



(12) 发明专利

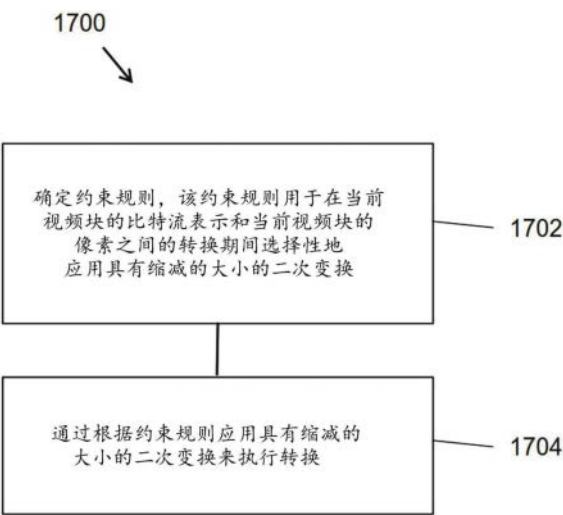
(10) 授权公告号 CN 113767627 B

(45) 授权公告日 2022. 11. 25

(21) 申请号 202080031268.5	(72) 发明人 张凯 张莉 刘鸿彬 许继征 王悦
(22) 申请日 2020.04.23	(74) 专利代理机构 北京市柳沈律师事务所 11105
(65) 同一申请的已公布的文献号 申请公布号 CN 113767627 A	专利代理师 张亮
(43) 申请公布日 2021.12.07	(51) Int.Cl. H04N 19/12 (2006.01)
(66) 本国优先权数据 PCT/CN2019/083853 2019.04.23 CN	(56) 对比文件 CN 109076242 A, 2018.12.21 CN 109076230 A, 2018.12.21 CN 108141596 A, 2018.06.08 CN 109644269 A, 2019.04.16 CN 108141594 A, 2018.06.08 US 2018103252 A1, 2018.04.12 US 2013003856 A1, 2013.01.03 US 2018041776 A1, 2018.02.08 US 2018338143 A1, 2018.11.22
(85) PCT国际申请进入国家阶段日 2021.10.25	审查员 黄驰
(86) PCT国际申请的申请数据 PCT/CN2020/086421 2020.04.23	权利要求书3页 说明书37页 附图31页
(87) PCT国际申请的公布数据 W02020/216296 EN 2020.10.29	
(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	

(54) 发明名称
基于二次变换的视频处理中的裁剪操作

(57) 摘要
一种视频处理方法包括：针对视频的块和视频的比特流表示之间的转换，确定来自具有缩减的大小的逆二次变换（例如，逆低频不可分变换）的输出值被约束在 $[min, max]$ 、包括 min 、 max 在内的范围内。逆二次变换适用于反量化步骤和逆主变换之间的块。缩减的大小是从块的大小缩减的，并且 min 和 max 是整数值。该方法还包括基于该确定来执行转换。



1. 一种处理视频数据的方法,包括:

为视频的当前块和所述视频的比特流之间的转换,确定逆二次变换是否被应用于所述当前块,其中来自所述逆二次变换的输出值被约束在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 的第一范围内,其中所述逆二次变换适用于去量化和逆主变换之间的当前块,并且其中 $\min$ 和 $\max$ 是整数值;以及

基于所述确定执行所述转换,

其中,所述去量化之后的系数被约束到 $[\text{qmin}, \text{qmax}]$ 的第二范围,包括 qmin 、 qmax , qmin 和 qmax 是整数,并且其中所述第一范围和所述第二范围具有如下至少一个关系:(1) $\min$ 等于 qmin ,或者(2) $\max$ 等于 qmax ,

其中,响应于利用非帧内预测模式的预测模式对所述当前块进行编解码,不对所述当前块应用所述逆二次变换,并且

其中,所述去量化中使用的量化矩阵是基于是否应用了所述逆二次变换来确定的。

2. 根据权利要求1所述的方法,其中,所述逆二次变换包括逆低频不可分变换。

3. 根据权利要求1所述的方法,其中,应用裁剪操作来将所述逆二次变换的输出值裁剪在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 的所述第一范围内。

4. 根据权利要求1所述的方法,其中, $\min$ 等于 $-(1 \ll 15)$,并且 $\max$ 等于 $(1 \ll 15) - 1$ 。

5. 根据权利要求1所述的方法,其中,用于所述逆二次变换的矩阵从四个变换集合中选择,所述四个变换集合中的每一个包括两个变换矩阵。

6. 根据权利要求5所述的方法,其中,响应于所述当前块是色度块并且三个跨分量线性模型帧内预测模式之一被用于所述当前块,为所述当前块选择变换集0。

7. 根据权利要求1所述的方法,其中,所述逆二次变换被应用于所述当前块的去量化变换系数。

8. 根据权利要求1所述的方法,其中,响应于利用变换跳过模式对所述当前块进行编解码,不对所述当前块应用所述逆二次变换。

9. 根据权利要求1所述的方法,还包括:

确定正向二次变换被应用于所述当前块,其中所述正向二次变换的输入值被约束在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 的所述第一范围内,其中所述正向二次变换可应用于正向主变换和量化之间的当前块。

10. 根据权利要求1所述的方法,其中,在适用于正向主变换和量化步骤之间的正向二次变换之后的当前块的系数被限制在第三范围内。

11. 根据权利要求1所述的方法,其中,响应于所述逆二次变换未被应用于所述当前块,指示与所述当前块中的二次变换相关的信息的语法元素未被包括在所述当前块的比特流中。

12. 根据权利要求1-11中任一项所述的方法,其中,所述转换包括将所述视频编码成所述比特流。

13. 根据权利要求1-11中任一项所述的方法,其中,所述转换包括从所述比特流解码所述视频。

14. 一种用于处理视频数据的装置,包括处理器和其上具有指令的非暂时性存储器,其中所述指令在由所述处理器执行时使得所述处理器:

为视频的当前块和所述视频的比特流之间的转换,确定逆二次变换是否被应用于所述当前块,其中来自所述逆二次变换的输出值被约束在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 的第一范围内,其中所述逆二次变换适用于去量化和逆主变换之间的当前块,并且其中 $\min$ 和 $\max$ 是整数值;以及

基于所述确定执行所述转换,

其中,所述去量化之后的系数被约束到 $[\text{qmin}, \text{qmax}]$ 的第二范围,包括 qmin 、 qmax , qmin 和 qmax 是整数,并且其中所述第一范围和所述第二范围具有如下至少一个关系:(1) $\min$ 等于 qmin ,或者(2) $\max$ 等于 qmax ,

其中,响应于利用非帧内预测模式的预测模式对所述当前块进行编解码,不对所述当前块应用所述逆二次变换,并且

其中,所述去量化中使用的量化矩阵是基于是否应用了所述逆二次变换来确定的。

15. 根据权利要求14所述的装置,其中,所述逆二次变换包括逆低频不可分变换。

16. 根据权利要求14所述的装置,其中, $\min$ 等于 $-(1 \ll 15)$,并且 $\max$ 等于 $(1 \ll 15) - 1$ 。

17. 一种存储指令的非暂时性计算机可读存储介质,所述指令使得处理器:

为视频的当前块和所述视频的比特流之间的转换,确定逆二次变换是否被应用于所述当前块,其中来自所述逆二次变换的输出值被约束在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 的第一范围内,其中所述逆二次变换适用于去量化和逆主变换之间的当前块,并且其中 $\min$ 和 $\max$ 是整数值;以及

基于所述确定执行所述转换,

其中,所述去量化之后的系数被约束到 $[\text{qmin}, \text{qmax}]$ 的第二范围,包括 qmin 、 qmax , qmin 和 qmax 是整数,并且其中所述第一范围和所述第二范围具有如下至少一个关系:(1) $\min$ 等于 qmin ,或者(2) $\max$ 等于 qmax ,

其中,响应于利用非帧内预测模式的预测模式对所述当前块进行编解码,不对所述当前块应用所述逆二次变换,并且

其中,所述去量化中使用的量化矩阵是基于是否应用了所述逆二次变换来确定的。

18. 一种存储视频的比特流的方法,包括:

确定逆二次变换是否被应用于视频的当前块,其中来自所述逆二次变换的输出值被约束在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 的第一范围内,其中所述逆二次变换适用于去量化和逆主变换之间的当前块,并且其中 $\min$ 和 $\max$ 是整数值;

基于所述确定生成所述视频的比特流,其中所述去量化之后的系数被约束到 $[\text{qmin}, \text{qmax}]$ 的第二范围,包括 qmin 、 qmax , qmin 和 qmax 是整数,并且其中所述第一范围和所述第二范围具有如下至少一个关系:(1) $\min$ 等于 qmin ,或者(2) $\max$ 等于 qmax ,其中响应于利用非帧内预测模式的预测模式对所述当前块进行编解码,不对所述当前块应用所述逆二次变换,并且其中所述去量化中使用的量化矩阵是基于是否应用了所述逆二次变换来确定的;以及

将所生成的所述比特流存储到非暂时性计算机可读记录介质中。

19. 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,其中所述指令在由所述处理器执行时,使所述处理器实施根据权利要求1-13中任一项所述的方法。

20. 一种非暂时性计算机可读介质,其上存储有计算机程序,所述计算机程序包括用于

实行根据权利要求1-13中任一项所述的方法的程序代码。

基于二次变换的视频处理中的裁剪操作

[0001] 相关申请的交叉引用

[0002] 根据巴黎公约适用的专利法和/或规则,本申请旨在及时要求2019年4月23日提交的国际专利申请号PCT/CN2019/083853的优先权和利益。出于法律规定的所有目的,上述申请的全部公开内容通过引用并入本文,作为本申请公开内容的一部分。

技术领域

[0003] 本专利文档涉及视频编解码技术、设备和系统。

背景技术

[0004] 尽管在视频压缩方面取得了进步,数字视频仍然占互联网和其他数字通信网络上的最大带宽使用。随着能够接收和显示视频的连接用户设备数量的增加,预计对数字视频使用的带宽需求将继续增长。

发明内容

[0005] 本文件描述了各种实施例和技术,其中在视频或图像的解码或编码期间使用二次变换(也称为低频不可分变换)。

[0006] 在一个示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频的比特流表示之间的转换,确定来自具有缩减的大小的逆二次变换的输出值被约束在 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 在内的范围内。逆二次变换适用于反量化步骤和逆主变换之间的块。缩减的大小是从块的大小缩减的,并且 $\min$ 和 $\max$ 是整数值。该方法还包括基于该确定来执行转换。

[0007] 在另一示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频的比特流表示之间的转换,基于二次变换适用的子块的数量,确定将具有缩减的大小的二次变换应用于该块的子块的方式。二次变换适用于正主变换与量化步骤之间或者反量化步骤与逆主变换之间的块。缩减的大小是从块的大小缩减的。该方法还包括基于该确定来执行转换。

[0008] 在另一示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频的比特流表示之间的转换,在块的大小满足条件的情况下,确定具有缩减的大小的二次变换适用于该块的单个子块。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。该方法还包括基于该确定来执行转换。

[0009] 在另一示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频的比特流表示之间的转换,确定具有缩减的大小的二次变换适用于具有 $K \times L$ 的大小的块中的区域。 K 和 L 是正整数,并且 K 不等于 L 。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。该方法还包括基于该确定来执行转换。

[0010] 在另一示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频

的比特流表示之间的转换,基于块的特性确定非零范围。非零范围对应于一范围,在该范围之外,与具有缩减的大小的二次变换相关联的系数被设置为零。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。该方法还包括基于该确定来执行转换。

[0011] 在另一示例方面,公开了一种视频编码的方法。该方法包括确定具有缩减的大小的二次变换适用于视频的块的两个相邻子块。两个相邻子块中的每一个具有 $M \times N$ 的大小, M 和 N 是正整数。二次变换在正主变换和量化步骤之间执行。缩减的大小是从块的大小缩减的。该方法还包括基于该确定生成视频的编解码表示。

[0012] 在另一示例方面,公开了一种视频解码的方法。该方法包括确定具有缩减的大小的二次变换适用于视频的块的两个相邻子块。两个相邻子块中的每一个具有 $M \times N$ 的大小, M 和 N 是正整数。二次变换在反量化步骤和逆主变换之间执行。缩减的大小是从块的大小缩减的。该方法还包括根据该确定通过解析视频的编解码表示来生成视频的块。

[0013] 在另一示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频的比特流表示之间的转换,根据规则,基于与该块相关联的特性,确定是否将具有缩减的大小的二次变换应用于该块。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。该方法还包括基于该确定来执行转换。

[0014] 在另一示例方面,公开了一种视频处理的方法。该方法包括:针对视频的块和视频的比特流表示之间的转换,确定适用于该块的具有缩减的大小的二次变换的一个或多个变换矩阵的系数的比特精度约束。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。该方法还包括基于该确定来执行转换。

[0015] 在另一示例方面,公开了一种视频处理的方法。该方法包括:确定约束规则,该约束规则用于在当前视频块的比特流表示和当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换;并且通过根据约束规则应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。在转换期间,具有缩减的大小的二次变换以特定的顺序与主变换一起应用。

[0016] 在另一示例方面,公开了视频处理的另一种方法。该方法包括:确定约束规则,该约束规则用于在当前视频块和临近视频区域的比特流表示与当前视频块的像素和临近区域的像素之间的转换期间选择性地应用具有缩减的大小的二次变换;并且通过根据约束规则应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块和临近视频区域的大小缩减的大小。在转换期间,具有缩减的大小的二次变换以特定的顺序与主变换一起应用。

[0017] 在又一示例方面,公开了另一视频处理方法。该方法包括:确定调零规则,该调零规则用于在当前视频块的比特流表示与当前视频块的像素的转换期间选择性地应用具有缩减的大小的二次变换;并且通过根据调零规则应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。调零规则规定了具有缩减的大小的二次变换使用的系数的最大数量。

[0018] 在又一示例方面,公开了另一视频处理方法。该方法包括:确定调零规则,该调零规则用于在当前视频块的比特流表示与当前视频块的像素的转换期间选择性地应用具有缩减的大小的二次变换;并且通过根据调零规则应用具有缩减的大小的二次变换来执行转

换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。调零规则规定了具有缩减的大小的二次变换使用的系数的最大数量。

[0019] 在又一示例方面,公开了另一视频处理方法。该方法包括:确定用于在当前视频块的比特流表示与当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换的条件;并且通过根据该条件应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。在比特流表示中信令通知该条件。

[0020] 在又一示例方面,公开了另一视频处理方法。该方法包括:在当前视频块的比特流表示和当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换,并且通过根据该条件应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。该转换包括基于共存规则选择性地应用位置相关的帧内预测组合(PDPC)。

[0021] 在又一示例方面,公开了另一视频处理方法。该方法包括:在当前视频块的比特流表示和当前视频块的像素之间的转换期间应用具有缩减的大小的二次变换,并且通过根据该条件应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。该应用控制在转换期间临近样点用于帧内预测的使用。

[0022] 在又一示例方面,公开了另一视频处理方法。该方法包括:在当前视频块的比特流表示和当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换,并且通过根据该条件应用具有缩减的大小的二次变换来执行转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。该选择性应用控制在转换期间量化矩阵的使用。

[0023] 在又一示例方面,公开了一种视频编码器。视频编码器包括被配置为实施上述方法中的一个或多个的处理器。

[0024] 在又一示例方面,公开了一种视频解码器。视频解码器包括被配置为实施上述方法中的一个或多个的处理器。

[0025] 在又一示例方面,公开了一种计算机可读介质。该介质包括用于实施存储在介质上的上述方法中的一个或多个的代码。

[0026] 本文件中描述了这些和其他方面。

附图说明

[0027] 图1示出了编码器框图的示例。

[0028] 图2示出了67个帧内预测模式的示例。

[0029] 图3A至图3B示出了用于广角帧内预测的参考样点的示例。

[0030] 图4是在方向超过45度的情况下的不连续问题的示例说明。

[0031] 图5A至图5D示出了由应用于对角和相邻角度帧内模式的PDPC的样点的示例说明。

[0032] 图6是划分 4×8 和 8×4 块的示例。

[0033] 图7是除 4×8 、 8×4 和 4×4 之外的所有块的划分示例。

[0034] 图8将 4×8 样点块划分成两个独立的可解码区域。

[0035] 图9示出了用于最大化具有竖直预测器的 $4 \times N$ 块的吞吐量的像素行的处理的示例顺序。

- [0036] 图10示出了二次变换的示例。
- [0037] 图11示出了所提出的缩减的二次变换 (Reduced Secondary Transform, RST) 的示例。
- [0038] 图12示出了正和反 (或逆) 缩减的变换的示例。
- [0039] 图13示出了利用16x48矩阵的正RST8x8过程的示例。
- [0040] 图14示出了扫描非零元素的位置17到64的示例。
- [0041] 图15是子块变换模式SBT-V和SBT-H的图示。
- [0042] 图16是用于实施本文档中描述的技术的示例硬件平台的框图。
- [0043] 图17是视频处理的示例方法的流程图。
- [0044] 图18是其中可以实施所公开的技术的示例视频处理系统的框图。
- [0045] 图19是根据本技术的视频处理的示例方法的流程图。
- [0046] 图20是根据本技术的视频处理的另一示例方法的流程图。
- [0047] 图21是根据本技术的视频处理的另一示例方法的流程图。
- [0048] 图22是根据本技术的视频处理的另一示例方法的流程图。
- [0049] 图23是根据本技术的视频处理的另一示例方法的流程图。
- [0050] 图24A是根据本技术的视频编码的示例方法的流程图。
- [0051] 图24B是根据本技术的视频解码的示例方法的流程图。
- [0052] 图25是根据本技术的视频处理的另一示例方法的流程图。
- [0053] 图26是根据本技术的视频处理的又一示例方法的流程图。

具体实施方式

[0054] 在本文件中使用章节标题是为了便于理解,并且不将章节中公开的实施例仅限于该章节。此外,尽管参考多功能视频编解码或其他特定视频编解码器描述了某些实施例,但是所公开的技术也适用于其他视频编解码技术。另外,虽然一些实施例详细描述了视频编解码步骤,但是应当理解的是,撤销编解码的相对应的解码步骤将由解码器实施。另外,术语视频处理涵括视频编解码或压缩、视频解码或解压缩以及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进行表示的视频转码。

[0055] 1. 概述

[0056] 本专利文档涉及视频编解码技术。具体地,它涉及视频编解码中的变换。它可以应用于现有的视频编解码标准,如HEVC,或即将要定案的标准 (Versatile Video Coding, 多功能视频编解码)。它也可以适用于未来的视频编解码标准或视频编解码器。

[0057] 2. 初步讨论

[0058] 视频编解码标准主要是通过熟知的ITU-T和ISO/IEC标准的发展而演变的。ITU-T制定了H.261和H.263, ISO/IEC制定了MPEG-1和MPEG-4可视化,并且这两个组织联合制定了H.262/MPEG-2视频和H.264/MPEG-4高级视频编解码 (Advanced Video Coding, AVC) 和H.265/HEVC[1]标准。自H.262以来,视频编解码标准基于混合视频编解码结构,其中利用时域预测加变换编解码。为了探索HEVC以外的未来视频编解码技术,VCEG和MPEG于2015年联合成立了联合视频探索组 (Joint Video Exploration Team, JVET)。此后, JVET采用了许多新方法,并将其放入名为联合探索模型 (Joint Exploration Model, JEM) 的参考软件中

[2]。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)成立,以致力于VVC标准,目标是与HEVC相比比特率降低50%。

[0059] 2.1色彩空间和色度子采样

[0060] 颜色空间,也称为颜色模型(或颜色系统),是这样的抽象数学模型,该抽象数学模型简单地将颜色范围描述为数字的元组,通常为3或4个值或颜色分量(例如,RGB)。基本上来说,色彩空间是坐标系和子空间的细化。

[0061] 对于视频压缩,最常用的颜色空间是YCbCr和RGB。

[0062] YCbCr、Y'CbCr或Y Pb/Cb Pr/Cr(也称为YBCCR或Y'CBCR)是在视频和数字摄影系统中用作彩色图像管线的一部分的色彩空间家族。Y'是亮度分量,而Cb和Cr是蓝差和红差色度分量。Y'(带有单撇号)不同于Y,Y是照度,这意味着光强是基于伽马校正的RGB原色进行非线性编码的。

[0063] 色度子采样是通过利用人类视觉系统对颜色差异的敏感度低于对亮度的敏感度来对色度信息实施比对亮度信息更低的分辨率来对图像进行编码的实践。

[0064] 2.1.1格式4:4:4

[0065] 三个Y'CbCr分量中的每一个具有相同的采样速率,因此没有色度子采样。这种方案有时用于高端电影扫描仪(high-end film scanner)和影片后期制作。

[0066] 2.1.2格式4:2:2

[0067] 两个色度分量以亮度的采样速率的一半进行采样:水平色度分辨率减半。这将未压缩视频信号的带宽缩减了三分之一,而有很少或没有视觉差异

[0068] 2.1.3格式4:2:0

[0069] 在4:2:0中,与4:1:1相比,水平采样加倍,但在该方案中,Cb和Cr通道仅在每条交替线上进行采样,因此垂直分辨率减半。因此,数据速率是相同的。Cb和Cr各自水平地和竖直地以2的因子进行子采样。存在具有不同的水平和竖直定位的4:2:0方案的三种变形。

[0070] 在MPEG-2中,Cb和Cr是水平共存。Cb和Cr在竖直方向上位于像素之间(填隙地位于)。

[0071] 在JPEG/JFIF、H.261和MPEG-1中,Cb和Cr填隙地位于交替的亮度样点中间。

[0072] 在4:2:0DV中,Cb和Cr在水平方向上共存。在竖直方向上,它们在交替线上在一处。

[0073] 2.2典型视频编解码器的编解码流程

[0074] 图1示出了VVC的编码器框图的示例,其中包含三个环路滤波模块:去方块滤波器(deblocking filter,DF)、样点自适应偏移(sample adaptive offset,SAO)和ALF。不同于使用预定义滤波器的DF,SAO和ALF利用当前图片的原始样点来分别通过添加偏移和通过应用有限脉冲响应(FIR)滤波器来缩减原始样点和重构样点之间的均方误差,并且编解码的侧信息信令通知偏移和滤波器系数。ALF位于每张图片的最后处理阶段,并且可以被看作是试图捕捉和修复前面阶段产生的伪像的工具。

[0075] 2.3具有67个帧内预测模式的帧内模式编解码

[0076] 为了捕获自然视频中呈现的任意边缘方向,方向性帧内模式的数量从HEVC中使用的33个扩展到65个。在图2中,附加的方向性模式被描绘为虚线箭头,并且平面和DC模式保持不变。这些更密集的方向性帧内预测模式应用于所有块尺寸以及应用于亮度和色度帧内预测。

[0077] 如图2所示,常规角度帧内预测方向被定义为在顺时针方向为从45度到-135度。在VTM2中,对于非正方形块,用广角帧内预测模式适应性地替换几种常规角度帧内预测模式。被替换的模式使用原始方法进行信令通知,并在解析后重新映射到广角模式的索引。帧内预测模式的总数不变,例如67个,并且帧内模式编解码不变。

[0078] 在HEVC,每个帧内编解码块具有正方形形状,并且其侧中的每一个的长度是2的幂。因此,使用DC模式生成帧内预测器不需要划分操作。在VVV2中,块可以具有矩形形状,这在一般情况下需要对每个块使用划分操作。为了避免DC预测的划分操作,只使用较长的边来计算非正方形块的平均值。

[0079] 2.4非正方形块的广角帧内预测

[0080] 常规角度帧内预测方向被定义为在顺时针方向为从45度到-135度。在VTM2中,对于非正方形块,用广角帧内预测模式适应性地替换几种常规角度帧内预测模式。被替换的模式使用原始方法进行信令通知,并在解析后重新映射到广角模式的索引。特定块的帧内预测模式的总数不变,例如67个,并且帧内模式编解码不变。

[0081] 为了支持这些预测方向,长度为 $2W+1$ 的顶部参考和长度为 $2H+1$ 的左侧参考如图3A至图3B所示定义。

[0082] 广角方向模式中被替换模式的模式号取决于块的纵横比。替换的帧内预测模式在表1中示出。

[0083] 表1被广角模式替换的帧内预测模式

条件	替换的帧内预测模式
$W/H = 2$	模式 2、3、4、5、6、7
$W/H > 2$	模式 2、3、4、5、6、7、8、9、10、11
[0084] $W/H = 1$	无
$H/W = 1/2$	模式 61、62、63、64、65、66
$H/W < 1/2$	模式 57、58、59、60、61、62、63、64、65、66

[0085] 如图4所示,在广角帧内预测的情况下,两个竖直相邻的预测样点可以使用两个不相邻的参考样点。因此,低通参考样点滤波器和侧平滑化被应用于广角预测,以缩减增加的间隙 Δp_q 的负面影响。

[0086] 2.5位置相关的帧内预测组合

[0087] 在VTM2中,通过位置相关的帧内预测组合(position dependent intra prediction combination, PDPC)方法进一步修改平面模式的帧内预测的结果。PDPC是这样的帧内预测方法,其调用未滤波的边界参考样点和HEVC风格帧内预测与经滤波的边界参考样点的组合。PDPC应用于以下没有信令通知的帧内模式:平面、DC、水平、竖直、左下角度模式及其八个相邻角度模式,以及右上角度模式及其八个相邻角度模式。

[0088] 预测样点 $\text{pred}(x, y)$ 是使用帧内预测模式(DC、平面、角度)和参考样点的线性组合根据以下等式进行预测的:

[0089] $\text{pred}(x, y) = (wL \times R_{-1, y} + wT \times R_{x, -1} - wTL \times R_{-1, -1} + (64 - wL - wT + wTL) \times \text{pred}(x, y) + 32) \gg 6$

[0090] 其中, $R_{x, -1}$ 、 $R_{-1, y}$ 分别表示位于当前样点 (x, y) 的顶部和左侧处的参考样点, 而 $R_{-1, -1}$ 表示位于当前块的左上角处的参考样点。

[0091] 如果PDPC应用于DC、平面、水平和垂直帧内模式, 则不需要附加的边界滤波器, 如HEVC DC模式边界滤波器或水平/垂直模式边缘滤波器的情况下所要求的那样。

[0092] 图5A至图5D示出了应用于各种预测模式上的PDPC的参考样点 ($R_{x, -1}$ 、 $R_{-1, y}$ 和 $R_{-1, -1}$) 的定义。预测样点 $\text{pred}(x', y')$ 位于预测块内的 (x', y') 处。参考样点 $R_{x, -1}$ 的坐标 x 由下式给出: $x = x' + y' + 1$, 并且参考样点 $R_{-1, y}$ 的坐标 y 类似地由下式给出: $y = x' + y' + 1$ 。

[0093] 图5A至图5D提供了由应用于对角和相邻角帧内模式的PDPC使用的样点的定义。

[0094] PDPC权重取决于预测模式, 并且在表2中示出。

[0095] 表2根据预测模式的PDPC权重的示例

预测模式	wT	wL	wTL
[0096] 对角右上方	$16 \gg ((y' \ll 1) \gg \text{移位})$	$16 \gg ((x' \ll 1) \gg \text{移位})$	0
	对角左下方	$16 \gg ((y' \ll 1) \gg \text{移位})$	$16 \gg ((x' \ll 1) \gg \text{移位})$
[0097]	相邻对角右上方	$32 \gg ((y' \ll 1) \gg \text{移位})$	0
	相邻对角左下方	0	$32 \gg ((x' \ll 1) \gg \text{移位})$

[0098] 2.6帧内子块分割(Intra subblock partitioning, ISP)

[0099] 在一些实施例中, 建议ISP根据块尺寸大小将亮度帧内预测块竖直或水平划分为2或4个子分区, 如表3所示。图6和图7示出了两种可能性的示例。所有子分区都满足至少有16个样点的条件。

[0100] 表3根据块尺寸的子分区的数量

块尺寸	子分区数量
4×4	未划分
4×8 以及 8×4	2
所有其他情况	4

[0102] 图6示出了 4×8 和 8×4 个块的划分的示例。

[0103] 图7示出了除 4×8 、 8×4 和 4×4 之外的所有块的划分的示例。

[0104] 对于这些子分区中的每一个, 通过对由编码器发送的系数进行熵解码, 并且然后对它们进行反量化和逆变换, 来生成残差信号。然后, 对子分区进行帧内预测, 并且最后通过将残差信号添加到预测信号来获得相对应的重构样点。因此, 每个子分区的重构值将可用于生成下一子分区的预测, 这将重复该过程, 依此类推。所有子分区共享相同的帧内模

式。

[0105] 基于帧内模式和所利用的划分,使用两种不同类别的处理顺序,它们被称为正常顺序和反向顺序。在正常顺序中,要处理的第一子分区是包含CU的左上样点并且然后继续向下(水平划分)或向右(竖直划分)的子分区。结果,用于生成子分区预测信号的参考样点仅位于线的左侧和上侧处。另一方面,反向处理顺序从包含CU的左下样点的子分区开始并继续向上,或者从包含CU的右上样点的子分区开始并继续向左。

[0106] 2.7块差分脉冲编码调制编解码(BDPCM)

[0107] 由于水平(竖直)预测器的形状,它们使用左(A)(右(B))像素对当前像素进行预测,处理块的最有吞吐量效率的方式是并行处理一列(行)的所有像素并顺序处理这些列(行)。为了增加吞吐量,我们引入了以下过程:当在该块上选择的预测器是竖直的时,宽度为4的块被划分具有水平边缘的两半,并且当在该块上选择的预测器是水平的时,高度为4的块被划分具有竖直边缘的两半。

[0108] 当块被划分时,来自一个区域的样点不允许使用来自另一区域的像素来计算预测:如果这种情况发生,则预测像素被预测方向上的参考像素代替。对于竖直预测的 4×8 个块中的当前像素X的不同位置,这种情况在图8中示出。

[0109] 图8示出了将具有 4×8 个样点的块划分成两个独立可解码区域的示例。

[0110] 由于这种属性,现在可以在2个周期内处理 4×4 块,在4个周期内处理 4×8 个或 8×4 个块,依此类推,如图9所示。

[0111] 图9示出了用于最大化具有竖直预测器的 $4 \times N$ 个块的吞吐量的像素行的处理顺序的示例。

[0112] 表4总结了根据块尺寸的处理块所需的周期数量。其繁琐地示出,两个大小都大于或等于8的任何块可以以每周期8个像素或更多的方式进行处理。

[0113] 表4尺寸为 $4 \times N$ 、 $N \times 4$ 的块的最坏情况吞吐量

[0114]

块尺寸	4x4	4x8, 8x4	4x16, 16x4	4x32, 32x4
周期	2	4	8	16
像素	16	32	64	128
吞吐量 (像素/周期)	8	8	8	8

[0115] 2.8量化残差域BDPCM

[0116] 在一些实施例中,提出了量化残差域BDPCM(以下称为RBDPCM)。与帧内预测类似,通过在预测方向(水平或竖直预测)上的样点复制,对整个块进行帧内预测。对残差进行量化,并对量化残差与其预测器(水平或竖直)量化值之间的差值进行编解码。

[0117] 对于尺寸为M(行) $\times$ N(列)的块,在使用来自上方或左侧块边界样点的未滤波样点水平(逐行复制预测块上的左临近像素值)或竖直(将顶部临近行复制到预测块中的每一行)执行帧内预测之后,让 $r_{i,j}$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ 成为预测残差。让 $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ 表示残差 $r_{i,j}$ 的量化版本,其中残差是原始块值和预测块值之间的差。然后将块DPCM应用于量化的残差样点,得到修改的具有元素 $\tilde{r}_{i,j}$ 的 $M \times N$ 阵列 $\tilde{R}$ 。当信令通知竖直

BDPCM时：

$$[0118] \quad \tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0, \quad 0 \leq j \leq (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \end{cases}^{\circ}$$

[0119] 对于水平预测,应用类似的规则,并且通过下式获得残差量化样点

$$[0120] \quad \tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M-1), \quad j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), \quad 1 \leq j \leq (N-1) \end{cases}^{\circ}$$

[0121] 将残差量化样点 $\tilde{r}_{i,j}$ 发送到解码器。

[0122] 在解码器侧上,反转上述计算以产生 $Q(r_{i,j})$, $0 \leq i \leq M-1, 0 \leq j \leq N-1$ 。对于竖直预测情况,

$$[0123] \quad Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1)。$$

[0124] 对于水平情况,

$$[0125] \quad Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1)。$$

[0126] 反量化残差 $Q^{-1}(Q(r_{i,j}))$ 被添加到帧内块预测值,以产生重构样点值。

[0127] 这个方案的主要优点是,可以在系数解析期间在运行中仅在解析系数时添加预测器的情况下完成反DPCM,或者可以在解析之后执行。

[0128] 在量化残差域BDPCM中总是使用变换跳过。

[0129] 2.9 VVC中的多变换集

[0130] 在VTM4中,可以进行的较大块尺寸(尺寸方面高达 64×64)的转换,这主要用于更高分辨率的视频,例如1080p和4K序列。对于尺寸(宽度或高度,或者宽度和高度)等于64的变换块,高频变换系数调零,使得仅保留低频系数。例如,对于 $M \times N$ 变换块,其中M为块宽度且N为块高度,当M等于64时,仅保留变换系数的左32列。类似地,当N等于64时,仅保留变换系数的前32行。当变换跳过模式用于较大块时,使用整个块而不调零任何值。

[0131] 除了已经在HEVC中采用的DCT-II之外,多变换选择(Multiple Transform Selection, MTS)方案被用于残差编解码帧间和帧内编解码的块两者。它使用从DCT8/DST7中选择的多个变换。新引入的变换矩阵是DST-VII和DCT-VIII。表5示出了所选择的DST/DCT的基函数。

[0132] 表5 DST/DCT的基函数选择

变换类型	基函数 $T_i(j)$, $i, j = 0, 1, \dots, N-1$
[0133] DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ $\text{其中, } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

[0134] 为了保持变换矩阵的正交性,与HEVC中的变换矩阵相比,变换矩阵被更精确地量化。为了将变换系数的中间值保持在16位范围内,在水平变换之后和竖直变换之后,所有系数将具有10位。

[0135] 为了控制MTS方案,在SPS级分别为帧内和帧间指定单独的使能标志。当在SPS处启用MTS时,将信令通知CU级标志以指示是否应用MTS。在此,MTS仅应用于亮度。当满足以下条件时,信令通知MTS CU级标志。

[0136] -宽度和高度两者小于或等于32。

[0137] -CBF标志等于1。

[0138] 如果MTS CU标志等于0,则在两个方向上应用DCT2。然而,如果MTS CU标志等于1,那么附加地信令通知另外两个标志以分别指示用于水平和竖直方向的变换类型。变换和信令通知映射表在表6中示出。在涉及到变换矩阵精度时,使用8位主变换核。因此,HEVC中使用的所有变换核保持不变,包括4点DCT-2和DST-7、8点、16点和32点DCT-2。而且,包括64点DCT-2、4点DCT-8、8点、16点、32点DST-7和DCT-8的其他变换核使用8位主变换核。

[0139] 表6变换和信令通知映射表

[0140]	MTS_CU_标志	MTS_Hor_标志	MTS_Ver_标志	帧内/帧间	
				水平	竖直
[0141]	0			DCT2	
	1	0	0	DST7	DST7
		0	1	DCT8	DST7
		1	0	DST7	DCT8
		1	1	DCT8	DCT8

[0142] 为了降低较大尺寸的DST-7和DCT-8的复杂性,对于尺寸(宽度或高度,或宽度和高度两者)等于32的DST-7和DCT-8块,高频变换系数调零。仅保留16x16低频区域内的系数。

[0143] 如在HEVC,块的残差可以用变换跳过模式编解码。为了避免语法编解码(syntax coding)的冗余,当CU级的MTS_CU_标志不等于零时,不信令通知变换跳过标志。变换跳过的块尺寸限制与JEM4中的MTS的块尺寸限制相同,这表明当块宽度和高度两者等于或小于32时,变换跳过适用于CU。

[0144] 2.10示例缩减的二次变换(RST)

[0145] 2.10.1示例不可分二次变换(Non-Separable Secondary Transform,NSST)

[0146] 在一些实施例中,二次变换(也称为不可分变换)被应用于正主变换和量化之间(在编码器处),以及反量化和主逆变换之间(在解码器侧)。如图10所示,根据块尺寸执行4x4(或8x8)二次变换。例如,4x4二次变换应用于较小块(例如,min(宽度,高度)<8),并且8x8二次变换应用于每个8x8块的较大块(例如,min(宽度,高度)>4)。

[0147] 图10示出了JEM中的二次变换的示例。

[0148] 如下使用输入作为示例描述不可分变换的应用。为了应用不可分变换,4x4输入块X

$$[0149] \quad X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

[0150] 首先表示为向量 $\vec{X}$:

$$[0151] \quad \vec{X} = [X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T$$

[0152] 不可分变换被计算为 $\vec{F} = T \cdot \vec{X}$,其中 $\vec{F}$ 指示变换系数向量,并且T是16x16变换矩阵。随后,使用该块的扫描顺序(水平、竖直或对角线),将16x1系数向量 $\vec{F}$ 重新组织为4x4块。具有较小索引的系数将与较小的扫描索引一起放置在4x4系数块中。总共有35个变换集,并且每个变换集使用3个不可分的变换矩阵(内核)。从帧内预测模式到变换集的映射是预先定义的。对于每个变换集,所选择的不可分的二次变换候选由明确信令通知的二次变换索引进一步指定。在变换系数之后,在比特流中每个帧内CU信令通知索引一次。

[0153] 2.10.2示例缩减的二次变换(RST)/低频不可分变换(Low Frequency Non-Separable Transform,LFNST)

[0154] 缩减的二次变换(RST),也称为低频不可分变换(LFNST),被引入为4个变换集(而不是35个变换集)映射。在一些实施例中,16x64(可以进一步缩减到16x48)和16x16矩阵分别用于8x8和4x4块。为了便于说明,16x64(可以进一步缩减为16x48)变换表示为RST8x8,并且16x16变换表示为RST4x4。图11示出了RST的示例。

[0155] 图11示出了所提出的缩减的二次变换的示例(RST)。

[0156] RST计算

[0157] 缩减的变换(Reduced Transform,RT)的主要思想是将N维向量映射到不同空间中的R维向量,其中R/N(R<N)是缩减因子。

[0158] RT矩阵是RxN矩阵,如下:

$$[0159] \quad T_{RxN} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \cdots & t_{RN} \end{bmatrix}$$

[0160] 其中变换的R行是N维空间的R个基。RT的逆变换矩阵是其正变换的转置。正和逆RT的示例在12中进行描绘。

[0161] 图12示出了正和反缩减的变换的示例。

[0162] 在一些实施例中,应用缩减因子为4 (1/4尺寸)的RST8x8。因此,使用16x64直接矩阵,而不是传统的8×8不可分变换矩阵尺寸的64x64直接矩阵。换句话说,在解码器侧处使用64×16的逆RST矩阵以在8×8的左上区域生成核心(主)变换系数。正RST8x8使用16×64 (或对于8x8块为8x64) 矩阵,使得它仅在给定的8×8区域内的左上4×4区域中产生非零系数。换句话说,如果应用RST,那么除了左上4×4区域之外的8×8区域将只有零系数。对于RST4x4,应用16x16 (或对于4x4块为8x16) 直接矩阵乘法。

[0163] 当满足以下两个条件时,有条件地应用逆RST:

[0164] a. 块尺寸大于或等于给定阈值 ($W \geq 4$ 且 $H \geq 4$);

[0165] b. 变换跳过模式标志等于零。

[0166] 如果变换系数块的宽度(W)和高度(H)两者大于4,则将RST8x8应用于变换系数块的左上8×8区域。否则,RST4x4应用于变换系数块的左上角 $\min(8, W) \times \min(8, H)$ 区域。

[0167] 如果RST索引等于0,则不应用RST。否则,应用RST,其核与RST索引一起进行选择。稍后将解释RST选择方法和RST索引的编解码。

[0168] 另外,RST应用于帧内和帧间条带两者中的帧内CU,以及应用于亮度和色度两者。如果启用了双树,则单独地信令通知用于亮度和色度的RST索引。对于帧间条带(双树被禁用),信令通知单个RST索引并将其用于亮度和色度两者。

[0169] 在一些实施例中,采用帧内子分区(Intra Sub-Partition, ISP)作为新的帧内预测模式。当选择ISP模式时,RST被禁用,并且不信令通知RST索引,因为即使RST被应用于每个可行的分区块,性能改善也是微不足道的。另外,对ISP预测的残差禁用RST可以降低编码复杂度。

[0170] RST选择

[0171] RST矩阵选自四个变换集,这些变换集中的每一个包括两个变换。应用哪一个变换集根据帧内预测模式如下确定:

[0172] (1) 如果指示了三种CCLM模式之一,则选择变换集0。

[0173] (2) 否则,根据表7执行变化集选择:

[0174] 表7变换集选择表

IntraPredMode	变换集索引
$\text{IntraPredMode} < 0$	1
$0 \leq \text{IntraPredMode} \leq 1$	0
$2 \leq \text{IntraPredMode} \leq 12$	1
$13 \leq \text{IntraPredMode} \leq 23$	2
$24 \leq \text{IntraPredMode} \leq 44$	3

$45 \leq \text{IntraPredMode} \leq 55$	2
$56 \leq \text{IntraPredMode}$	1

[0176] 访问该表的索引(表示为IntraPredMode)的范围为 $[-14, 83]$,这是用于广角帧内预测的变换模式索引。

[0177] 缩减的大小的RST矩阵

[0178] 作为进一步的缩减,在相同的变换集配置的情况下应用 16×48 矩阵而不是 16×64 矩阵,这些矩阵中的每一个从左上 8×8 块(不包括右下 4×4 块)中的三个 4×4 块获取48个输入数据(图13)。

[0179] 图13示出了利用 16×48 矩阵的正RST 8×8 过程的示例。

[0180] RST信令通知

[0181] $R=16$ 的正向RST 8×8 使用 16×64 矩阵,使得它仅在给定的 8×8 区域内的左上 4×4 区域中产生非零系数。换句话说,如果应用RST,那么除了左上 4×4 区域之外的 8×8 区域将只生成零系数。因此,当在除左上 4×4 (其描绘在图14中)之外的 8×8 块区域内检测到任何非零元素时,RST索引不会被编解码,因为这意味着没有应用RST。在这种情况下,RST索引被推断为零

[0182] 图14示出了扫描非零元素的位置17到64的示例。

[0183] 调零范围

[0184] 通常,在对 4×4 子块应用逆RST之前, 4×4 子块中的任何系数都可以是非零的。然而,受约束的是,在某些情况下,在对子块应用逆RST之前, 4×4 子块中的一些系数必须为零。

[0185] 让nonZeroSize成为变量。要求索引不小于nonZeroSize的任何系数在逆RST之前其被重新排列成1-D阵列时必须为零。

[0186] 当nonZeroSize等于16时,对于左上 4×4 子块中的系数没有调零约束。

[0187] 在一些实施例中,当当前块尺寸是 4×4 或 8×8 时,nonZeroSize被设置为等于8。对于其他块大小,nonZeroSize设置为16。

[0188] RST的示例描述

[0189] 在下面的表格和描述中,粗体斜体文本用于表示可以对当前语法进行以适应本文中描述的某些实施例的改变。

[0190] 序列参数集RBSP语法

[0191]	<code>seq_parameter_set_rbsp() {</code>	描述符
	<code>.....</code>	
	<i>sps_mts_enabled_flag</i>	<i>u(1)</i>
	<code>if(<i>sps_mts_enabled_flag</i>) {</code>	

[0192]

sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
.....	
sps_st_enabled_flag	u(1)
.....	
}	

[0193] 残差编解码语法

[0194]

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	描述符
...	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1--	
if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
if(!transform_skip_flag[x0][y0]) {	
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3 && log2TbHeight == 3) && n >= 8 && i == 0) ((log2TbWidth => 3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
abs_level_gt1_flag[n]	ae(v)
...	

[0195] 编解码单元语法

[0196]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	

[0197]

if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag	
&&	
!ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ	
allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ	
allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ)	
(!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
numZeroOutSigCoeff = 0	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
if(Min(cbWidth, cbHeight) >= 4 && sps_st_enabled_flag == 1 &&	
CuPredMode[x0][y0] == MODE_INTRA	
&& IntraSubPartitionsSplitType == ISP_NO_SPLIT) {	
if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) &&	
numZeroOutSigCoeff == 0) {	
st_idx[x0][y0]	ae(v)
}	
}	
}	
}	
}	

[0198] 序列参数集RBSP语义

[0199]

[0200] 等于 1 的 sps_st_enabled_flag 指定 st_idx 可能存在于帧内编解码单元的残差编解码语法中。等于 0 的 sps_st_enabled_flag 指定 st_idx 不存在于帧内

[0201] 编解码单元的残差编解码语法中。

[0202]

[0203] 编解码单元语义

[0204]

[0205] ***$st_idx[x0][y0]$ 指定在所选择的变换集中的两个候选内核之间应用哪个二次变换内核。等于0的 $st_idx[x0][y0]$ 指定不应用二次变换。阵列索引 $x0$ 、 $y0$ 指定所考虑的变换块的左上样点相对于图片的左上角样点的位置($x0, y0$)。***

当 $st_idx[x0][y0]$ 不存在时， $st_idx[x0][y0]$ 被推断为等于0。

[0206] 缩放变换系数的变换过程

[0207] 综述

[0208] 这一过程的输入是：

[0209] -亮度位置($xTbY, yTbY$)，其相对于当前图片的左-上亮度样点指定当前亮度变换块的左上样点，

[0210] -变量 $nTbW$ ，其指定当前变换块宽度，

[0211] -变量 $nTbH$ ，其指定当前变换块高度，

[0212] -变量 $cIdx$ ，其指定当前块的颜色分量，

[0213] - $x=0..nTbW-1, y=0..nTbH-1$ 的缩放变换系数的 $(nTbW) \times (nTbH)$ 阵列 $d[x][y]$ 。

[0214] 这个过程的输出是 $x=0..nTbW-1, y=0..nTbH-1$ 的残差样点的 $(nTbW) \times (nTbH)$ 阵列 $r[x][y]$ 。

如果 $st_idx[xTbY][yTbY]$ 不等于0，则以下内容使用：

1. 变量 $nStSize$ 、 $\log2StSize$ 、 $numStX$ 、 $numStY$ 和 $nonZeroSize$ 的推导如下：

- ***如果 $nTbW$ 和 $nTbH$ 两者大于或等于8，则 $\log2StSize$ 设置为3，且 $nStOutSize$ 设置为48。***
- [0215] - ***否则， $\log2StSize$ 设置为2，且 $nStOutSize$ 设置为16。***
- ***$nStSize$ 设置为 $(1 \ll \log2StSize)$ 。***
- ***如果 $nTbH$ 等于4且 $nTbW$ 大于8，则 $numStX$ 设置为等于2。***
- ***否则， $numStX$ 设置为等于1。***
- ***如果 $nTbW$ 等于4且 $nTbH$ 大于8，则 $numStY$ 设置为等于2。***
- ***否则， $numStY$ 设置为等于1。***

- 如果 $nTbW$ 和 $nTbH$ 两者等于 4, 或者 $nTbW$ 和 $nTbH$ 两者等于 8, 则 $nonZeroSize$ 设置为等于 8。
- 否则, $nonZeroSize$ 设置为等于 16。

2. 对于 $xSbIdx = 0..numStX - 1$ 且 $ySbIdx = 0..numStY - 1$, 以下内容适用:

- $x = 0..nonZeroSize - 1$ 的变量阵列 $u[x]$ 推导如下:

$$xC = (xSbIdx \ll \log_2 StSize) + DiagScanOrder[\log_2 StSize][\log_2 StSize][x][0]$$

$$yC = (ySbIdx \ll \log_2 StSize) + DiagScanOrder[\log_2 StSize][\log_2 StSize][x][1]$$

$$u[x] = d[xC][yC]$$

- 通过调用条款 8.7.4.4 中指定的一维变换过程, 将 $x = 0..nonZeroSize - 1$ 的 $u[x]$ 变换为 $x = 0..nStOutSize - 1$ 的变量阵列 $v[x]$, 其中缩放的变换系数 $nonZeroSize$ 的变换输入长度、 $x = 0..nonZeroSize - 1$ 的列表 $u[x]$ 的变换输出长度 $nStOutSize$ 、变换集选择的索引 $stPredModeIntra$ 以及在变换集中的变换选择 $st_idx[xTbY][yTbY]$ 的索引作为输入, 并且输出是 $x = 0..nStOutSize - 1$ 的列表 $v[x]$ 。变量 $stPredModeIntra$ 设置为条款 8.4.4.2.1 中指定的 $predModeIntra$ 。

- $x = 0..nStSize - 1$, $y = 0..nStSize - 1$ 的阵列 $d[(xSbIdx \ll \log_2 StSize) + x][(ySbIdx \ll \log_2 StSize) + y]$ 推导如下:
 - 如果 $stPredModeIntra$ 小于或等于 34, 或等于 $INTRA_LT_CCLM$, $INTRA_T_CCLM$ 或 $INTRA_L_CCLM$, 则以下内容适用:

$$d[(xSbIdx \ll \log_2 StSize) + x][(ySbIdx \ll \log_2 StSize) + y] = (y < 4) ? v[x + (y \ll \log_2 StSize)] : ((x < 4) ? v[32 + x + ((y - 4) \ll 2)] : d[(xSbIdx \ll \log_2 StSize) + x][(ySbIdx \ll \log_2 StSize) + y])$$

- 否则, 以下内容适用:

$$d[(xSbIdx \ll \log_2 StSize) + x][(ySbIdx \ll \log_2 StSize) + y] = (y < 4) ? v[y + (x \ll \log_2 StSize)] : ((x < 4) ? v[32 + (y - 4) + (x \ll 2)] : d[(xSbIdx \ll \log_2 StSize) + x][(ySbIdx \ll \log_2 StSize) + y])$$

[0216]

[0217] 二次变换过程

这一过程的输入是:

- 变量 $nTrS$, 其指定变换输出长度,
- 变量 $nonZeroSize$, 其指定变换输入长度,
- $j = 0..nonZeroSize - 1$ 的变换输入列表 $x[j]$,
- 变量 $stPredModeIntra$, 其指定变换集选择的索引,
- 变量 $stIdx$, 其指定变换选择在集合中的索引。

该过程的输出是 $i = 0..nTrS - 1$ 的变换的样点列表 $y[i]$ 。

[0218] 条款 8.7.4.5 中指定的变换矩阵推导过程涉及变换输出长度 $nTrS$ 、变换集选择索引 $stPredModeIntra$ 、变换集 $stIdx$ 中的变换选择索引 $stIdx$ 作为输入以及变换矩阵 $secTransMatrix$ 作为输出。

$i = 0..nTrS - 1$ 的变换样点列表 $y[i]$ 推导如下:

$$y[i] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, ((\sum_{j=0}^{nonZeroSize-1} secTransMatrix[j][i] * x[j]) + 64) \gg 7) \text{ 其中 } i = 0..nTrS - 1$$

$\text{CoeffMin} = -(1 \ll 15)$ 以及 $\text{CoeffMax} = (1 \ll 15) - 1$;

[0219] 二次变换矩阵推导过程

这一过程的输入是:

- 变量 $nTrS$, 其指定变换输出长度,
- 变量 $stPredModeIntra$, 其指定变换集选择的索引,
- 变量 $stIdx$, 其指定指定变换集中的变换选择的索引。

这个过程的输出是变换矩阵 $secTransMatrix$ 。

变量 $stTrSetIdx$ 推导如下:

$stPredModeIntra$	$stTrSetIdx$
$stPredModeIntra < 0$	1
$0 \leq stPredModeIntra \leq 1$	0
$2 \leq stPredModeIntra \leq 12$	1
$13 \leq stPredModeIntra \leq 23$	2
$24 \leq stPredModeIntra \leq 44$	3
$45 \leq stPredModeIntra \leq 55$	2

[0220]

$56 \leq stPredModeIntra$	1
---------------------------	---

[0221]

变换矩阵 $secTransMatrix$ 基于 $nTrS$ 、 $stTrSetIdx$ 和 $stIdx$ 导出如下:

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 0 且 $stIdx$ 等于 1, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 0 且 $stIdx$ 等于 2, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 1 且 $stIdx$ 等于 1, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 1、 $stIdx$ 等于 2, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 2 且 $stIdx$ 等于 1, 则以下内容适用:

[0222]

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 2 且 $stIdx$ 等于 2, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 3 且 $stIdx$ 等于 1, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 16、 $stTrSetIdx$ 等于 3 且 $stIdx$ 等于 2, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 48、 $stTrSetIdx$ 等于 0 且 $stIdx$ 等于 1, 则以下内容适用:

$secTransMatrix [m][n] = \dots$

- 如果 $nTrS$ 等于 48、 $stTrSetIdx$ 等于 0 且 $stIdx$ 等于 2, 则以下

内容适用:

secTransMatrix [m][n] = ...

- 如果 *nTrS* 等于 48、*stTrSetIdx* 等于 1 且 *stIdx* 等于 1, 则以下内容适用:

secTransMatrix [m][n] = ...

- 如果 *nTrS* 等于 48、*stTrSetIdx* 等于 1 且 *stIdx* 等于 2, 则以下内容适用:

secTransMatrix [m][n] = ...

- 如果 *nTrS* 等于 48、*stTrSetIdx* 等于 2 且 *stIdx* 等于 1, 则以下内容适用:

[0223]

secTransMatrix [m][n] = ...

- 如果 *nTrS* 等于 48、*stTrSetIdx* 等于 2 且 *stIdx* 等于 2, 则以下内容适用:

secTransMatrix [m][n] = ...

- 如果 *nTrS* 等于 48、*stTrSetIdx* 等于 3 且 *stIdx* 等于 1, 则以下内容适用:

secTransMatrix [m][n] = ...

- 如果 *nTrS* 等于 48、*stTrSetIdx* 等于 3 且 *stIdx* 等于 2, 则以下内容适用:

secTransMatrix [m][n] = ...

[0224] 2.11 HEVC中反量化裁剪

[0225] 在HEVC, 缩放的变换系数 d' 被计算为 $d' = \text{Clip3}(\text{coeffMin}, \text{coeffMax}, d)$, 其中 d 是裁剪之前的缩放变换系数。

[0226] 对于亮度分量, $\text{coeffMin} = \text{CoeffMinY}$; $\text{coeffMax} = \text{CoeffMaxY}$ 。对于色度分量, $\text{coeffMin} = \text{CoeffMinC}$; $\text{coeffMax} = \text{CoeffMaxC}$; 其中

[0227] $\text{CoeffMinY} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15))$

[0228] $\text{CoeffMinC} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15))$

[0229] $\text{CoeffMaxY} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15)) - 1$

[0230] $\text{CoeffMaxC} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15)) - 1$

[0231] “extended_precision_processing_flag”是在SPS中信令通知的语法元素。

[0232] 2.12仿射线性加权帧内预测(ALWIP, a.k.a. 基于矩阵的帧内预测, MIP)

[0233] 在一些实施例中,进行两个测试。在测试1中,ALWIP被设计为具有8K字节的内存限制,并且每个样点最多4次乘法。测试2类似于测试1,但在内存需求和模型架构方面进一步缩减了设计。

[0234] *所有块形状的单组矩阵和偏移向量。

[0235] *对于所有块形状模式数量缩减到19。

[0236] *将内存需求缩减到5760个10位值,即7.20千字节。

[0237] *预测样点的线性内插在每个方向的单个步骤中执行,如第一次测试中那样代替迭代内插。

[0238] 2.13子块变换

[0239] 对于cu_cbf等于1的帧间预测的CU,可以信令通知cu_sbt_flag来指示是解码整个残差块还是残差块的子部分。在前一种情况下,将进一步解析帧间MTS信息,以确定CU的变换类型。在后一种情况下,残差块的一部分用推断的自适应变换编解码,并且残差块的另一部分被调零。SBT不应用于组合帧内-帧间模式。

[0240] 在子块变换中,位置相关的变换被应用于SBT-V和SBT-H中的亮度变换块(色度TB总是使用DCT-2)。SBT-H和SBT-V的两个位置与不同的核变换相关联。更具体地说,每个SBT位置的水平和垂直变换规定在图15中。例如,SBT-V位置0的水平和垂直变换分别是DCT-8和DST-7。当残差TU的一侧大于32时,相对应的变换被设置为DCT-2。因此,子块变换联合规定了残差块的TU平铺、cbf以及水平和垂直变换,这对于块的主要残差部分在块的一侧的情况可以被认为是语法快捷方式。

[0241] 图15是子块变换模式SBT-V和SBT-H的图示。

[0242] 3.由实施例解决的问题的示例

[0243] 当前的设计具有以下问题:

[0244] (1) MTS/RST中的裁剪和移位/取整操作可能不是最佳的。

[0245] (2) 应用在两个相邻的 4×4 块上的RST可能是昂贵的。

[0246] (3) 对于不同的颜色分量,RST可以以不同的方式进行。

[0247] (4) RST对于屏幕内容编解码可能不会很好地工作。

[0248] (5) RST和其他编解码工具之间的相互作用尚不清楚。

[0249] (6) 可以更有效地存储RST的变换矩阵。

[0250] (7) 如何在RST上应用量化矩阵尚不清楚。

[0251] 4.示例实施例和技术

[0252] 下面列出的实施例应该被认为是用于解释总体构思的示例。这些实施例不应狭义地解释。此外,这些实施例可以以任何方式组合。

[0253] 在以下描述中,编解码信息可以包括预测模式(例如,帧内/帧间/IBC模式)、运动向量、参考图片、帧间预测方向、帧内预测模式、CIIP(combined intra inter prediction,组合帧内帧间预测)模式、ISP模式、仿射帧内模式、采用的变换核、变换跳过标志等,例如编解码块时所需的信息。

[0254] 在以下讨论中,SatShift(x,n)定义为

[0255]
$$\text{SatShift}(x,n) = \begin{cases} (x + \text{offset0}) \gg n & \text{if } x \geq 0 \\ -((-x + \text{offset1}) \gg n) & \text{if } x < 0 \end{cases}$$

[0256] Shift(x,n) 定义为 $\text{Shift}(x,n) = (x + \text{offset0}) \gg n$ 。

[0257] 在一个示例中, offset0和/或offset1被设置为 $(1 \ll n) \gg 1 \text{ or } (1 \ll (n-1))$ 。在另一示例中, offset0和/或offset1被设置为0。

[0258] 在另一示例中, $\text{offset0} = \text{offset1} = ((1 \ll n) \gg 1) - 1 \text{ or } ((1 \ll (n-1))) - 1$ 。

[0259] Clip3(min,max,x) 定义为

$$[0260] \quad \text{Clip3}(\text{Min}, \text{Max}, x) = \begin{cases} \text{Min} & \text{if } x < \text{Min} \\ \text{Max} & \text{if } x > \text{Max} \\ x & \text{Otherwise} \end{cases}$$

[0261] 1. 在逆RST之后, 输出值应该被裁剪到[MinCoef, MaxCoef]、包括MinCoef、MaxCoef的范围内, 其中MinCoef和/或MaxCoef是两个可能可变的整数值。

[0262] a. 在一个示例中, 假设反量化后的系数被裁剪到, [QMinCoef, QMaxCoef]、包括QMinCoef、QMaxCoef的范围, 则MinCoef可以被设置为等于QMinCoef和/或MaxCoef可以被设置为等于QMaxCoef。

[0263] b. 在一个示例中, MinCoef和/或MaxCoef可以取决于颜色分量。

[0264] i. 在一个示例中, MinCoef和/或MaxCoef可以取决于相对应的颜色分量的比特深度。

[0265] c. 在一个示例中, MinCoef和/或MaxCoef可以取决于块的形状(例如, 正方形或非正方形)和/或块大小。

[0266] d. 在一个示例中, 可以信令通知MinCoef和/或MaxCoef的值或候选值的选择, 诸如以SPS、PPS、条带头/片组头/CTU/CU。

[0267] e. 在一个示例中, 对于亮度分量, MinCoef和/或MaxCoef可以推导为:

[0268] $\text{MinCoef} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15))$

[0269] $\text{MaxCoef} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15)) - 1$

[0270] , 其中BitDepthY是亮度分量的比特深度, 并且可以诸如以SPS信令通知extended_precision_processing_flag。

[0271] f. 在一个示例中, 对于分量, MinCoef和/或MaxCoef可以推导为:

[0272] $\text{MinCoef} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15))$

[0273] $\text{MaxCoef} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15)) - 1$,

[0274] 其中BitDepthC是色度分量的比特深度, 并且可以诸如以SPS信令通知extended_precision_processing_flag。

[0275] g. 在一些实施例中, MinCoef is $-(1 \ll 15)$, 且MaxCoef is $(1 \ll 15) - 1$ 。

[0276] h. 在一个示例中, 一致性比特流应当满足正RST之后的变换系数应当在给定范围内。

[0277] 2. 建议对系数的M×N子块应用正RST和/或逆RST的方式可以取决于应用正RST和/或逆RST的子块的数量, 例如, M=N=4。

- [0278] a. 在一个示例中, 调零范围可以取决于应用RST的子块索引。
- [0279] i. 可替代地, 调零范围可以取决于应用RST的子块的数量。
- [0280] b. 在一个示例中, 当在整个系数块中存在应用正RST和/或逆RST的S个子块时 (其中 $S > 1$, 例如 $S = 2$), 对系数的第一子块和对系数的第二子块应用正RST和/或逆RST的方式可以不同。例如, 第一 $M \times N$ 子块可以是左上 $M \times N$ 子块。
- [0281] i. 在一个示例中, 对于系数的第一 $M \times N$ 子块 (表示为nonZeroSize0) 和系数的第二 $M \times N$ 子块 (表示为nonZeroSize1), 2.10节中描述的nonZeroSize可以不同。
- [0282] 1) 在一个示例中, nonZeroSize0可以大于nonZeroSize1。例如, nonZeroSize0=16而nonZeroSize1=8。
- [0283] ii. 在一个示例中, 当只有一个 $M \times N$ 子块将被应用正RST和/或逆RST时, 或者有多于一个的 $M \times N$ 子块将被应用正RST和/或逆RST时, 2.10节中描述的nonZeroSize可以不同。
- [0284] 1) 在一个示例中, 如果有多于一个的 $M \times N$ 子块将被应用正RST和/或逆RST, 则nonZeroSize可以等于8。
- [0285] 3. 建议如果当前块尺寸为 $4 \times H$ 或 $W \times 4$, 其中 $H > 8$ 且 $W > 8$, 则仅对系数的一个 $M \times N$ 子块 (诸如左上 $M \times N$ 子块) 应用正RST和/或逆RST。例如, $M = N = 4$ 。
- [0286] a. 在一个示例中, 如果 $H > T1$ 和/或 $W > T2$ (例如, $T1 = T2 = 16$), 则仅对系数的一个 $M \times N$ 子块应用正RST和/或逆RST。
- [0287] b. 在一个示例中, 如果 $H < T1$ 和/或 $W < T2$ (例如, $T1 = T2 = 32$), 则仅对系数的一个 $M \times N$ 子块应用正RST和/或逆RST。
- [0288] c. 在一个示例中, 对于所有 $H > 8$ 和/或 $W > 8$, 仅对系数的一个 $M \times N$ 子块应用正RST和/或逆RST。
- [0289] d. 在一个示例中, 如果当前块尺寸是 $M \times H$ 或 $W \times N$, 其中 $H \geq N$ 且 $W \geq M$ (例如 $M = N = 4$), 则仅对一个 $M \times N$ 子块 (诸如左上 $M \times N$ 子块) 应用正RST和/或逆RST。
- [0290] 4. RST可以应用于非正方形区域。假设区域尺寸由 $K \times L$ 表示, 其中 K 不等于 L 。
- [0291] a. 可替代地, 另外, 可以在正RST之后对变换系数应用调零, 使得满足非零系数的最大数量。
- [0292] i. 在一个示例中, 如果变换系数位于左上 $M \times M$ 区域之外 (其中 M 不大于 K 并且 M 不大于 L), 则变换系数可以被设置为0。
- [0293] 5. 建议两个相邻的 $M \times N$ 子块中的系数可以包含在单个正RST和/或逆RST中。例如, $M = N = 4$ 。
- [0294] a. 在一个示例中, 可以在编码器处进行如下的一个或几个操作。操作可以按顺序进行。
- [0295] i. 两个相邻的 $M \times N$ 子块中的系数被重新排列成具有 $2 \times M \times N$ 个元素的1-D向量。
- [0296] ii. 对该1-D向量应用具有 $2 \times M \times N$ 列和 $M \times N$ 行 (或 $M \times N$ 列和 $2 \times M \times N$ 行) 的变换矩阵的正RST。
- [0297] iii. 将具有 $M \times N$ 个元素的变换后的1-D向量重新排列成第一 $M \times N$ 子块 (诸如左上子块)。
- [0298] iv. 第二 $M \times N$ 子块中的所有系数可以设置为零。
- [0299] b. 在一个示例中, 可以在解码器处进行如下的一个或几个操作。操作可以按顺序

进行。

[0300] i. 第一 $M \times N$ 子块 (诸如左上子块) 中的系数被重新排列成具有 $M \times N$ 个元素的1-D向量。

[0301] ii. 对1-D向量应用具有 $M \times N$ 列和 $2 \times M \times N$ 行 (或 $2 \times M \times N$ 列和 $M \times N$ 行) 的变换矩阵的逆RST。

[0302] iii. 将具有 $2 \times M \times N$ 个元素的变换后的1-D向量重新排列成两个相邻的 $M \times N$ 子块。

[0303] c. 在一个示例中, 块可以被划分成 K ($K > 1$) 个子块, 并且可以在子块级别下执行主变换和二次变换两者。

[0304] 6. 调零范围 (例如2.10节中所述的nonZeroSize) 可以取决于颜色分量。

[0305] a. 在一个示例中, 对于相同的块大小, 对于亮度和色度分量来说范围可以不同。

[0306] 7. 调零范围 (例如, 2.10节中描述的nonZeroSize) 可能取决于编解码信息。

[0307] a. 在一个示例中, 它可以取决于编解码模式, 诸如帧内模式或非帧内模式。

[0308] b. 在一个示例中, 它可以取决于编解码模式, 诸如帧内模式或帧间模式或IBC模式。

[0309] c. 在一个示例中, 它可以取决于参考图片/运动信息。

[0310] 8. 建议特定块大小的调零范围 (如2.10节中描述的nonZeroSize) 可以取决于量化参数 (QP)。

[0311] a. 在一个示例中, 假设当QP等于QPA时, nonZeroSize等于nonZeroSizeA, 并且当QP等于QPB时, nonZeroSize等于nonZeroSizeB。如果QPA不比QPB小, 那么nonZeroSizeA不比nonZeroSizeB大。

[0312] b. 不同的变换/逆变换矩阵可以用于不同的nonZeroSize。

[0313] 9. 建议可以信令通知调零范围 (例如, 2.10节中描述的nonZeroSize), 诸如以SPS、PPS、图片头、条带头、片组头、CTU行、CTU、CU或任何视频数据单元。

[0314] a. 可替代地, 可以定义多个范围。并且可以信令通知选择哪个候选nonZeroSize的指示, 诸如以SPS、PPS、图片头、条带头、片组头、CTU行、CTU和CU。

[0315] 10. 是否和/或如何应用RST可以取决于颜色格式、和/或分离平面编解码 (separate plane coding) 的使用、和/或颜色分量。

[0316] a. 在一个示例中, RST可以不应用于色度分量 (诸如Cb和/或Cr)。

[0317] b. 在一个示例中, 如果颜色格式是4:0:0, 则可以不对色度分量应用RST。

[0318] c. 在一个示例中, 如果使用分离平面编解码, 则可以不对色度分量应用RST。

[0319] d. 在一个示例中, 特定块大小的nonZeroSize可以取决于颜色分量。

[0320] i. 在一个示例中, 对于相同的块大小, 色度分量上的nonZeroSize可以小于亮度分量上的nonZeroSize。

[0321] 11. 建议当用单个编解码结构树对亮度和色度分量进行编解码时, 对于亮度和色度分量可以单独地信令通知RST控制信息 (诸如是否应用RST、和/或选择哪组变换矩阵)。

[0322] 12. 是否以及如何应用RST可以取决于当前块和/或临近块的编解码信息 (诸如编解码模式)。

[0323] a. 在一个示例中, RST不能用于一个或多个特定的帧内预测模式。

[0324] i. 例如, RST不能用于LM模式。

- [0325] ii. 例如, RST不能用于LM-T模式。
- [0326] iii. 例如, RST不能用于LM-A模式。
- [0327] iv. 例如, RST不能用于广角帧内预测模式。
- [0328] v. 例如, RST不能用于BDPCM模式或/和DPCM模式或/和RBDPCM模式。
- [0329] vi. 例如, RST不能用于ALWIP模式。
- [0330] vii. 例如, RST不能用于某些特定的角度帧内预测模式 (诸如DC、平面、竖直、水平等)。
- [0331] viii. 例如, 在LM模式或/和LM-T模式或/和LM-A模式中, RST可以用于亮度分量, 但不用于色度分量。
- [0332] ix. 例如, 当应用联合色度残差编解码时, RST可能不用于色度分量。
- [0333] b. 如果不能应用RST, 则可能不会信令通知用于指示当前块中信息相关的RST的语法元素。
- [0334] 13. 建议对未进行帧内编解码的块应用RST。
- [0335] a. 在一个示例中, RST可以应用于帧间编解码的块。
- [0336] b. 在一个示例中, RST可以应用于帧内块复制 (intra block copy, IBC) 编解码的块。
- [0337] c. 在一个示例中, 可以对利用组合帧间帧内预测 (CIIP) 编解码的块应用RST。
- [0338] 14. 建议可以在不同级别处对RST进行控制。
- [0339] a. 例如, 用于指示RST (诸如控制标志) 是否适用的信息可以以PPS、条带头、图片头、片组头、片、CTU行、CTU信令通知。
- [0340] b. RST是否适用可能取决于标准配置文件/级别/层级。
- [0341] 15. 建议是否应用位置相关的帧内预测组合 (PDPC) 可以取决于是否应用RST。
- [0342] a. 在一个示例中, 如果当前块应用了RST, 则可能不应用PDPC。
- [0343] b. 在一个示例中, 如果当前块应用了RST, 则可以应用PDPC。
- [0344] c. 可替代地, 是否适用RST可能取决于是否适用PDPC。
- [0345] i. 在一个示例中, 当应用PDPC时, 不应用RST。
- [0346] ii. 如果不能应用RST, 则可能不会信令通知用于指示当前块中信息相关的RST的语法元素。
- [0347] 16. 建议对用于帧内预测的临近样点是否进行滤波可以取决于是否应用RST。
- [0348] a. 在一个示例中, 如果当前块应用了RST, 则可以不滤波临近样点。
- [0349] b. 在一个示例中, 如果当前块应用了RST, 则可以滤波临近样点。
- [0350] c. 可替代性地, 是否应用RST可以取决于用于帧内预测的临近样点是否被滤波。
- [0351] i. 在一个示例中, 当用于帧内预测的临近样点被滤波时, 不应用RST。
- [0352] ii. 在一个示例中, 当用于帧内预测的临近样点未被滤波时, 不应用RST。
- [0353] iii. 如果不能应用RST, 则可能不会信令通知用于指示当前块中信息相关的RST的语法元素。
- [0354] 17. 建议当用变换跳过对当前块进行编解码时, 可以应用RST。
- [0355] a. 例如, 跳过主变换, 但仍可以应用二次变换。
- [0356] b. 变换跳过模式中使用的二次变换矩阵可以不同于无变换跳过模式中使用的二

次变换矩阵。

[0357] 18. 建议用于RST的变换矩阵可以以小于8的比特宽进行存储。例如,用于RST的变换矩阵可以以比特宽6或4进行存储。

[0358] 19. 建议用于RST的变换矩阵可以以预测的方式进行存储。

[0359] a. 在一个示例中,RST的第一变换矩阵中的第一元素可以通过RST的第一变换矩阵中的第二元素进行预测。

[0360] i. 例如,可以存储两个元素之间的差异。

[0361] ii. 例如,差异可以以小于8的比特宽进行存储,诸如6或4。

[0362] b. 在一个示例中,RST的第一变换矩阵中的第一元素可以通过RST的第二变换矩阵中的第二元素进行预测。

[0363] i. 例如,可以存储两个元素之间的差异。

[0364] ii. 例如,差异可以以小于8的比特宽进行存储,诸如6或4。20. 建议可以从RST的第二变换矩阵中推导RST的第一变换矩阵。

[0365] a. 在一个示例中,可以挑选RST的第二变换矩阵的部分元素来构建RST的第一变换矩阵。

[0366] b. 在一个示例中,RST的第一变换矩阵通过旋转或翻转RST的第二变换矩阵的全部或一部分而推导出。

[0367] c. 在一个示例中,通过对RST的第二变换矩阵进行下采样或上采样来推导RST的第一变换矩阵。

[0368] 21. 建议用于指示当前块中信息相关的RST的语法元素可以在信令通知残差(可以被变换)之前进行信令通知。

[0369] a. 在一个示例中,RST相关的信息的信令通知可以不取决于在解析残差时计数的非零或零系数。

[0370] b. 在一个示例中,当解析残差时,可以不计数非零或零系数。

[0371] c. 在一个示例中,可以不信令通知由RST设置为全零的子块的编解码块标志(coded block flag,cbf)标志并将其推断为0。

[0372] d. 在一个示例中,可以不信令通知由RST设置为0的系数的有效标志并将其推断为0。

[0373] e. 解析残差块的扫描顺序可能取决于是否以及如何应用RST。

[0374] i. 在一个示例中,可以不扫描由RST设置为零的系数。

[0375] f. 用于解析残差块的算术编解码上下文可以取决于是否以及如何应用RST。

[0376] 22. 建议是否以及如何应用量化矩阵可能取决于是否以及如何应用RST。

[0377] a. 在一个示例中,无论是否应用RST,都可以应用不同的量化矩阵。

[0378] b. 可替代地,是否以及如何应用RST可以取决于是否以及如何应用量化矩阵。

[0379] i. 在一个示例中,当对块应用量化矩阵时,可以不应用RST。

[0380] 23. 建议RST可以应用于量化系数/残差。

[0381] a. 在一个示例中,当使用变换跳过时,RST可以应用于残差。

[0382] b. 在一个示例中,RST可以应用于块的量化变换后的系数。

[0383] 24. 建议将RST应用于子块变换块。

[0384] a. 在一个示例中, RST可以应用于由子块变换生成的左上系数。

[0385] 图16是视频处理装置1600的框图。装置1600可以用于实施本文描述的方法中的一种或多种。装置1600可以实现在智能手机、平板电脑、计算机、物联网(Internet of Things, IoT)接收器等中。装置1600可以包括一个或多个处理器1602、一个或多个存储器1604和视频处理硬件1606。处理器1602可以被配置成实施本文档中描述的一种或多种方法。存储器(多个存储器)1604可以用于存储用于实施本文描述的方法和技术的代码。视频处理硬件1606可以用于在硬件电路中实施本文档中描述的一些技术。

[0386] 图17是视频处理的示例方法1700的流程图。方法1700包括确定(1702)约束规则, 该约束规则用于在当前视频块的比特流表示和当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换。方法1700包括通过根据约束规则应用具有缩减的大小的二次变换来执行(1704)转换。具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小。在转换期间, 具有缩减的大小的二次变换以特定的顺序与主变换一起应用。

[0387] 附加实施例和技术如下列示例中所述。

[0388] 1. 一种视频处理方法, 包括: 确定约束规则, 该约束规则用于在当前视频块的比特流表示和当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换; 并且通过根据约束规则应用具有缩减的大小的二次变换来执行转换; 其中具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小, 并且其中在转换期间, 具有缩减的大小的二次变换以特定的顺序与主变换一起应用。

[0389] 2. 根据示例1的方法, 其中转换包括将当前视频块编码成比特流表示, 并且其中特定顺序包括首先在正方向上应用主变换, 随后有选择地在正方向上应用具有缩减的大小的二次变换, 随后在正方向上量化具有缩减的大小的二次变换的输出。

[0390] 3. 根据示例1的方法, 其中转换包括从比特流表示解码当前视频块, 并且其中特定顺序包括首先对比特流表示应用反量化, 随后选择性地逆方向上应用具有缩减的大小的二次变换, 随后在逆方向上对具有缩减的大小的二次变换在逆方向上的输出应用主变换。

[0391] 4. 根据示例1至3中任一项的方法, 其中约束规则规定将具有缩减的大小的二次变换在逆方向上的输出范围裁剪为[MinCoef, MaxCoef]、包括MinCoef、MaxCoef的范围, 其中MinCoef和/或MaxCoef是作为当前视频块的条件函数的两个整数值。

[0392] 5. 根据示例4的方法, 其中当前视频块的条件是由当前视频块表示的颜色或亮度分量的类型。

[0393] 6. 根据示例1的方法, 其中约束规则规定将具有缩减的大小的二次变换应用于当前视频块的一个或多个MxN子块, 并将当前视频块的剩余子块调零。

[0394] 7. 根据示例1的方法, 其中约束规则规定不同地将具有缩减的大小的二次变换应用于当前视频块的不同子块。

[0395] 8. 根据示例1至5中任一项所述的方法, 其中, 由于当前视频块具有 $4 \times H$ 或 $W \times 4$ 的尺寸, 约束规则规定将具有缩减的大小的二次变换应用于当前视频块的恰好一个MxN子块, 其中, H是以整数像素为单位的高度, W是以整数像素为单位的宽度。

[0396] 9. 根据示例8的方法, 其中 $H > 8$ 或 $W > 8$ 。

[0397] 10. 根据示例1至9中任一项的方法, 其中当前视频块是视频的非正方形区域。

[0398] 11. 根据示例2或3的方法, 其中约束规则规定在正方向上将主变换的变换系数调

零,或者在反方向上将零系数填充到二次变换的输出。

[0399] 示例1至5的另外的实施例在第4节的第1项中描述。示例6至7的另外的实施例在第4节的第2项中描述。示例8至9的另外的实施例在第4节的第3项中描述。示例10至11的另外的实施例在第4节的第4项中描述。

[0400] 12.一种视频处理方法,包括:确定约束规则,该约束规则用于在当前视频块和临近视频区域的比特流表示与当前视频块的像素和临近视频区域的像素之间的转换期间选择性地应用具有缩减的大小的二次变换;并且通过根据约束规则应用具有缩减的大小的二次变换来执行转换;其中具有缩减的大小的二次变换具有从当前视频块和临近视频区域的大小缩减的大小,并且其中在转换期间,具有缩减的大小的二次变换以特定的顺序与主变换一起应用。

[0401] 13.根据示例12的方法,其中临近视频区域包括当前视频块的左上块。

[0402] 14.根据实例12的方法,其中当前视频块和临近视频区域对应于父视频块的子块。

[0403] 示例12至14的另外的实施例在第4节的第5项中描述。

[0404] 15.一种视频处理方法,包括:确定调零规则,该调零规则用于在当前视频块的比特流表示与当前视频块的像素的转换期间选择性地应用具有缩减的大小的二次变换;并且通过根据调零规则应用具有缩减的大小的二次变换来执行转换;其中具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小;其中调零规则规定了具有缩减的大小的二次变换使用的系数的最大数量。

[0405] 16.根据示例15的方法,其中系数的最大数量是当前视频块的分量标识的函数。

[0406] 17.根据示例16的方法,其中对于亮度视频块和色度视频块,系数的最大数量是不同的。

[0407] 18.根据示例15至17中任一项的方法,其中调零规则规定调零范围,该调零范围是当前视频块的编解码信息的函数。

[0408] 19.根据示例15至17中任一项的方法,其中调零规则规定调零范围,该调范围是当前视频块的量化参数的函数。

[0409] 20.根据示例15至19中任一项的方法,其中调零范围在比特流表示中由包括在序列参数集级别、或图像参数集级别、或图像头、或条带头、或片组头、或编解码树单元行、或编解码树单元、或编解码单元或在视频数据单元级别的字段来指示。

[0410] 示例15至17的另外的实施例在第4节的第6项中描述。示例18的其他实施例在第4节的第7项中描述。示例19的其他实施例在第4节的第8项中描述。示例20的其他实施例在第4节的第9项中描述。

[0411] 21.一种视频处理方法,包括:确定用于在当前视频块的比特流表示与当前视频块的像素之间的转换期间选择性地应用具有缩减的大小的二次变换的条件;并且通过根据该条件应用具有缩减的大小的二次变换来执行转换;其中具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小;并且其中在比特流表示中信令通知该条件。

[0412] 22.根据示例21的方法,其中条件是颜色格式或分离平面编解码的使用或基于当前视频块的颜色标识。

[0413] 示例21至22的另外的实施例在第4节的第10项中描述。

[0414] 23.根据示例21至22中任一项的方法,其中对于色度分量和亮度分量,分别在比特

流表示中信令通知该条件。

[0415] 示例23的其他实施例在第4节的第11项中描述。

[0416] 24. 根据示例21至23中任一项的方法, 其中该条件取决于当前视频块和临近视频区域的编解码信息。

[0417] 25. 根据示例24的方法, 其中该条件不包括应用使用特定帧内预测模式编解码的当前视频块。

[0418] 示例24至25的另外的实施例在第4节的第12项中描述。

[0419] 26. 根据示例24的方法, 其中该条件规定应用于被帧间编解码的当前视频块。

[0420] 27. 根据示例24的方法, 其中该条件规定应用于使用帧内块复制模式编解码的当前视频块。

[0421] 示例25至26的另外的实施例在第4节的第13项中描述。

[0422] 28. 根据示例21的方法, 其中在比特流表示中以某一级别信令通知该条件, 使得该级别内的所有块符合该条件, 其中该级别是序列参数集级别、或图像参数集级别、或图像头、或条带头、或片组头、或编解码树单元行、或编解码树单元、或编解码单元或视频数据单元级别。

[0423] 示例28的其他实施例在第4节的第14项中描述。

[0424] 29. 根据示例21的方法, 其中该条件是使用变换跳过模式对当前视频块进行编解码。

[0425] 示例29的其他实施例在第4节的第17项中描述。

[0426] 30. 一种视频处理方法, 包括: 在当前视频块的比特流表示和当前视频块的像素之间的转换期间, 选择性地应用具有缩减的大小的二次变换, 并且通过根据条件应用具有缩减的大小的二次变换来执行转换; 其中具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小; 并且其中转换包括基于共存规则选择性地应用位置相关的帧内预测组合(PDPC)。

[0427] 31. 根据示例30的方法, 其中共存规则不包括由于应用了二次变换而将PDPC应用于当前视频块。

[0428] 32. 根据示例30的方法, 其中共存规则规定由于应用二次变换而将PDPC应用于当前视频块。

[0429] 33. 根据示例30的方法, 其中对使用PDPC的当前视频块执行选择性地应用二次变换。

[0430] 示例30至33的另外的实施例在第4节的第15项中描述。

[0431] 34. 一种视频处理方法, 包括: 在当前视频块的比特流表示和当前视频块的像素之间的转换期间应用具有缩减的大小的二次变换, 并且通过根据条件应用具有缩减的大小的二次变换来执行转换; 其中具有缩减的大小的二次变换具有从当前视频块的大小缩减的大小; 并且其中该应用控制在转换期间临近样点用于帧内预测的使用。

[0432] 示例34的其他实施例在第4节的第16项中描述。

[0433] 35. 一种视频处理方法, 包括: 在当前视频块的比特流表示和当前视频块的像素之间的转换期间, 选择性地应用具有缩减的大小的二次变换, 并且通过根据条件应用具有缩减的大小的二次变换来执行转换; 其中具有缩减的大小的二次变换具有从当前视频块的大

小缩减的大小;并且其中该选择性应用控制在转换期间量化矩阵的使用。

[0434] 36. 根据示例35的方法,其中量化矩阵的使用仅由于应用二次变换而发生。

[0435] 示例35至36的另外的实施例在第4节的第22项中描述。

[0436] 37. 根据示例1至36中任一项的方法,其中主变换和二次变换被存储为具有小于8的比特宽的变换矩阵。

[0437] 38. 根据示例1至36中任一项的方法,其中主变换和二次变换被存储为预测变换矩阵。

[0438] 39. 根据示例1至36中任一项的方法,其中主变换可使用第一规则从二次变换导出,或者其中二次变换可使用第二规则从主变换导出。

[0439] 40. 根据示例1至36中任一项的方法,其中比特流表示包括关于当前视频块的残差信息之前的二次变换或主变换的信息。

[0440] 示例37至40的另外的实施例在第4节的第18、19、20和21项中描述

[0441] 41. 一种视频处理装置,包括处理器,该处理器被配置为实施示例1至40中的一个或多个。

[0442] 42. 一种其上存储有代码的计算机可读介质,当由处理器执行时,该代码使得处理器实施示例1至40中的任何一个或多个中阐述的方法。

[0443] 应当理解的是,所公开的技术可以实现在视频编码器或解码器中,以使用包括使用缩减的大小的二次变换的技术来提高压缩效率。

[0444] 图18是示出其中可以实施本文公开的各种技术的示例视频处理系统1800的框图。各种实施方式可以包括系统1800的组件中一些或全部。系统1800可以包括用于接收视频内容的输入1802。视频内容可以以原始或未压缩格式接收,例如8或10位多分量像素值,或者可以呈压缩或编码格式。输入1902可以代表网络接口、外围总线接口或存储接口。网络接口的示例包括有线接口(诸如以太网、无源光网络(passive optical network,PON)等)和无线接口(诸如Wi-Fi或蜂窝接口)。

[0445] 系统1800可以包括编解码组件1804,该编解码组件可以实施本文档中描述的各种编解码或编码方法。编解码组件1804可以降低从编解码组件1804输入1802到输出的视频的平均比特率,以产生视频的编解码表示。因此,编解码技术有时被称为视频压缩或视频转码技术。编解码组件1804的输出可以被存储,或者经由连接的通信传输,如组件1806所示。可以由组件1808使用在输入1802处接收的视频的存储或通信传送的比特流(或经编解码的)表示来生成被发送到显示接口1810的像素值或可显示视频。根据比特流表示生成用户可观看的视频的过程有时被称为视频解压缩。另外,虽然某些视频处理操作被称为“编解码”操作或工具,但是应当理解的是,编解码工具或操作在编码器处使用,并且将由解码器来执行与编解码的结果相反的相对应的解码工具或操作。

[0446] 外围总线接口或显示接口的示例可以包括通用串行总线(universal serial bus,USB)或高清晰度多媒体接口(high definition multimedia interface,HDMI)或显示端口(Displayport)等。存储接口的示例包括SATA(serial advanced technology attachment,串行高级技术附件)、PCI、IDE接口等。本文档中描述的技术可以实现在各种电子设备中,诸如移动电话、膝上型电脑、智能手机或能够执行数字数据处理和/或视频显示的其他设备。

[0447] 图19是根据本技术的视频处理的示例方法1900的流程图。方法1900包括,在操作1910,针对视频的块和视频的比特流表示之间的转换,确定来自具有缩减的大小的逆二次变换的输出值被约束在范围 $[\min, \max]$ 、包括 $\min$ 、 $\max$ 内。逆二次变换适用于反量化步骤和逆主变换之间的块。缩减的大小是从块的大小缩减的, $\min$ 和 $\max$ 是整数值。方法1900包括,在操作1920,基于该确定执行转换。在一些实施例中,具有缩减的大小的逆二次变换包括逆低频不可分变换,其中低频对应于缩减的大小。

[0448] 在一些实施例中,反量化步骤之后的系数被约束为 $[\text{qmin}, \text{qmax}]$,包括 qmin 、 qmax , qmin 和 qmax 是正整数。满足(1) $\min$ 等于 qmin ,或(2) $\max$ 等于 qmax 中的至少一个。在一些实施例中,该范围基于块的颜色分量。在一些实施例中, $\min$ 或 $\max$ 中的至少一个基于颜色分量的比特深度。在一些实施例中,该范围基于块的形状。在一些实施例中,该范围基于该块具有正方形形状还是非正方形形状。在一些实施例中,该范围基于块的大小。在一些实施例中,在比特流表示中信令通知 $\min$ 或 $\max$ 中的至少一个。在一些实施例中,该范围在序列参数集、图片参数集、条带头、片组头、编解码树单元或编解码单元中被信令通知。

[0449] 在一些实施例中,对于块的亮度分量 $\min$ 是 $-(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15))$ 并且 $\max$ 是 $(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15))$ 。 BitDepthY 是亮度分量的比特深度,并且其中 $\text{extended_precision_processing_flag}$ 是在比特流表示中信令通知的变量。

[0450] 在一些实施例中,对于块的色度分量, $\min$ 是 $-(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15))$ 并且 $\max$ 是 $(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15))$ 。 BitDepthC 是亮度分量的比特深度,不强其中 $\text{extended_precision_processing_flag}$ 是在比特流表示中信令通知的变量。

[0451] 在一些实施例中, $\min$ 等于 $-(1 \ll 15)$,并且 $\max$ 等于 $(1 \ll 15) - 1$ 。

[0452] 在一些实施例中,在序列参数集中信令通知 $\text{extended_precision_processing_flag}$ 。在一些实施例中,适用于正主变换和量化步骤之间的二次变换之后的块的系数被限制在范围内。

[0453] 图20是根据本技术的视频处理的示例方法2000的流程图。方法2000包括,在操作2010,对于视频的块和视频的比特流表示之间的转换,基于二次变换应用的子块的数量,确定将具有缩减的大小的二次变换适用于该块的子块的方式。二次变换适用于正主变换与量化步骤之间或者反量化步骤与逆主变换之间的块。缩减的大小是从块的大小缩减的。方法2000还包括,在操作2020,基于该确定执行转换。

[0454] 在一些实施例中,具有缩减的大小的二次变换包括低频不可分变换,其中低频对应于缩减的大小。在一些实施例中,缩减的大小对应于子块的大小。

[0455] 在一些实施例中,子块具有 4×4 的大小。在一些实施例中,子块与子块索引相关联。子块的非零范围之外的系数被设置为零,并且基于子块索引来确定调零范围。在一些实施例中,子块的非零范围之外的系数被设置为零。非零范围基于二次变换适用的子块的数量来确定。

[0456] 在一些实施例中,二次变换适用的子块的数量大于1。二次变换以第一方式应用于第一子块,并且二次变换以不同于第一方式的第二方式应用于第二子块。在一些实施例中,第一子块的第一非零范围之外的系数被设置为零。第二子块的第二非零范围之外的系数被

设置为零,并且第一非零范围不同于第二非零范围。在一些实施例中,第一非零范围大于第二非零范围。在一些实施例中,第一非零范围表示为16,并且第二非零范围表示为8。

[0457] 在一些实施例中,在二次变换仅应用于一个子块的情况下,该唯一的一个子块的第一非零范围之外的系数被设置为零。在二次变换被应用于多个子块的情况下,多个子块的第二非零范围之外的系数被设置为零。在一些实施例中,第一非零范围不同于第二非零范围。在一些实施例中,第二非零范围表示为8。

[0458] 图21是根据本技术的视频处理的另一示例方法的流程图。方法2100包括,在操作2110,为视频的块和视频的比特流表示之间的转换,在块的大小满足条件的情况下,确定具有缩减的大小的二次变换适用于块的单个子块。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。方法2100还包括,在操作2120,基于该确定执行转换。

[0459] 在一些实施例中,缩减的大小对应于子块的大小。在一些实施例中,二次变换适用的单个子块是当前块的左上子块。在一些实施例中,单个子块具有 $M \times N$ 的大小, M 和 N 是正整数。在一些实施例中, $M=N=4$ 。在一些实施例中,条件规定块的大小为 $4 \times H$ 或 $W \times 4$,并且其中 $H>8$ 且 $W>8$ 。在一些实施例中,满足(1) $H>T_1$ 或(2) $W>T_2$ 中的至少一个, T_1 和 T_2 大于8。在一些实施例中, $T_1=T_2=16$ 。在一些实施例中,满足(1) $H<T_1$,或(2) $W<T_2$ 中的至少一个, T_1 和 T_2 大于8。在一些实施例中, $T_1=T_2=32$ 。在一些实施例中,条件规定块的大小是 $M \times H$ 或 $W \times N$,并且其中 $H \geq N$ 且 $W \geq M$ 。

[0460] 图22是根据本技术的视频处理的另一示例方法的流程图。方法2200包括,在操作2210,为视频的块和视频的比特流表示之间的转换,确定具有缩减的大小的二次变换适用于具有 $K \times L$ 的大小的块中的区域。 K 和 L 是正整数,并且 K 不等于 L 。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。方法2200还包括,在操作2220,基于该确定执行转换。

[0461] 在一些实施例中,缩减的大小对应于区域的大小。在一些实施例中,区域的非零范围之外的系数被设置为零。在一些实施例中,非零范围被表示为块中的左上区域,左上区域具有 $M \times M$ 的大小, M 小于或等于 K 和 L 。

[0462] 图23是根据本技术的视频处理的另一示例方法的流程图。方法2300包括,在操作2310,为视频的块和视频的比特流表示之间的转换,基于块的特性确定非零范围。非零范围对应于一范围,在该范围之外,与具有缩减的大小的二次变换相关联的系数被设置为零。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。方法2300还包括,在操作2320,基于该确定执行转换。

[0463] 在一些实施例中,块的特性包括块的颜色分量。在一些实施例中,块的亮度分量的第一非零范围不同于块的色度分量的第二非零范围。在一些实施例中,块的特性包括块的编解码信息。在一些实施例中,编解码信息包括指示块是以帧内模式还是非帧内模式编解码的信息。在一些实施例中,编解码信息包括指示块是否以帧内模式、帧间模式、帧间块复制模式编解码的信息。在一些实施例中,编解码信息包括运动信息的参考图片。在一些实施例中,块的特性包括块的量化参数。在一些实施例中,第一非零范围对应于第一量化参数,并且第二非零范围对应于第二量化参数,并且其中在第一量化参数大于或等于第二量化参数的情况下,第一非零范围小于或等于第二非零范围。

[0464] 在一些实施例中,不同的非零范围与二次变换的不同变换矩阵相关联。在一些实施例中,非零范围在比特流表示中以序列参数集、图片参数集、图片头、条带头、片组头、编解码树单元(coding tree unit,CTU)行、CTU或编解码单元信令通知。在一些实施例中,多个非零范围适用于二次变换,并且指示多个非零范围之一的选择的值在比特流表示中以序列参数集、图片参数集、图片头、条带头、片组头、编解码树单元(CTU)行、CTU或编解码单元信令通知。

[0465] 在一些实施例中,执行转换包括基于视频的块生成比特流表示。在一些实施例中,执行转换包括根据比特流表示生成视频的块。

[0466] 图24A是根据本技术的视频编码的示例方法的流程图。方法2400包括,在操作2410,确定具有缩减的大小的二次变换适用于视频的块的两个相邻子块。两个相邻子块中的每一个具有 $M \times N$ 的大小, M 和 N 是正整数。二次变换在正主变换和量化步骤之间执行。缩减的大小是从块的大小缩减的。方法2400还包括,在操作2420,基于该确定生成视频的编解码表示。

[0467] 在一些实施例中,缩减的大小对应于两个相邻块的大小。在一些实施例中,该方法包括将两个相邻子块的系数排列成具有 $2 \times M \times N$ 个元素的一维向量。在一些实施例中,该方法包括通过使用变换矩阵对一维向量应用二次变换来获得 $M \times N$ 个变换元素。变换矩阵具有 $2 \times M \times N$ 个元素的第一大小和 $M \times N$ 个元素的第二大小。在一些实施例中,该方法包括将 $M \times N$ 个变换后的元素重新排列成两个相邻子块中的第一子块。在一些实施例中,该方法包括将两个相邻子块的第二子块中的元素设置为零。在一些实施例中,正主变换和二次变换两者在子块级别上执行。

[0468] 图24B是根据本技术的视频解码的示例方法的流程图。方法2450包括,在操作2460,确定具有缩减的大小的二次变换适用于视频的块的两个相邻子块。两个相邻子块中的每一个具有 $M \times N$ 的大小, M 和 N 是正整数。二次变换在反量化步骤和逆主变换之间执行。缩减的大小是从块的大小缩减的。方法2450还包括,在操作2470,根据该确定通过解析视频的编解码表示来生成视频的块。

[0469] 在一些实施例中,缩减的大小对应于两个相邻块的大小。在一些实施例中,该方法包括将两个相邻子块的第一子块的系数排列成具有 $M \times N$ 个元素的一维向量。在一些实施例中,该方法包括通过使用变换矩阵对一维向量应用二次变换来获得 $2 \times M \times N$ 个变换元素。变换矩阵具有 $M \times N$ 个元素的第一大小和 $2 \times M \times N$ 个元素的第二大小。在一些实施例中,该方法包括将 $2 \times M \times N$ 个变换后的元素重新排列成两个相邻的子块。在一些实施例中, $M=N=4$ 。

[0470] 在一些实施例中,具有缩减的大小的二次变换包括低频不可分变换,低频对应于缩减的大小。

[0471] 图25是根据本技术的视频处理的另一示例方法的流程图。方法2500包括,在操作2510,为视频的块和视频的比特流表示之间的转换,根据规则,基于与该块相关联的特性,确定是否将具有缩减的大小的二次变换应用于该块。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。方法2500包括,在操作2520,基于该确定执行转换。

[0472] 在一些实施例中,与块相关联的特性包括块的编解码信息或临近块的编解码信息。在一些实施例中,该规则规定在编解码信息指示块或临近块以一种或多种特定编解码

模式编解码的情况下,二次变换不适用于该块。在一些实施例中,一种或多种特定编解码模式包括以下中的至少一种:线性模式(linear mode,LM)模式、LM-T模式、LM-A模式、一种或多种广角帧内预测模式、块差分脉冲编码调制(block differential pulse-code modulation,BDPCM)模式、差分脉冲编码调制(differential pulse-code modulation,DPCM)模式、残差域块差分脉冲编码调制(residual domain block differential pulse-code modulation,RBDPCM)模式、基于矩阵的帧内预测(matrix-based intra prediction,MIP)模式或一种或多种角度帧内预测模式。在一些实施例中,该规则规定在块以联合色度残差编解码模式编解码的情况下,二次变换不适用于块的色度分量。以联合色度残差编解码模式对块进行编解码包括确定联合残差,该联合残差是与块的色度分量相关联的残差的平均值。在一些实施例中,该规则规定二次变换适用于块的亮度分量,而不适用于以LM模式、LM-T模式或LM-A模式编解码的块的色度分量。

[0473] 在一些实施例中,与块相关联的特性包括量化或反量化步骤之后的块的系数或残差。在一些实施例中,该规则规定在使用变换跳过模式对块进行编解码的情况下,将二次变换应用于残差。变换跳过模式是跳过正或逆主变换的模式。在一些实施例中,该规则规定将二次变换应用于块的量化后的变换系数。

[0474] 在一些实施例中,与块相关联的特性包括该块是否使用帧内编解码工具进行编解码。在一些实施例中,该规则规定在使用帧间编解码工具对块进行编解码的情况下,二次变换适用于该块。在一些实施例中,该规则规定在使用帧内块复制编解码工具对块进行编解码的情况下,二次变换适用于该块。在一些实施例中,该规则规定在使用组合帧间帧内预测编解码工具对块进行编解码的情况下,二次变换适用于该块。

[0475] 在一些实施例中,与块相关联的特性包括与块的色度格式相关联的信息。在一些实施例中,该规则规定二次变换不适用于块的色度分量。在一些实施例中,该规则规定在块的色度格式是4:0:0的情况下,二次变换不适用于块的色度分量。在一些实施例中,规则规定在色度格式的色度分量被单独地编解码的情况下,二次变换不适用于块的色度分量。在一些实施例中,该规则规定二次变换适用于该块。基于块的颜色分量来确定与块的大小相关联的二次变换的非零范围,该非零范围是块在其外的系数被设置为零的范围。在一些实施例中,对于相同大小的块,块的色度分量的第一非零范围小于块的亮度分量的第二非零范围。

[0476] 在一些实施例中,基于二次变换是否适用来确定位置相关的帧内预测组合(PDPC)编解码步骤是否适用于该块。在一些实施例中,在二次变换适用于该块的情况下,不应用PDPC编解码步骤。在一些实施例中,在二次变换适用于该块的情况下,应用PDPC编解码步骤。

[0477] 在一些实施例中,与该块相关联的特性包括位置相关的帧内预测组合(PDPC)编解码步骤是否适用于该块。在一些实施例中,该规则规定在PDPC编解码步骤适用的情况下,不将二次变换应用于该块。在一些实施例中,基于二次变换是否适用于块来确定是否为帧内预测编解码步骤滤波块的临近样点。在一些实施例中,在对块应用二次变换的情况下,不滤波临近样点。在一些实施例中,在对块应用二次变换的情况下,对临近样点进行滤波。

[0478] 在一些实施例中,与块相关联的特性包括是否为应用于块的帧内预测编解码步骤滤波块的临近样点。在一些实施例中,该规则规定在滤波临近样点的情况下二次变换不适

用。在一些实施例中,该规则规定在未滤波临近样点的情况下二次变换不适用。

[0479] 在一些实施例中,与该块相关联的特性包括是否用跳过正或逆主变换的变换跳过模式对块进行编解码。在一些实施例中,用变换跳过模式对块进行编解码,并且其中二次变换适用于该块。在一些实施例中,在启用变换跳过模式的情况下用于二次变换的第一变换矩阵不同于在禁用变换跳过模式的情况下用于二次变换的第二变换矩阵。在一些实施例中,基于是否应用了二次变换来确定量化矩阵是否适用于该块。在一些实施例中,在二次变换适用的情况下应用第一量化矩阵,并且其中在二次变换不适用的情况下应用不同的第二量化矩阵。

[0480] 在一些实施例中,与块相关联的特性包括量化矩阵是否适用于该块。在一些实施例中,该规则规定在应用量化矩阵的情况下二次变换不适用。在一些实施例中,与块相关联的特性包括子块级别变换是否适用于该块。在一些实施例中,该规则规定二次变换适用于由子块级别变换生成的块的左上子块的系数。在一些实施例中,在量化或反量化步骤之后解析残差块的扫描顺序是基于二次变换是否应用于该块来确定的。在一些实施例中,不扫描由二次变换设置为零的系数。在一些实施例中,用于在量化或反量化步骤之后解析残差块的算术编解码上下文是基于二次变换是否应用于该块来确定的。

[0481] 在一些实施例中,在比特流表示中在一个或多个级别下信令通知与二次变换相关的信息,该一个或多个级别包括图片参数集、条带、标头、图片头、片组头、片、编解码树单元行或编解码树单元。在一些实施例中,二次变换是否适用是基于信令通知信息的一个或多个级别。在一些实施例中,对于在编解码树单元内编解码的亮度分量和色度分量,单独地信令通知信息。在一些实施例中,在二次变换不适用于块的情况下,与二次变换相关的一个或多个语法元素被排除在块的比特流表示中。在一些实施例中,在比特流表示中的量化变换残差之前,信令通知与二次变换相关的一个或多个语法元素。在一些实施例中,一个或多个语法元素独立于解析量化残差时确定的系数的数量而被信令通知。在一些实施例中,当解析量化残差时,不计数系数的数量。在一些实施例中,指示所有子块被二次变换设置为零的语法标志被排除在比特流表示中,语法标志的值被暗示为0。在一些实施例中,指示系数通过二次变换被设置为零的语法标志被排除在比特流表示中,语法标志的值被暗示为0。

[0482] 图26是根据本技术的视频处理的另一示例方法的流程图。方法2600包括,在操作2610,为视频的块和视频的比特流表示之间的转换,确定适用于该块的具有缩减的大小的二次变换的一个或多个变换矩阵的系数的比特精度约束。二次变换在正主变换与量化步骤之间或者反量化步骤与逆主变换之间执行。缩减的大小是从块的大小缩减的。方法2600还包括,在操作2620,基于该确定执行转换。

[0483] 在一些实施例中,比特精度约束包括一个或多个变换矩阵的系数可用小于8的比特宽来存储。在一些实施例中,比特精度约束包括一个或多个变换矩阵的系数可基于一个或多个变换矩阵之间的关联性来存储。在一些实施例中,存储变换矩阵中第一元素和第二元素之间的差异,其中第一元素基于第二元素推导出。在一些实施例中,存储第一变换矩阵中的第一元素和第二变换矩阵中的第二元素之间的差异,其中第一元素基于第二元素推导出。在一些实施例中,差异由小于8的比特宽表示。在一些实施例中,比特宽是6或4。

[0484] 在一些实施例中,具有缩减的大小的二次变换包括低频不可分变换,低频对应于缩减的大小。

[0485] 在一些实施例中,执行转换包括基于视频的块生成比特流表示。在一些实施例中,执行转换包括根据比特流表示生成视频的块。

[0486] 所公开技术的一些实施例包括做出启用视频处理工具或模式的决定或确定。在示例中,当启用视频处理工具或模式时,编码器将在视频的块的处理中使用或实施该工具或模式,但是不一定基于该工具或模式的使用来修改所得到的比特流。也就是说,当视频处理工具或模式基于决定或确定被启用时,从视频的块到视频的比特流表示的转换将使用视频处理工具或模式。在另一示例中,当启用视频处理工具或模式时,解码器将在知道比特流已经基于视频处理工具或模式被修改的情况下处理比特流。也就是说,将使用基于决定或确定而启用的视频处理工具或模式来执行从视频的比特流表示到视频的块的转换。

[0487] 所公开的技术的一些实施例包括做出禁用视频处理工具或模式的决定或确定。在示例中,当视频处理工具或模式被禁用时,编码器将不会在视频的块到视频的比特流表示的转换中使用该工具或模式。在另一示例中,当视频处理工具或模式被禁用时,解码器将在知道比特流没有使用基于决定或确定而启用的视频处理工具或模式进行修改的情况下处理比特流。

[0488] 在本文件中,术语“视频处理”可以指视频编码、视频解码、视频压缩或视频解压缩。例如,视频压缩算法可以在从视频的像素表示到相对应的比特流表示的转换期间应用,反之亦然。当前视频块的比特流表示可以例如对应于比特流中的共同定位或分布在不同位置的比特,如语法所定义的那样。例如,宏块可以根据变换的和编解码的误差残差值来编码,并且还使用比特流中的标头和其他字段中的比特来编码。

[0489] 本文档中描述的所公开的和其它技术方案、示例、实施例、模块和功能操作可以在数字电子电路中实施,或者在计算机软件、固件或硬件中实施,包括本文档中公开的结构及其结构等同物,或者在它们中的一个或多个的组合中实施。所公开的和其它实施例可以实施为一个或多个计算机程序产品,例如编码在计算机可读介质上的计算机程序指令的一个或多个模块,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、机器可读存储基底、存储器设备、实现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“数据处理装置”涵盖用于处理数据的所有装置、设备和机器,作为示例包括可编程处理器、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运行环境的代码,例如,构成处理器固件、协议栈、数据库管理系统、操作系统或它们中的一个或多个的组的代码。传播信号是人工生成的信号,例如,机器生成的电信号、光信号或电磁信号,其被生成来编码信息以传输到合适的接收器装置。

[0490] 计算机程序(也称为程序、软件、软件应用、脚本或代码)可以以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部署,包括作为独立程序或作为模块、组件、子例程或适合在计算环境中使用的其它单元。计算机程序不一定对应于文件系统中的文件。程序可以存储在保存其它程序或数据的文件的一部分中(例如,存储在标志语言文档中的一个或多个脚本),存储在专用于所讨论的程序的单个文件中,或者存储在多个协调文件(例如,存储一个或多个模块、子程序或部分代码的文件)中。计算机程序可以被部署为位于一个站点或分布在多个站点并通过通信网络互连的一个或多个计算机上执行。

[0491] 本文档中描述的过程和逻辑流程可以由一个或多个可编程处理器运行,该处理器

运行一个或多个计算机程序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,并且装置也可以实施为专用逻辑电路,例如FPGA (field programmable gate array,现场可编程门阵列)或ASIC(application specific integrated circuit,专用集成电路)。

[0492] 举例来说,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,或者被可操作地耦合以从一个或多个大容量存储设备接收数据或向其传送数据或两者兼有。然而,计算机不需要这样的设备。适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,例如,包括半导体存储器设备,例如,EPROM、EEPROM和闪存设备;磁盘,例如,内部硬盘或可移动磁盘;磁光盘;以及CD ROM和DVD-ROM盘。处理器和存储器可以由专用逻辑电路来补充或并入专用逻辑电路。

[0493] 虽然本专利文档包含许多细节,但这些不应被解释为对任何主题或所要求保护的范围的限制,而是对特定于特定技术的特定实施例的特征的描述。本专利文档中在单独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施。相反,在单个实施例的上下文中描述的各种特征也可以在多个实施例中单独或以任何合适的子组合来实施。此外,尽管上述特征可以被描述为以某些组合起作用,甚至最初被要求这样来保护,但是在某些情况下,可以从所要求保护的组合中删除该组合中的一个或多个特征,并且所要求保护的组合可以指向子组合或子组合的变体。

[0494] 类似地,尽管在附图中以特定顺序描述了操作,但这不应理解为要求以所示的特定顺序或顺序地执行这些操作,或者要求执行所有所示的操作,以获得期望的结果。此外,在本专利文档中描述的实施例中的各种系统组件的分离不应该理解为在所有实施例中都需要这种分离。

[0495] 仅描述了几个实施方式和示例,并且可以基于本专利文件中描述和示出的内容进行其他实施、增强和变化。

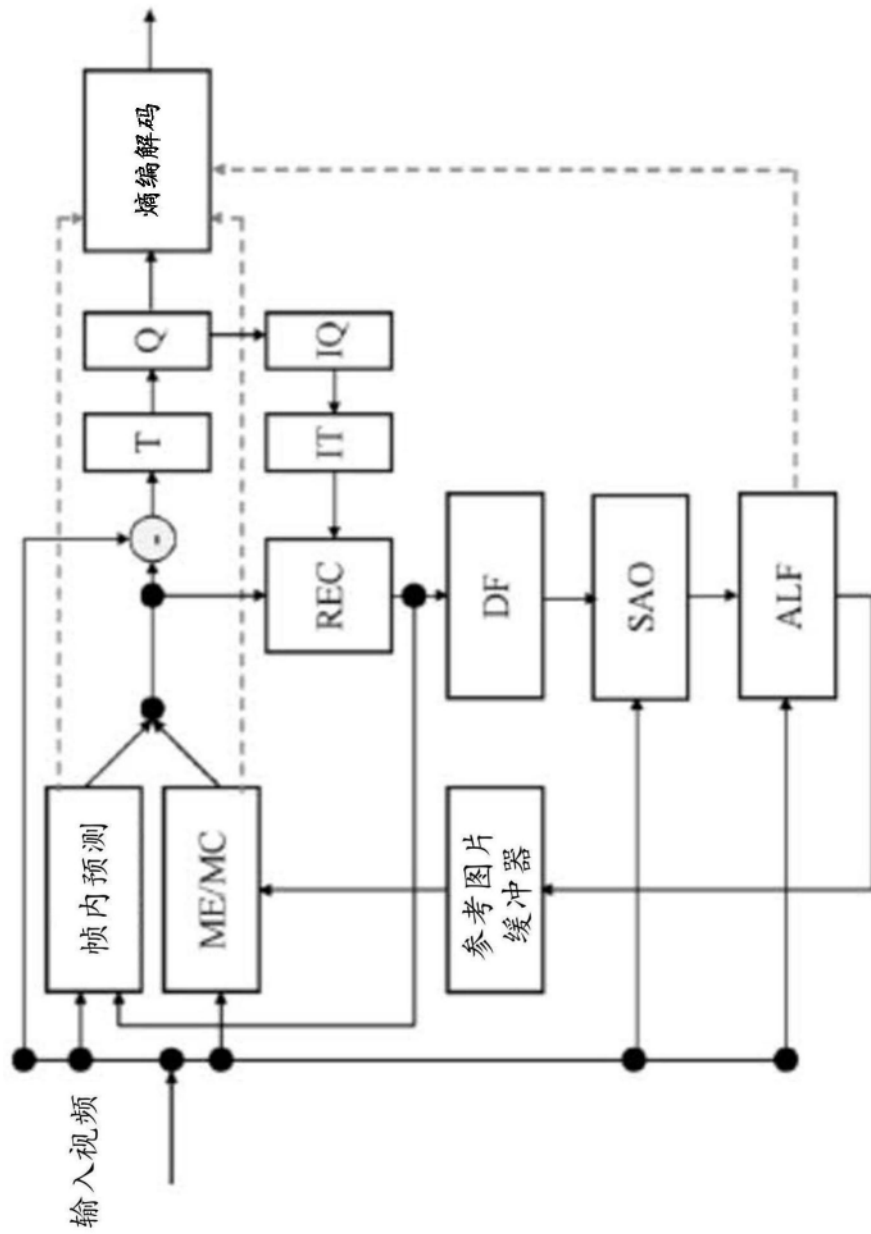


图1

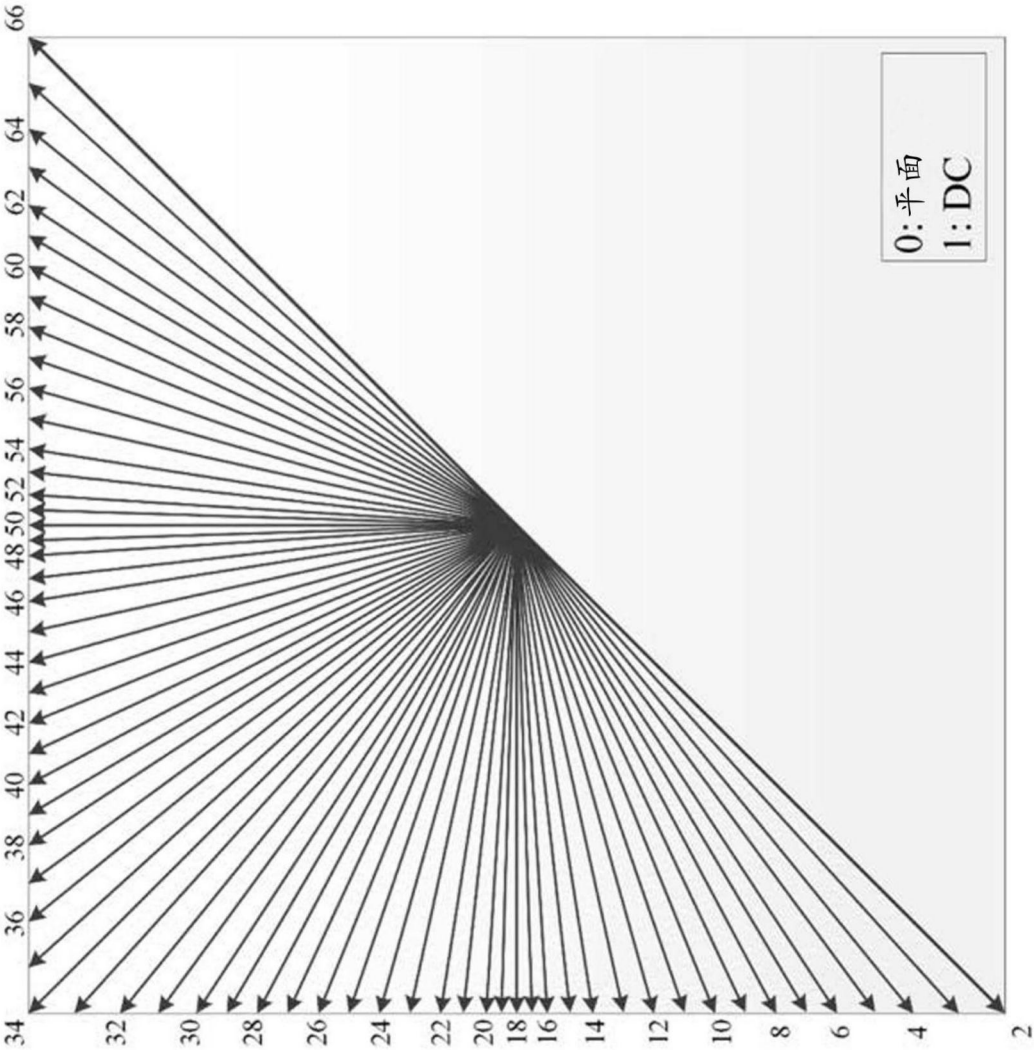


图2

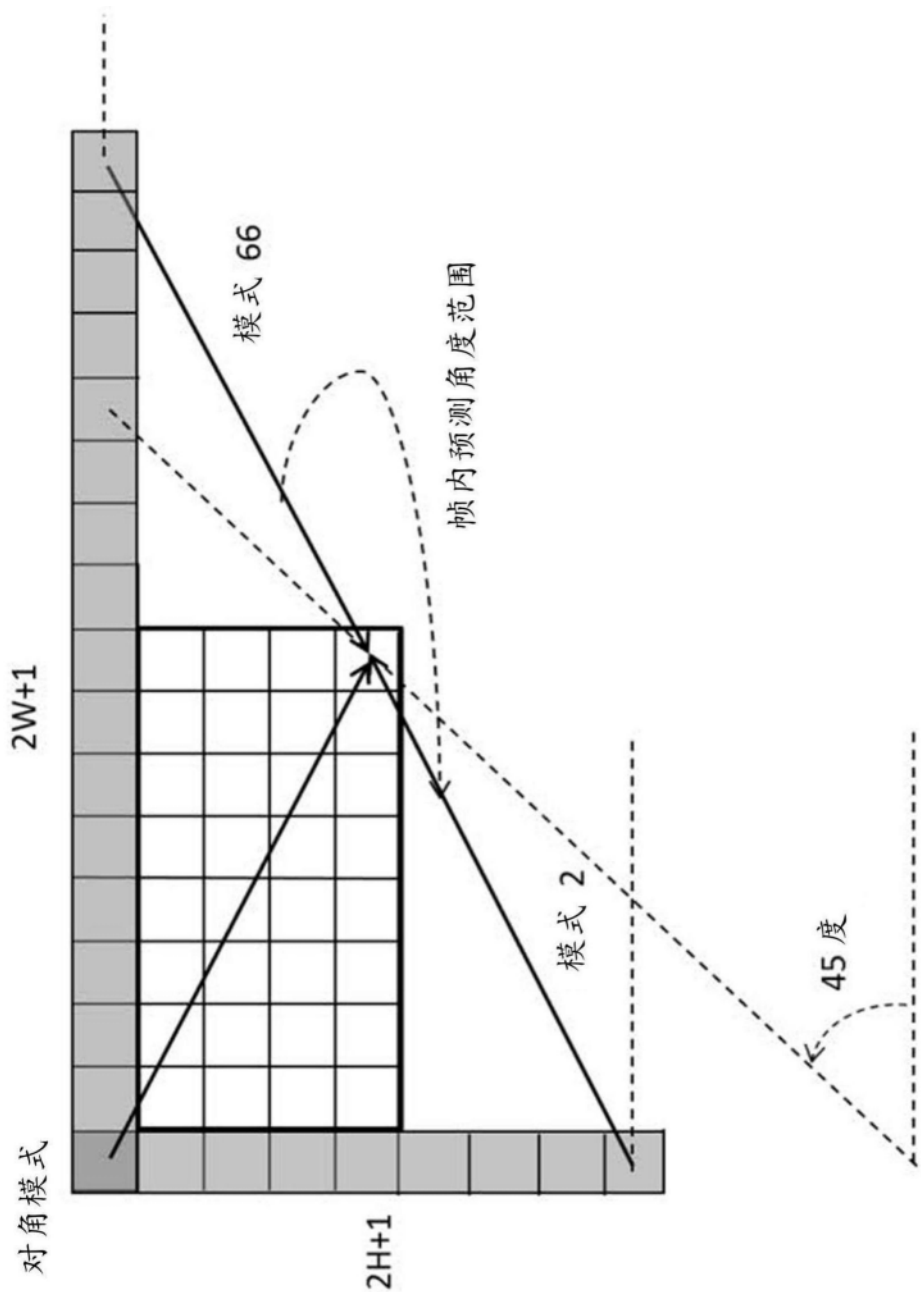


图3A

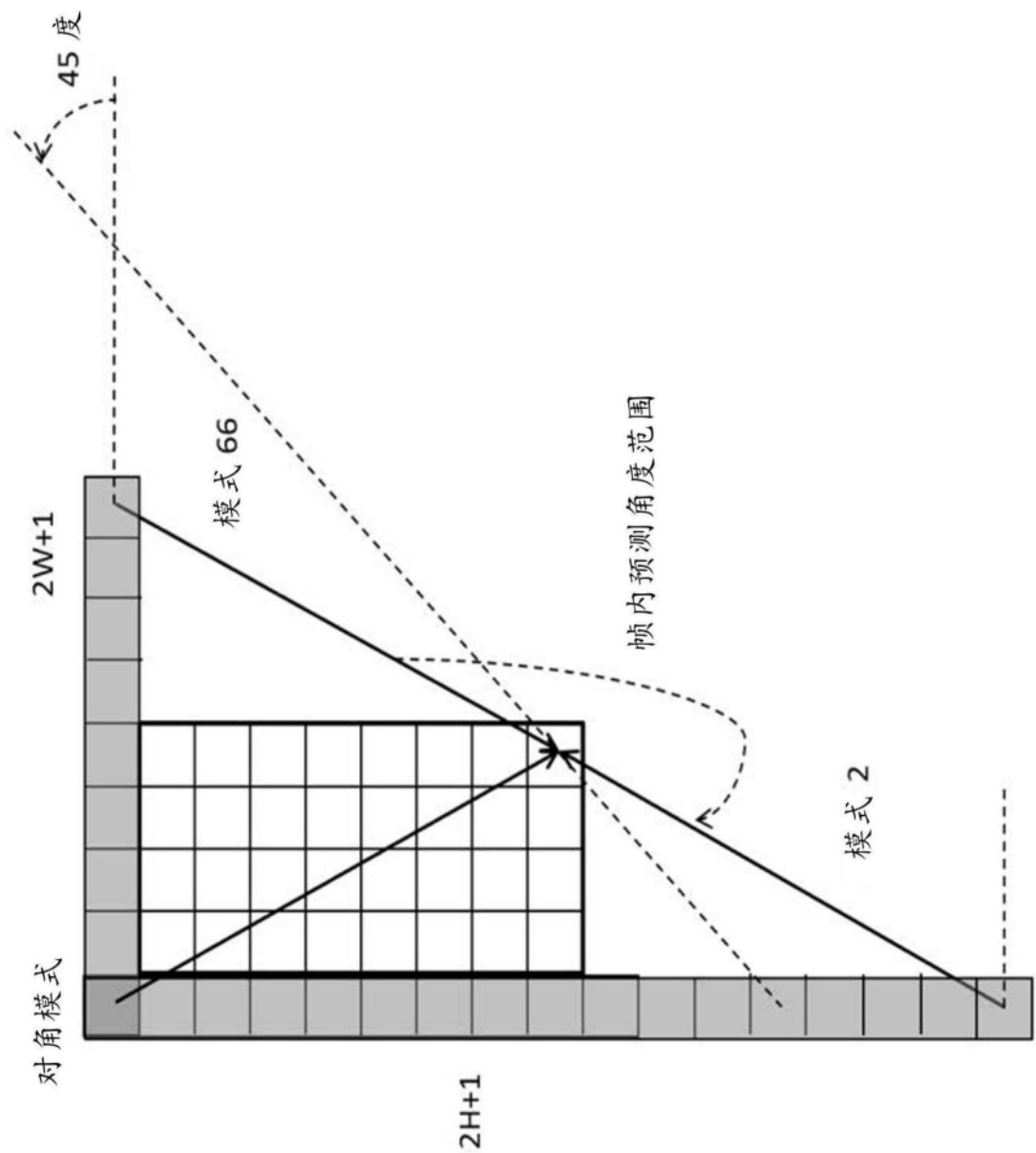


图3B

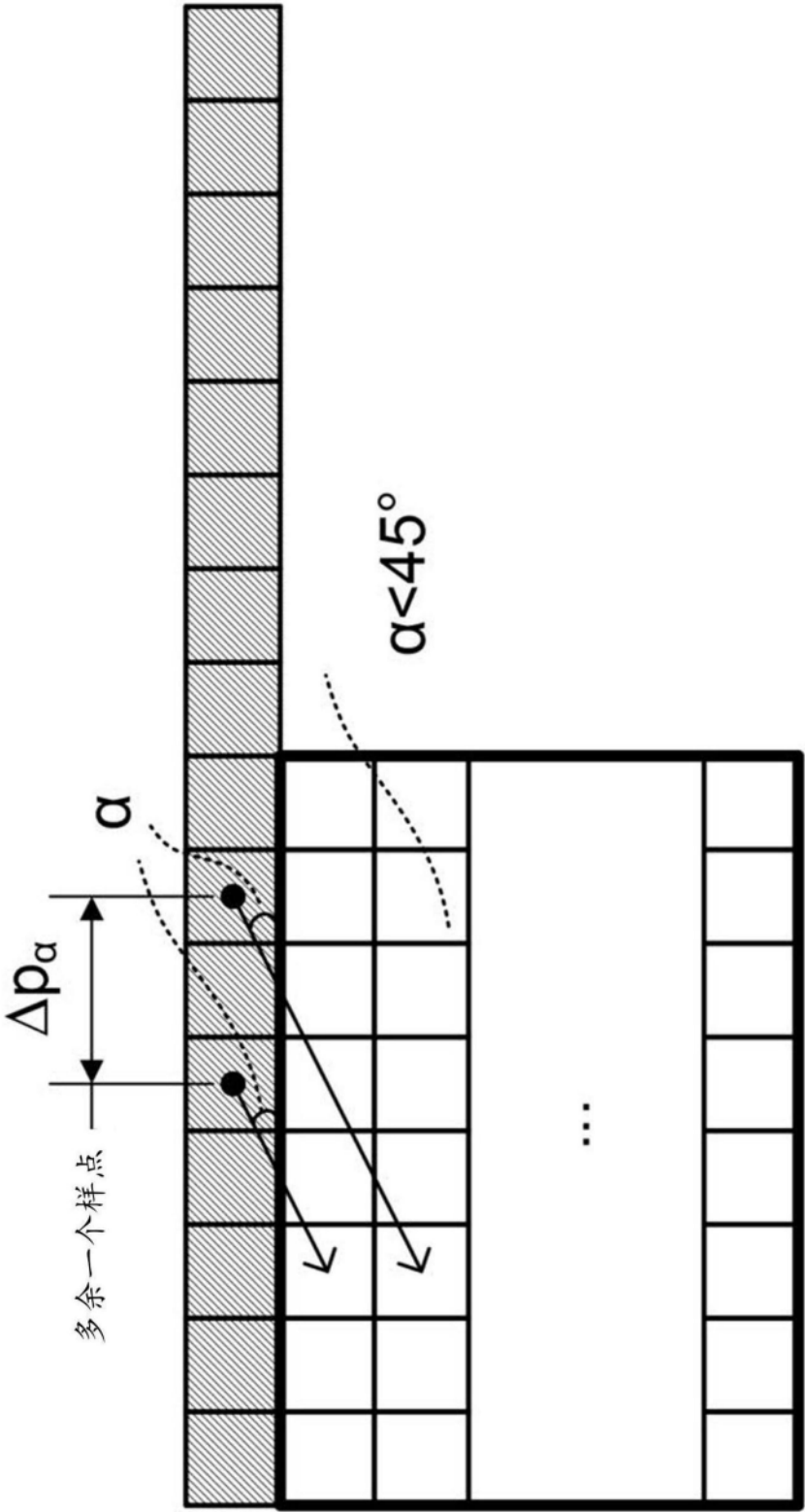


图4

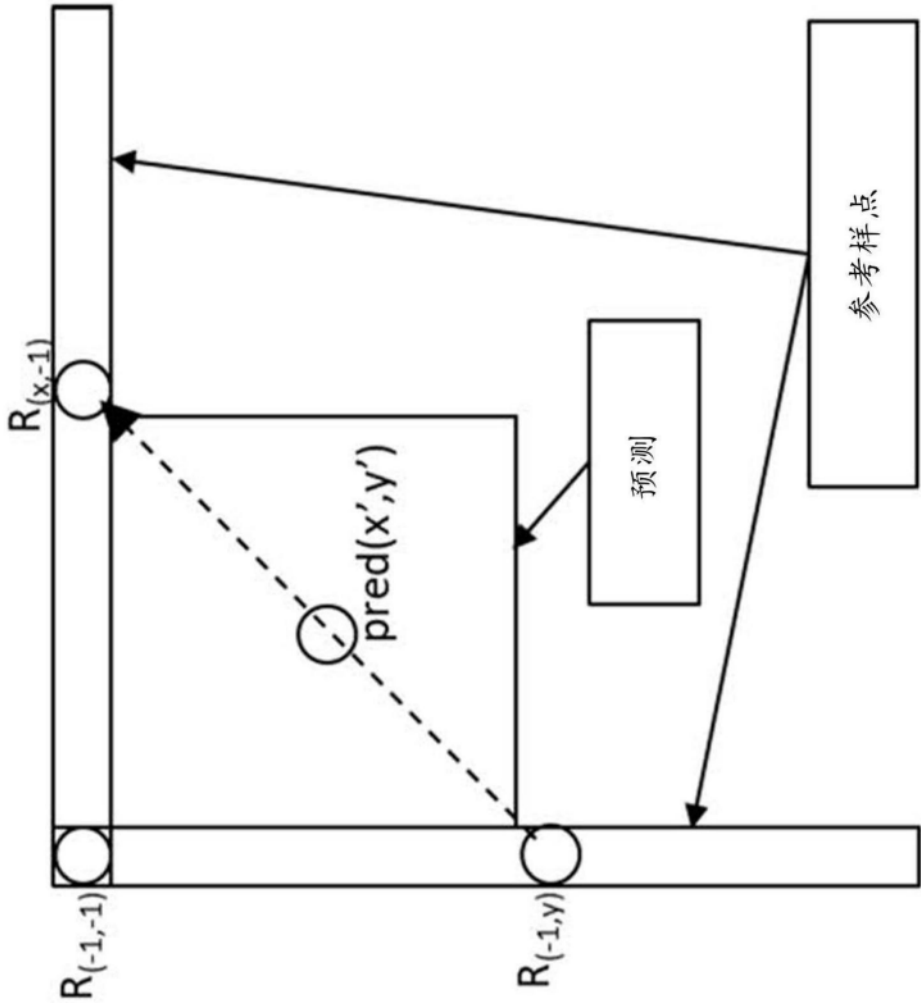


图5A

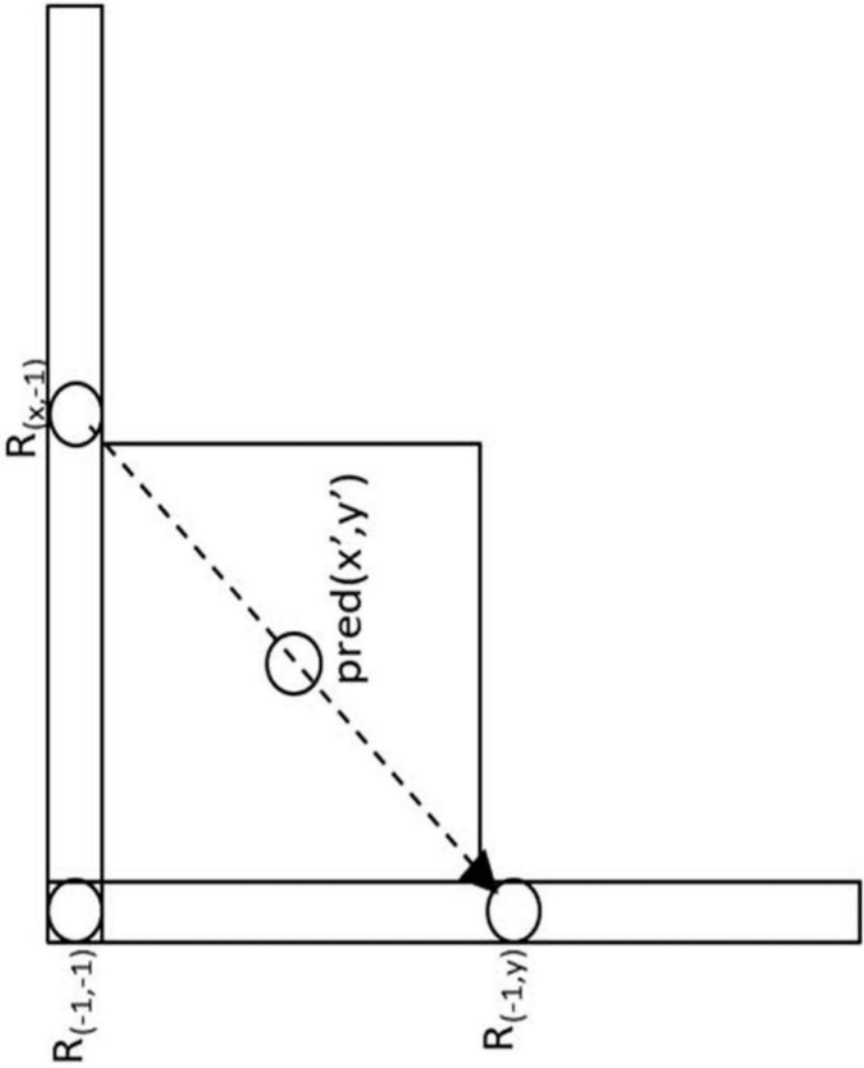


图5B

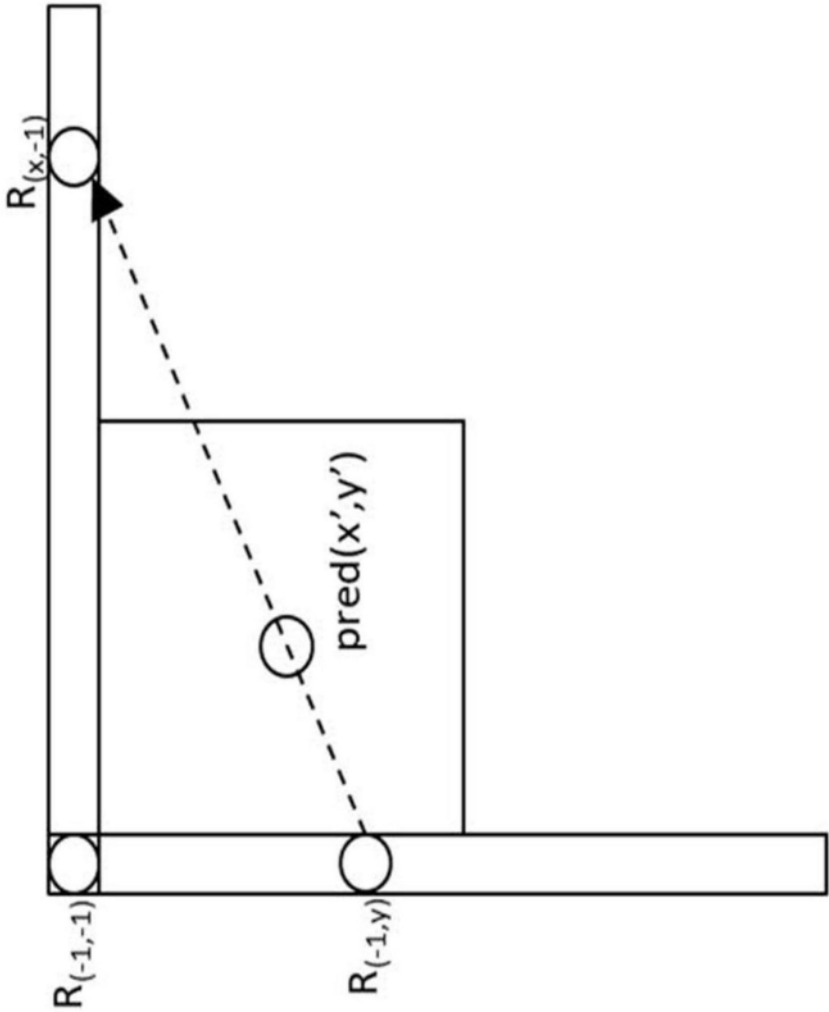


图5C

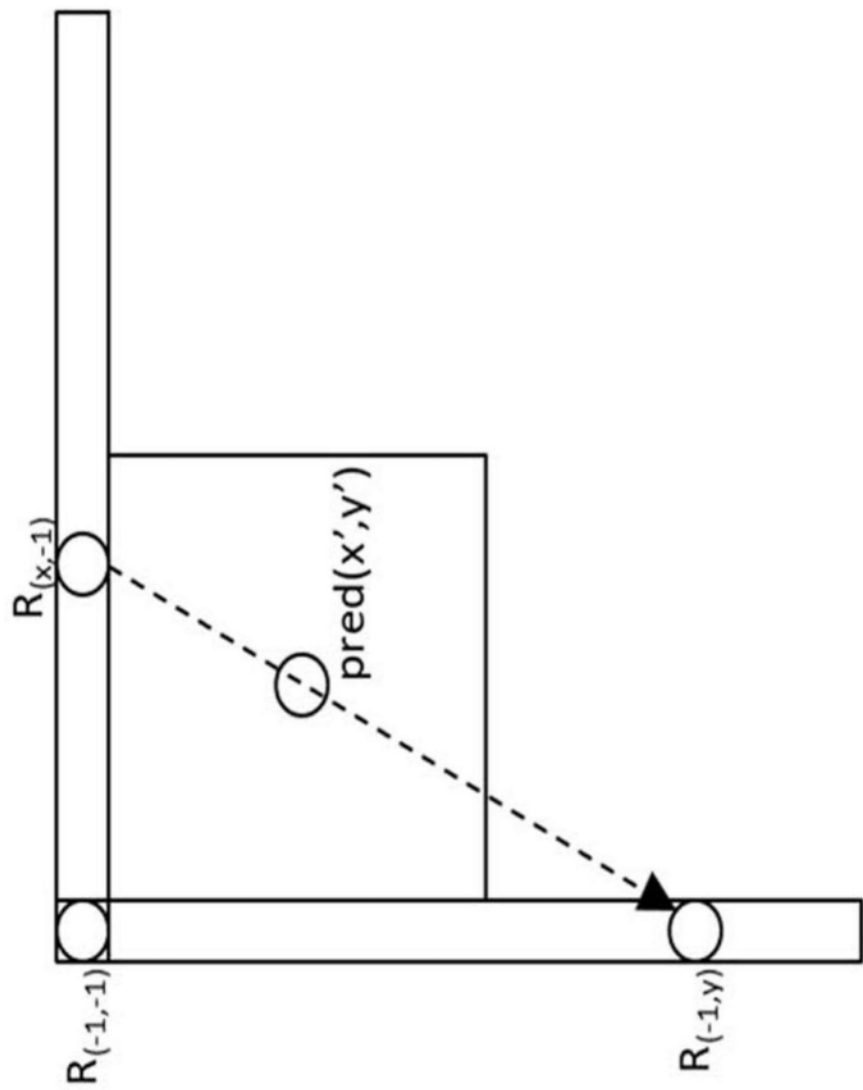


图5D

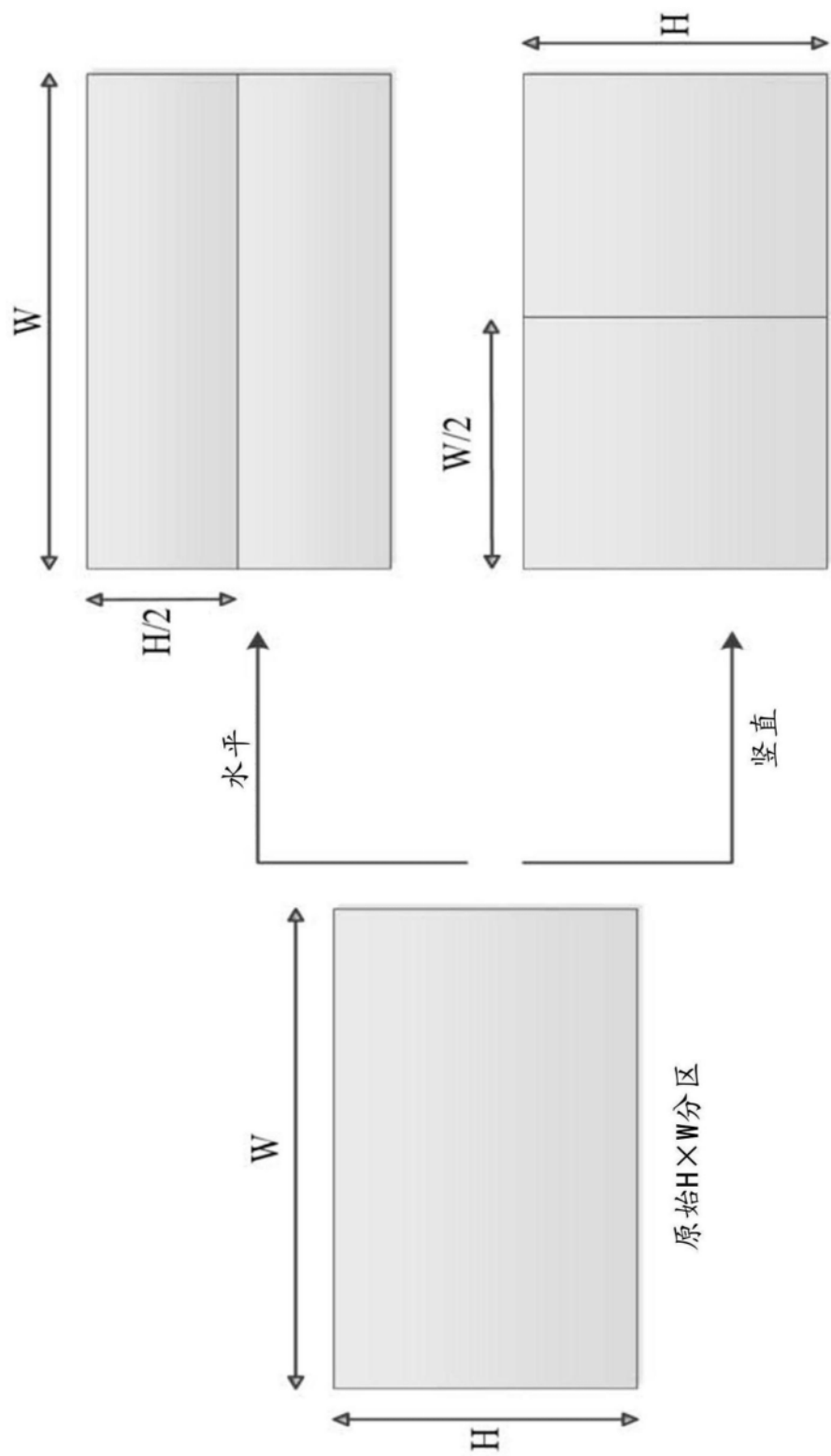


图6

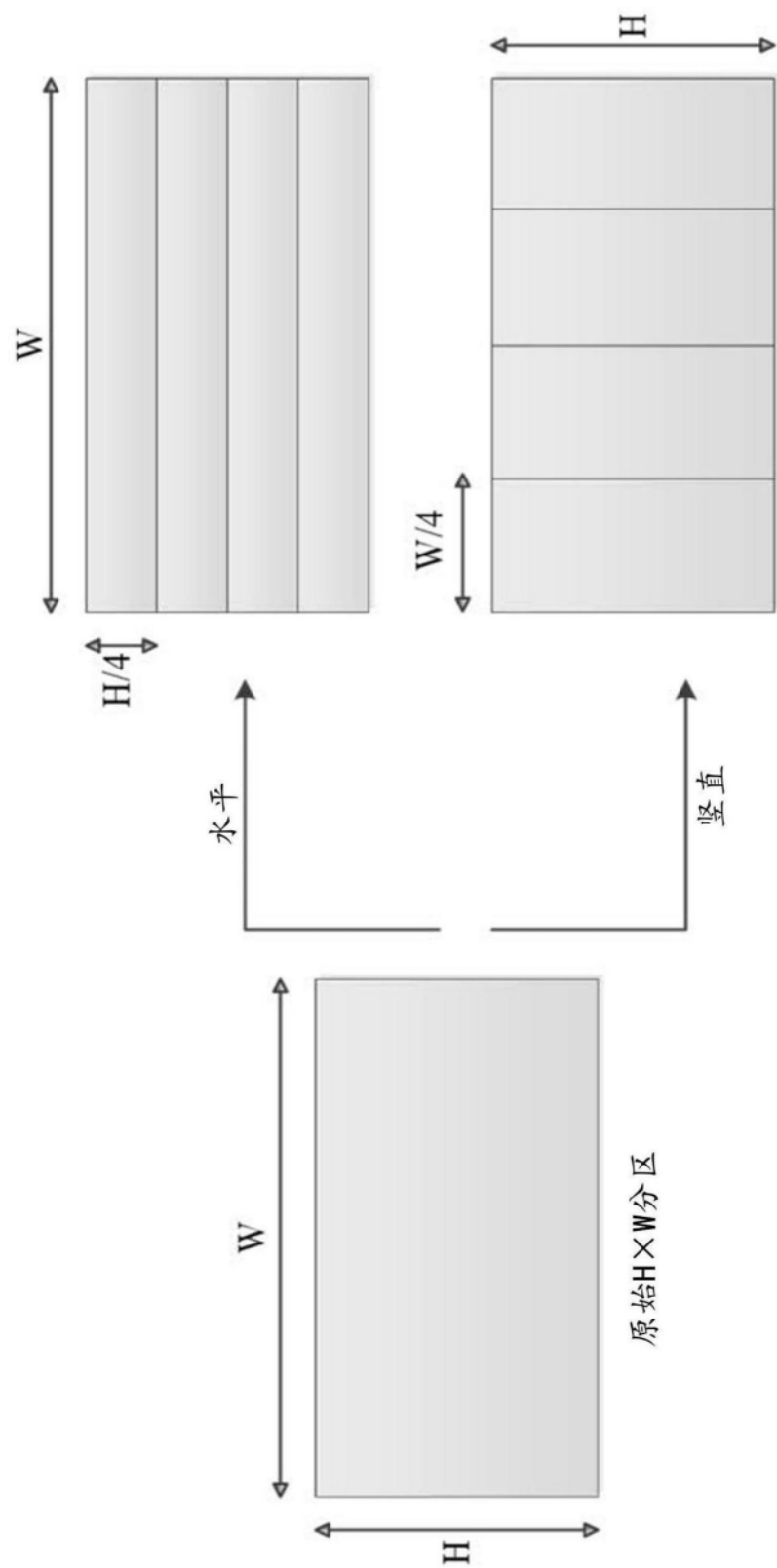


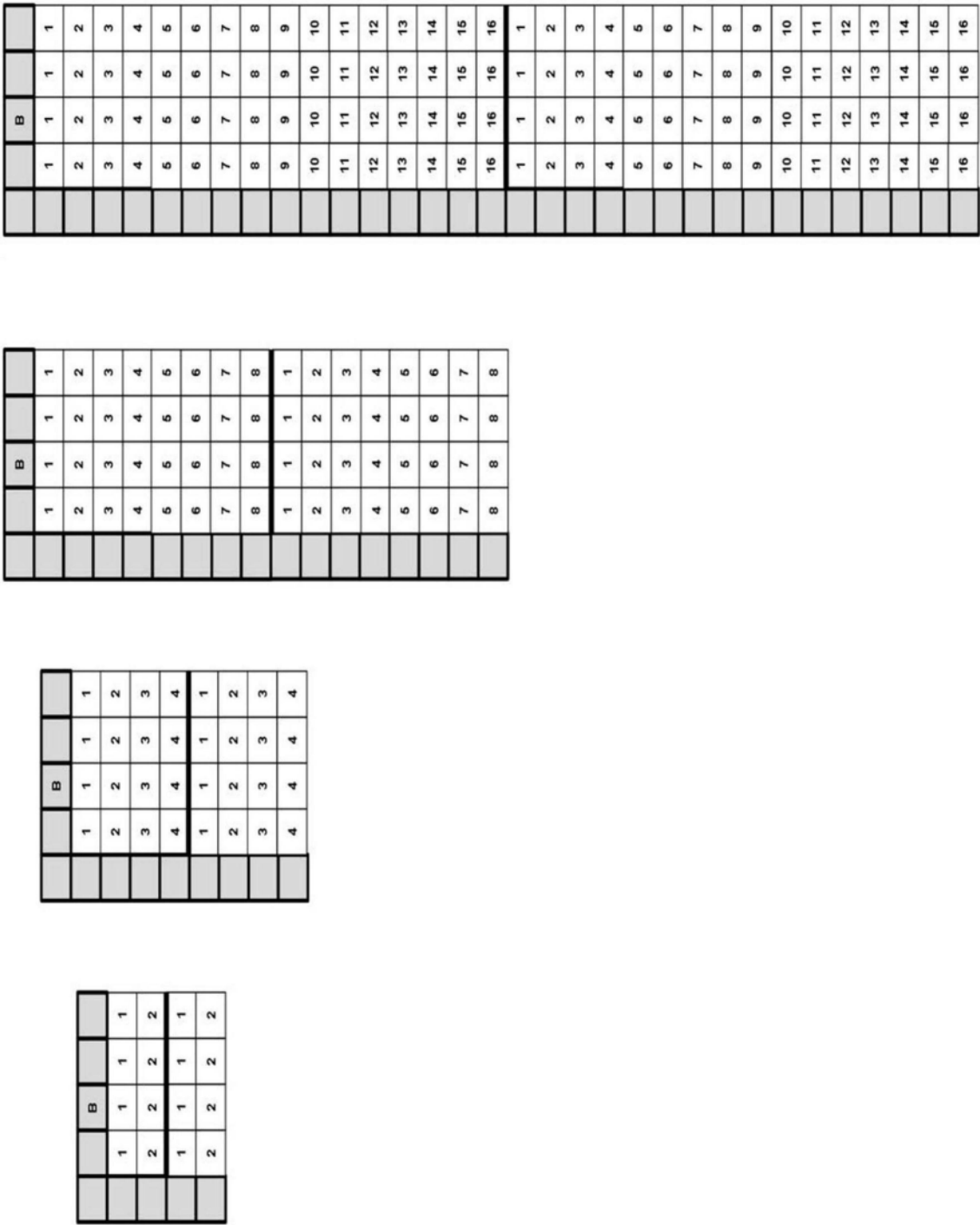
图7

					B	X		

B					X			

B	X							

图8



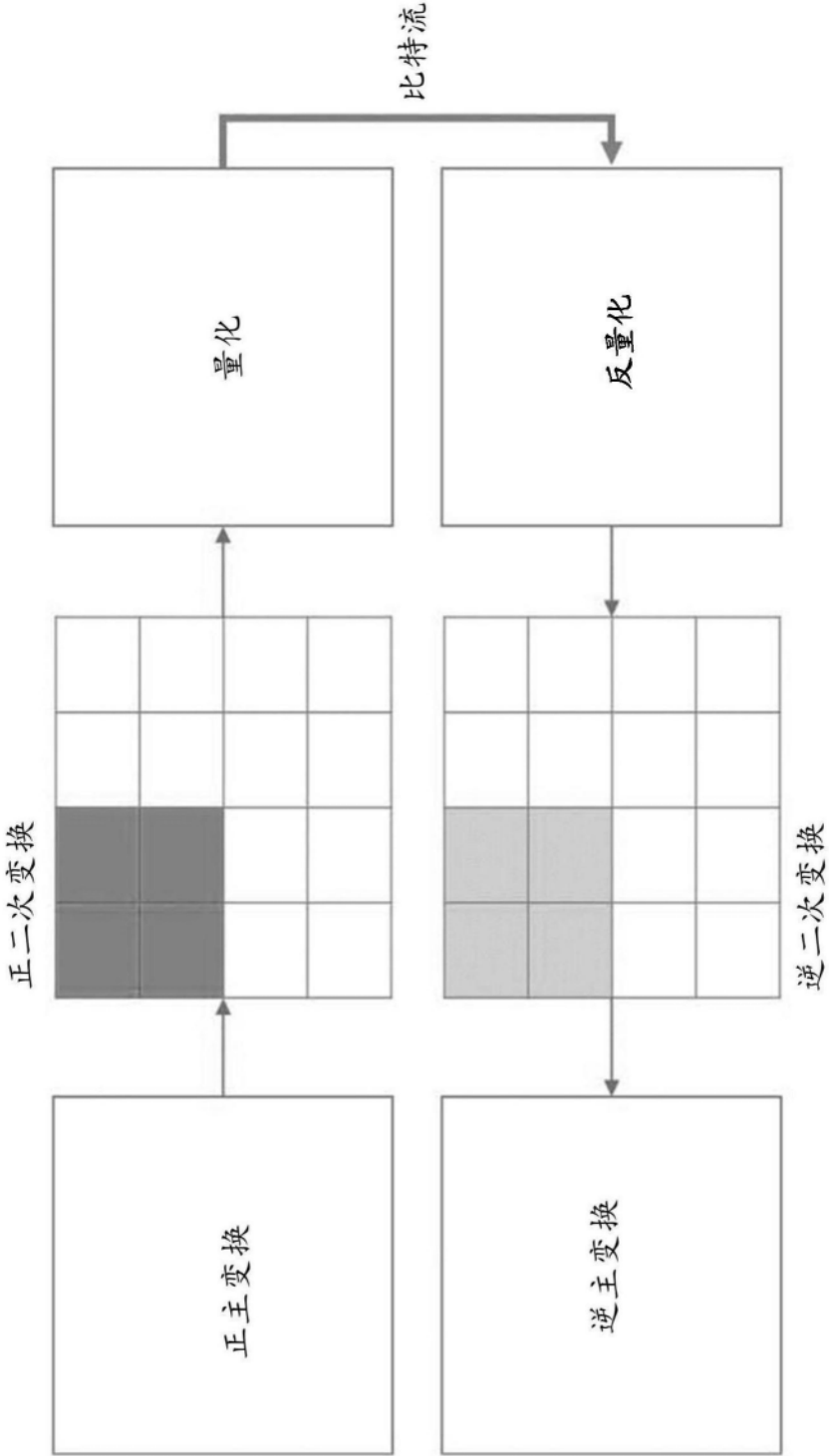


图10

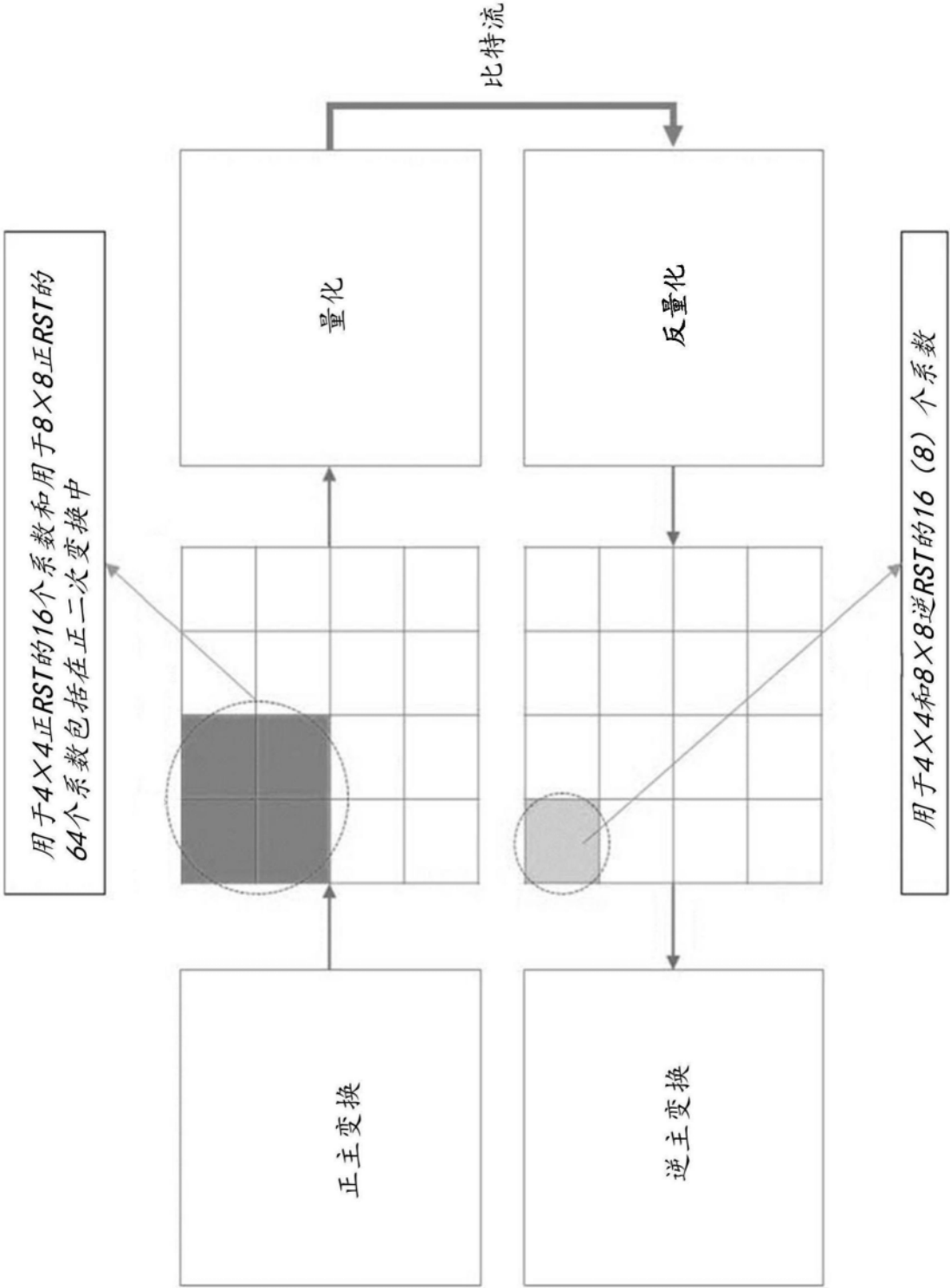


图11

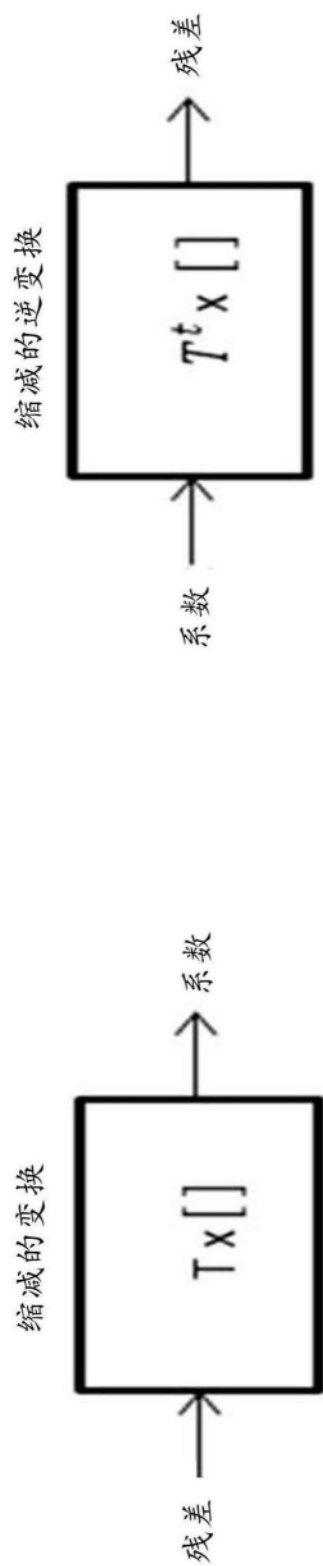


图12

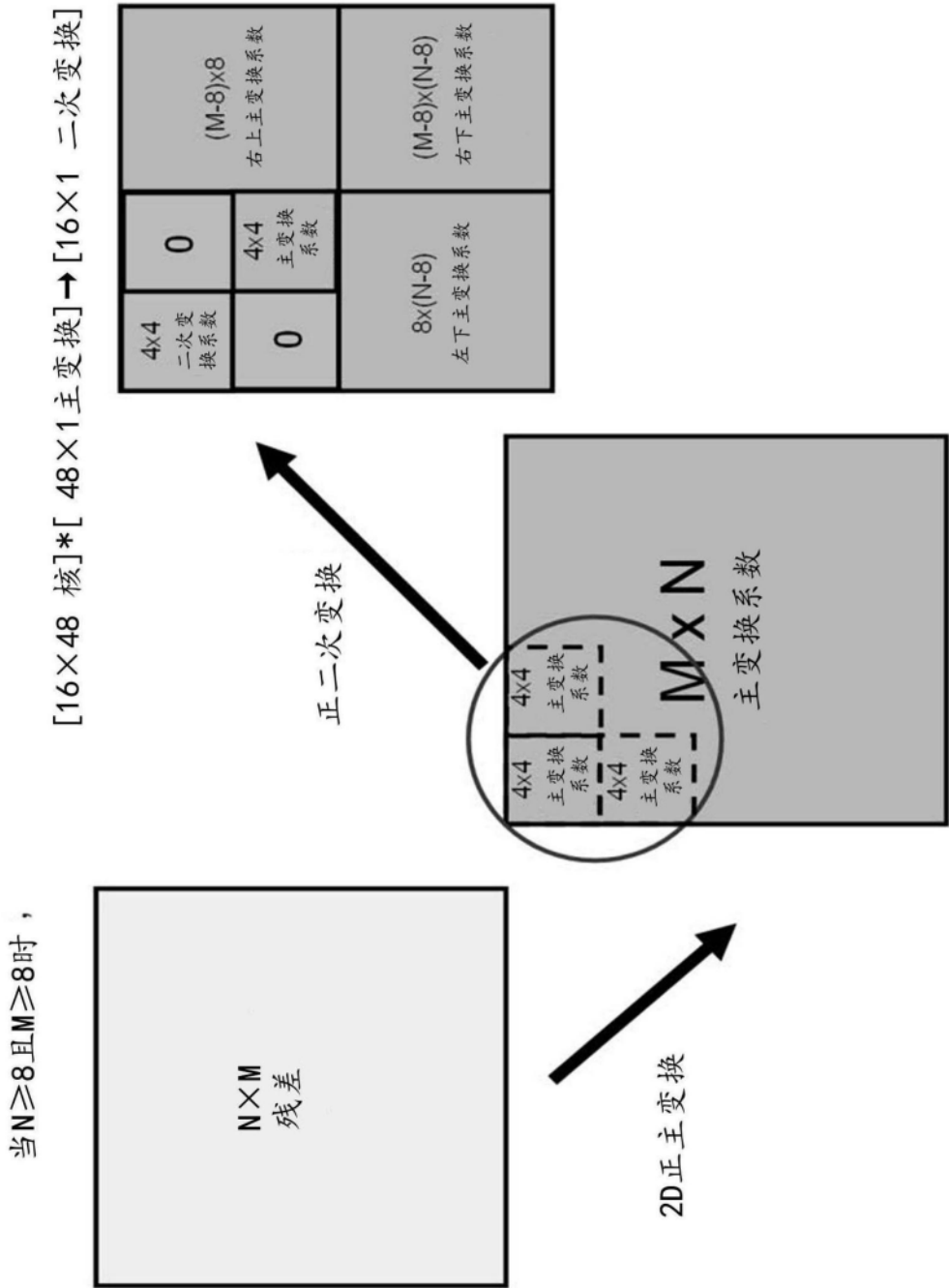


图13

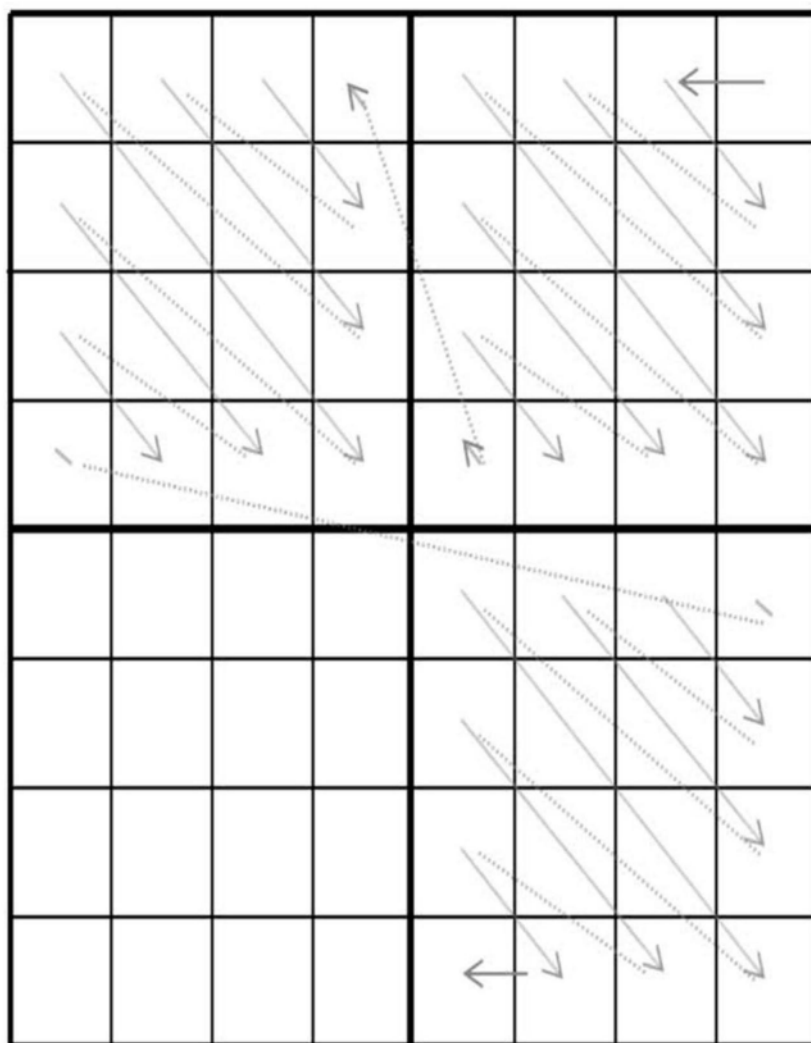


图14

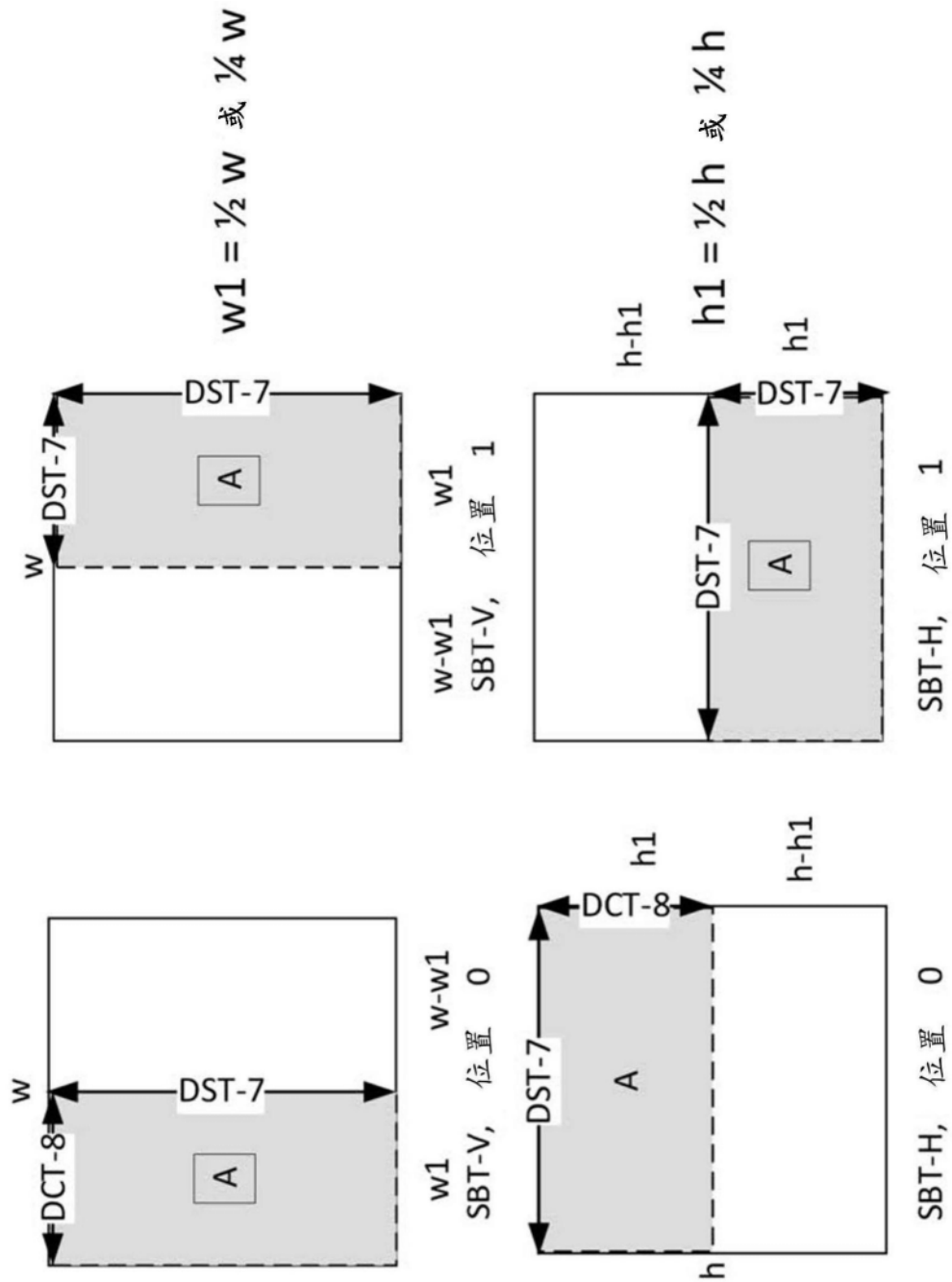


图15

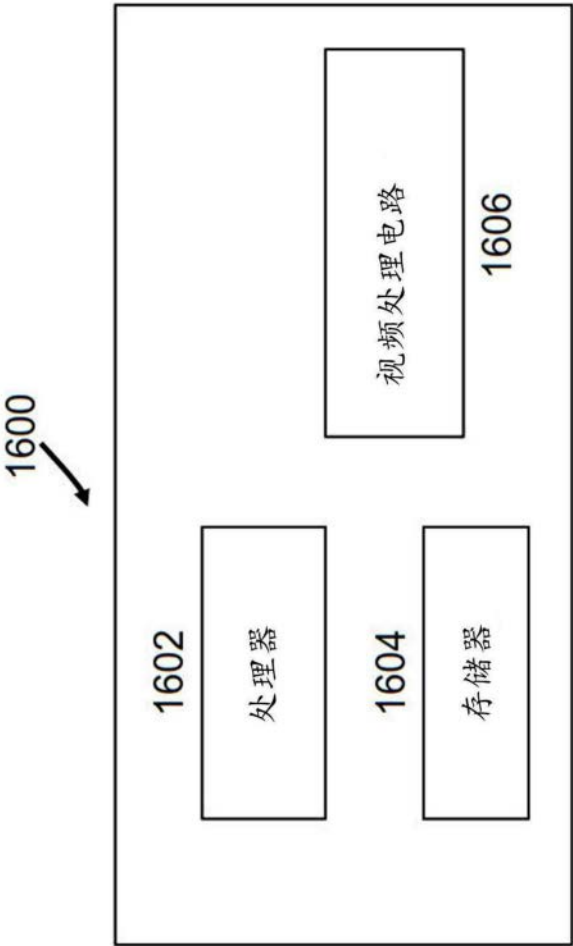


图16

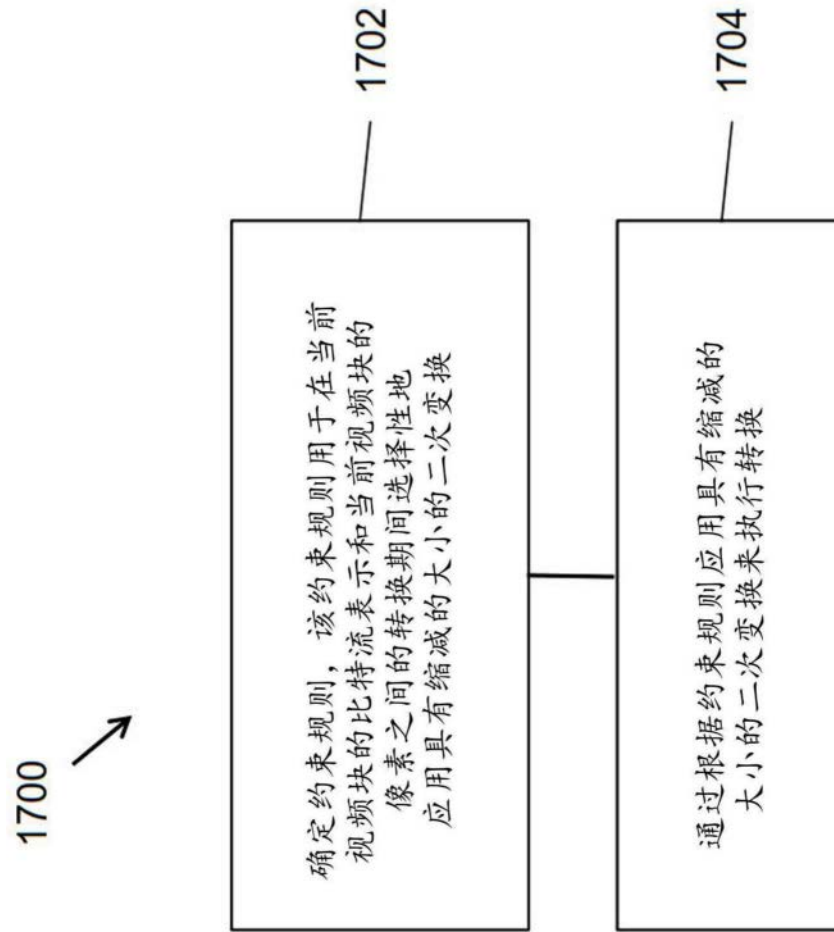


图17

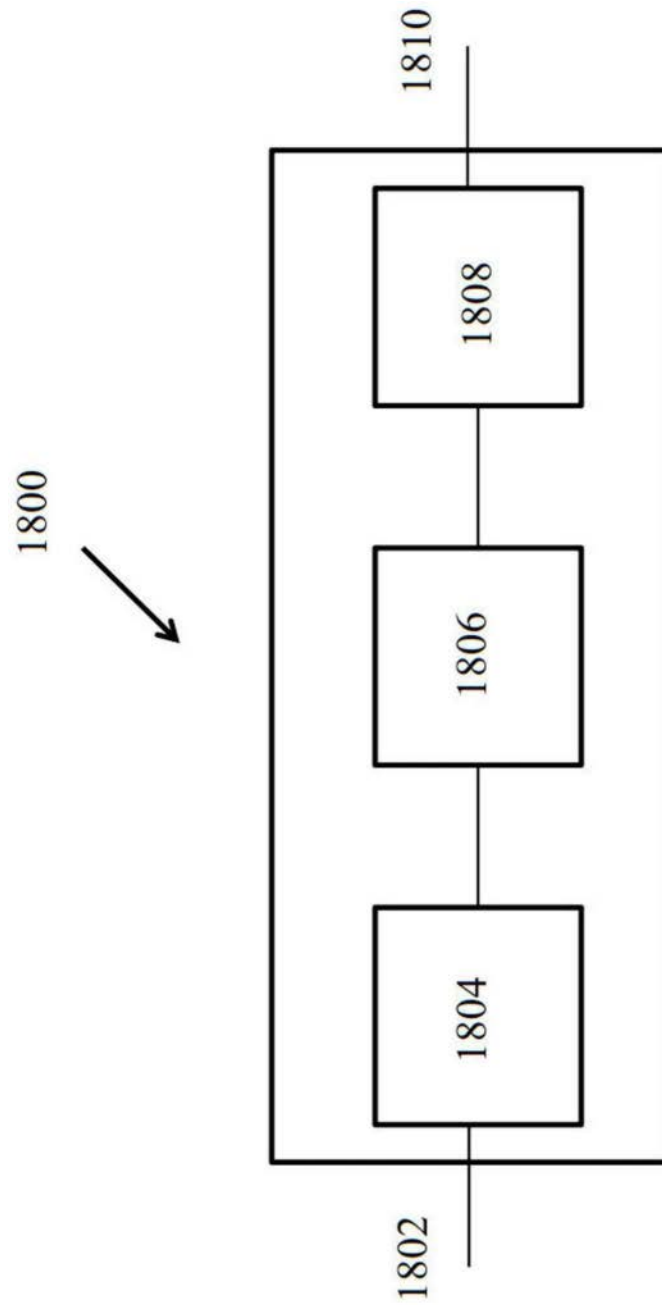


图18

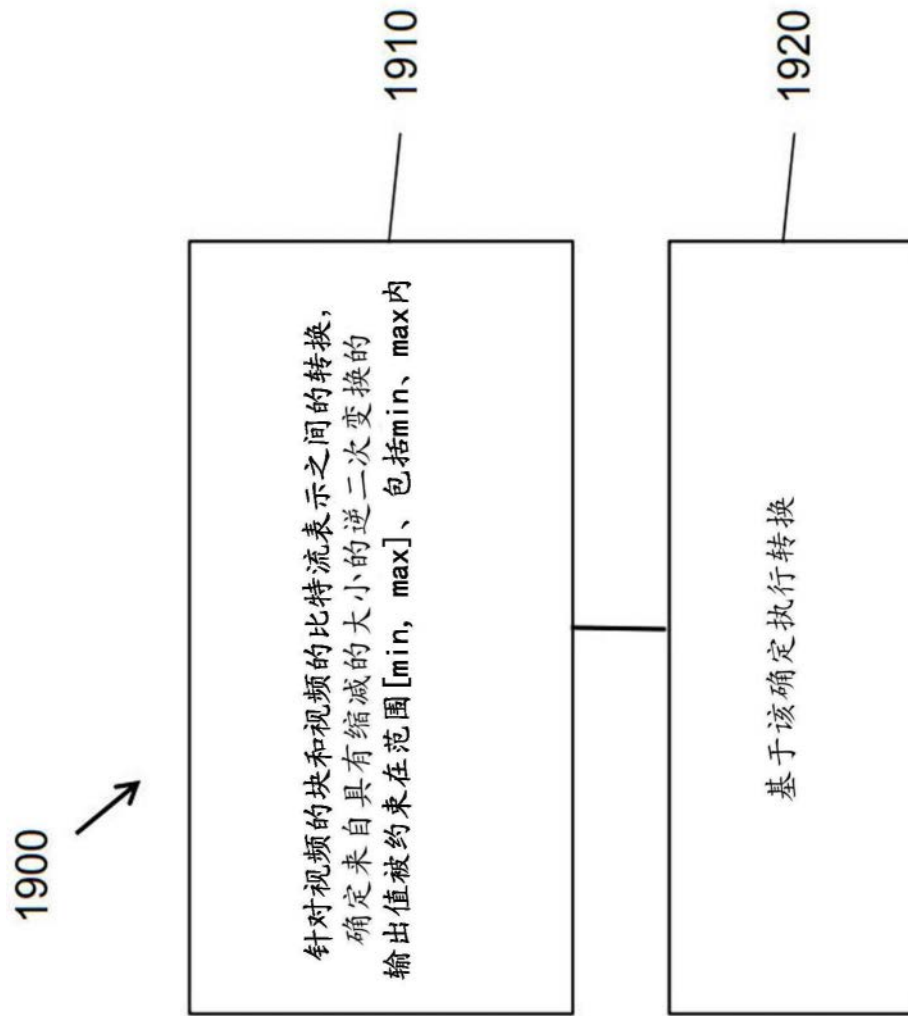


图19

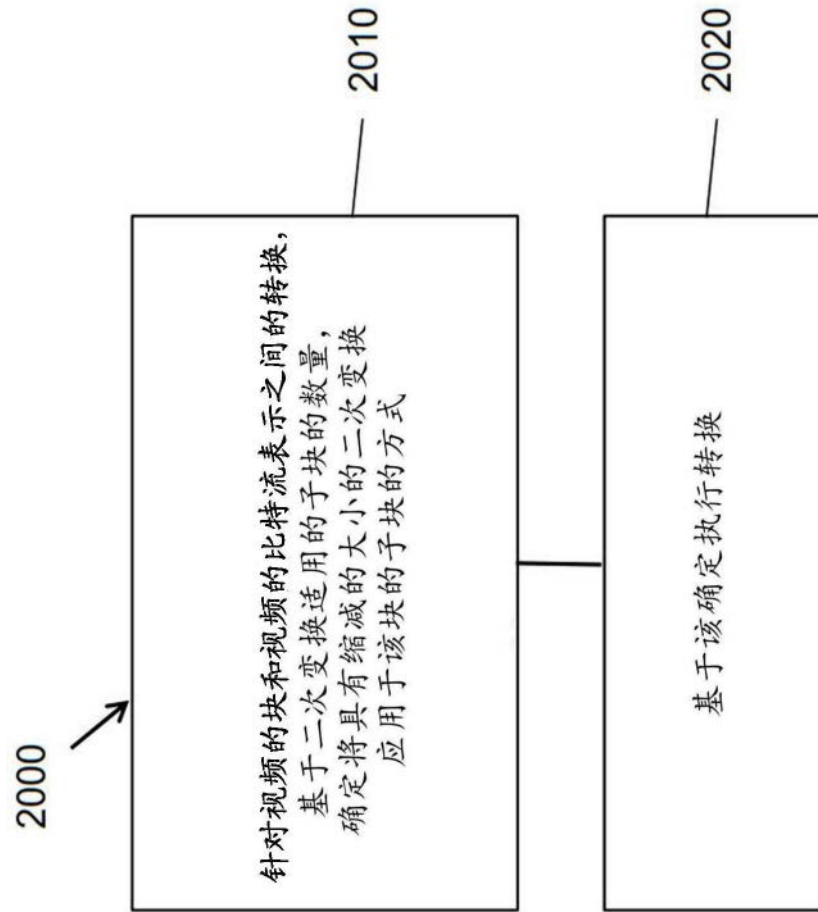


图20

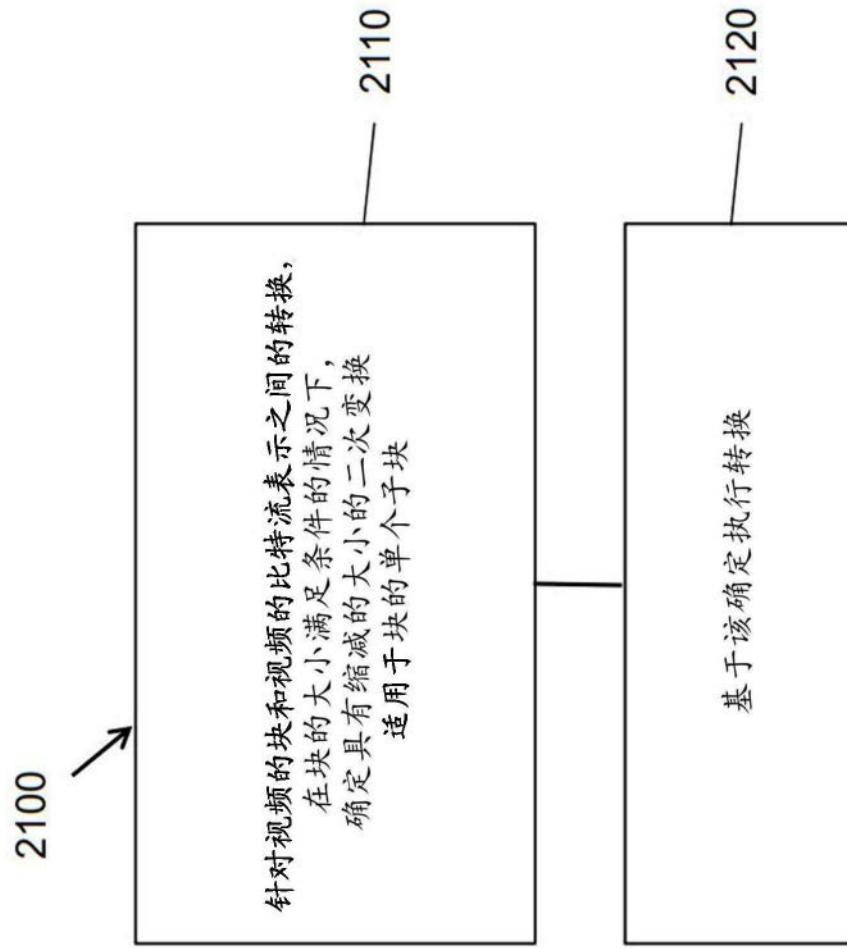


图21

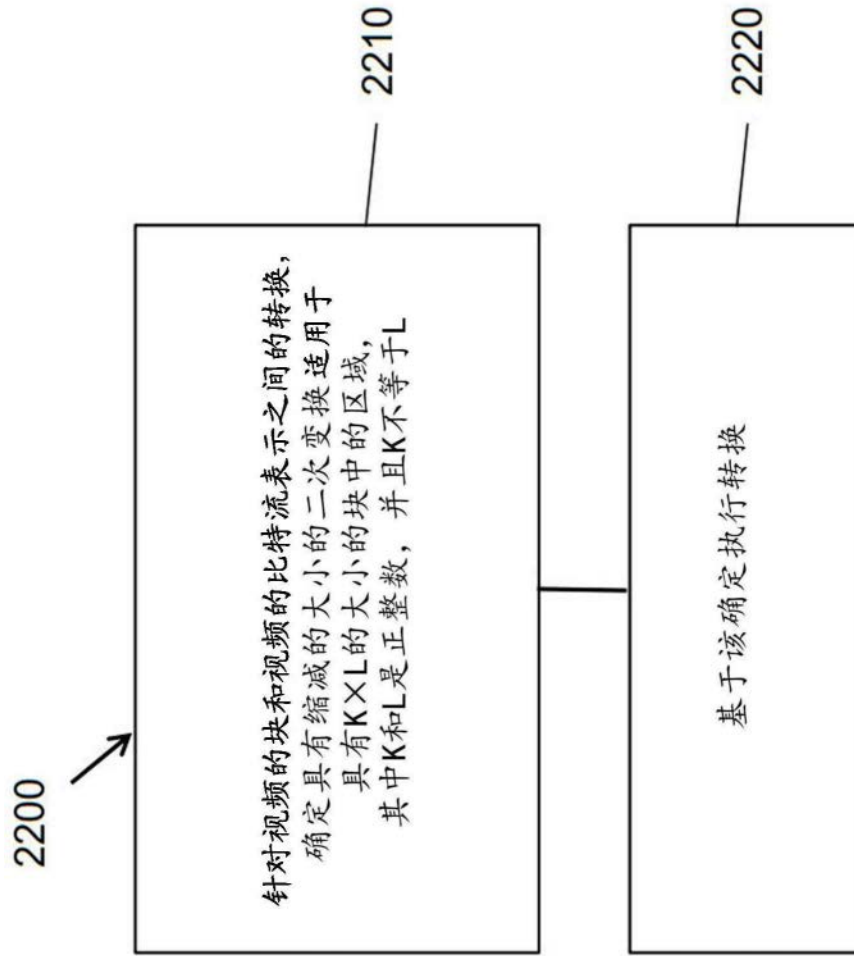


图22

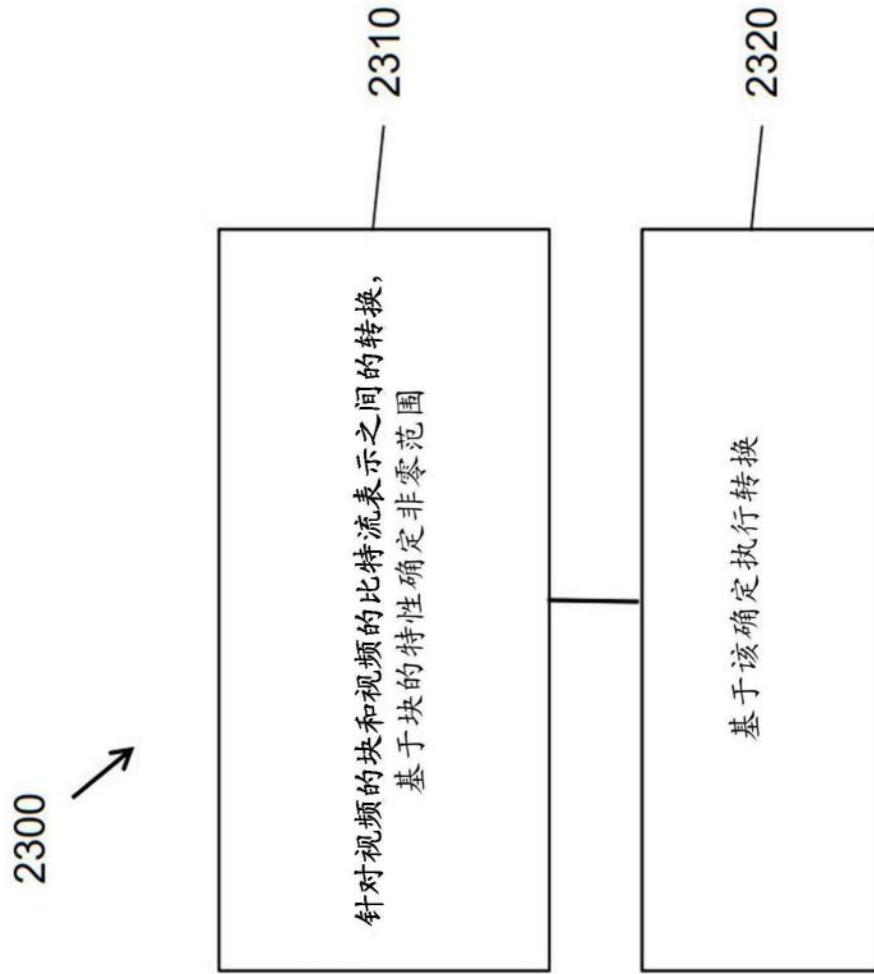


图23

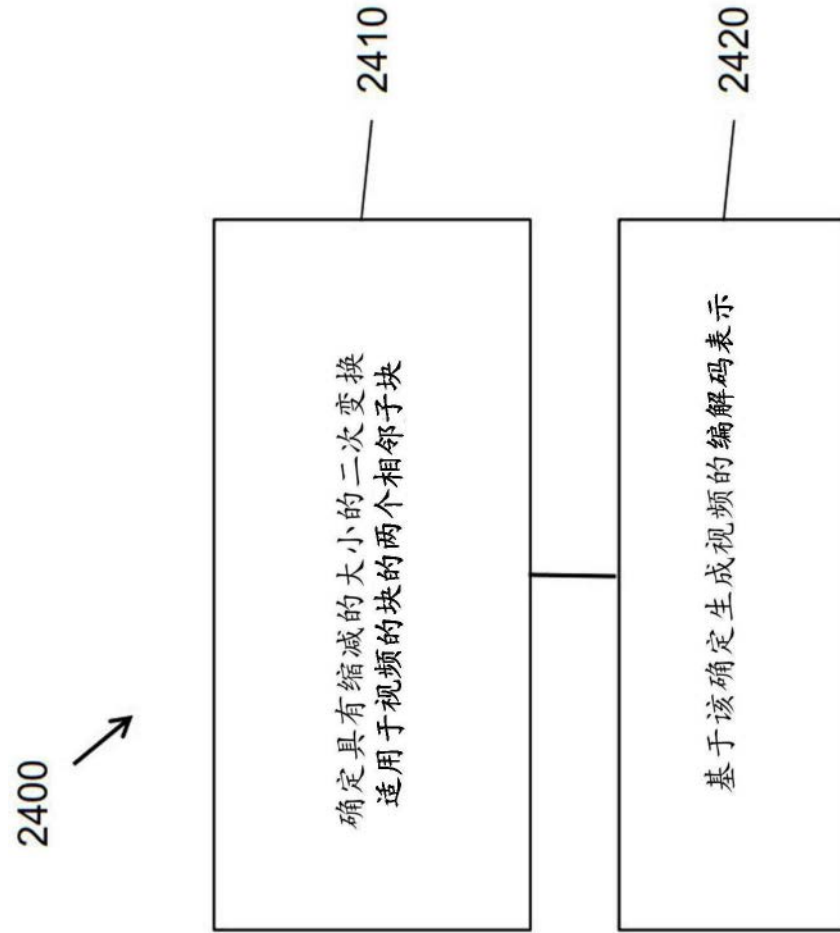


图24A

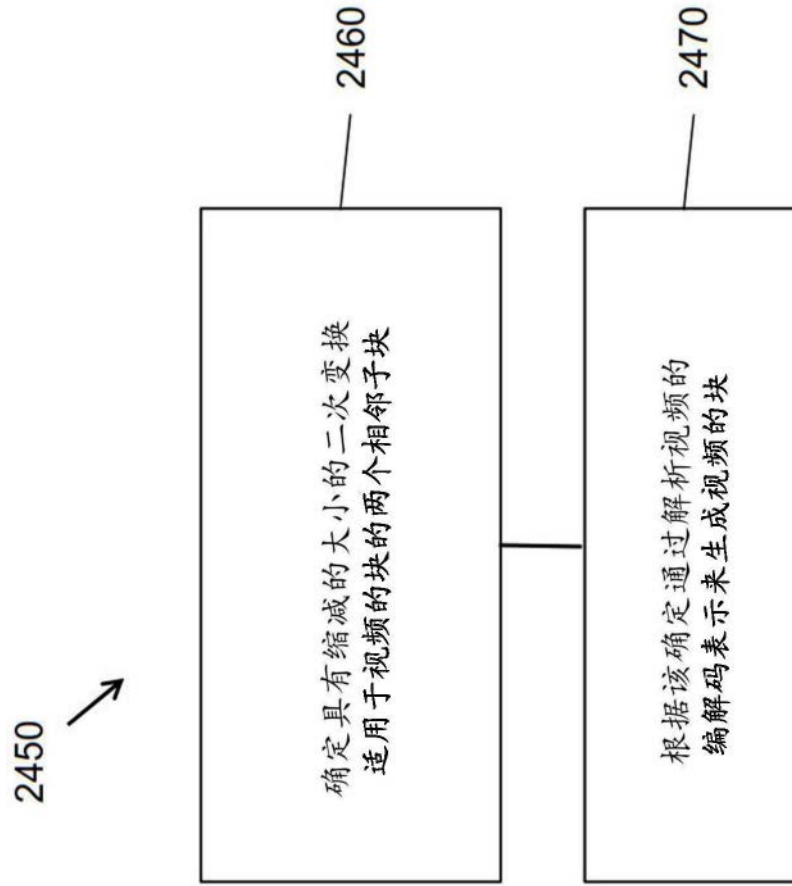


图24B

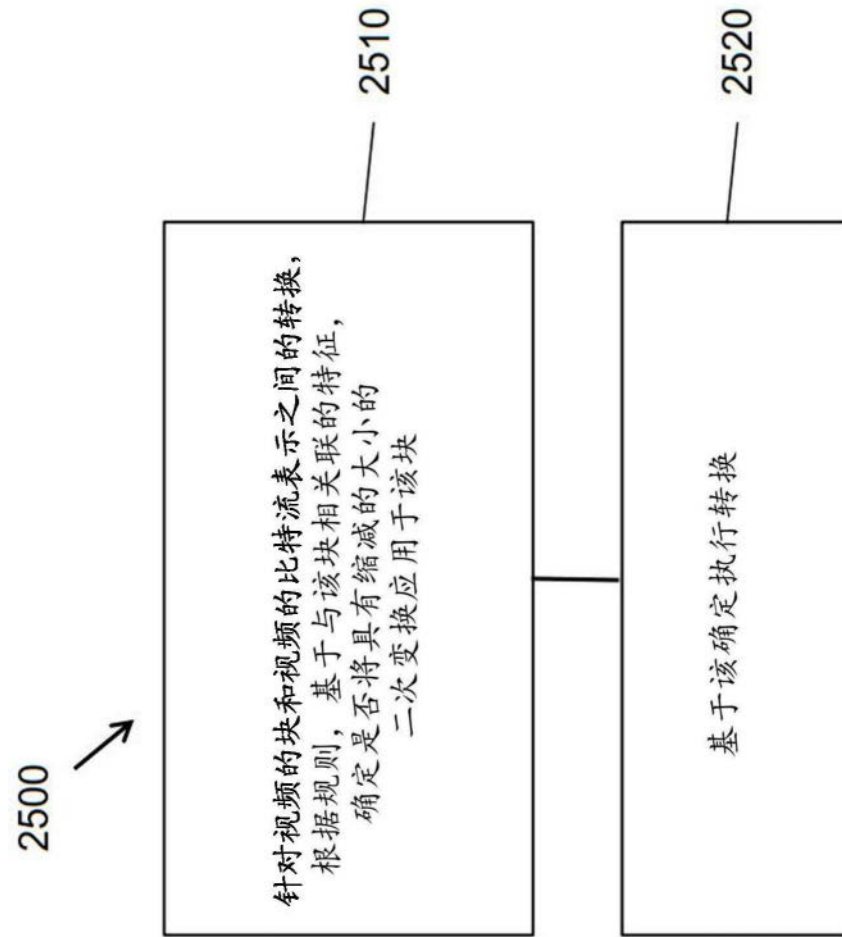


图25

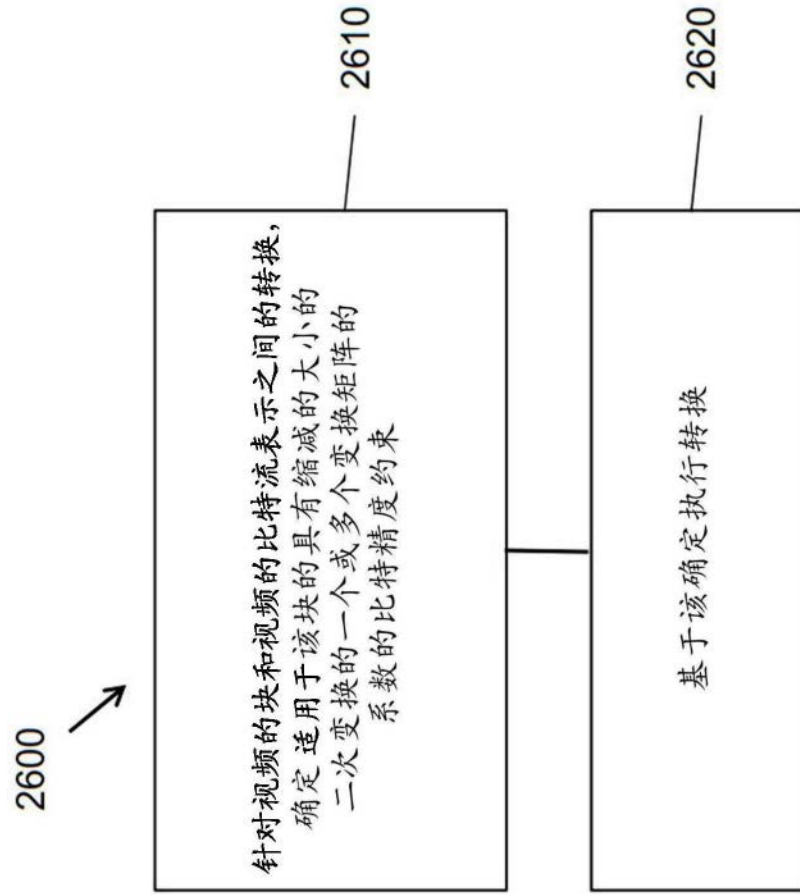


图26