



(12)发明专利申请

(10)申请公布号 CN 110662050 A

(43)申请公布日 2020.01.07

(21)申请号 201910584267.2

(51)Int.Cl.

(22)申请日 2019.07.01

H04N 19/176(2014.01)

(30)优先权数据

H04N 19/172(2014.01)

62/693,415 2018.07.02 US

H04N 19/174(2014.01)

H04N 19/119(2014.01)

(66)本国优先权数据

PCT/CN2018/093631 2018.06.29 CN

(71)申请人 北京字节跳动网络技术有限公司

地址 100041 北京市石景山区实兴大街30

号院3号楼2层B-0035房间

申请人 字节跳动有限公司

(72)发明人 张凯 张莉 刘鸿彬 庄孝强

王悦

(74)专利代理机构 北京市柳沈律师事务所

11105

代理人 彭久云

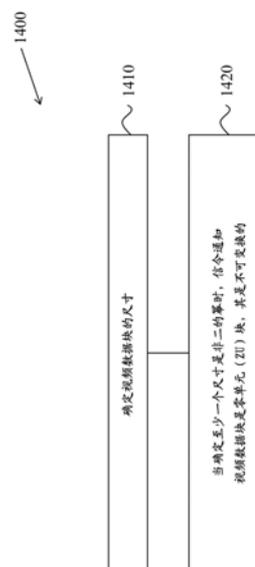
权利要求书3页 说明书18页 附图24页

(54)发明名称

零单元的定义

(57)摘要

描述了在视频和图像编码中使用零单元的装置、系统和方法。在一个代表性方面,用于视频编码的方法包括基于至少一个维度是非二的幂,而确定具有该维度的视频数据块是零单元(ZU)块;响应于确定视频数据块是ZU块,编码ZU块数据而不使用变换系数编码;以及在比特流中信令通知编码的ZU。在另一代表性方面,一种视频编码方法包括,接收对应于视频数据块的比特流,视频数据块具有维度,接收指示视频数据块是零单元(ZU)块的信令,并且基于信令,解码对应于ZU块的比特流,以在不使用变换系数解码的情况下重建视频数据块,其中ZU块的至少一个维度是非二的幂。



1. 一种视频编码方法,包括:
基于至少一个维度是非二的幂数来确定视频数据块是零单元 (ZU) 块;
响应于确定所述视频数据块是ZU块,对所有残差被设为零的所述ZU块进行编码;以及在比特流中信令通知被编码的ZU块。
2. 一种视频解码方法,包括:
接收对应于视频数据块的比特流;
由于所述块的至少一个维度为非二的幂数,确定所述视频数据块被编码为所有残差设为零的零单元 (ZU) 块;以及
基于所述决定解码所述块。
3. 如权利要求1或2所述的方法,其中,所述视频数据块的维度的任一个是 2^N 形式的偶数。
4. 如权利要求1或2所述的方法,其中,所述视频数据块的维度的任一个为 2^{KN} 形式,其中K是整数。
5. 如权利要求1或2所述的方法,其中,不对所述ZU块执行变换操作或逆变换操作。
6. 如权利要求1或2所述的方法,其中,不对所述ZU块执行量化操作或去量化操作。
7. 如权利要求1或2所述的方法,其中,所述信令不包括任何指示预测模式的语法元素。
8. 如权利要求7所述的方法,其中,推断所述ZU块以Skip模式编码。
9. 如权利要求7或8所述的方法,其中,所述信令不包括merge索引。
10. 如权利要求1或2所述的方法,其中,所述ZU块的运动信息从相邻块的运动信息继承。
11. 如权利要求1或2所述的方法,其中,所述ZU块的运动信息从大小为 $2^N \times 2^M$ 的相邻块的运动信息继承。
12. 如权利要求11所述的方法,其中,所述相邻块的侧边与接触图片边界的所述ZU块的侧边平行。
13. 如权利要求11所述的方法,其中,所述ZU块与所述相邻块具有连续性。
14. 如权利要求1或2所述的方法,其中,所述信令不包括cbf_flag并且所述ZU块的cbf_flag被推断为零。
15. 如权利要求1或2所述的方法,其中,所述信令不包括残差信息。
16. 如权利要求1或2所述的方法,其中,所述ZU块具有残差并且未对所述ZU块执行变换或逆变换操作。
17. 如权利要求1或2所述的方法,其中,用变换跳过 (transform-skip) 模式编码所述ZU块中的残差。
18. 如权利要求1或2所述的方法,其中,所述信令不包括transform-skip标志。
19. 如权利要求1或2所述的方法,其中,所述ZU块是在非对称二叉树 (ABT) 划分中分割的编码单元 (CU)。
20. 如权利要求1或2所述的方法,其中,所述ZU块是图片、条带或片的边界处的块。
21. 如权利要求1或2所述的方法,其中,在图片、条带或片的边界处的每个编码单元是ZU块。
22. 如权利要求19所述的方法,其中,所述信令包括区分所述CU和所述ZU块的标志。

23. 如权利要求1或2所述的方法,其中,所述信令包括所述ZU块的至少一个维度的最大值或最小值。

24. 如权利要求23所述的方法,其中,所述信令在视频参数集(VPS)、序列参数集(SPS)、图片参数集(PPS)、条带报头、编码树单元(CTU)或编码单元(CU)中。

25. 如权利要求1或2所述的方法,其中,所述ZU块可以仅是帧间编码的块。

26. 如权利要求25所述的方法,其中,所述ZU块可以仅在P/B图片/条带中使用。

27. 如权利要求25所述的方法,其中,所述信令不包括预测模式标志。

28. 如权利要求25所述的方法,其中,所述ZU块结合几何划分实现。

29. 如权利要求28所述的方法,其中,所述几何划分包括基于父划分的形状的划分。

30. 如权利要求28所述的方法,其中,所述几何划分包括仅使用Merge模式。

31. 如权利要求28所述的方法,其中,所述信令包括指示用于子运动划分的Merge模式或常规AMVP模式之一的标志。

32. 如权利要求25所述的方法,其中,使用2x2 ATMVP模式预测所述ZU块。

33. 如权利要求25所述的方法,其中,使用2x2仿射模式预测所述ZU块。

34. 一种视频编码方法,包括:

基于块的维度是二的幂数,确定视频数据块是零单元(ZU)块;

基于决定该块为ZU块,对比特流中的所述视频数据块进行编码;以及在比特率中信令通知所述ZU块。

35. 一种视频解码方法,包括:

接收对应于视频数据块的比特流;

基于所述块的维度为二的幂数来确定所述视频数据块为零单元(ZU)块;以及

基于所述信令解码所述比特流以重建所述视频数据块。

36. 如权利要求34或35所述的方法,其中,不对所述ZU块执行变换操作或逆变换操作。

37. 如权利要求34或35所述的方法,其中,不对所述ZU块执行量化操作或去量化操作。

38. 如权利要求34到37的任一项所述的方法,其中,所述确定还包括由于所述视频数据块的样本的总数大于或等于阈值数T,确定所述视频数据块为所述ZU块。

39. 如权利要求34到37的任一项所述的方法,其中,所述确定还包括由于所述维度进一步大于所述视频数据块的宽度的第一阈值或者所述视频数据块的高度的第二阈值,确定所述视频数据块是所述ZU块。

40. 一种视频编码方法,包括:

基于具有小于或等于第一整数的宽度并具有小于或等于第二整数的高度来确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,

基于决定该块为ZU块,将所述视频数据块编码为ZU块;以及在比特流中信令通知所述ZU块。

41. 一种视频解码方法,包括:

接收对应于具有宽度和高度的视频数据块的比特流;

基于具有大于或等于第一整数的宽度并具有大于或等于第二整数的高度来确定视频数据块将被解码为零单元(ZU)块;以及

基于所述信令解码所述比特流以重建所述视频数据块。

42. 如权利要求40或41所述的方法,其中,所述第一整数和所述第二整数为128或256的一个。

43. 一种视频编码方法,包括:

基于具有大于或等于第一整数的宽度并且具有大于或等于第二整数的高度来确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,

基于决定该块为ZU块,将所述视频数据块编码为ZU块;以及
在比特流中信令通知所述ZU块。

44. 一种视频解码方法,包括:

接收对应于具有宽度和高度的视频数据块的比特流;

基于具有大于或等于第一整数的宽度或具有大于或等于第二整数的高度来确定视频数据块将被解码为零单元(ZU)块;以及

基于所述信令解码所述比特流以重建所述视频数据块。

45. 如权利要求43或44所述的方法,其中,所述第一整数和所述第二整数为128或256的一个。

46. 一种视频编码方法,包括:

基于具有大于或等于整数的宽度和高度的乘积,确定具有所述宽度和所述高度的视频数据块将被编码为零单元(ZU)块,

基于决定该块为ZU块,将所述视频数据块编码为ZU块;以及
在比特流中信令通知所述ZU块。

47. 一种视频解码方法,包括:

接收对应于具有宽度和高度的视频数据块的比特流;

基于具有大于或等于整数的所述宽度和所述高度的乘积来确定视频数据块将被解码为零单元(ZU)块;以及

基于所述信令解码所述比特流以重建所述视频数据块。

48. 如权利要求46或47所述的方法,其中,所述整数是16384或65536中的一个。

49. 一种视频系统中的装置,包括处理器和其上具有指令的非瞬时性的存储器,其中,当所述指令由所述处理器执行时,使得所述处理器实现如权利要求1到48中的一个或多个所述的方法。

50. 一种储存在非瞬时性的计算机可读介质上的计算机程序产品,所述计算机程序产品包括用于执行如权利要求1到48中的一个或多个所述的方法的程序代码。

51. 如本文件中所描述的方法、装置或系统。

零单元的定义

[0001] 相关申请的交叉引用

[0002] 根据适用的专利法和/或依据巴黎公约的规则,本申请及时要求于2018年6月29日提交的国际专利申请第PCT/CN2018/093631号和于2018年7月2日提交的美国临时专利申请第62/693,415号的优先权和权益。出于根据美国法律的所有目的,该国际专利申请第PCT/CN2018/093631号和美国临时申请第62/693,415号的全部公开内容通过引用并入作为本申请的公开内容的一部分。

技术领域

[0003] 本专利文件一般地涉及图像和视频编码技术。

背景技术

[0004] 数字视频占据因特网和其它数字通信网络上最大的带宽使用。随着能够接收和显示视频的所连接的用户设备的数量增加,预计数字视频使用的带宽需求将继续增长。

发明内容

[0005] 描述了与专用编码单元(CU)和/或编码树单元(CTU)相关的用于提高编码效率的设备、系统和方法。具体地,本公开的技术公开了提供增强例如处理位于视频数据块的边界处的子块(例如,在图片(picture)、条带(slice)、片(tile)等中)的零单元。所描述的方法可以应用于现有的视频编码标准(例如,高效视频编码(HEVC))和未来的视频编码标准或视频编解码器两者。

[0006] 在一个示例方面中,公开了一种视频处理的方法。该方法包括基于至少一个维度是非二的幂数来确定视频数据块是零单元(ZU)块,响应于确定视频数据块是ZU块,对所有残差被设为零的ZU块进行编码;以及在比特流中信令通知编码的ZU块。

[0007] 在另一示例方面中,公开了另一种视频处理方法。该方法包括接收对应于视频数据块的比特流;由于块的至少一个维度为非二的幂数,确定视频数据块被编码为所有残差设为零的零单元(ZU)块;以及基于决定而解码视频数据块。

[0008] 在另一示例方面中,公开了另一种视频处理方法。该方法包括基于块的维度是二的幂数,确定视频数据块是零单元(ZU)块;基于确定该块为ZU块,对比特流中的视频数据块进行编码;以及在比特率中信令通知ZU块。

[0009] 在另一示例方面中,公开了另一种视频处理方法。该方法包括接收对应于视频数据块的比特流;基于块的维度为二的幂数来确定视频数据块为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。

[0010] 在另一示例方面中,公开了另一种视频处理方法。该方法包括,基于具有小于或等于第一整数的宽度并具有小于或等于第二整数的高度,确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,基于决定该块为ZU块,将视频数据块编码为ZU块;以及在比特流中信令通知ZU块。

[0011] 在另一示例方面中,公开了另一种视频处理方法。该方法包括接收对应于具有宽度和高度的视频数据块的比特流;基于具有大于或等于第一整数的宽度并具有大于或等于第二整数的高度来确定视频数据块将被解码为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。

[0012] 在另一示例方面中,公开了另一种视频处理方法。该方法包括基于具有大于或等于第一整数的宽度并且具有大于或等于第二整数的高度来确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,基于决定该块为ZU块,将视频数据块编码为ZU块;以及在比特流中信令通知ZU块。

[0013] 在另一示例方面中,公开了另一种视频处理方法。该方法包括接收对应于具有宽度和高度的视频数据块的比特流;基于具有大于或等于第一整数的宽度或具有大于或等于第二整数的高度来确定视频数据块将被解码为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。

[0014] 在另一示例方面中,公开了另一种视频处理方法。该方法包括,基于具有大于或等于整数的宽度和高度的乘积,确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,基于决定该块为ZU块,将视频数据块编码为ZU块;以及在比特流中信令通知ZU块。

[0015] 在另一示例方面中,公开了另一种视频处理方法。该方法包括接收对应于具有宽度和高度的视频数据块的比特流;基于具有大于或等于整数的宽度和高度的乘积来确定视频数据块将被解码为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。

[0016] 在另一代表性方面中,上述方法以存储用于实现该方法的处理器可执行代码的计算机可读介质的形式体现。

[0017] 在另一代表性方面中,公开了一种配置为或可操作为执行上述方法的设备。该设备可以包括处理器,该处理器被编程以实现该方法。

[0018] 在另一代表性方面中,视频解码器装置可实现如本文所述的方法。

[0019] 在附图、说明书和权利要求中更详细地描述了本公开的技术的上述和其它方面和特征。

附图说明

[0020] 图1示出了典型的高效视频编码(HEVC)视频编码器和解码器的示例框图。

[0021] 图2示出了H.264/AVC中的宏块(MB)划分的示例。

[0022] 图3示出了将编码块(CB)分割成预测块(PB)的示例。

[0023] 图4A和4B分别示出了将编码树块(CTB)细分为CB和变换块(TB)以及相应的二叉树的示例。

[0024] 图5示出了一帧的划分结构的示例。

[0025] 图6A和6B分别示出了在图5中的示例性帧中突出显示的CTB的细分和信令方法。

[0026] 图7A至7B是视频处理方法示例的流程图。

[0027] 图8A至8B是视频处理方法示例的流程图。

[0028] 图9A至9C是视频处理方法示例的流程图。

[0029] 图10A至10I示出了基于QTBT的CB的细分示例。

[0030] 图11示出了树类型信令的示例。

- [0031] 图12A至12C示出了跨越图片边界的CTB的示例。
- [0032] 图13示出了图片边界处的零单元的示例。
- [0033] 图14示出了根据本公开的技术的用于视频编码的示例方法的流程图。
- [0034] 图15示出了根据本公开的技术的用于视频解码的另一示例方法的流程图。
- [0035] 图16是示出可用于实现本公开技术的各个部分的计算机系统或其它控制设备的架构的示例的框图。
- [0036] 图17示出了可用于实现本公开技术的各个部分的移动设备的示例实施例的框图。
- [0037] 图18A和18B示出了将ZU块分割为子块以进行自适应环路控制 (ALF) 开/关控制的示例。

具体实施方式

[0038] 由于对更高分辨率视频的需求日益增加,视频编码方法和技术在现代技术中无处不在。视频编解码器通常包括压缩或解压缩数字视频的电子电路或软件,并且视频编解码器被不断改进以提供更高的编码效率。视频编解码器将未压缩视频转换为压缩格式,或反之亦然。视频质量、用于表示视频的数据量(由比特率确定)、编码和解码算法的复杂性、对数据丢失和错误的敏感性、编辑便利性、随机访问以及端到端延迟(延迟)之间存在复杂的关系。压缩格式通常符合标准视频压缩规范,例如高效视频编码(HEVC)标准(也称为H.265或MPEG-H第2部分)、即将最终确定的通用视频编码标准,或其它当前和/或未来的视频编码标准。

[0039] 所公开的技术的实施例可以应用于现有的视频编码标准(例如,HEVC,H.265)和未来的标准以改善压缩性能。在本文件中使用章节标题来提高描述的可读性而不是以任何方式将讨论或实施例(和/或实现方式)仅限于各个章节。

[0040] 在本文件中使用章节标题以便于理解,并且不将章节中公开的实施例仅限于该章节。此外,虽然参考通用视频编码或其它特定视频编解码器描述了某些实施例,但是所公开的技术还可应用于其它视频编码技术。此外,虽然一些实施例详细描述了视频编码步骤,但是应当理解的是,撤销编码的相应解码步骤将由解码器实现。此外,术语视频处理包括视频编码或压缩、视频解码或解压缩以及视频转码,其中视频像素从一种压缩格式表示为另一压缩格式或以不同的压缩比特率表示。

[0041] 1. 视频编码的示例实施例

[0042] 图1示出了典型HEVC视频编码器和解码器的示例性框图。产生符合HEVC的比特流的编码算法通常如下进行。每个图片被分割为块状区域,其中精确的块划分被传送到解码器。视频序列的第一图片(以及在对视频序列的每个干净随机访问点处的第一图片)仅使用帧内预测(在同一图片内使用区域到区域的空间上的数据的一些预测,而不基于其它图片)进行编码。对于序列的所有剩余图片或随机访问点之间的图片,对于大多数块通常使用帧间时间预测编码模式。帧间预测的编码过程包括选择包含选定的参考图片和运动矢量(MV)的运动数据,该参考图片和运动矢量(MV)将被应用于预测每个块的样本。编码器和解码器通过应用运动补偿(MC)来生成相同的帧间预测信号,该运动补偿(MC)使用作为辅助信息发送的MV和模式确定数据。

[0043] 通过线性空间变换对帧内预测或帧间预测的残差信号进行变换,其中该残差信号

是原始块与其预测之间的差异。然后对变换系数进行缩放、量化、熵编码并与预测信息一起发送。

[0044] 编码器复制解码器处理环路(见图1中的灰色阴影框),使得两者都将为后续数据生成相同的预测。因此,量化的变换系数通过逆缩放构造,并且然后进行逆变换以复制残差信号的解码的近似值。然后将残差加入到预测中,并且然后可以将该加法的结果输入一个或两个环路滤波器,以平滑由逐块处理和量化引起的伪影(artifact)。最终的图片表示(即解码器输出的复制)存储在解码的图片缓冲器中以用于后续图片的预测。通常,图片的编码或解码处理的顺序常常不同于它们从来源到达的顺序,需要区分解码器的解码顺序(即比特流顺序)和输出顺序(即显示顺序)。

[0045] 通常期望将由HEVC编码的视频材料作为逐行扫描图像输入(由于源视频源自该格式或者由于由编码之前的去隔行(deinterlacing)产生)。在HEVC设计中不存在显式编码特征以支持隔行扫描的使用,因为隔行扫描不再用于显示器,并且在分布时变得非常不常见。但是,已经在HEVC中提供了元数据语法以允许编码器指示,隔行扫描的视频已经通过将隔行扫描视频的每个场(即每个视频帧的偶数或奇数行)编码为单独的图片而被发送,或者隔行扫描的视频已经通过将每个隔行扫描的帧编码为HEVC编码的图片而被发送。这提供了一种有效的编码隔行扫描的视频的方法,而无需加重解码器需支持隔行扫描的视频的特殊的解码过程的负担。

[0046] 1.1.H.264/AVC中的划分树结构的示例

[0047] 先前标准中的编码层的核心是宏块,包含 16×16 的亮度(luma)样本块、以及在4:2:0颜色采样的通常情况下的两个对应的 8×8 的色度(chroma)样本块。

[0048] 帧内编码块使用空间预测来利用像素之间的空间相关性。两种划分被定义为: 16×16 和 4×4 。

[0049] 帧间编码块通过估计图片之间的运动来使用时间预测而不是空间预测。可以对于 16×16 宏块或其任何子宏块划分 16×8 、 8×16 、 8×8 、 8×4 、 4×8 、 4×4 来独立地估计运动,如图2所示。每子宏块划分仅允许一个运动矢量(MV)。

[0050] 1.2.HEVC中的划分树结构的示例

[0051] 在HEVC中,通过使用表示为编码树的二叉树结构将编码树单元(CTU)划分成编码单元(CU),以适应各种局部特性。使用帧间(时间)预测还是帧内(空间)预测来对图片区域进行编码的确定是在CU级做出的。根据预测单元(PU)划分类型,每个CU可以进一步划分成一个、两个或四个PU。在一个PU内部,应用相同的预测过程,并且在PU的基础上将相关信息发送到解码器。在通过基于PU划分类型应用预测过程而获得了残差块之后,可以根据类似于CU的编码树的另一个二叉树结构将CU划分为变换单元(TU)。HEVC结构的关键特征之一是它具有多个划分概念,包括CU、PU和TU。

[0052] 对使用HEVC的混合视频编码中涉及的某些特征包括:

[0053] (1) 编码树单元(CTU)和编码树块(CTB)结构:HEVC中的类似结构是编码树单元(CTU),其具有由编码器选择的尺寸并且可以大于传统宏块。CTU由亮度CTB和相应的色度CTB以及语法元素组成。亮度CTB的尺寸 $L \times L$ 可以被选择为 $L = 16, 32$ 或 64 个样本,较大的尺寸通常能够实现更好的压缩。然后,HEVC支持使用树结构和类似二叉树的信令来将CTB划分为更小的块。

[0054] (2) 编码单元 (CU) 和编码块 (CB) :CTU的四叉树语法指定其亮度CB和色度CB的尺寸和位置。四叉树的根与CTU相关联。因此,亮度CTB的尺寸是亮度CB的最大支持尺寸。将CTU划分成亮度CB和色度CB是信令联合的。一个亮度CB和通常两个色度CB以及相关语法一起形成编码单元 (CU) 。CTB可以仅包含一个CU或者可以被划分以形成多个CU,并且每个CU具有相关联的、向预测单元 (PU) 以及变换单元 (TU) 的树的划分。

[0055] (3) 预测单元和预测块 (PB) :使用帧间图片预测还是帧内图片预测来对图片区域编码的决策是在CU级进行的。PU划分结构的根在CU级。取决于基本预测类型决策,然后可以在尺寸上进一步划分亮度CB和色度CB,并根据亮度和色度预测块 (PB) 对其进行预测。HEVC支持从 64×64 到 4×4 样本的可变PB尺寸。图3示出了对于 $M \times M$ CU所允许的PB的示例。

[0056] (4) 变换单元 (TU) 和变换块:使用块变换对预测残差进行编码。TU树结构的根在CU级。亮度CB残差可以与亮度变换块 (TB) 相同,或者可以被进一步划分成更小的亮度TB。这同样适用于色度TB。对于正方形TB尺寸 4×4 、 8×8 、 16×16 和 32×32 ,定义类似于离散余弦变换 (DCT) 的整数基函数。对于亮度帧内图片预测残差的 4×4 变换,可以替代地指定从离散正弦变换 (DST) 的形式导出的整数变换。

[0057] 1.2.1. 树结构的划分到TB和TU的示例

[0058] 对于残差编码,可以将CB递归地划分为变换块 (TB) 。由残差四叉树信令通知划分。仅指定正方形CB和TB划分,其中块可以被递归地划分到象限 (quadrant) ,如图4A到4B所示。对于尺寸为 $M \times M$ 的给定亮度CB,标志 (flag) 表明是否将该CB划分成四个尺寸为 $M/2 \times M/2$ 的块。如果如SPS中指示的残差四叉树的最大深度所信令通知的,每个象限能够进一步划分,则为每个象限分配标志,该标志指示是否将其划分成四个象限。由残差四叉树产生的叶节点块是变换块,通过变换编码对其进一步处理。编码器指示它将使用的最大和最小亮度TB尺寸。当CB尺寸大于最大TB尺寸时,划分是隐含的。当划分将导致亮度TB尺寸小于所指示的最小值时,不划分是隐含的。除了当亮度TB尺寸为 4×4 时,色度TB尺寸在每个维度上是亮度TB尺寸的一半,在亮度TB尺寸为 4×4 的情况下,单个 4×4 色度TB被用于由四个 4×4 亮度TB覆盖的区域。在帧内图片预测的CU的情况下,最近邻的TB (在CB内或在CB外) 的解码后样本被用于帧内图片预测的参考数据。

[0059] 与之前的标准相反,HEVC设计允许TB跨越多个PB以用于帧间预测的CU,以使四叉树结构的TB划分的潜在编码效率益处最大化。

[0060] 1.2.2. 图片边界编码的示例

[0061] 图片的边界以最小允许的亮度CB大小为单位定义。因此,在图片的右边界和底边界,一些CTU可能会覆盖部分位于图片边界之外的区域。该条件被解码器检测到,并且CTU四叉树根据需要进行隐式地分割,以将CB大小减小到整个CB将适合到图片中的程度。

[0062] 图5示出了一个帧的划分结构的示例,其中分辨率为 416×240 像素,尺寸为 $7CTB \times 4CTB$,其中CTB的大小为 64×64 。如图5所示,部分位于右边界和底边界之外的CTB具有隐含的分割 (虚线,表示为502),并且完全落在外面的CU就被跳过 (不被编码) 。

[0063] 在图5所示的示例中,突出显示的CTB (504),其行CTB索引等于2且列CTB索引等于3,在当前图片内具有 64×48 像素,并且不适合 64×64 CTB。因此,它被强制分割为 32×32 而没有分割标志信号。对于左上角的 32×32 ,它被帧完全覆盖。当它选择根据速率-失真成本以较小的块编码时 (对于左上角的 16×16 用 8×8 ,并且其余的以 16×16 编码),需要编码几个分割标

志。这些分割标志(用于是否将左上角的32x32分割为四个16x16块的一个标志,以及用于信令通知一个16x16是否被进一步分割,以及对于在左上角的16x16中的四个8x8块中的每一个是否进一步分割8x8的标志)必须被明确地信令通知。右上角的32x32块存在类似情况。对于两个底部32x32块,因为它们部分位于图片边界(506)的外部,需要应用进一步的QT分割而不用信令通知。图6A和6B分别示出了图5中突出显示的CTB(504)的细分和信令方法。

[0064] 1.2.3.CTB大小指示的示例

[0065] 表1中示出了用于一般序列参数集的示例RBSP(原始字节序列有效载荷)语法表。

[0066] 表1:RBSP语法结构

	seq_parameter_set_rbsp () {	描述符
	sps_video_parameter_set_id	u (4)
	sps_max_sub_layers_minus1	u (3)
	sps_temporal_id_nesting_flag	u (1)
[0067]	profile_tier_level (1, sps_max_sub_layers_minus1)	
	...	
	log2_min_luma_coding_block_size_minus3	ue (v)
	log2_diff_max_min_luma_coding_block_size	ue (v)
	log2_min_luma_transform_block_size_minus2	ue (v)
	log2_diff_max_min_luma_transform_block_size	ue (v)
[0068]	..	
	}	

[0069] 相应的语义包括:

[0070] $\log2_min_luma_coding_block_size_minus3+3$ 指定最小亮度编码块大小;以及

[0071] $\log2_diff_max_min_luma_coding_block_size$ 指定最大亮度编码块大小和最小亮度编码块大小之间的差异。

[0072] 变量: $MinCbLog2SizeY$, $CtbLog2SizeY$, $MinCbSizeY$, $CtbSizeY$, $PicWidthInMinCbsY$, $PicWidthInCtbsY$, $PicHeightInMinCbsY$, $PicHeightInCtbsY$, $PicSizeInMinCbsY$, $PicSizeInCtbsY$, $PicSizeInSamplesY$, $PicWidthInSamplesC$ and $PicHeightInSamplesC$ 通过以下得到:

[0073] $MinCbLog2SizeY = \log2_min_luma_coding_block_size_minus3 + 3$

[0074] $CtbLog2SizeY = MinCbLog2SizeY + \log2_diff_max_min_luma_coding_block_size$

[0075] $MinCbSizeY = 1 \ll MinCbLog2SizeY$

[0076] $CtbSizeY = 1 \ll CtbLog2SizeY$

[0077] $PicWidthInMinCbsY = pic_width_in_luma_samples / MinCbSizeY$

[0078] $PicWidthInCtbsY = Ceil(pic_width_in_luma_samples \div CtbSizeY)$

- [0079] $\text{PicHeightInMinCbsY} = \text{pic_height_in_luma_samples} / \text{MinCbSizeY}$
- [0080] $\text{PicHeightInCtbsY} = \text{Ceil}(\text{pic_height_in_luma_samples} \div \text{CtbSizeY})$
- [0081] $\text{PicSizeInMinCbsY} = \text{PicWidthInMinCbsY} * \text{PicHeightInMinCbsY}$
- [0082] $\text{PicSizeInCtbsY} = \text{PicWidthInCtbsY} * \text{PicHeightInCtbsY}$
- [0083] $\text{PicSizeInSamplesY} = \text{pic_width_in_luma_samples} * \text{pic_height_in_luma_samples}$
- [0084] $\text{PicWidthInSamplesC} = \text{pic_width_in_luma_samples} / \text{SubWidthC}$
- [0085] $\text{PicHeightInSamplesC} = \text{pic_height_in_luma_samples} / \text{SubHeightC}$
- [0086] 变量CtbWidthC和CtbHeightC,其分别指定每个色度CTB阵列的宽度和高度,通过以下得到:
- [0087] 如果chroma_format_idc等于0(单色)或者separate_colour_plane_flag等于1,则CtbWidthC和CtbHeightC都等于0;
- [0088] 否则,CtbWidthC和CtbHeightC通过以下得到:
- [0089] $\text{CtbWidthC} = \text{CtbSizeY} / \text{SubWidthC}$
- [0090] $\text{CtbHeightC} = \text{CtbSizeY} / \text{SubHeightC}$
- [0091] 1.3. 在JEM中具有较大CTU的四叉树加二叉树块结构的示例
- [0092] 在一些实施例中,使用被称为联合探索模型(JEM)(参考[4])的参考软件探索未来的视频编码技术(参考[3])。除二叉树结构外,JEM还描述了四叉树加二叉树(QTBT)和三叉树(TT)结构。
- [0093] 1.5. 替代视频编码技术中的分割结构的示例
- [0094] 在一些实施例中,支持称为多树类型(MTT)的树结构(其是QTBT的一般化)。在QTBT中,如图11所示,首先用四叉树结构划分编码树单元(CTU)。四叉树叶节点进一步由二叉树结构划分。
- [0095] MTT的结构由两种类型的树节点构成:区域树(RT)和预测树(PT),支持九种类型的划分,如图10A至10I所示。区域树可以递归地将CTU分割成方形块,直到 4×4 大小的区域的树叶节点。在区域树中的每个节点处,可以从三种树类型中的一个形成预测树:二叉树,三叉树和非对称二叉树。在PT分割中,禁止在预测树的分支中具有四叉树划分。与在JEM中一样,亮度树和色度树在I切片中分开。
- [0096] 通常,除了上下文推导之外,RT信令与JEM中的QT信令相同。对于PT信令,需要多达4个额外的二进制位(bin),如图11所示。第一个二进制位指示PT是否被进一步分割。该二进制位的上下文基于对进一步分割的可能性与当前块与其相邻块的相对大小高度相关的观察来计算。如果进一步分割PT,则第二个二进制位指示其是水平划分还是垂直划分。在一些实施例中,中心侧三叉树和非对称二叉树(ABT)的存在增加了“高”块或“宽”块的出现。第三个二进制位表示划分的树的类型,即它是二叉树/三叉树还是非对称二叉树。在二叉树/三叉树的情况下,第四个二进制位表示树的类型。在非对称二叉树的情况下,第四个二进制位指示对于水平划分树的向上或向下类型,对于垂直划分树的向右或向左类型。
- [0097] 1.5.1. 图片边界处的限制的示例
- [0098] 在一些实施例中,如果CTB/LCU大小由 $M \times N$ 指示(通常M等于N,如HEVC/JEM中所定义),并且对于位于图片(或图块或条带或其它类型)边界的CTB, $K \times L$ 个样本处于图片边界

内。

[0099] 在图片底边界和右边界上的CU分割规则可以应用于任何编码树配置QTBT+TT、QTBT+ABT或QTBT+TT+ABT。它们包括以下两个方面：

[0100] (1) 如果给定编码树节点(CU)的一部分位于图片外部,则总是允许沿着相关边界方向(沿着底边界的水平分割定向,如图12A所示,沿右边界的垂直分割定向,如图12B所示)的CU的二元对称分割。如果当前CU的右下角在帧外(如图12C所示),则只允许CU的四叉树分割。另外,如果当前二叉树深度大于最大二叉树深度并且当前CU在帧边界上,则启用二元分割以确保到达帧边界。

[0101] (2) 关于三叉树分割过程,在产生的子CU之间的第一边界或第二边界恰好位于图片的边界上的情况下,允许三叉树分割。如果分割线(由分割产生的两个子CU之间的边界)与图片边界完全匹配,则允许不对称二叉树分割。

[0102] 2. 现有实现方式的示例

[0103] 现有实现方式,CTU或CU的宽度或高度可能不等于 2^N ,其中N是正整数。这些情况很难处理。具体来说,如果行数或列数不是 2^N 的形式,则可能难以设计具有不包括除法的整数运算的变换。

[0104] 在一个示例中,为了避免CTU或CU具有不等于 2^N 的宽度或高度,CTU或CU被强制分割为更小,直到宽度和高度两者都是 2^N 的形式或通过填充或使用变换跳过。如果以更灵活的方式处理这些块,则可以进一步改善编码增益。

[0105] 在另一个示例中,对于宽度或高度不是 2^N 形式的CU定义变换。这种变换在实际的视频编码应用中是不期望的。

[0106] 3. 使用基于本公开的技术的零单元的示例方法

[0107] 本公开的技术的实施例克服了现有实施方式的缺点,从而为视频编码提供了更高的效率。具体地,提出零单元块作为特殊的CU/CTU,并且块在当且仅当其高度和/或宽度不是 2^N 的形式时被解释为零单元。

[0108] 在以下针对各种实现方式所描述的示例中阐明了使用零单元来提高视频编码效率并增强现有和未来的视频编码标准。下面提供的本公开的技术的示例解释了一般概念,并不意欲被解释为限制。在示例中,除非明确地相反指示,否则可以组合这些示例中描述的各种特征。在另一示例中,这些示例中描述的各种特征可以应用于使用向后兼容的块大小的图片边界编码和使用划分树进行可视媒体编码的方法。

[0109] 示例1. 在一个示例中,块的宽度、高度或两者都可以等于不是 2^N 的形式的任意正整数。这样的块被定义为零单元(ZU),其中所有残差被设置为0。图13示出了图片边界处的零单元的示例。

[0110] (a) 在一个示例中,ZU的宽度和高度必须是偶数(以 $2N$ 的形式)。

[0111] (b) 在一个示例中,ZU的宽度和高度必须是 $2^k N$ 的形式(例如,K等于1,2,3,4等)。

[0112] 示例2. 在一个示例中,提出对于零单元,不调用变换、逆变换、量化和去量化操作。

[0113] (a) 在一个示例中,零单元被推断为用Skip模式编码;在这种情况下,不需要信令通知skip标志、以及帧内或帧间或其它模式的指示。

[0114] (b) 可替代地,此外,也可以跳过merge索引。

[0115] (i) 零块可以从其相邻块之中的一个继承运动信息,该相邻块的大小等于 $2^N \times 2^M$ 。

[0116] (ii) 在一个示例中,相邻块被定义为与边界平行的块,例如对于位于底边界的零单元,它可以从其上面的块继承运动信息。

[0117] (iii) 在一个示例中,可以应用某些规则来选择其相邻块的一个,诸如零单元与其相邻块之间的连续性。

[0118] (iv) 在一个示例中,运动信息可以从相邻块的运动信息得到。

[0119] (c) 在一个示例中,零单元可以用Skip模式或non-skip模式、和/或帧内或帧间模式编码。在这种情况下,模式信息的传统信令保持不变,但是,不信令通知cbf_flag并且零单元的所有cbf_flag都被推断为零。对零单元,不信令通知诸如量化系数或transform_skip_flag的残差信息。

[0120] 示例3. 在一个示例中,零单元中可能存在残差。但是对于零单元没有变换和逆变换操作。

[0121] (a) 在一个示例中,零单元的残差总是用transform-skip模式编码。对于零单元,不信令通知Transform_skip_flag并且Transform_skip_flag被推断为一。

[0122] 示例4. ABT划分中的分割CU可以是零单元。

[0123] 示例5. 图片/条带/片边界处的CTU或CU可以是零单元。

[0124] (a) 在一个示例中,不需要信令通知用于边界CTU/CU的标志。对于位于图片边界上的CU,只允许ZU。

[0125] (b) 在一个示例中,所有边界CTU/CU需要标志来区分正常CU和ZU。在另一个示例中,该标志可以结合示例6和7中描述的对大小的限制条件来考虑。

[0126] 示例6. 可以预先定义零块的宽度/高度的最大值和最小值,或者可以从编码器向解码器发信令通知零块的宽度/高度的最大值和最小值。在一个示例中,它们可以在视频参数集(VPS)、序列参数集(SPS)、图片参数集(PPS)、条带报头、编码树单元(CTU)或编码单元(CU)中信令通知。

[0127] 示例7. 零单元可以进一步分割为两个单元(BT或ABT)、三个单元(TT、FTT)或四个单元(QT、EQT)。从零单元分割的分割单元可以是零单元,或者它可以是正常CU,其具有 2^N 的形式的宽度或高度。假设零单位Z的大小为 $S \times T$ 。

[0128] (a) 在一个示例中,Z可以用BT分割为两个单元,二者具有大小为 $S/2 \times T$ 。

[0129] (b) 在一个示例中,Z可以用BT分割为两个单元,二者具有大小为 $S \times T/2$ 。

[0130] (c) 在一个示例中,假设 $2^N < S \leq 2^{N+1}$,Z可以用BT分割为两个单元,其具有大小为 $2^N \times T$ 和 $(S-2^N) \times T$,或 $(S-2^N) \times T$ 和 $2^N \times T$ 。

[0131] (d) 在一个示例中,假设 $2^N < T \leq 2^{N+1}$,Z可以用BT分割为两个单元,其具有大小为 $S \times 2^N$ 和 $S \times (T-2^N)$,或 $S \times (T-2^N)$ 和 $S \times 2^N$ 。

[0132] (e) 在一个示例中,Z可以用TT分割为三个单元,其具有大小为 $S/4 \times T$ 、 $S/2 \times T$ 和 $S/4 \times T$ 。

[0133] (f) 在一个示例中,Z可以用TT分割为三个单元,其具有大小为 $S \times T/4$ 、 $S \times T/2$ 和 $S \times T/4$ 。

[0134] (g) 在一个示例中,假设 $2^N < S < 2^{N+1}$,Z可以用TT分割为三个单元,其具有大小为 $2^{N-1} \times T$ 、 $2^{N-1} \times T$ 和 $(S-2^N) \times T$,或 $2^{N-1} \times T$ 、 $(S-2^N) \times T$ 和 $2^{N-1} \times T$,或 $(S-2^N) \times T$ 、 $2^{N-1} \times T$ 和 $2^{N-1} \times T$ 。

[0135] (h) 在一个示例中,假设 $2^N < T \leq 2^{N+1}$,Z可以用TT分割为三个单元,其具有大小为 $S \times$

2^{N-1} 、 $S \times 2^{N-1}$ 和 $S \times (T-2^N)$ ，或 $S \times 2^{N-1}$ 、 $S \times (T-2^N)$ 和 $S \times 2^{N-1}$ ，或 $S \times (T-2^N)$ 、 $S \times 2^{N-1}$ 和 $S \times 2^{N-1}$ 。

[0136] (i) 在一个示例中，Z 可以用 QT 分割为四个单元，均具有大小为 $S/2 \times T/2$ 。

[0137] (j) 在一个示例中，假设 $2^N < S \leq 2^{N+1}$ ，Z 可以用 QT 分割为四个单元，具有大小为 $2^N \times T/2$ 、 $2^N \times T/2$ 、 $(S-2^N) \times T/2$ 和 $(S-2^N) \times T/2$ ，或 $(S-2^N) \times T/2$ 、 $(S-2^N) \times T/2$ 、 $2^N \times T/2$ 和 $2^N \times T/2$ 。

[0138] (k) 在一个示例中，假设 $2^N < T \leq 2^{N+1}$ ，Z 可以用 QT 分割为四个单元，具有大小为 $S/2 \times 2^N$ 、 $S/2 \times 2^N$ 、 $S/2 \times (T-2^N)$ 和 $S/2 \times (T-2^N)$ ，或 $S/2 \times (T-2^N)$ 、 $S/2 \times (T-2^N)$ 、 $S/2 \times 2^N$ 和 $S/2 \times 2^N$ 。

[0139] (l) 在一个示例中，假设 $2^N < S \leq 2^{N+1}$ 和 $2^M < T \leq 2^{M+1}$ ，Z 可以用 QT 分割为四个单元，具有大小为 $2^N \times 2^M$ 、 $2^N \times 2^M$ 、 $(S-2^N) \times (T-2^M)$ 和 $(S-2^N) \times (T-2^M)$ ，或 $(S-2^N) \times (T-2^M)$ 、 $(S-2^N) \times (T-2^M)$ 、 $2^N \times 2^M$ 和 $2^N \times 2^M$ ，或 $2^N \times (T-2^M)$ 、 $2^N \times (T-2^M)$ 、 $(S-2^N) \times 2^M$ 和 $(S-2^N) \times 2^M$ ，或 $(S-2^N) \times 2^M$ 、 $(S-2^N) \times 2^M$ 、 $2^N \times (T-2^M)$ 和 $2^N \times (T-2^M)$ 。

[0140] (m) 在一个示例中，所有分割单元的宽度/高度应为偶数。如果一个划分结构导致单元的宽度或高度为奇数，则自动禁止这样的划分结构。

[0141] 或者，此外，跳过这种划分结构的信令。

[0142] (n) 在一个示例中，Z 可以用 TT 分割为三个单元。

[0143] 在一个示例中，假设 $3 \times 2^N < S < 3 \times 2^{N+1}$ ，三个单元的大小分别为 $2^N \times T$ 、 $2^{N+1} \times T$ 和 $(S-3 \times 2^N) \times T$ 。

[0144] 在一个示例中，假设 $3 \times 2^N < T < 3 \times 2^{N+1}$ ，三个单元的大小分别为 $S \times 2^N$ 、 $S \times 2^{N+1}$ 和 $S \times (T-3 \times 2^N)$ 。

[0145] (o) 在一个示例中，所有分割单元的宽度和/或高度应为 $K \times M$ 的形式，其中 M 是允许的编码单元/预测单元的最小宽度和/或高度，诸如 4；K 是大于 0 的整数。如果一个划分结构导致单元的宽度或高度不是这样的形式，则自动禁止该划分结构。

[0146] 例如，假设划分结构中分割单元的宽度和高度为 W 和 H，如果 $W < M$ 或 $H < M$ 或 $(W \& (M-1)) \neq 0$ 或 $(H \& (M-1)) \neq 0$ ，则禁止该划分结构。

[0147] 或者，此外，跳过这种划分结构的信令。

[0148] 或者，所有分割的非 ZU 的宽度和/或高度应为 $K \times M$ 的形式，其中 M 是允许的编码单元/预测单元的最小宽度和/或高度，诸如 4。在这种情况下，如果分割的零单元不遵循此限制但非 ZU 遵循此限制，则仍允许划分结构。

[0149] 示例 8. ZU 的分割信令方法与正常 CU 的分割信令方法相同。

[0150] a. 在一个示例中，可以使用不同的上下文来编码 ZU 或非 ZU。

[0151] b. 或者，对于 ZU，仅允许正常 CU 的部分分割方法。

[0152] i. 对 ZU 所允许的正常 CU 的分割方法的子集由 ZU 大小和/或图片/条带/片边界位置（底部、右部、右下部等）和/或条带类型确定。

[0153] ii. 在一个示例中，对于 ZU 只允许 QT 和 BT 划分结构。

[0154] iii. 或者，此外，在 ZU 分割信息中不发信号通知是否使用 TT 以及如何使用 TT（以及除 QT/BT 之外的其它种类的划分结构）。

[0155] iv. 或者，此外，ZU 的分割信令方法仍然与正常 CU 的分割信令方法保持相同，然而，TT（或其它类型的划分结构）的指示的上下文可以进一步取决于当前块是否是 ZU。

[0156] 示例 9. 在一个实施例中，宽度和高度均等于 $(2^N \times 2^M)$ 的块也可以定义为 ZU。

[0157] (a) 在这种情况下，与其它 ZU（其中宽度或高度不等于 2^N ）一样，如第 2 项所述，不调

用变换、逆变换、量化和去量化操作。

[0158] (b) 假设CU的大小是 $W \times H$, 则当以下条件成立时, CU被视为ZU,

[0159] (i) $W \geq T0$ 且 $H \geq T1$ 。 $T0$ 和/或 $T1$ 是诸如128或256的整数。

[0160] (ii) $W \geq T0$ 或 $H \geq T1$ 。 $T0$ 和/或 $T1$ 是诸如128或256的整数。

[0161] (iii) $W \times H \geq T$ 。 T 是诸如16384或65536的整数。

[0162] 示例10. 在一个实施例中, ZU必须是帧间编码的块。

[0163] (a) 在一个示例中, ZU只能用于P/B图片/条带中。

[0164] (b) 在一个示例中, 对于ZU, 不信令通知prediction_mode_flag, 并且总是被推断为被帧间编码。

[0165] (c) 在一个示例中, ZU可以与用于运动预测的几何划分结合使用(参考[6]), 使得帧间预测可以生成更好地适合于视频的运动和纹理的预测器。

[0166] (i) 在一个示例中, 允许的划分类型可以取决于父划分的形状。例如, 如果块的横纵比大于或等于4(宽度)比1(高度), 应禁止接近水平线的角度。如果横纵比小于或等于4(宽度)比1(高度), 则应禁止接近垂直线的角度。

[0167] (ii) 在一个示例中, 仅Merge模式可用于子运动划分以节省开销比特。

[0168] (iii) 在一个示例中, Merge模式和常规AMVP模式都可用于子运动划分。是否允许常规AMVP模式也可以通过条带报头来信令通知, 使得有效编码模式可以适应于视频内容。

[0169] (d) 在一个示例中, 可以使用运动单元的更小的组来更好地预测ZU。例如, 可以允许 2×2 ATMVP和 2×2 仿射模式用于ZU。这也有助于当ZU的宽度或高度为2时, 最小化角落情况的影响。

[0170] 示例11. 在一个示例中, ZU必须被划分为I条带或帧内编码的图片。

[0171] (a) 在一个示例中, ZU的宽度或高度不是 $2N$ 的形式。

[0172] (b) 在一个示例中, 当以下条件成立时, CU被视为ZU,

[0173] i. $W \geq T0$ 且 $H \geq T1$ 。 $T0$ 和/或 $T1$ 是诸如128或256的整数。

[0174] ii. $W \geq T0$ 或 $H \geq T1$ 。 $T0$ 和/或 $T1$ 是诸如128或256的整数。

[0175] iii. $W \times H \geq T$ 。 T 是诸如16384或65536的整数。

[0176] 示例12. 对于ZU, 环路过滤的强度应设置为强。

[0177] (a) 在一个示例中, 用于去方块(de-blocking)的强滤波器应该被用于ZU。

[0178] (b) 在一个示例中, 强双边滤波器应该被用于ZU。

[0179] (c) 在一个示例中, 可以使用运动补偿平滑滤波器(例如, 重叠块运动补偿)。

[0180] 示例13. 对于ZU和正常CU, 可以以不同方式实现启用/禁用适应性环路滤波器(ALF)。

[0181] (a) 在一个示例中, 如果CTU是ZU, 则整个ZU被控制为整体上进行ALF或不进行ALF, 而不是分割成子块来进行ALF开/关控制。相应地, 对于这样的CTU, 仅信令通知一个alf_control_flag。

[0182] (b) 在另一个示例中, 如果CTU是ZU, 则ZU被分割为N个子块以进行ALF开/关控制。该分割不依赖于CTU内的ZU或CU划分。相应地, 对于这样的CTU, 信令通知多个(例如, 高达N个)alf_control_flag。

[0183] (i) 例如, 如果CTU大小是 $W \times H$, 则子块大小是 $w \times h$ 。

[0184] (1) 在一个示例中,CTU被分割为 $(W+w-1)/w$ 列和 $(H+h-1)/h$ 行的子块(所有除法均是在C语言中定义的整除)。最后一行/最后一列中的子块可能比其它块更小。

[0185] (2) 在一个示例中,CTU被分割为 W/w 列和 H/h 行的子块(所有除法均是在C语言中定义的整除)。最后一行/最后一列中的子块可能比其它块更大。

[0186] 图18A和18B示出了将ZU块划分成子块以用于自适应环路控制(ALF)开/关控制的示例。

[0187] (c) 如果一个ZU位于图像边界处,则ALF将被自动禁用而无需任何信令通知。

[0188] (d) 上述方法可以扩展到需要块级开/关控制标志的信令通知的其它种类的滤波方法。

[0189] 示例14. 对于位于图像边界处的CU/PU/TU,可以自动禁用ALF或需要块级开/关控制标志的其它种类的滤波方法。

[0190] (a) 在这种情况下,对于这些块的开/关控制标志的信令被跳过。

[0191] 上述示例可以结合在下面描述的方法的上下文中——例如,方法1400和1500,其可以在视频解码器和/或视频编码器处实现。

[0192] 图14示出了用于视频编码的示例性方法的流程图,其可以在视频编码器中实现。方法1400包括,在步骤1410处,确定视频数据块的维度。

[0193] 方法1400包括,在步骤1420处,在确定至少一个维度是非二的幂时,将视频数据块作为零单元(ZU)块来信令通知,其是不可变换的。

[0194] 在一些实施例中,非二的幂是不能以 2^N 的形式表示的任何非零数字。例如,不包括二的幂的整数(例如,1,3,5,6,7,9,10,11,12,13,14,15,17,18,...)的每个都是非二的幂。

[0195] 在一些实施例中,可以定义不可变换,使得不对零单元调用变换、逆变换、量化和去量化操作。例如,零单元的不可变换属性是其被推断为用跳过模式进行编码,并且因此,不需要明确地用信令通知跳过模式。在其它实施例中,可以在示例3的上下文中定义不可变换,使得尽管可能存在非零残差,但对于零单元没有定义变换和逆变换操作。

[0196] 图15示出了用于视频编码的另一示例性方法的流程图,该方法可以在视频解码器中实现。该流程图包括与图14中所示并且在上面描述的类似的一些特征和/或步骤。本章节中可能未单独描述这些特征和/或步骤中的至少一些。

[0197] 方法1500包括,在步骤1510处,接收对应于视频数据块的比特流。

[0198] 方法1500包括,在步骤1520处,接收信令,其指示视频数据块是零单元(ZU)块,该零单元(ZU)块是不可变换的,并且具有非二的幂的至少一个维度。

[0199] 方法1500包括,在步骤1530处,基于信令解码比特流以重建视频数据块。

[0200] 在一些实施例中,方法1400和1500,以及如在示例1的上下文中所描述的,可以进一步包括视频数据块的维度是偶数,具有 $2N$ 形式,或者具有 $2^k N$ 的形式,其中 $k=1,2,3,4\dots$ 。在其它实施例中,信令可以不包括合并索引或跳过标记、和/或不包括prediction_mode_flag、和/或包括ZU块的至少一个维度的最大值或最小值。在示例中,信令在视频参数集(VPS)、序列参数集(SPS)、图片参数集(PPS)、条带报头、编码树单元(CTU)或编码单元(CU)中。

[0201] 在一些实施例中,ZU块的运动信息从大小为 $2^N \times 2^M$ 的相邻块的运动信息继承。

[0202] 在一个实施例中,并且如在示例7的上下文中所描述的,ZU块被分割为两个或更多

个单元。在示例中,两个或更多个单元中的至少一个是零单元。在另一个示例中,在两个或更多个单元中的至少一个是尺寸为 $2^N \times 2^M$ 的编码单元(CU)。

[0203] 在一些实施例中,并且如在示例10的上下文中所描述的,ZU块的环路滤波强度被设置为强。在HEVC标准的示例中,“强”滤波器修改所感兴趣像素的每一侧上的三个像素,而“弱”滤波器修改每一侧上的一个或两个像素。例如,环路滤波包括双边滤波、去方块滤波和/或使用运动补偿的平滑滤波器。

[0204] 4. 所公开的技术的示例实现方式

[0205] 图16是示出可以用于实现本公开的技术的各个部分的计算机系统或其它控制设备1600的架构的示例的框图,本公开的技术包括(但不限于)方法1400和1500。在图16中,计算机系统1600包括经由互连1625连接的一个或多个处理器1605和存储器1610。互连1625可以代表通过适当的桥接器、适配器连接的任何一个或多个单独的物理总线、点对点连接或两者。因此,互连1625可以包括例如,系统总线、外围组件互连(PCI)总线、双向传输总线(HyperTransport)或工业标准架构(ISA)总线、小型计算机系统接口(SCSI)总线、通用串行总线(USB)、IIC(I2C)总线或电气和电子工程师协会(IEEE)标准674总线,有时也称为“Firewire”。

[0206] (一个或多个)处理器1605可以包括中央处理单元(CPU)以控制例如主计算机的整体操作。在某些实施例中,(一个或多个)处理器1605通过执行存储器1610中存储的软件或固件来实现这一点。(一个或多个)处理器1605可以是或可以包括一个或多个可编程通用或专用微处理器、数字信号处理器(DSP)、可编程控制器、专用集成电路(ASIC)、可编程逻辑器件(PLD)等,或这些器件的组合。

[0207] 存储器1610可以是或可以包括计算机系统的主存储器。存储器1610表示任何合适形式的随机存取存储器(RAM)、只读存储器(ROM)、闪存存储器等,或其组合。在使用中,存储器1610可以包含机器指令集,当由处理器1605执行该机器指令集时,使得处理器1605进行操作以实现本公开技术的实施例。

[0208] 通过互连1625也连接到(一个或多个)处理器1605的是(可选的)网络适配器1615。网络适配器1615向计算机系统1600提供与远程设备通信的能力,远程设备例如存储器客户端和/或其它存储器服务器,并且网络适配器1615可以是例如以太网适配器或光纤信道适配器。

[0209] 图17示出了可用于实现本公开技术的各个部分的移动设备1700的示例实施例的框图,包括(但不限于)方法700、750、800、850、900、950、600、1400和1500。移动设备1700可以是膝上型电脑、智能电话、平板电脑、便携式摄像机或能够处理视频的其它类型的设备。移动设备1700包括用于处理数据的处理器或控制器1701,以及与处理器1701通信以存储和/或缓冲数据的存储器1702。例如,处理器1701可以包括中央处理单元(CPU)或微控制器单元(MCU)。在一些实现方式中,处理器1701可以包括现场可编程门阵列(FPGA)。在一些实现方式中,移动设备1700包括用于智能电话设备的各种可视和/或通信数据处理功能的图形处理单元(GPU)、视频处理单元(VPU)和/或无线通信单元或与之通信。例如,存储器1702可以包括并存储处理器可执行代码,其在由处理器1701执行时配置移动设备1700以执行各种操作,例如接收信息、命令和/或数据,处理信息和数据,并且将处理后的信息/数据发送或提供给诸如致动器或外部显示器的另一个设备。

[0210] 为了支持移动设备1700的各种功能,存储器1702可以存储信息和数据,诸如指令、软件、值、图像以及由处理器1701处理或参考的其它数据。例如,各种类型的随机存取存储器(RAM)设备、只读存储器(ROM)设备、闪存设备和其它合适的存储介质可已被用于实现存储器1702的存储功能。在一些实现方式中,移动设备1700包括输入/输出(I/O)单元1703,以将处理器1701和/或存储器1702与其它模块、单元或设备接口。例如,I/O单元1703可以利用与典型数据通信标准(例如,在云中的一个或多个计算机与用户设备之间的)兼容的各种类型的无线接口,以将处理器1701与存储器1702接口。在一些实现方式中,移动设备1700可以经由I/O单元1703使用有线连接与其它设备接口。移动设备1700还可以与其它外部接口(例如数据存储器和/或视觉或音频显示设备1704接口,以取回和传送数据和信息,该数据和信息可由处理器处理、存储在存储器中或在显示设备1704或外部设备的输出单元上展示。例如,显示设备1704可以显示包括块(CU、PU或TU)的视频帧,该块基于是否使用运动补偿算法并根据所公开的技术对块进行编码来应用块内复制。

[0211] 在一些实施例中,视频解码器设备可以实现使用如本文中所描述的零单元的方法以用于视频解码。该方法的各种特征可以类似于上述方法1500。

[0212] 在一些实施例中,视频解码方法可以使用解码设备来实现,该解码设备在如图16和图17中所述的硬件平台上实现。

[0213] 可以使用以下示例解决方案的列表来捕获本文描述的一些实施例。在解决方案中,接收操作可以包括通过网络连接或从存储比特流的存储单元(存储器、光盘驱动器等)接收比特流。

[0214] 1.一种视频编码方法,包括:基于至少一个维度是非二的幂数来确定视频数据块是零单元(ZU)块;响应于确定视频数据块是ZU块,对所有残差被设为零的ZU块进行编码;以及在比特流中信令通知编码的ZU块。

[0215] 2.一种视频解码方法,包括:接收对应于视频数据块的比特流;由于块的至少一个维度为非二的幂数,确定视频数据块被编码为所有残差设为零的零单元(ZU)块;以及基于决定来解码视频数据块。

[0216] 3.如解决方案1或2所述的方法,其中,视频数据块的维度的任一个是 2^N 形式的偶数。

[0217] 4.如解决方案1或2所述的方法,其中,视频数据块的维度的任一个为 2^K 形式,其中K是整数。

[0218] 5.如解决方案1或2所述的方法,其中,不在ZU块上进行变换操作或逆变换操作。

[0219] 6.如解决方案1或2所述的方法,其中,不在ZU块上进行量化操作或去量化操作。

[0220] 7.如解决方案1或2所述的方法,其中,信令不包括任何指示预测模式的语法元素。

[0221] 8.如解决方案7所述的方法,其中,推断ZU块以Skip模式编码。

[0222] 9.如解决方案7或8所述的方法,其中,信令不包括merge索引。

[0223] 10.如解决方案1或2所述的方法,其中,ZU块的运动信息从相邻块的运动信息继承。

[0224] 11.如解决方案1或2所述的方法,其中,ZU块的运动信息从大小为 $2^N \times 2^M$ 的相邻块的运动信息继承。

[0225] 12.如解决方案11所述的方法,其中,相邻块的侧边与接触图片边界的ZU块的侧边

平行。

[0226] 13.如解决方案11所述的方法,其中,ZU块与相邻块具有连续性。

[0227] 14.如解决方案1或2所述的方法,其中,信令不包括cbf_flag并且ZU块的cbf_flag被推断为零。

[0228] 15.如解决方案1或2所述的方法,其中,信令不包括残差信息。

[0229] 16.如解决方案1或2所述的方法,其中,ZU块具有残差并且未在ZU块上进行变换或逆变换操作。

[0230] 17.如解决方案1或2所述的方法,其中,用transform-skip模式编码ZU块中的残差。

[0231] 18.如解决方案1或2所述的方法,其中,信令不包括transform-skip标志。

[0232] 19.如解决方案1或2所述的方法,其中,ZU块是在非对称二叉树 (ABT) 划分中分割的编码单元 (CU)。

[0233] 20.如解决方案1或2所述的方法,其中,ZU块是图片、条带或片边界处的块。

[0234] 21.如解决方案1或2所述的方法,其中,在图片、条带或片的边界处的每个编码单元是ZU块。

[0235] 22.如解决方案19所述的方法,其中,信令包括区分CU和ZU块的标志。

[0236] 23.如解决方案1或2所述的方法,其中,信令包括ZU块的至少一个维度的最大值或最小值。

[0237] 24.如解决方案23所述的方法,其中,信令在视频参数集 (VPS)、序列参数集 (SPS)、图片参数集 (PPS)、条带报头、编码树单元 (CTU) 或编码单元 (CU) 中。

[0238] 25.如解决方案1或2所述的方法,其中,ZU块可以仅是帧间编码的块。

[0239] 26.如解决方案25所述的方法,其中,ZU块可以仅在P/B图片/条带中使用。

[0240] 27.如解决方案25所述的方法,其中,信令不包括预测模式标志。

[0241] 28.如解决方案25所述的方法,其中,ZU块结合几何划分实现。

[0242] 29.如解决方案28所述的方法,其中,几何划分包括基于父划分的形状的划分。

[0243] 30.如解决方案28所述的方法,其中,几何划分包括仅使用Merge模式。

[0244] 31.如解决方案28所述的方法,其中,信令包括指示用于自运动划分的Merge模式或常规AMVP模式的一个的标志。

[0245] 32.如解决方案25所述的方法,其中,使用2x2ATMVP模式预测ZU块。

[0246] 33.如解决方案25所述的方法,其中,使用2x2仿射模式预测ZU块。

[0247] 如上所述的示例1-7提供了解决方案1至33的附加特征和变化。

[0248] 34.一种视频编码方法,包括:基于块的维度是二的幂数,确定视频数据块是零单元 (ZU) 块;基于决定该块为ZU块,对比特流中的视频数据块进行编码;以及在比特率中信令通知ZU块。

[0249] 35.一种视频解码方法,包括:接收对应于视频数据块的比特流;基于块的维度为二的幂数来确定视频数据块为零单元 (ZU) 块;以及基于信令解码比特流以重建视频数据块。

[0250] 36.如解决方案34或35所述的方法,其中,不在ZU块上进行变换操作或逆变换操作。

- [0251] 37.如解决方案34或35所述的方法,其中,不在ZU块上进行量化操作或去量化操作。
- [0252] 38.如解决方案34到37的任一个所述的方法,其中,确定还包括由于视频数据块的样本的总数大于或等于阈值数T,确定视频数据块为ZU块。
- [0253] 39.如解决方案34到37的任一个所述的方法,其中,确定还包括由于维度进一步大于视频数据块的宽度的第一阈值或者视频数据块的高度的第二阈值,确定视频数据块是ZU块。
- [0254] 如上所述的示例8提供了解决方案34至39的附加特征和变化。
- [0255] 40.一种视频编码方法,包括:基于具有小于或等于第一整数的宽度并具有小于或等于第二整数的高度来确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,基于决定该块为ZU块,将视频数据块编码为ZU块;以及在比特流中信令通知ZU块。
- [0256] 41.一种视频解码方法,包括:接收对应于具有宽度和高度的视频数据块的比特流;基于具有大于或等于第一整数的宽度并具有大于或等于第二整数的高度来确定视频数据块将被解码为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。
- [0257] 42.如解决方案40或41所述的方法,其中,第一整数和第二整数为128或256的一个。
- [0258] 43.一种视频编码方法,包括:基于具有大于或等于第一整数的宽度并且具有大于或等于第二整数的高度来确定具有该宽度和高度的视频数据块将被编码为零单元(ZU)块,基于决定该块为ZU块,将视频数据块编码为ZU块;以及在比特流中信令通知ZU块。
- [0259] 44.一种视频解码方法,包括:接收对应于具有宽度和高度的视频数据块的比特流;基于具有大于或等于第一整数的宽度或具有大于或等于第二整数的高度来确定视频数据块将被解码为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。
- [0260] 45.如解决方案43或44所述的方法,其中,第一整数和第二整数为128或256的一个。
- [0261] 46.一种视频编码方法,包括:基于具有大于或等于整数的宽度和高度的乘积来确定具该宽度和高度的视频数据块将被编码为零单元(ZU)块,基于决定该块为ZU块,将视频数据块编码为ZU块;以及在比特流中信令通知ZU块。
- [0262] 47.一种视频解码方法,包括:接收对应于具有宽度和高度的视频数据块的比特流;基于具有大于或等于整数的宽度和高度的乘积来确定视频数据块将被解码为零单元(ZU)块;以及基于信令解码比特流以重建视频数据块。
- [0263] 48.如解决方案46或47所述的方法,其中,整数是16384或65536的一个。
- [0264] 如上所述的示例1-8提供了解决方案40至48的附加特征和变化。
- [0265] 49.一种视频系统中的装置,包括处理器和其上具有指令的非瞬时性的存储器,其中,当由处理器执行指令时,使得处理器实现如解决方案1到48中的任一个所述的方法。
- [0266] 50.一种储存在非瞬时性的计算机可读介质上的计算机程序产品,计算机程序产品包括用于执行解决方案1到48中的任一个中的方法的程序代码。
- [0267] 51.如本文件中所描述的方法、装置或系统。
- [0268] 从前述内容可以理解,本文已经出于说明的目的描述了本公开技术的具体实施例,但是可以在不脱离本发明范围的情况下进行各种修改。因此,本公开的技术除了所附权

利要求外不受限制。

[0269] 本专利文件中描述的主题和功能操作的实现方式可以以各种系统、数字电子电路实现,或者以计算机软件、固件或硬件实现,包括说明书中公开的结构及其结构等同物,或者以它们中的一个或多个的组合实现。公开和其它实施例可以实现为一个或多个计算机程序产品,即,在有形且非瞬时性的计算机可读介质上编码的一个或多个计算机程序指令模块,用于由数据处理装置执行或控制数据处理装置的操作。计算机可读介质可以是机器可读存储设备、机器可读存储基板、存储器设备、影响机器可读传播信号的物质组合、或者它们中的一个或多个的组合。术语“数据处理单元”或“数据处理装置”涵盖用于处理数据的所有装置、设备和机器,包括例如可编程处理器、计算机或多个处理器或计算机。除了硬件之外,该装置还可以包括为所讨论的计算机程序创建执行环境的代码,例如,构成处理器固件、协议栈、数据库管理系统、操作系统、或者它们中的一个或多个的组的代码。

[0270] 计算机程序(也称为程序、软件、软件应用、脚本或代码)可以用任何形式的编程语言编写,包括编译或解释语言,并且可以以任何形式来部署计算机程序,包括独立程序或适合在计算环境中使用的模块、组件、子例程或其它单元。计算机程序并不必需对应于文件系统中的文件。程序可以存储在文件的保存其它程序或数据(例如,存储在标记语言文件中的一个或多个脚本)的部分中,存储在专用于所讨论的程序的单个文件中,或存储在多个协调文件中(例如,存储一个或多个模块、子程序或代码部分的文件)。可以部署计算机程序以在一个计算机上或在位于一个站点上或分布在多个站点上并通过通信网络互连的多个计算机上执行。

[0271] 本说明书中描述的过程和逻辑流程可以由执行一个或多个计算机程序的一个或多个可编程处理器执行,以通过对输入数据进行操作并生成输出来执行功能。过程和逻辑流程也可以由专用逻辑电路执行,并且装置也可以实现为专用逻辑电路,例如FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0272] 举例来说,适合于执行计算机程序的处理器包括通用和专用微处理器、以及任何种类的数字计算机的任何一个或多个处理器。通常,处理器将从只读存储器或随机存取存储器或两者接收指令和数据。计算机的基本元件是用于执行指令的处理器和用于存储指令和数据的一个或多个存储器设备。通常,计算机还将包括或可操作地耦合到用于存储数据的一个或多个大容量存储设备,例如磁盘、磁光盘或光盘,以从该一个或多个大容量存储设备接收数据,或将数据传输到该一个或多个大容量存储设备,或者既接收又传递数据。然而,计算机不需要具有这样的设备。适用于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储器设备,包括例如半导体存储器设备,例如EPROM、EEPROM和闪存设备。处理器和存储器可以由专用逻辑电路补充或并入专用逻辑电路中。

[0273] 说明书和附图旨在被认为仅是示例性的,其中示例性意味着示例。如这里所使用的,单数形式的“一”,“一个”和“该”旨在也包括复数形式,除非上下文另有明确说明。另外,除非上下文另有明确说明,否则“或”的使用旨在包括“和/或”。

[0274] 虽然本专利文件包含许多细节,但这些细节不应被解释为对任何发明或可要求保护的范围的限制,而是作为特定于特定发明的特定实施例的特征的描述。在本专利文件中,在单独的实施例的上下文中描述的某些特征也可以在单个实施例中组合实现。相反,在单

个实施例的上下文中描述的各种特征也可以单独地或以任何合适的子组合在多个实施例中实现。此外,尽管上面的特征可以描述为以某些组合起作用并且甚至最初如此要求权利保护,但是在某些情况下,可以从所要求保护的组合中去除来自该组合的一个或多个特征,并且所要求保护的组合可以指向子组合或子组合的变型。

[0275] 类似地,虽然在附图中以特定顺序描绘了操作,但是这不应该被理解为要求以所示的特定顺序或按顺序执行这样的操作,或者执行所有示出的操作,以实现期望的结果。此外,在本专利文件中描述的实施例中的各种系统组件的分离不应被理解为在所有实施例中都要求这种分离。

[0276] 仅描述了几个实现方式和示例,并且可以基于本专利文件中描述和示出的内容来做出其它实现方式、增强和变型。

300

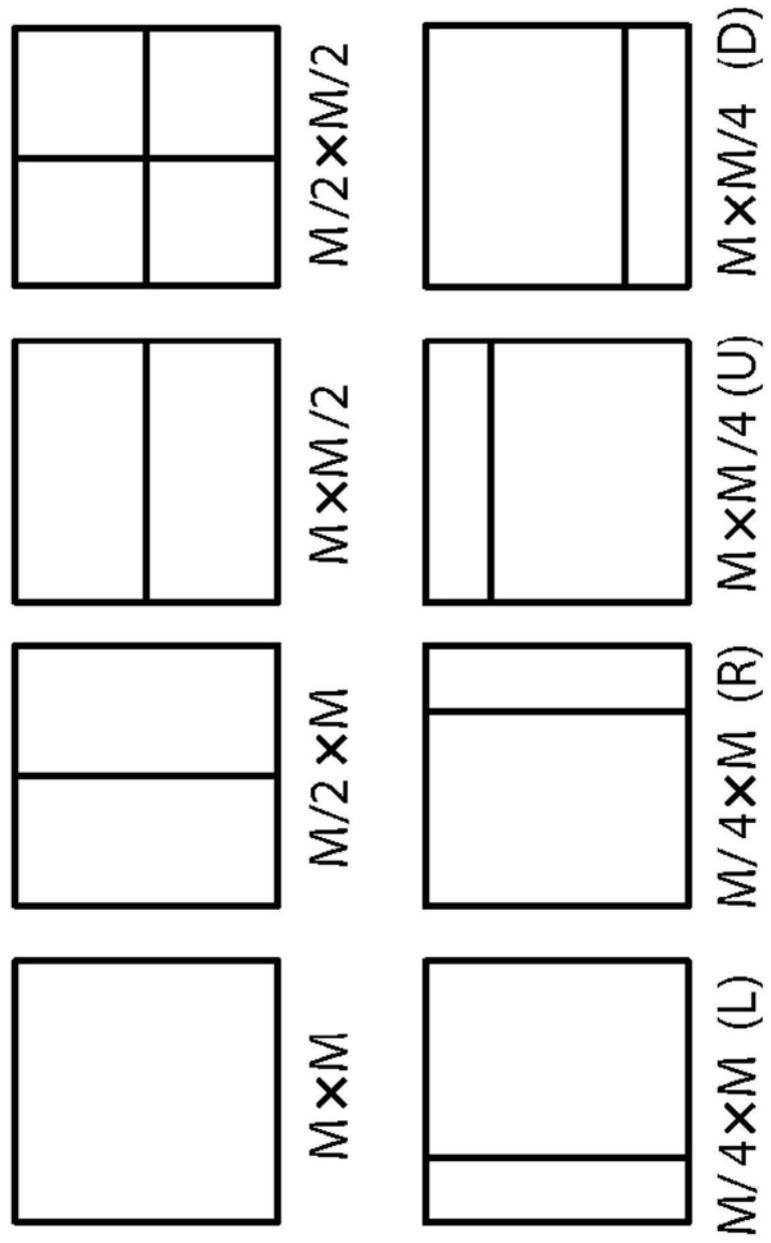


图3

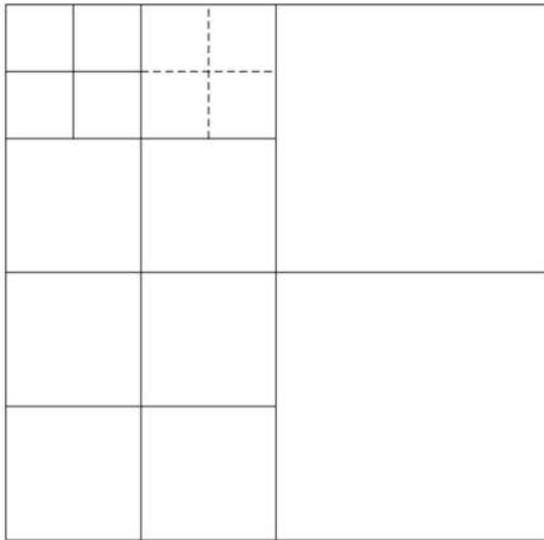


图 4A

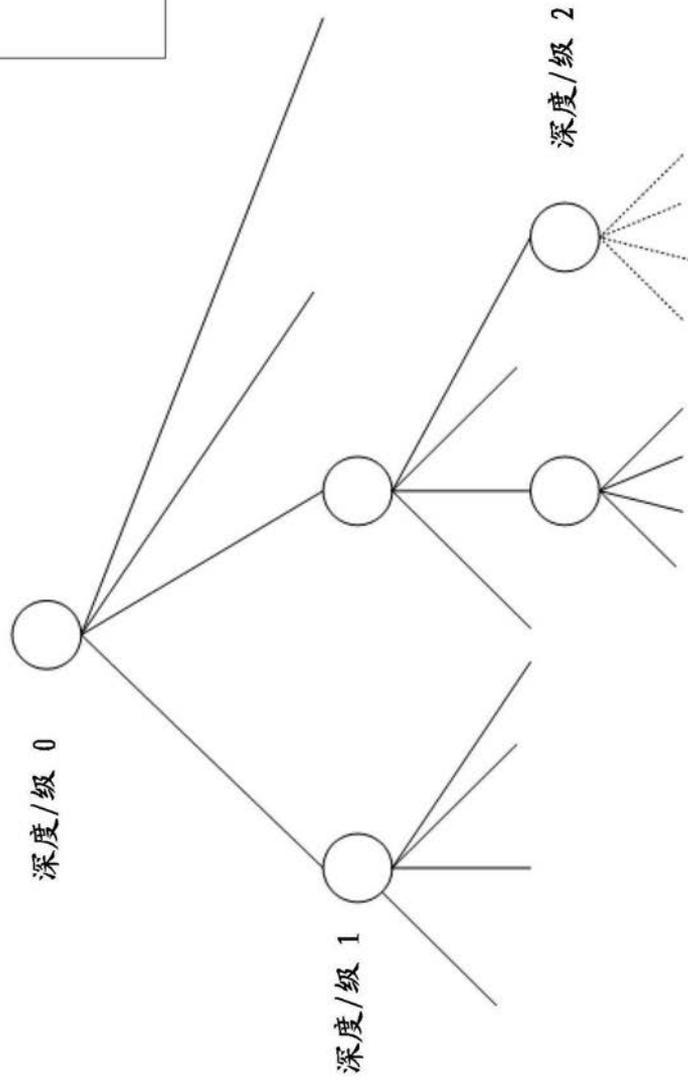


图 4B

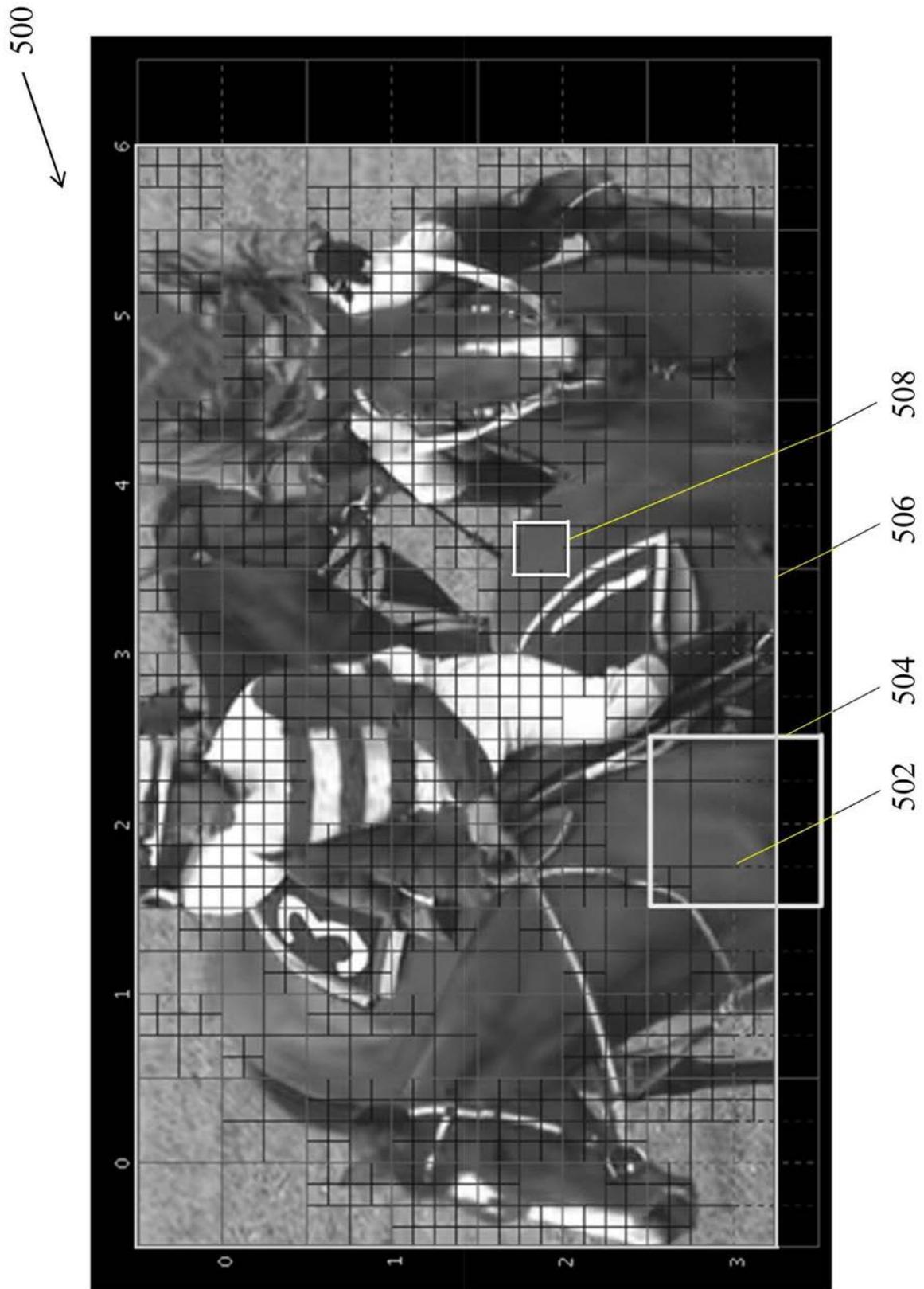


图5

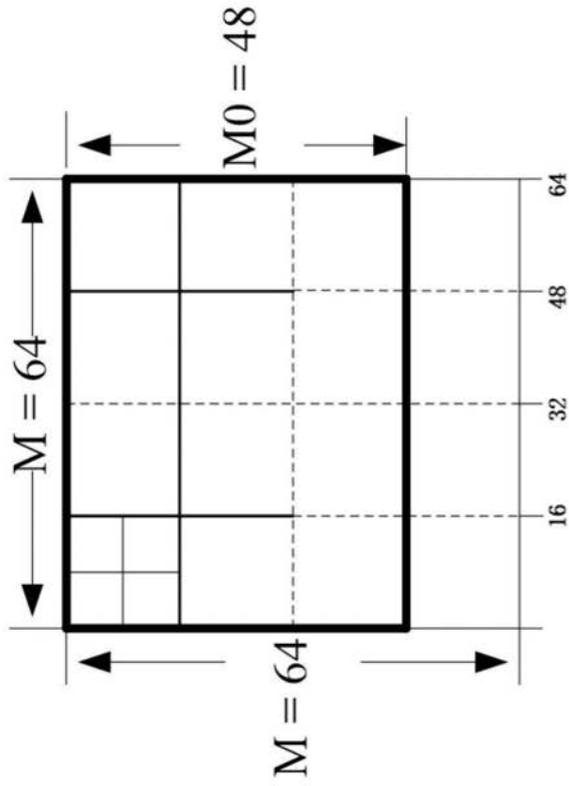


图 6A

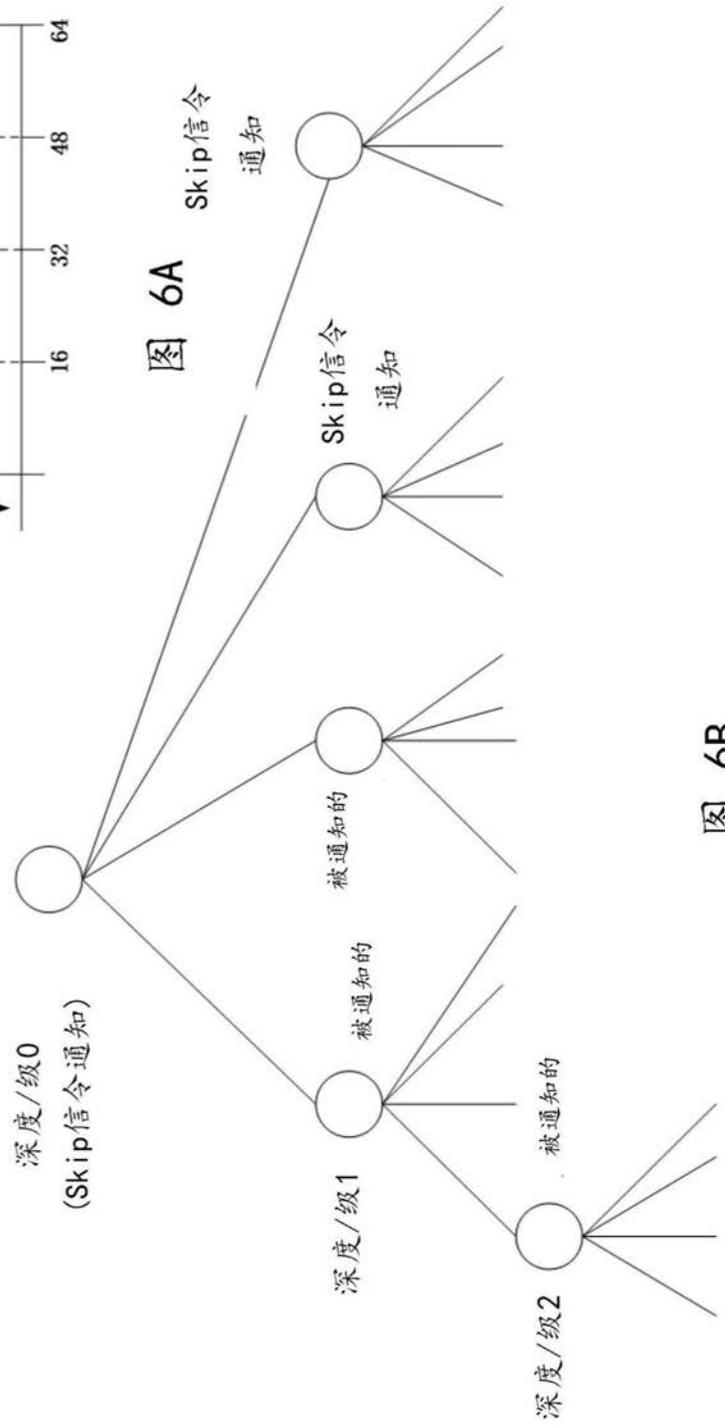


图 6B

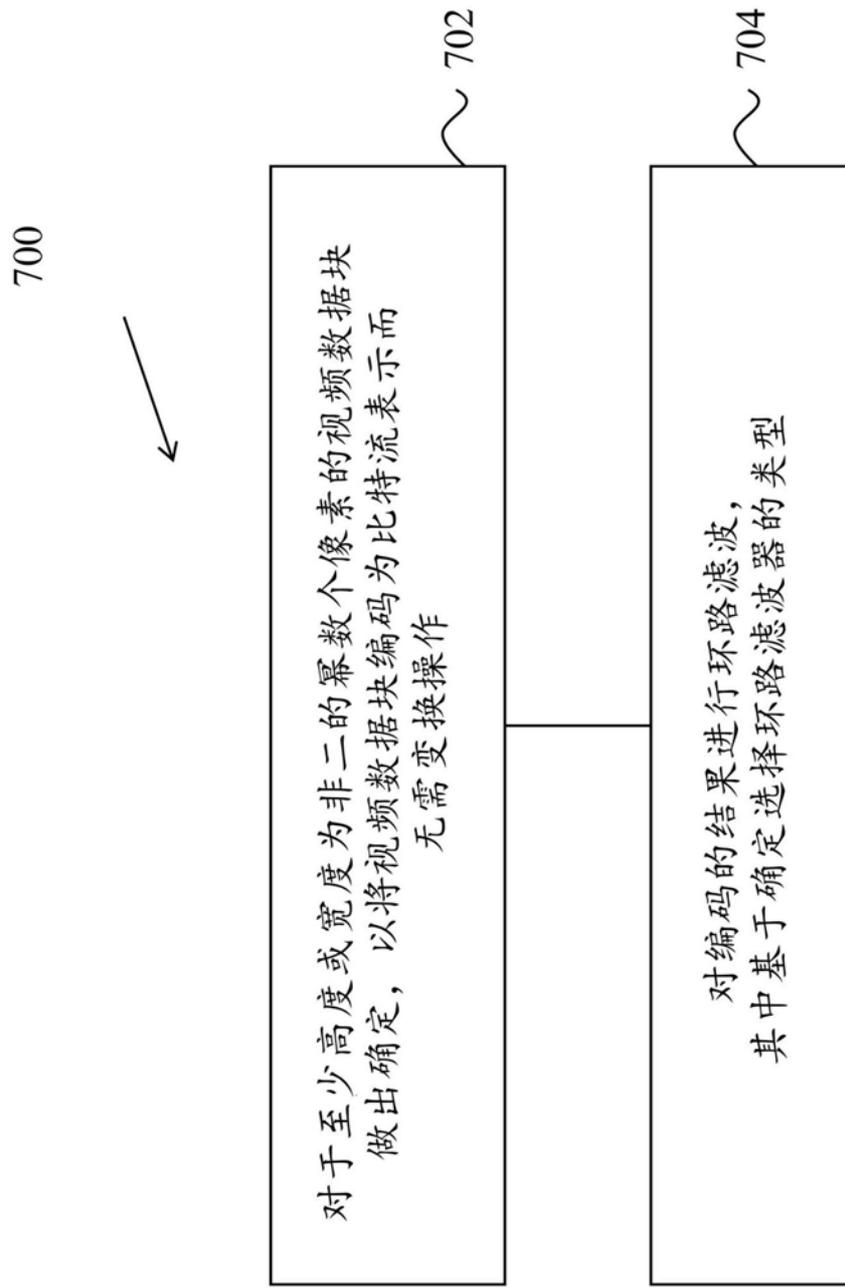


图7A

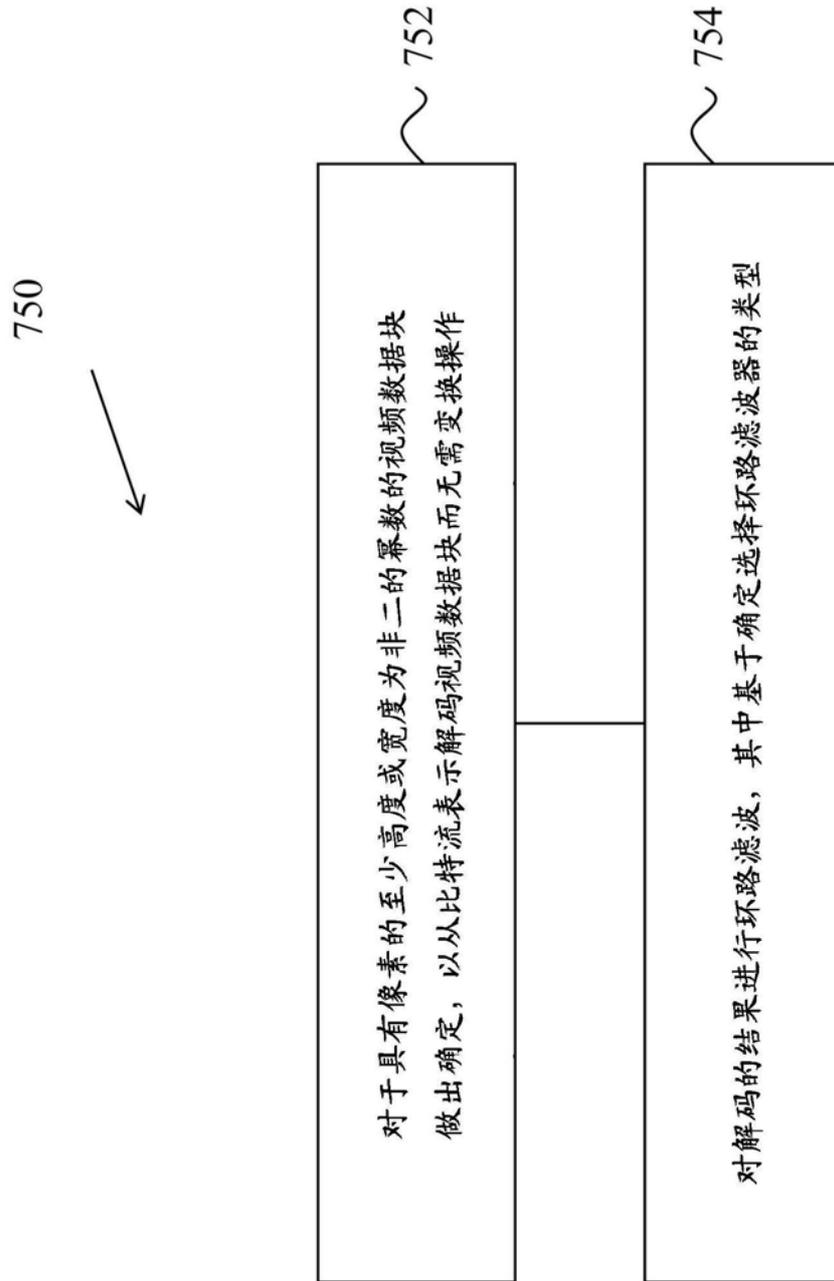


图7B

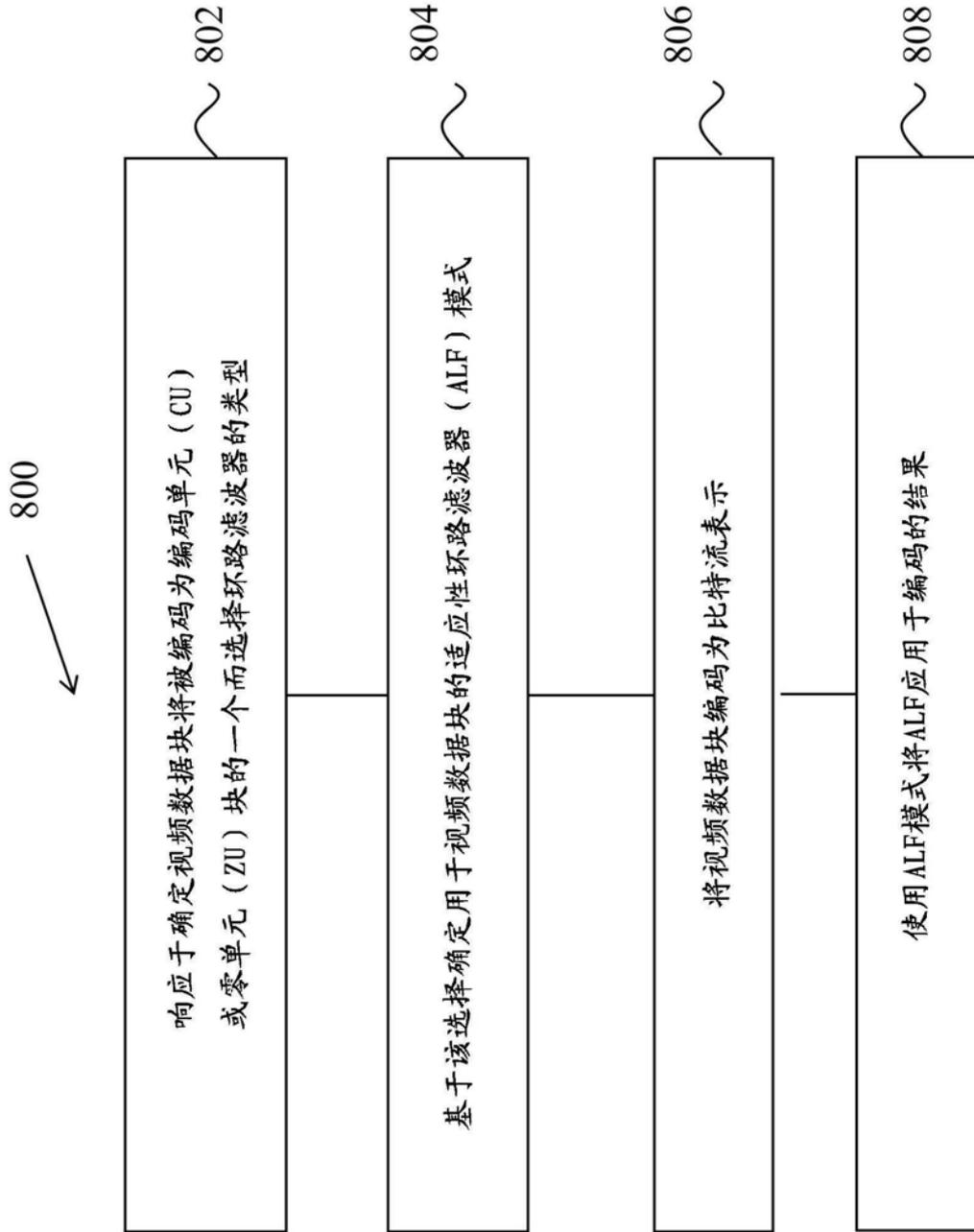


图8A

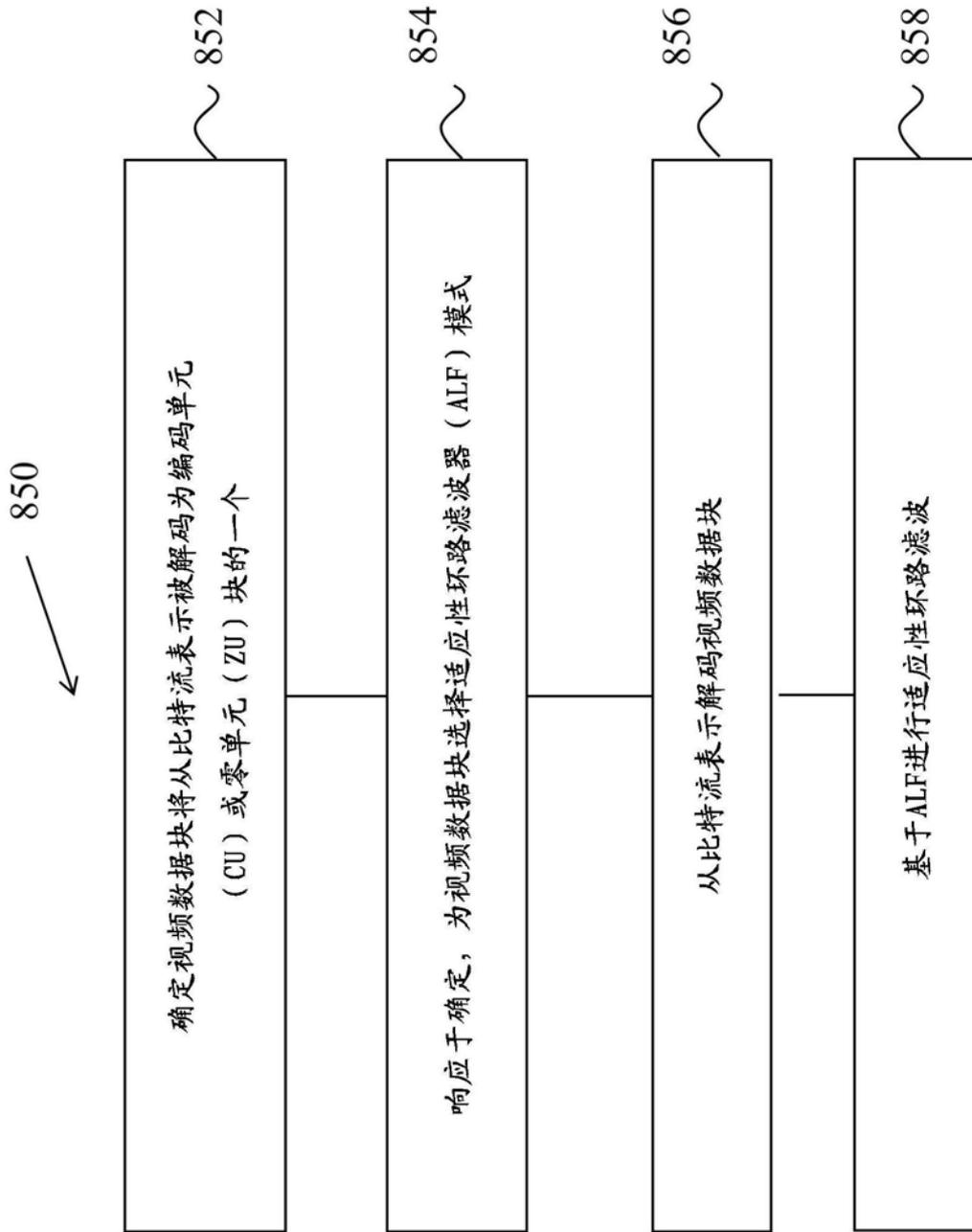


图8B

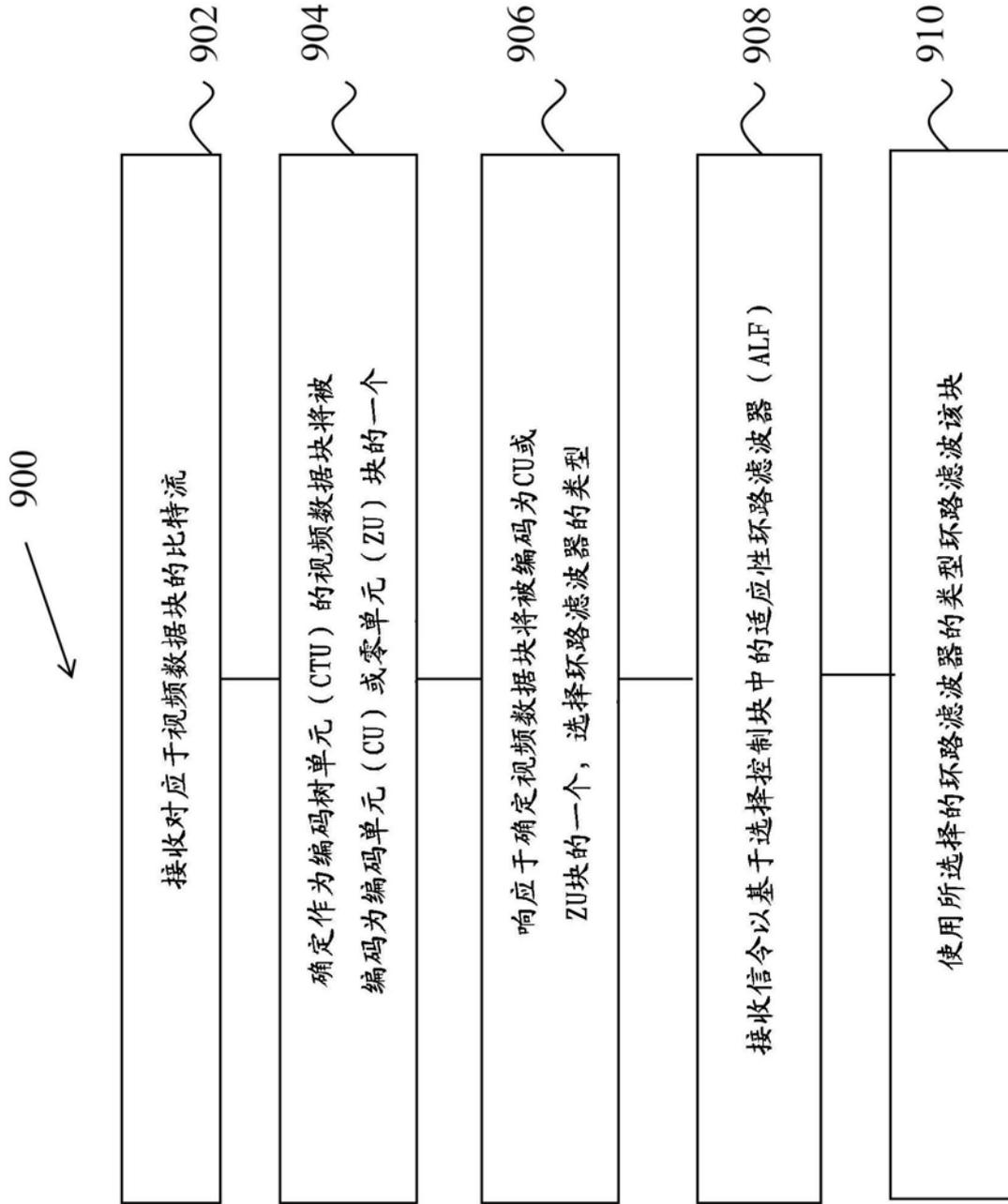


图9A

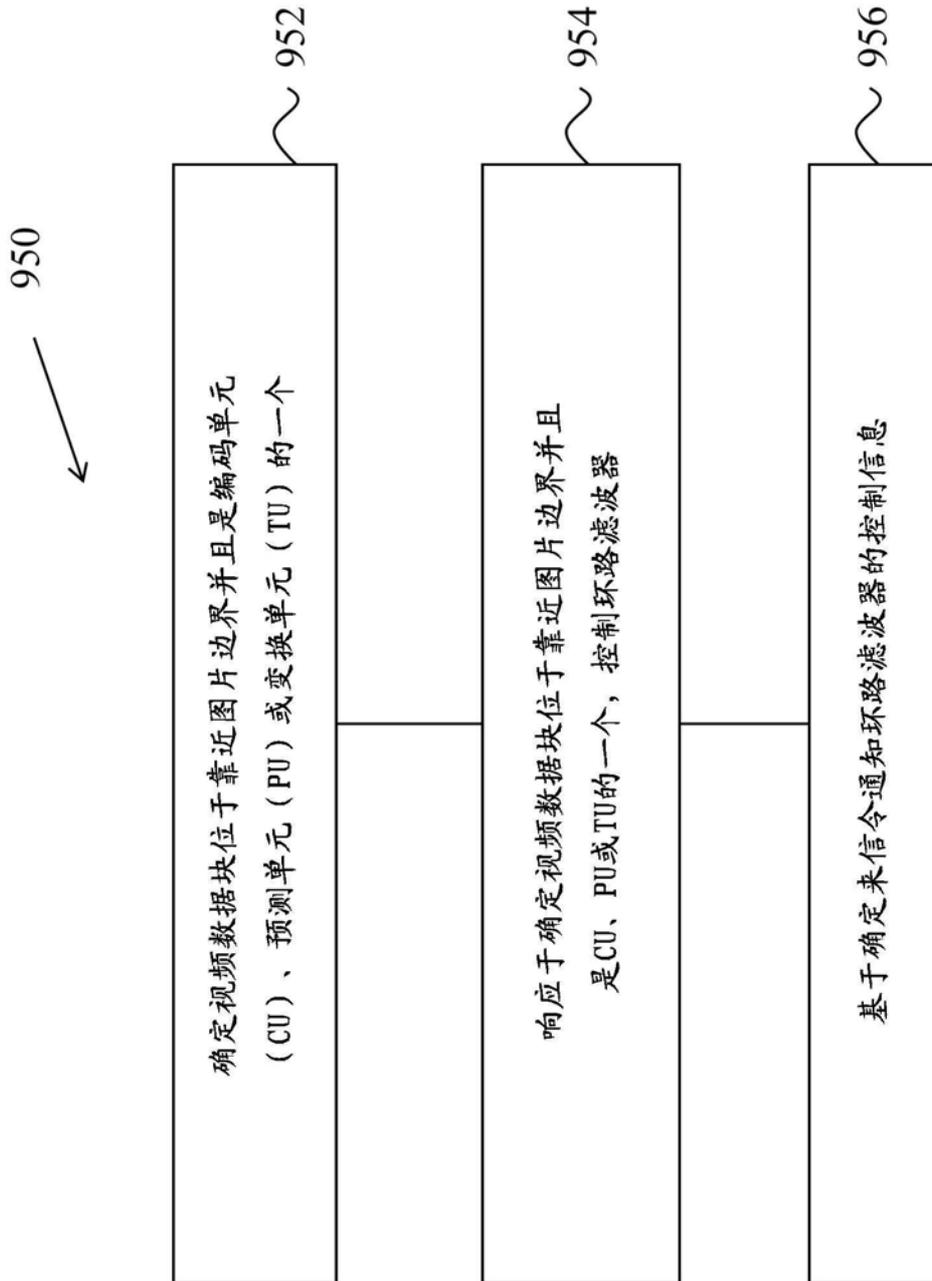


图9B

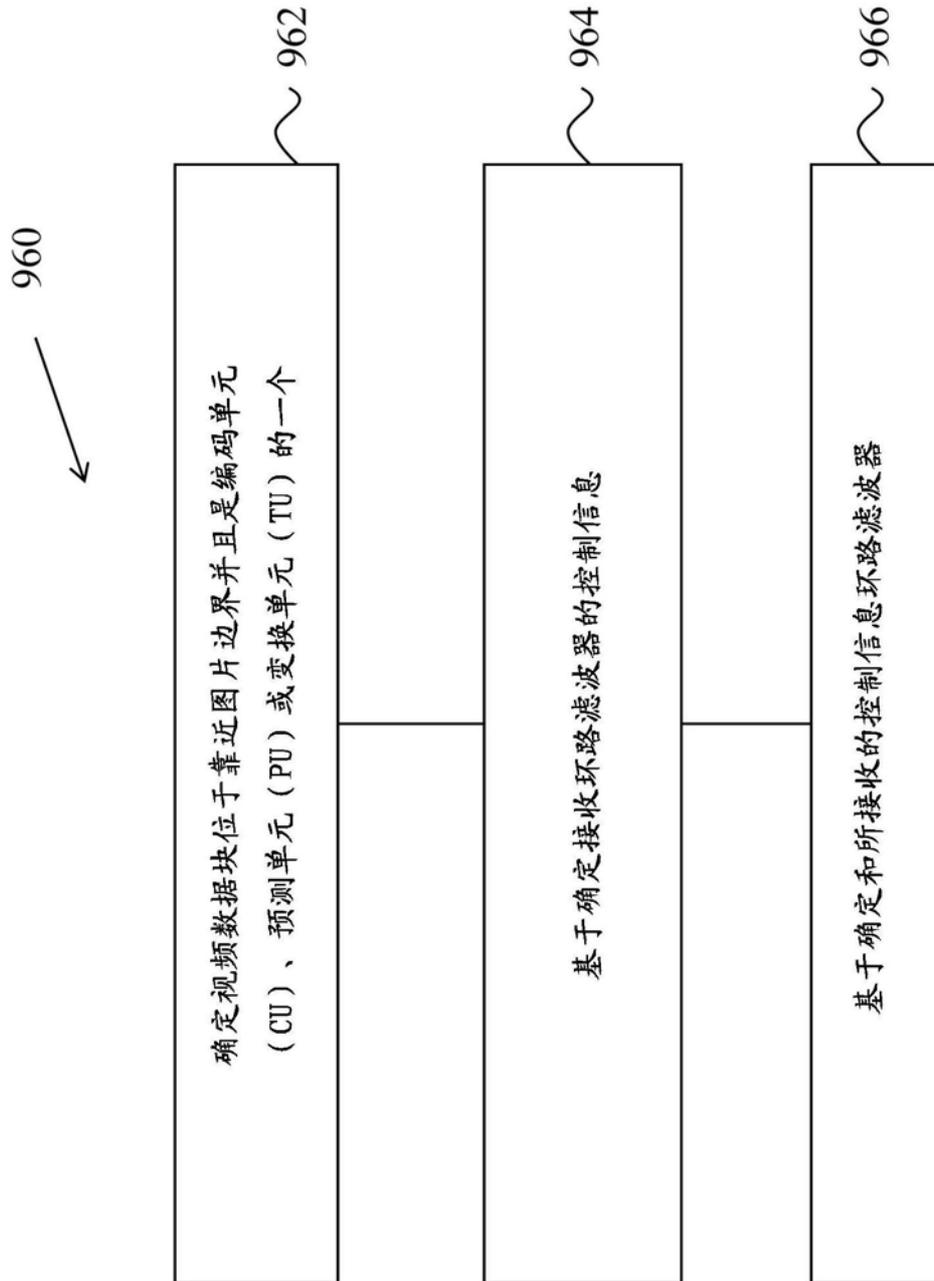


图9C

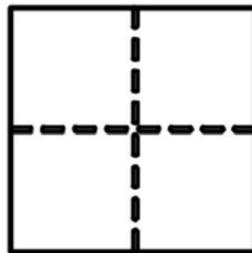


图10A

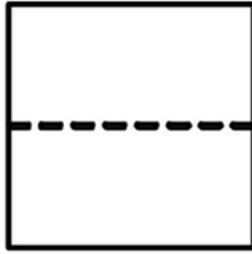


图10B

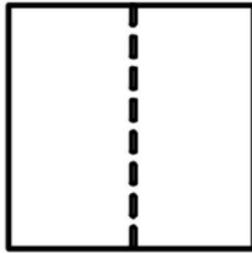


图10C

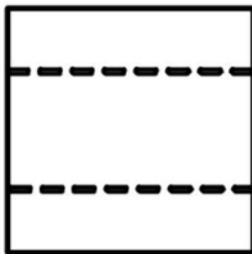


图10D

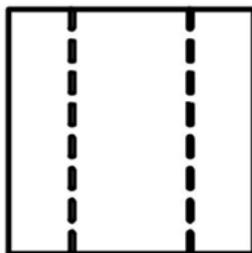


图10E

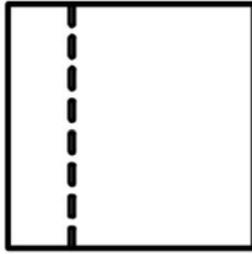


图10F

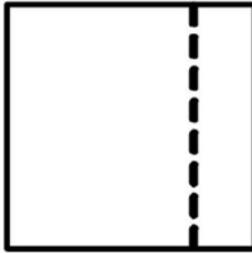


图10G

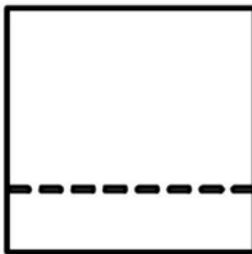


图10H

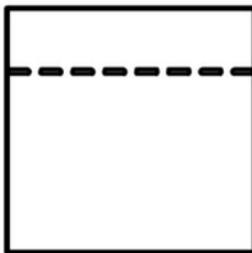


图10I

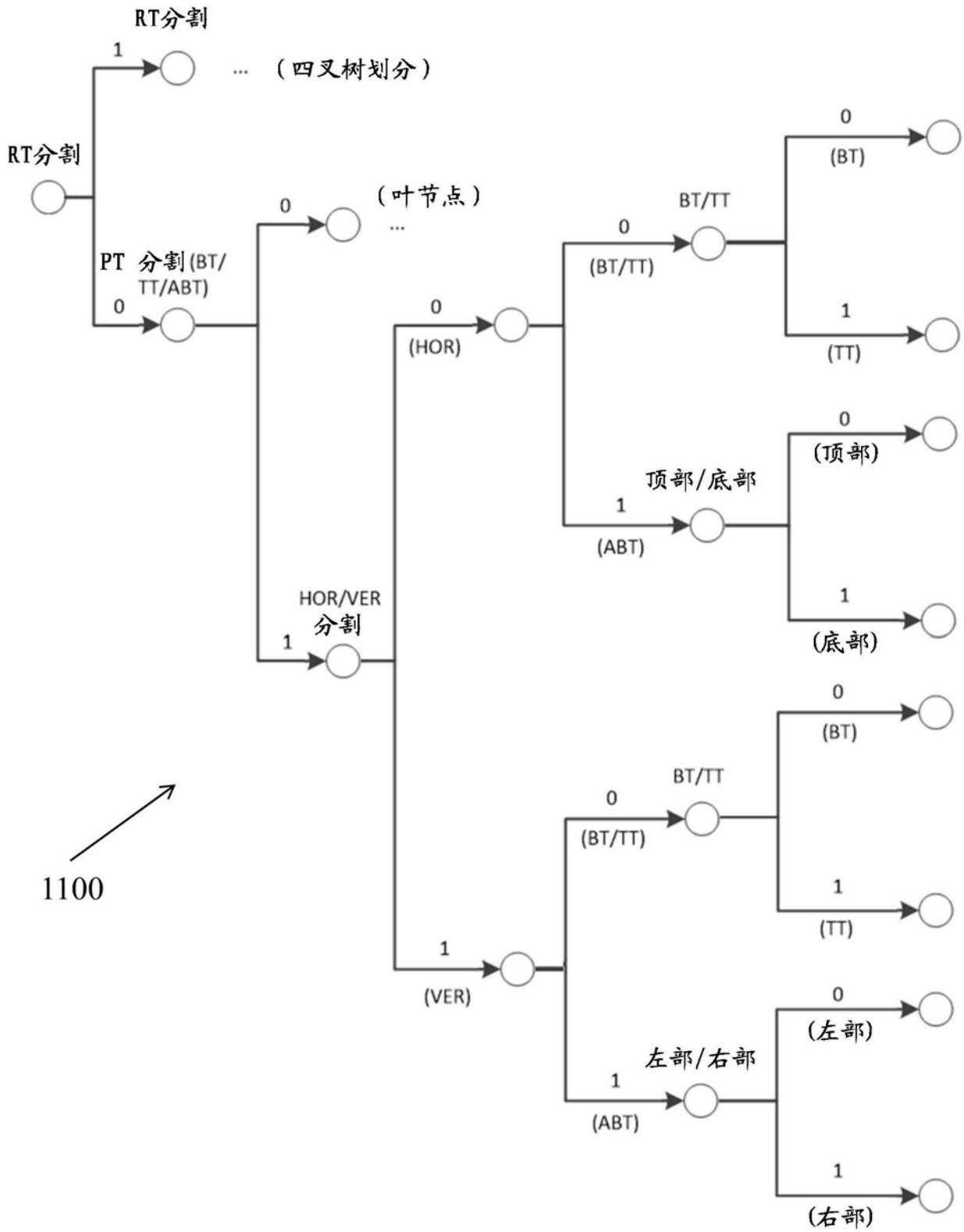


图11

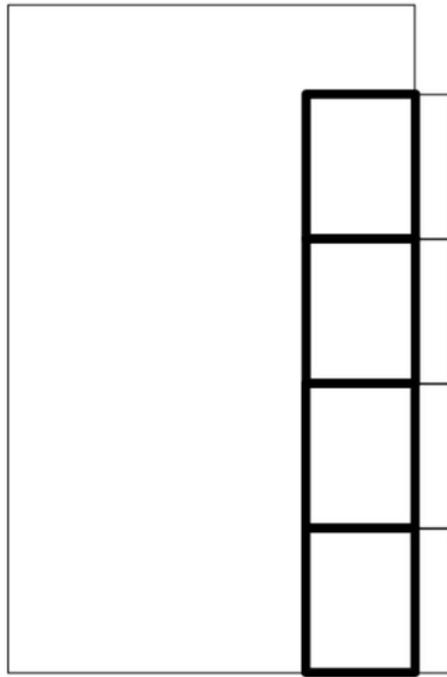


图12A



图12B

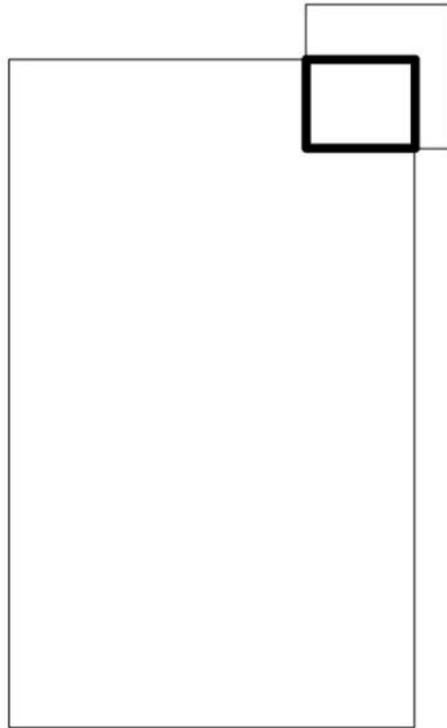


图12C

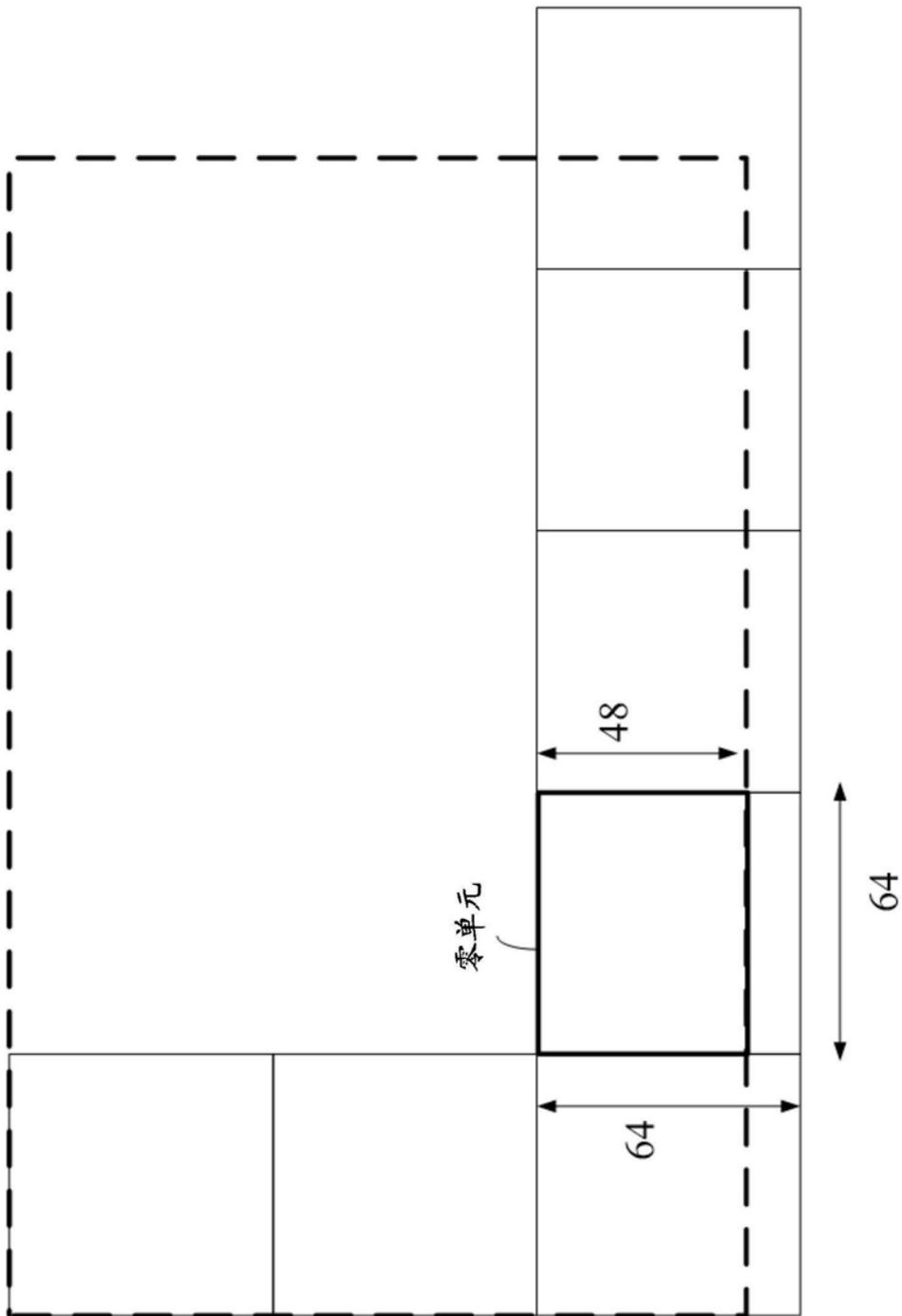


图13

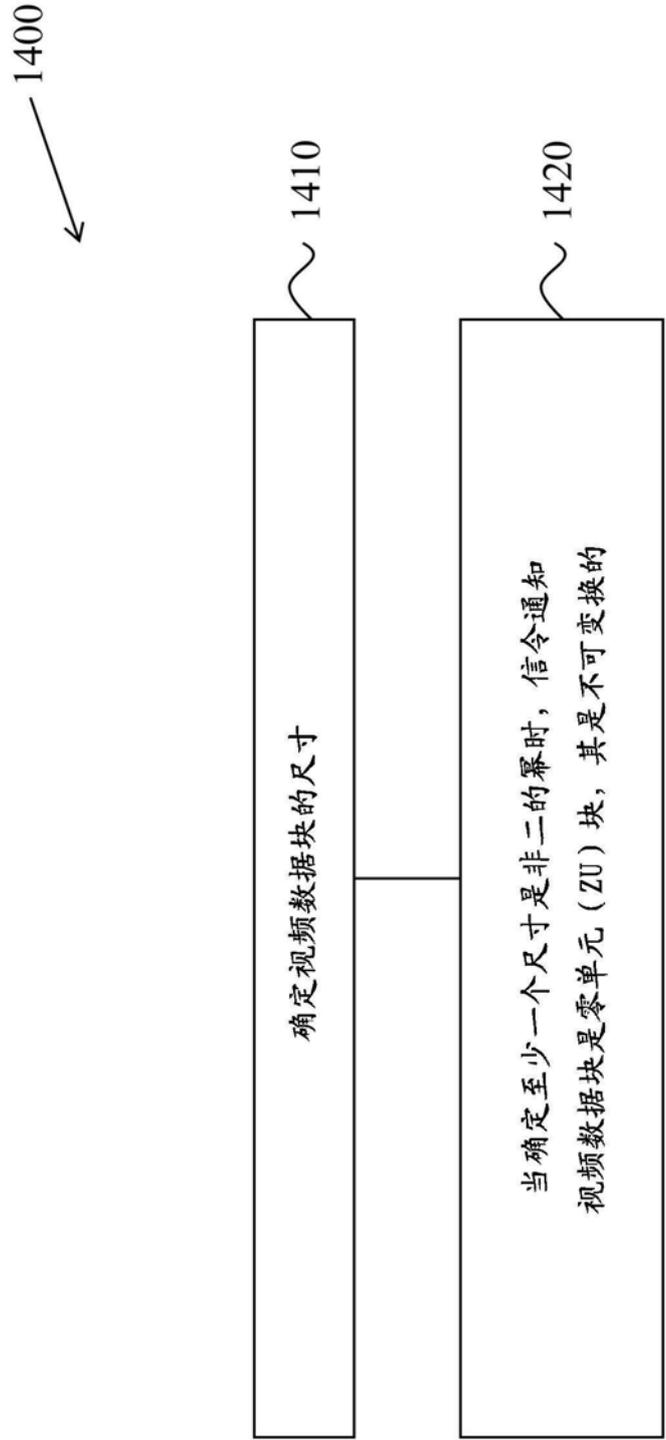


图14

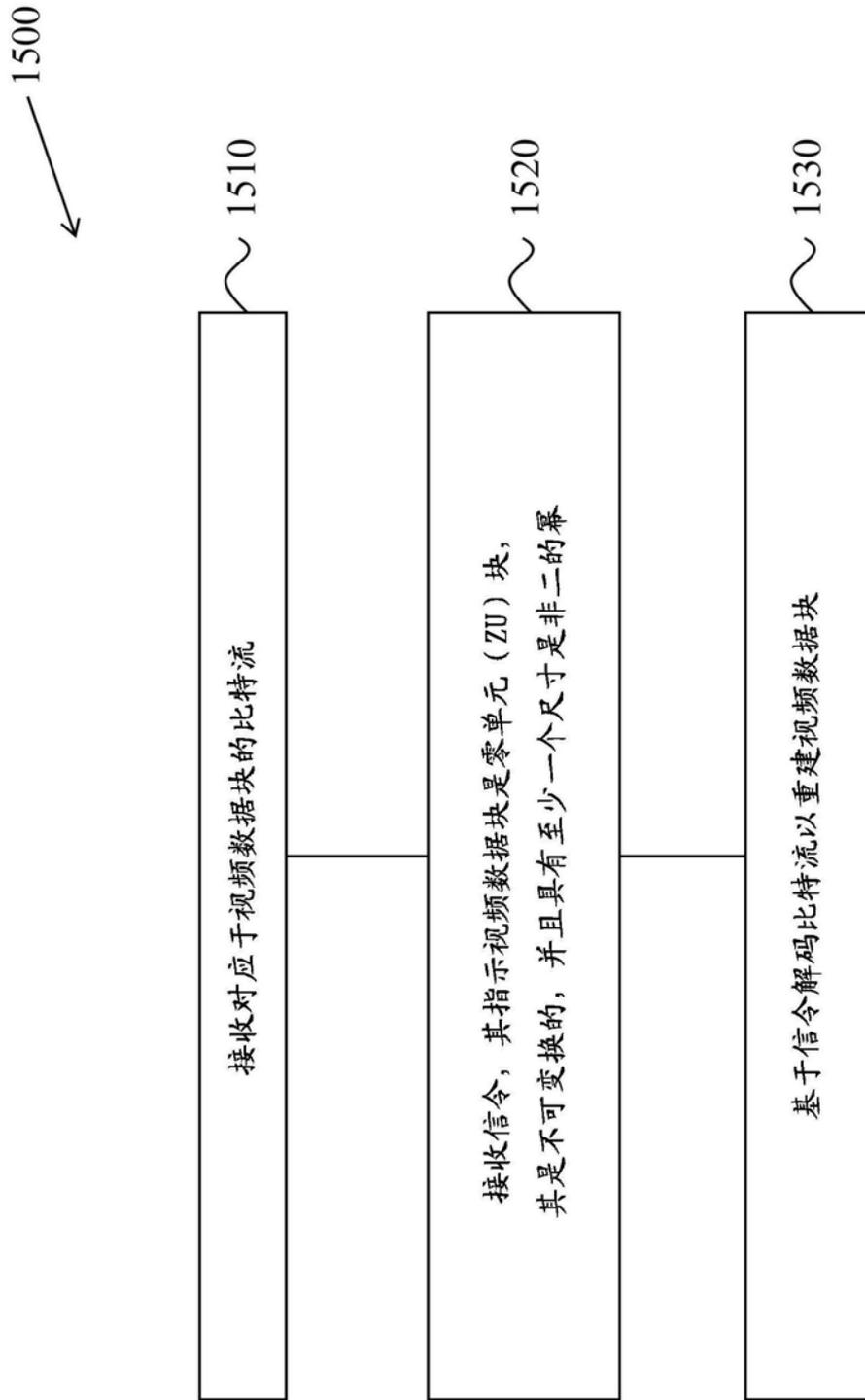


图15

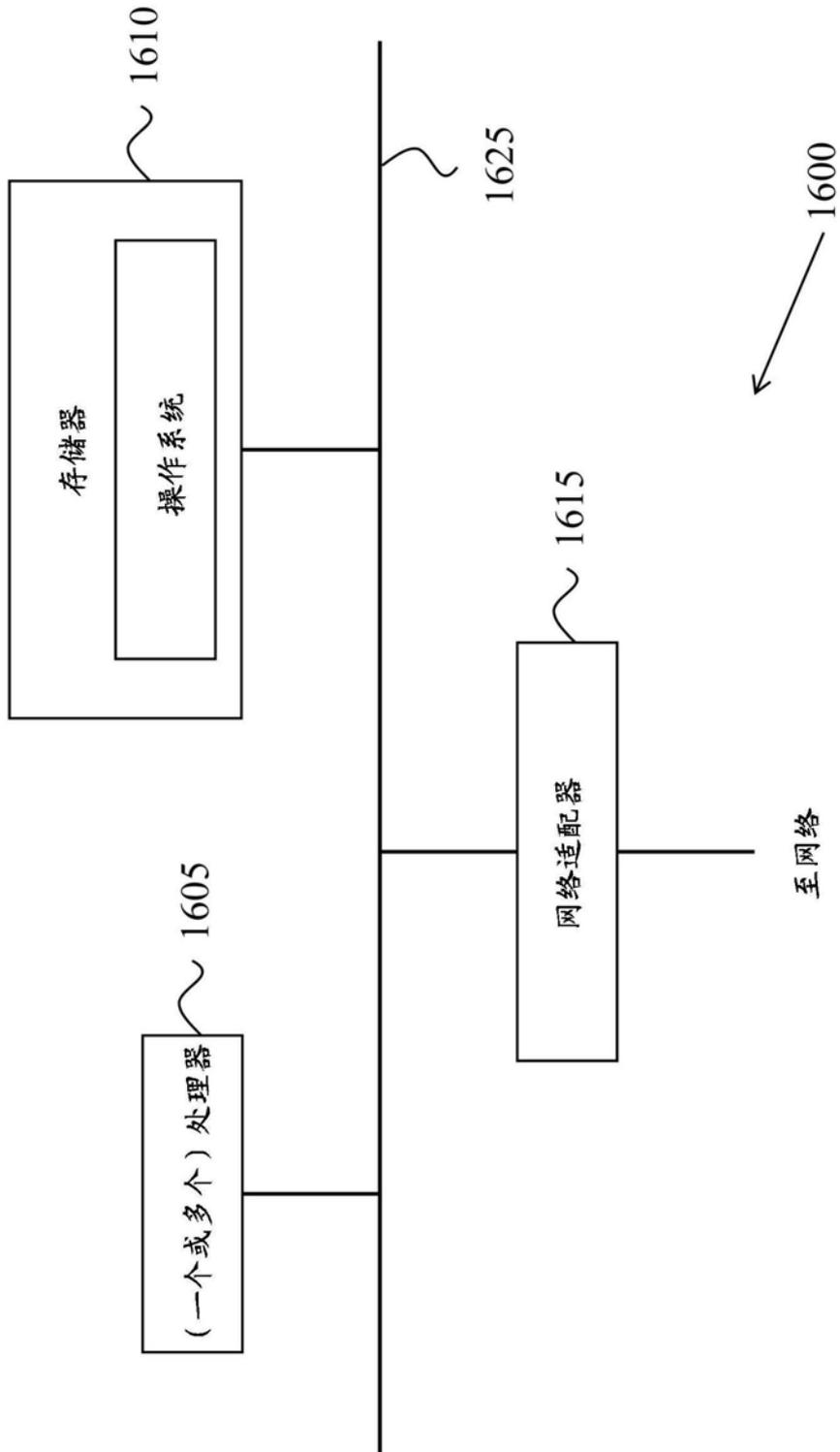


图16

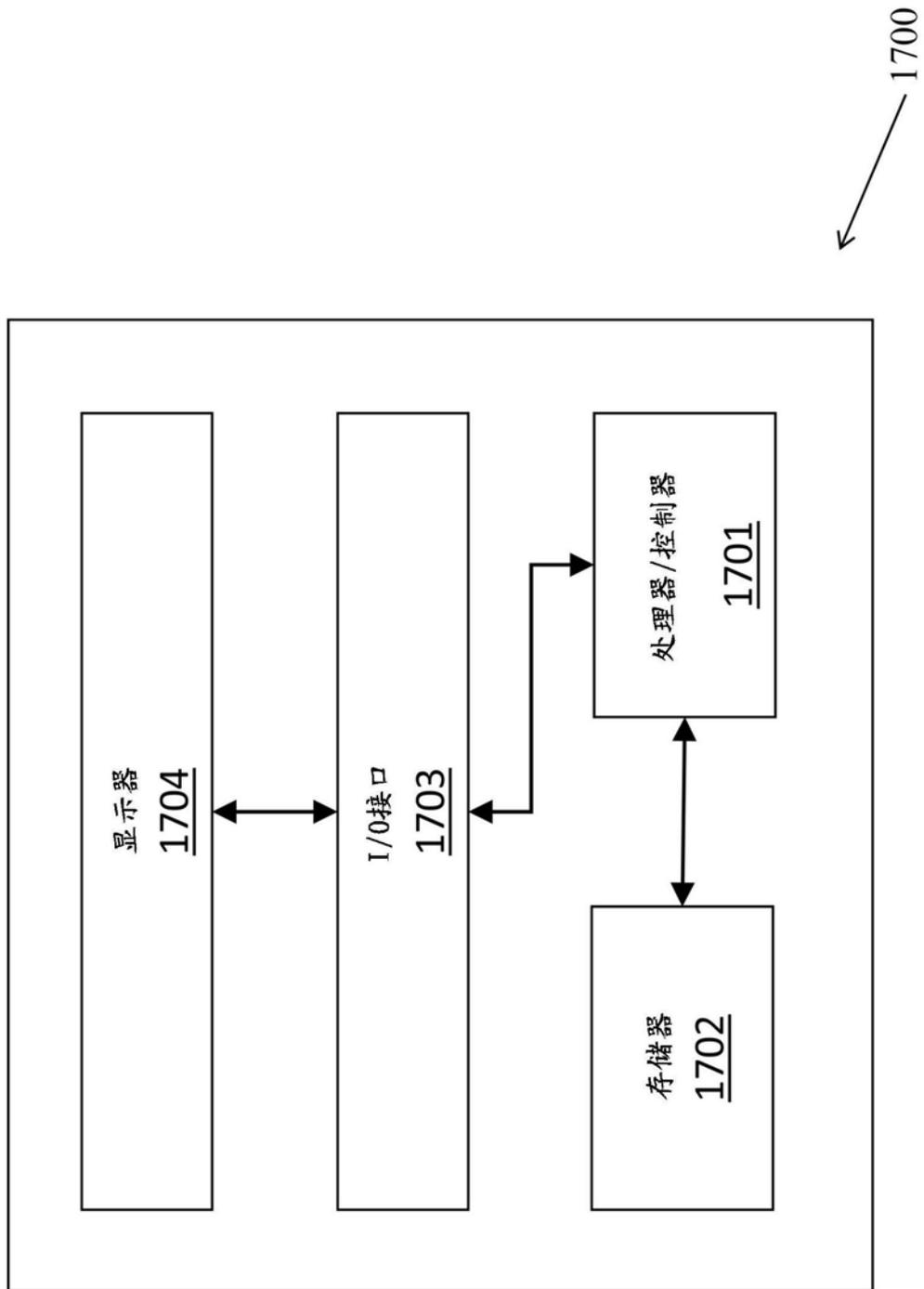


图17

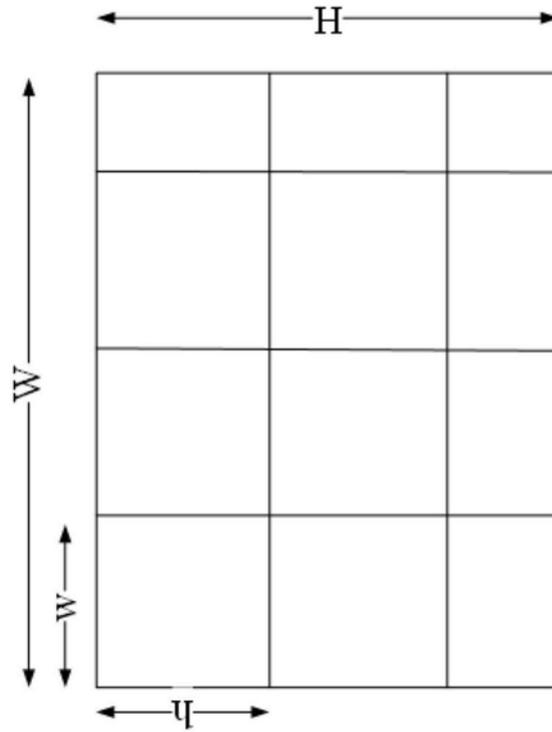


图18A

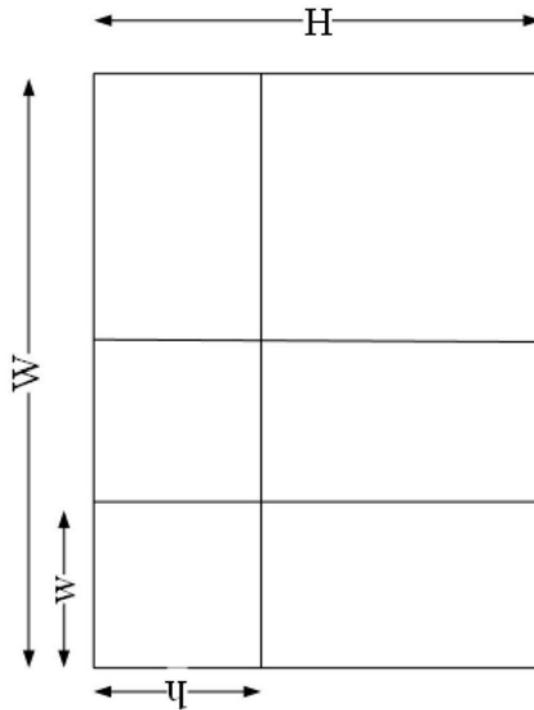


图18B