拆分成多个编码块,所述编码块中的各个编码块包括预测块;确定所述编码块中的各个编码块的预测块的矩阵帧内预测标志,各个矩阵帧内预测标志指示针对所述编码块之一的预测块是否使用矩阵帧内预测,所述确定(i)在所述区域满足阈值的情况下基于所述区域的面积,或者(ii)在所述区域的面积不满足所述阈值的情况下基于所述区域的预算;针对根据所确定的标志被确定为使用矩阵帧内预测的各个预测块,从存储器读取矩阵系数;以及使用预测块对所述编码单元进行解码,所述预测块是使用各个预测块的参考样本和所述矩阵系数针对所述区域中的各个编码单元生成的。

[0025] 本公开的另一方面提供一种系统,包括:存储器;以及处理器,其中所述处理器被配置为执行存储在所述存储器上的代码,以实现从视频位流中解码图像帧的编码树的编码单元的方法,所述方法包括:将所述编码树的区域拆分成多个编码块,所述编码块中的各个编码块包括预测块;确定所述编码块中的各个编码块的预测块的矩阵帧内预测标志,各个矩阵帧内预测标志指示针对所述编码块之一的预测块是否使用矩阵帧内预测,所述确定(i)在所述区域满足阈值的情况下基于所述区域的面积,或者(ii)在所述区域的面积不满足所述阈值的情况下基于所述区域的预算;针对根据所确定的标志被确定为使用矩阵帧内预测的各个预测块,从存储器读取矩阵系数;以及使用预测块对所述编码单元进行解码,所述预测块是使用各个预测块的参考样本和所述矩阵系数针对所述区域中的各个编码单元生成的。

[0026] 还公开了其它方面。

附图说明

[0027] 现在将参考以下附图和附录描述本发明的至少一个实施例,其中:

[0028] 图1是示出视频编码和解码系统的示意性框图;

[0029] 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;

[0030] 图3是示出视频编码器的功能模块的示意框图;

[0031] 图4是示出视频解码器的功能模块的示意框图;

[0032] 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;

[0033] 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;

[0034] 图7A和7B示出编码树单元(CTU)向多个编码单元(CU)的示例分割;

[0035] 图8示出矩阵帧内预测模块的操作的示例;

[0036] 图9A和9B示出具有应用了用于矩阵帧内预测模式的存储器访问带宽的区域的示例CTU;

[0037] 图10示出与图9A和9B的示例相对应的编码树;

[0038] 图11示出用于将图像帧的编码单元编码到视频位流中的方法,编码单元包括使用

矩阵帧内预测的编码单元；

[0039] 图12示出如图11中使用的用于确定跨越编码树单元的多个编码单元的矩阵帧内预测的存储器访问所用的预算的方法；

[0040] 图13示出如图11中使用的用于确定用于将图像帧的一部分编码在视频位流中的编码单元的方法；

[0041] 图14示出如图11中使用的用于将图像帧的编码单元编码到视频位流的方法；

[0042] 图15示出用于从视频位流解码图像帧的编码树的方法；

[0043] 图16示出如图15中使用的用于从视频位流解码图像帧的编码单元的方法；

[0044] 图17示出如图11中使用的用于确定用于将图像帧的一部分编码到视频位流中的编码单元的替代方法；

[0045] 图18示出如图11中使用的用于将图像帧的编码单元编码到视频位流的替代方法；
以及

[0046] 图19示出如图15中所使用的用于从视频位流解码图像帧的编码单元的替代方法。

具体实施方式

[0047] 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。

[0048] 如上所述,帧内预测的计算复杂度是限制性的,特别是对于矩阵帧内预测(MIP)。虽然MIP可以在最小化误差方面提供有效的解决方案,但是应用MIP的计算复杂度特别影响最差情况(例如仅由 4×4 块组成的帧)下的块处理速率。块处理速率需要足以支持目标应用的分辨率和帧频。仅考虑亮度通道,每秒120帧的“8K”分辨率帧(7680×4320)需要每秒处理 248.8×10^6 个 4×4 块。即使在整個帧或视频序列中没有实现最差情况,也需要处理达到最差情况的局部化区域,而不延迟完全解码帧的递送以在显示器上呈现。复杂度的一个方面涉及提取根据矩阵帧内预测(MIP)模式选择的矩阵系数所需的存储器带宽,其可能在沒有约束的情况下在块与块之间变化。

[0049] 图1是示出视频编码和解码系统100的功能模块的示意性框图。系统100可以利用对MIP模式的应用的约束来建立用于选择或读取矩阵系数的最差情况存储器带宽,以允许实际实现和/或与由MIP模式所实现的编码优点相称。

[0050] 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)、以及在一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。

[0051] 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。

[0052] 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的非暂时性存储装置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。例如,编码视频数据可以在需要时经由广域网(WAN)供应给顾客以用于视频流式传输应用。

[0053] 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。解码后的帧数据135具有与帧数据113相同的色度格式。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。

[0054] 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。

[0055] 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

(Ethernet™) 电路卡、蓝牙 (Bluetooth™) 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。

[0056] 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 Disc™))、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 可以体现在计算机系统 200 中。

[0057] 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 Mac™ 或相似的计算机系统。

[0058] 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。特别地,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 (参考图 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。

[0059] 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。

[0060] 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。

[0061] 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。

[0062] 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。

[0063] 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。

[0064] 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ROM 249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,BIOS 251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。

[0065] 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存储器冲突的情况下执行的充足存储器。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。

[0066] 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。

[0067] 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。

[0068] 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。

[0069] 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。

[0070] 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:

[0071] 提取操作,用于从存储器位置228、229、230提取或读取指令231;

[0072] 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

[0073] 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。

[0074] 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。

[0075] 要说明的图11至19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、ALU 240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。

[0076] 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310-392,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。

[0077] 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。帧数据113可以处于任何色度格式,例如4:0:0、4:2:0、4:2:2或4:4:4色度格式。块分区器310首先将帧数据113分割成CTU,

CTU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,CTU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分割成与亮度编码树或色度编码树相对应的一个或多个CB。CB具有各种大小并且可以包括正方形和非正方形长宽比这两者的。参考图11至19进一步描述块分区器310的操作。然而,在VVC标准中,CB、CU、PU和TU总是具有2的幂的边长。因而,从块分区器310输出当前CB(表示为312),从而根据对CTU的一个或多个块的迭代、根据CTU的亮度编码树和色度编码树而前进。以下参考图5和6来进一步说明用于将CTU分区为CB的选项。虽然操作大体上是在逐个CTU的基础上描述,但视频编码器114和视频解码器134可以在较小大小的区域上操作以减少存储器消耗。例如,各个CTU可以被划分成更小的区域,被称为大小为 64×64 的‘虚拟管线数据单元’(VPDU)。VPDU形成更适于硬件架构中的流水线处理的数据粒度,其中,与对完整CTU进行操作相比,存储器占用空间的减少减小了硅面积并且因此减小了成本。

[0078] 从帧数据113的第一次分割得到的CTU可以是按光栅扫描顺序扫描的,并可以被组成一个或多个“条带(slice)”。条带可以是“帧内”(或“I”)条带。帧内条带(I条带)指示条带中的每个CU都是帧内预测的。可选地,条带可以是单预测或双预测的(分别为“P”或“B”条带),分别指示条带中的单预测和双预测的附加可用性。

[0079] 对于各CTU,视频编码器114在两个阶段中操作。在第一阶段(称为“搜索”阶段),块分区器310测试编码树的各种潜在配置。编码树的各个潜在配置具有关联的“候选”CB。第一阶段涉及测试各种候选CB,以选择提供相对高压缩效率和相对低失真的CB。该测试通常涉及拉格朗日优化,由此基于速率(编码成本)和失真(关于输入帧数据113的误差)的加权组合来评价候选CB。选择“最佳”候选CB(具有最低评价速率/失真的CB)以用于随后编码在位流115中。候选CB的评价中包括如下选项:将CB用于给定区域,或者根据各种拆分选项来拆分该区域并利用其它CB来编码各个较小所得区域或更进一步拆分区。结果,在搜索阶段中选择CB和编码树本身这两者。

[0080] 视频编码器114针对各CB(例如,CB 312)产生由箭头320指示的预测块(PB)。PB 320是关联的CB 312的内容的预测。减法器模块322产生PB 320和CB 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PB 320和CB 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指示的变换块(TB)。通常从多个可能候选CB之一(例如,基于所评价的成本或失真)选择PB 320和关联的TB 336。

[0081] 候选编码块(CB)是针对关联的PB和所得到的残差从视频编码器114可用的预测模式其中之一得到的CB。TB 336是差324的量化和变换表示。当与视频解码器114中的预测PB组合时,TB 336以位流中的附加信令为代价来减少解码CB和原始CB 312之间的差。

[0082] 因而,各候选编码块(CB)(即预测块(PB)与变换块(TB)的组合)具有关联的编码成本(或“速率”)和关联的差(或“失真”)。速率通常是以位为单位测量的。CB的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。可以通过模式选择器386使用差324确定从各候选PB得到的估计,以确定预测模式387。预测模式387指示针对当前CB使用帧内预测、帧间预测或矩阵帧内预测(MIP)的决策。预测模式387被确定为可能模式中的所选择的模式,可能模式包括具有相关联的运动矢量的帧内预测(包括矩阵帧内预测、DC、平面和角度帧内预测)或帧间预测。预测模式387通常通过最小化失真度量来选择,该失真度量源自与利用Lambda值来缩放相关联的速率的结果进行求和的由各个候选模式产生的失真

的拉格朗日优化。当矩阵帧内预测处于使用中时,矩阵帧内预测模式(由箭头388表示)也被确定为指示若干可用矩阵帧内预测模式中的哪一个用于当前CB。如参考图11至14以及在替代实现图17和18中所描述的,用于决定块(特别是相对小的块)的MIP模式的使用和选择的搜索可被约束,以与不受约束的搜索相比实现用于提取矩阵系数的较低的最差情况存储器带宽。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。

[0083] 确定率失真方面的最佳模式通常是使用拉格朗日优化的变形来实现的。选择矩阵帧内预测模式388通常涉及确定从应用特定矩阵帧内预测模式得到的残差数据的编码成本。可以通过使用“绝对变换差和”(SATD)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。在使用相对简单的变换的一些实现中,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。在具有单调相关的估计成本的实现中,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。例如,可以从可用于残差数据的编码的其它模式决策得到非单调性。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。

[0084] 其它模式决策包括跳过正向变换的能力(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。对于“跳过的变换”,即使不进行变换本身,仍会对残差系数进行编码。

[0085] 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CB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。最低成本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附加模式。在包括可应用于给定块和更小同位置块的模式的情况下,候选的选择的处理隐含地也是用于确定CTU向CB的最佳层级分解的处理。

[0086] 在视频编码器114的操作的第二阶段(称为“编码”阶段)中,在视频编码器114中对所选择的亮度编码树和所选择的色度编码树(以及因而对各个所选择的CB)的迭代。如这里进一步描述的,在迭代中,将CB编码在位流115中。

[0087] 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具有关联预测(或“可能”或“最大概率”)值和关联概率(被称为“上下文”)。当要编码的实际bin与预测值匹配时,对“最大概率符号”(MPS)进行编码。对最大概率符号进行编码在位流115中的消耗位方面相对便宜,包括总计小于一个离散位的成本。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。

[0088] 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。可以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。每次对上下文编码bin进行编码时,以反映出新bin值的方式来更新针对该bin(如果有)选择的上下文。如此,二进制算术编码方案被认为是自适应的。

[0089] 视频编码器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。因而,各个bin占据位流115中的一位。不存在上下文节省了存储器并降低了复杂度,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作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。

[0090] 熵编码器338使用上下文编码和旁路编码的bin的组合来对预测模式387(并且,如果用于当前CB则是矩阵帧内预测模式388)进行编码。大小为 4×4 的块具有35个可能的矩阵帧内预测模式,否则大小不超过 8×8 的块(即, 4×8 、 8×4 和 8×8)具有19个可能的矩阵帧内预测模式。对于其它块大小,存在11种可能的矩阵帧内预测模式。通常,在视频编码器114中生成“最大概率矩阵模式”(MPM)的列表。最大概率模式的列表通常是诸如三个模式等的固定长度。最大概率模式的列表可以包括在与当前块相邻的较早块中遇到的模式。例如,如果在当前块的上方或左侧的块使用MIP模式,则对应的模式作为当前块的MPM存在。如果当前块的上方或左侧的块使用角度帧内预测,则当前块的MPM列表可以填充有经由将角度帧内预测模式映射到MIP模式的查找表所导出的MIP模式。另外,使用常规帧内预测(DC、平面或角度)预测的后续CU的MPM列表可以包括利用从MIP模式到候选常规帧内预测模式的映射表从使用MIP模式编码的CU所导出的候选帧内预测模式。上下文编码的bin对指示帧内预测模式是否是最大概率模式之一的标志进行编码。如果帧内预测模式388是最大概率模式其中之一,则使用旁路编码bin对进一步信令进行编码。例如,编码后的进一步信令使用截断一元bin串来指示哪个最大概率模式与矩阵帧内预测模式388相对应。否则,将帧内预测模式388编码为“剩余模式”。编码为剩余模式使用诸如固定长度代码等的替代句法,也使用旁路编码bin来编码,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。

[0091] 一些布置可以避免针对(使用MIP模式编码的)当前CU和可以使用常规帧内预测模式编码的后续CU这两者的MPM列表构建的复杂度,并且可以包括从任何MIP模式编码的相邻块导出的候选帧内预测模式。当MPM列表构建被省略时,使用截断二进码进行MIP模式的二值化以表示MIP模式。对于各个MIP模式,截断二进码得到相对均匀的编码成本,而对于添加到MPM列表中的各个MIP,MPM列表具有低编码成本。与MPM列表上未包含的MIP模式相比,从MIP模式选择获得的统计没有示出针对选择MPM列表上包含的MIP模式的强偏置,从而指示省略MPM列表的生成不会降低压缩效率。对于 4×4 块,35个可能的MIP模式可以使用5位或6位码来编码,其中5位用于MIP模式0-28并且6位用于MIP模式29-34。对于 4×8 、 8×4 和 8×8

块,19个可能的MIP模式可以使用4位或5位码来编码,其中4位用于MIP模式0-12并且5位用于MIP模式13-18。对于其它大小的块,11个可能的MIP模式可以使用3位或4位码来编码,其中3位用于MIP模式0-4并且4位用于MIP模式5-10。

[0092] 多路复用器模块384根据选自各候选CB的所测试的预测模式中的所确定的最佳预测模式387来输出PB 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。

[0093] 在帧间预测中,使用来自按位流中的编码帧的顺序在当前帧之前的一个或两个帧的样本来产生块的预测。此外,对于帧间预测,单个编码树通常用于亮度通道和色度通道这两者。位流中的编码帧的顺序可能不同于当捕获或显示时的帧的顺序。当一个帧用于预测时,该块被称为“单预测”并且具有两个关联的运动矢量。当两个帧用于预测时,该块被称为“双预测”并且具有两个关联的运动矢量。对于P条带,各CU可以被帧内预测或单预测。对于B条带,各CU可以被帧内预测、单预测或双预测。通常使用“图片组”结构来编码帧,从而实现帧的时间层级结构。帧的时间层级结构允许帧参考按显示帧的顺序的先前和后续的图片。图像按确保满足解码各帧的依赖关系所必需的顺序来编码。

[0094] 根据运动矢量和参考图片索引来选择样本。运动矢量和参考图片索引适用于所有颜色通道,因而主要就对PU而非PB的操作来描述帧间预测。在各类别内(即,帧内和帧间预测),可以应用不同技术来生成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和生成处理生成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索引(其指示将使用来自参考帧列表的哪些参考帧加上参考帧各自的空间平移),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。

[0095] 在确定并选择了PB 320、并在减法器322处从原始样本块中减去了PB 320的情况下,获得编码成本最低的残差(表示为324)并对该残差进行有损压缩。有损压缩处理包括变换、量化和熵编码的步骤。正向主变换模块326对差324应用正向变换,从而将差324从空间域转换到频率域并且产生由箭头328表示的主变换系数。主变换系数328被传递到正向二次变换模块330以通过进行不可分离二次变换(NSST)操作来产生由箭头332表示的变换系数。正向主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。正向主变换模块326通常使用II型离散余弦变换(DCT-2),但VII型离散余弦变换(DST-7)和VIII型离散余弦变换(DCT-8)也可能是可用的,例如水平方向上针对不超过16个样本的块宽度和垂直方向上针对不超过16个样本的块高度。通过首先对块的各行应用一维变换以产生中间结果然后对中间结果的各列应用一维变换以产生最终结果来进行各组行和列的变换。模块330的正向二次变换通常是不可分离变换,其仅适用于帧内预测CU的残差,但也可能被绕过。正向二次变换对16个样本(排列为主变换系数328的左上 4×4 子块)或64个样本(排列为左上 8×8 系数,排列为主变换系数328的四个 4×4 子块)进行操作。

[0096] 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TB是恒定的,因而得到针对TB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TB的大小的缩放矩阵中的相应条目的组合

导出的。缩放矩阵可以具有小于TB的大小的尺寸,并且当应用于TB时,最近邻方法被用于从大小小于TB大小的缩放矩阵为各个残差系数提供缩放值。残差系数336被供给至熵编码器338以编码在位流115中。通常,根据扫描模式对各TB的残差系数以及TU的至少一个有效残差系数进行扫描以生成值的有序列表。扫描模式通常将TB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TB的大小。另外,预测模式387、矩阵帧内预测模式(如果使用)388和相应块分区也编码在位流115中。

[0097] 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通过逆二次变换模块344以产生由箭头346表示的中间逆变换系数。中间逆变换系数346被传递到逆主变换模块348以产生TU的由箭头350表示的残差样本。逆二次变换模块344进行的逆变换类型与由正向二次变换模块330进行的正向变换类型相对应。逆主变换模块348进行的逆变换类型与由主变换模块326进行的主变换类型相对应。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)。

[0098] 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PB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。样本块366由模块364使用诸如DC、平面或角度帧内预测等、但不使用矩阵帧内预测的技术来生成。

[0099] 如果模式选择器386选择用于当前CB的矩阵帧内预测,则矩阵帧内预测模式388用于从系数存储器392中选择(读取)矩阵系数363。矩阵系数363被传递到矩阵帧内预测模块390。矩阵帧内预测模块390使用矩阵系数363和参考样本358进行矩阵乘法,以产生矩阵帧内预测块393。多路复用器384输出矩阵帧内预测块393作为PB 320。系数存储器392具有可用于提供矩阵系数393的有限带宽。具体地,不同矩阵帧内预测模式可用于各个连续块,从而建立最差情况存储器带宽要求。如关于图11至14的操作所描述的,模式选择器386可操作以在减小系数存储器392的最差情况存储器带宽的约束下针对块选择MIP模式。减小系数存储器392的最差情况存储器带宽降低了复杂度(例如,存储器的硬件区域),而不会导致视频编码器114的编码效率的成比例降级。由于无约束搜索中的MIP模式选择的统计通常不会触发施加到模式选择器386中的MIP模式选择的约束,因此在编码效率没有成比例降级的情况下实现了降低的复杂度。因此,在没有编码性能的相称损失的情况下减小了最差情况存储器带宽。

[0100] 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

配类别在样本级别应用偏移来工作。

[0101] 从环内滤波器模块368输出由箭头370表示的滤波样本。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。由于所需的大存储器消耗,因此帧缓冲器372通常不使用片上存储器来存储。如此,对帧缓冲器372的访问在存储器带宽方面是昂贵的。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。

[0102] 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CB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,PB 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)相比可以进行简化滤波处理,以实现降低的计算复杂度。当视频编码器114针对CU选择帧间预测时,运动矢量378被编码在位流115中。

[0103] 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)。

[0104] 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)视频解码流水线的示例,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。

[0105] 位流133被输入到熵解码器模块420。熵解码器模块420通过解码“bin”序列从位流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使用算术解码引擎来将各个句法元素解码为一个或多个bin的序列。各个bin可以使用一个或多个“上下文”,其中上下文描述用于为bin编码“一”和“零”值的概率水平。在多个上下文可用于给定bin的情况下,进行“上下文建模”或“上下文选择”步骤以选择可用上下文之一来解码bin。对bin进行解码的处理形成了一个顺序反馈循环。反馈循环中的操作数量优选地被最小化以使得熵解码器420能够实现以bin/秒为单位的高吞吐量。上下文建模取决于在选择上下文时视频解码器134已知的位流的其它特性,即当前bin之前的特性。例如,可以基于编码树中当前CU的四叉树深度来选择上下文。依赖性优选地基于在解码bin之前熟知的特性,或者在不需要长的顺序处理的情况下确定。

[0106] 编码树的四叉树深度是容易知道的上下文建模的依赖性的示例。帧内预测模式(特别是矩阵帧内预测模式)是确定起来相对困难或计算密集的上下文建模和二值化的依

赖性的示例。矩阵帧内预测模式被编码为“最大概率模式” (MPM) 列表的索引或“剩余模式”列表的索引,其中根据所解码的上下文编码标志在MPM和剩余模式之间进行选择。其它帧内预测模式编码为到‘最大概率模式’ (MPM) 的列表的索引或到‘剩余模式’的列表的索引,其中根据所解码的帧内亮度MPM标志在MPM与剩余模式之间进行选择。当MPM用于编码矩阵帧内预测模式时,具有范围0至2的截断一元bin串从MPM列表选择MPM之一。当使用剩余模式时,解码固定长度码字以选择要使用剩余(非MPM)模式中的哪个。根据块大小,可用MIP模式的数量是35、19和11之一。因此,对于长度3的MPM列表,剩余模式的数量分别是32、16或8。剩余模式可分别使用长度为5、4或3的固定长度码字来有效地表示。确定最大概率模式和剩余模式这两者需要大量的操作并且包括对相邻块的帧内预测模式的依赖。例如,相邻块可以是当前块上方和左侧的块。如果相邻块使用角度帧内预测,则可以进行表查找以将角度帧内预测模式映射到矩阵帧内预测模式,以用于填充MPM列表。可替代地,布置可以使用截断二进制码字来针对各个情况(即模式的数量是35、19还是11)编码MIP模式。熵解码器模块420应用算术编码算法,例如“上下文自适应二进制算术编码”(CABAC),以从位流133解码句法元素。所解码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等的模式选择信息(由箭头458表示)。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CB的分区。参数用于通常与来自先前解码的CB的样本数据组合生成PB。

[0107] 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进行逆量化(或“缩放”),以根据量化参数创建重建中间变换系数(由箭头432表示)。重建中间变换系数432被传递到逆二次变换模块436,在该二次变换模块处应用二次变换或无操作(旁路)。逆二次变换模块436产生重建变换系数440。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,并且根据将缩放因子排列成矩阵。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432。

[0108] 重建变换系数440被传递到逆主变换模块444。模块444将系数从频率域变换回到空间域。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的块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加到表示为452的解码PB,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。

[0109] 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CB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。使用诸如DC、平面或角度帧内预测等而非矩阵帧内预测的模式来生成样本块480。

[0110] 当在位流133中指示CB的预测模式使用帧内预测(而不是矩阵帧内预测)时,帧内预测样本480经由复用器模块484形成解码的PB 452。帧内预测产生样本的预测块(PB),即,

使用同一颜色分量中的“相邻样本”导出的一个颜色分量中的块。相邻样本是与当前块相邻的样本,并且由于在块解码顺序中处于前面因此已经被重建。在亮度和色度块并置的情况下,亮度和色度块可以使用不同的帧内预测模式。然而,两个色度通道各自共享相同的帧内预测模式。帧内预测分为三种类型。“DC帧内预测”涉及利用表示相邻样本的平均值的单个值填充PB。“平面帧内预测”涉及利用与平面相对应的样本填充PB,其中,DC偏移以及垂直和水平梯度源自相邻样本。“角度帧内预测”涉及利用以特定方向(或“角度”)在PB上滤波和传播的相邻样本填充PB。在VVC中,支持65个角度,其中矩形块能够利用正方形块不可用的附加角度来产生总共87个角度。第四种类型的帧内预测可用于色度PB,由此根据“交叉分量线性模型”(CCLM)模式从并置的亮度重建样本生成PB。三种不同的CCLM模式是可用的,各种模式使用从相邻亮度和色度样本导出的不同模型。然后使用导出模型来从并置的亮度样本生成色度PB的样本块。

[0111] 当在位流133中指示CB的预测模式是矩阵帧内预测时,矩阵帧内预测模式458被解码并且被供给到系数存储器486和矩阵帧内预测模块482。从系数存储器486读取用于所选矩阵帧内预测模式的矩阵系数481且将其传递到矩阵帧内预测模块482。矩阵系数的选择涉及来自系数存储器486的存储器读取操作,其中通过对于给定块大小选择MIP模式的频率来建立存储器访问的最差情况存储器带宽限制,如参考图8和9所描述的。

[0112] 当在位流133中指示CB的预测模式是帧间预测时,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中选择和滤波样本块498,来产生表示为438的帧间预测样本的块。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PB 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DBF、ALF和SAO滤波操作。通常,运动矢量被应用于亮度和色度通道这两者,但用于子样本插值亮度和色度通道的滤波处理是不同的。

[0113] 图5是示出通用视频编码的树结构中的区域向一个或多个子区域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利用以根据如通过拉格朗日优化所确定的编码数将各CTU分割成一个或多个CU或CB。

[0114] 尽管集合500仅示出将正方形区域分割成其它可能非正方形子区域,但应当理解,图500正示出潜在分割、而并未要求包含区域为正方形。如果包含区域为非正方形,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。一旦区域未被进一步拆分,即,在编码树的叶节点处,CU占据该区域。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”。

[0115] 将区域子分割成子区域的处理必须在所得到的子区域达到最小CU大小时终止。除了将CU约束为禁止块区域小于例如16个样本的预定最小大小之外,将CU约束为具有为四的最小宽度或高度。就宽度和高度或者就宽度或高度这两而言其它最小值也是可能的。子分割处理也可以在分解的最深层之前终止,从而得到大于最小CU大小的CU。有可能不发生拆分,从而得到占据整个CTU的单个CU。占据整个CTU的单个CU是最大可用编码单元大小。由于使用经子采样的色度格式(诸如4:2:0等),视频编码器114和视频解码器134的布置可以比亮度通道中更早地终止色度通道中的区域的拆分。

[0116] 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,其中各节点是形成一个CU的叶节点、或者包含向更小区域的进一步拆分的非叶节点。在编码树的各个叶节点处,针对各个颜色通道存在一个编码块。针对亮度和色度这两者在相同的深度处终止的拆分得到三个并置CB。针对亮度比针对色度在更深的深度处终止的拆分得到与色度通道的CB并置的多个亮度CB。

[0117] 如图5所示,四叉树拆分512将包含区域分割成四个大小相等的区域。与HEVC相比,通用视频编码(VVC)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大小相等的区域。分割沿着包含块内的水平边界(514)或垂直边界(516)。

[0118] 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“QTBT”。“树的根包括零个或多个四叉树拆分(树的“QT”部分)。一旦QT部分终止,可以发生零个或多个二元或三元拆分(“多树”或树的“MT”部分),最终在树的叶节点处的CB或CU中结束。在树描述所有颜色通道的情况下,树的叶节点为CU。在树描述亮度通道或色度通道的情况下,树的叶节点为CB。

[0119] 与仅支持四叉树、因而仅支持正方形块的HEVC相比,QTBT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(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,在所描述的布置中,约束可以单独应用于色度通道的块。拆分选项对色度通道的约束的应用可能针对亮度vs色度(例如,在帧数据采用4:2:0色度格式或4:2:2色度格式时)得到不同的最小块大小。各拆分产生具有相对于包含区域不变的、二等分的或四等分的边尺寸的子区域。然后,由于CTU大小为2的幂,因此所有CU的边尺寸也为2的幂。

[0120] 图6是示出在通用视频编码中使用的QTBT(或“编码树”)结构的数据流600的示意图。将QTBT结构用于各CTU以定义CTU向一个或多个CU的分割。各CTU的QTBT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。

[0121] 从层级结构的顶层开始、即在CTU处,首先进行零个或多个四叉树分割。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。

[0122] 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。

[0123] 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。

[0124] 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。

[0125] 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。

[0126] 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。

[0127] 图7A和7B提供CTU 710向多个CU或CB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。

[0128] 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的扫描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二、三或四个新节点的列表,直到到达叶节点(CU)为止。

[0129] 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,利用视频编码器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TB。

[0130] 作为将CTU拆分成CU的结果,小的大小的CU(诸如 4×4 、 4×8 或 8×4)看起来在帧中空间上相邻。此外,借助于编码树的分层Z顺序扫描,在时间上相邻地处理小的大小的CU。特别地, 8×8 区域的四叉树拆分得到四个 4×4 CU的序列, 8×4 或 4×8 区域的二元拆分得到一对 4×4 CU,并且 8×8 区域的二元拆分得到一对 4×8 区域或一对 8×4 区域。所得到的各个区域(大小为 4×8 或 8×4)可以形成CU或可以进一步拆分成 4×4 CU。在64个样本的区域中应用的三元拆分得到大小为16、32和16(4×4 、 4×8 或 8×4 和 4×4)的三个区域。在128个样本的区域中应用的三元拆分得到大小为32、64和32个样本(例如 4×8 、 8×8 和 4×8)的三个区域。小的块(例如 4×4 、 4×8 、 8×4)在空间上和通过CTU的Z顺序扫描这两者中一起被看到,因

为它们是拆分大小诸如为64或128等的区域的结果。

[0131] 图8示出提供分别使用 8×8 块的视频编码器114和视频解码器134的矩阵帧内预测模块390和482的操作细节的数据流800。在图8的示例中,仅在亮度通道中进行矩阵帧内预测,并且因此不需要考虑由使用色度格式(诸如4:2:0或4:2:2等)得到的不同块大小。模块390和482分别输出矩阵预测样本块,诸如393或483等。当模式选择器386针对CU选择矩阵帧内预测时,块393用作视频编码器114中的PB 320。当通过从位流133解码CU的预测模式来指示使用矩阵帧内预测时,块483用作PB 483。

[0132] 模块390和482的操作涉及三个步骤:1. 平均化,2. 矩阵相乘和偏移(偏置)相加,以及3. 双线性插值。仅当CU大小大于 4×4 时,进行双线性插值。平均化步骤按照如下操作。在数据流800处接收参考样本802(例如,358或464),并将其分配为上方样本822和左侧样本820。当亮度CB的宽度和高度大于4时,将上方样本822划分为四个集合(图8的示例 8×8 块中的四对)。四个集合中各自的内容的值被平均化以产生四个经滤波的上方样本826。类似地,左侧样本820被划分为四个集合(图8的示例 8×8 块中的四对)。四个集合中各自的内容的值被平均化以产生四个经滤波的左侧样本824。因此,总共八个经滤波样本作为矩阵乘法模块828的输入。

[0133] 当块宽度和高度等于四(而不是如图8所示的8)时,将四个上方样本822分割成两对,对各个对进行平均化以产生两个经滤波上方样本826。类似地,将四个左侧样本820分割成两对,对各个对进行平均化以产生两个经滤波左侧样本824,总共四个经滤波样本作为矩阵乘法模块828的输入。

[0134] 在第二步(矩阵相乘)中,根据矩阵帧内预测模式(388或458)选择接收到的矩阵系数(363或481),还将其输入到矩阵乘法模块828。此外,矩阵乘法模块的输入是偏移或偏置值的集合。偏移或偏置值被加到矩阵乘法的结果以引入任何期望的DC移位。矩阵系数和偏置值是预定的。换句话说,矩阵系数和偏置值是“离线”训练处理的结果,并且被视频编码器114和视频解码器134认为是恒定值。

[0135] 对于 4×4 CB,35个MIP模式可用,具有18个矩阵系数和偏置值的集合(集合A)。对于 4×8 、 8×4 和 8×8 CB,19个MIP模式可用,具有10个矩阵系数和偏置值的集合(集合B)。对于其它CB大小,11个MIP模式可用,具有6个矩阵系数和偏置值的集合(集合C)。矩阵系数和偏置值的给定集合可以用于两个MIP模式。一个MIP模式使用所提供的值,而另一个MIP模式使用所提供的值的转置。此外,在一种情况下,矩阵系数和偏置值的集合专用于单个MIP模式。三种情况各自适用于集合A-C中的每个。集合A-C的大小如下,还示出由362或481选择的用于生成一个PB(即,将MIP模式应用于一个CU)的字的数量,其针对集合A-C中的每一个给出如下:

[0136] 集合A:大小为 16×4 的18个矩阵,大小为16的18个偏移矢量。CB的所选值的大小为 $16 \times 4 + 16 = 80$ 个字。

[0137] 集合B:大小为 16×8 的10个矩阵,大小为16的10个偏移矢量。CB的所选值的大小为 $16 \times 8 + 16 = 144$ 个字。

[0138] 集合C:大小为 64×8 的6个矩阵,大小为64的10个偏移矢量。CB的所选值的大小为 $64 \times 8 + 64 = 576$ 个字。

[0139] 视频编码器114和视频解码器134以由帧大小和帧频确定的像素速率处理视频数

据。另外,亮度CB大小在宽度和高度上是四的倍数。因此,集合A-C的存储器带宽要求可以在对 4×4 亮度样本的访问方面来表达。当对各个CU的MIP模式的使用没有约束时,针对最差情况(即,相应集合的最小块大小)的集合A-C各自的访问密度如下:

[0140] 集合A: 4×4 CB需要每 4×4 样本区域80个字。

[0141] 集合B: 4×8 和 8×4 CB需要每 4×4 样本区域 $144 \div 2 = 72$ 个字, 8×8 CB需要每 4×4 样本区域 $144 \div 4 = 36$ 个字。

[0142] 集合C: 8×16 和 16×8 CB需要每 4×4 样本区域 $576 \div 8 = 72$ 个字, 8×32 、 16×16 、 32×8 CB需要每 4×4 样本区域 $576 \div 16 = 36$ 个字,较大的CB大小需要每 4×4 样本区域更少的字。

[0143] 如上面的访问密度所示,当对MIP模式的使用没有约束时,在最差情况下,所有CB可能使用MIP模式,并且编码树可能将各个CTU分解为上面用于说明最差情况系数存储器(486、392)带宽的小的CB大小。矩阵系数和偏置值的标称字大小是16位,但是更少的位(例如10位)可能也是足够的。在组中提取字是可能的实现选择。然而,存储器带宽负担仍然有些高。

[0144] MIP模式选择的统计示出:通常,在(如JVET公共测试条件文档JVET-N1010中所定义的)宽测试集上的20%的情况下,当考虑上方和左侧块时,针对相邻定位的CB选择MIP模式。因此,对MIP模式的应用频率的限制可以减轻系数存储器486和392所需的最差情况存储器带宽,而不会导致从MIP模式的可用性来看的压缩效率增益的相称降低。参考图11至16描述对MIP模式选择的约束。

[0145] 矩阵乘法模块828使用矩阵系数(363或481)和经滤波参考样本(即824和826)的集合来进行矩阵乘法。偏置值被加到矩阵乘法的输出,从而形成稀疏块830。稀疏块830被示出为部分填充393或483的阴影样本。使用双线性插值导出393或483样本的剩余部分,其贡献来自上方参考样本822和左侧经滤波参考样本824或者左侧参考样本820和上方经滤波样本822(如图8所示)。

[0146] 图9A示出示例CTU 900。CTU 900具有应用MIP模式的存储器访问带宽的区域。在VPDU大小的区域(即 128×128 CTU 900的 64×64 象限)内应用矩阵帧内预测。例如,预算适用的512个亮度样本区域包括(i) 32×16 区域912、(ii) 16×32 区域914及(iii) 64×8 区域916。区域912被进一步分解为各种CU,即图9B中的912a-912h。在图9B的示例中,CU 912a的大小为 16×4 且因此属于集合C,CU 912d的大小为 8×8 且因此如同912b、912c及912e(全部大小均为 8×4)那样属于集合B。CU 912f和912g各自为 4×4 且因此属于集合A。如关于图11至16所描述,512个亮度样本的区域面积可在所描述的布置中用作阈值以确定是否针对MIP模式应用约束。使用512个样本的区域面积适合作为约束,因为上述集合A至C通常可以由区域的拆分产生。然而,阈值可以涉及如下所述的不同面积。

[0147] 图10示出与图9A的示例CTU 900相对应的编码树1000。区域912对应于编码树中的节点,在该节点处,针对MIP矩阵系数读取,建立所包含的CU的预算,区域914和916也是如此。为了便于参考,未示出在区域中分解成一个或多个CU。如下文参考图11至16所描述的,对于被配置为使用MIP模式的各个CU,从区域(即,912、914、916)预算中扣除根据CU大小映射到集合A-C之一的预算量。

[0148] 图11示出了用于将图像帧的编码单元编码在视频位流115中的方法1100。方法

1100可以由诸如配置的FPGA、ASIC或ASSP等的装置来体现。另外,方法1100可以在处理器205的执行下由视频编码器114进行。因此,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。进行方法1100以将图像帧作为编码单元序列编码到位流115中,这对矩阵帧内预测的使用应用了约束。因此,与(如没有约束的情况下的)最差情况潜在存储器带宽相比,方法1100的操作使得对矩阵帧内预测所需的存储器带宽进行限制。方法1100开始于将帧分割成CTU的步骤1110。

[0149] 在将帧分割成CTU的步骤1110处,块分区器310在处理器205的执行下将帧数据113的当前帧分割成CTU阵列。开始进行随着从分割得到的CTU的编码。处理器中的控制从步骤1110前进到确定编码树的步骤1120。

[0150] 在确定编码树的步骤1120处,视频编码器114在处理器205的执行下结合确定编码单元的步骤1130的操作来测试各种拆分选项(如参考图5-7所述),以得到CTU的编码树。参考图12描述步骤1120的操作。处理器205中的控制从步骤1120前进到确定编码单元的预测模式的步骤1130。

[0151] 在确定编码单元的步骤1130处,视频编码器114在处理器205的执行下确定要在将所选择的编码单元编码在位流115中时使用的预测模式。可以根据扫描图案来选择编码单元。参考图13进一步描述步骤1130的操作。一旦针对编码单元选择了预测模式,处理器205中的控制就从步骤1130前进到对编码单元进行编码的步骤1140。在选择编码单元的预测模式时,随着编码单元本身由编码树的拆分的层级产生,还选择要达到给定编码单元的拆分的特定组合,从而确定编码树。

[0152] 在对编码单元进行编码的步骤1140处,熵编码器338在处理器205的执行下将在步骤1130处确定的编码单元编码在位流115中。在步骤1140处,由熵编码器338在处理器205的执行下,使用“拆分标志”和用以指示所选择的拆分的其它句法元素将所确定的编码树有效地编码在位流115中,如图5和6所示。

[0153] 参考图14进一步描述步骤1140处的操作。处理器205中的控制从步骤1140前进到最后编码单元测试步骤1150。

[0154] 在最后编码单元测试步骤1150处,处理器205测试当前编码单元是否是步骤1120的编码树中的最后编码单元。如果当前编码单元是步骤1120的编码树中的最后一个(步骤1150为“是”),则处理器205中的控制前进到最后CTU测试步骤1160。如果当前编码单元不是步骤1120的编码树中的最后一个(步骤1150为“否”),则使用用于确定和编码的扫描图案来选择步骤1120的编码树中的下一个编码单元,并且处理器205中的控制前进到步骤1130。因此,对于从在步骤1120确定的编码树得到的各个CU,进行步骤1130。

[0155] 在最后CTU测试步骤1160处,处理器205测试当前CTU是否是条带或帧中的最后CTU。如果不是(步骤1160为“否”),则视频编码器114前进到帧中的下一个CTU,并且处理器205中的控制从步骤1160前进回到步骤1120以继续处理帧中的剩余CTU。如果CTU是帧或条带中的最后一个,则步骤1160返回“是”,并且方法1100终止。作为方法1100的操作的结果,整个图像帧作为CTU序列被编码在位流中。

[0156] 对视频序列中的各个图像帧执行方法1100。方法1100可以在逐个CTU的基础上确定CU。换句话说,CTU的CU可以首先在一遍或流水线阶段中确定,接着是用于编码在位流115中的第二阶段。方法1100还可以以更精细的粒度(例如,在逐个VPDU的基础上)确定CU,以使

得由于与CTU相比更小的VPDU面积而减少存储器消耗。

[0157] 图12示出了如在步骤1120实现的用于确定CTU的编码树的方法1200。方法1200执行以接收CTU并生成候选拆分和候选编码单元以用于评价和最终选择为要编码在位流115中的拆分和编码单元。特别地,方法1200在编码树中的特定节点或区域处建立MIP模式存储器带宽的预算,该特定节点或区域约束在节点下方(即,空间区域内)的编码单元中的MIP模式的应用。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的执行下由视频编码器114进行。因此,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。针对候选编码树中的各个节点(区域)调用方法1200。方法1200开始于区域面积测试步骤1210。

[0158] 在区域面积测试步骤1210处,视频编码器114在处理器205的执行下根据可用拆分选项(使用关于图5描述的拆分选项)测试编码树中的候选节点所占用的面积。在所描述的布置的上下文中,区域涉及可在拆分停止之前进一步拆分的编码树中的节点,从而形成CU。区域不需要与当前编码单元的直接父区域相对应,即,设置MIP模式存储器访问带宽预算的父区域可以是编码树中的当前节点上方的多个节点。例如,512个亮度样本的区域可以拆分成许多小编码单元,例如具有诸如 4×4 、 4×8 、 8×4 等的大小。随着编码器搜索的进行,测试由候选拆分产生的不同候选节点。在各个候选节点内,测试用于所得编码单元的各种预测模式。当由区域占据的亮度样本中的面积不满足阈值时,区域面积测试评价为真,并且处理器205中的控制从步骤1210前进到设置预算步骤1220。当由区域占据的亮度样本中的面积满足阈值时,区域面积测试评价为假,并且处理器205中的控制从步骤1210前进到生成拆分的步骤1230。在一个实现中,如图11至16的示例中所描述的,阈值为512个样本,且当区域具有大于512个样本的面积时满足阈值。可以如下所述使用其它阈值。阈值通常是基于编码器114和解码器134的所需性能而预先确定的。

[0159] 在设置预算步骤1220处,视频编码器114在处理器205的执行下针对与编码树中的当前节点的面积相对应的区域设置预算。该预算可用于当前区域内的任何子节点(子区域),该子节点(子区域)是通过将当前区域拆分成更小的区域而产生的。对于小于或等于512个亮度样本的所有区域,当正在评价编码单元预测模式时,可以在MIP模式的应用中考虑该预算。512个亮度样本的面积可经受例如三元拆分,从而得到面积128、256和128个亮度样本的编码单元。面积128个亮度样本的编码单元可以具有 8×16 或 16×8 的尺寸,在这种情况下,可能达到最差情况存储器带宽限制,从而导致禁止将MIP模式用于区域中的其它编码单元,包括由区域的进一步细分得到的那些编码单元。当512个亮度样本的面积被二元拆分成256个亮度样本面积的两个编码单元(例如, 16×16)时,没有达到最差情况存储器带宽限制,并且对所得CU的MIP模式的使用没有约束。然而,将各个256样本面积区域进一步分割成较小CU(即,大小为 16×8 、 8×16 、 8×8 、 8×4 、 4×8 及 4×4 的CU)可能导致由于较早所得CU消耗预算而约束针对较晚所得CU的MIP模式的使用。

[0160] 预算可以表示为针对每 4×4 块面积的最大字读取,因为 4×4 是最小亮度编码单元的大小,并且所有其它编码单元是该大小的整数倍,因此所有区域也是该大小的倍数。预算可以是针对 4×4 块的40个字读取,其针对512个亮度样本面积对应于 $(512 \div (4 \times 4)) * 40 = 1280$ 个字读取,作为512个亮度样本区域中的许可读取操作的最大数量。每当MIP模式用于CU时,从预算中扣除所需的读取数量,并且当预算不足以进一步使用MIP模式时,禁止其它

CU使用MIP模式,如关于图13和14所述。以这种方式,较大大小的CU可以使用MIP模式,例如,面积512个样本的单个CU或面积256个样本的一对CU可以使用MIP模式而不从约束引入任何限制。当编码树将512个样本的面积分解成较大数量的较小CU时,预算的消耗可能禁止区域中的其它CU使用MIP模式。具有每 4×4 40个字的预算的512个样本面积得到 $512 \div (4 \times 4) * 40 = 1280$ 个字的区域的总预算。在将512个样本面积分割成大小为 8×16 或 16×8 的四个CU时,两个CU使用MIP模式消耗针对每CU的576个字或1152个字。128个字读取的剩余预算不足以使区域中的其它两个CU使用MIP模式。处理器205中的控制从步骤1220前进到生成拆分的步骤1230。

[0161] 在生成拆分的步骤1230中,块分区器310在处理器205的执行下针对编码树中的当前节点生成候选拆分集合。拆分如图5所示,并且相关联的句法元素在图6中示出并在图7A和图7B中例示。在生成“无拆分”(即,510)的情况下,稍后生成编码单元。在生成其它类型的拆分(即,512-520)的情况下,根据各个拆分生成编码树中的附加节点。当在方法1200的稍后遍历上生成“无拆分”情况时,重复方法1200以生成附加节点的编码单元。按顺序针对CTU的各个节点重复方法1200。因此,确定CTU的所有拆分。此外,拆分内的拆分的这种生成的递归性质使得在对最小CU大小的约束和深度递归约束(其可以限制使用二元、三元和四叉树拆分的递归次数)内搜索给定CTU中的所有可能的编码单元。处理器205中的控制从步骤1230前进到生成CU的步骤1240。

[0162] 在生成CU的步骤1240处,块分区器310在处理器205的执行下针对在生成拆分的步骤1230处生成“无拆分”的各个情况,生成候选CU。在步骤1240处,候选CU的预测模式有待确定,但是Z顺序扫描中在前的CU的预测模式是已知的,尽管最终模式尚未被选择。因此,相邻(就CU中的位置而言)参考样本可用于来自(可能来自与包含当前CU的拆分相同的拆分操作的)相邻CU、或来自不同父区域得到的相邻区域、或来自完全不同的CTU的帧内预测。步骤1240有效地将编码树的区域拆分成编码块,各个编码块包括如关于图3所述的预测块。方法1200在步骤1240处终止,其中处理器205中的控制返回到方法1100,在方法1100处确定所生成的CU的预测模式。

[0163] 图13示出如在步骤1130处实现的用于确定编码单元的方法1300。方法1300涉及确定如通过进行方法1200生成的编码单元的预测模式。预测模式包括帧内预测、帧间预测和MIP模式(根据在包含当前编码单元的区域使用MIP模式的预算)。在方法1200中建立的预算用于控制是否针对当前CU测试MIP模式。每当区域中的CU具有最差情况块的大小(如上文关于情况A-C所描述)时,就消耗预算的一部分,这取决于CU中的码字的数量。一旦消耗了预算,就不搜索该区域中的其它编码单元以寻找MIP模式的潜在使用,即,方法1300的调用在方法1300的较早调用尚未耗尽被设置为应用于编码树中的公共父节点的存储器访问带宽预算时仅进行MIP模式的测试。

[0164] 该区域不需要与当前编码单元的直接父区域相对应,即,设置了MIP模式存储器访问带宽预算的父区域可以是编码树中的当前节点上方的多个节点。例如,512个亮度样本的区域可以拆分成许多小编码单元,例如具有诸如 4×4 、 4×8 、 8×4 等的大小。对于各个CU,MIP模式的使用受512个亮度样本的区域的剩余预算约束。方法1300可以由模式选择器386来实现,或者部分地由模块3960来实现。方法1300可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,方法1300可以在处理器205的执行下由视频编码器114进行。因此,方法

1300可以存储在计算机可读存储介质上和/或存储器206中。方法1300开始于测试帧内预测模式的步骤1310。

[0165] 在测试帧内预测模式的步骤1310处,视频编码器114在处理器205的执行下测试“常规”帧内预测模式(即DC、平面和角度帧内预测模式),以潜在用于对当前编码单元进行编码。通常,进行拉格朗日优化以在CU的可用帧内预测模式中选择最佳帧内预测模式。还测试了二次变换(即330和344)的应用,包括变换跳过情况的不同类型的一次变换(DCT-2、DCT-8、DST-7)也是如此。处理器205中的控制从步骤1310前进到测试帧间预测模式的步骤1320。

[0166] 在测试帧间预测模式的步骤1320处,视频编码器114在处理器205的执行下测试各种运动矢量以生成帧间预测PU。当评价帧间预测的使用时,从一组候选运动矢量中选择运动矢量。根据搜索模式生成候选运动矢量。在对评价中的候选运动矢量的所提取的参考块进行失真测试时,考虑了编码树中的禁止色度拆分的应用。如果在色度中禁止拆分并且在亮度中允许拆分,则所得到的亮度CB可以使用帧间预测。仅对亮度通道应用运动补偿,因此失真计算考虑了亮度失真而不考虑色度失真。由于在色度拆分被禁止时在色度通道中不进行运动补偿,因此不考虑色度失真。对于色度,考虑了由所考虑的帧内预测模式和编码色度TB(如果有的话)引起的失真。在考虑亮度和色度这两者时,帧间预测搜索可以首先基于亮度失真选择运动矢量,然后通过考虑色度失真来“细化”运动矢量。细化通常考虑运动矢量值的小变化,诸如子像素位移等。特定运动矢量可以是通过‘合并模式’生成的运动矢量,由此从来自相邻CU的运动矢量导出当前CU的运动矢量。与相对于所选择的‘运动矢量预测子’应用的可能需要用信号通知‘运动矢量增量(delta)’的其它运动矢量相比,在位流句法中更紧凑地表达合并模式。运动矢量预测子通常从空间或时间上相邻的CU导出。对于帧内编码的条带(例如帧序列的第一帧),帧间预测不可用,并且不进行步骤1320。处理器205中的控制从步骤1320前进到预算内测试的步骤1330。

[0167] 在预算内测试的步骤1330处,视频编码器114在处理器205的执行下测试MIP模式存储器访问带宽预算是否适用于当前CU。在1330处执行的测试确定当前CU是否包含在512个亮度样本区域内,在步骤1220处为该512个亮度样本区域建立了MIP模式预算。如果当前CU大于512个亮度样本区域,则不存在适用的预算约束。因此,CU不受使用MIP模式时的进一步约束,并且处理器205中的控制前进到测试MIP模式的步骤1340(1330为“真”)。如果当前CU的大小等于或小于512个亮度样本,则将针对当前CU使用MIP模式所需的预算(如参考图8所描述)与该区域的剩余预算进行比较。如果没有足够的预算可用于将MIP模式应用于当前CU,则处理器205中的控制从步骤1330前进到选择模式的步骤1350(步骤1330为“假”)。如果有足够的预算可用于将MIP模式应用于当前CU(步骤1330为“真”),则处理器205中的控制从步骤1330前进到步骤1340。

[0168] 在测试MIP模式的步骤1340处,模式选择器386测试各种MIP模式,以在当前CU大小的可用MIP模式中确定用于预测该CU的最佳MIP模式。与在步骤1310处的测试帧内预测模式一样,可以进行拉格朗日优化以权衡失真与所测试的MIP模式及其相关联残差的编码成本。处理器205中的控制从步骤1340前进到选择模式的步骤1350。

[0169] 在选择模式的步骤1350处,模式选择器386在处理器205的执行下从由步骤1310、1320和1350得到的候选中选择CU的最终模式。方法1300在处理器205中的控制返回到方法

1100的状态下终止。

[0170] 图14示出如在步骤1140处实现的用于将CTU的编码树的编码单元编码在视频位流115中的方法1400。方法1400可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,方法1400可以在处理器205的执行下由视频编码器114(例如由熵编码器338)进行。因此,方法1400可以存储在计算机可读存储介质上和/或存储器206中。方法1400开始于编码预测模式的步骤1410。

[0171] 在编码预测模式的步骤1410处,熵编码器338在处理器205的执行下使用指示使用(如在步骤1350处确定并由预测模式387表示的)帧内预测(包括常规帧内预测模式使用或MIP模式使用两者)或帧间预测的上下文编码bin对标志进行编码。在步骤1410处对标志的编码不区分常规帧内预测和MIP模式。常规帧内预测和MIP帧内预测(当适用时)之间的区别在编码MIP模式标志的步骤1430处被编码。处理器205中的控制从步骤1410前进到预算测试的步骤1420。

[0172] 在预算测试的步骤1420处,视频编码器114在处理器205的执行下测试MIP模式存储器访问带宽预算是否适用于当前CU。该测试确定当前CU是否包含在512个亮度样本区域内,在步骤1220处为该512个亮度样本区域建立了MIP模式预算。如果当前CU包含在大于512个亮度样本区域的区域中,则不存在适用的预算约束且CU不受使用MIP模式时的进一步约束。处理器205中的控制前进到编码MIP模式标志的步骤1430(1420处为“真”)。

[0173] 如果当前CU包含在大小等于或小于512个亮度样本的区域中,则将如参考图8所描述的针对当前CU使用MIP模式所需的预算与该区域的剩余预算进行比较。如果没有足够的预算可用于将MIP模式应用于当前CU,则处理器205中的控制从步骤1420前进到编码TB的步骤1440(步骤1420为“假”)。然而,如果有足够的预算可用于将MIP模式应用于当前CU(1420处为“真”),则即使当前CU在小于或等于512个样本的区域中,处理器205中的控制也从步骤1420前进到步骤1430。步骤1420的操作对应于步骤1330的操作,并且因此,针对在方法1300中搜索了MIP模式的CU,在方法1400中仅对MIP标志进行编码。

[0174] 在编码MIP模式标志的步骤1430处,熵编码器338在处理器205的执行下将指示是否选择如在步骤1350处确定的MIP模式的上下文编码bin编码在位流115中。参考步骤1420描述用于对bin进行编码的上下文。在选择MIP模式的情况下,熵编码器338还将使用哪个特定MIP模式的选择编码在位流338中。可使用截断二数码字代替使用“最大概率模式”与剩余模式之间的选择来编码MIP模式。使用截断二数码字避免了导出最大概率模式的列表的必要性(包括从相邻角度帧内预测CU导出最大概率模式的潜在表查找)。处理器205中的控制从步骤1430前进到编码TB的步骤1440。

[0175] 在编码TB的步骤1440处,熵编码器338在处理器205的执行下将与当前CU相关联的TB的残差系数编码在位流中。通常,各个TB的标志用信号通知存在至少一个有效系数,根据从最后一个有效系数位置前进回到DC(左上)系数位置的扫描图案而逐一地编码系数。然后,方法1400终止,并且处理器205中的控制返回到方法1100。

[0176] 图15示出用于从视频位流133中解码图像帧的编码单元和变换块的方法1500。方法1500可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,方法1500可以在处理器205的执行下由视频解码器134进行。因此,方法1500可以存储在计算机可读存储介质上和/或存储器206中。方法1500开始于将帧分割成CTU的步骤1510。

[0177] 在将帧分割成CTU的步骤1510处,视频解码器134在处理器205的执行下将(要解码的)帧数据133的当前帧分割成CTU阵列。开始进行随着从分割所得到的CTU的解码。处理器205中的控制从步骤1510前进到对编码单元进行解码的步骤1520。

[0178] 在对编码单元进行解码的步骤1520处,熵解码器420在处理器205的执行下根据如参考图5-7所述的编码树从位流133中解码拆分标志。解码拆分标志允许步骤1520操作以即根据CTU的编码树确定CTU内的CU的大小和位置。方法1300的进展涉及在步骤1520上的迭代,从而得到在CTU的编码树上的遍历(各个CU被解码)。处理器205中的控制从步骤1520前进到对编码单元进行解码的步骤1530。

[0179] 在对编码单元进行解码的步骤1530处,熵解码器420在处理器205的执行下从位流133解码编码单元。步骤1530调用下文关于图16描述的方法1600来解码CU。处理器205中的控制从步骤1530前进到最后编码单元测试的步骤1540。

[0180] 在最后编码单元测试的步骤1540处,处理器205测试当前编码单元是否是如从在步骤1520处解码拆分标志所确定的CTU中的最后编码单元。如果当前编码单元是CTU中的最后一个(步骤1540为“是”),则处理器中的控制前进到最后CTU测试步骤1550。如果当前编码单元不是步骤1520的编码树中的最后一个(步骤1540为“否”),则选择步骤1520的编码树中的下一个编码单元进行解码,并且处理器205中的控制前进到步骤1520。

[0181] 在最后CTU测试步骤1550处,处理器205测试当前CTU是否是条带或帧中的最后CTU。如果当前CU不是最后一个(步骤1550为“否”),则视频解码器134前进到帧或条带中的下一个CTU,并且处理器205中的控制从步骤1550前进回到步骤1520以继续处理帧中的剩余CTU。如果CTU是帧或条带中的最后一个,则步骤1550返回“是”,并且方法1500终止。作为方法1500的结果,整个图像帧被解码为来自位流的CTU序列。

[0182] 图16示出如在步骤1530处实现的用于从视频位流133解码编码单元的方法1600。方法1600可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,方法1600可以在处理器205的执行下由视频解码器134进行。因此,方法1600可以存储在计算机可读存储介质上和/或存储器206中。方法1600开始于解码pred_mode标志步骤1602。

[0183] 在解码pred_mode标志步骤1602处,熵解码器420对上下文编码bin进行解码,以确定当前编码单元是使用帧间预测还是帧内预测(包括MIP模式)。处理器205中的控制从步骤1602继续到帧间预测测试步骤1604。执行步骤1604以从pred_mode标志确定是否使用帧间预测。如果当前编码单元使用帧间预测(帧间预测测试步骤1604为“真”),则处理器205中的控制从步骤1602继续到进行帧间预测步骤1606。在步骤1606处进行帧间预测,使得提取参考块(434)并进行滤波以产生PU,然后解码各个颜色通道的TB并将TB(1690)添加到PU以解码CU。处理器205中的控制从步骤1606继续到添加TB的步骤1690。

[0184] 如果当前编码单元使用帧内预测或MIP模式(步骤1604为“假”),则处理器205上的控制继续到MIP模式预算测试的步骤1610。执行MIP模式预算测试的步骤1610以根据预算确定MIP模式是否可用于当前CU。该预算对应于参考步骤1330或1420所描述的预算。在所描述的示例中,如果包含当前CU的区域大于面积阈值(512个亮度样本),则不存在适用的预算约束且MIP模式可用(步骤1610为“真”)。处理器205中的控制从步骤1610前进到解码MIP模式标志的步骤1620。

[0185] 如果在步骤1610处,包含当前CU的区域的面积等于或小于阈值(512个亮度样本),

则将如参考图8所描述的针对当前CU使用MIP模式所需的预算与该区域的剩余预算进行比较。如果有足够的预算可用于将MIP模式应用于当前CU,则MIP模式可用(步骤1610为“真”),并且方法1600继续到步骤1620。如果没有足够的预算可用,则MIP模式不可用于当前CU(MIP模式预算测试步骤1610为“假”),并且处理器205中的控制继续到解码帧内预测模式的步骤1612。

[0186] 在步骤1612处,使用常规帧内预测(DC,平面或角度),从解码帧内预测模式开始。处理器205中的控制从步骤1612继续到进行帧内预测的步骤1614,并且步骤1614操作以进行帧内预测。处理器205中的控制从步骤1614继续到添加TB的步骤1690。

[0187] 在步骤1620的执行中,通过熵解码器420使用单个上下文编码bin对MIP标志进行解码。bin使用一个上下文,即,不需要上下文选择,例如这取决于针对相邻CU使用MIP模式或块大小或可用于从多个可能上下文中选择用于解码单个bin的一个上下文的其它参数。处理器中的控制从步骤1620继续到MIP模式选择测试步骤1630。

[0188] 步骤1630操作以确定是否选择MIP模式。如果解码的MIP模式标志指示MIP模式未被使用(MIP模式选择测试步骤1630为“假”),则控制前进到步骤1612以使用常规帧内预测模式之一对块进行解码。如果解码的MIP模式标志指示MIP模式用于CU(步骤1630为“真”),则处理器205中的控制操作以继续到解码MIP模式的步骤1640。

[0189] 在步骤1640的执行中,从位流133解码MIP模式。处理器205中的控制从步骤1640继续到读取矩阵系数步骤1650。在步骤1650处,解码的MIP模式用于从矩阵系数存储器486读取矩阵系数481的集合。借助于通过使用预算在步骤1610处限制使用MIP模式的可用性,矩阵系数存储器486提供矩阵系数481所需的最大存储器带宽消耗减少。

[0190] 方法1600从步骤1650继续到滤波相邻样本的步骤1660。在步骤1660处,对相邻参考样本464进行滤波。方法1600从步骤1660继续到矩阵乘法步骤1670。步骤1600可以由模块482例如使用矩阵系数481和样本464来实现。类似于图8的示例828,在步骤1670处将经滤波的参考样本和矩阵系数481相乘。

[0191] 方法1600从步骤1670继续到插值PB的步骤1680。从步骤1670的执行所确定的稀疏块(即,830)用于利用步骤1680处的插值操作填充PB(即,483)。方法1600从插值PB的步骤1680继续到添加TB的步骤1690。解码的残差用于生成TB,该TB在步骤1690处被添加到PB 483以解码CU。使用在步骤1690处生成的预测块对编码单元进行解码。方法1600在步骤1690完成时终止。

[0192] 在进行方法1600时,视频解码器134以对用于提取矩阵系数的最差情况存储器带宽的约束来实现对矩阵帧内预测的支持。与实际上未使用限制的情况相比,在影响错误方面,所描述的约束不会过度地限制MIP模式的选择。不会过度限制选择的操作是由于通常仅20%的MIP模式编码的CU具有也使用MIP模式的左侧或上方邻居的所确定的统计可能性。存在比作用于建立存储器访问预算的粒度的区域(即,512个亮度样本)更大的区域上的边界和20%发现的测量结果中所包括的CTU上的边界。

[0193] 在图11至16的示例实现中,是否使用MIP模式是基于约束。例如在图12中,基于当前CU的区域的面积来实现约束,并且矩阵帧内预测标志是否被编码或解码取决于当前CU的区域的面积是否满足阈值面积。在所描述的实现中,(i)如果区域满足阈值,则矩阵帧内预测标志是否被编码取决于区域的面积,或者(ii)如关于步骤1330、1420和1610所描述的,如

果区域的面积不满足阈值,则矩阵帧内预测标志是否被编码取决于该区域的预算。因此,矩阵帧内预测标志是否被编码(或解码)至少取决于区域的面积是否满足阈值面积。

[0194] 在图11至14中描述的实现中,如关于步骤1420和1430所描述的,仅在满足预算测试时才将MIP模式标志编码在位流115中。相应地,在图16中解码位流时,仅在MIP模式预算测试步骤返回真时才解码MIP模式标志。换句话说,仅在使用矩阵帧内预测时才针对CU来解码矩阵帧内预测标志。

[0195] 在视频编码器114和视频解码器134的替代布置中,如参考图12和13所述,限制MIP模式的搜索。然而,在替代布置中,无论各个CU的预算消耗的状态如何,针对各个CU都将MIP模式的信令包括在位流中。换句话说,无论是否使用矩阵帧内预测,针对CU都解码矩阵帧内预测标志。在替代布置中,在视频编码器114中,省略步骤1420且控制前进到步骤1430(步骤1420处为“真”),如使用图14中的虚线所示的箭头1411所指示的。在替代布置中,在步骤1430的所有情况下都包括MIP模式标志,但是如果在步骤1350处没有选择MIP模式,则可以将其设置为零。同样,在视频解码器134中,省略步骤1610,并且控制前进到步骤1620(实际上,步骤1610总是返回“真”)。在步骤1620处解码MIP模式标志,并且如果在编码位流的步骤1350处没有选择MIP模式,则MIP模式标志为零。避免了在熵编码器338和熵解码器420中对是否编码各个CU的MIP标志进行解码的额外负担。尽管位流现在包括各个CU的MIP模式标志,但是如方法1200和1300中进行的搜索确保了用于从视频解码器134中的系数存储器486读取矩阵系数481的存储器带宽仍然受到限制,从而减少了用于应对MIP模式的这种(受约束的)最差情况使用的资源的必要供应。

[0196] 上文关于图12至17所描述的实现基于包含CU的区域的面积应用对MIP模式的使用的约束。在视频编码器114和视频解码器134的又一布置中,图12处实现的约束禁止了针对与最差情况存储器带宽相对应的块大小而非基于区域的面积而使用矩阵帧内预测。实际上,基于各个编码块的大小使用矩阵帧内预测(以及编码或解码的矩阵帧内预测标志)。 4×4 、 4×8 、 8×4 、 8×8 、 8×16 和 16×8 的块大小得到针对每 4×4 块80个字的最高访问密度(例如,如相对于 4×4 亮度样本块所测量的)。在先前的块大小集合中的某些块大小具有每 4×4 块72个字的最差情况,然而出于资源供应的目的,这可以在与每 4×4 块80个字的情况相同的类别中被对待。禁止最差情况块大小建立了每 4×4 块40个字的最差情况(某些块大小具有每 4×4 块36个字的最差情况)。

[0197] 图17示出如步骤1130实现的用于确定编码单元的方法1700。方法1700提供了用于约束涉及禁止针对与最差情况存储器带宽相对应的块大小使用矩阵帧内预测的实现的方法1300的替代方案。方法1700涉及确定如通过进行方法1200生成的编码单元的预测模式。预测模式包括帧内预测、帧间预测和MIP模式(根据包含当前编码单元的区域内的MIP模式的允许使用)。方法1700开始于测试帧内预测模式的步骤1710。

[0198] 在测试帧内预测模式的步骤1710处,视频编码器114在处理器205的执行下测试“常规”帧内预测模式(即,DC、平面和角度帧内预测模式)以潜在用于对当前编码单元进行编码,并以针对步骤1310所描述的方式操作。处理器205中的控制从步骤1710前进到测试帧间预测模式的步骤1720。

[0199] 在测试帧间预测模式的步骤1720处,视频编码器114在处理器205的执行下测试各种运动矢量以生成帧间预测的PU。当评价帧间预测的使用时,从候选运动矢量集合中选择

运动矢量。步骤1720以与步骤1320相同的方式操作。处理器205中的控制从步骤1720前进到MIP可允许大小测试的步骤1730。

[0200] 在MIP可允许大小测试的步骤1730处,视频编码器114在处理器205的执行下测试CU是否具有允许MIP模式的大小。在一些实现中,除了最差情况大小 4×4 、 4×8 、 8×4 、 8×8 、 8×16 和 16×8 之外,MIP模式可允许用于所有大小。如果当前CU是可允许的大小(1730为“真”),则处理器205中的控制从步骤1730前进到测试MIP模式的步骤1740。如果当前CU是禁止的大小(步骤1730为“假”),则处理器205中的控制从步骤1730前进到选择模式步骤1750。

[0201] 在测试MIP模式的步骤1740处,模式选择器386测试各种MIP模式,以在针对CU的大小可用的MIP模式中确定用于预测当前CU的最佳MIP模式。步骤1740以与步骤1340相同的方式操作,处理器205从步骤1740前进到选择模式的步骤1750。

[0202] 在选择模式的步骤1750处,模式选择器386在处理器205的执行下从由步骤1710、1720和1750得到的候选中选择CU的最终模式。方法1700在处理器205中的控制返回到方法1100的状态下终止。

[0203] 图18示出在约束涉及禁止针对与最差情况存储器带宽相对应的块大小使用矩阵帧内预测的实现中如在步骤1140处实现的用于将CTU的编码树的编码单元编码在视频位流115中的方法1800。方法1800可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,方法1800可以在处理器205的执行下由视频编码器114进行。因此,方法1800可以存储在计算机可读存储介质上和/或存储器206中。方法1800开始于编码预测模式的步骤1810。

[0204] 在编码预测模式的步骤1810处,熵编码器338在处理器205的执行下使用指示使用如在步骤1750处确定的帧内预测(包括常规帧内预测模式使用或MIP模式使用两者)或帧间预测的上下文编码bin对标志进行编码。处理器205中的控制从步骤1810前进到MIP可允许大小测试的步骤1820。

[0205] 在MIP可允许大小测试的步骤1820处,视频编码器114在处理器205的执行下测试当前CU是否是可允许的大小。可允许的大小和禁止的大小与方法1700的步骤1730的相同。如果当前CU是可允许的大小,则处理器205中的控制前进到编码MIP模式标志的步骤1830(1820处为“真”)。

[0206] 如果当前CU不是可允许的大小(步骤1820为“假”),则处理器205中的控制从步骤1820前进到编码TB的步骤1840。

[0207] 在编码MIP模式标志的步骤1830处,熵编码器338在处理器205的执行下将指示是否选择如在步骤1350处确定的MIP模式的上下文编码bin编码在位流115中。步骤1830以与步骤1430相同的方式操作。处理器205中的控制从步骤1830前进到编码TB的步骤1840。

[0208] 在编码TB的步骤1840处,熵编码器338在处理器205的执行下将与当前CU相关联的TB的残差系数编码在位流中。步骤1840以与步骤1440类似的方式操作。然后,方法1800终止,并且处理器205中的控制返回到方法1100。

[0209] 图19示出如在步骤1530处针对约束涉及最差情况块禁止MIP模式的实现所实现的用于从视频位流133中解码编码单元的方法1900。方法1900可以由诸如配置的FPGA、ASIC或ASSP等的设备来体现。另外,方法1900可以在处理器205的执行下由视频解码器134进行。因此,方法1900可以存储在计算机可读存储介质上和/或存储器206中。方法1900开始于解码pred_mode标志的步骤1902。

[0210] 在解码pred_mode标志的步骤1902处,熵解码器420解码上下文编码bin以确定当前编码单元是使用帧间预测还是帧内预测(包括MIP模式)。如果当前编码单元使用帧间预测(帧间预测测试的步骤1904为“真”),则处理器中的控制继续进行帧间预测的步骤1906。在步骤1606处进行帧间预测,使得提取参考块(434)并进行滤波以产生PU,然后解码各个颜色通道的TB并将TB(1690)添加到PU以解码CU。类似地,处理器205中的控制从步骤1906继续到添加TB的步骤1990。

[0211] 如果当前编码单元使用帧内预测或MIP模式(步骤1904为“假”),则处理器205的控制继续到MIP可允许大小测试的步骤1910。执行MIP可允许大小测试的步骤1910以确定MIP模式是否可允许用于当前CU大小。参考步骤1730或1820描述MIP可允许大小。如果当前CU是可允许大小(步骤1910为“真”),则处理器205中的控制从步骤1910前进到解码MIP模式标志步骤1920。如果在步骤1910处当前CU不是可允许大小(步骤1910为“假”),则处理器205中的控制继续到解码帧内预测模式的步骤1912。

[0212] 在步骤1912处,对常规帧内预测(DC、平面或角度)进行解码,并且方法1900继续到解码帧内预测的步骤1914。步骤1912和1914以关于步骤1612和1614所描述的方式操作。处理器205中的控制从步骤1914继续到添加TB的步骤1990。

[0213] 在步骤1920的执行中,MIP标志被解码。处理器205中的控制从步骤1920继续到MIP模式选择测试的步骤1930。

[0214] 步骤1930操作以确定是否选择MIP模式。如果解码的MIP模式标志指示MIP模式未使用(MIP模式选择测试的步骤1930为“假”),则控制前进到步骤1912以使用常规帧内预测模式之一对块进行解码。如果解码的MIP模式标志指示针对CU使用MIP模式(步骤1930为“真”),则处理器中的控制操作以继续到解码MIP模式的步骤1940。

[0215] 在步骤1940的执行中,从位流133解码MIP模式。处理器205中的控制从步骤1940继续到读取矩阵系数的步骤1950。在步骤1950处,解码的MIP模式用于从矩阵系数存储器486读取矩阵系数481的集合。如在步骤1650处,针对被确定为使用MIP模式的各个预测块读取矩阵系数。

[0216] 方法1900从步骤1950继续到滤波相邻样本的步骤1960。在步骤1960处,对相邻参考样本464进行滤波。方法1900从步骤1960继续到矩阵乘法步骤1970。类似于图8的示例828,在步骤1970处将经滤波的参考样本和矩阵系数481相乘。

[0217] 方法1900从步骤1970继续到插值PB的步骤1980。从步骤1970的执行所确定的稀疏块(即,830)用于在步骤1980处利用插值操作填充PB(即,483)。方法1900从插值PB的步骤1980继续到添加TB的步骤1990。类似于步骤1690,解码的残差用于生成TB,该TB在步骤1990处被添加到PB 483以解码CU。方法1900在步骤1990完成时终止。

[0218] 对方法1300、1400和1600描述的修改也可以分别应用于方法1700、1800和1900。

[0219] 如在方法1700中,当块大小用作MIP模式可用性的标准时,不需要建立和更新存储器预算,即省略步骤1210和1220。对于某些块大小,去除MIP模式虽然比预算存储器访问更易于实现,但由于许多流行的块大小不存在MIP模式而导致较低的压缩性能。由于不存在 4×4 块意味着不需要存储“集合A”(18个矩阵系数和偏置值的集合)的矩阵系数,因此系数存储器(即,392、486)的大小减小。从 4×4 、 4×8 、 8×4 、 8×8 、 8×16 和 16×8 (“小块”)中去除MIP模式还具有这样的优点,即MIP模式的反馈环路(其是相对复杂的操作)不需要支持这些

小块。

[0220] 在又一布置中,对于(如上所列)小块的子集禁止使用MIP模式。例如,在步骤1730、1820和1910处,仅针对 4×4 块禁止MIP模式,但针对所有其它块大小都允许MIP模式,或者针对 4×4 、 4×8 和 8×4 块禁止MIP模式,但针对所有其它块大小都允许MIP模式。换句话说,不使用矩阵帧内预测,并且根据实现,如果编码单元的大小是所禁止的大小之一,则对矩阵帧内预测标志进行解码。与完全缺乏对MIP模式的使用的限制相比,最差情况的存储器带宽没有减少,但是由于排除了这些非常小的块大小,因此帧内重建反馈环路的严重性降低。 4×4 的去除还消除了从系数存储器(即392、486)存储与“集合A”相关联的矩阵系数的需要。可替代地,所禁止的块大小集合可以是 4×4 、 4×8 、 8×4 和 8×8 ,在这种情况下,在视频编码器114或视频解码器134中不存在集合A和B。去除集合A和B使得存储器消耗减少,因为不需要与集合A和B相关联的矩阵系数,代价是压缩性能降低。

[0221] 在又一布置中,存储器预算仍然低于每 4×4 亮度样本面积40个字,例如每 4×4 亮度样本面积20个或甚至10个字。预算是在较大区域大小(诸如在与1024或2048亮度样本的区域大小相对应的节点处等)上建立的。与上述布置一样,根据这些区域内较早CU的MIP模式的使用,受限内的较晚CU在存储器预算的可用性方面受到约束。存储器带宽的进一步减小是以较低的压缩效率为代价实现的。

[0222] 在又一布置中,在步骤1220、1330、1420和1610以64个亮度样本的区域大小建立存储器预算,以应用于大小为 4×4 、 4×8 和 8×4 的CU(“小CU存储器预算”)。以512的区域大小建立单独的存储器预算,但仅适用于大小超过 8×8 的CU,特别是 8×16 和 16×8 的CU(“较大的CU存储器预算”)。两个预算都以每 4×4 亮度样本面积40个字来设置。预算形成总矩阵存储器带宽的加法预算,因为小CU存储器预算仅适用于包含在64个亮度样本面积内的CU,而较大CU存储器预算仅适用于大于64个亮度样本的CU。

[0223] 尽管本文公开的布置以每 4×4 存储器区域的字来描述存储器带宽,但是应当理解,对存储器的访问可能以某种SIMD形式对字进行分组,以允许读取矩阵系数而不需要相关联存储器的过多时钟频率。然而,这种更宽的存储器本身成本高,并且矩阵系数可以与同一存储器中的其它数据共享,使得通过对MIP模式的使用的限制而减少访问争用。

[0224] 以区域为单位限制MIP模式的使用以限制最差情况存储器带宽可能引入朝向各个区域的较早(更靠近上方或左侧)的CU的偏置以使用MIP模式,而给定区域中较晚的CU不能使用MIP模式,因为该区域的可用预算被较早遇到的CU消耗。通常不会遇到这种偏置,因为选择MIP模式的CU的分布通常有些稀疏。

[0225] 所描述的布置实现了对何时可以使用MIP模式的约束,从而与允许在没有约束的情况下实现MIP模式相比降低了计算复杂度。由于在无约束搜索中的MIP模式选择的统计通常不会触发施加在模式选择器386中的MIP模式选择上的约束,因此在编码效率没有成比例降低的情况下实现了降低的复杂度。因此,在编码性能没有相称的损失的情况下减少了最差情况存储器带宽。

[0226] 产业上的可利用性

[0227] 所描述的布置适用于计算机和数据处理行业,并且特别地适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,从而实现高压缩效率。

[0228] 本文描述的布置允许残差编码和解码使用基于网格的状态机,该基于网格的状态

机根据系数奇偶性来更新并且为系数选择上下文和量化器。所描述的布置允许在不会由于状态更新的顺序性质而强加过度延迟的情况下实现基于网格的状态机。

[0229] 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。

[0230] 相关专利申请的引用

[0231] 本申请根据35U.S.C. §119要求于2019年6月24日提交的澳大利亚专利申请2019204437的优先权益,这里出于所有目的将该专利申请的全部内容通过引用合并于此。

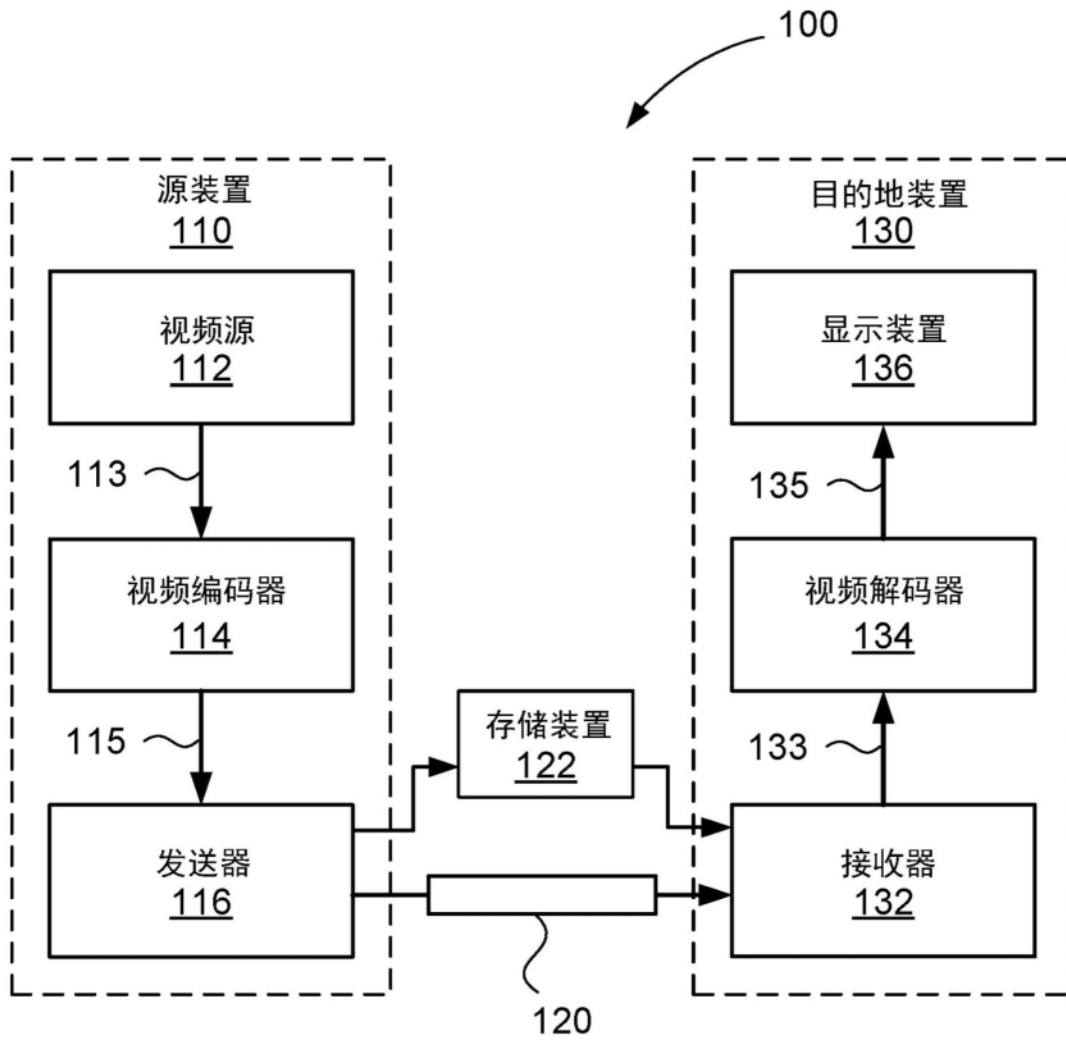


图1

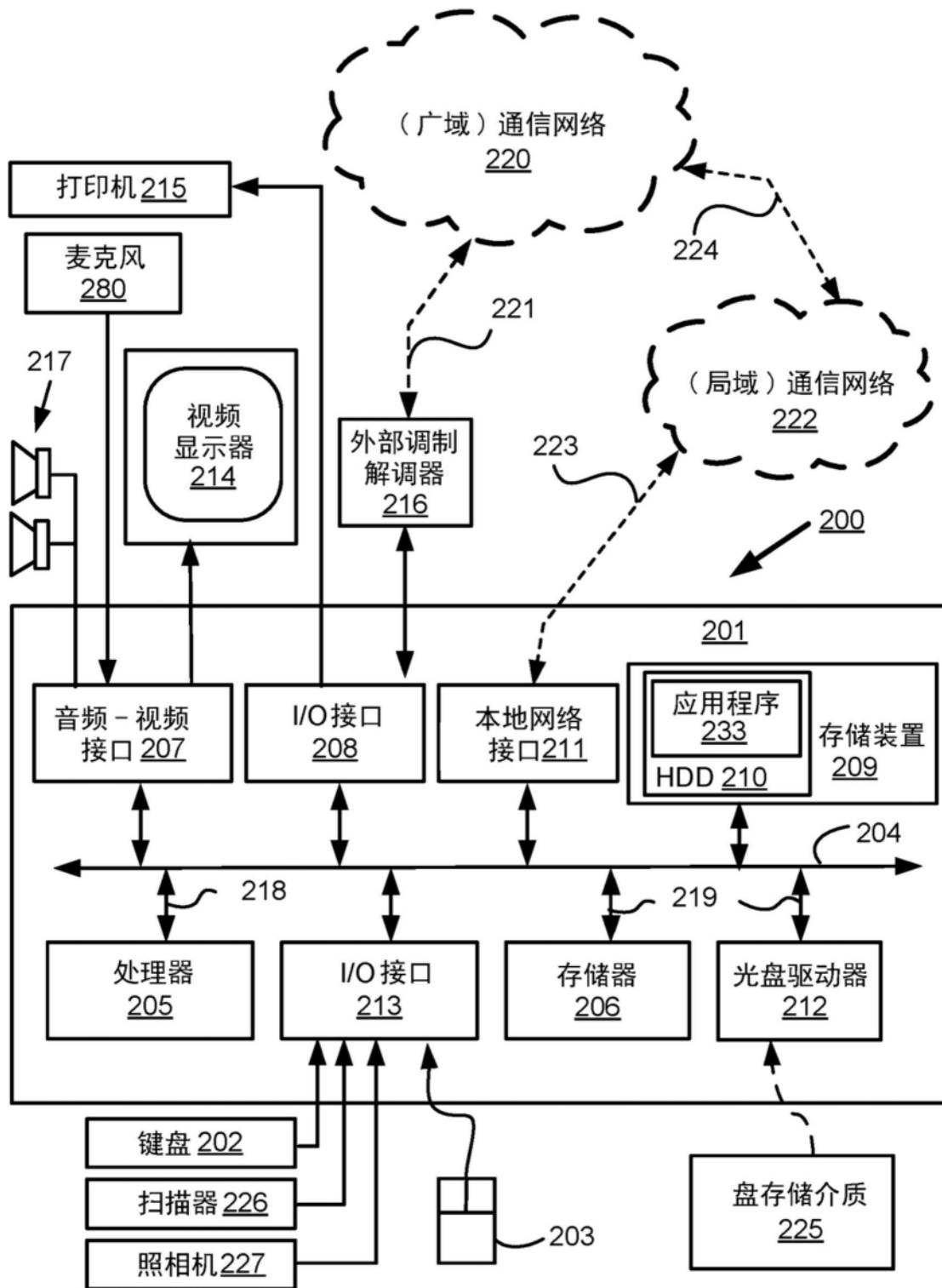


图2A

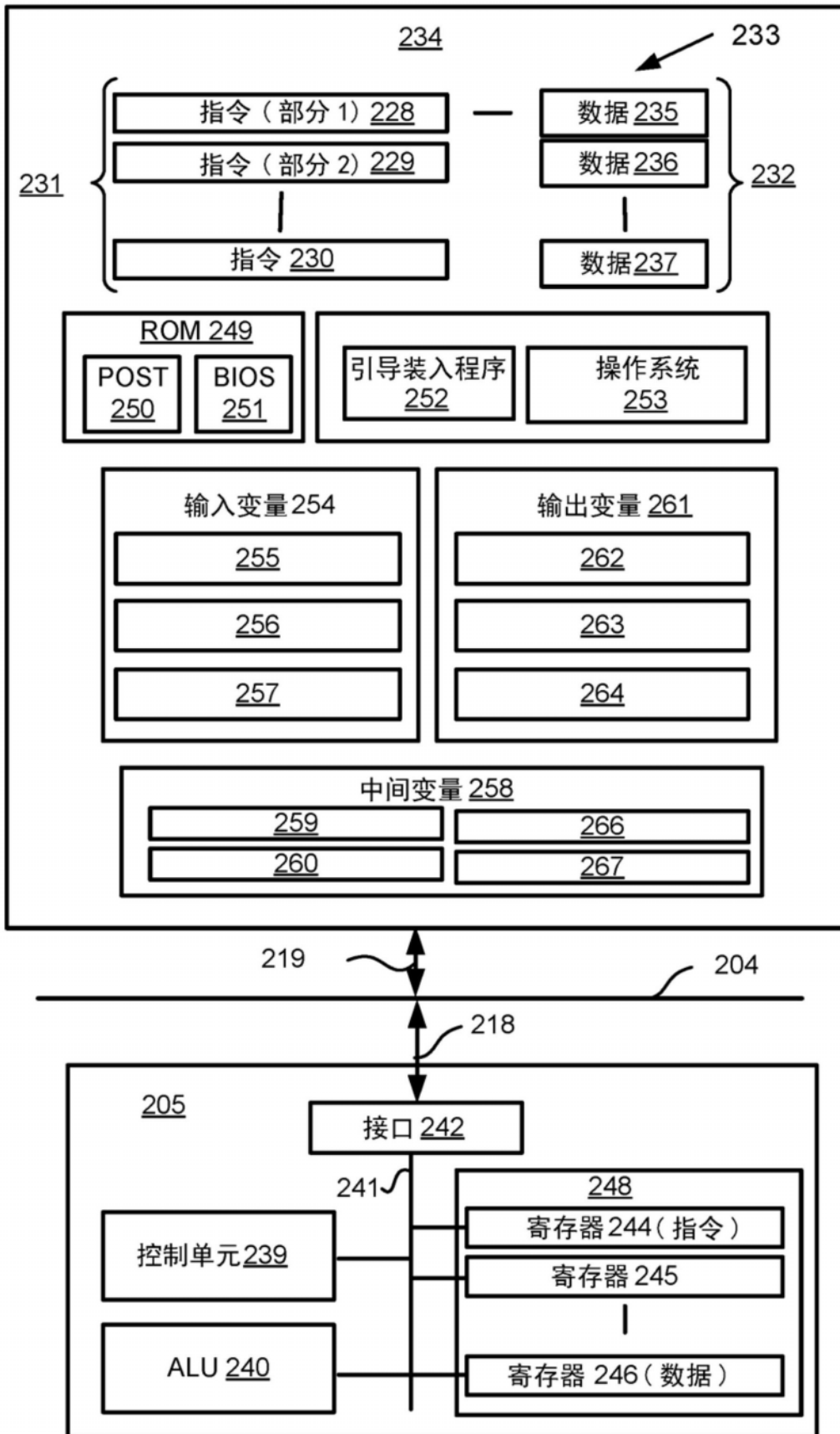


图2B

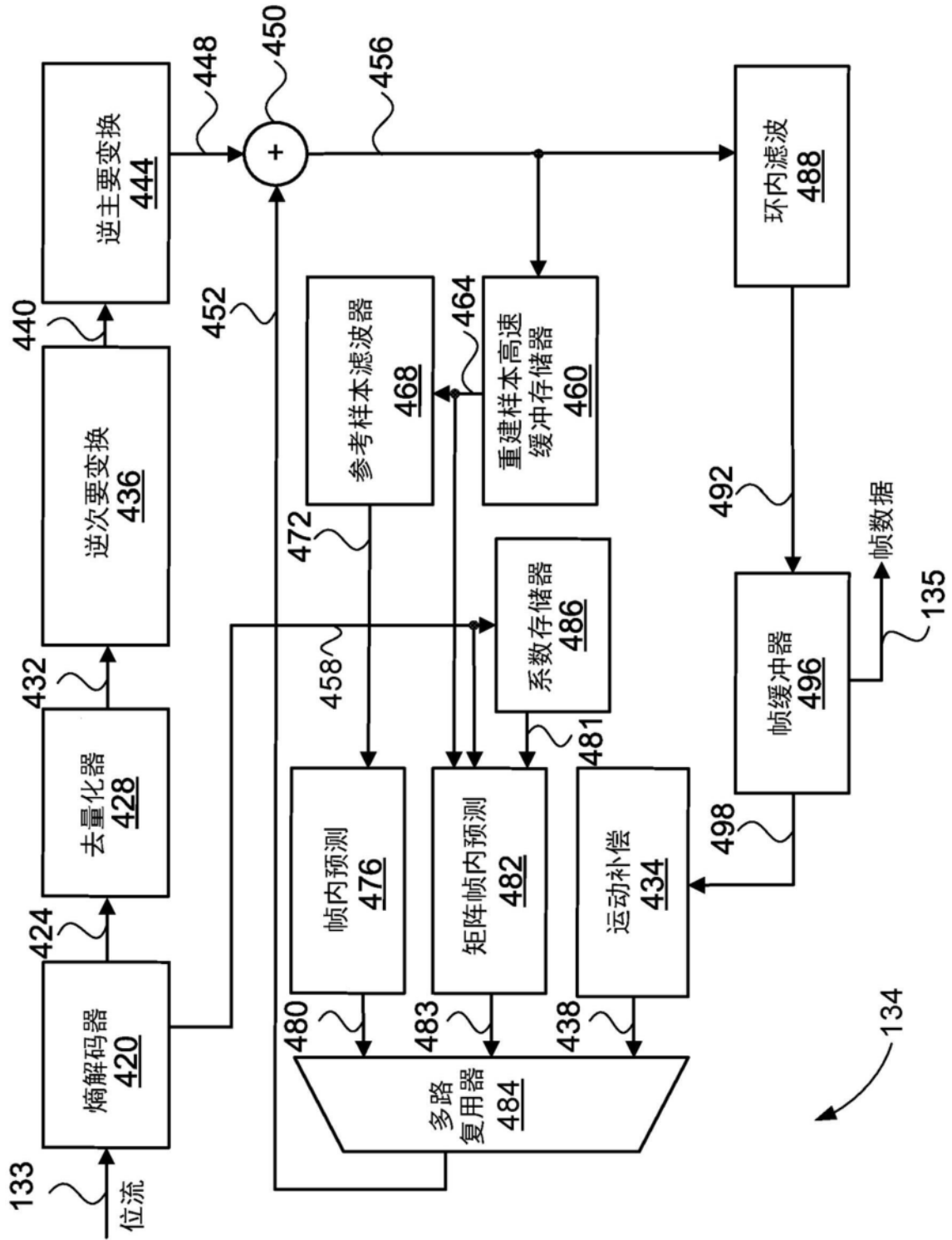


图4

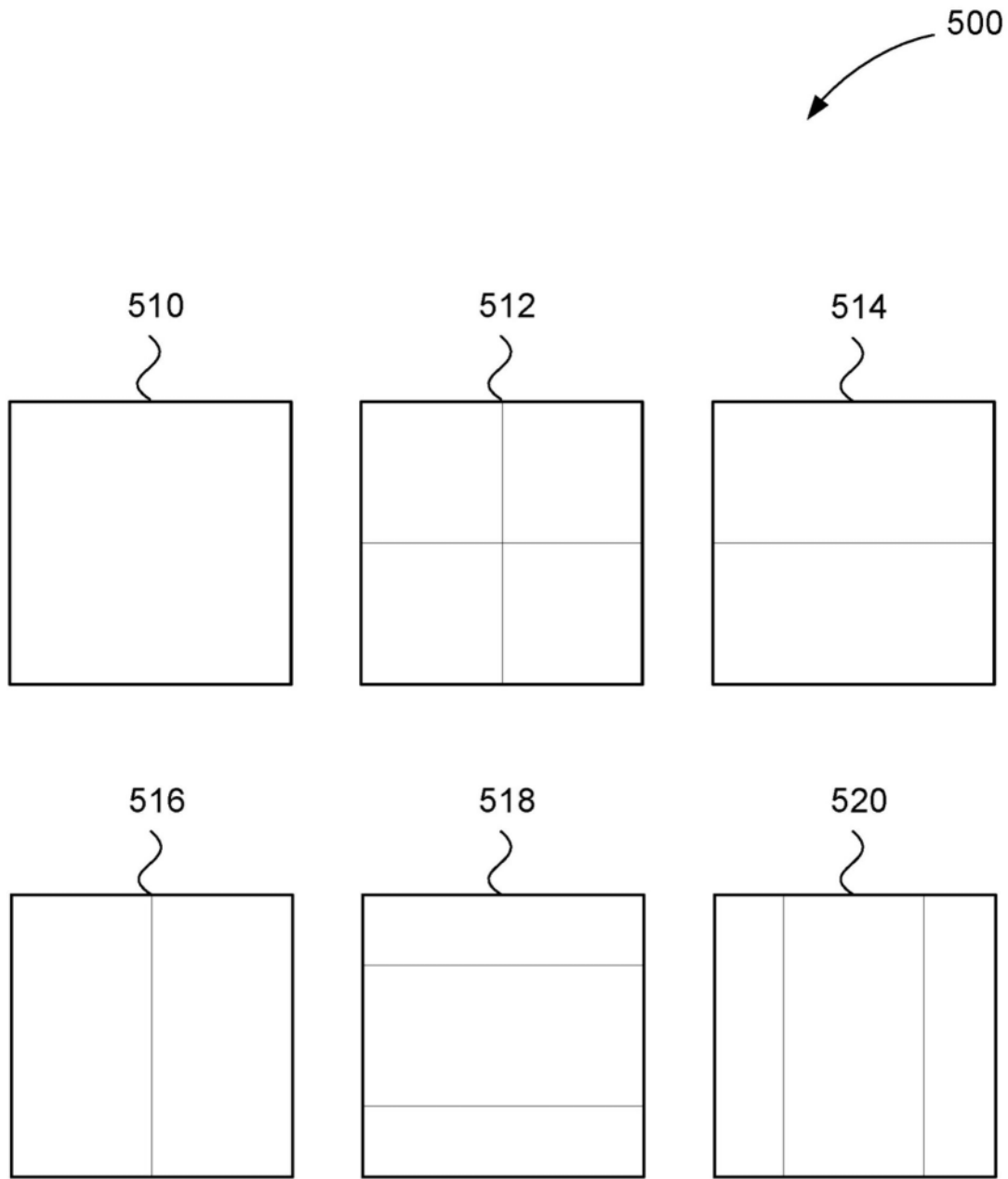


图5

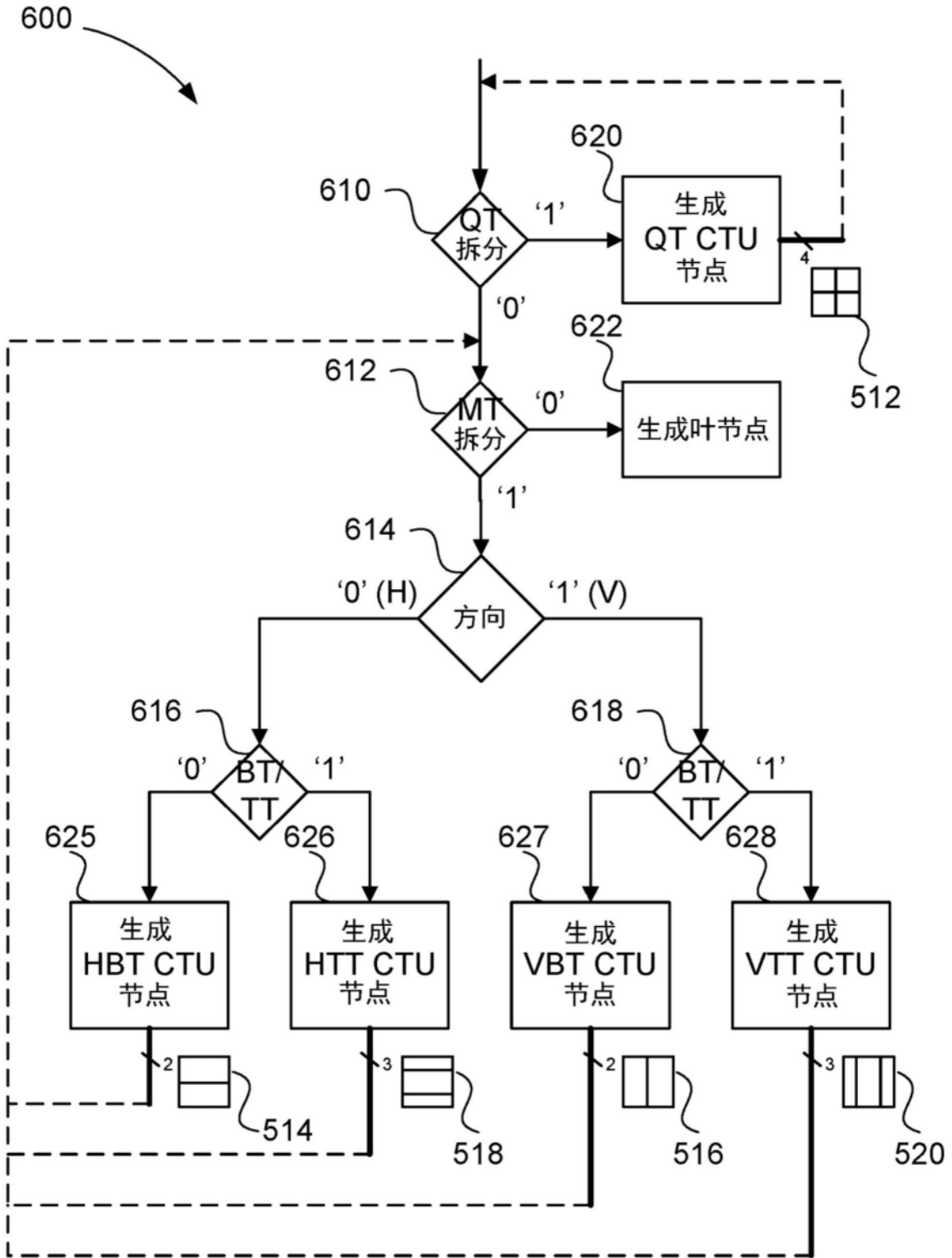


图6

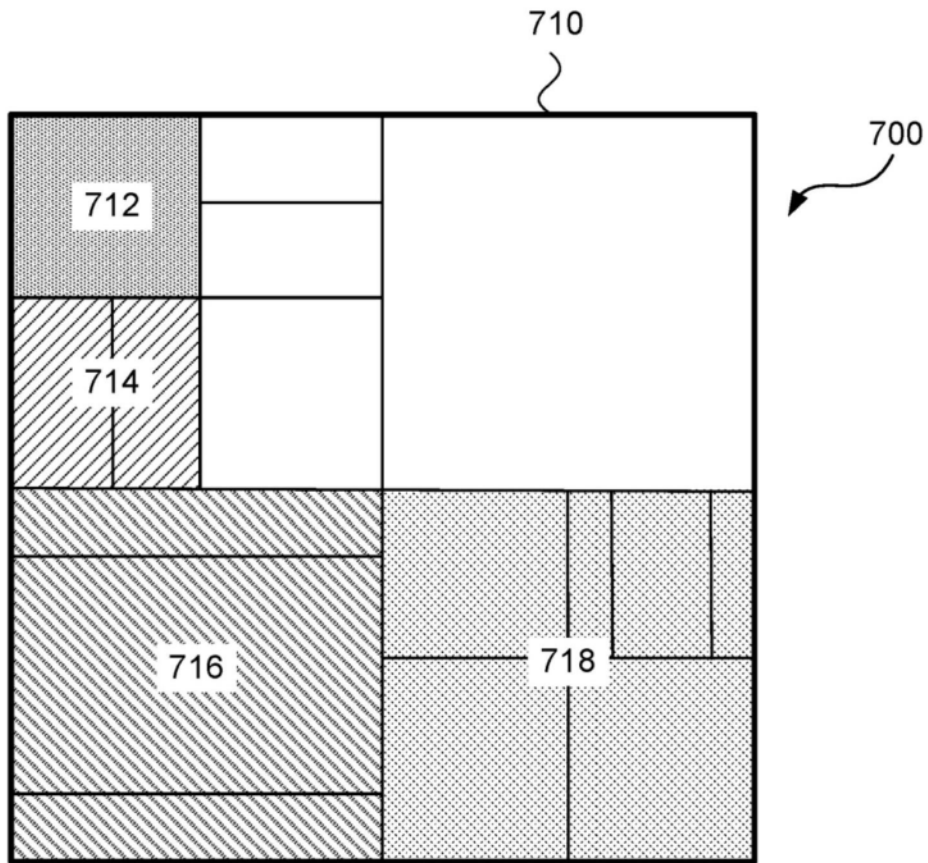


图7A

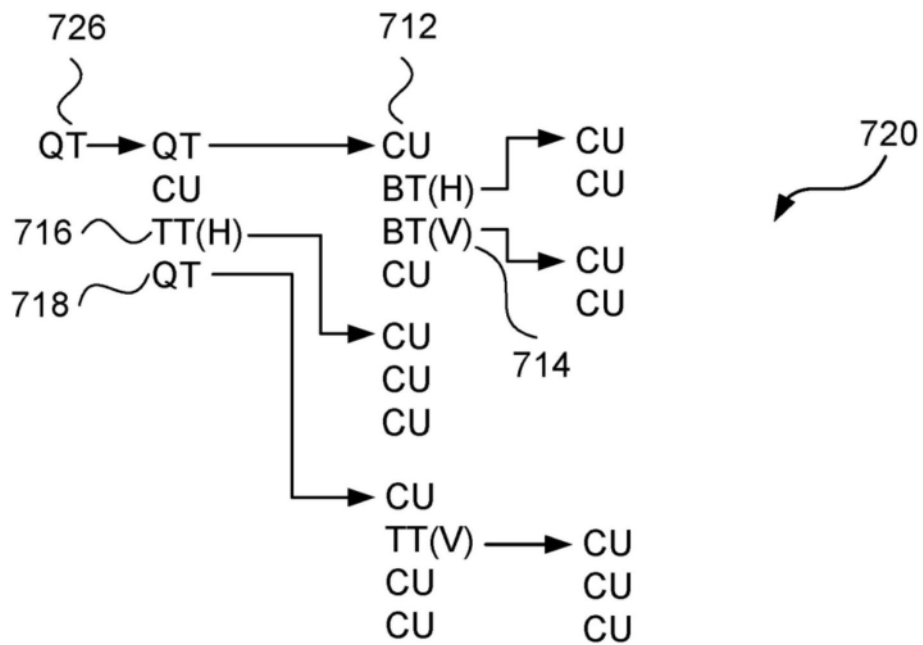


图7B

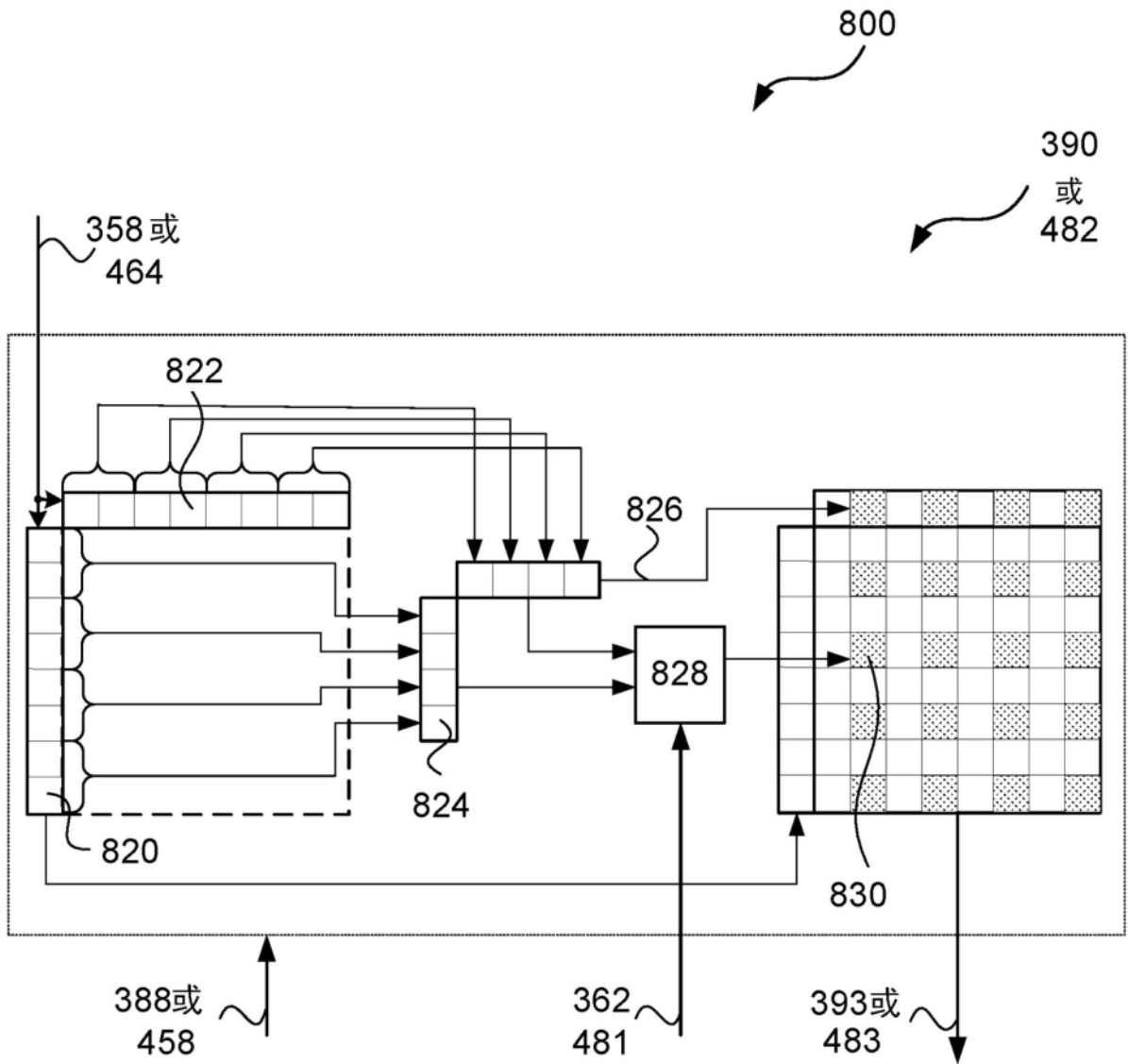


图8

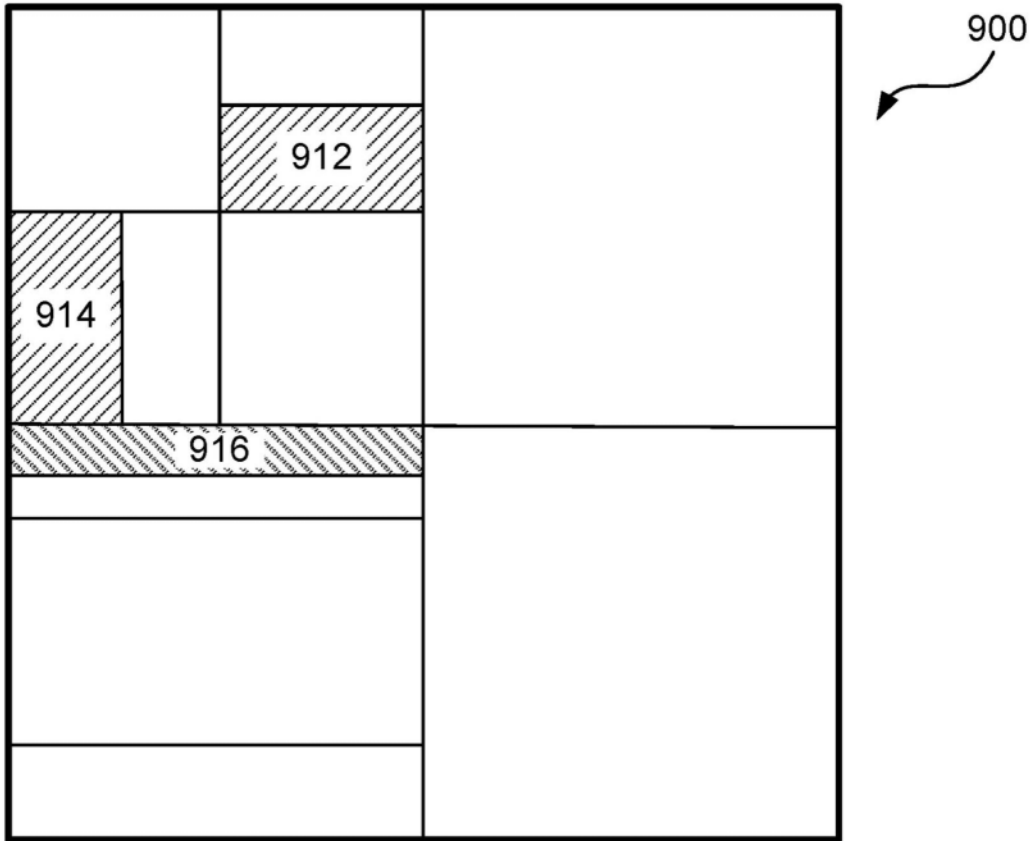


图9A

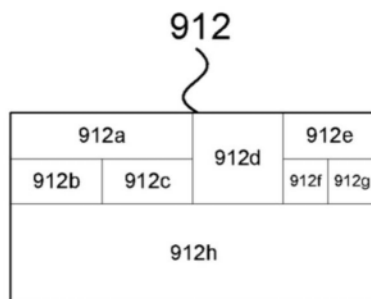


图9B

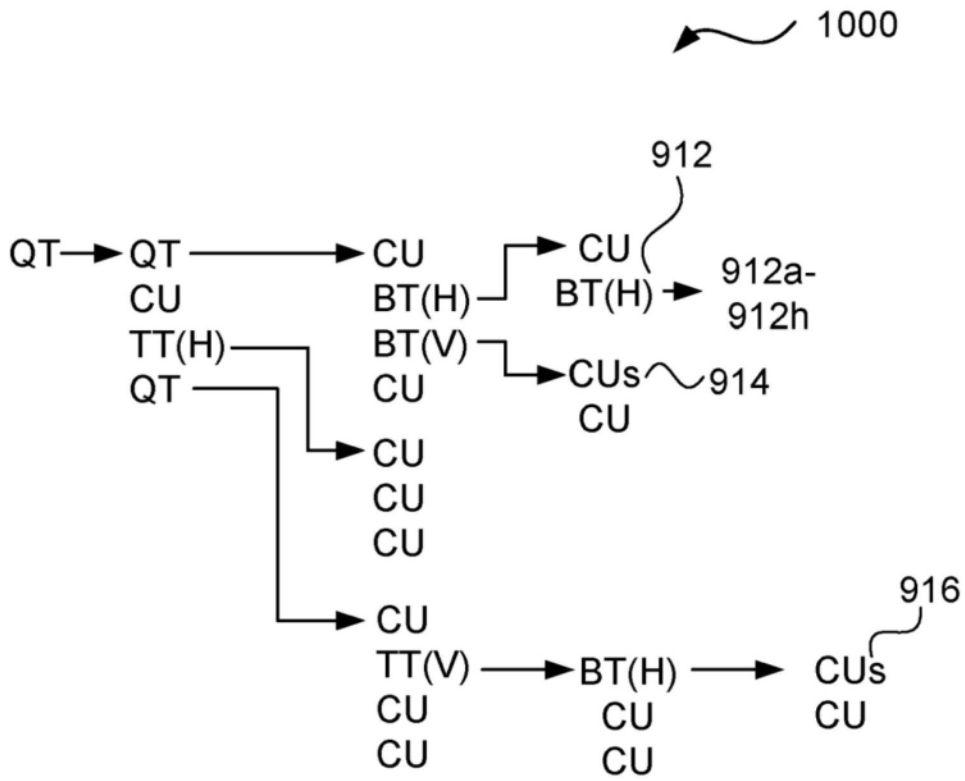


图10

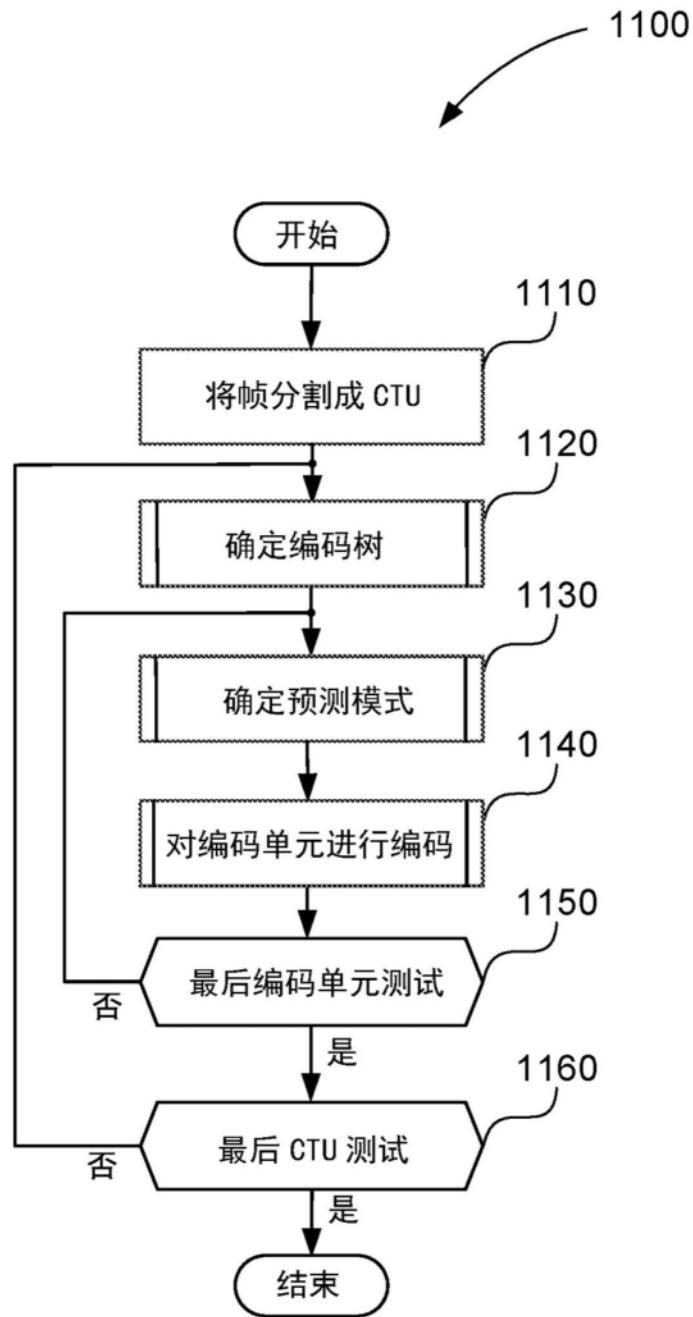


图11

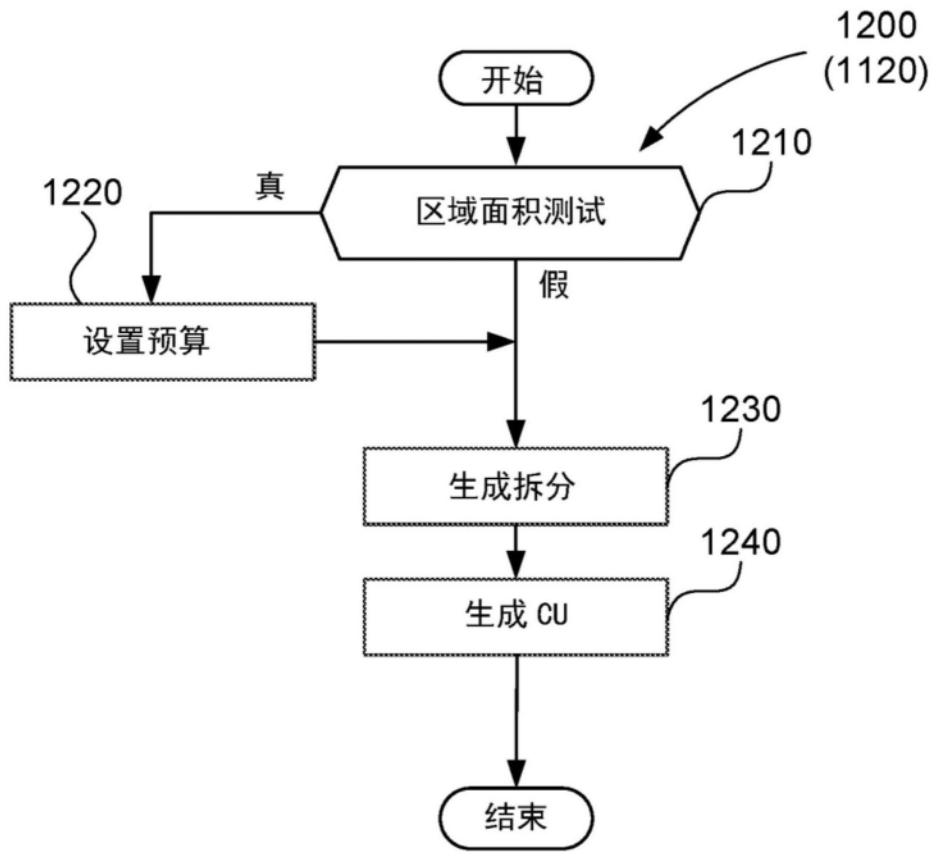


图12

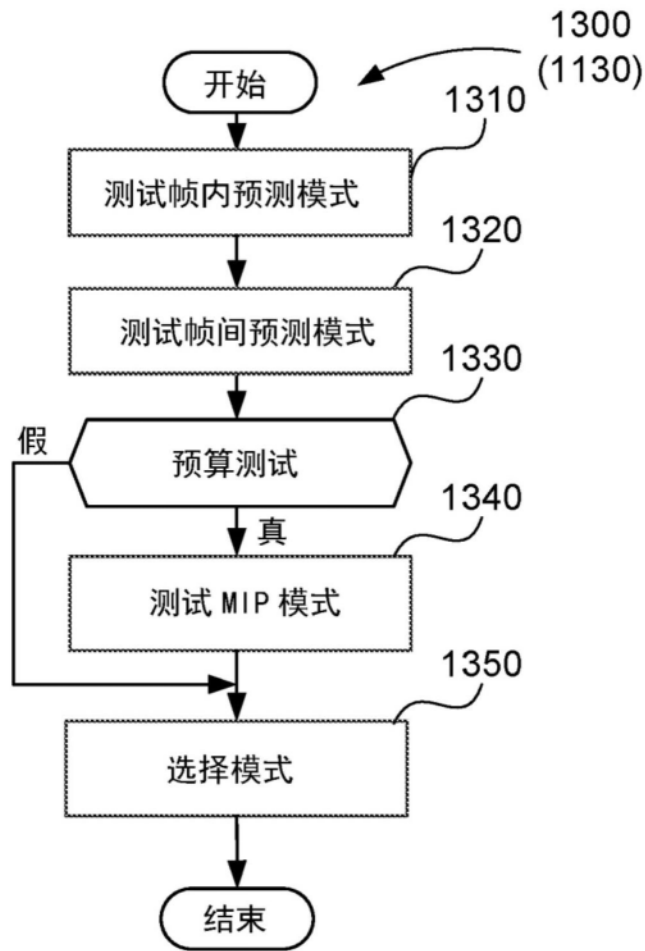


图13

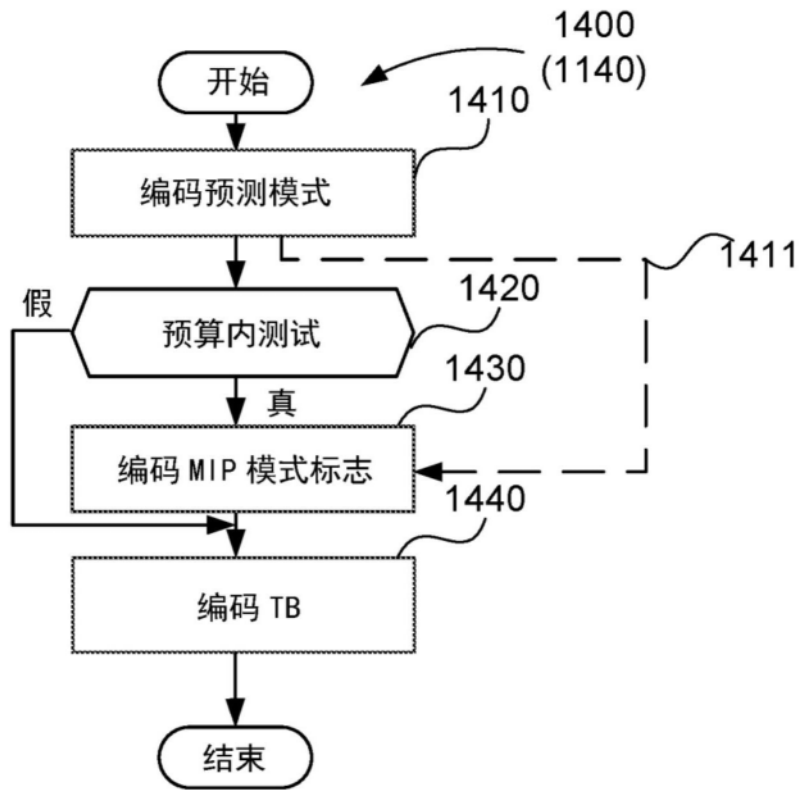


图14

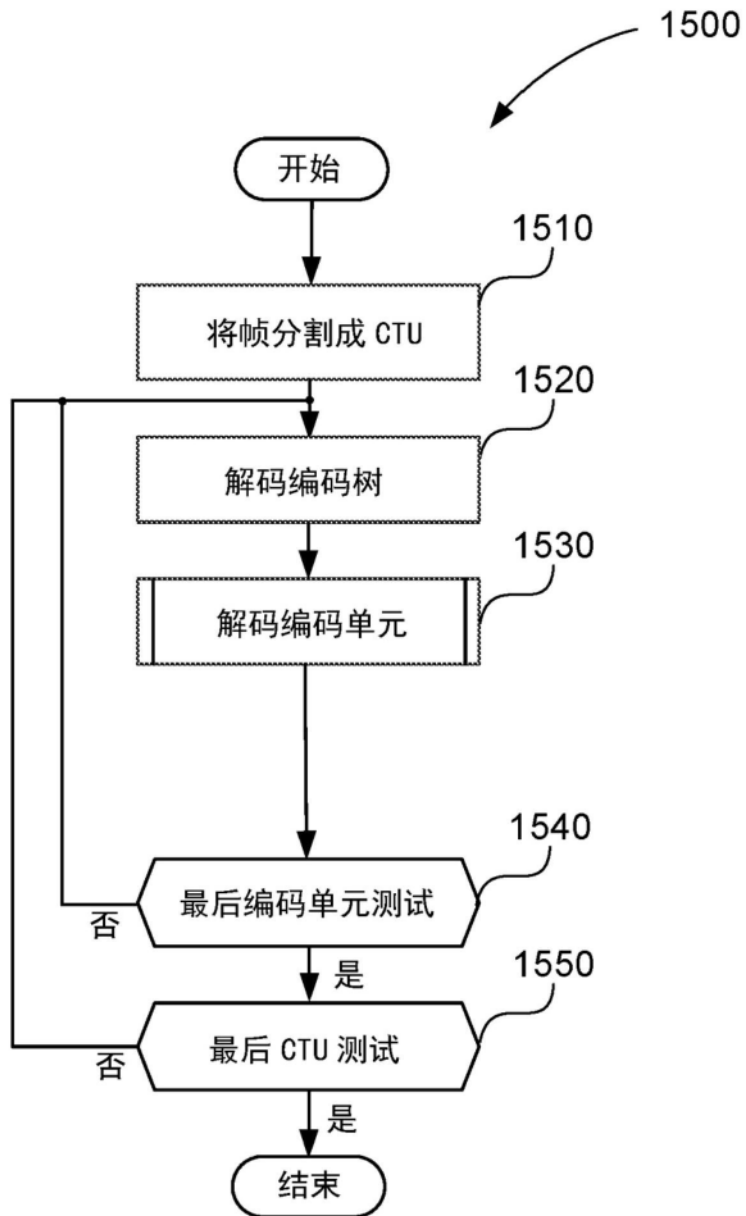


图15

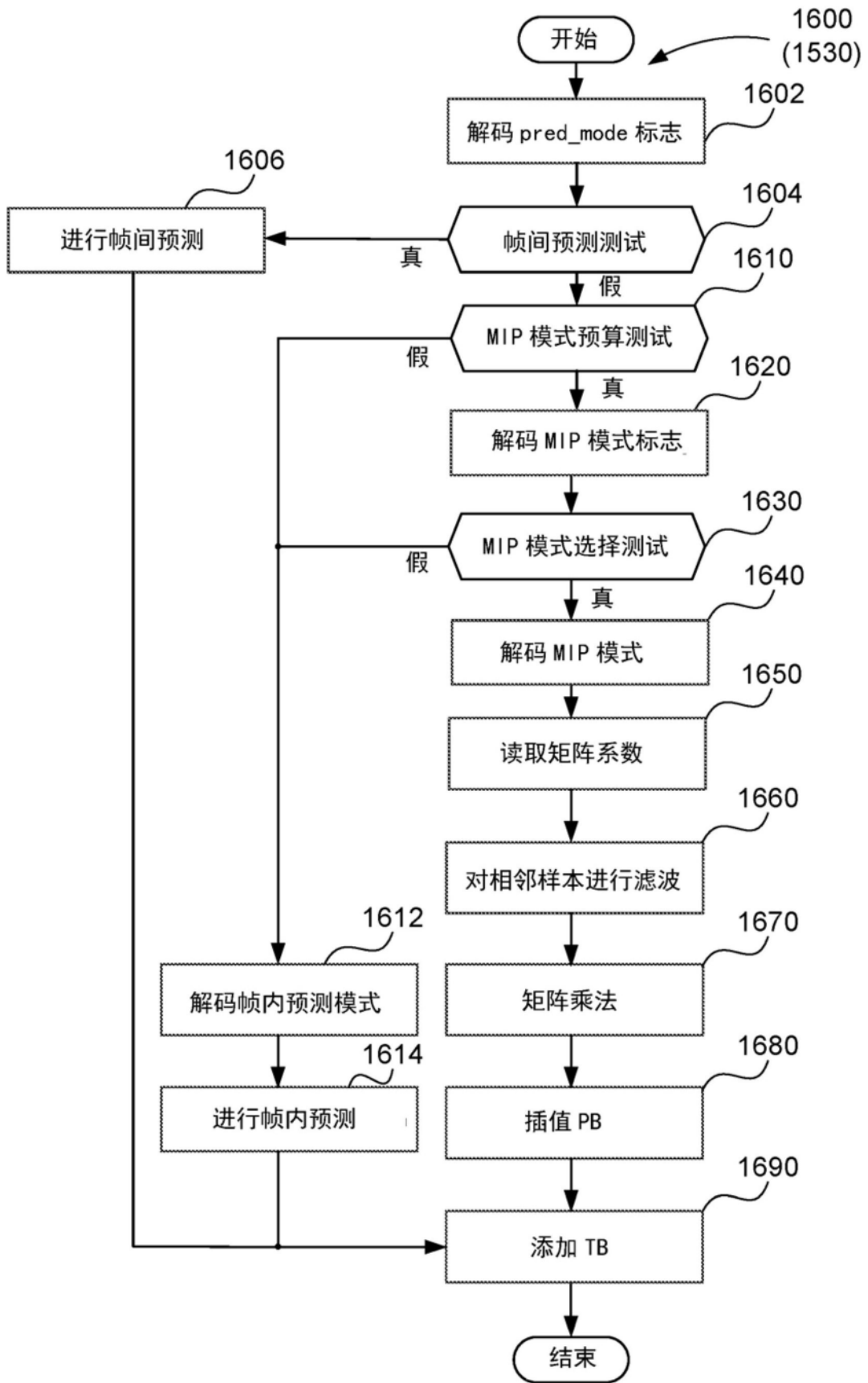


图16

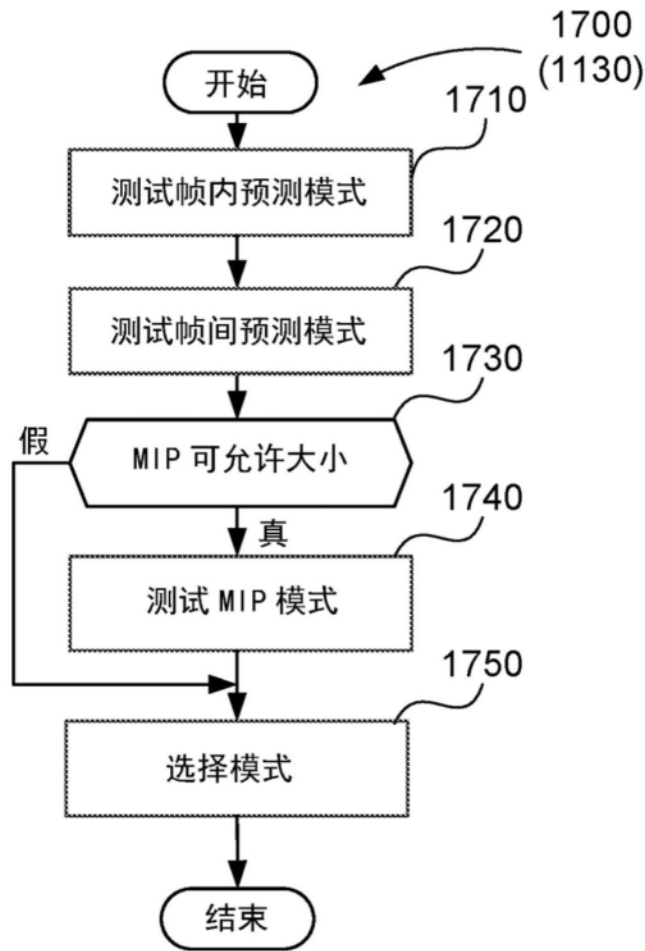


图17

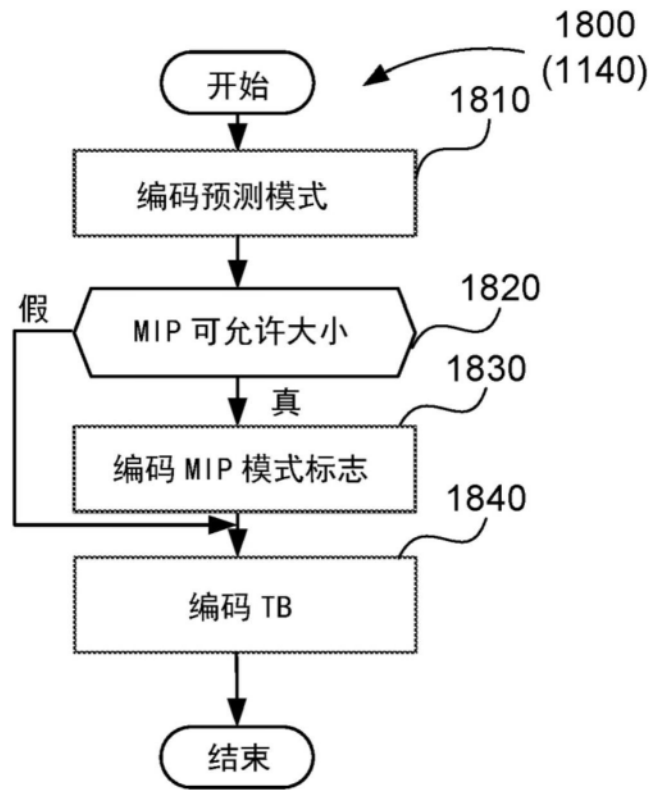


图18

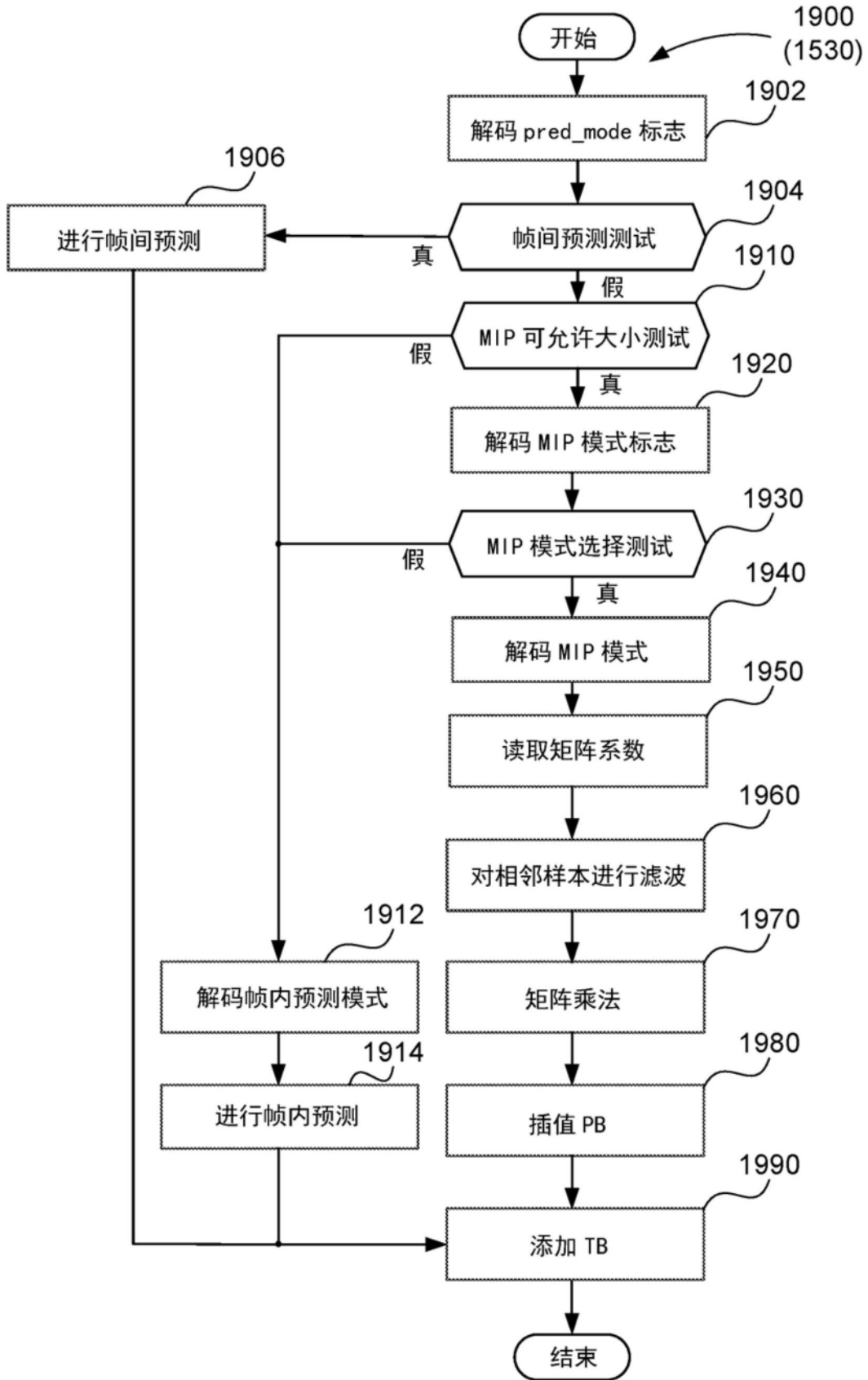


图19