



(12) 发明专利

(10) 授权公告号 CN 110719466 B

(45) 授权公告日 2022.12.23

(21) 申请号 201910637842.0

H04N 19/117 (2014.01)

(22) 申请日 2019.07.15

H04N 19/139 (2014.01)

(65) 同一申请的已公布的文献号

H04N 19/176 (2014.01)

申请公布号 CN 110719466 A

H04N 19/513 (2014.01)

H04N 19/186 (2014.01)

(43) 申请公布日 2020.01.21

(66) 本国优先权数据

PCT/CN2018/095576 2018.07.13 CN

(73) 专利权人 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

专利权人 字节跳动有限公司

(72) 发明人 刘鸿彬 张莉 张凯 王悦

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 彭久云

(56) 对比文件

US 2012230393 A1, 2012.09.13

CN 1666429 A, 2005.09.07

CN 1666429 A, 2005.09.07

US 2015382005 A1, 2015.12.31

US 6807231 B1, 2004.10.19

WINKEN, Martin等. JVET-J0041-v2: Multi-Hypothesis Inter Prediction.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11》. 2018, 全文.

审查员 贺艳娟

(51) Int. Cl.

H04N 19/103 (2014.01)

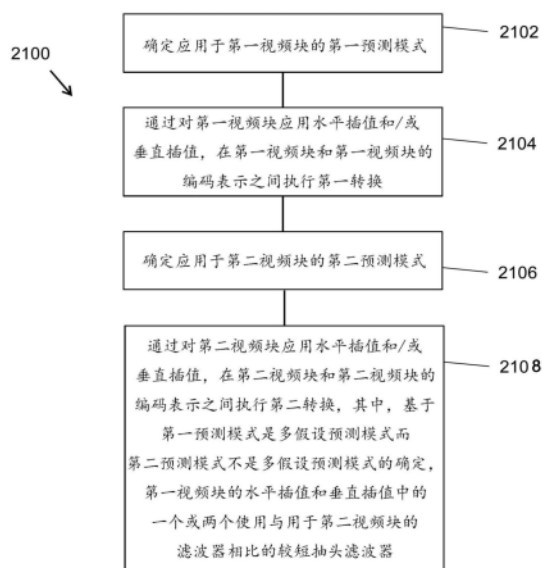
权利要求书1页 说明书17页 附图21页

(54) 发明名称

用于视频处理的方法、装置和存储介质

(57) 摘要

本申请提供了一种视频处理方法,包括:确定应用于第一视频块的第一预测模式;通过对第一视频块应用水平插值和/或垂直插值,在第一视频块和第一视频块的编码表示之间执行第一转换;确定应用于第二视频块的第二预测模式;通过对第二视频块应用水平插值和/或垂直插值,在第二视频块和第二视频块的编码表示之间执行第二转换,其中,基于第一预测模式是多假设预测模式而第二预测模式不是多假设预测模式的确定,第一视频块的水平插值和垂直插值中的一个或两个使用与用于第二视频块的滤波器相比的较短抽头滤波器。



1. 一种视频处理方法,包括:  
确定应用于第一视频块的第一预测模式;  
通过对所述第一视频块应用水平插值和/或垂直插值,在所述第一视频块和所述第一视频块的编解码表示之间执行第一转换;  
确定应用于第二视频块的第二预测模式;  
通过对所述第二视频块应用水平插值和/或垂直插值,在所述第二视频块和所述第二视频块的编解码表示之间执行第二转换,  
其中,基于所述第一预测模式是多假设预测模式而所述第二预测模式是正常预测模式的确定,所述第一视频块的所述水平插值和所述垂直插值中的一个或两个使用与用于所述第二视频块的滤波器相比的较短抽头滤波器,  
其中,在多假设预测模式下,所述水平插值和所述垂直插值中的一个或两个对与多个参考块相关的参考图片列表使用所述较短抽头滤波器。
2. 根据权利要求1所述的方法,其中,所述第一视频块利用多于两个参考块进行转换以用于双向预测,并且至少对于一个参考图片列表,其使用至少两个参考块。
3. 根据权利要求1所述的方法,其中利用多于一个参考块对所述第一视频块进行转换以用于单向预测。
4. 根据权利要求1-3中任一项所述的方法,其中所述较短抽头滤波器是双线性滤波器。
5. 根据权利要求1-4中任一项所述的方法,其中,在多假设预测模式下,当参考图片列表与单个参考块相关时,所述水平插值或所述垂直插值中的一个或两个使用与用于所述正常预测模式相同的滤波器。
6. 根据权利要求1-5中任一项所述的方法,其中,基于以下中的一个或多个的确定来应用所述方法:时域层的使用、包含所述视频块的一个或多个块、片、条带或图片的量化参数在阈值范围内。
7. 根据权利要求6所述的方法,其中在阈值范围内的量化参数包括大于阈值的量化参数。
8. 根据权利要求5所述的方法,其中,所述正常预测模式包括单向预测或双向预测帧间预测模式,所述单向预测使用具有至多一个运动矢量和一个参考索引的帧间预测来预测块中的样本的样本值,所述双向预测帧间预测模式使用具有至多两个运动矢量和参考索引的帧间预测来预测块中样本的样本值。
9. 一种视频解码装置,包括处理器,被配置为实现权利要求1至8中的任一项所述的方法。
10. 一种视频编码装置,包括处理器,被配置为实现权利要求1至8中的任一项所述的方法。
11. 一种其上存储了代码的计算机可读程序介质,所述代码包括指令,当处理器执行所述指令时,使所述处理器实现权利要求1至8中的任一项所述的方法。

## 用于视频处理的方法、装置和存储介质

[0001] 相关申请的交叉引用

[0002] 根据适用的《专利法》和/或《巴黎公约》的规定,本申请及时要求于2018年7月13日提交的国际专利申请号PCT/CN2018/095576的优先权和利益。将国际专利申请号PCT/CN2018/095576的全部公开以引用方式并入本文,作为本申请公开的一部分。

### 技术领域

[0003] 本申请文件涉及视频编码技术、设备和系统。

### 背景技术

[0004] 尽管视频压缩有所进步,数字视频在互联网和其它数字通信网络上使用的带宽仍然最大。随着能够接收和显示视频的连接用户设备数量的增加,预计数字视频使用的带宽需求将继续增长。

### 发明内容

[0005] 所公开的技术可以由视频解码器或编码器实施例使用,其中使用块形插值顺序技术来改进插值。

[0006] 在一个示例方面,公开了一种视频比特流处理方法。该方法包括:确定视频块的形状;基于视频块的形状确定插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以重建视频块的解码表示。

[0007] 在另一个示例方面,视频比特流处理方法包括:确定与视频块相关的运动矢量的特征;基于运动矢量的特征确定插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以重建视频块的解码表示。

[0008] 在另一示例方面,公开了一种视频比特流处理方法。该方法包括:确定视频块的形状;基于视频块的形状确定插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以构造视频块的编码表示。

[0009] 在另一示例方面,公开了一种视频比特流处理方法。该方法包括:确定与视频块相关的运动矢量的特征;基于运动矢量的特征确定插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以构造视频块的编码表示。

[0010] 在一个示例方面,公开了一种视频处理方法。该方法包括:确定应用于第一视频块的第一预测模式;通过对第一视频块应用水平插值和/或垂直插值,在第一视频块和第一视频块的编码表示之间执行第一转换,确定应用于第二视频块的第二预测模式;通过对第二视频块应用水平插值和/或垂直插值,在第二视频块和第二视频块的编码表示之间执行第二转换,其中,基于第一预测模式是多假设预测模式而第二预测模式不是多假设预测模式的确定,第一视频块的水平插值和垂直插值中的一个或两个使用与用于第二视频块的滤波

器相比的较短抽头滤波器。

[0011] 在另一示例方面,公开了一种实现本文所述的视频处理方法的视频解码装置。

[0012] 在又一示例方面,公开了一种实现本文所述的视频处理方法的视频编码装置。

[0013] 在又一典型的方面,本文所述的各种技术可以实施为存储在非暂时性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本文所述方法的程序代码。

[0014] 在又一示例方面,公开了一种视频系统中的装置。该装置包括处理器和其上具有指令的非暂时性存储器,其中由处理器执行的指令使处理器实现上述方法。

[0015] 在附件、附图和下面的描述中阐述了一个或多个实现的细节。其它特征将从说明书和附图以及权利要求书中显而易见。

## 附图说明

[0016] 图1是四叉树二叉树(QTBT)结构的图示。

[0017] 图2示出了Merge候选列表构造的示例推导过程。

[0018] 图3示出了空间Merge候选的示例位置。

[0019] 图4示出了对于空间Merge候选的冗余检查考虑的候选对的示例。

[0020] 图5示出了 $N \times 2N$ 和 $2N \times N$ 分割的第二个预测单元(PU)的位置的示例。

[0021] 图6是时域Merge候选的运动矢量缩放的图示。

[0022] 图7示出了时域Merge候选C0和C1的示例候选位置。

[0023] 图8示出了组合的双向预测Merge候选的示例。

[0024] 图9示出了运动矢量预测候选的推导过程的示例。

[0025] 图10是空间运动矢量候选的运动矢量缩放的图示。

[0026] 图11示出了编码单元(CU)的高级时间运动矢量预测(ATMVP)运动预测的示例。

[0027] 图12示出了具有四个子块(A-D)及其相邻块(a-d)的一个CU的示例。

[0028] 图13示出了J0021中提出的非相邻Merge候选。

[0029] 图14示出了J0058中提出的非相邻Merge候选。

[0030] 图15示出了J0059中提出的非相邻Merge候选。

[0031] 图16示出了用于四分之一样本亮度插值的整数样本和分数样本位置的示例。

[0032] 图17是视频处理装置的示例的框图。

[0033] 图18示出了视频编码器的示例实现的框图。

[0034] 图19是视频比特流处理方法的示例的流程图。

[0035] 图20是视频比特流处理方法的示例的流程图。

[0036] 图21是视频处理方法的示例的流程图。

[0037] 图22是视频比特流处理方法的示例的流程图。

[0038] 图23是视频比特流处理方法的示例的流程图。

## 具体实施方式

[0039] 本文件提供了可由视频比特流的解码器使用以改进解压缩或解码的数字视频的质量的各种技术。此外,视频编码器还可在编码过程期间实现这些技术,以便重建用于进一步编码的所解码帧。

[0040] 为了便于理解,在本文件中使用章节标题,并且不将实施例和技术限制于相应的部分。这样,来自一个章节的实施例可以与来自其他章节的实施例组合。

[0041] 1. 总结

[0042] 本发明涉及视频编码技术。具体地,涉及视频编码中的插值。可应用于现有的视频编码标准,比如HEVC,或待最终确定的标准(多功能视频编码)。也可能适用于未来的视频编码标准或视频编解码器。

[0043] 2. 背景

[0044] 视频编码标准主要是通过开发公知的ITU-T和ISO/IEC标准而发展起来的。ITU-T开发了H.261和H.263,ISO/IEC开发了MPEG-1和MPEG-4视觉,并且两个组织联合开发了H.262/MPEG-2视频、H.264/MPEG-4高级视频编码(AVC)和H.265/HEVC标准。自H.262以来,视频编码标准基于混合视频编码结构,其中采用了时域预测加变换编码。为了探索HEVC之外的未来视频编码技术,联合视频探索团队(JVET)由VCEG和MPEG于2015年联合成立。从那时起,JVET采用了许多新方法并将其引入名为联合探索模型(JEM)的参考软件中。在2018年4月,VCEG(Q6/16)和ISO/IEC JTC1 SC29/WG11(MPEG)之间的联合视频专家组(JVET)被创建用于研究VVC标准,目标是与HEVC相比降低50%的比特率。

[0045] 图18是视频编码器的示例实现的框图。

[0046] 2.1具有较大CTU的四叉树加二叉树(QTBT)块结构

[0047] 在HEVC中,通过使用四叉树结构(表示为编码树)将CTU划分成CU来适应各种局部特性。在CU级别决定是使用帧间(时域)预测还是帧内(空间)预测对图片区域进行编码。根据PU的分割类型,每个CU可以进一步划分成一个、两个或四个PU。在一个PU中,应用相同的预测处理,并且相关信息以PU为基础传输到解码器。在基于PU分割类型通过应用预测处理获得残差块后,可以根据与CU的编码树相似的另一个四叉树结构将CU分割成变换单元(TU)。HEVC结构的一个重要特征是它具有多个分割概念,包括CU、PU以及TU。

[0048] QTBT结构消除了多个分割类型的概念,即QTBT结构消除了CU、PU和TU概念的分离,并支持CU分割形状的更多灵活性。在QTBT块结构中,CU可以是方形或矩形。如图1所示,首先用四叉树结构对编码树单元(CTU)进行分割。四叉树叶节点进一步被二叉树结构分割。在二叉树划分中有两种分割类型:对称的水平划分和对称的垂直划分。二叉树叶节点被称为编码单元(CU),该划分用于预测和转换处理,而无需进一步分割。这意味着在QTBT编码块结构中CU、PU和TU具有相同的块尺寸。在JEM中,CU有时由不同颜色分量的编码块(CB)组成,例如,在4:2:0色度格式的P条带和B条带中,一个CU包含一个亮度CB和两个色度CB,并且CU有时由单个分量的CB组成,例如,在I条带的情况下,一个CU仅包含一个亮度CB或仅包含两个色度CB。

[0049] 为QTBT分割方案定义了以下参数。

[0050] -CTU尺寸:四叉树的根节点尺寸,与HEVC中的概念相同。

[0051] -MinQTSIZE:最小允许的四叉树叶节点尺寸

[0052] -MaxBTSize:最大允许的二叉树根节点尺寸

[0053] -MaxBTDePTh:最大允许的二叉树深度

[0054] -MinBTSize:最小允许的二叉树叶节点尺寸

[0055] 在QTBT分割结构的一个示例中,CTU尺寸被设置为具有两个对应的64×64色度样

点块的 $128 \times 128$ 个亮度样点,MinQTSIZE被设置为 $16 \times 16$ ,MaxBTSIZE被设置为 $64 \times 64$ ,MinBTSIZE(宽度和高度)被设置为 $4 \times 4$ ,MaxBTSIZE被设置为4。四叉树分割首先应用于CTU,以生成四叉树叶节点。四叉树叶节点的尺寸可以具有从 $16 \times 16$ (即,MinQTSIZE)到 $128 \times 128$ (即,CTU尺寸)的尺寸。如果叶四叉树节点是 $128 \times 128$ ,则其不会被二叉树进一步划分,因为其尺寸超过了MaxBTSIZE(例如, $64 \times 64$ )。否则,叶四叉树节点可以被二叉树进一步分割。因此,四叉树叶节点也是二叉树的根节点,并且其二叉树深度为0。当二叉树深度达到MaxBTDePTh(即,4)时,不考虑进一步划分。当二叉树节点的宽度等于MinBTSIZE(即,4)时,不考虑进一步的水平划分。同样,当二叉树节点的高度等于MinBTSIZE时,不考虑进一步的垂直划分。通过预测和变换处理进一步处理二叉树的叶节点,而不需要进一步的分割。在JEM中,最大CTU尺寸为 $256 \times 256$ 个亮度样点。

[0056] 图1(左侧)图示了通过使用QTBT进行块分割的示例,图1(右侧)图示了相应的树表示。实线表示四叉树分割,并且虚线表示二叉树分割。在二叉树的每个划分(即,非叶)节点中,会对一个标志发信令来指示使用哪种分割类型(即,水平或垂直),其中0表示水平划分,1表示垂直划分。对于四叉树分割,不需要指明分割类型,因为四叉树分割总是水平和垂直划分一个块,以生成尺寸相同的4个子块。

[0057] 此外,QTBT方案支持亮度和色度具有单独的QTBT结构的能力。目前,对于P条带和B条带,一个CTU中的亮度和色度CTB共享相同的QTBT结构。然而,对于I条带,用QTBT结构将亮度CTB分割为CU,用另一个QTBT结构将色度CTB分割为色度CU。这意味着I条带中的CU由亮度分量的编码块或两个色度分量的编码块组成,P条带或B条带中的CU由所有三种颜色分量的编码块组成。

[0058] 在HEVC中,为了减少运动补偿的内存访问,限制小块的帧间预测,使得 $4 \times 8$ 和 $8 \times 4$ 块不支持双向预测,并且 $4 \times 4$ 块不支持帧间预测。在JEM的QTBT中,这些限制被移除。

[0059] 2.2HEVC/H.265中的帧间预测

[0060] 每个帧间预测的PU具有一个或两个参考图片列表的运动参数。运动参数包括运动矢量和参考图片索引。对两个参考图片列表中的一个的使用也可以使用inter\_pred\_idc发信令。运动矢量可以相对于预测值显式地编码为增量。

[0061] 当CU采用跳跃模式编码时,一个PU与CU相关联,并且没有显著的残差系数、没有编码的运动矢量增量或参考图片索引。指定了一种Merge模式,通过该模式,可以从相邻的PU(包括空间和时域候选)中获取当前PU的运动参数。Merge模式可以应用于任何帧间预测的PU,而不仅仅是跳跃模式。Merge模式的另一种选择是运动参数的显式传输,其中运动矢量(更准确地说,与运动矢量预测值相比的运动矢量差)、每个参考图片列表对应的参考图片索引和参考图片列表的使用都会在每个PU中显式地发信令。在本文件中,这种模式被称为高级运动矢量预测(AMVP)。

[0062] 当信令指示要使用两个参考图片列表中的一个时,从一个样点块中生成PU。这被称为“单向预测”。单向预测对P条带和B条带都可用。

[0063] 当信令指示要使用两个参考图片列表时,从两个样点块中生成PU。这被称为“双向预测”。双向预测仅对B条带可用。

[0064] 下面文本提供了HEVC中规定的帧间预测模式的细节。描述将从Merge模式开始。

[0065] 2.2.1Merge模式

[0066] 2.2.1.1 Merge模式的候选的推导

[0067] 当使用Merge模式预测PU时,从比特流分析指向Merge候选列表中条目的索引,并用于检索运动信息。该列表的结构在HEVC标准中有规定,并且可以按照以下步骤顺序进行概括:

[0068] 步骤1:初始候选推导

[0069] 步骤1.1:空域候选推导

[0070] 步骤1.2:空域候选的冗余检查

[0071] 步骤1.3:时域候选推导

[0072] 步骤2:附加候选插入

[0073] 步骤2.1:双向预测候选的创建

[0074] 步骤2.2:零运动候选的插入

[0075] 在图2中也示意性描述了这些步骤。对于空间Merge候选推导,在位于五个不同位置的候选中最多选择四个Merge候选。对于时域Merge候选推导,在两个候选中最多选择一个Merge候选。由于在解码器处假定每个PU的候选数为常量,因此当从步骤1获得的候选数未达到条带报头中发信令的最大Merge候选数(MaxNumMergeCand)时,生成附加的候选。由于候选数是恒定的,所以最佳Merge候选的索引使用截断的一元二值化(TU)进行编码。如果CU的大小等于8,则当前CU的所有PU都共享一个Merge候选列表,这与 $2N \times 2N$ 预测单元的Merge候选列表相同。

[0076] 下面详细介绍与上述步骤相关的操作。

[0077] 2.2.1.2空域候选推导

[0078] 在空间Merge候选的推导中,在位于图3所示位置的候选中最多选择四个Merge候选。推导顺序为A1、B1、B0、A0和B2。只有当位置A1、B1、B0、A0的任何PU不可用(例如,因为它属于另一个条带或片)或是内部编码时,才考虑位置B2。在增加A1位置的候选后,对其余候选的增加进行冗余检查,其确保具有相同运动信息的候选被排除在列表之外,从而提高编码效率。为了降低计算的复杂度,在所提到的冗余检查中并不考虑所有可能的候选对。相反,只有与图4中的箭头链接的对才会被考虑,并且只有当用于冗余检查的对应候选没有相同的运动信息时,才将候选添加到列表中。复制运动信息的另一个来源是与 $2N \times 2N$ 不同的分区相关的“第二PU”。例如,图5分别描述了 $N \times 2N$ 和 $2N \times N$ 情况下的第二PU。当当前的PU被划分为 $N \times 2N$ 时,对于列表构建不考虑A1位置的候选。在一些实施例中,添加此候选可能导致两个具有相同运动信息的预测单元,这对于在编码单元中仅具有一个PU是冗余的。同样地,当当前PU被划分为 $2N \times N$ 时,不考虑位置B1。

[0079] 2.2.1.3时域候选推导

[0080] 在此步骤中,只有一个候选添加到列表中。特别地,在这个时域Merge候选的推导中,基于与给定参考图片列表中当前图片具有最小图片顺序计数POC差异的并置PU推导了缩放运动矢量。用于推导并置PU的参考图片列表在条带报头中显式地发信令。图6中的虚线示出了时域Merge候选的缩放运动矢量的获得,其使用POC距离 $t_b$ 和 $t_d$ 从并置PU的运动矢量进行缩放,其中 $t_b$ 定义为当前图片的参考图片和当前图片之间的POC差异,并且 $t_d$ 定义为并置图片的参考图片与并置图片之间的POC差异。时域Merge候选的参考图片索引设置为零。HEVC规范中描述了缩放处理的实际实现。对于B条带,得到两个运动矢量(一个是对于参考

图片列表0,另一个是对于参考图片列表1)并将其组合使其成为双向预测Merge候选。

[0081] 图6是用于时域Merge候选的运动矢量缩放的说明。

[0082] 在属于参考帧的并置PU(Y)中,在候选 $C_0$ 和 $C_1$ 之间选择时域候选的位置,如图7所示。如果位置 $C_0$ 处的PU不可用、内部编码或在当前CTU行之外,则使用位置 $C_1$ 。否则,位置 $C_0$ 被用于时域Merge候选的推导。

[0083] 2.2.1.4附加候选插入

[0084] 除了空间和时域Merge候选,还有两种附加类型的Merge候选:组合双向预测Merge候选和零Merge候选。组合双向预测Merge候选是利用空间和时域Merge候选生成的。组合双向预测Merge候选仅用于B条带。通过将初始候选的第一参考图片列表运动参数与另一候选的第二参考图片列表运动参数相结合,生成组合双向预测候选。如果这两个元组提供不同的运动假设,则它们将形成新的双向预测候选。作为示例,图8示出了原始列表中(在左侧)的两个候选被用于创建添加到最终列表(在右侧)中的组合双向预测Merge候选的情况,其具有MvL0和refIdxL0或MvL1和refIdxL1的两个候选。现有技术中定义了许多关于组合的规则需要考虑以生成这些附加Merge候选。

[0085] 插入零运动候选以填充Merge候选列表中的其余条目,从而达到MaxNumMergeCand的容量。这些候选具有零空间位移和从零开始并且每次将新的零运动候选添加到列表中时都会增加的参考图片索引。这些候选使用的参考帧的数目对于单向预测和双向预测分别是1帧和2帧。最后,对这些候选不执行冗余检查。

[0086] 2.2.1.5并行处理的运动估计区域

[0087] 为了加快编码处理,可以并行执行运动估计,从而同时推导给定区域内所有预测单元的运动矢量。从空间邻域推导Merge候选可能会干扰并行处理,因为一个预测单元在完成相关运动估计之前无法从相邻的PU推导运动参数。为了缓和编码效率和处理延迟之间的平衡,HEVC定义了运动估计区域(MER),可使用语法元素“log2\_parallel\_merge\_level\_minus2”在图片参数集中对MER的尺寸中发信令。当定义MER时,落入同一区域的Merge候选标记为不可用,并且因此在列表构建中不考虑。

[0088] 2.2.2AMVP

[0089] AMVP利用运动矢量与相邻的PU的空时相关性,其用于运动参数的显式传输。对于每个参考图片列表,首先通过检查左上方的时域相邻的PU位置的可用性、去掉多余的候选位置并且加上零矢量以使候选列表长度恒定来构建运动矢量候选列表。然后,编码器可以从候选列表中选择最佳的预测值,并发送指示所选候选的对应索引。与Merge索引信令类似,最佳运动矢量候选的索引使用截断的一元进行编码。在这种情况下要编码的最大值是2(参照图9)。在下面的章节中,将详细介绍运动矢量预测候选的推导过程。

[0090] 2.2.2.1AMVP候选的推导

[0091] 图9概括了运动矢量预测候选的推导过程。

[0092] 在运动矢量预测中,考虑了两种类型的运动矢量候选:空间运动矢量候选和时域运动矢量候选。对于空间运动矢量候选的推导,基于位于图3所示的五个不同位置的每个PU的运动矢量最终推导出两个运动矢量候选。

[0093] 对于时域运动矢量候选的推导,从两个候选中选择一个运动矢量候选,这两个候选是基于两个不同的并置位置推导的。在作出第一个空时候选列表后,移除列表中重复的



运动矢量候选。如果潜在候选的数量大于二,则从列表中移除相关联的参考图片列表中参考图片索引大于1的运动矢量候选。如果空时运动矢量候选数小于二,则会在列表中添加附加的零运动矢量候选。

#### [0094] 2.2.2.2空间运动矢量候选

[0095] 在推导空间运动矢量候选时,在五个潜在候选中最多考虑两个候选,这五个候选来自图3所描绘位置上的PU,这些位置与运动Merge的位置相同。当前PU左侧的推导顺序定义为 $A_0$ 、 $A_1$ 、以及缩放的 $A_0$ 、缩放的 $A_1$ 。当前PU上面的推导顺序定义为 $B_0$ 、 $B_1$ 、 $B_2$ 、缩放的 $B_0$ 、缩放的 $B_1$ 、缩放的 $B_2$ 。因此,每侧有四种情况可以用作运动矢量候选,其中两种情况不需要使用空间缩放,并且两种情况使用空间缩放。四种不同的情况概括如下:

#### [0096] --无空间缩放

[0097] (1) 相同的参考图片列表,并且相同的参考图片索引(相同的POC)

[0098] (2) 不同的参考图片列表,但是相同的参考图片索引(相同的POC)

#### [0099] --空间缩放

[0100] (3) 相同的参考图片列表,但是不同的参考图片索引(不同的POC)

[0101] (4) 不同的参考图片列表,并且不同的参考图片索引(不同的POC)

[0102] 首先检查无空间缩放的情况,然后检查空间缩放。当POC在相邻PU的参考图片与当前PU的参考图片之间不同时,都会考虑空间缩放,而不考虑参考图片列表。如果左侧候选的所有PU都不可用或是内部编码,则允许对上述运动矢量进行缩放,以帮助左侧和上方MV候选的平行推导。否则,不允许对上述运动矢量进行空间缩放。

[0103] 图10是空间运动矢量候选的运动矢量缩放的说明。

[0104] 在空间缩放处理中,相邻PU的运动矢量以与时域缩放相似的方式缩放,如图10所示。主要区别在于,给出了当前PU的参考图片列表和索引作为输入,实际缩放处理与时域缩放处理相同。

#### [0105] 2.2.2.3时域运动矢量候选

[0106] 除了参考图片索引的推导外,时域Merge候选的所有推导过程与空间运动矢量候选的推导过程相同(参见图7)。向解码器发参考图片索引的信令。

#### [0107] 2.3JEM中新的帧间Merge候选

##### [0108] 2.3.1基于子CU的运动矢量预测

[0109] 在具有QTBT的JEM中,每个CU对于每个预测方向最多可以具有一组运动参数。通过将大的CU分割成子CU并推导该大CU的所有子CU的运动信息,编码器中考虑了两种子CU级的运动矢量预测方法。可选时域运动矢量预测(ATMVP)方法允许每个CU从多个小于并置参考图片中当前CU的块中获取多组运动信息。在空时运动矢量预测(STMVP)方法中,通过利用时域运动矢量预测值和空间邻接运动矢量递归地推导子CU的运动矢量。

[0110] 为了为子CU运动预测的保持更精确的运动场,当前禁用参考帧的运动压缩。

##### [0111] 2.3.1.1可选时域运动矢量预测

[0112] 在可选时域运动矢量预测(ATMVP)方法中,运动矢量时域运动矢量预测(TMVP)是通过从小于当前CU的块中提取多组运动信息(包括运动矢量和参考索引)来修改的。如图11所示,子CU为方形 $N \times N$ 块(默认N设置为4)。

[0113] ATMVP分两步预测CU内的子CU的运动矢量。第一步是用所谓的时域矢量识别参考

图片中的对应块。参考图片称为运动源图片。第二步是将当前CU划分成子CU,并从每个子CU对应的块中获取运动矢量以及每个子CU的参考索引,如图11所示。

[0114] 在第一步中,参考图片和对应的块由当前CU的空间相邻块的运动信息确定。为了避免相邻块的重复扫描处理,使用当前CU的Merge候选列表中的第一个Merge候选。第一个可用的运动矢量及其相关联的参考索引被设置为时域矢量和运动源图片的索引。这样,在ATMVP中,与TMVP相比,可以更准确地识别对应的块,其中对应的块(有时称为并置块)始终位于相对于当前CU的右下角或中心位置。

[0115] 在第二步中,通过将时域矢量添加到当前CU的坐标中,通过运动源图片中的时域矢量识别子CU的对应块。对于每个子CU,使用其对应块的运动信息(覆盖中心样点的最小运动网格)来推导子CU的运动信息。在识别出对应 $N \times N$ 块的运动信息后,将其转换为当前子CU的运动矢量和参考索引,与HEVC的TMVP方法相同,其中应用运动缩放和其它处理。例如,解码器检查是否满足低延迟条件(例如,当前图片的所有参考图片的POC都小于当前图片的POC),并可能使用运动矢量 $MV_x$ (与参考图片列表X对应的运动矢量)来为每个子CU预测运动矢量 $MV_y$ ( $X$ 等于0或1且 $Y$ 等于 $1-X$ )。

[0116] 2.3.1.2空时运动矢量预测

[0117] 在这种方法中,子CU的运动矢量是按照光栅扫描顺序递归推导的。图12说明了该概念。我们来考虑一个 $8 \times 8$ 的CU,它包含四个 $4 \times 4$ 的子CU A、B、C和D。当前帧中相邻的 $4 \times 4$ 的块标记为a、b、c和d。

[0118] 子CU A的运动推导由识别其两个空间邻居开始。第一个邻居是子CU A上方的 $N \times N$ 块(块c)。如果该块c不可用或内部编码,则检查子CU A上方的其它 $N \times N$ 块(从左到右,从块c处开始)。第二个邻居是子CU A左侧的一个块(块b)。如果块b不可用或是内部编码,则检查子CU A左侧的其它块(从上到下,从块b处开始)。每个列表从相邻块获得的运动信息被缩放到给定列表的第一个参考帧。接下来,按照HEVC中规定的与TMVP相同的程序,推导出子块A的时域运动矢量预测(TMVP)。提取位置D处的并置块的运动信息并进行相应的缩放。最后,在检索和缩放运动信息后,对每个参考列表分别平均所有可用的运动矢量(最多3个)。将平均运动矢量指定为当前子CU的运动矢量。

[0119] 2.3.1.3子CU运动预测模式信令

[0120] 作为附加merge候选而启用子CU模式,并且不需要附加的语法元素来发信令通知该模式。对每个CU的merge候选列表添加两个附加的merge候选,以表示ATMVP模式和STMVP模式。如果序列参数集合指示启用ATMVP和STMVP,则使用多达七个merge候选。额外merge候选的编码逻辑与HM中的merge候选相同,这意味着,对于P或B条带中的每个CU,对于两个附加merge候选需要多两个RD检查。

[0121] 在JEM中,CABAC对merge索引的所有二元位进行上下文编码。而在HEVC中,仅对第一个二元位进行上下文编码,而对剩余二元位上下文旁路编码。

[0122] 2.3.2非相邻Merge候选

[0123] 在J0021中,高通提出从如在图13中标记为6到49的非相邻的相邻位置推导附加空间Merge候选。将所推导的候选添加在Merge候选列表中的TMVP候选之后。

[0124] 在J0058中,腾讯提出从相对于当前块具有偏移(-96, -96)的外部参考区域中的位置推导附加空间Merge候选。

[0125] 如图14所示,位置标记为A(i, j)、B(i, j)、C(i, j)、D(i, j)和E(i, j)。与其先前的B或C候选相比,每个候选B(i, j)或C(i, j)在垂直方向上具有16的偏移。与其先前的A或D候选相比,每个候选A(i, j)或D(i, j)在水平方向上具有16的偏移。与其先前的E候选相比,每个E(i, j)在水平方向和垂直方向上具有16的偏移。对候选从内到外进行检查。并且候选的顺序是A(i, j)、B(i, j)、C(i, j)、D(i, j)和E(i, j)。进一步研究merge候选的数量是否可以进一步减少。将候选添加在merge候选列表中的TMVP候选之后。

[0126] 在J0059中,根据它们在时域候选之后的数字顺序,对图15中从6到27的扩展空间位置进行检查。为了节省MV行缓冲,所有空间候选都被限制在两个CTU行内。

[0127] 2.4JEM中的帧内预测

[0128] 2.4.1具有67个帧内预测模式的帧内模式编码

[0129] 对于亮度插值滤波,将8抽头可分离的基于DCT的插值滤波器用于2/4精度样本,并且将7抽头可分离的基于DCT的插值滤波器用于1/4精度样本,如表1所示。

[0130] 表1:用于1/4亮度插值的8抽头DCT-IF系数。

置	滤波器系数
[0131] /4	1 { -1, 4, -10, 58, 17, -5, 1 }
/4	2 { -1, 4, -11, 40, 40, -11, 4, -1 }
/4	3 { 1, -5, 17, 58, -10, 4, -1 }

[0132] 类似地,将4抽头可分离的基于DCT的插值滤波器用于色度插值滤波器,如表2所示。

[0133] 表2:用于1/8色度插值的4抽头DCT-IF系数。

置	滤波器系数
[0134]	1 { -2, 58,

	/8	10, -2 }
	2	{ -4, 54,
	/8	16, -2 }
	3	{ -6, 46,
	/8	28, -4 }
[0135]	4	{ -4, 36,
	/8	36, -4 }
	5	{ -4, 28,
	/8	46, -6 }
	6	{ -2, 16,
	/8	54, -4 }
	7	{ -2, 10,
	/8	58, -2 }

[0136] 对于4:2:2的垂直插值以及4:4:4色度通道的水平和垂直插值,不使用表2中的奇数位置,导致1/4色度插值。

[0137] 对于双向预测,在对两个预测信号求平均之前,无论源比特深度如何,插值滤波器的输出的比特深度都保持为14比特精度。实际平均过程是隐式地通过比特深度降低过程完成的:

[0138]  $\text{predSamples}[x,y] = \text{predSamplesL0}[x,y] + \text{predSamplesL1}[x,y] + \text{offset} \gg \text{shift}$

[0139] 其中 $\text{shift} = (15 - \text{BitDepth})$ 且 $\text{offset} = 1 \ll (\text{shift} - 1)$

[0140] 如果运动矢量的水平分量和垂直分量都指向子像素位置,则总是先执行水平插值,然后执行垂直插值。例如为了对图16中所示的子像素 $j_0, 0$ 进行插值,首先,根据等式2-1对 $b_0, k$  ( $k = -3, -2, \dots, 3$ )进行插值,然后根据等式2-2对 $j_0, 0$ 进行插值。这里, $\text{shift1} = \text{Min}(4, \text{BitDepthY} - 8)$ ,且 $\text{shift2} = 6$ ,其中 $\text{BitDepthY}$ 是视频块的比特深度,更具体地说,是视频块的亮度分量的比特深度。

[0141]  $b_0, k = (-A - 3, k + 4 * A - 2, k - 11 * A - 1, k + 40 * A_0, k + 40 * A_1, k - 11 * A_2, k + 4 * A_3, k - A_4, k) \gg \text{shift1}$  (2-1)

[0142]  $j_0, 0 = (-b_0, -3 + 4 * b_0, -2 - 11 * b_0, -1 + 40 * b_0, 0 + 40 * b_0, 1 - 11 * b_0, 2 + 4 * b_0, 3 - b_0, 4) \gg \text{shift2}$  (2-2)

[0143] 可替代地,我们可以先执行垂直插值,然后执行水平插值。在这种情况下,为了对 $j_0, 0$ 进行插值,首先,根据等式2-3对 $h_k, 0$  ( $k = -3, -2, \dots, 3$ )进行插值,然后根据等式2-4对 $j_0, 0$ 进行插值。当 $\text{BitDepthY}$ 小于或等于8时, $\text{shift1}$ 为0,在第一个插值阶段没有任何损失,因此,最终插值结果不会被插值顺序改变。然而,当 $\text{BitDepthY}$ 大于8时, $\text{shift1}$ 大于0。在这

种情况下,当应用不同的插值顺序时,最终的插值结果可能不同。

[0144]  $hk,0 = (-Ak, -3+4*Ak, -2-11*Ak, -1+40*Ak, 0+40*Ak, 1-11*Ak, 2+4*Ak, 3-Ak, 4) \gg \text{shift1} \quad (2-3)$

[0145]  $j0,0 = (-h-3, 0+4*h-2, 0-11*h-1, 0+40*h0, 0+40*h1, 0-11*h2, 0+4*h3, 0-h4, 0) \gg \text{shift2} \quad (2-4)$

[0146] 3. 实施例解决的问题的示例

[0147] 对于亮度块尺寸WxH,如果我们总是先执行水平插值,则所需的插值(每个像素)在表3中示出。

[0148] 表3: HEVC/JEM对WxH亮度分量所需的插值

MV 的水平分量	MV 的垂直分量	插值
1/4 像素	1/4 像素	$((H + 6) \times W + W \times H) / (W \times H) = 2 + 6 / H$
1/4 像素	1/2 像素	$((H + 7) \times W + W \times H) / (W \times H) = 2 + 7 / H$
1/2 像素	1/4 像素	$((H + 6) \times W + W \times H) / (W \times H) = 2 + 6 / H$
1/2 像素	1/2 像素	$((H + 7) \times W + W \times H) / (W \times H) = 2 + 7 / H$

[0150] 另一方面,如果我们先执行垂直插值,则表4中示出了所需的插值。显然,最佳插值顺序是在表3和表4之间需要较小插值次数的插值顺序。

[0151] 表4: 当插值顺序颠倒时,WxH亮度分量所需的插值

MV 的水平分量	MV 的垂直分量	插值
1/4 像素	1/4 像素	$((W + 6) \times H + W \times H) / (W \times H) = 2 + 6 / W$
1/4 像素	1/2 像素	$((W + 6) \times H + W \times H) / (W \times H) = 2 + 6 / W$
1/2 像素	1/4 像素	$((W + 7) \times H + W \times H) / (W \times H) = 2 + 7 / W$
1/2 像素	1/2 像素	$((W + 7) \times H + W \times H) / (W \times H) = 2 + 7 / W$

[0153] 对于色度分量,如果我们总是先执行水平插值,则所需插值为 $((H+3) \times W+W \times H) / (W \times H) = 2+3/H$ 。如果我们总是先执行垂直插值,所需插值为 $((W+3) \times H+W \times H) / (W \times H) = 2+3/W$ 。

[0154] 如上所述,当输入视频的比特深度大于8时,不同的插值顺序可导致不同的插值结果。因此,插值顺序应在编码器和解码器中隐式地定义。

[0155] 4. 实施例的示例

[0156] 为了解决这些问题并提供其他益处,我们提出了形状相关的插值顺序。

[0157] 以下详细示例应被视为解释一般概念的示例。不应以狭隘的方式解释这些发明。此外,这些发明可以以任何方式组合。

[0158] 1. 提出插值顺序取决于当前编码块形状(例如,编码块是CU)。

[0159] a. 在一个示例中,对于宽度>高度的块(诸如在基于子块的预测(如仿射、ATMVP或BIO)中使用的CU、PU或子块),首先执行垂直插值,然后执行水平插值,例如,首先对像素 $d_{k,0}, h_{k,0}$ 和 $n_{k,0}$ 进行插值,然后对 $e_{0,0}$ 至 $r_{0,0}$ 进行插值。等式2-3和2-4中示出了 $j_{0,0}$ 的示例。

[0160] i. 可替代地,对于宽度 $\geq$ 高度的块(诸如在基于子块的预测(如仿射、ATMVP或BIO)中使用的CU、PU或子块),先执行垂直插值,然后执行水平插值。

[0161] b. 在一个示例中,对于宽度 $\leq$ 高度的块(诸如在基于子块的预测(如仿射、ATMVP或BIO)中使用的CU、PU或子块),先执行水平插值,然后执行垂直插值。

[0162] i. 可替代地,对于宽度 $<$ 高度的块(诸如在基于子块的预测(如仿射、ATMVP或BIO)中使用的CU、PU或子块),先执行水平插值,然后执行垂直插值。

[0163] c. 在一个示例中,亮度分量和色度分量都遵循相同的插值顺序。

[0164] d. 可替代地,当一个色度编码块对应于多个亮度编码块时(例如,对于4:2:0的颜色格式,一个色度 $4 \times 4$ 块可对应于两个 $8 \times 4$ 或 $4 \times 8$ 亮度块),亮度和色度可使用不同的插值顺序。

[0165] e. 在一个示例中,当利用不同的插值顺序时,可以相应地进一步改变多个阶段中的缩放因子(即,shift1和shift2)。

[0166] 2. 可替代地,另外,提出亮度分量的插值顺序还可以取决于MV。

[0167] a. 在一个示例中,如果垂直MV分量指向四分之一像素位置并且水平MV分量指向半像素位置,则先执行水平插值,然后执行垂直插值。

[0168] b. 在一个示例中,如果垂直MV分量指向半像素位置并且水平MV分量指向四分之一像素位置,则先执行垂直插值,然后执行水平插值。

[0169] c. 在一个示例中,所提出的方法仅应用于方形编码块。

[0170] 3. 所提出的方法可以应用于某些模式、块尺寸/形状和/或某些子块尺寸。

[0171] a. 所提出的方法可以应用于某些模式,诸如双向预测模式。

[0172] b. 所提出的方法可以应用于某些块尺寸。

[0173] i. 在一个示例中,它仅应用于 $w \times h \leq T1$ 的块,其中 $w$ 和 $h$ 是当前块的宽度和高度,并且 $T1$ 是第一阈值,其可以是取决于设计要求的预定义值,诸如16、32或64。

[0174] ii. 在一个示例中,它仅应用于 $h \leq T2$ 的块,并且 $T2$ 是第二阈值,其可以是取决于设计要求的预定义值,诸如4或8。

[0175] c. 所提出的方法可以应用于某些颜色分量(诸如仅亮度分量)。

[0176] 4. 提出当对一个块应用多假设预测时,与应用于普通预测模式的那些滤波器相比,可以应用短抽头或不同的插值滤波器。

[0177] a. 在一个示例中,可以使用双线性滤波器。

[0178] b. 短抽头或第二插值滤波器可以应用于涉及多个参考块的参考图片列表,而对于仅具有一个参考块的另一参考图片,可以应用与用于普通预测模式的滤波器相同的滤波器。

[0179] c. 所提出的方法可以在某些条件下应用,诸如包含该块的某些(一个或多个)时域层、块/片/条带/图片的量化参数在范围内(诸如大于阈值)。

[0180] 图17是视频处理装置1700的框图。装置1700可以用于实现本文描述的一个或多个方法。装置1700可以嵌入在智能电话、平板电脑、计算机、物联网(IoT)接收机等中。装置1700可以包括一个或多个处理器1702、一个或多个存储器1704和视频处理硬件1706。(一个或多个)处理器1702可以被配置为实现本文件中描述的一个或多个方法。(一个或多个)存储器1704可以用于存储用于实现本文描述的方法和技术的代码和数据。视频处理硬件1706可用于在硬件电路中实现本文件中描述的一些技术。

[0181] 图19是视频比特流处理的方法1900的流程图。方法1900包括确定(1905)视频块的形状,基于视频块确定(1910)插值顺序,该插值顺序指示执行水平插值和垂直插值的序列,并根据视频块的插值顺序执行水平插值和垂直插值,以重建(1915)视频块的解码表示。

[0182] 图20是视频比特流处理的方法2000的流程图。方法2000包括确定(2005)与视频块相关的运动矢量的特性,基于运动矢量的特性确定(2010)视频块的插值顺序,该插值顺序指示执行水平插值和垂直插值的序列,并根据视频块的插值顺序执行水平插值和垂直插值,以重建(2015)视频块的解码表示。

[0183] 参考方法1900和2000,在本文件的第4章节中描述了一些执行水平插值和垂直插值的序列的示例及其使用。例如,如第4章节所述,在视频块的不同形状下,可以优先首先执行水平插值或垂直插值中的一个。在一些实施例中,水平插值先于垂直插值执行,并且在一些实施例中,垂直插值先于水平插值执行。

[0184] 参考方法1900和2000,可以在视频比特流中对视频块进行编码,其中可以通过使用与插值顺序相关的比特流生成规则来实现比特效率,该插值顺序也取决于视频块的形状。

[0185] 应当理解,所公开的技术可以嵌入在视频编码器或解码器中,以在被压缩的编码单元具有与传统方形块或半方形矩形块明显不同的形状时改进压缩效率。例如,使用诸如 $4 \times 32$ 或 $32 \times 4$ 尺寸单元的长或高编码单元的新编码工具可受益于所公开的技术。

[0186] 图21是视频处理方法2100的示例的流程图。方法2100包括:确定(2102)应用于第一视频块的第一预测模式;通过对第一视频块应用水平插值和/或垂直插值,在第一视频块和第一视频块的编码表示之间执行(2104)第一转换;确定(2106)应用于第二视频块的第二预测模式;通过对第二视频块应用水平插值和/或垂直插值,在第二视频块和第二视频块的编码表示之间执行(2108)第二转换,其中,基于第一预测模式是多假设预测模式而第二预测模式不是多假设预测模式的确定,第一视频块的水平插值和垂直插值中的一个或两个使用与用于第二视频块的滤波器相比的较短抽头滤波器。

[0187] 图22是视频比特流处理的方法2200的流程图。方法包括:确定(2205)视频块的形

状;基于视频块的形状确定(2210)插值顺序,插值顺序指示执行水平插值和垂直插值的序列,以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以构造(2215)视频块的编码表示。

[0188] 图23是视频比特流处理的方法2300的流程图。该方法包括:确定(2305)与视频块相关的运动矢量的特征;基于运动矢量的特征确定(2310)插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以构造(2315)视频块的编码表示。

[0189] 可以在以下实施例列表中描述本文件中公开的各种实施例和技术。

[0190] 1.一种视频处理方法,包括:确定应用于第一视频块的第一预测模式;通过对第一视频块应用水平插值和/或垂直插值,在第一视频块和第一视频块的编码表示之间执行第一转换;确定应用于第二视频块的第二预测模式;通过对第二视频块应用水平插值和/或垂直插值,在第二视频块和第二视频块的编码表示之间执行第二转换,其中,基于第一预测模式是多假设预测模式而第二预测模式不是多假设预测模式的确定,第一视频块的水平插值和垂直插值中的一个或两个使用与用于第二视频块的滤波器相比的较短抽头滤波器。

[0191] 2.根据示例1的方法,其中,第一视频块利用多于两个参考块进行转换以用于双向预测,并至少对于一个参考图片列表,其使用多于两个参考块。

[0192] 3.根据示例1的方法,其中利用多于一个参考块对第一视频块进行转换以用于单向预测。

[0193] 4.根据示例1-3中任一个的方法,其中较短抽头滤波器是双线性滤波器。

[0194] 5.根据示例1-3中任一个的方法,其中水平插值和垂直插值中的一个或两个对与多个参考块相关的参考图片列表使用较短抽头滤波器。

[0195] 6.根据示例1-5中任一个的方法,其中,当参考图片列表与单个参考块相关时,水平插值或垂直插值中的一个或两个使用与用于普通预测模式相同的滤波器。

[0196] 7.根据示例1-6中任一个的方法,其中,基于以下中的一个或多个的确定来应用方法:时域层的使用、包含视频块的一个或多个块、片、条带或图片的量化参数在阈值范围内。

[0197] 8.根据示例7的方法,其中在阈值范围内的量化参数包括大于阈值的量化参数。

[0198] 9.根据示例6的方法,其中,普通预测模式包括单向预测或双向预测帧间预测模式,单向预测使用具有至多一个运动矢量和一个参考索引的帧间预测来预测块中的样本的样本值,双向预测帧间预测模式使用具有至多两个运动矢量和参考索引的帧间预测来预测块中样本的样本值。

[0199] 10.一种视频解码装置,包括处理器,被配置为实现示例1至9的一个或多个的方法。

[0200] 11.一种视频编码装置,包括处理器,被配置为实现示例1至9的一个或多个的方法。

[0201] 12.一种其上存储了代码的计算机可读程序介质,代码包括指令,当处理器执行指令时,使处理器实现示例1至9的一个或多个的方法。

[0202] 13.一种视频比特流处理方法,包括:确定视频块的形状;基于视频块的形状确定插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以重建视频块的解码表示。



[0203] 14. 根据示例13的方法,其中,视频块的形状由视频块的宽度和高度表示,并且确定插值顺序的步骤还包括:

[0204] 当视频块的宽度大于视频块的高度时,确定在水平插值之前执行垂直插值作为插值顺序。

[0205] 15. 根据示例13的方法,其中,视频块的形状由宽度和高度表示,并且确定插值顺序的步骤还包括:

[0206] 当视频块的宽度大于或等于视频块的高度时,确定在水平插值之前执行垂直插值作为插值顺序。

[0207] 16. 根据示例13的方法,其中,视频块的形状由宽度和高度表示,并且确定插值顺序的步骤还包括:

[0208] 当视频块的高度大于或等于视频块的宽度时,确定在垂直插值之前执行水平插值作为插值顺序。

[0209] 17. 根据示例1的方法,其中,视频块的形状由宽度和高度表示,并且确定插值顺序的步骤还包括:

[0210] 当视频块的高度大于视频块的宽度时,确定在垂直插值之前执行水平插值作为插值顺序。

[0211] 18. 根据示例1的方法,其中,基于所述插值顺序或基于不同的插值顺序,对视频块的亮度分量和色度分量进行插值。

[0212] 19. 根据示例1的方法,其中,当色度分量的每个色度块对应于亮度分量的多个亮度块时,使用不同的插值顺序对视频块的亮度分量和色度分量进行插值。

[0213] 20. 根据示例13的方法,其中使用不同的插值顺序对视频块的亮度分量和色度分量进行插值,并且其中对于亮度分量和色度分量,在水平插值和垂直插值中使用的缩放因子不同。

[0214] 21. 一种视频比特流处理方法,包括:确定与视频块相关的运动矢量的特征;基于运动矢量的特征确定插值顺序,插值顺序指示执行水平插值和垂直插值的序列;以及按由插值顺序指示的序列对视频块执行水平插值和垂直插值,以重建视频块的解码表示。

[0215] 22. 根据示例21的方法,其中,运动矢量的特征由运动矢量指向的四分之一像素位置和半像素位置表示,运动矢量包括垂直分量和水平分量,并且确定插值顺序包括:当垂直分量指向四分之一像素位置并且水平分量指向半像素位置时,确定在垂直插值之前执行水平插值作为插值顺序。

[0216] 23. 根据示例21的方法,其中,运动矢量的特征由运动矢量指向的四分之一像素位置和半像素位置表示,运动矢量包括垂直分量和水平分量,并且确定插值顺序包括:当垂直分量指向半像素位置并且水平分量指向四分之一像素位置时,确定在水平插值之前执行垂直插值。

[0217] 24. 根据示例21-23中任一个的方法,其中视频块的形状是正方形。

[0218] 25. 根据示例21-24中任一个的方法,其中方法应用于双预测模式。

[0219] 26. 根据示例21-25中任一个的方法,其中,当视频块的高度乘以视频块的宽度小于或等于 $T1$ 时,应用方法, $T1$ 是第一阈值。

[0220] 27. 根据示例21-25中任一个的方法,其中,当视频块具有小于或等于 $T2$ 的高度时,

应用方法, T2是第二阈值。

[0221] 28. 根据示例21-25中任一个的方法, 其中将方法应用于视频块的亮度分量。

[0222] 29. 一种视频比特流处理方法, 包括:

[0223] 确定视频块的形状;

[0224] 基于视频块的形状确定插值顺序, 插值顺序指示执行水平插值和垂直插值的序列; 以及

[0225] 按由插值顺序指示的序列对视频块执行水平插值和垂直插值, 以构造视频块的编码表示。

[0226] 30. 一种视频比特流处理方法, 包括:

[0227] 确定与视频块相关的运动矢量的特征;

[0228] 基于运动矢量的特征确定插值顺序, 插值顺序指示执行水平插值和垂直插值的序列; 以及

[0229] 按由插值顺序指示的序列对视频块执行水平插值和垂直插值, 以构造视频块的编码表示。

[0230] 31. 一种视频解码装置, 包括处理器, 其被配置为实现示例21至28的一个或多个的方法。

[0231] 32. 一种视频编码装置, 包括处理器, 其被配置为实现示例29或30的方法。

[0232] 33. 一种计算机程序产品, 其上存储有计算机代码, 代码在由处理器执行时使处理器实现示例13至30中任一个的方法。

[0233] 34. 一种视频系统中的装置, 包括处理器和其上具有指令的非暂时性存储器, 其中指令在由处理器执行时使处理器实现示例13至30中任一个的方法。

[0234] 从上述来看, 应当理解的是, 为了便于说明, 本发明公开的技术的具体实施例已经在本文中进行了描述, 但是可以在不偏离本发明范围的情况下进行各种修改。因此, 除了的之外, 本发明公开的技术不限于权利要求的限定。

[0235] 本专利文件中主题名称的实现和功能操作可以在各种系统、数字电子电路、或计算机软件、固件或硬件中实现, 包括本说明书中所公开的结构及其结构等效体, 或其中一个或多个的组合。在本说明书中描述的主题的实现可以实现为一个或多个计算机程序产品, 即一个或多个编码在暂时性和非暂时性计算机可读介质上的计算机程序指令的模块, 以供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、存储设备、影响机器可读传播信号的物质组成或其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”包括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、计算机或多处理器或计算机组。除硬件外, 该装置还可以包括为计算机程序创建执行环境的代码, 例如, 构成处理器固件的代码、协议栈、数据库管理系统、操作系统或其中一个或多个的组合。

[0236] 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(包括编译语言或解释语言)编写, 并且可以以任何形式部署, 包括作为独立程序或作为模块、组件、子程序或其他适合在计算环境中使用的单元。计算机程序不一定与文件系统中的文件对应。程序可以存储在保存其他程序或数据的文件的部分中(例如, 存储在标记语言文档中的一个或多个脚本)、专用于该程序的单个文件中、或多个协调文件(例如, 存储一个

或多个模块、子程序或部分代码的文件)中。计算机程序可以部署在一台或多台计算机上来执行,这些计算机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。

[0237] 本说明书中描述的处理和逻辑流可以通过一个或多个可编程处理器执行,该处理器执行一个或多个计算机程序,通过在输入数据上操作并生成输出来执行功能。处理和逻辑流也可以通过特殊用途的逻辑电路来执行,并且装置也可以实现为特殊用途的逻辑电路,例如,FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0238] 例如,适于执行计算机程序的处理器包括通用和专用微处理器,以及任何类型数字计算机的任何一个或多个。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执行指令的处理器和存储指令和数据的一个或多个存储设备。通常,计算机还将包括一个或多个用于存储数据的大容量存储设备,例如,磁盘、磁光盘或光盘,或通过操作耦合到一个或多个大容量存储设备来从其接收数据或将数据传输到一个或多个大容量存储设备,或两者兼有。然而,计算机不一定具有这样的设备。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来补充,或合并到专用逻辑电路中。

[0239] 说明书和附图旨在被认为是示例性的,其中示例性意味着示例。如这里所使用的,单数形式“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,“或”的使用旨在包括“和/或”,除非上下文另有明确说明。

[0240] 虽然本专利文件包含许多细节,但不应将其解释为对任何发明或权利要求范围的限制,而应解释为对特定发明的特定实施例的特征的描述。本专利文件在单独实施例的上下文描述的一些特征也可以在单个实施例中组合实施。相反,在单个实施例的上下文中描述的各种功能也可以在多个实施例中单独实施,或在任何合适的子组合中实施。此外,尽管上述特征可以描述为在一些组合中起作用,甚至最初要求是这样,但在一些情况下,可以从组合中移除权利要求组合中的一个或多个特征,并且权利要求的组合可以指向子组合或子组合的变体。

[0241] 同样,尽管图纸中以特定顺序描述了操作,但这不应理解为要获得想要的结果必须按照所示的特定顺序或顺序执行此类操作,或执行所有说明的操作。此外,本专利文件所述实施例中各种系统组件的分离不应理解为在所有实施例中都需要这样的分离。

[0242] 仅描述了一些实现和示例,其他实现、增强和变体可以基于本专利文件中描述和说明的内容做出。

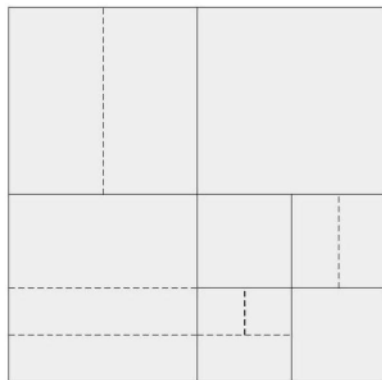
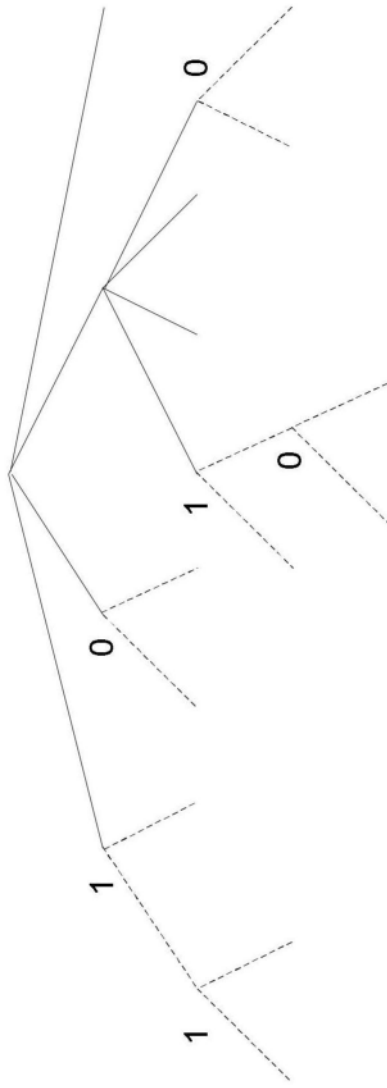


图1

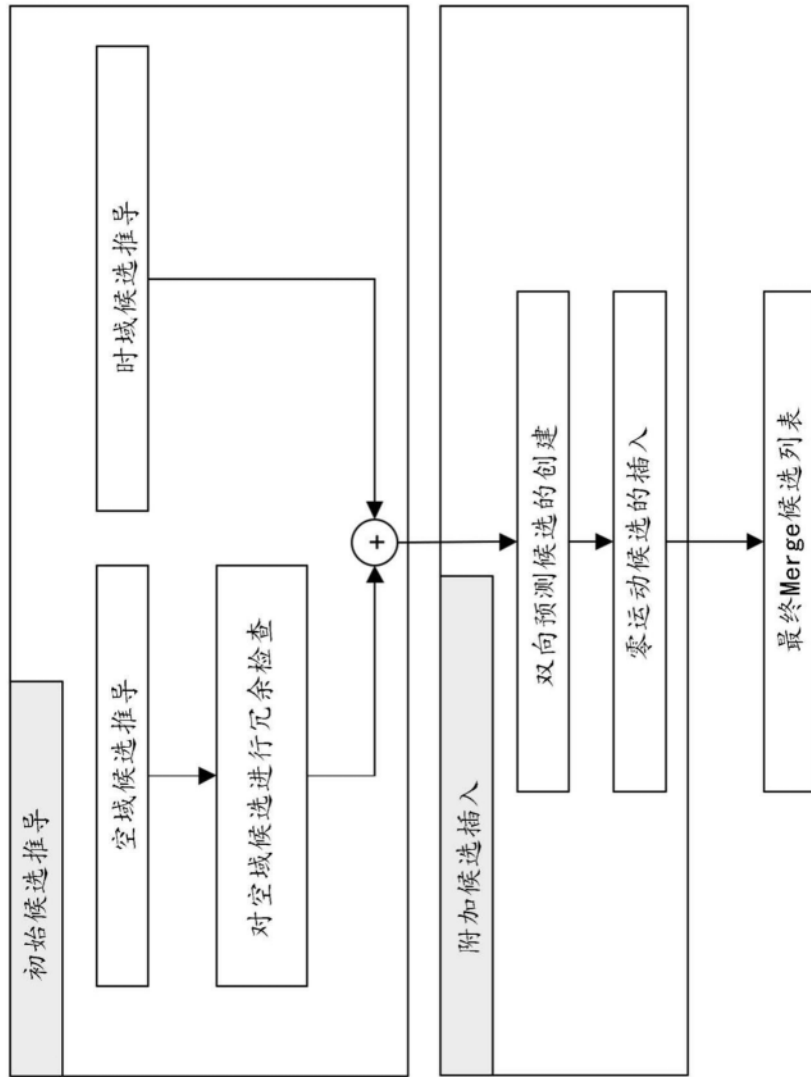


图2

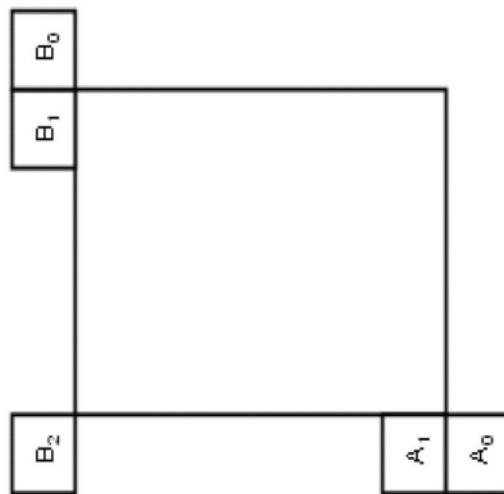


图3

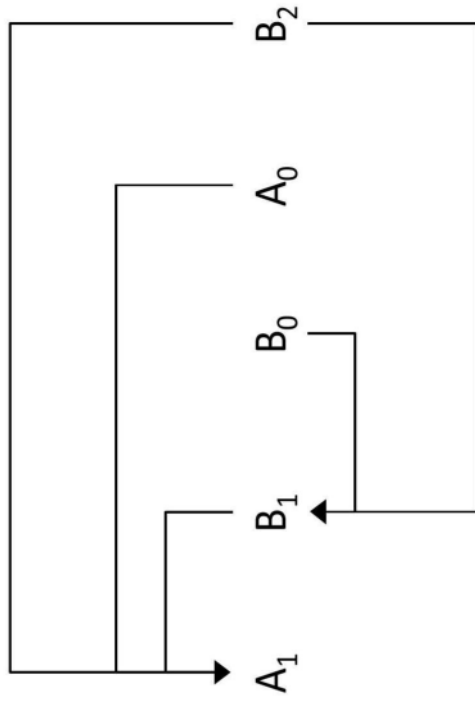


图4

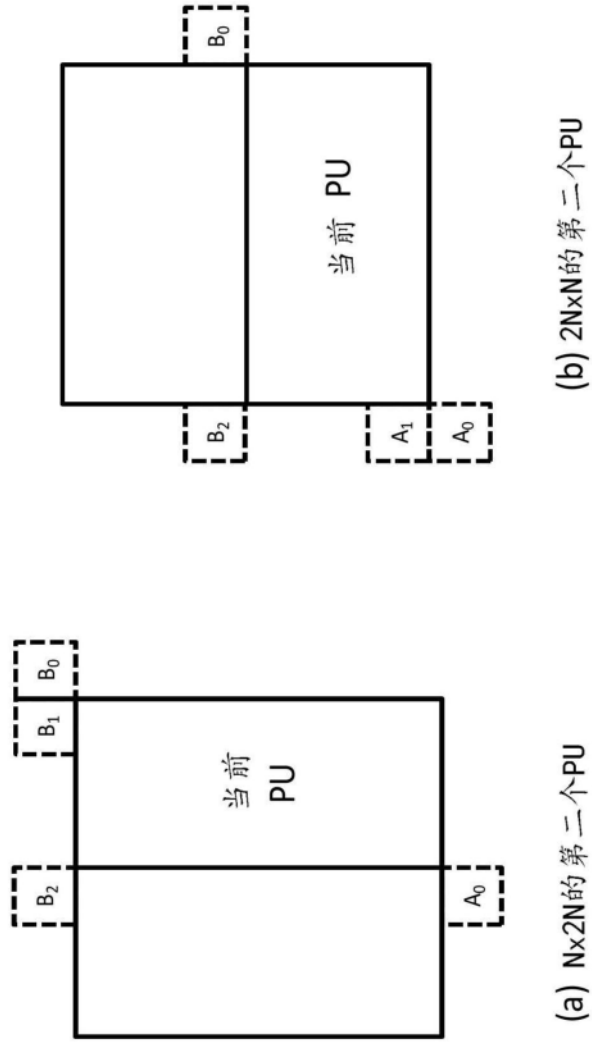


图5

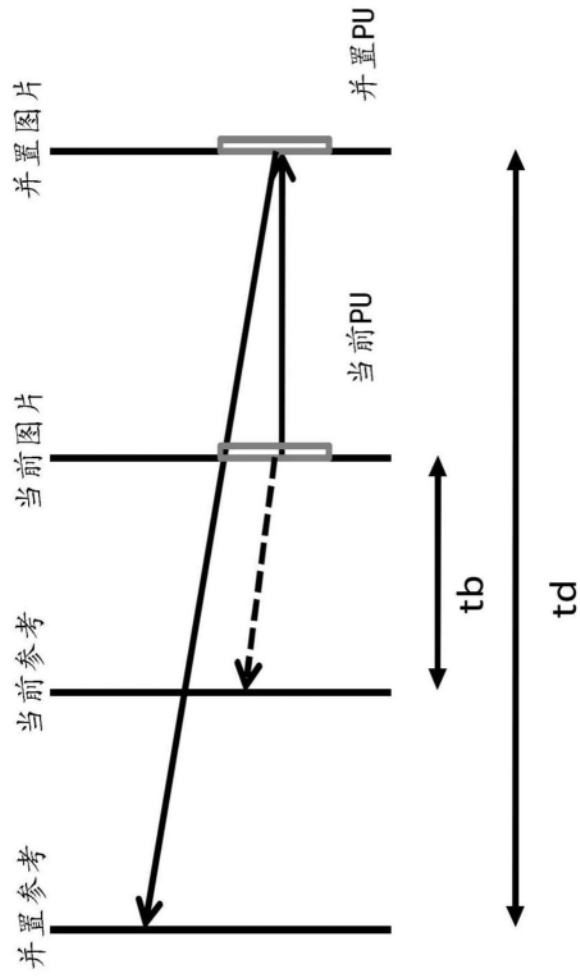


图6



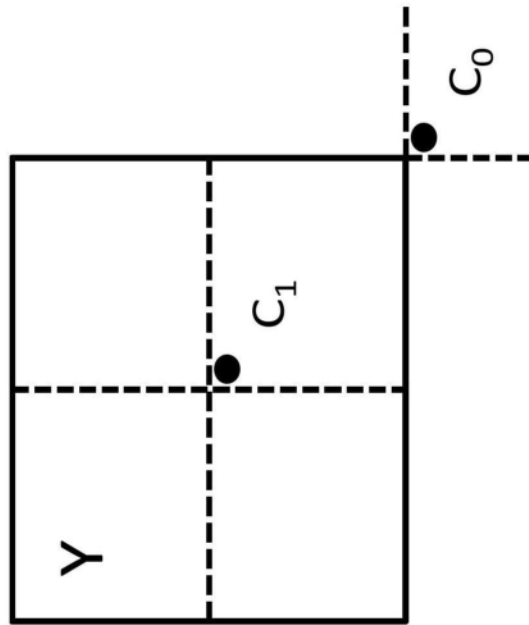


图7

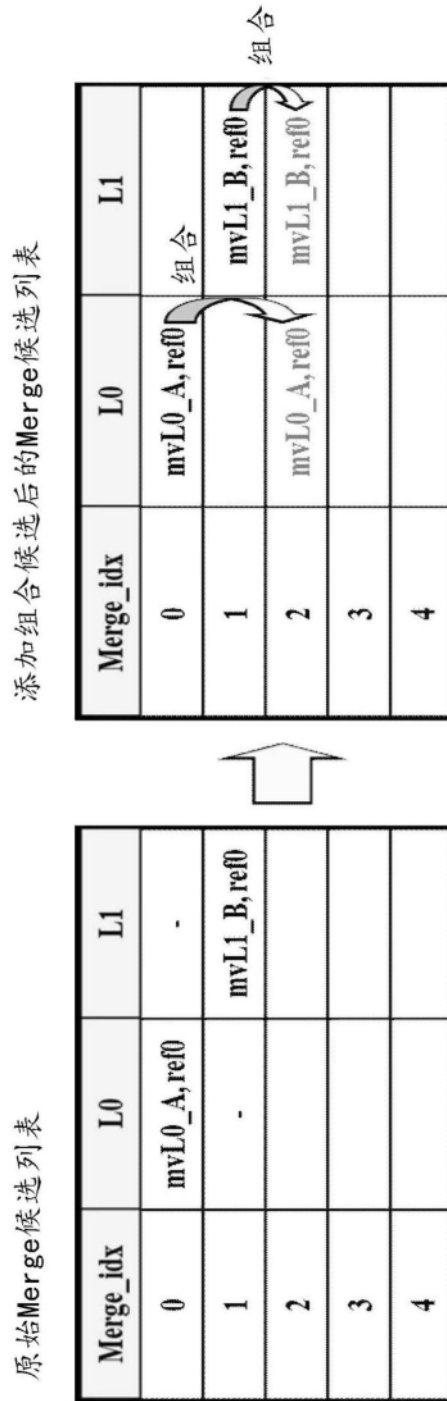


图8

对于refidx作为输入的每个参考图片列表

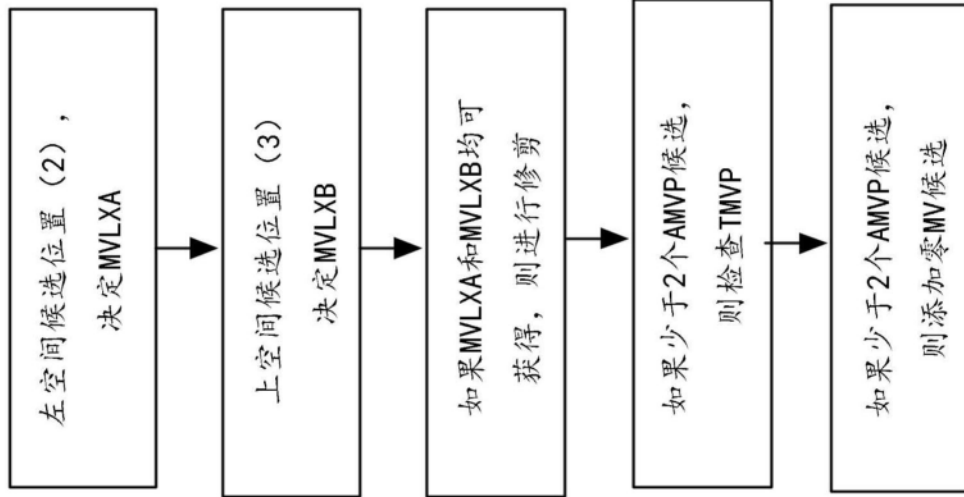


图9

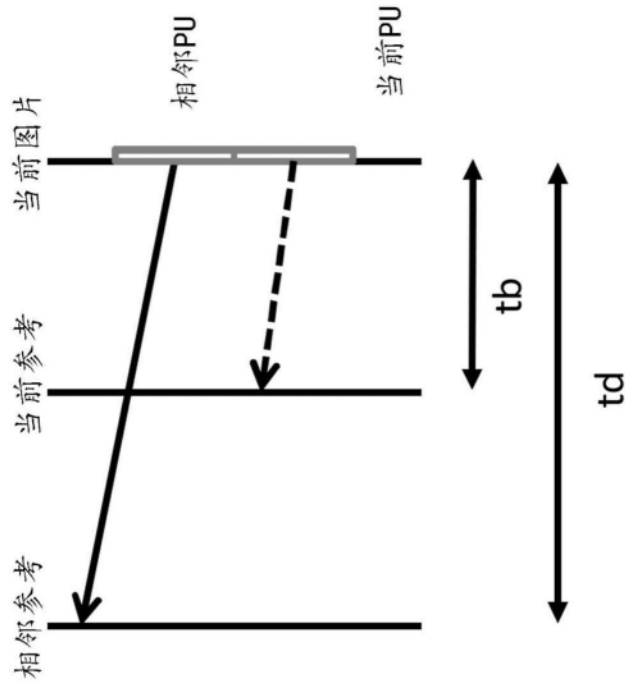


图10

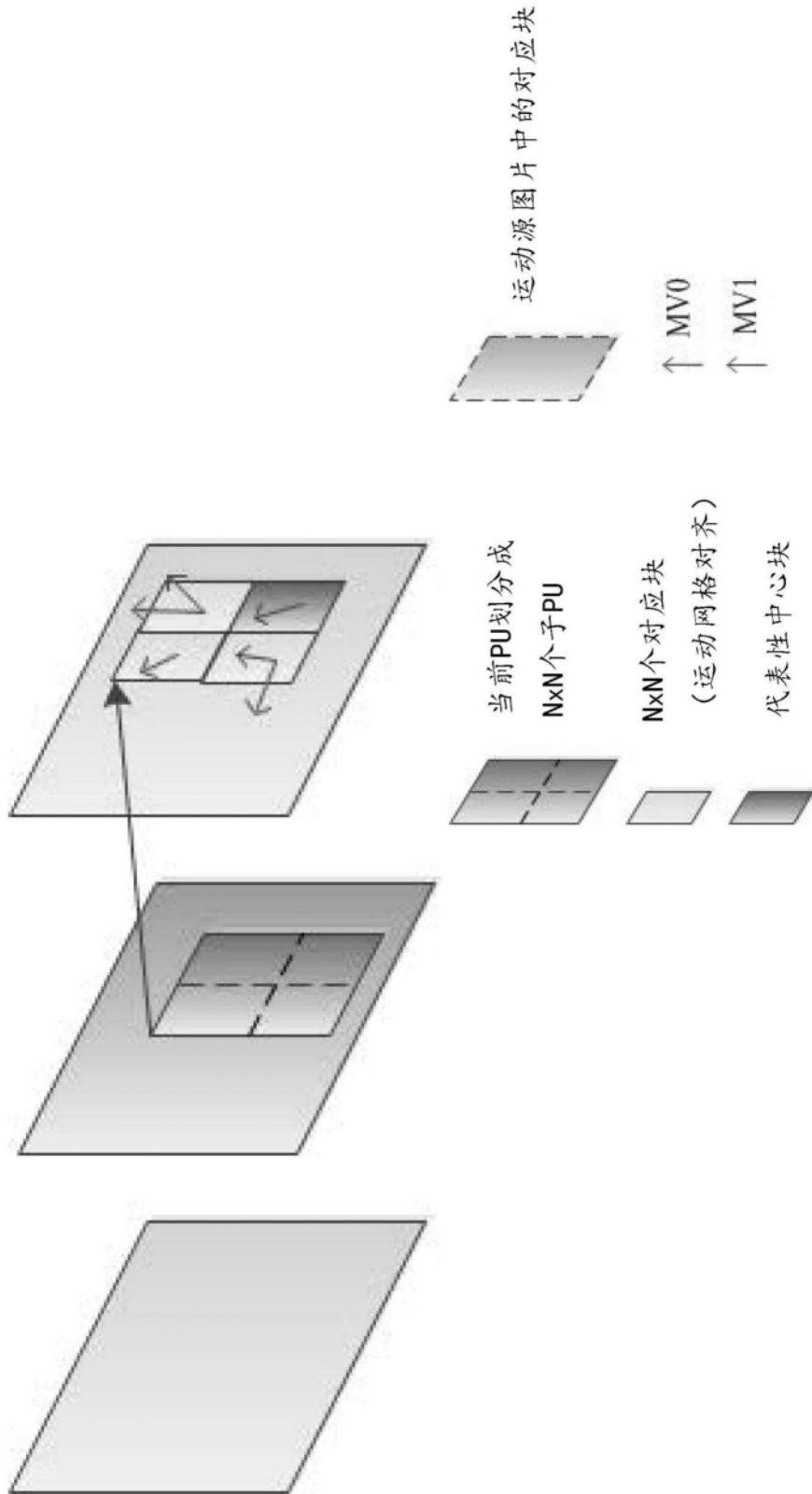


图11

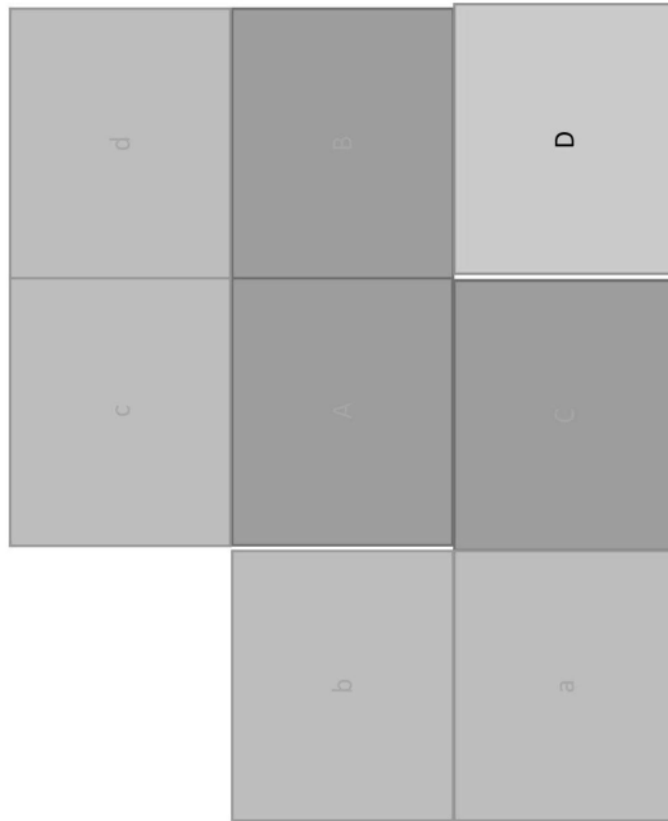


图12

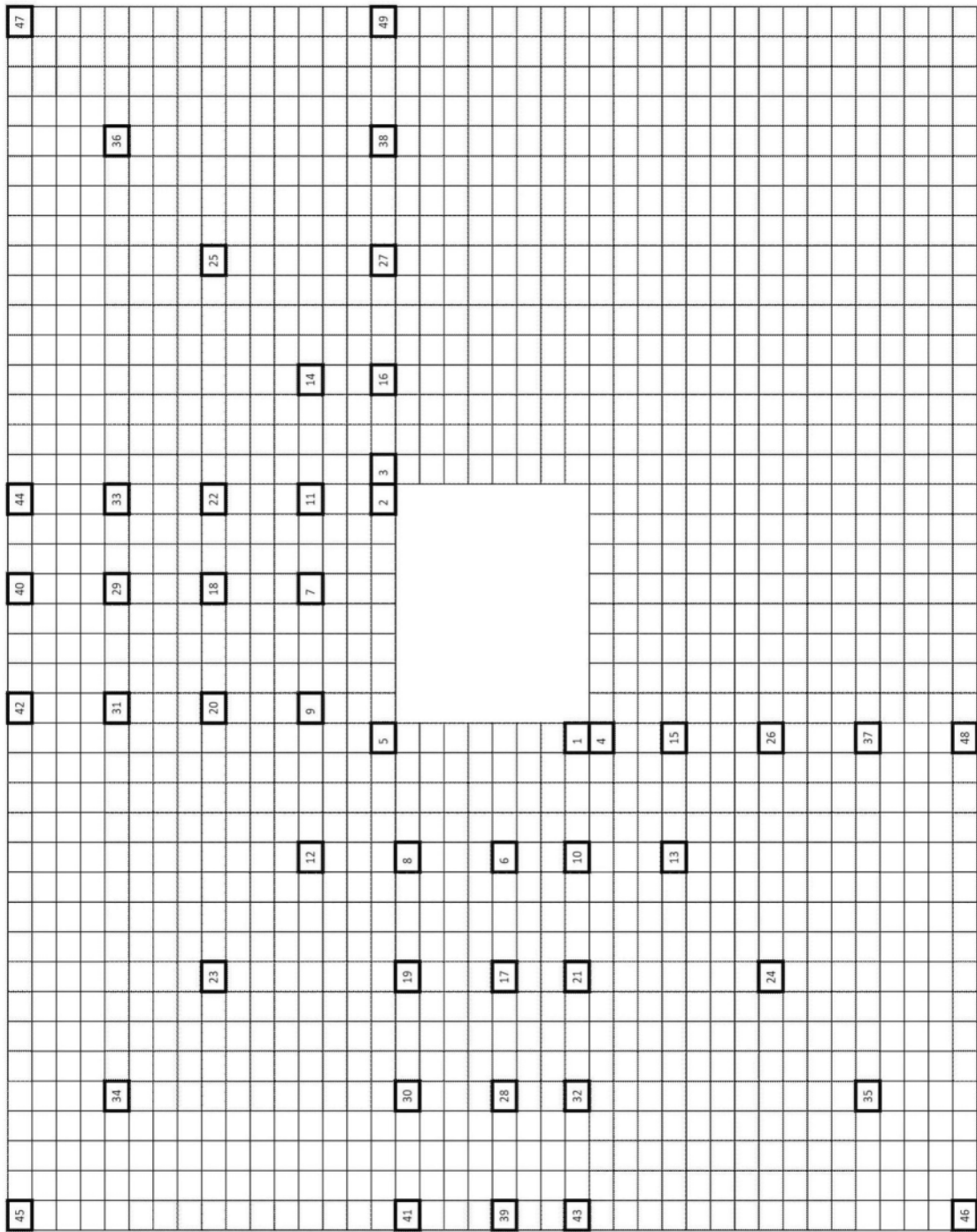


图13

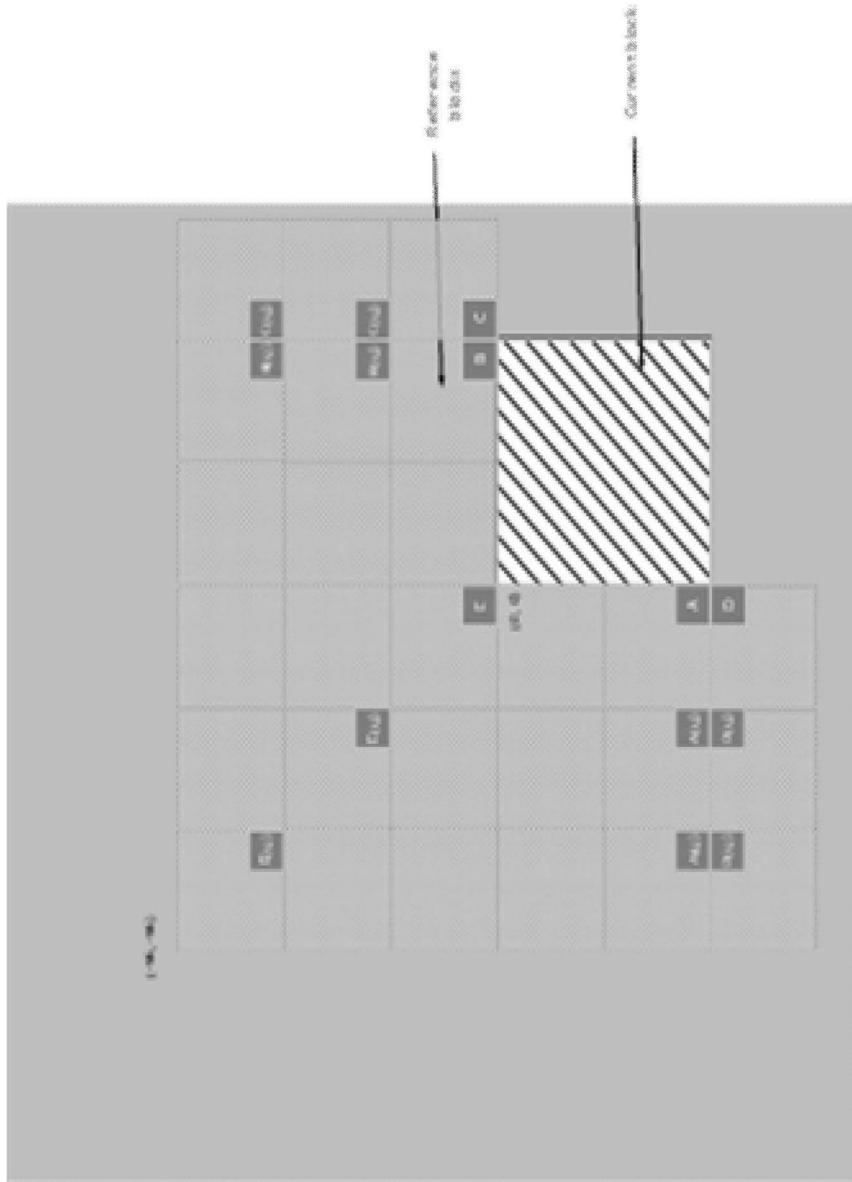


图14

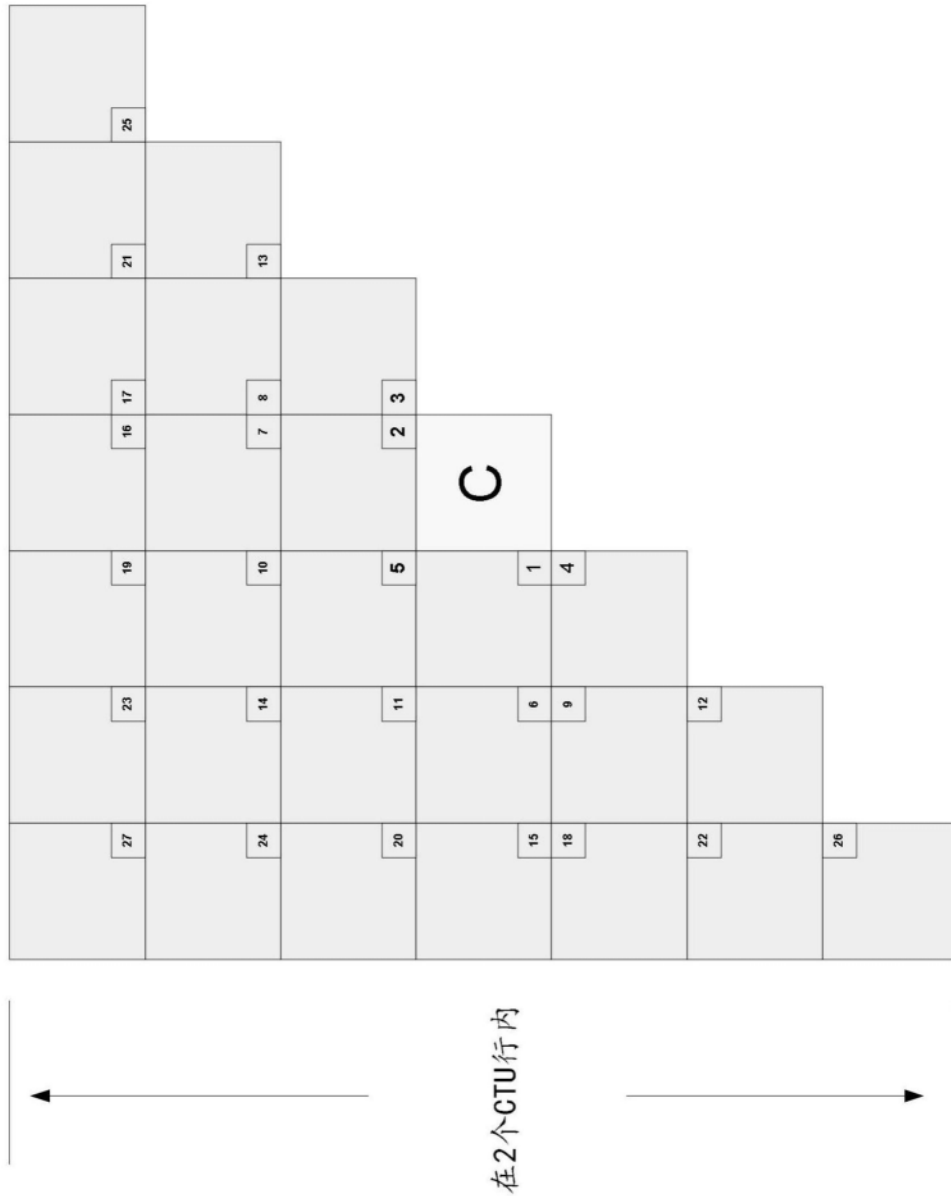


图15





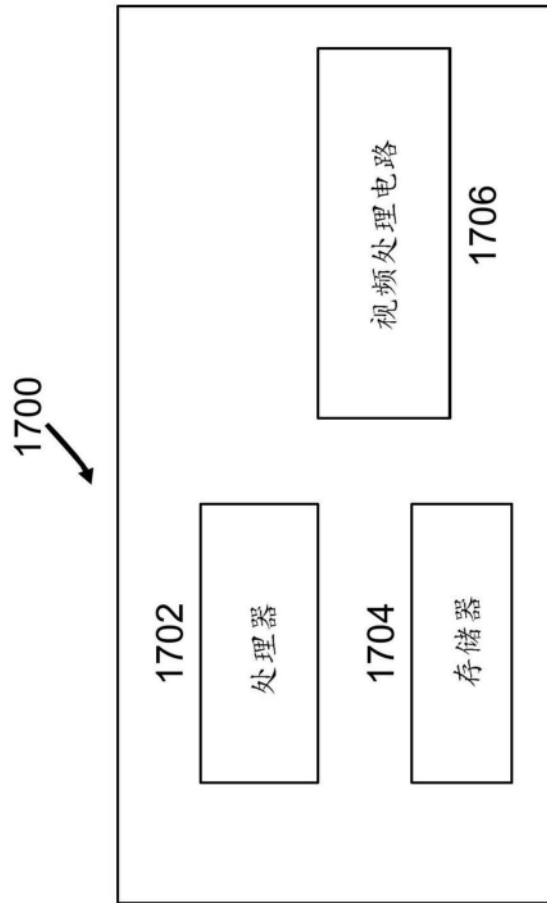


图17

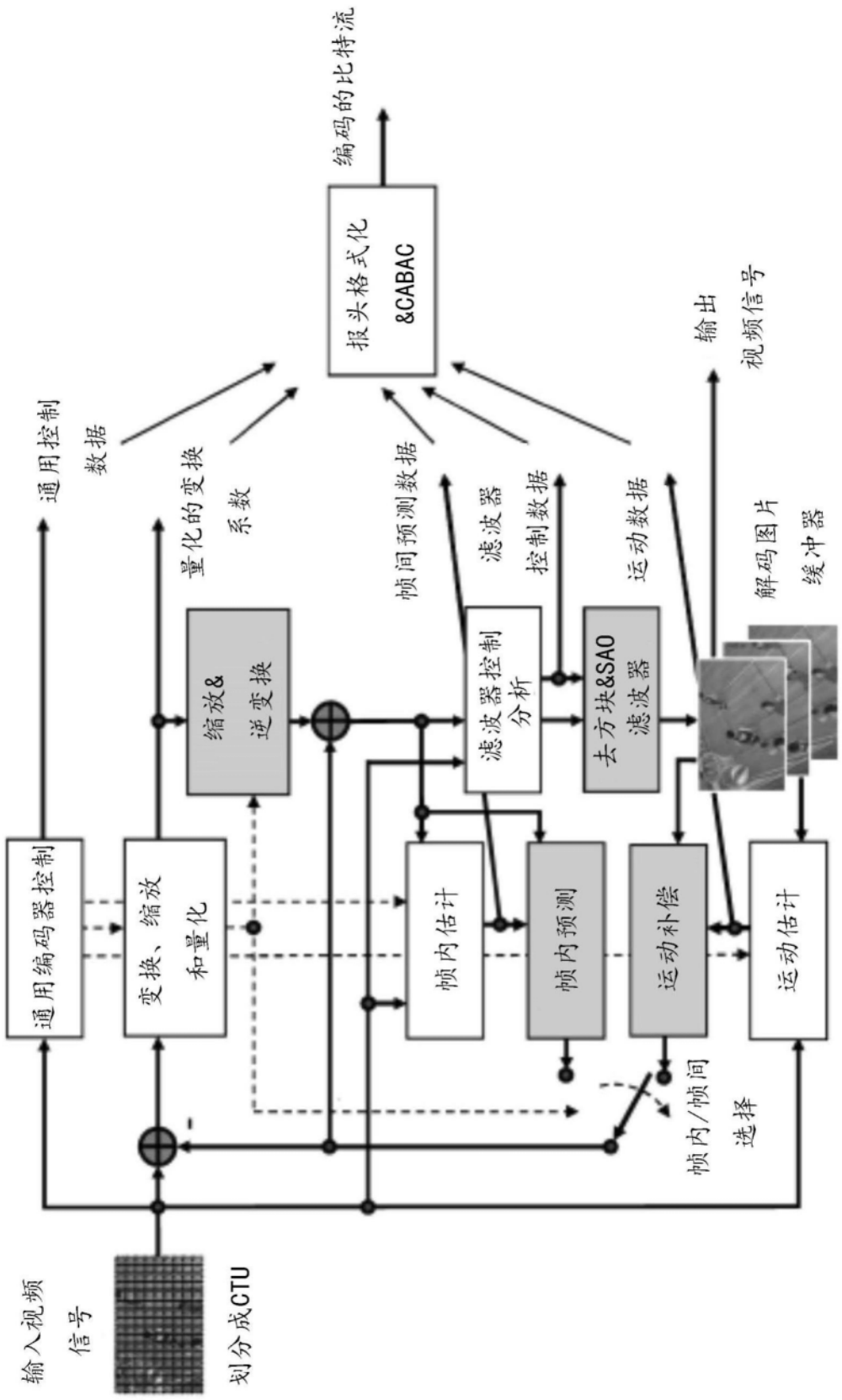


图18

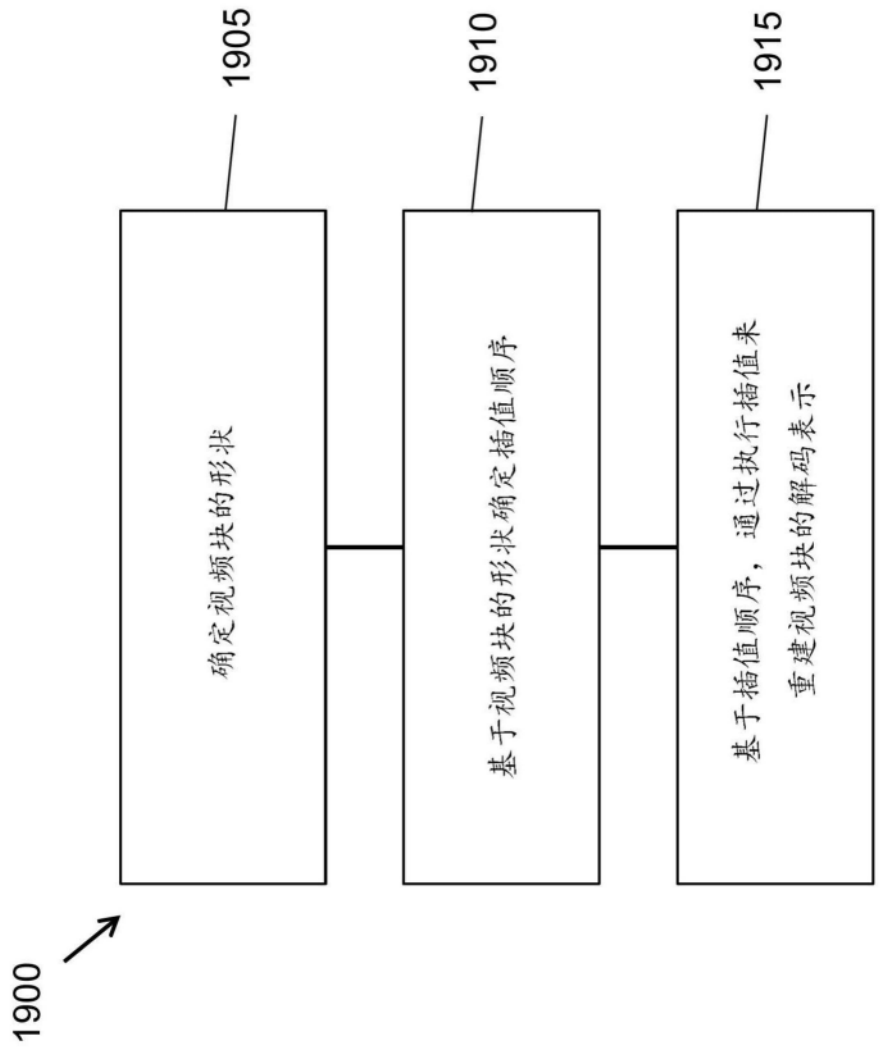


图19

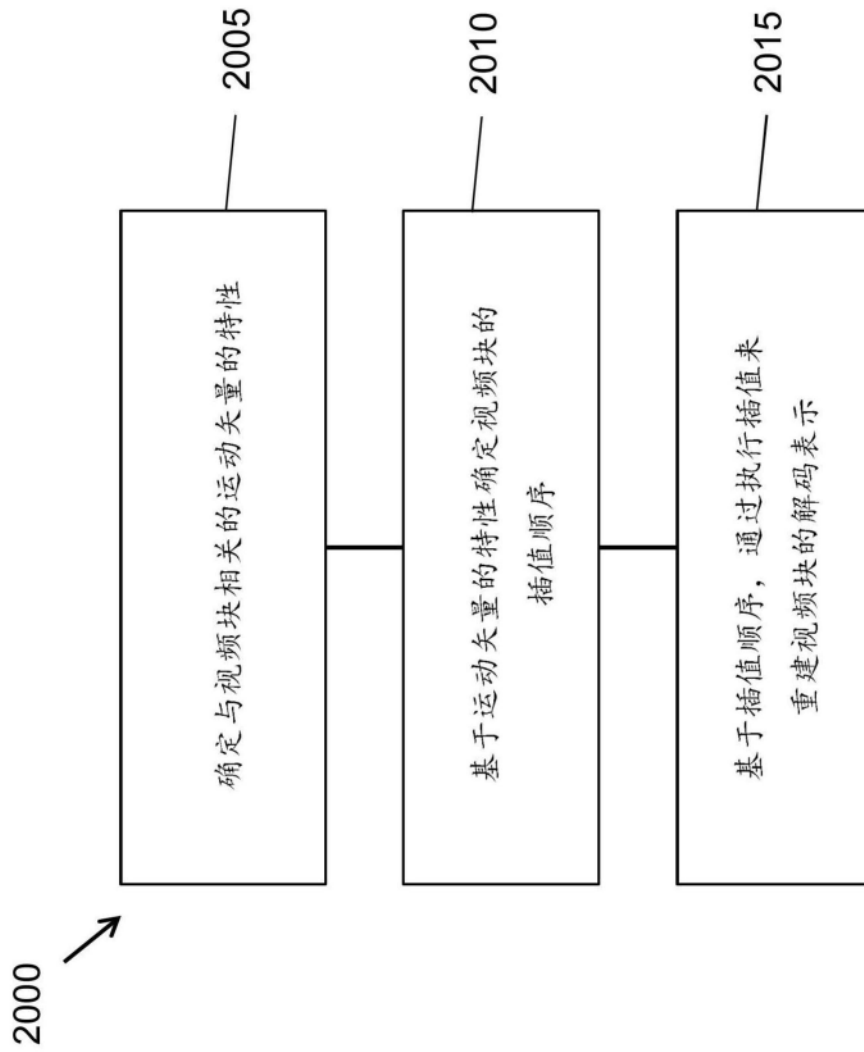


图20

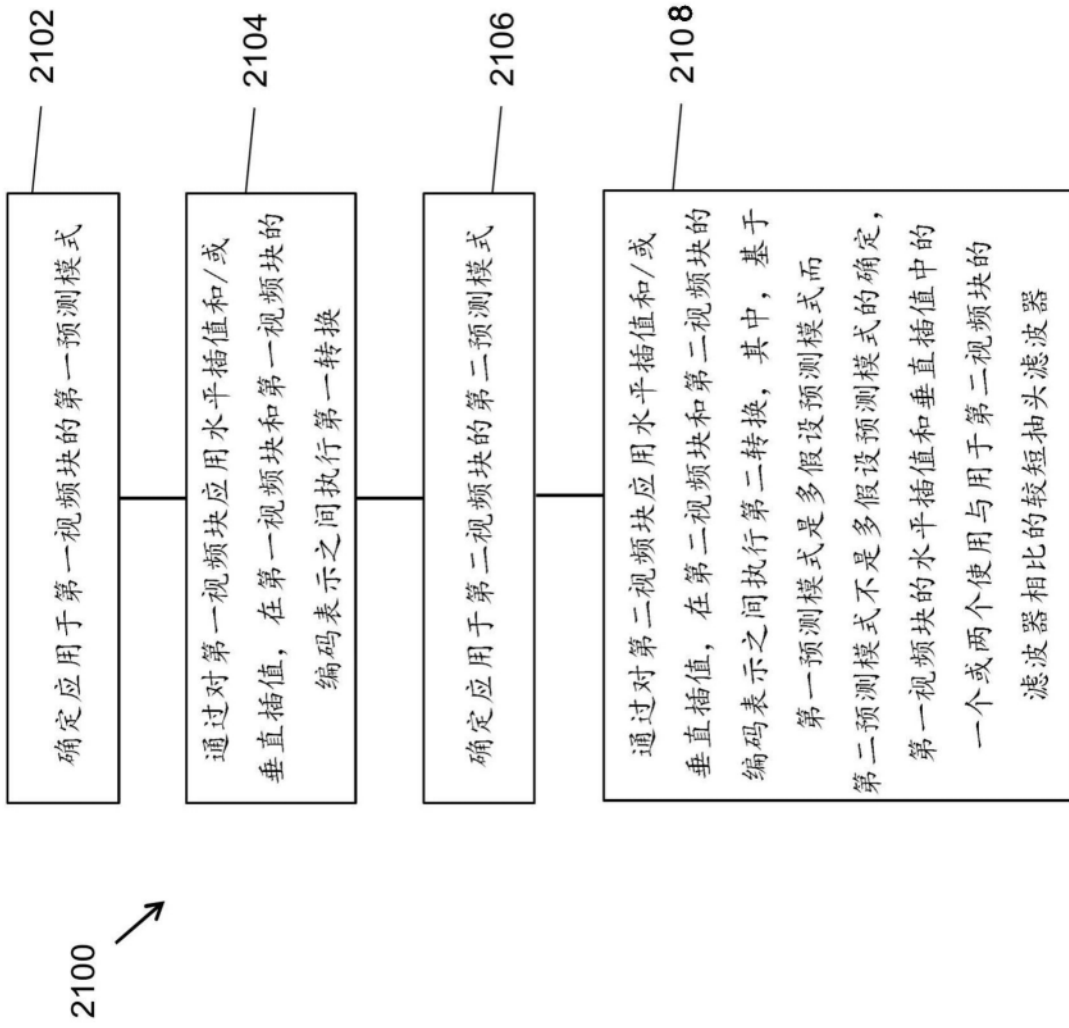


图21

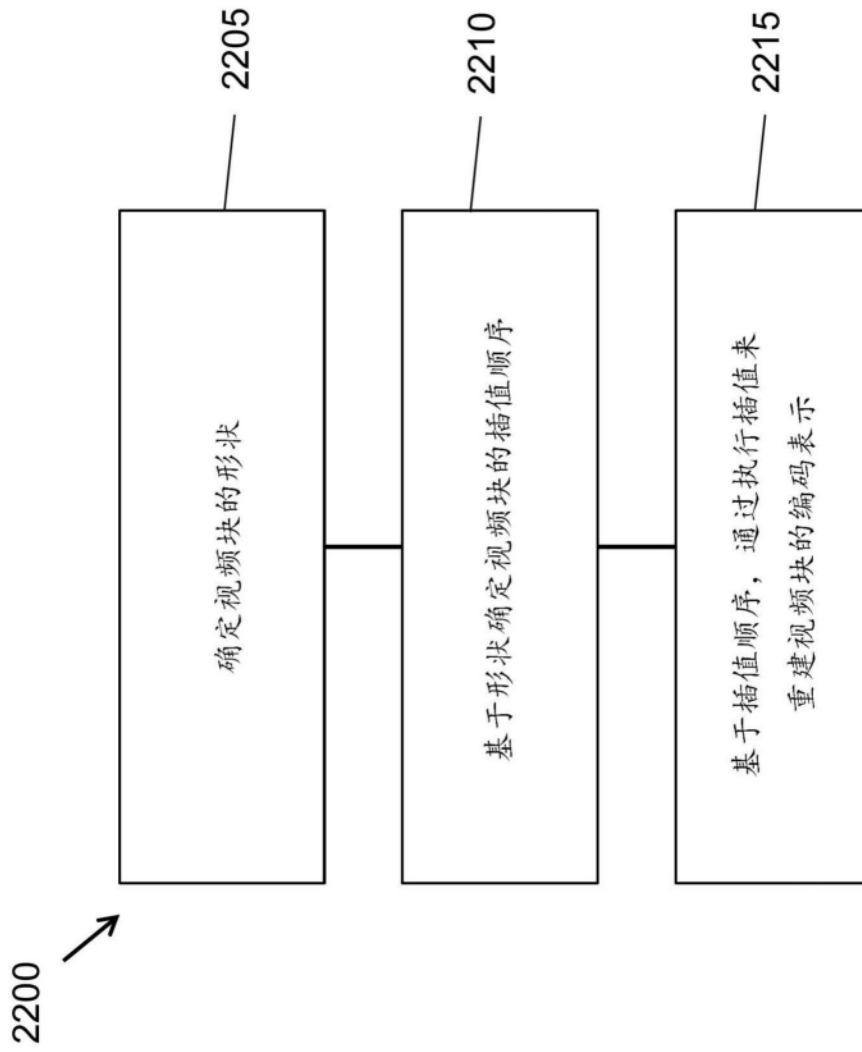


图22

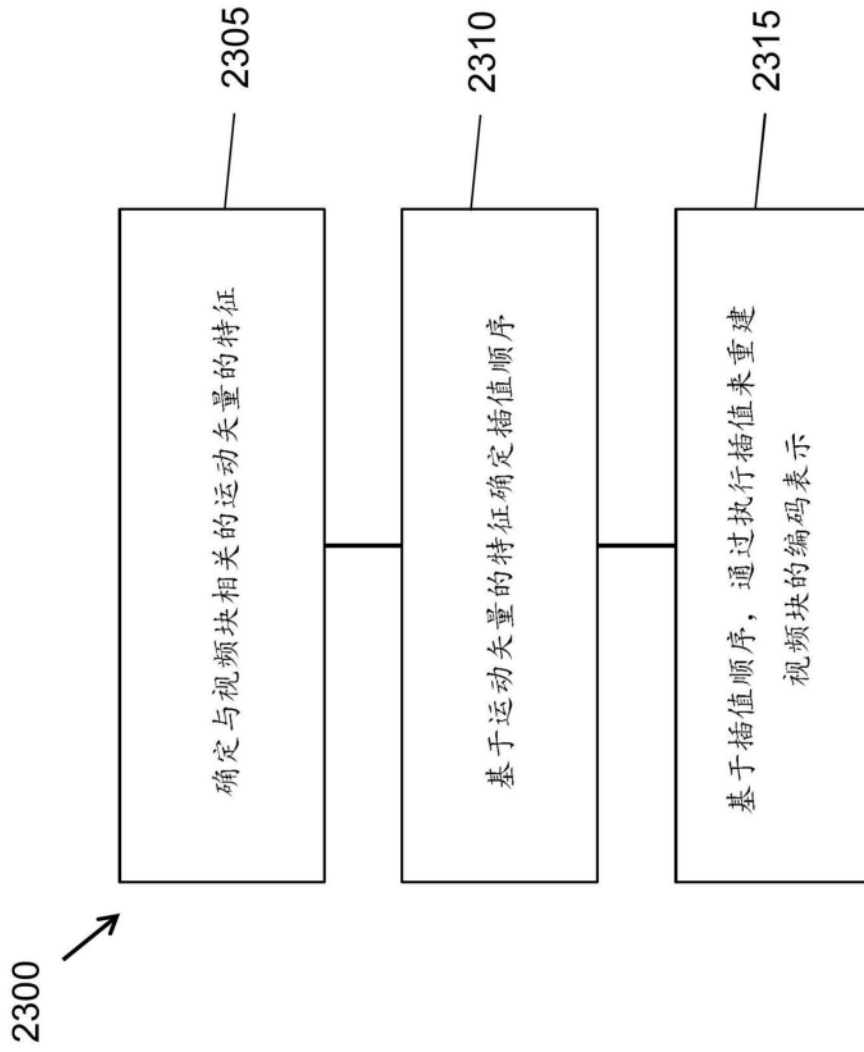


图23