



(12) 发明专利

(10) 授权公告号 CN 116724552 B

(45) 授权公告日 2025. 02. 07

(21) 申请号 202280008585.4

(22) 申请日 2022.09.14

(65) 同一申请的已公布的文献号  
申请公布号 CN 116724552 A

(43) 申请公布日 2023.09.08

(30) 优先权数据  
63/245,678 2021.09.17 US  
17/939,239 2022.09.07 US

(85) PCT国际申请进入国家阶段日  
2023.06.27

(86) PCT国际申请的申请数据  
PCT/US2022/043507 2022.09.14

(87) PCT国际申请的公布数据  
W02023/043824 EN 2023.03.23

(73) 专利权人 腾讯美国有限责任公司  
地址 美国加利福尼亚州帕洛阿尔托公园大道2747号

(72) 发明人 赵欣 许晓中 刘杉

(74) 专利代理机构 北京派特恩知识产权代理有限公司 11270  
专利代理师 刘晖铭 胡春光

(51) Int.Cl.  
H04N 19/593 (2006.01)  
H04N 19/176 (2006.01)  
H04N 19/159 (2006.01)  
H04N 19/105 (2006.01)  
H04N 19/52 (2006.01)

(56) 对比文件  
US 2019208217 A1, 2019.07.04  
Jiahao Li等. "Intra Block Copy for Screen Content in the Emerging AV1 Video Codec". 2018 Data Compression Conference. 2018, 355-364页.

审查员 李嘉蓉

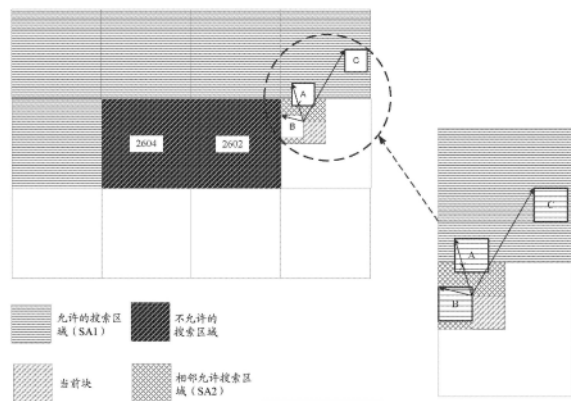
权利要求书4页 说明书35页 附图27页

(54) 发明名称

用于处理视频数据的方法、装置、设备及介质

(57) 摘要

本公开总体上涉及视频编解码,尤其涉及帧内块复制编解码模式。例如,公开了一种用于处理视频数据的方法,所述方法包括:接收视频比特流,所述视频比特流包括视频帧的至少一个块,所述至少一个块包括当前块;确定所述视频帧中的第一搜索区域;确定第二搜索区域,其中所述第二搜索区域是用于定位IntraBC预测块的第二候选区域;识别由块矢量参考的所述IntraBC预测块;基于所述IntraBC预测块、所述第一搜索区域和所述第二搜索区域之间的空间关系来确定预测;以及基于所确定的预测对所述当前块的至少部分进行解码。



1. 一种用于处理视频数据的方法,其特征在于,所述方法包括:

接收视频比特流,所述视频比特流包括视频帧的至少一个块,所述至少一个块包括当前块;

确定所述视频帧中的第一搜索区域,其中,所述第一搜索区域是用于帧内块复制(IntraBC)预测块搜索和选择的非局部搜索区域,其中,所述第一搜索区域与所述当前块不重叠、并且包括块列表,所述IntraBC预测块是用于对所述当前块的至少部分进行IntraBC预测的候选块;

确定第二搜索区域,其中,所述第二搜索区域是用于所述IntraBC预测块搜索和选择的局部搜索区域,所述第二搜索区域包括以下至少之一:(i)所述当前块的子块,以及(ii)所述当前块的已重建的相邻块;所述第一搜索区域和所述第二搜索区域不重叠;所述第一搜索区域中的样本存储在片外存储器中,所述第二搜索区域中的样本存储在片上存储器中;

识别由块矢量参考的所述IntraBC预测块;响应于所述IntraBC预测块部分地位于所述第一搜索区域中并且部分地位于所述第二搜索区域时,将所述IntraBC预测块与所述第一搜索区域重叠的区域确定为第一重叠区域,将所述IntraBC预测块与所述第二搜索区域重叠的区域确定为第二重叠区域;

基于所述第一重叠区域的大小和所述第二重叠区域的大小,从所述第一重叠区域和所述第二重叠区域中选择其中之一作为目标重叠区域;以及

使用与所述IntraBC预测块中的所述目标重叠区域相邻的边界样本替换所述目标重叠区域中的样本,以获得更新的IntraBC预测块;以及

基于所述更新的IntraBC预测块解码所述当前块的至少部分。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

响应于所述IntraBC预测块是所述第一搜索区域的子集或所述第二搜索区域的子集,允许使用所述IntraBC预测块对所述当前块的至少部分进行IntraBC预测,所述第一搜索区域和所述第二搜索区域不重叠。

3. 根据权利要求2所述的方法,其特征在于,所述当前块包括超级块。

4. 根据权利要求1所述的方法,其特征在于,所述当前块的相邻块与所述当前块在同一行中且位于所述当前块的左侧,以及所述当前块的相邻块具有与所述当前块相同的大小。

5. 根据权利要求1至4任一项所述的方法,其特征在于:

所述当前块的左上角像素的坐标位置是 $(x_0, y_0)$ ;

所述块列表中每个块的左上角像素的坐标位置是 $(x, y)$ ;

$y < y_0$ ; 以及

$x < [x_0 + 2(y_0 - y) - D]$ ,

其中, $x_0$ 、 $y_0$ 、 $x$ 和 $y$ 是非负数, $D$ 是针对IntraBC模式所限制的立即重建块的数量。

6. 根据权利要求5所述的方法,其特征在于, $D$ 等于2。

7. 根据权利要求1所述的方法,其特征在于,所述从所述第一重叠区域和所述第二重叠区域中选择其中之一作为目标重叠区域包括:

响应于所述第二重叠区域的大小大于或等于所述第一重叠区域的大小,选择所述第一重叠区域作为所述目标重叠区域;以及

响应于所述第一重叠区域的大小大于所述第二重叠区域的大小,选择所述第二重叠区

域作为所述目标重叠区域。

8. 根据权利要求1所述的方法,其特征在于,所述从所述第一重叠区域和所述第二重叠区域中选择其中之一作为目标重叠区域包括:

响应于所述第二重叠区域的大小大于或等于第一阈值,选择所述第一重叠区域作为所述目标重叠区域,其中,所述第一阈值是所述第一重叠区域的大小与预定义的第一偏置因子的乘积;以及

响应于所述第一重叠区域的大小大于或等于第二阈值,选择所述第二重叠区域作为所述目标重叠区域,其中,所述第二阈值是所述第二重叠区域的大小与预定义的第二偏置因子的乘积。

9. 根据权利要求8所述的方法,其特征在于,所述第一重叠区域和所述第二重叠区域的所述大小是由每个相应区域中的样本的数目表示的。

10. 根据权利要求9所述的方法,其中,所述样本的数目包括以下之一:

亮度样本的数目;或者

色度样本的数目。

11. 根据权利要求8至10任一项所述的方法,其特征在于,所述替换所述目标重叠区域中的样本包括:

响应于所述目标重叠区域是所述第一重叠区域,使用来自所述第二重叠区域中的样本替换所述目标重叠区域中的样本;以及

响应于所述目标重叠区域是所述第二重叠区域,使用来自所述第一重叠区域中的样本替换所述目标重叠区域中的样本。

12. 根据权利要求1至4任一项所述的方法,其特征在于,所述第二搜索区域中的样本存储在处理所述视频数据的视频编解码器的片上存储器中。

13. 一种用于处理视频数据的设备,其特征在于,包括用于存储计算机指令的存储器和与所述存储器通信的处理器,其中,当所述处理器执行所述计算机指令时,所述处理器被配置为使得所述设备:

接收视频比特流,所述视频比特流包括视频帧的至少一个块,所述至少一个块包括当前块;

确定所述视频帧中的第一搜索区域,其中,所述第一搜索区域是用于帧内块复制(IntraBC)预测块搜索和选择的非局部搜索区域,其中,所述第一搜索区域与所述当前块不重叠、并且包括块列表,所述IntraBC预测块是用于对所述当前块的至少部分进行IntraBC预测的候选块;

确定第二搜索区域,其中,所述第二搜索区域是用于所述IntraBC预测块搜索和选择的局部搜索区域,所述第二搜索区域包括以下至少之一:(i)所述当前块的子块,以及(ii)所述当前块的已重建的相邻块;所述第一搜索区域和所述第二搜索区域不重叠;所述第一搜索区域中的样本存储在片外存储器中,所述第二搜索区域中的样本存储在片上存储器中;

识别由块矢量参考的所述IntraBC预测块;响应于所述IntraBC预测块部分地位于所述第一搜索区域中并且部分地位于所述第二搜索区域时,将所述IntraBC预测块与所述第一搜索区域重叠的区域确定为第一重叠区域,将所述IntraBC预测块与所述第二搜索区域重叠的区域确定为第二重叠区域;

基于所述第一重叠区域的大小和所述第二重叠区域的大小,从所述第一重叠区域和所述第二重叠区域中选择其中之一作为目标重叠区域;以及

使用与所述IntraBC预测块中的所述目标重叠区域相邻的边界样本替换所述目标重叠区域中的样本,以获得更新的IntraBC预测块;以及

基于所述更新的IntraBC预测块解码所述当前块的至少部分。

14. 根据权利要求13所述的设备,其特征在于,当所述处理器被配置为使得所述设备选择所述目标重叠区域时,所述处理器被配置为使得所述设备:

响应于所述第二重叠区域的大小大于或等于所述第一重叠区域的大小,选择所述第一重叠区域作为所述目标重叠区域;以及

响应于所述第一重叠区域的大小大于所述第二重叠区域的大小,选择所述第二重叠区域作为所述目标重叠区域。

15. 根据权利要求13所述的设备,其特征在于,当所述处理器被配置为使得所述设备选择所述目标重叠区域时,所述处理器被配置为使得所述设备:

响应于所述第二重叠区域的大小大于或等于第一阈值,选择所述第一重叠区域作为所述目标重叠区域,其中,所述第一阈值是所述第一重叠区域的大小与预定义的第一偏置因子的乘积;以及

响应于所述第一重叠区域的大小大于或等于第二阈值,选择所述第二重叠区域作为所述目标重叠区域,其中,所述第二阈值是所述第二重叠区域的大小和预定义的第二偏置因子的乘积。

16. 一种非暂时性存储介质,其特征在于,用于存储计算机可读指令,当所述计算机可读指令由处理器执行时,使得所述处理器:

接收视频比特流,所述视频比特流包括视频帧的至少一个块,所述至少一个块包括当前块;

确定所述视频帧中的第一搜索区域,其中,所述第一搜索区域是用于帧内块复制(IntraBC)预测块搜索和选择的非局部搜索区域,其中,所述第一搜索区域与所述当前块不重叠、并且包括块列表,所述IntraBC预测块是用于对所述当前块的至少部分进行IntraBC预测的候选块;

确定第二搜索区域,其中,所述第二搜索区域是用于所述IntraBC预测块搜索和选择的局部搜索区域,所述第二搜索区域包括以下至少之一:(i)所述当前块的子块,以及(ii)所述当前块的已重建的相邻块;所述第一搜索区域和所述第二搜索区域不重叠;所述第一搜索区域中的样本存储在片外存储器中,所述第二搜索区域中的样本存储在片上存储器中;

识别由块矢量参考的所述IntraBC预测块;响应于所述IntraBC预测块部分地位于所述第一搜索区域中并且部分地位于所述第二搜索区域时,将所述IntraBC预测块与所述第一搜索区域重叠的区域确定为第一重叠区域,将所述IntraBC预测块与所述第二搜索区域重叠的区域确定为第二重叠区域;

基于所述第一重叠区域的大小和所述第二重叠区域的大小,从所述第一重叠区域和所述第二重叠区域中选择其中之一作为目标重叠区域;以及

使用与所述IntraBC预测块中的所述目标重叠区域相邻的边界样本替换所述目标重叠区域中的样本,以获得更新的IntraBC预测块;以及

基于所述更新的IntraBC预测块解码所述当前块的至少部分。

17.一种处理视频码流的方法,其特征在于,所述视频码流基于权利要求1至12任一项所述的用于处理视频数据的方法进行解码。

## 用于处理视频数据的方法、装置、设备及介质

[0001] 交叉引用

[0002] 本申请是基于且要求于2022年9月7日提交的标题为“Method and Apparatus for Intra Block Copy (IntraBC) Mode Coding with Search Range Restrictions”的美国非临时申请第17/939,239号优先权权益,该美国非临时申请是基于且要求于2021年9月17日提交的标题为“Method and Apparatus for Intra Block Copy (IntraBC) Mode Coding with Search Range Restrictions”的美国临时申请第63/245,678号优先权权益,这两个申请中的每一个均通过引用整体并入本文中。

### 技术领域

[0003] 本公开描述了一组高级视频编码技术。更具体地,所公开的技术涉及视频编码和解码中的帧内块复制(IntraBC,或IBC)的实现和增强。

### 背景技术

[0004] 本文所提供的背景描述是出于总体上呈现本公开的内容的目的。在该背景部分中描述的范围,目前命名的发明人的作品以及在本申请提交之时不可另行具备现有技术资格的本描述的各方面既未明确、亦未默示地承认为本公开的现有技术。

[0005] 可以使用具有运动补偿的帧间图片预测来执行视频编码和解码。未压缩的数字视频可以包括一系列图片,每个图片具有例如为1920x1080的亮度样本及相关的全采样或子采样的色度样本的空间维度。该系列图片可以具有例如每秒60张图片或每秒60帧的固定或可变图片速率(或者称为帧率)。未压缩的视频对于流式传输或数据处理具有很高的位速率要求。例如,在每个颜色通道每像素8比特下,像素分辨率为1920x1080、帧率为60帧/秒、并且色度子采样为4:2:0的视频需要接近1.5Gbit/s的带宽。一小时的此类视频需要600GB以上的存储空间。

[0006] 视频编码和解码的一个目的可以是通过压缩来减少未压缩的输入视频信号中的冗余。压缩可以帮助减少上述带宽和/或存储空间需求,在某些情况下,可以减少两个数量级或更多。可以采用无损压缩和有损压缩,以及它们的组合。无损压缩是指通过解码过程可以从已压缩的原始信号中重建原始信号的精确副本的技术。有损压缩是指在编码过程中原始视频信息不能完全保留并且在解码过程中不能完全恢复的编码/解码过程。当使用有损压缩时,已重建的信号可能与原始信号不相同,尽管有一些信息损失,但是原始信号和重建的信号之间的失真足够小,以使得已重建的信号可用于预期的应用。在视频的情况下,在很多应用中广泛使用有损压缩。可容忍的失真量取决于应用。例如,某些消费者视频流应用的用户相比于电影或电视广播应用的用户来说可以容忍更高的失真。可以选择或调整特定编码算法可实现的压缩比以反映各种失真容限:更高的可容限失真通常会允许产生更高损耗和更高压缩比的编码算法。

[0007] 视频编码器和解码器可以利用来自若干广泛类别和步骤的技术,包括例如运动补偿、傅里叶变换、量化和熵编码。

[0008] 视频编解码器技术可以包括称为帧内编码的技术。在帧内编码中,在不参考来自先前已重建的参考图片的样本或其他数据的情况下表示样本值。在某些视频编解码器中,图片在空间上细分为样本块。当所有的样本块都以帧内模式编码时,该图片可以是帧内图片。帧内图片及其派生方式(例如独立的解码器刷新图片)可以用于重置解码器状态,并且因此可以用作已编码视频比特流和视频会话中的第一张图片,或者用作静止图像。然后,可以对帧内预测之后的块的样本进行到频域的变换,并且可以在熵编码之前对如此生成的变换系数进行量化。帧内预测表示一种使预变换域中的样本值最小化的技术。在某些情况下,变换后的DC值越小,且AC系数越小,则在给定的量化步长尺寸下就需要越少的比特来表示熵编码后的块。

[0009] 诸如从例如MPEG-2代编码技术已知的传统帧内编码不使用帧内预测。然而,一些较新的视频压缩技术包括基于例如从例如周围样本数据和/或元数据尝试对块进行编码/解码的技术,该周围样本数据和/或元数据是对空间上相邻的数据块进行编码/解码期间获得的、且在解码顺序上先于正在帧内编码或解码的数据块。此类技术此后称为“帧内预测”技术。注意,至少在某些情况下,帧内预测仅使用来自正在重建的当前图片的参考数据,而不使用来自其他参考图片的参考数据。

[0010] 帧内预测可以有許多不同的形式。当在给定的视频编码技术中不止一种这样的技术可用时,所使用的技术可以称之为帧内预测模式。可以在特定编解码器中提供一个或多个帧内预测模式。在某些情况下,模式可以具有子模式和/或可以与各种参数相关联,并且用于视频块的模式/子模式和帧内编码参数可以单独编码或共同地包含在模式码字中。针对给定的模式/子模式/参数组合使用哪个码字可以通过帧内预测对编码效率增益产生影响,且用于将码字转换为比特流的熵编码技术也是如此。

[0011] H.264引入了某种帧内预测模式,在H.265中对其进行了改进,并在诸如联合探索模型(Joint Exploration Model, JEM)、下一代视频编码(Versatile Video Coding, VVC)、基准集(Benchmark Set, BMS)等新的编码技术中进一步进行了改进。通常,对于帧内预测,可以使用已经成为可用的相邻样本值来形成预测器块。例如,可以将沿着特定方向和/或线的特定相邻样本集的可用值复制到预测器块中。对所使用方向的参考可以编码在比特流中,或者可以对其本身进行预测。

[0012] 参考图1A,右下方描绘的是在H.265的33种可能的预测器方向(对应于H.265中指定的35种帧内模式中的33种角度模式)中指定9个预测器方向的子集。箭头会聚的点(101)表示正在被预测的样本。箭头表示使用相邻样本来预测101处的样本的方向。例如,箭头(102)指示样本(101)是根据与水平方向成45度角的右上方向的一个或多个相邻样本来预测的。同样,箭头(103)指示样本(101)是根据与水平方向成22.5度角的样本(101)的左下方方向的一个或多个相邻样本来预测的。

[0013] 仍参考图1A,在左上角描绘了一个4x4个样本的正方形块(104)(由粗体虚线表示)。正方形块(104)包含16个样本,每个样本使用“S”及其在Y维度上的位置(例如,行索引)和其在X维度上的位置(例如列索引)来标记。例如,样本S21是Y维度上(从顶部开始)的第二个样本,以及X维度上(从左侧开始)的第一个样本。类似地,样本S44在Y维度和X维度上都是块(104)中的第四个样本。由于块的大小为4x4个样本,因此S44在右下角。还示出了遵循类似编号方案的示例参考样本。参考样本用R及其相对于块(104)的Y位置(例如行索引)和X位

置(列索引)来标记。在H.264和H.265二者中,使用与正在重建的块邻接相邻的预测样本。

[0014] 块104的帧内图片预测可以通过根据信号通知的预测方向从相邻样本复制参考样本值开始。例如,假设已编码视频比特流包括针对该块104指示箭头(102)的预测方向的信令,也就是说,样本是根据与水平方向成45度角的右上方向的一个或多个预测样本进行预测的。在这种情况下,根据同一个参考样本R05来预测样本S41、S32、S23和S14。然后,根据参考样本R08来预测样本S44。

[0015] 在某些情况下,可以例如通过插值来组合多个参考样本的值,以便计算参考样本;尤其是当方向不能被45度整除时,可以这么做。

[0016] 随着视频编码技术的持续发展,可能的方向的数量已经增加。例如,在H.264(2003年)中,九个不同的方向可用于帧内预测。这一数字在H.265(2013年)增加到了33个,而在本公开时,JEM/VVC/BMS中可支持多达65个方向。已经进行了实验研究以帮助识别最合适的帧内预测方向,并且熵编码中的某些技术可用来以少量的比特来编码那些最合适的方向,对不太可能的方向接受一定的代价。此外,有时可以根据在已经解码的相邻块的帧内预测中使用的相邻方向来预测方向本身。

[0017] 图1B是示出了描绘根据JEM的65个帧内预测方向的示意图(180),以示出了随着时间发展的各种编码技术中不断增加的预测方向的数量。

[0018] 表示预测方向的帧内预测方向比特在已编码视频码流中的映射的方式可以随视频编码技术的不同而不同,并且,例如,它的范围可以从预测方向的简单直接映射到帧内预测模式再到代码字,再到涉及最可能的模式和类似技术的复杂自适应方案。但是,在所有情况下,可能存在某些方向,与某些其他方向相比,在视频内容中统计出现的可能性较小。由于视频压缩的目标是减少冗余,因此,在设计良好的视频编码技术中,那些不太可能出现的方向相比可能出现的方向将由更多数量的比特表示。

[0019] 帧间图片预测或帧间预测可以基于运动补偿。在运动补偿中,来自先前已重建的图片或其部分(参考图片)的样本数据在沿由运动矢量(此后称为MV)指示的方向上空间偏移之后,可用于预测新重建的图片或图片部分(例如,块)。在一些情况下,参考图片可以与当前正在重建的图片相同。MV可以具有X和Y两个维度,或具有三个维度,第三个维度指示正在使用的参考图片(类似于时间维度)。

[0020] 在一些视频压缩技术中,适用于样本数据的某个区域的当前MV可以根据其他MV(例如根据在空间上与正在重建的区域相邻的样本数据的另一个区域相关的、且解码顺序在当前MV之前的那些其他MV)来预测。这样做可以通过依赖于移除相关MV中的冗余来显著减少编码MV所需的总数据量,从而提高压缩率。MV预测可以有效地工作,例如,由于在对从相机导出的输入视频信号(称为自然视频)进行编码时,存在以下统计可能性:比适用单个MV的区域更大的区域在视频序列中以相似的方向移动,因此,在某些情况下,可以使用从相邻区域的MV导出的相似运动矢量来预测该更大的区域。这使得给定区域的实际MV与根据周围MV所预测的MV相似或相同。进而,在熵编码之后,这样的MV可以用比直接编码MV而不是从相邻MV预测MV时使用的比特数更少的比特数来表示。在一些情况下,MV预测可以是无损压缩从原始信号(即样本流)中导出的信号(即MV)的示例。在其他情况下,例如由于根据多个周围MV计算预测值时出现舍入误差,MV预测本身可以是有损的。

[0021] H.265/HEVC(ITU-T H.265建议书,“高效视频编解码(High Efficiency Video

Coding)”,2016年12月)中描述了各种MV预测机制。在H.265指定的多种MV预测机制中,下面描述的是下文称作“空间合并”的技术。

[0022] 具体地,参考图2,当前块(201)包括在运动搜索过程期间已由编码器发现的样本,可以根据已产生空间偏移的相同大小的先前块来预测该样本。代替对MV直接编码,该MV可以使用与标记为A0、A1和B0、B1、B2(分别对应202到206)的五个周围样本中的任一样本关联的MV,从与一个或多个参考图片(例如从(按解码次序)最近的参考图片)相关联的元数据中导出。在H.265中,MV预测可以使用来自相邻块使用的相同参考图片的预测值。

## 发明内容

[0023] 本公开的各方面总体上涉及视频编码和解码,特别地,涉及帧内块复制模式下的视频编码和解码。

[0024] 本公开的各方面提供了一种用于处理视频数据的方法。所述方法包括接收视频比特流,所述视频比特流包括视频帧的至少一个块,所述至少一个块包括当前块;确定所述视频帧中的第一搜索区域,其中,所述第一搜索区域是用于定位帧内块复制(IntraBC)预测块的第一候选区域,其中,所述第一搜索区域与所述当前块不重叠、并且包括块列表,所述IntraBC预测块是用于对所述当前块的至少部分进行IntraBC预测的候选块;确定第二搜索区域,其中,所述第二搜索区域是用于定位所述IntraBC预测块的第二候选区域,所述第二搜索区域包括以下至少之一:(i)所述当前块的子块,以及(ii)所述当前块的相邻块;识别由块矢量参考的所述IntraBC预测块;基于所述IntraBC预测块、所述第一搜索区域和所述第二搜索区域之间的空间关系来确定预测;以及基于所确定的预测对所述当前块的至少部分进行解码。

[0025] 本公开的各方面还提供了一种视频编码或解码设备或装置,包括被配置为执行上述方法实现中的任何一个的电路。

[0026] 本公开的各方面还提供了非暂时性计算机可读介质,存储有指令,当所述指令由计算机执行用于视频解码和/或编码时,使得计算机执行用于视频解码和/或编码的方法。

## 附图说明

[0027] 通过以下详细描述和附图,所公开的主题的其他特征、性质和各种优势将更加明显,在附图中:

[0028] 图1A是帧内预测方向模式示例性子集的示意图;

[0029] 图1B是示例性的帧内预测方向的图示;

[0030] 图2示出了在一个示例中用于运动矢量预测的当前块及其周围空间合并候选的示意图;

[0031] 图3示出了根据示例实施例的通信系统的简化框图的示意图;

[0032] 图4示出了根据示例实施例的通信系统的简化框图的示意图;

[0033] 图5示出了根据示例实施例的视频解码器的简化框图的示意图;

[0034] 图6示出了根据示例实施例的视频编码器的简化框图的示意图;

[0035] 图7示出了根据另一示例实施例的视频编码器的框图;

[0036] 图8示出了根据另一示例实施例的视频解码器的框图;

- [0037] 图9示出了根据本公开的示例实施例的编码块分区的方案；
- [0038] 图10示出了根据本公开的示例实施例的编码块分区的另一方案；
- [0039] 图11示出了根据本公开的示例实施例的编码块分区的另一方案；
- [0040] 图12示出了根据示例分区方案的将基本块分区为编码块的示例；
- [0041] 图13示出了示例性三元分区方案；
- [0042] 图14示出了示例性四叉树二叉树编码块分区方案；
- [0043] 图15示出了根据本公开的示例实施例的用于将编码块分区为多个变换块以及变换块的编码顺序的方案；
- [0044] 图16示出了根据本公开的示例实施例的用于将编码块分区为多个变换块以及变换块的编码顺序的另一方案；
- [0045] 图17示出了根据本公开的示例实施例的用于将编码块分区为多个变换块的另一方案；
- [0046] 图18示出了使用同一帧中的已重建编码块来预测当前编码块的帧内块复制 (IBC) 的概念；
- [0047] 图19示出了可作为IBC的参考样本的示例性已重建样本；
- [0048] 图20示出了具有一些示例限制的情况下可用作IBC的参考样本的示例性已重建样本；
- [0049] 图21示出了用于IBC的示例性片上参考样本存储器 (RSM) 更新机制；
- [0050] 图22示出了图21的示例性片上RSM更新机制的空间视图；
- [0051] 图23示出了用于IBC的另一示例性片上参考样本存储器 (RSM) 更新机制；
- [0052] 图24示出了用于水平分割超级块和垂直分割超级块的IBC的示例性RSM更新机制的空间视图的比较；
- [0053] 图25示出了IBC参考块的示例性非局部和局部搜索区域；
- [0054] 图26示出了采用局部和非局部参考块搜索区域的IBC的示例性预测块和所提出的对预测块选择的限制；
- [0055] 图27示出了根据本公开的示例实施例的方法的流程图；以及
- [0056] 图28示出了根据本公开的示例实施例的计算机系统的示意图。

### 具体实施方式

[0057] 下面将参考附图详细描述本发明,附图是本发明的一部分,并且通过图示的方式示出了实施例的具体示例。然而,请注意,本发明可以以各种不同的形式实施,因此,所涵盖或要求保护的主题旨在被解释为不限于下面阐述的任何实施例。还请注意,本发明可以体现为方法、设备、组件或系统。因此,本发明的实施例可以采取例如硬件、软件、固件或其任意组合的形式。

[0058] 在整个说明书和权利要求书中,术语可以具有在上下文中暗示或隐含的超出明确陈述的含义的细微含义。本文中使用的短语“在一个实施例中”或“在一些实施例中”不一定指相同的实施例,并且本文中使用的短语“在另一个实施例中”或“在其他实施例中”不一定指不同的实施例。同样,这里使用的短语“在一个实施中”或“在一些实施中”不一定指相同的实施,并且这里使用的短语“在另一个实施中”或“在其他实施中”不一定指不同的实施。

例如,所要求保护的主体包括全部或部分示例性实施例/实现的组合。

[0059] 一般来说,术语可以至少部分地从上下文中的用法来理解。例如,这里使用的诸如“和”、“或”或“和/或”的术语可以包括各种含义,这些含义可以至少部分取决于使用这些术语的上下文。通常,“或”如果用于关联诸如A、B或C之类的列表,则意在表示A、B和C(此处用于包含意义)以及A、B或C(此处用于排斥意义)。此外,在此使用的术语“一个或多个”或“至少一个”,至少部分取决于上下文,可用于以单数意义描述任何特征、结构或特征,或可用于以复数意义描述特征、结构或特征的组合。类似地,诸如“一”、“一个”或“该/所述”的术语也可以被理解为传达单数用法或传达复数用法,这至少部分取决于上下文。此外,“基于”或“由…确定”一词可被理解为不一定意在传达一组排他性因素,而是可能允许存在不一定明确描述的其他因素,这也至少部分取决于上下文。

[0060] 图3示出了根据本公开的实施例的通信系统(300)的简化框图。通信系统(300)包括多个终端装置,该终端装置可通过例如网络(350)彼此通信。举例来说,通信系统(300)包括通过网络(350)互连的第一终端装置对(310)和(320)。在图3的示例中,第一终端装置对(310)和(320)可以执行单向数据传输。举例来说,终端装置(310)可对视频数据(例如由终端装置(310)采集的视频图片流)进行编码以通过网络(350)传输到另一终端装置(320)。已编码视频数据以一个或多个已编码视频比特流形式传输。终端装置(320)可从网络(350)接收已编码视频数据,对已编码视频数据进行解码以恢复视频图片,并根据恢复的视频数据 display 视频图片。单向数据传输可以在媒体服务等应用中实现。

[0061] 在另一示例中,通信系统(300)包括执行已编码视频数据的双向传输的第二终端装置对(330)和(340),该双向传输可例如在视频会议应用期间实现。对于双向数据传输,在一示例中,终端装置(330)和终端装置(340)中的每个终端装置可对视频数据(例如由终端装置采集的视频图片流)进行编码,以通过网络(350)传输到终端装置(330)和终端装置(340)中的另一终端装置。终端装置(330)和终端装置(340)中的每个终端装置还可接收由终端装置(330)和终端装置(340)中的另一终端装置传输的已编码视频数据,且可对该已编码视频数据进行解码以恢复视频图片,且可根据恢复的视频数据在可访问的显示装置上显示视频图片。

[0062] 在图3的示例中,终端装置(310)、终端装置(320)、终端装置(330)和终端装置(340)可被实现为服务器、个人计算机和智能电话,但本公开的基本原理的适用性可不限于此。本公开的实施例可以在桌面计算机、膝上型计算机、平板电脑、媒体播放器、可穿戴计算机和/或专用视频会议设备等中实现。网络(350)表示在终端装置(310)、终端装置(320)、终端装置(330)和终端装置(340)之间传送已编码视频数据的任何数目或类型的网络,包括例如有线(连线的)和/或无线通信网络。通信网络(350)可在电路交换分组交换和/或其他类型的信道中交换数据。代表性的网络可包括电信网络、局域网、广域网和/或互联网。出于本讨论的目的,除非本文明确解释,否则网络(350)的架构和拓扑对于本公开的操作来说可能是无关紧要的。

[0063] 作为所公开主题的应用的示例,图4是通信系统(400)的示例性框图。具体地,图4示出了视频编码器和视频解码器在视频流式传输环境中的放置方式。所公开主题可同等地适用于其它视频应用,包括例如视频会议、数字电视广播、游戏、虚拟现实、在包括CD、DVD、存储棒等的数字介质上存储压缩视频等等。

[0064] 视频流式传输系统可包括视频采集子系统(413),该视频采集子系统可包括例如数码相机的视频源(401),用于创建未压缩的视频图片或图像流(402)。在示例中,视频图片流(402)包括由视频源401的数码相机记录的样本。相较于已编码视频数据(404)(或已编码视频比特流),被描绘为粗线以强调高数据量的视频图片流(402)可由电子装置(420)处理,该电子装置(420)包括耦接到视频源(401)的视频编码器(403)。视频编码器(403)可包括硬件、软件或软硬件组合以实现或实施如下文更详细地描述的所公开主题的各方面。相较于未压缩的视频图片流(402),被描绘为细线以强调较低数据量的已编码视频数据(404)(或已编码视频比特流(404))可存储在流式传输服务器(405)上以供将来使用或直接存储到下游视频设备(未示出)中。一个或多个流式传输客户端子系统,例如图4中的客户端子系统(406)和客户端子系统(408),可访问流式传输服务器(405)以检索已编码视频数据(404)的副本(407)和副本(409)。客户端子系统(406)可包括例如电子装置(430)中的视频解码器(410)。视频解码器(410)对已编码视频数据的传入副本(407)进行解码,且产生未被压缩的且可在显示器(412)(例如显示屏)或另一呈现装置(未描绘)上呈现的输出视频图片流(411)。视频解码器410可以被配置为执行本公开中描述的各种功能中的一些或全部。在一些流式传输系统中,可根据某些视频编码/压缩标准对已编码视频数据(404)、(407)和(409)(例如视频比特流)进行编码。这些标准的示例包括ITU-TH.265建议书。在示例中,正在开发的视频编码标准非正式地称为下一代视频编码(VersatileVideoCoding,VVC),所公开的主题可用于VVC和其他视频编码标准的上下文中。

[0065] 应注意,电子装置(420)和电子装置(430)可包括其它组件(未示出)。举例来说,电子装置(420)可包括视频解码器(未示出),且电子装置(430)还可包括视频编码器(未示出)。

[0066] 图5是根据本公开的下述任何实施例的视频解码器(510)的框图。视频解码器(510)可包括在电子装置(530)中。电子装置(530)可包括接收器(531)(例如接收电路)。视频解码器(510)可用于代替图4的示例中的视频解码器(510)。

[0067] 接收器(531)可接收将由视频解码器(510)解码的一个或多个已编码视频序列。在同一实施例或另一实施例中,一次解码一个已编码视频序列,其中每个已编码视频序列的解码独立于其它已编码视频序列。每个视频序列可以与多个视频帧或图像相关联。可从信道(501)接收已编码视频序列,该信道可以是通向存储已编码视频数据的存储装置或者是发送已编码视频数据的流式传输源的硬件/软件链路。接收器(531)可接收可转发到它们各自的处理电路(未描绘)的已编码视频数据以及其它数据,例如,已编码音频数据和/或辅助数据流。接收器(531)可将已编码视频序列与其它数据分开。为了防止网络抖动,缓冲存储器(515)可设置在接收器(531)与熵解码器/解析器(520)(此后称为“解析器(520)”)之间。在某些应用中,缓冲存储器(515)可以实现为视频解码器(510)的一部分。在其他应用中,该缓冲存储器(515)可设置在视频解码器(510)之外并与其分离(未示出)。在其它应用中,在视频解码器(510)的外部可设置缓冲存储器(未描绘)以例如防止网络抖动,且在视频解码器(510)的内部可配置另一附加的缓冲存储器(515)以例如处理播放定时。当接收器(531)从具有足够带宽和可控性的存储/转发装置或从等时同步网络接收数据时,也可能不需要配置缓冲存储器(515),或可以将该缓冲存储器做得较小。为了在诸如互联网之类的尽力服务的分组网络上使用,也可能需要足够大小的缓冲存储器(515),该缓冲存储器的大小可相

对较大。这种缓冲存储器可以以自适应性大小来实现,且可至少部分地实施于操作系统或视频解码器(510)外部的类似元件(未描绘)中。

[0068] 视频解码器(510)可包括解析器(520)以根据已编码视频序列重建符号(521)。这些符号的类别包括用于管理视频解码器(510)的操作的信息,以及用以控制诸如显示器(512)(例如,显示屏)等显示装置的潜在信息,该显示装置可以是或不是电子装置(530)的整体部分,但可耦接到电子装置(530),如图5中所示。用于显示装置的控制信息可以是辅助增强信息(Supplemental Enhancement Information,SEI消息)或视频可用性信息(Video Usability Information,VUI)的参数集片段(未示出)。解析器(520)可对由解析器(520)接收到的已编码视频序列进行解析/熵解码。已编码视频序列的熵编码可根据视频编码技术或标准进行,且可遵循各种原理,包括可变长度编码、霍夫曼编码(Huffman coding)、具有或不具有上下文灵敏度的算术编码等等。解析器(520)可基于对应于子群的至少一个参数,从已编码视频序列提取用于视频解码器中的像素的子群中的至少一个子群的子群参数集。子群可包括图片群组(Group of Pictures,GOP)、图片、图块、切片、宏块、编码单元(Coding Unit,CU)、块、变换单元(Transform Unit,TU)、预测单元(Prediction Unit,PU)等等。解析器(520)还可从已编码视频序列提取信息,例如变换系数(例如,傅里叶变换系数)、量化器参数值、运动矢量等等。

[0069] 解析器(520)可对从缓冲存储器(515)接收的视频序列执行熵解码/解析操作,从而创建符号(521)。

[0070] 取决于已编码视频图片或一部分已编码视频图片(例如:帧间图片和帧内图片、帧间块和帧内块)的类型以及其它因素,符号(521)的重建可涉及多个不同处理或功能单元。涉及的单元以及涉及方式可由解析器(520)从已编码视频序列解析的子群控制信息来控制。为了简洁起见,未描述解析器(520)与下文的多个处理或功能单元之间的此类子群控制信息流。

[0071] 除已经提及的功能块以外,视频解码器(510)可在概念上细分成如下文所描述的数个功能单元。在商业约束下运行的实际实施例中,这些功能单元中的许多单元彼此紧密交互并且可以至少部分地彼此集成。然而,出于清楚地描述所公开主题的各种功能的目的,本公开在下文中采用概念上细分成功能单元。

[0072] 第一单元可包括缩放器/逆变换单元(551)。缩放器/逆变换单元(551)可从解析器(520)接收作为符号(521)的量化变换系数以及控制信息,包括指示使用哪种类型的逆变换、块大小、量化因子/参数、量化定标矩阵等的信息。缩放器/逆变换单元(551)可输出包括样本值的块,该样本值可输入到聚合器(555)中。

[0073] 在一些情况下,缩放器/逆变换单元(551)的输出样本可属于帧内编码块,即,不使用来自先前已重建的图像的预测性信息,但可使用来自当前图像的先前已重建部分的预测性信息的块。此类预测性信息可由帧内图片预测单元(552)提供。在一些情况下,帧内图片预测单元(552)可以采用已经重建且存储在当前图片缓冲器(558)中的周围块信息生成大小和形状与正在重建的块相同的块。举例来说,当前图片缓冲器(558)缓冲部分重建的当前图片和/或完全重建的当前图片。在一些实施中,聚合器(555)可以基于每个样本,将帧内预测单元(552)生成的预测信息添加到由缩放器/逆变换单元(551)提供的输出样本信息中。

[0074] 在其它情况下,缩放器/逆变换单元(551)的输出样本可属于帧间编码的和潜在运

动补偿的块。在此情况下,运动补偿预测单元(553)可访问参考图片存储器(557)以提取用于帧内图片预测的样本。在根据属于块的符号(521)对提取的样本进行运动补偿之后,这些样本可由聚合器(555)添加到缩放器/逆变换单元(551)的输出(单元551的输出被称作残差样本或残差信号),从而生成输出样本信息。运动补偿预测单元(553)从中提取预测样本的参考图片存储器(557)内的地址受运动矢量控制,且该运动矢量以符号(521)的形式而供运动补偿预测单元(553)使用,该符号(521)可以具有例如X、Y分量(偏移)和参考图片分量(时间)。运动补偿还可包括在使用子样本精确运动矢量时,从参考图片存储器(557)提取的样本值的内插,并且还可以与运动矢量预测机制等相关联。

[0075] 聚合器(555)的输出样本可经受环路滤波器单元(556)中的各种环路滤波技术。视频压缩技术可包括环路内滤波器技术,该环路内滤波器技术受控于包括在已编码视频序列(也称作已编码视频比特流)中并且作为来自解析器(520)的符号(521)可用于环路滤波器单元(556)的参数,然而,视频压缩技术还可响应于在解码已编码图片或已编码视频序列的先前(按解码次序)部分期间获得的元信息,以及响应于先前已重建且经过环路滤波的样本值。几种类型的环路滤波器可以以各种顺序被包括作为环路滤波器单元556的一部分,如下面将进一步详细描述。

[0076] 环路滤波器单元(556)的输出可以是样本流,该样本流可输出到显示装置(512)以及存储在参考图片存储器(557)中,以用于后续的帧间图片预测。

[0077] 一旦完全重建,某些已编码图片就可用作参考图片以用于将来帧间图片预测。举例来说,一旦对应于当前图片的已编码图片被完全重建,且已编码图片(通过例如解析器(520))被识别为参考图片,则当前图片缓冲器(558)可变为参考图片存储器(557)的一部分,且可在开始重建后续已编码图片之前重新分配新的当前图片缓冲器。

[0078] 视频解码器(510)可根据例如ITU-T H.265建议书标准中采用的预定视频压缩技术执行解码操作。在已编码视频序列遵循视频压缩技术或标准的语法以及视频压缩技术或标准中记录的配置文件的意义下,已编码视频序列可符合所使用的视频压缩技术或标准指定的语法。具体地说,配置文件可从视频压缩技术或标准中可用的所有工具中选择某些工具作为在该配置文件下可供使用的仅有工具。对于合规性,还要求已编码视频序列的复杂度处于视频压缩技术或标准的层级所限定的范围内。在一些情况下,层级限制最大图片大小、最大帧率、最大重建取样率(以例如每秒兆(mega)个样本为单位进行测量)、最大参考图片大小等。在一些情况下,由层级设定的限制可通过假想参考解码器(Hypothetical Reference Decoder,HRD)规范和在已编码视频序列中用信号通知的HRD缓冲器管理的元数据来进一步限定。

[0079] 在一些示例实施例中,接收器(531)可连同已编码视频一起接收附加(冗余)数据。该附加数据可以被包括作为已编码视频序列的一部分。该附加数据可由视频解码器(510)用以对数据进行适当解码和/或较准确地重建原始视频数据。附加数据可为例如时间、空间或信噪比(signal noise ratio,SNR)增强层、冗余切片、冗余图片、前向纠错码等形式。

[0080] 图6示出了根据本公开的示例实施例的视频编码器(603)的框图。视频编码器(603)可包括在电子装置(620)中。电子装置(620)还可包括传输器(640)(例如传输电路)。视频编码器(603)可用于代替图4的示例中的视频编码器(603)。

[0081] 视频编码器(603)可从视频源(601)(并非图6示例中的电子装置(620)的一部分)

接收视频样本,该视频源可采集将由视频编码器(603)编码的视频图像。在另一示例中,视频源(601)可以实现为电子装置(620)的一部分。

[0082] 视频源(601)可提供将由视频编码器(603)编码的呈数字视频样本流形式的源视频序列,该数字视频样本流可具有任何合适位深度(例如:8位、10位、12位……)、任何色彩空间(例如BT.601YCrCb、RGB、XYZ……)和任何合适采样结构(例如YCrCb 4:2:0、YCrCb 4:4:4)。在媒体服务系统中,视频源(601)可以是能够存储先前已准备的视频的存储装置。在视频会议系统中,视频源(601)可以是采集本地图像信息作为视频序列的相机。可将视频数据提供为多个单独的图片或图像,当按顺序观看时,这些图片或图像被赋予运动。图片自身可构造为空间像素阵列,其中取决于所用的取样结构、色彩空间等,每个像素可包括一个或多个样本。所属领域的技术人员可以很容易理解像素与样本之间的关系。下文侧重于描述样本。

[0083] 根据一些示例实施例,视频编码器(603)可实时或在由应用所要求的任何其它时间约束下,将源视频序列的图片编码且压缩成已编码视频序列(643)。施行适当的编码速度构成了控制器(650)的一个功能。在一些实施例中,控制器(650)可以在功能上耦接到如下文所描述的其它功能单元且控制所述其它功能单元。为了简洁起见,图中未标示耦接。由控制器(650)设置的参数可包括速率控制相关参数(图片跳过、量化器、率失真优化技术的 $\lambda$ 值……)、图片大小、图片群组(GOP)布局、最大运动矢量搜索范围等。控制器(650)可被配置成具有其它合适的功能,这些功能涉及针对某一系统设计优化的视频编码器(603)。

[0084] 在一些示例实施例中,视频编码器(603)可被配置成在编码环路中进行操作。作为简单的描述,在示例中,编码环路可包括源编码器(630)(例如,负责基于待编码的输入图片和参考图片创建符号,例如符号流)和嵌入于视频编码器(603)中的(本地)解码器(633)。解码器(633)以类似于(远程)解码器创建样本数据的方式来重建符号以创建样本数据,即使嵌入的解码器633处理由源编码器630在没有熵编码的情况下的已编码视频流(因为在所公开主题所考虑的视频压缩技术中,熵编码中的符号与已编码视频比特流之间的任何压缩可以是无损的)。将重建的样本流(样本数据)输入到参考图片存储器(634)。由于符号流的解码产生与解码器位置(本地或远程)无关的位精确结果,因此参考图片存储器(634)中的内容在本地编码器与远程编码器之间也是按比特位精确对应的。换句话说,编码器的预测部分“看到”的参考图片样本与解码器将在解码期间使用预测时所“看到”的样本值完全相同。这种参考图片同步性的基本原理(以及在例如因信道误差而无法维持同步性的情况下产生的漂移)被用于改进编码质量。

[0085] “本地”解码器(633)的操作可与例如已在上文结合图5详细描述的视频解码器(510)的“远程”解码器相同。然而,另外简要参考图5,当符号可用且熵编码器(645)和解析器(520)能够无损地将符号编码/解码为已编码视频序列时,包括缓冲存储器(515)和解析器(520)在内的视频解码器(510)的熵解码部分,可能无法完全在编码器中的本地解码器(633)中实施。

[0086] 此时可以观察到,除可能仅存在于解码器中的解析/熵解码之外的任何解码器技术也可能必定需要以基本上相同的功能形式存在于对应的编码器中。出于此原因,所公开主题有时可能集中在解码器操作,其与编码侧的解码部分相关。从而可简化编码器技术的描述,因为编码器技术与全面地描述的解码器技术互逆。以下仅在某些区域或方面提供了

编码器的更详细的描述。

[0087] 在操作期间,在一些示例实现中,源编码器(630)可执行运动补偿预测编码。参考来自视频序列中被指定为“参考图片”的一个或多个先前已编码图片,该运动补偿预测编码对输入图片进行预测性编码。以此方式,编码引擎(632)对输入图片的像素块与参考图片的像素块之间的颜色通道差异(或者残差)进行编码,该参考图片可被选作该输入图片的预测参考。术语“残差”及其形容词形式“残差的”可以互换使用。

[0088] 本地视频解码器(633)可基于源编码器(630)创建的符号,对可指定为参考图片的图片的已编码视频数据进行解码。编码引擎(632)的操作可有利地为有损过程。当已编码视频数据可在视频解码器(图6中未示出)处被解码时,重建的视频序列通常可以是带有一些误差的源视频序列的副本。本地视频解码器(633)复制解码过程,该解码过程可由视频解码器对参考图片执行,且可使重建的参考图片存储在参考图片存储器(634)中。以此方式,视频编码器(603)可在本地存储重建的参考图片的副本,该副本与将由远端视频解码器获得的重建的参考图片具有共同内容(不存在传输误差)。

[0089] 预测器(635)可针对编码引擎(632)执行预测搜索。即,对于将要编码的新图片,预测器(635)可在参考图片存储器(634)中搜索可作为该新图片的适当预测参考的样本数据(作为候选参考像素块)或某些元数据,例如参考图片运动矢量、块形状等。预测器(635)可基于样本块逐像素块操作,以找到合适的预测参考。在一些情况下,如由预测器(635)获得的搜索结果所确定的那样,输入图片可具有从参考图片存储器(634)中存储的多个参考图片取得的预测参考。

[0090] 控制器(650)可管理源编码器(630)的编码操作,包括例如设置用于对视频数据进行编码的参数和子群参数。

[0091] 可在熵编码器(645)中对所有上述功能单元的输出进行熵编码。熵编码器(645)根据例如霍夫曼编码、可变长度编码、算术编码等技术对各种功能单元生成的符号进行无损压缩,从而将该符号变换成已编码视频序列。

[0092] 传输器(640)可缓冲由熵编码器(645)创建的已编码视频序列,从而为通过通信信道(660)进行传输做准备,该通信信道可以是通向将存储已编码视频数据的存储装置的硬件/软件链路。传输器(640)可将来自视频编码器(603)的已编码视频数据与要传输的其它数据合并,该其它数据例如是已编码音频数据和/或辅助数据流(未示出来源)。

[0093] 控制器(650)可管理视频编码器(603)的操作。在编码期间,控制器(650)可以为每个已编码图片分配某一已编码图片类型,但这可能影响可应用于相应的图片的编码技术。例如,通常可将图片分配为以下任一种图片类型:

[0094] 帧内图片(I图片),其可以是不将序列中的任何其它图片用作预测源就可被编码和解码的图片。一些视频编解码器容许不同类型的帧内图片,包括例如独立解码器刷新(Independent Decoder Refresh,“IDR”)图片。所属领域的普通技术人员了解I图片的变体及其相应的应用和特征。

[0095] 预测性图片(P图片),其可以是可使用帧内预测或帧间预测进行编码和解码的图片,该帧内预测或帧间预测使用至多一个运动矢量和参考索引来预测每个块的样本值。

[0096] 双向预测性图片(B图片),其可以是可使用帧内预测或帧间预测进行编码和解码的图片,该帧内预测或帧间预测使用至多两个运动矢量和参考索引来预测每个块的样本

值。类似地,多个预测性图片可使用多于两个参考图片和相关联元数据以用于重建单个块。

[0097] 源图片通常可在空间上细分成多个样本编码块(例如,  $4 \times 4$ 、 $8 \times 8$ 、 $4 \times 8$  或  $16 \times 16$  个样本的块),且逐块进行编码。这些块可参考其它(已编码)块进行预测性编码,该其它块由应用于块的相应图片的编码分配来确定。举例来说,I图片的块可进行非预测性编码,或该块可参考同一图片的已编码的块来进行预测性编码(空间预测或帧内预测)。P图片的像素块可参考一个先前编码的参考图片通过空间预测或通过时间预测进行预测性编码。B图片的块可参考一个或两个先前编码的参考图片通过空间预测或通过时间预测进行预测性编码。为了其他目的,源图片或中间处理图片可以被细分成其他类型的块。编码块和其他类型的块的划分可以遵循或不遵循相同的方式,见下面进一步的详细描述。

[0098] 视频编码器(603)可根据例如ITU-T H.265建议书的预定视频编码技术或标准执行编码操作。在操作中,视频编码器(603)可执行各种压缩操作,包括利用输入视频序列中的时间和空间冗余的预测编码操作。因此,已编码视频数据可符合所使用的视频编码技术或标准指定的语法。

[0099] 在一些示例性实施例中,传输器(640)可在传输已编码视频时传输附加数据。源编码器(630)可包括此类数据作为已编码视频序列的一部分。附加数据可包括时间/空间/SNR增强层、冗余图片和切片等其它形式的冗余数据、SEI消息、VUI参数集片段等。

[0100] 采集到的视频可作为呈时间序列的多个源图片(视频图片)。帧内图片预测(常常简化为帧内预测)利用给定图片中的空间相关性,而帧间图片预测则利用图片之间的时间或其它相关性。例如,将正在编码/解码的特定图片分割成块,正在编码/解码的特定图片被称作当前图片。在当前图片中的块类似于视频中先前已编码且仍被缓冲的参考图片中的参考块时,可通过称作运动矢量的矢量对当前图片中的块进行编码。该运动矢量指向参考图片中的参考块,且在使用多个参考图片的情况下,该运动矢量可具有识别参考图片的第三维度。

[0101] 在一些示例实施例中,双向预测技术可用于帧间图片预测中。根据双向预测技术,使用两个参考图片,例如按解码次序都在视频中的当前图片之前(但按显示次序可能分别是过去和将来)的第一参考图片和第二参考图片。可通过指向第一参考图片中的第一参考块的第一运动矢量和指向第二参考图片中的第二参考块的第二运动矢量对当前图片中的块进行编码。可通过第一参考块和第二参考块的组合来联合预测该块。

[0102] 此外,合并模式技术可用于帧间图片预测中以改善编码效率。

[0103] 根据本公开的一些示例实施例,例如帧间图片预测和帧内图片预测的预测以块为单位执行。例如,将视频图片序列中的图片分区为编码树单元(coding tree unit,CTU)以用于压缩,图片中的CTU可具有相同大小,例如  $128 \times 128$  像素、 $64 \times 64$  像素、 $32 \times 32$  像素或  $16 \times 16$  像素。通常,CTU可以包括三个并行编码树块(CTB):一个亮度CTB和两个色度CTB。可将每个CTU递归地二叉树拆分为一个或多个编码单元(coding unit,CU)。例如,可将  $64 \times 64$  像素的CTU拆分为一个  $64 \times 64$  像素的CU,或4个  $32 \times 32$  像素的CU。 $32 \times 32$  块中的一个或多个块中的每一个可以被进一步拆分成  $16 \times 16$  像素的4个CU。在一些示例性实施例中,可以在编码期间分析每个CU以在各种预测类型(例如帧间预测类型或帧内预测类型)中确定用于CU的预测类型。取决于时间和/或空间可预测性,可以将CU拆分为一个或多个预测单元(prediction unit,PU)。通常,每个PU包括一个亮度预测块(prediction block,PB)和两个

色度PB。在一个实施例中,编解码(编码/解码)中的预测操作以预测块为单位来执行。可以以各种空间模式执行将CU拆分成PU(或不同颜色通道的PB)。例如,亮度或色度PB可以包括样本的值(例如,亮度值)的矩阵,例如 $8 \times 8$ 像素、 $16 \times 16$ 像素、 $8 \times 16$ 像素、 $16 \times 8$ 像素等。

[0104] 图7示出了根据本公开的另一示例实施例的视频编码器(703)的图。视频编码器(703)被配置成接收视频图片序列中的当前视频图片内的样本值的处理块(例如预测块),且将该处理块编码到作为已编码视频序列的一部分的已编码图片中。示例的视频编码器(703)可以用于代替图4示例中的视频编码器(403)。

[0105] 例如,视频编码器(703)接收用于处理块的样本值的矩阵,该处理块为例如 $8 \times 8$ 样本的预测块等。然后视频编码器(703)使用例如率失真优化(rate-distortion optimization, RDO)来确定是否使用帧内模式、帧间模式或双向预测模式来最佳地编码该处理块。当确定在帧内模式中编码处理块时,视频编码器(703)可使用帧内预测技术以将该处理块编码到已编码图片中;且当确定在帧间模式或双向预测模式中编码处理块时,视频编码器(703)可分别使用帧间预测或双向预测技术将该处理块编码到已编码图片中。在一些示例实施例中,合并模式可以用作帧间图片预测的子模式,其中,在不借助预测器外部的已编码运动矢量分量的情况下,从一个或多个运动矢量预测器导出运动矢量。在一些其他示例实施例中,可存在适用于主题块的运动矢量分量。因此,视频编码器(703)可以包括图7中未明确示出的组件,例如用于确定处理块的预测模式的模式决策模块。

[0106] 在图7的示例中,视频编码器(703)包括如图7示例性布置中所示的耦接到一起的帧间编码器(730)、帧内编码器(722)、残差计算器(723)、开关(726)、残差编码器(724)、通用控制器(721)和熵编码器(725)。

[0107] 帧间编码器(730)被配置成接收当前块(例如处理块)的样本、比较该块与参考图片中的一个或多个参考块(例如按显示顺序在先前图片和后来图片中的块)、生成帧间预测信息(例如根据帧间编码技术的冗余信息描述、运动矢量、合并模式信息)、以及基于帧间预测信息使用任何合适的技术计算帧间预测结果(例如已预测块)。在一些示例中,参考图片是使用嵌入在图6的示例编码器620中的解码单元633(如图7的残差解码器728所示,如下面进一步详细描述)的基于已编码视频信息解码的已解码参考图片。

[0108] 帧内编码器(722)被配置成接收当前块(例如处理块)的样本、比较该块与同一图片中已编码的块、在变换之后生成量化系数、以及在一些情况下还生成帧内预测信息(例如根据一个或多个帧内编码技术的帧内预测方向信息)。帧内编码器(722)可以基于帧内预测信息和同一图片中的参考块计算帧内预测结果(例如已预测块)。

[0109] 通用控制器(721)可以被配置成确定通用控制数据,且基于该通用控制数据控制视频编码器(703)的其它组件。在示例中,通用控制器(721)确定块的预测模式,且基于该预测模式将控制信号提供到开关(726)。举例来说,当该预测模式是帧内模式时,通用控制器(721)控制开关(726)以选择供残差计算器(723)使用的帧内模式结果,且控制熵编码器(725)以选择帧内预测信息且将该帧内预测信息添加在比特流中;以及当用于块的该预测模式是帧间模式时,通用控制器(721)控制开关(726)以选择供残差计算器(723)使用的帧间预测结果,且控制熵编码器(725)以选择帧间预测信息且将该帧间预测信息添加在比特流中。

[0110] 残差计算器(723)可以被配置成计算所接收的块与选自帧内编码器(722)或帧间

编码器(730)的块的预测结果之间的差(残差数据)。残差编码器(724)可以被配置成对残差数据进行编码以生成变换系数。例如,残差编码器(724)可以被配置成将残差数据从空间域变换至频域,以生成变换系数。变换系数接着经受量化处理以获得量化的变换系数。在各种示例实施例中,视频编码器(703)还包括残差解码器(728)。残差解码器(728)被配置成执行逆变换,且生成已解码残差数据。已解码残差数据可适当地由帧内编码器(722)和帧间编码器(730)使用。举例来说,帧间编码器(730)可基于已解码残差数据和帧间预测信息生成已解码块,且帧内编码器(722)可基于已解码残差数据和帧内预测信息生成已解码块。适当处理已解码块以生成已解码图片,且该已解码图片可在存储器电路(未示出)中缓冲并用作参考图片。

[0111] 熵编码器(725)可以被配置成将比特流格式化以包括已编码块且执行熵编码。熵编码器(725)被配置成在比特流中包括各种信息。例如,熵编码器(725)可以被配置成将通用控制数据、所选预测信息(例如帧内预测信息或帧间预测信息)、残差信息和其它合适的信息包括在比特流中。当在帧间模式或双向预测模式的合并子模式中对块进行编码时,不存在残差信息。

[0112] 图8示出了根据本公开的另一实施例的示例视频解码器(810)的图。视频解码器(810)被配置成接收作为已编码视频序列的一部分的已编码图片,且对该已编码图片进行解码以生成重建的图片。在示例中,视频解码器(810)可以用于代替图4的示例中的视频解码器(410)。

[0113] 在图8的示例中,视频解码器(810)包括如图8的示例布置中所示的耦接到一起的熵解码器(871)、帧间解码器(880)、残差解码器(873)、重建模块(874)和帧内解码器(872)。

[0114] 熵解码器(871)可被配置成根据已编码图片来重建某些符号,这些符号表示构成该已编码图片的语法元素。此类符号可包括例如用于对该块进行编码的模式(例如,帧内模式、帧间模式、双向预测模式、合并子模式或其他子模式)、可识别供帧内解码器(872)或帧间解码器(880)用以进行预测的某些样本或元数据的预测信息(例如帧内预测信息或帧间预测信息)、呈例如量化的变换系数形式的残差信息等等。在示例中,当预测模式是帧间或双向预测模式时,将帧间预测信息提供到帧间解码器(880);以及当预测类型是帧内预测类型时,将帧内预测信息提供到帧内解码器(872)。残差信息可经受逆量化并提供到残差解码器(873)。

[0115] 帧间解码器(880)可以被配置成接收帧间预测信息,且基于该帧间预测信息生成帧间预测结果。

[0116] 帧内解码器(872)可以被配置成接收帧内预测信息,且基于该帧内预测信息生成预测结果。

[0117] 残差解码器(873)可以被配置成执行逆量化以提取解量化的变换系数,且处理该解量化的变换系数,以将残差从频域变换到空间域。残差解码器(873)还可能利用某些控制信息(用以包括量化器参数QP),且该信息可由熵解码器(871)提供(未标示数据路径,因为这仅仅是低数据量控制信息)。

[0118] 重建模块(874)可以被配置成在空间域中组合由残差解码器(873)输出的残差与预测结果(根据具体情况,可由帧间预测模块或帧内预测模块输出)以形成重建的块,该重建的块形成了重建的图片的一部分,该重建的图片继而作为重建的视频的一部分。应注意,

也可执行诸如解块操作等其它合适的操作来改善视觉质量。

[0119] 应注意,可使用任何合适的技术来实施视频编码器(403)、(603)和(703)以及视频解码器(410)、(510)和(810)。在一些示例实施例中,可使用一个或多个集成电路来实施视频编码器(403)、(603)和(703)以及视频解码器(410)、(510)和(810)。在另一实施例中,可使用执行软件指令的一个或多个处理器来实施视频编码器(403)、(603)和(703)以及视频解码器(410)、(510)和(810)。

[0120] 转向用于编码和解码的块分区,一般分区可以从基本块开始,并且可以遵循预定义的规则集、特定模式、分区树或任何分区结构或方案。分区可以是分层的和递归的。在遵循下面描述的任何示例分区过程或其他过程或其组合来对基本块进行划分或分区(partitioning)之后,可以获得分区(partition)或编码块的最终集合。这些分区中的每一个可以处于分区层次结构中的各种分区级别中的一个,并且可以具有各种形状。每个分区可以被称为编码块(CB)。对于下面进一步描述的各种示例分区实施,每个结果CB可以是任意允许的大小和分区级别。这样的分区被称为编码块,因为它们可以形成可以对其做出一些基本的编码/解码决策的单元,并且可以在已编码的视频比特流中优化、确定和用信号发送编码/解码参数。最终分区中的最高或最深级别代表树的编码块分区结构的深度。编码块可以是亮度编码块或色度编码块。每种颜色的CB树结构可以被称为编码块树(CBT)。

[0121] 所有颜色通道的编码块可以统称为编码单元(CU)。所有颜色通道的分层结构可以统称为编码树单元(CTU)。CTU中各种颜色通道的分区模式或结构可能相同,也可能不同。

[0122] 在一些实施中,用于亮度和色度通道的分区树方案或结构可能不需要相同。换句话说,亮度和色度通道可以具有单独的编码树结构或模式。此外,亮度和色度通道是否使用相同或不同的编码分区树结构以及实际使用的编码分区树结构可以取决于被编码的切片是P、B还是I切片。例如,对于I切片,色度通道和亮度通道可以具有单独的编码分区树结构或编码分区树结构模式,而对于P或B切片,亮度和色度通道可以共享相同的编码分区树方案。当应用单独的编码分区树结构或模式时,亮度通道可以通过一个编码分区树结构被分区成CBs,色度通道可以通过另一个编码分区树结构被分区成色度CBs。

[0123] 在一些示例实现中,预定的分区模式可以应用于基本块。如图9所示,示例的4路分区树可以从第一预定义级别(例如,64x64块级别或其他大小,作为基本块大小)开始,并且基本块可以被分层地向下分区到预定义的最低级别(例如,4x4级别)。例如,基本块可以受902、904、906和908指示的四个预定义分区选项或模式的影响,其中被指定为R的分区可以用于递归分区,因为如图9所示的相同分区选项可以以较低的规模重复直到最低级别(例如,4x4级别)。在一些实施中,额外的限制可以应用于图9的分区方案。在图9的实施中,可以是矩形分区(例如,1:2/2:1矩形分区),但是这些矩形分区不可以是递归的,而正方形分区可以是递归的。如果需要的话,继图9的递归分区会生成最终的一组编码块。可以进一步定义编码树深度以指示来自根节点或根块的拆分深度。例如,根节点或根块(例如,64x64块)的编码树深度可以设置为0,并且在根块根据图9被进一步拆分一次之后,编码树深度增加1。对于上述方案,从64x64基本块到4x4的最小分区的最大或最深级别将是4(从级别0开始)。这种分区方案可以应用于一个或多个颜色通道。每个颜色通道可以按照图9的方案独立地进行分区(例如,对于每个分层级别的每个颜色通道,可以独立地确定预定义模式(pattern)中的分区模式或选项)。可替换地,两个或多个颜色通道可以共享图9的相同分

层模式树(例如,可以为每个分层级别的两个或更多个颜色通道选择预定义模式中的相同分区模式或选项)。

[0124] 图10示出了可以递归分区以形成分区树的另一个示例预定义分区模式。如图10所示,可以预定义示例的10路分区结构或模式。根块可以从预定义的级别开始(例如,从 $128 \times 128$ 级别或 $64 \times 64$ 级别的基本块开始)。图10的示例分区结构包括各种 $2:1/1:2$ 和 $4:1/1:4$ 矩形分区。图10的第二行中表示为1002、1004、1006和1008的具有3个子分区的分区类型可以称为“T型”分区。“T型”分区1002、1004、1006和1008可以被称为左T型、上T型、右T型和下T型。在一些示例实现中,不允许进一步细分图10的矩形分区中的任何一个。可以进一步定义编码树深度以指示来自根节点或根块的拆分深度。例如,根节点或根块(例如, $128 \times 128$ 块)的编码树深度可以设置为0,并且在根块根据图10被进一步拆分一次之后,编码树深度增加1。在一些实施中,只有1010中的全正方形分区可以按照图10的模式递归地分区到分区树的下一级。换句话说,对于T型模式1002、1004、1006和1008内的正方形分区,可能不允许进行递归分区。如果需要的话,继图10的递归分区过程会生成最终的一组编码块。这种方案可以应用于一个或多个颜色通道。在一些实施中,可以使低于 $8 \times 8$ 级别的分区的使用更加灵活。例如,在某些情况下可以使用 $2 \times 2$ 色度帧间预测。

[0125] 在用于编码块分区的一些其它实施中,四叉树结构可用于将基本块或中间块拆分成四叉树分区。这种四叉树拆分可以分层地和递归地应用于任何正方形分区。基本块、中间块或分区是否可以进一步的四叉树拆分可以适用于基本块、中间块或分区的各种局部特性。可以进一步调整在图片边界处的四叉树分区。例如,可以在图片边界处执行隐式四叉树拆分,以使块保持四叉树拆分直到大小适合图片边界。

[0126] 在一些其他示例实施中,可以使用来自基本块的分层二元分区。对于这样的方案,基本块或中间级块可以被分成两个分区。二元分区可以是水平的,也可以是垂直的。例如,水平二元分区可以将基本块或中间块拆分成相等的右分区和左分区。同样,垂直二元分区可以将基本块或中间块拆分成相等的上分区和下分区。这种二元分区可以是分层的和递归的。可以在基本块或中间块中的每一个处决定是否应当继续二元分区方案,并且如果确定进一步实施该方案,则应当使用水平还是垂直二元分区。在一些实施中,可以在预定义的最低分区大小处(在一个维度或两个维度中)停止进一步的分区。可替换地,一旦达到来自基本块的预定义分区级别或深度,则可以停止进一步的分区。在一些实施中,分区的纵横比(aspect ratio)可以被限制。例如,分区的纵横比可以不小于 $1:4$ (或大于 $4:1$ )。因此,垂直与水平纵横比为 $4:1$ 的垂直条带分区只能进一步被垂直地二元分区为上分区和下分区,每个分区的垂直与水平纵横比为 $2:1$ 。

[0127] 在其他一些例子中,三元分区方案可以用于对基本块或任何中间块进行分区,如图13所示。三元模式可以如图13的1302所示的垂直实现,或如图13的1304所示的水平实现。虽然无论是垂直还是水平,图13中的示例拆分比率都显示为 $1:2:1$ ,但也可以预先定义其他比率。在一些实施中,可以预定义两个或更多个不同的比率。这种三元分区方案可以用于补充四叉树或二元分区结构,因为这种三叉树分区能够在连续分区中捕获位于块中心的对象,而四叉树和二叉树总是沿着块中心拆分,从而将对象分割成单独的分区。在一些实施中,为了避免额外的变换,示例三叉树的分区的宽度和高度总是2的幂。

[0128] 上述分区方案可以在不同的分区级别上以任何方式组合。作为一个示例,可以将

上述四叉树和二元分区方案组合以将基本块分区为四叉树-二叉树(quadtree-binary-tree, QTBT)结构。在这样的方案中,基本块或中间块/分区可以是四叉树拆分或二元拆分,如果指定的话,服从一组预定义的条件。一个特定的例子如图14所示。在图14的例子中,基本块首先被四叉树拆分成四个分区,如1402、1404、1406和1408所示。此后,所得到的分区中的每一个要么被四叉树拆分为四个分区(例如1408),要么在下一级被二元拆分为两个进一步的分区(例如水平地或垂直地,例如1402或1406,两者都是对称的),要么不被拆分(例如1404)。对于正方形分区,可以递归地进行二元或四叉树拆分,如1410的整体示例分区模式和1420中的相应树结构/表示所示,其中实线表示四叉树拆分,虚线表示二元拆分。可以为每个二元拆分节点(非叶二元分区)使用标志来指示二元拆分是水平的还是垂直的。例如,如1420所示,与1410的分区结构一致,标志“0”可以表示水平二元拆分,标志“1”可以表示垂直二元拆分。对于四叉树-拆分分区,不需要指示拆分类型,因为四叉树拆分总是水平和垂直拆分块或分区,以产生大小相等的4个子块/分区。在一些实施中,标志“1”可以表示水平二元拆分,标志“0”可以表示垂直二元拆分。

[0129] 在QTBT的一些示例实施中,四叉树和二元拆分规则集可以由以下预定义参数和与其相关联的相应函数来表示:

[0130] -CTU大小:四叉树的根节点大小(基本块的大小)

[0131] -MinQTSIZE:允许的最小四叉树叶节点大小

[0132] -MaxBTSIZE:允许的最大二叉树根节点大小

[0133] -MaxBTDepth:允许的最大二叉树深度

[0134] -MinBTSIZE:允许的最小二叉树叶节点大小

[0135] 在QTBT分区结构的一些示例实施中,CTU大小可以被设置为 $128 \times 128$ 亮度样本,和两个对应的 $64 \times 64$ 色度样本块(当考虑并使用示例色度子采样时),MinQTSIZE可以被设置为 $16 \times 16$ ,MaxBTSIZE可以被设置为 $64 \times 64$ ,MinBTSIZE(对于宽度和高度)可以被设置为 $4 \times 4$ ,并且MaxBTDepth可以被设置为4。四叉树分区可以首先应用于CTU以生成四叉树叶节点。四叉树叶节点可以具有从其最小允许大小 $16 \times 16$ (即,MinQTSIZE)到 $128 \times 128$ (即,CTU大小)。如果节点是 $128 \times 128$ ,它将不会首先通过二叉树进行拆分,因为它的大小超过了MaxBTSIZE(即 $64 \times 64$ )。否则,不超过MaxBTSIZE的节点可以通过二叉树进行分区。在图14的例子中,基本块为 $128 \times 128$ 。根据预定义的规则集,基本块只能进行四叉树拆分。基本块的分区深度为0。得到的四个分区中的每一个都是 $64 \times 64$ ,不超过MaxBTSIZE,可以在级别1进一步四叉树或二叉树分割。这一过程仍在继续。当二叉树深度达到MaxBTDepth(即4)时,可以不考虑进一步的拆分。当二叉树节点的宽度等于MinBTSIZE(即4)时,可以不考虑进一步的水平拆分。类似地,当二叉树节点的高度等于MinBTSIZE时,不考虑进一步的垂直拆分。

[0136] 在一些示例实施中,上述QTBT方案可以被配置为支持亮度和色度具有相同QTBT结构或分开的QTBT结构的灵活性。例如,对于P和B切片,一个CTU中的亮度和色度CTB可以共享相同的QTBT结构。然而,对于I切片,亮度CTB可以通过QTBT结构分区为CB,色度CTB可以通过另一QTBT结构分区为色度CB。这意味着CU可以用于指代I切片中的不同颜色通道,例如,I切片可以由亮度分量的编码块或两个色度分量的编码块组成,并且P或B切片中的CU可以由所有三个颜色分量的编码块组成。

[0137] 在一些其他实施中,QTBT方案可以用上述三元方案来补充。这种实施可以被称为

多类型树 (MTT) 结构。例如,除了节点的二元拆分之外,可以选择图13的三元分区模式之一。在一些实施中,只有正方形节点可以进行三元拆分。可以使用附加标志来指示三元分区是水平的还是垂直的。

[0138] 两级或多级树的设计,例如QTBT实现和由三元拆分补充的QTBT实现,可能主要是出于降低复杂性的动机。理论上,遍历树的复杂度是 $T^D$ ,其中T表示拆分类型的数量,并且D是树的深度。可以在减少深度(D)的同时通过使用多种类型(T)来进行折衷。

[0139] 在一些实施中,CB可以被进一步分区。例如,为了在编码和解码过程期间进行帧内或帧间预测的目的,可将CB进一步分区为多个预测块(PB)。换句话说,CB可以进一步划分为不同的子分区,其中可以进行单独的预测决策/配置。并行地,为了描绘执行视频数据的变换或逆变换的级别,可将CB进一步分区为多个变换块(TB)。CB到PB和TB的分区方案可以相同,也可以不相同。例如,每个分区方案可以基于例如视频数据的各种特征使用其自己的过程来执行。在一些示例实施中,PB和TB分区方案可以是独立的。在一些其他示例实施中,PB和TB分区方案和边界可以是相关的。在一些实施中,例如,TB可以在PB分区之后被分区,并且具体地,每个PB在继编码块的分区后被确定之后,可以被进一步分区为一个或多个TB。例如,在一些实施中,PB可以被拆分成一个、两个、四个或其他数量的TB。

[0140] 在一些实施中,为了将基本块分区为编码块并进一步分区为预测块和/或变换块,亮度通道和色度通道可以被不同地处理。例如,在一些实施中,对于亮度通道,可以将编码块分区为预测块和/或变换块,而对于色度通道,可以不将编码块分区为预测块和/或变换块。因此,在这样的实施中,可以仅在编码块级别执行亮度块的变换和/或预测。对于另一个示例,亮度通道和色度通道的最小变换块大小可以不同,例如,亮度通道的编码块可以被分区为比色度通道更小的变换和/或预测块。对于又一示例,将编码块分区为变换块和/或预测块的最大深度在亮度通道和色度通道之间可以不同,例如,可以将亮度通道的编码块分区为比色度通道更深的变换块和/或预测块。对于特定示例,亮度编码块可以被分区为多个大小的变换块,这些变换块可以由递归分区向下最多2级来表示,并且可以允许诸如正方形、2:1/1:2和4:1/1:4的变换块形状以及从 $4 \times 4$ 到 $64 \times 64$ 的变换块大小。然而,对于色度块,可以只允许为亮度块指定的最大可能的变换块。

[0141] 在用于将编码块分区为PB的一些示例实施中,PB分区的深度、形状和/或其他特征可以取决于PB是帧内编码还是帧间编码。

[0142] 编码块(或预测块)到变换块的分区可以在各种示例方案中实现,包括但不限于递归地或非递归地四叉树拆分和预定模式拆分,并且在编码块或预测块的边界处附加考虑变换块。通常,得到的变换块可以处于不同的拆分级别,可以不具有相同的大小,并且可以不需要在形状上是正方形的(例如,它们可以是具有某些允许的大小和纵横比的矩形)。下面将结合图15、16和17进一步详细地描述更多的示例。

[0143] 然而,在一些其它实施中,经由上述任何分区方案获得的CB可用作用于预测和/或变换的基本或最小编码块。换句话说,为了执行帧间预测/帧内预测目的和/或变换目的,不进行进一步的拆分。例如,从上述QTBT方案获得的CB可以直接用作进行预测的单位。具体地,这种QTBT结构消除了多种分区类型的概念,即,它消除了CU、PU和TU的分离,并支持如上所述的CU/CB分区形状的更大灵活性。在这种QTBT块结构中,CU/CB可以是正方形或矩形形状。这种QTBT的叶节点被用作预测和变换处理的单元,而无需任何进一步的分区。这意味着

在这种示例性QTBT编码块结构中,CU、PU和TU具有相同的块大小。

[0144] 可以以任何方式将上述各种CB分区方案和将CB进一步分区为PB和/或TB(不包括PB/TB分区)进行组合。提供以下特定实施作为非限制性示例。

[0145] 下面描述编码块和变换块分区的具体示例实施。在这样的示例性实施中,可以使用递归二叉树拆分或上述预定义的拆分模式(例如图9和10中示出的)将基本块拆分为编码块。在每个级别,可以由本地视频数据特性来确定是否继续对特定分区进行进一步的二叉树拆分。所得到的CB可以处于各种二叉树拆分级别,并且具有各种大小。关于是否使用帧间(时间)或帧内(空间)预测来编码图片区域的决策可以在CB级别(或对于所有三色通道的CU级别)做出。每个CB可以根据预定义的PB拆分类型进一步拆分成一个、两个、四个或其他数量的PB。在一个PB内,可以应用相同的预测过程,并且可以基于PB将相关信息发送到解码器。在通过应用基于PB拆分类型的预测过程获得残差块之后,可以根据类似于CB的编码块的另一二叉树结构将CB分区成TB。在该特定实施中,CB或TB可以但不必限于正方形。此外,在该特定示例中,对于帧间预测,PB可以是正方形或矩形,并且对于帧内预测,PB可以仅是正方形。编码块可以被拆分成例如四个正方形TB。每个TB可以被进一步递归地拆分(使用二叉树拆分)成更小的TB,称为残差二叉树(RQT)。

[0146] 下面进一步描述用于将基本块分区为CB、PB和/或TB的另一示例实施。例如,代替使用诸如图9或图10所示的多个分区单元类型,可以使用具有嵌套多类型树的二叉树,该二叉树使用二元和三元拆分分割结构(例如,如上所述的QTBT或具有三元拆分的QTBT)。可以放弃CB、PB和TB的分离(即,将CB分区为PB和/或TB,以及将PB分区为TB),除非需要大小对于最大变换长度来说太大的CB,其中这样的CB可能需要进一步拆分。该示例分区方案可以被设计成支持CB分区形状的更大灵活性,使得预测和变换都可以在CB级别上执行,而无需进一步分区。在这种编码树结构中,CB的形状可以是正方形或矩形。具体地,编码树块(CTB)可以首先通过二叉树结构分区。然后,二叉树叶节点可以通过嵌套的多类型树结构进一步被分区。使用二元或三元拆分的嵌套多类型树结构的示例如图11所示。具体地,图11的示例性多类型树结构包括四种拆分类型,称为垂直二元拆分(SPLIT\_BT\_VER)(1102)、水平二元拆分(SPLIT\_BT\_HOR)(1104)、垂直三元拆分(SPLIT\_TT\_VER)(1106)和水平三元拆分(SPLIT\_TT\_HOR)(1108)。然后,CB对应于多类型树的叶子。在该示例实施中,除非CB对于最大变换长度来说太大,否则该分割用于预测和变换处理,而无需任何进一步的分区。这意味着,在大多数情况下,在具有嵌套多类型树编码块结构的二叉树中,CB、PB和TB具有相同的块大小。当支持的最大变换长度小于CB的颜色分量的宽度或高度时,会发生异常。在一些实施中,除了二元或三元拆分之外,图11的嵌套模式还可以包括二叉树拆分。

[0147] 图12示出了一个基本块的块分区(包括二叉树、二元和三元拆分选项)的具有嵌套多类型树编码块结构的二叉树的一个具体示例。更详细地说,图12示出了基本块1200被二叉树拆分成四个正方形分区1202、1204、1206和1208。对于每一个二叉树-拆分区,决定进一步使用图11的多类型树结构和二叉树进行进一步拆分。在图12的例子中,分区1204没有被进一步拆分。分区1202和1208各自采用另一二叉树拆分。对于分区1202,第二级二叉树-拆分左上、右上、左下和右下分区分别采用二叉树的第三级拆分、图11的水平二元拆分1104、不拆分和图11的水平三元拆分1108。分区1208采用另一个二叉树拆分,并且第二级二叉树-拆分左上、右上、左下和右下分区分别采用图11的垂直三元拆分1106的第三级拆分、

不拆分、不拆分和图11的水平二元拆分1104。分别根据图11的水平二元拆分1104和水平三元拆分1108进一步拆分1208的第三级左上分区的两个子分区。分区1206采用第二级拆分模式,其遵循图11的垂直二元拆分1102成两个分区,这两个分区根据图11的水平三元拆分1108和垂直二元拆分1102在第三级进一步拆分。根据图11的水平二元拆分1104,第四级拆分进一步应用于它们中的一个。

[0148] 对于上面的特定示例,最大亮度变换大小可以是 $64 \times 64$ ,并且支持的最大色度变换大小可以不同于例如 $32 \times 32$ 的亮度。即使上面在图12中的示例CB通常不被进一步拆分成更小的PB和/或TB,但当亮度编码块或色度编码块的宽度或高度大于最大变换宽度或高度时,亮度编码块或色度编码块可在水平和/或垂直方向上被自动拆分以满足该方向上的变换尺寸限制。

[0149] 在上述用于将基本块分区为CB的具体示例中,并且如上所述,编码树方案可以支持亮度和色度具有单独的块树结构的能力。例如,对于P和B切片,一个CTU中的亮度和色度CTB可以共享相同的编码树结构。例如,对于I切片,亮度和色度可以具有单独的编码块树结构。当应用单独的块树结构时,亮度CTB可以通过一个编码树结构被分区为亮度CB,并且色度CTB通过另一个编码树结构被分区为色度CB。这意味着I切片中的CU可以由亮度分量的编码块或两个色度分量的编码块组成,并且P或B切片中的CU总是由所有三个颜色分量的编码块组成,除非视频是单色的。

[0150] 当编码块被进一步分区为多个变换块时,其中的变换块可以按照各种顺序或扫描方式在比特流中排序。用于将编码块或预测块分区为变换块的示例实施以及变换块的编码顺序在下面进一步详细描述。在一些示例实施中,如上所述,变换分区可以支持多个形状的变换块,例如1:1(正方形)、1:2/2:1和1:4/4:1,变换块大小的范围从例如 $4 \times 4$ 到 $64 \times 64$ 。在一些实施中,如果编码块小于或等于 $64 \times 64$ ,则变换块分区可以仅应用于亮度分量,使得对于色度块,变换块大小与编码块大小相同。否则,如果编码块宽度或高度大于64,则亮度和色度编码块可以分别被隐式地拆分成 $\min(W, 64) \times \min(H, 64)$ 和 $\min(W, 32) \times \min(H, 32)$ 变换块的倍数。

[0151] 在变换块分区的一些示例实现中,对于帧内和帧间编码块,编码块可以被进一步分区成多个变换块,其分区深度高达预定数目的级别(例如,2个级别)。变换块分区深度和大小可以相关联。对于一些示例实施,从当前深度的变换大小到下一个深度的变换大小的映射如下面的表1中所示。

[0152] 表1:变换分区大小设置

当前深度的变换大小	下一个深度的变换大小
TX_4X4	TX_4X4
TX_8X8	TX_4X4
TX_16X16	TX_8X8
TX_32X32	TX_16X16
TX_64X64	TX_32X32
TX_4X8	TX_4X4
TX_8X4	TX_4X4
TX_8X16	TX_8X8
[0153] TX_16X8	TX_8X8
TX_16X32	TX_16X16
TX_32X16	TX_16X16
TX_32X64	TX_32X32
TX_64X32	TX_32X32
TX_4X16	TX_4X8
TX_16X4	TX_8X4
TX_8X32	TX_8X16
TX_32X8	TX_16X8
TX_16X64	TX_16X32
TX_64X16	TX_32X16

[0154] 基于表1的示例映射,对于1:1正方形块,下一级变换拆分可以创建四个1:1正方形子变换块。例如,变换分区可以在4x4处停止。因此,当前深度为4×4的变换大小对应于下一深度的相同大小4×4。在表1的示例中,对于1:2/2:1的非正方形块,下一级变换拆分可以创建两个1:1的正方形子变换块,而对于1:4/4:1的非正方形块,下一级变换拆分可以创建两个1:2/2:1的子变换块。

[0155] 在一些示例实施中,对于帧内编码块的亮度分量,可以相对于变换块分区应用附加限制。例如,对于变换分区的每一级,所有子变换块可以被限制为具有相等的大小。例如,对于32x16编码块,级别1变换拆分创建2个16x16子变换块,级别2变换拆分创建8个8x8子变换块。换句话说,第二级拆分必须应用于所有第一级子块,以保持变换单元大小相等。图15中示出了遵循表1的用于帧内已编码的正方形块的变换块分区的示例以及箭头所示的编码顺序。具体地,1502示出了正方形编码块。在1504中示出了根据表1第一级拆分成4个大小相等的变换块,其编码顺序由箭头指示。1506中示出了根据表1将所有第一级相等大小的块第二级拆分成16个相等大小的变换块,其编码顺序由箭头指示。

[0156] 在一些示例实施中,对于帧间编码块的亮度分量,可以不应用上述帧内编码的限制。例如,在第一级变换拆分之后,子变换块中的任何一个子变换块可以被进一步独立地再拆分一个以上的级别。因此,得到的变换块可以具有相同或不同的大小。图16示出了将帧间已编码块拆分成具有其编码顺序的变换块的示例。在图16的例子中,帧间已编码块1602根据表1被拆分成两级的变换块。在第一级,帧间已编码块被拆分成大小相等的四个变换块。

然后,仅四个变换块中的一个(不是全部)被进一步拆分成四个子变换块,得到具有两种不同大小的总共7个变换块,如1604所示。这7个变换块的示例编码顺序由图16的1604中的箭头示出。

[0157] 在一些示例实施中,对于色度分量,可以应用对变换块的一些附加限制。例如,对于色度分量,变换块大小可以与编码块大小一样大,但不小于预定义的大小,例如 $8 \times 8$ 。

[0158] 在一些其他示例实施中,对于宽度(W)或高度(H)大于64的编码块,亮度和色度编码块都可以分别被隐式地拆分成 $\min(W, 64) \times \min(H, 64)$ 和 $\min(W, 32) \times \min(H, 32)$ 变换单元的倍数。这里,在本公开中,“ $\min(a, b)$ ”可以返回a和b之间的较小值。

[0159] 图17进一步示出了用于将编码块或预测块分区为变换块的另一替代示例方案。如图17所示,代替使用递归变换分区,可以根据编码块的变换类型将预定义的分区类型集合应用于编码块。在图17所示的特定示例中,可以应用6种示例分区类型中的一种来将编码块拆分为各种数量的变换块。这种生成变换块分区的方案可以应用于编码块或预测块。

[0160] 更详细地,图17的分区方案为任何给定的变换类型提供了多达6个示例分区类型(变换类型是指例如主变换的类型,例如ADST和其他)。在该方案中,每个编码块或预测块可以基于例如率失真成本被分配变换分区类型。在示例中,可基于编码块或预测块的变换类型来确定分配给编码块或预测块的变换分区类型。特定的变换分区类型可以对应于变换块拆分大小和模式,如图17中的6种变换分区类型所示。可以预定义各种变换类型和各种变换分区类型之间的对应关系。下面示出了一个示例,其中大写标签指示可基于率失真成本分配给编码块或预测块的变换分区类型:

[0161] • PARTITION\_NONE:分配变换大小,其等于块大小。

[0162] • PARTITION\_SPLIT:分配变换大小,其宽度为块大小的宽度的1/2且高度为块大小的高度的1/2。

[0163] • PARTITION\_HORZ:分配变换大小,其具有与块大小相同的宽度且高度为块大小的高度的1/2。

[0164] • PARTITION\_VERT:分配变换大小,其宽度为块大小的宽度的1/2且高度与块大小的高度相同。

[0165] • PARTITION\_HORZ4:分配变换大小,,其宽度与块大小的宽度相同且高度为块大小的高度的1/4。

[0166] • PARTITION\_VERT4:分配变换大小,其宽度为块大小的宽度的1/4且高度与块大小的高度相同。

[0167] 在上面的例子中,如图17所示的变换分区类型都包含用于已分区变换块的统一变换大小。这只是一个例子,而不是用来限制本发明。在一些其他实施中,混合变换块大小可以用于特定分区类型(或模式)中的已分区变换块。

[0168] 视频块(PB或CB,当没有被进一步分区成多个预测块时也称为PB)可以以各种方式预测,而不是直接编码,从而利用视频数据中的各种相关性和冗余来提高压缩效率。相应地,这种预测可以以各种模式执行。例如,可以通过帧内预测或帧间预测来预测视频块。特别是在帧间预测模式中,视频块可以由来自一个或多个其他帧的一个或多个其他参考块或帧间预测器块通过单参考或复合参考帧间预测来进行预测。为了实现帧间预测,参考块可以由其帧标识符(参考块的时间位置)和指示正在编码或解码的当前块和参考块之间的空

间偏移的运动矢量(参考块的空间位置)来指定。可以在比特流中用信号通知参考帧标识和运动矢量。作为空间块偏移的运动矢量可以直接用信号通知,或者可以由另一参考运动矢量或预测器运动矢量本身预测。例如,可以通过(例如候选相邻块的)参考运动矢量直接预测当前运动矢量,或者通过参考运动矢量和当前运动矢量与参考运动矢量之间的运动矢量差(motion vector difference, MVD)的组合来预测当前运动矢量。后者可以被称为具有运动矢量差的合并模式(merge mode with motion vector difference, MMVD)。在比特流中,参考运动矢量可以被识别为指向例如当前块的空间相邻块或时间相邻但空间同位块的指针。

[0169] 在一些其他示例实施中,可以采用帧内块复制(intra-block copy, IBC)预测。在IBC中,当前帧中的当前块可以使用当前帧中的另一个块(而不是时间上不同的帧,因此使用术语“帧内”)结合块矢量(block vector, BV)来预测,该块矢量用于指示帧内预测器或参考块的位置相对于正被预测的块的位置的偏移。编码块的位置可以由例如相对于当前帧(或切片)的左上角的左上角的像素坐标来表示。因此,IBC模式在当前帧内使用类似帧间预测概念。例如,一BV可通过其他参考BV直接或结合当前BV和参考BV之间的BV差来预测,这类似于在帧间预测中使用参考MV和MV差来预测MV。IBC有助于提供改进的编解码效率,特别是用于编码和解码具有屏幕内容的视频帧,例如,该屏幕内容含有大量重复模式,例如文本信息,其中,相同的文本分段(字母、符号、单词、短语等)出现在同一帧的不同部分中,并且可以用来相互预测。

[0170] 在一些实施中,IBC可以被视为除了正常的帧内预测模式和正常的帧间预测模式之外的单独的预测模式。因此,可以在三种不同的预测模式:帧内预测、帧间预测和IBC模式中进行特定块的预测模式的选择并用信号通知该选择的预测模式。在这些实施中,可以在这些模式的每一个中建立灵活性,以优化这些模式中的每一个的编解码效率。在一些其它实施中,可以使用类似的运动矢量确定、参考和编解码机制,将IBC视为帧间预测模式内的子模式或分支。在这样的实施中(集成帧间预测模式和IBC模式),为了协调一般帧间预测模式和IBC模式,IBC的灵活性可能在一定程度上受到限制。然而,这种实施不太复杂,同时仍然可以利用IBC来提高例如以屏幕内容为特征的视频帧的编解码效率。在一些示例实施中,利用现有的用于单独的帧间预测模式和帧内预测模式的预先指定的机制,可以扩展帧间预测模式以支持IBC。

[0171] 这些预测模式的选择可以在各种级别进行,包括但不限于序列级别、帧级别、图片级别、切片级别、CTU级别、CT级别、CU级别、CB级别或PB级别。例如,为了IBC的目的,关于是否采用IBC模式的决定可以在CTU级别做出并用信号通知。如果CTU被信号通知为采用IBC模式,那么整个CTU中的所有编码块都可以由IBC预测。在一些其它实施中,IBC预测可以在超级块(SB或超块)级别确定。每个SB可以以各种方式(例如,二叉树划分)被拆分成多个CTU或分区。下面进一步提供示例。

[0172] 图18从解码器的角度示出了包含多个CTU的当前帧的部分的示例简要说明。每个方块(如1802)代表一个CTU。CTU大小可以是以上详细描述的各种预定义大小之一,例如SB。每个CTU可以包括一个或多个编码块(或预测块,用于特定的颜色通道)。使用水平线遮蔽的CTU代表那些已经重建的CTU。CTU 1804代表正在重建的当前CTU。在当前CTU 1804中,使用水平线遮蔽的编码块表示在当前CTU中已经被重建的那些块,使用斜线遮蔽的编码块1806

当前正在被重建,而当前CTU 1804中无遮蔽的编码块正在等待重建。其他无遮蔽的CTU尚未处理。

[0173] 如图18中的示例箭头所示,IBC中用于预测当前编码块的参考块(相对于当前块)的位置或偏移可以由BV指示。例如,BV可以以矢量形式指示参考块(在图18中标记为“Ref”)的左上角与当前块的左上角之间的位置差。而图18是使用CTU作为基本IBC单元来说明的。基本原则适用于将SB用作基本IBC单元的实施。在这样的实施中,如下面更详细描述,每个超级块可以被划分成多个CTU,并且每个CTU可以被进一步划分成多个编码块。

[0174] 如下面进一步更详细地公开的,取决于用于IBC的参考CTU/SB相对于当前CTU/SB的位置,参考CTU/SB可以被称为局部CTU/SB或非局部CTU/SB。局部CTU/SB可以指与当前CTU/SB重合的CTU/SB,或者指靠近当前CTU/SB并且已经被重建的CTU/SB(例如,当前CTU/SB的左相邻CTU/SB)。非局部CTU/SB可以指距离当前CTU/SB更远的CTU/SB。当执行当前编码块的IBC预测时,可以搜索局部CTU/SB和非局部CTU/SB中的任一个或全部以寻找参考块。实现IBC的具体方式可取决于参考CTU/SB是局部的还是非局部的,因为用于局部或非局部CTU/SB参考的重建样本(例如片外图片缓冲器(DPB)和/或片上存储器)的片上和片外存储管理可能不同。例如,已重建的局部CTU/SB样本可适于存储在用于IBC的编码器或解码器的片上存储器中。例如,已重建的非局部CTU/SB样本可以存储在片外DPB存储器或外部存储器中。

[0175] 在一些实施中,可以被用作当前编码块1804的参考块的已重建块的位置可以受到限制。这种限制可以是各种因素的结果,并且可以取决于IBC是作为一般帧间预测模式的集成部分、帧间预测模式的特殊扩展还是单独和独立的IBC模式。在一些示例中,可以仅搜索当前已重建的CTU/SB样本来识别IBC参考块。在一些其他示例中,如图18的粗虚线框1808所示,当前已重建的CTU/SB样本和另一个相邻已重建的CTU/SB样本(例如,左侧相邻的CTU/SB)可用于参考块搜索和选择。对于这样的实施,只有局部已重建的CTU/SB样本可以用于IBC参考块搜索和选择。在一些其他示例中,由于各种其他原因,某些CTU/SB可能不可用于IBC参考块搜索和选择。例如,在图18中用交叉线标记的CTU/SB 1810可能不可用于搜索和选择当前块1804的参考块,因为它们可以用于特殊目的(例如,波前并行处理),如下面进一步描述的。

[0176] 在一些实施中,如图18所示的,由粗虚线框1808形成的区域可以被称为局部搜索区域。局部搜索区域中的样本可以存储在片上存储器中。

[0177] 在一些实施中,当允许帧内块复制(IntraBC)时,环路滤波器被禁用,环路滤波器包括去块滤波器、约束方向增强滤波器(Constrained Directional Enhancement Filter, CDEF)和环路恢复(Loop Restoration, LR)。通过这样做,可以避免专用于启用/支持IntraBC的第二图片缓冲器。

[0178] 在一些实施中,采用并行解码(即,同时解码多于一个的编码块)可能导致对于被允许用于提供IBC参考块或参考样本的已经重建的CTU/SB的限制。在如图19所示的例子中,每个方块代表一个CTU/SB。可以实现并行解码,其中,可以在并行处理中重建多个连续行和每隔一列(每两列)中的多个CTU/SB,如图19中使用斜线遮蔽的CTU/SB所示。其他水平线遮蔽的CTU/SB已经重建,无遮蔽的CTU/SB是尚未构建的。通过这种并行处理,对于左上坐标为 $(x_0, y_0)$ 的当前并行处理的CTU/SB,只有当垂直坐标 $y$ 小于 $y_0$ 且水平坐标 $x$ 小于 $x_0 + 2(y_0 - y)$ 时,才可以在IBC中访问 $(x, y)$ 处的已重建样本以预测当前CTU/SB,因此,使用水平线遮蔽的已

经构建的CTU/SB可以作为并行处理的当前块的参考。注意,诸如 $(x_0, y_0)$ 和 $(x, y)$ 的坐标单位可以包括像素、块(例如,SB)等。

[0179] 在一些实施中,将立即重建的样本写入片外DPB的写回时延可以对可用于为当前块提供IBC参考样本的CTU/SB施加进一步的限制,特别是当片外DPB用于保持(hold)IBC参考样本时。在图20中示出了一个示例,其中可以在图19所示的那些限制之上施加附加的限制。具体地,为了允许硬件写回时延,IBC预测可以不访问立即重建的区域来搜索和选择参考块。限制或禁止的立即重建区域的数量可以是 $1 \sim n$  CTU/SBs,其中 $n$ 是正数, $n$ 可以与写回时延的持续时间正相关。因此,在图19的特定并行处理限制之上,对于当前CTU/SB(使用斜线遮蔽),如果一个当前CTU/SB的左上位置的坐标是 $(x_0, y_0)$ ,则如果垂直坐标 $y$ 小于 $y_0$ 并且水平坐标小于 $x_0 + 2(y_0 - y) - D$ ,则可以通过IBC访问位置 $(x, y)$ 处的预测,其中 $D$ 表示被限制/禁止用作IBC参考的立即重建区域(例如,在当前CTU/SB的左侧)的数目。图20示出了 $D=2$ (以块为单位,或者当每个块是 $128 \times 128$ SB时以像素为单位的 $2 \times 128$ 像素)的情况下被限制在作为IBC参考样本之外的这些附加CTU/SB。这些不能作为IBC参考的附加CTU/SB由反向斜线阴影表示。

[0180] 在一些实施中,如图20所示,由使用用水平线遮蔽的块形成的区域可以被称为非局部搜索区域,并且该区域中的样本可以被存储在外部存储器中。

[0181] 在一些实施中,也在下面进一步详细描述,局部搜索区域和非局部CTU/SB搜索区域都可以用于IBC参考块搜索和选择。另外,当使用片上存储器时,关于已经构建的CTU/SB作为IBC参考的可用性的限制中关于写回时延的一些限制可以被放宽或移除。在一些进一步的实施中,局部CTU/SB和非局部CTU/SB共存时的使用方式可以不同,这是由于例如使用片上存储器或片外存储器对参考块的缓冲管理的不同。在下面的公开中更详细地描述这些实施。

[0182] 在一些实施中,IBC可以实现为帧间预测模式的扩展,帧间预测模式中将当前帧视为参考帧,使得当前帧内的块可以用作预测参考。这样的IBC实现因此可以遵循用于帧间预测的编码路径,即使IBC过程仅涉及当前帧。在这样的实施中,帧间预测模式的参考结构可以适于IBC,其中使用BV对参考样本的寻址机制的表示可以类似于帧间预测中的运动矢量(MV)。因此,依赖于与基于当前帧作为参考帧的帧间预测模式相似或相同的语法结构和解码过程,IBC可以被实现为特殊的帧间预测模式。

[0183] 在这样的实施中,由于IBC可以被视为帧间预测模式,因此仅帧内预测切片必须成为允许使用IBC的预测切片。换句话说,仅帧内预测切片不会被帧间预测(因为帧内预测模式不会调用任何帧间预测处理路径),因此IBC不会被允许用于这种仅帧内切片中的预测。当IBC适用时,编码器(coder)将参考图片列表扩展一个用于指向当前图片的指针的条目。因此,当前图片可以占用共享解码图片缓冲器(decoded picture buffer,DPB)的多达一个图片大小的缓冲器。使用IBC的信令可以隐含在帧间预测模式中的参考帧的选择中。例如,当所选择的参考图片指向当前图片时,如果需要并且可用,编码单元将采用具有类似帧间预测的编码路径的IBC和特殊的IBC扩展。在一些特定实施中,与常规帧间预测相反,IBC过程中的参考样本在用于预测之前可以不被环路滤波。此外,相应的参考当前图片可以是长期参考帧,因为它将在待编码或解码的下一帧附近。在一些实施中,为了最小化存储器需求,编码器可以在重建当前图像之后立即释放缓冲器。当重建图片的滤波版本在真正的帧

间预测中成为后续帧的参考图片时,编码器可以将其填充回DPB中作为短期参考,即使其在用于IBC时可能是未滤波的。

[0184] 在上面的示例实施中,即使IBC可能仅仅是帧间预测模式的扩展,IBC也可以利用可能偏离正常的帧间预测的几个特殊过程来处理。例如,IBC参考样本也可以不经过滤波。换句话说,包括去块滤波(deblocking filtering, DBF)、样本自适应偏移(Sample Adaptive Offset, SAO)滤波、跨分量样本偏移(Cross-Component Sample Offset, CCSO)滤波等在内的环内滤波过程之前的重建样本可用于IBC预测,而正常的帧间预测模式采用已滤波样本用于预测。又例如,可以不执行IBC的无亮度样本插值,并且仅当色度BV在从亮度BV导出时是非整数时,色度样本插值才可能是必要的。再例如,当色度BV为非整数,并且用于IBC的参考块在用于IBC参考的可用区域的边界附近时,周围的已重建样本可以在边界之外以执行色度内插。指向一条毗邻边界的线的BV不可能避免这种情况。

[0185] 在这样的实施中,通过IBC对当前块的预测可以重用(reuse)帧间预测过程的预测和编码机制,包括使用参考BV来预测当前BV和例如附加BV差。然而,在一些特定实施中,亮度BV可以以整数分辨率而不是如在用于常规帧间预测的MV中的分数精度来实现。

[0186] 在一些实施中,图18中用水平阴影线指示的所有CTU和SB可用于搜索和选择IBC参考块,除了图18中的1810所示的、在当前CTU的右侧和上方的两个CTU(图18中用交叉线指示)用于允许波前并行处理(Wavefront Parallel Processing, WPP)。因此,除了出于并行处理目的的一些例外,几乎是当前图片的整个已经重建的区域。

[0187] 在一些其它实施中,可以从中搜索和选择IBC参考块的区域可以被限制为局部CTU/SB。一个示例由图18的粗虚线框1808指示。在这样的示例中,当前CTU左侧的CTU/SB可以在当前CTU的重建过程的开始处用作IBC的参考样本区域。当使用这样的局部参考区域时,代替在DPB中分配额外的外部存储器空间,可以分配片上存储器空间来保持局部CTU/SB用于IBC参考。在一些实施中,固定的片上存储器可以用于IBC,从而降低在硬件架构中实现IBC的复杂性。因此,可以实现独立于正常的帧间预测的专用IBC模式用于利用片上存储器,而不是仅仅实现为帧间预测模式的扩展。

[0188] 例如,对于每个颜色分量,用于存储局部IBC参考样本(例如左CTU或SB)的固定片上存储器大小可以是 $128 \times 128$ 。在一些实施中,最大CTU大小也可以是 $128 \times 128$ 。在这种情况下,参考样本存储器(reference sample memory, RSM)可以保持具有单个CTU大小的样本。在一些其他替代实施中,CTU大小可以更小。例如,CTU大小可以是 $64 \times 64$ 。因此,RSM可以同时保持多个(在该示例情况下为4个)CTU。在又一些其它实现中,RSM可以保持多个SB,每个SB可以包括一个或多个CTU,并且每个CTU可以包括多个编码块。

[0189] 在局部片上IBC参考的一些实施中,片上RSM保持一个CTU,并且可以实现连续更新机制,以用于使用当前CTU的已重建样本替换左相邻CTU的已重建样本。图21示出了在重建过程期间的四个中间时间处的这种连续RSM更新机制的简化示例。在图21的例子中,RSM具有保持一个CTU的固定大小。CTU可能包括隐式划分。例如,CTU可以被隐式地划分为四个分离的区域(例如,四叉树划分)。每个区域可以包括多个编码块。CTU的大小可以是 $128 \times 128$ ,而对于示例四叉树划分,每个示例区域或分区的大小可以是 $64 \times 64$ 。在每个中间时间中用水平线遮蔽的RSM的区域/分区保持左相邻CTU的对应的已重建参考样本,并且用垂直线灰色遮蔽的区域/分区保持当前CTU的对应的已重建参考样本。用斜线遮蔽的RSM的编码块表

示当前区域内的当前编码块正在被编码/解码/重建。

[0190] 如2102所示,在表示当前CTU重建的开始的第一中间时间,RSM可仅针对四个示例区域中的每一个包括左相邻CTU的重建参考样本。在其他三个中间时间中,重建过程逐渐用当前CTU的已重建样本取代左侧相邻CTU的已重建参考样本。当编码器处理该区域/分区的第一个编码块时,RSM中 $64 \times 64$ 的区域/分区发生重置(reset)。在重置RSM的区域时,该区域被认为是空白的,并且被认为没有保持任何用于IBC的已重建参考样本(换句话说,RSM的该区域还没有准备好用作IBC参考样本)。当处理该区域中的相应当前编码块时,RSM中的相应块用当前CTU的相应块的已重建样本填充,以用作下一个当前块的IBC的参考样本,如图21中针对中间时间2104、2106和2108所示。一旦对应于RSM的区域/分区处理了所有编码块,该整个区域就用这些当前编码块的已重建样本作为IBC参考样本来填充,如图21中在各个中间时间处用垂直线完全遮蔽的区域所示。因此,在中间时间2104和2106中,RSM中的一些区域/分区保持来自相邻CTU的IBC参考样本,一些其他区域/分区完全保持来自当前CTU的参考样本,而一些区域/分区部分地保持来自当前CTU的参考样本并且部分空白(由于上述重置处理的结果,空白部分不用于IBC参考)。当最后一个区域(例如,右下角区域)正在被处理时,所有其他三个区域将保持当前CTU的已重建样本作为IBC的参考样本,而最后一个区域/分区部分地保持当前CTU中对应编码块的已重建样本并且部分空白,直到CTU的最后编码块被重建,此时整个RSM保持当前CTU的已重建样本并且如果仍以IBC模式编码,RSM准备用于下一个CTU。

[0191] 图22示出了RSM在特定中间时间处在空间上的上述连续更新的实施,即,左相邻CTU和具有当前编码块(斜阴影线遮蔽的块)的当前CTU都被示出。这两个CTU对应的、在RSM中并且有效地作为当前编码块的IBC参考样本的已重建样本由水平和垂直阴影线示出。在该示例中的特定重建时间,在RSM中,该处理已经用由垂直阴影线遮蔽的当前CTU的区域替换了由左侧相邻CTU中的无阴影区域覆盖的样本。来自相邻CTU的剩余有效样本显示为水平线阴影。

[0192] 在上面的示例实现中,当固定RSM大小与CTU大小相同时,RSM被实现为包含一个CTU。在CTU大小较小的一些其它实施中,RSM可以包含多于一个CTU。例如,CTU的大小可以是 $32 \times 32$ ,而固定RSM的大小可以是 $128 \times 128$ 。因此,RSM可以保持16个CTU的样本。遵循上述相同的基础RSM更新原则,RSM可以在被重建之前保持当前 $128 \times 128$ 小块(patch)的16个相邻CTU。一旦当前 $128 \times 128$ 小块的第一个编码块的处理开始,RSM中的初始使用一个相邻CTU的已重建样本填充的第一个 $32 \times 32$ 区域可以如上所述针对保持单个CTU的RSM那样进行更新。剩余的15个 $32 \times 32$ 区域包含15个相邻的CTU作为IBC的参考样本。一旦与正被解码的当前 $128 \times 128$ 小块的第一个 $32 \times 32$ 区域对应的CTU被重建,则用该CTU的已重建样本更新RSM的第一个 $32 \times 32$ 区域。然后,与当前 $128 \times 128$ 小块的第二个 $32 \times 32$ 区域相对应的CTU可以被处理并且最终用已重建样本来进行更新。该过程继续直到RSM的16个 $32 \times 32$ 区域包含当前 $128 \times 128$ 小块(全部15个CTU)的已重建样本。然后,解码过程进行到下一个 $128 \times 128$ 小块。

[0193] 在一些其他实施中,作为图21和22的扩展,RSM可以保持一组相邻的CTU。一次处理一个当前CTU,保持最远相邻CTU的RSM部分按照上述方式利用已重建的当前CTU进行更新。对于下一个当前CTU,同样,RSM中最远的相邻CTU被更新和替换。因此,保持在固定大小RSM中的多个CTU随着相邻CTU的移动窗口进行更新以用于IBC。

[0194] 图23示出了使用片上RSM的局部IBC的另一个具体示例实施。在该示例中,IBC模式的最大块大小可能受到限制。例如,最大的IBC块可以是 $64 \times 64$ 。片上RSM可以被配置有与超级块(SB)相对应的固定大小,例如 $128 \times 128$ 。图23的RSM实现使用与图21和图22的实现类似的基本原理。在图23中,RSM可以保持多个相邻和/或当前CTU作为IBC参考样本。在图23的例子中,SB可以被二叉树拆分。相应地,RSM可以二叉树拆分成4个区域或单元,每个区域或单元为 $64 \times 64$ 。这些区域中的每一个可以保持一个或多个编码块。可替换地,这些区域中的每一个可以保持一个或多个CTU,并且每个CTU可以保持一个或多个编码块。二叉树区域的编码顺序可以是预定义的。例如,编码顺序可以是左上、右上、左下、右下。图23的SB的二叉树拆分仅仅是一个示例。在一些其他可替换的实现方式中,SB可以按照任何其他方案进行拆分。本文所描述的用于局部IBC的RSM更新实现适用于那些替代的拆分方案。

[0195] 在这种局部IBC实施中,可用于IBC预测的局部参考块可被限制。例如,可能要求参考块和当前块应该在同一SB行中。具体地,局部参考块可以仅位于当前SB中或当前SB左侧的一个SB中。由图23中的虚线箭头示出了由另一个允许的编码块在IBC中预测的示例当前块。当当前SB或左侧SB用于IBC参考时,RSM中的参考样本更新过程可以遵循上述重置过程。例如,当 $64 \times 64$ 单元参考样本存储器中的任何一个开始用来自当前SB的已重建样本进行更新时,整个 $64 \times 64$ 单元中先前存储的参考样本(来自左侧SB)被标记为不可用于生成IBC预测样本,并且它是用当前块的已重建样本逐渐更新的。

[0196] 图23示出了在面板2302中对当前SB进行局部IBC解码期间RSM的5个示例状态。同样,在每个示例状态中使用水平线遮蔽的RSM的区域保持左相邻SB的相应二叉树区域的相应参考样本,并且使用垂直线灰色遮蔽的区域/分区保持当前SB的相应参考样本。使用斜线遮蔽的RSM的编码块表示当前二叉树区域内的当前编码块正在被编码/解码。在每个当前SB的编码开始时,RSM存储先前已编码SB的样本(图23的RSM状态(0))。当当前块位于当前SB中的四个 $64 \times 64$ 二叉树区域之一时,RSM中的相应区域被重置并用于存储当前 $64 \times 64$ 编码区域的样本。这样,RSM的每个 $64 \times 64$ 二叉树区域中的样本由当前SB中的样本逐渐更新(状态(1)-状态(3))。当当前SB已经被完全编码时,整个RSM使用当前SB的所有样本填充(状态(4))。

[0197] 图23的面板2302中的 $64 \times 64$ 区域中的每一个区域用空间编码序列号来标记。序列号0-3表示左邻居SB的4个 $64 \times 64$ 二叉树区域,而序列号4-7表示当前SB面板的4个 $64 \times 64$ 二叉树区域。在图23中,面板2304进一步示出了对于图23的面板2302的RSM状态(1)、状态(2)和状态(3), $128 \times 128$ RSM中参考样本的左相邻和当前SB中的对应空间分布。没有交叉线的阴影区域表示RSM中具有已重建样本的区域。有交叉线的阴影区域表示RSM中具有左侧SB的已重建样本被重置的区域(因此不可用作局部IBC的参考样本)。

[0198]  $64 \times 64$ 区域的编码顺序和相应的RSM更新顺序可以遵循水平扫描(如上面图23所示)或垂直扫描。水平扫描从左上开始,到右上、左下和右下。垂直扫描从左上开始、到左下、右上和右下。在图24的面板2402和2404中分别示出了用于水平扫描和垂直扫描的左相邻SB和当前SB参考样本更新过程,用于在重建当前SB的四个 $64 \times 64$ 区域中的每一个时进行比较。在图24中,用水平线遮蔽而没有交叉线的 $64 \times 64$ 区域表示具有可用于IBC的样本的区域。用水平线遮蔽且带有交叉线的区域表示左侧相邻SB的已经被更新为当前SB的相应已重建样本的区域。无阴影区域表示当前SB的未处理区域。用斜线遮蔽的块表示正在处理的当前编

码块。

[0199] 如图24所示,根据当前编码块相对于当前SB的位置,可以应用以下关于IBC的参考块的限制。

[0200] 如果当前块落入当前SB的左上 $64 \times 64$ 区域,那么除了当前SB中已经重建的样本之外,它还可以参考左侧SB的右下、左下和右上 $64 \times 64$ 块中的参考样本,如图24的2412(用于水平扫描)和2422(用于垂直扫描)所示。

[0201] 如果当前块落入当前SB的右上 $64 \times 64$ 块中,那么除了当前SB中已经重建的样本之外,如果相对于当前SB位于(0,64)处的亮度样本尚未被重建,则当前块还可以参考左侧SB的左下 $64 \times 64$ 块和右下 $64 \times 64$ 块中的参考样本(图24的2414)。否则,当前块还可以参考左侧SB的右下 $64 \times 64$ 块中的参考样本(图24的2426)用于IBC。

[0202] 如果当前块落入当前SB的左下 $64 \times 64$ 块中,那么除了当前SB中已经重建的样本之外,如果相对于当前SB的亮度位置(64,0)尚未被重建,则当前块还可以参考左侧SB的右上 $64 \times 64$ 块和右下 $64 \times 64$ 块中的参考样本(图24的2424)。否则,当前块还可以参考左侧SB(图24的2416)的右下 $64 \times 64$ 块中的参考样本用于IBC。

[0203] 如果当前块落入当前SB的右下 $64 \times 64$ 块中,则它只能参考当前SB中已经重建的样本(图24的2418和2428)用于IBC。

[0204] 如上所述,在一些示例实现中,基于局部和非局部的CTU/SB中的任一个或两者都可用于IBC参考块搜索和选择。此外,当片上RSM用于局部参考时,关于已经构建的CTU/SB作为IBC参考的可用性的限制中关于写回时延的一些限制可以放宽或移除。无论是否采用并行解码,都可以应用这样的实施。

[0205] 图25示出了可用于IBC的局部和非局部参考CTU/SB的示例实现,其中同样的,每个方块代表一个CTU/SB。用斜线遮蔽的CTU/SB代表当前CTU/SB(标记为“0”),而用水平线遮蔽的CTU/SB(标记为“1”)、垂直线遮蔽的CTU/SB(标记为“2”)和反向斜线遮蔽的CTU/SB(标记为“3”)代表已经构建的区域。无阴影的CTU/SB代表尚未重建的区域。假设使用类似于图19和20的并行解码。用垂直线(“2”)和反向斜线(“3”)遮蔽的CTU/SB表示通常被限制在作为当前CTU/SB的IBC参考之外的示例区域,这是由于当仅片外存储器用于IBC参考时到DPB的写回时延(参见图20)。当使用片上RSM时,图20的限制区域中的一个或多个可以直接从RSM参考,因此可以不需要将其限制出去。现在可以通过RSM访问以供IBC参考的限制区域的数量可能取决于RSM的大小。在图25的例子中,假设RSM可以保持一个CTU/SB,并采用上述RSM更新机制。因此,以垂直线遮蔽并标记为“2”的左侧相邻CTU/SB可用于局部参考。然后,RSM保持来自左侧CTU/SB和当前CTU/SB的样本。因此,在图25的例子中,可用于非局部IBC参考块的搜索区域包括标记为“1”的CTU/SB(搜索区域1(SA1),或非局部搜索区域),可用于局部IBC参考块的搜索区域包括标记为“2”和“0”的CTU/SB(该搜索区域可被称为搜索区域2(SA2),或局部搜索区域),并且用于IBC参考块的限制出/禁止区域包括由于写回时延的标记为“3”的CTU/SB。在一些其它实现中,使用足够的可以保持整个受限的CTU/SB的片上RSM大小,所有这些潜在限制区域都可包括在RSM中以供局部参考。例如,标记为“2”和“3”的左相邻块都可以包括在局部搜索区域中。

[0206] 在一些其它实施中,只有标记为“0”的当前CTU/SB或当前CTU/SB的一部分可以被包括在RSM中用于局部参考。

[0207] 在一些示例实施中,SA1中的样本可以存储在外部存储器中。

[0208] 在一些示例实施中,SA2中的样本可以存储在片上存储器中。

[0209] 在一些示例实施中,外部存储器和片上存储器具有不同的硬件特性,例如访问速度、访问时钟、访问带宽等。

[0210] 当执行帧内块复制(IntraBC)预测时,当块矢量指向部分地位于SA1且部分地位于SA2中的块时,可能发生特殊情况。在这种特殊情况下,在将该块用作预测块之前,可能需要应用进一步的限制或处理。

[0211] 在一些示例实施中,在这种特殊情况下,不允许或排除由块矢量指向的块被用作IntraBC的预测块。

[0212] 图26示出了由各个块矢量指向的各种示例块。块A不被允许用作预测块,因为它与SA1和SA2都重叠;块B被允许用作预测块,因为它被完全包括在SA2中;并且块C被允许用作预测块,因为它被完全包括在SA1中。

[0213] 在一些示例实施中,如果IntraBC的块矢量指向部分位于SA1且部分位于SA2的块(由B表示),则建议替换B中与SA1重叠的样本或替换B中与SA2重叠的样本。样本的替换可以通过扩展可用于预测的边界样本来完成。例如,为了替换B中与SA1重叠的样本,可以使用SA2中的边界样本;为了替换B中与SA2重叠的样本,可以使用SA1中的边界样本。

[0214] 在一些示例实施中,重叠区域大小可以用于确定样本替换应该应用于哪个重叠区域。如果B和SA1之间的重叠区域大小大于B和SA2之间的重叠区域大小,则替换位于B和SA2之间的重叠部分中的样本,反之亦然。

[0215] 在一些示例实施中,样本的数目可以用于确定样本替换应该应用于哪个重叠区域。将B和SA1之间的重叠部分覆盖的样本数目表示为 $S_1$ ,将B和SA2之间的重叠部分覆盖的样本数目表示为 $S_2$ 。如果 $S_1$ 大于 $S_2$ 乘以加权因子( $t_1$ ) (即, $S_1 > S_2 * t_1$ ),则替换位于B和SA2之间的重叠部分中的样本,其中 $t_1$ 可以是预定义的,或者是动态地用信号通知的。类似地,如果 $S_2$ 大于 $S_1$ 乘以加权因子( $t_2$ ),则替换位于B和SA1之间的重叠部分中的样本,其中 $t_2$ 可以是预定义的,或者是动态地用信号通知的。

[0216] 在一些示例实施中,当样本替换应用于B和SA1之间的重叠部分时,SA2中的样本可用于替换重叠部分中的样本。类似地,当样本替换施加在B和SA2之间的重叠部分上时,SA1中的样本可以用于替换重叠部分中的样本。

[0217] 在一些示例实施中,如图26所示,CTU/SB 2602和2604形成不允许区域。然而,在一些其他实现中,CTU/SB 2602和/或2604也可以是局部搜索区域(或相邻允许搜索区域,SA2)的一部分。例如,如果片上存储器大到足以保持/容纳2602和/或2604中的样本。

[0218] 本公开描述了用于视频编码/解码的方法、装置和计算机可读介质。本公开解决了IntraBC的各种问题。本公开中描述的方法、设备和计算机可读介质可以增强视频编解码器的性能,优化编解码器中片上存储器的使用,并且有助于选择和调整IntraBC的预测区域。

[0219] 图27示出了用于处理视频数据的示例性方法2700。方法1900可以包括以下步骤的部分或全部:步骤2710,接收视频比特流,所述视频比特流包括视频帧的至少一个块,所述至少一个块包括当前块;步骤2720,确定所述视频帧中的第一搜索区域,其中,所述第一搜索区域是用于定位帧内块复制(IntraBC)预测块的第一候选区域,其中,所述第一搜索区域与所述当前块不重叠、并且包括块列表,所述IntraBC预测块是用于对所述当前块的至少部

分进行IntraBC预测的候选块;步骤2730,确定第二搜索区域,其中,所述第二搜索区域是用于定位所述IntraBC预测块的第二候选区域,所述第二搜索区域包括以下至少之一:(i)所述当前块的子块,以及(ii)所述当前块的相邻块;步骤2740,识别由块矢量所参考的所述IntraBC预测块;步骤2750,基于所述IntraBC预测块、所述第一搜索区域和所述第二搜索区域之间的空间关系来确定预测策略;以及步骤2760,基于所确定的预测对所述当前块的至少部分进行解码。

[0220] 在本公开的实施例和实现中,可以根据需要以任何数量或顺序组合或布置任何步骤和/或操作。步骤和/或操作中的两个或更多个可以并行执行。本公开中的实施例和实现可以单独使用或以任何顺序组合使用。此外,方法(或实施例)、编码器和解码器中的每一个都可以通过处理电路(例如,一个或多个处理器或一个或多个集成电路)来实现。在一个示例中,一个或多个处理器执行存储在非暂时性计算机可读介质中的程序。本公开中的实施例可以应用于亮度块或色度块。术语‘块’可以被解释为预测块、编码块或编码单元,即CU。这里的术语‘块’也可以用来指代变换块。在以下项目中,当说块大小时,它可以指块的宽度或高度,或宽度和高度的最大值,或宽度和高度的最小值,或块的面积大小(宽度\*高度),或纵横比(宽度:高度,或高度:宽度)。

[0221] 上述技术可以使用计算机可读指令实现为计算机软件,并且物理地存储在一个或多个计算机可读介质中。例如,图28示出了适用于实现所公开主题的某些实施例的计算机系统(2800)。

[0222] 可以使用任何合适的机器代码或计算机语言来对计算机软件进行编码,任何合适的机器代码或计算机语言可以经受汇编、编译、链接或类似的机制以创建包括指令的代码,该指令可以由一个或多个计算机中央处理单元(CPU)、图形处理单元(GPU)等直接执行或通过解释、微代码等执行。

[0223] 这些指令可以在各种类型的计算机或其组件上执行,包括例如个人计算机、平板计算机、服务器、智能手机、游戏设备、物联网设备等。

[0224] 图28所示的计算机系统(2800)的组件在本质上是示例性的,并且不旨在对实现本公开的实施例的计算机软件的使用范围或功能提出任何限制。组件的配置也不应被解释为具有与计算机系统(2800)的示例性实施例中所示的任何一个组件或组件组合相关的任何依赖性要求。

[0225] 计算机系统(2800)可以包括某些人机接口输入设备。这样的人机接口输入设备可以通过例如触觉输入(例如:击键、滑动、数据手套运动)、音频输入(例如:语音、拍手)、视觉输入(例如:手势)、嗅觉输入(未示出)来响应一个或多个个人机接口用户的输入。人机接口设备还可以用于捕捉不一定与人类的有意识输入直接相关的某些媒体,例如音频(例如:语音、音乐、环境声音)、图像(例如:扫描图像、从静止图像相机获得的摄影图像)、视频(例如二维视频、包括立体视频的三维视频)。

[0226] 输入人机接口设备可以包括键盘(2801)、鼠标(2802)、触控板(2803)、触摸屏(2810)、数据手套(未示出)、操纵杆(2805)、麦克风(2806)、扫描仪(2807)、照相机(2808)中的一个或多个(每种仅示出一个)。

[0227] 计算机系统(2800)还可以包括某些人机接口输出设备。这种人机接口输出设备可以通过例如触觉输出、声音、光和嗅觉/味觉来刺激一个或多个人类用户的感官。这样的人

机接口输出设备可以包括触觉输出设备(例如触摸屏(2810)的触觉反馈,数据手套(未示出)或操纵杆(2805),但也可以有不用作输入设备的触觉反馈设备)、音频输出设备(例如:扬声器(2809)、耳机(未示出))、视觉输出设备(例如屏幕(2810),包括CRT屏幕、LCD屏幕、等离子屏幕、OLED屏幕,每个具有或不具有触摸屏输入能力,每个具有或不具有触觉反馈能力——其中一些可以能够通过诸如立体输出的装置输出二维视觉输出或大于三维输出;虚拟现实眼镜(未描绘)、全息显示器和烟雾罐(未描绘))以及打印机(未描绘)。

[0228] 计算机系统(2800)也可以包括人类可访问存储装置及其关联介质:例如包括具有CD/DVD等介质(2821)的CD/DVD ROM/RW(2820)的光学介质、指状驱动器(2822)、可拆卸硬盘驱动器或固态驱动器(2823)、诸如磁带和软盘之类的传统磁性介质(未示出)、诸如安全软件狗之类的基于专用ROM/ASIC/PLD的装置(未示出)等。

[0229] 本领域技术人员还应该理解,结合当前公开的主题使用的术语“计算机可读介质”不涵盖传输介质、载波或其他瞬时信号。

[0230] 计算机系统(2800)还可以包括到一个或多个通信网络(2855)的接口(2854)。网络例如可以是无线网络、有线网络、光网络。网络还可以是本地网络、广域网络、城域网络、车辆和工业网络、实时网络、耐延迟网络等。网络的示例包括诸如以太网之类的局域网、无线LAN、包括GSM、3G、4G、5G、LTE等的蜂窝网络、包括有线电视、卫星电视和地面广播电视的电视有线或无线广域数字网络、包括CANBus的车辆和工业用电视等等。某些网络通常需要连接到某些通用数据端口或外围总线(2849)的外部网络接口适配器(例如计算机系统(2800)的USB端口);如下所述,其他网络接口通常通过连接到系统总线而集成到计算机系统(2800)的内核中(例如,连接PC计算机系统以太网接口或连接到智能手机计算机系统蜂窝网络接口)。计算机系统(2800)可以使用这些网络中的任何一个与其他实体通信。此类通信可以是仅单向接收的(例如,广播电视)、仅单向发送的(例如,连接到某些CANbus装置的CANbus)或双向的,例如,使用局域网或广域网数字网络连接到其他计算机系统。如上所述,可以在那些网络和网络接口的每一个上使用某些协议和协议栈。

[0231] 前述人机接口设备、人机交互存储设备和网络接口可以连接到计算机系统(2800)的内核(2840)。

[0232] 内核(2840)可以包括一个或多个中央处理单元(CPU)(2841)、图形处理单元(GPU)(2842)、现场可编程门区域(FPGA)(2843)形式的专用可编程处理单元、用于某些任务的硬件加速器(2844)、图形适配器(2850)等。这些装置以及只读存储器(ROM)(2845)、随机存取存储器(2846)、诸如内部非用户可访问的硬盘驱动器、SSD等之类的内部大容量存储器(2847)可以通过系统总线(2848)连接。在一些计算机系统中,可以以一个或多个物理插头的形式访问系统总线(2848),以能够通过附加的CPU、GPU等进行扩展。外围装置可以直接连接到内核的系统总线(2848)或通过外围总线(2849)连接到内核的系统总线。在一个示例中,屏幕(2810)可以连接到图形适配器(2850)。外围总线的体系结构包括PCI、USB等。

[0233] CPU(2841)、GPU(2842)、FPGA(2843)和加速器(2844)可以执行某些指令,这些指令可以组合来构成上述计算机代码。该计算机代码可以存储在ROM(2845)或RAM(2846)中。过渡数据也可以存储在RAM(2846)中,而永久数据可以例如存储在内部大容量存储器(2847)中。可以通过使用高速缓存来进行到任何存储装置的快速存储及检索,该高速缓存可以与下述紧密关联:一个或多个CPU(2841)、GPU(2842)、大容量存储(2847)、ROM(2845)、RAM

(2846) 等。

[0234] 计算机可读介质可以在其上具有用于执行各种由计算机实现的操作的计算机代码。介质和计算机代码可以是出于本公开的目的而专门设计和构造的介质和计算机代码,或者介质和计算机代码可以是计算机软件领域的技术人员公知且可用的类型。

[0235] 作为非限制性示例,可以由于一个或多个处理器(包括CPU、GPU、FPGA、加速器等)执行包含在一种或多种有形的计算机可读介质中的软件而使得具有架构(2800),特别是内核(2840)的计算机系统提供功能。此类计算机可读介质可以是与如上所述的用户可访问的大容量存储相关联的介质,以及某些非暂时性的内核(2840)的存储器,例如内核内部大容量存储器(2847)或ROM(2845)。可以将实施本公开的各种实施例的软件存储在此类装置中并由内核(2840)执行。根据特定需要,计算机可读介质可以包括一个或多个存储装置或芯片。软件可以引起内核(2840),特别是其中的处理器(包括CPU、GPU、FPGA等)执行本文所描述的特定过程或特定过程的特定部分,包括定义存储在RAM(2846)中的数据结构以及根据由软件定义的过程来修改此类数据结构。附加地或替换地,可以由于硬连线或以其他方式体现在电路(例如,加速器(2844))中的逻辑而使得计算机系统提供功能,该电路可以替换软件或与软件一起运行以执行本文描述的特定过程或特定过程的特定部分。在适当的情况下,提及软件的部分可以包含逻辑,反之亦然。在适当的情况下,提及计算机可读介质的部分可以包括存储用于执行的软件的电路(例如集成电路(IC))、体现用于执行的逻辑的电路或两者都包括。本公开包括硬件和软件的任何合适的组合。

[0236] 尽管本公开已经描述了多个示例性实施例,但是存在落入本公开的范围内的修改、置换和各种替换等效物。因此,应当理解,本领域技术人员将能够设计出许多系统和方法,这些系统和方法虽然未在本文中明确示出或描述,但其体现了本公开的原理,因此落入本公开的精神和范围内。

[0237] 附录A:缩略语

[0238] IBC:帧内块复制

[0239] IntraBC:帧内块复制

[0240] JEM:联合探索模型

[0241] VVC:下一代视频编码

[0242] BMS:基准集

[0243] MV:运动矢量

[0244] HEVC:高效视频编码

[0245] SEI:补充增强信息

[0246] VUI:视频可用性信息

[0247] GOPs:图片群组

[0248] TUs:变换单元

[0249] PUs:预测单元

[0250] CTUs:编码树单元

[0251] CTBs:编码树块

[0252] PBs:预测块

[0253] HRD:假想参考解码器

- [0254] SNR:信噪比
- [0255] CPU:中央处理单元
- [0256] GPUs:图形处理单元
- [0257] CRT:阴极射线管
- [0258] LCD:液晶显示器
- [0259] OLED:有机发光二极管
- [0260] CD:光盘
- [0261] DVD:数字视频光盘
- [0262] ROM:只读存储器
- [0263] RAM:随机存取存储器
- [0264] ASIC:专用集成电路
- [0265] PLD:可编程逻辑器件
- [0266] LAN:局域网
- [0267] GSM:全球移动通信系统
- [0268] LTE:长期演进
- [0269] CANBus:控制器区域网络总线
- [0270] USB:通用串行总线
- [0271] PCI:互连外围设备
- [0272] FPGA:现场可编程门区域
- [0273] SSD:固态驱动器
- [0274] IC:集成电路
- [0275] HDR:高动态范围
- [0276] SDR:标准动态范围
- [0277] JVET:联合视频勘探组
- [0278] MPM:最可能模式
- [0279] WAIP:广角帧内预测
- [0280] CU:编码单元
- [0281] PU:预测单元
- [0282] TU:变换单元
- [0283] CTU:编码树单元
- [0284] PDPC:位置相关预测组合
- [0285] ISP:帧内子分区
- [0286] SPS:序列参数集
- [0287] PPS:图片参数集
- [0288] APS:自适应参数集
- [0289] VPS:视频参数集
- [0290] DPS:解码参数集
- [0291] ALF:自适应环路滤波器
- [0292] SAO:样本自适应偏移

- [0293] CC-ALF:跨分量自适应环路滤波器
- [0294] CDEF:约束方向增强滤波器
- [0295] CCS0:跨分量样本偏移
- [0296] LS0:局部样本偏移
- [0297] LR:环路恢复滤波器
- [0298] AV1:开放媒体联盟(AOMedia)视频1
- [0299] AV2:AOMedia视频2
- [0300] RPS:参考图片集
- [0301] DPB:解码图片缓冲区
- [0302] MMVD:具有运动矢量差的合并模式
- [0303] IntraBC or IBC:帧内块复制
- [0304] BV:块矢量
- [0305] BVD:块矢量差
- [0306] RSM:参考样本存储器

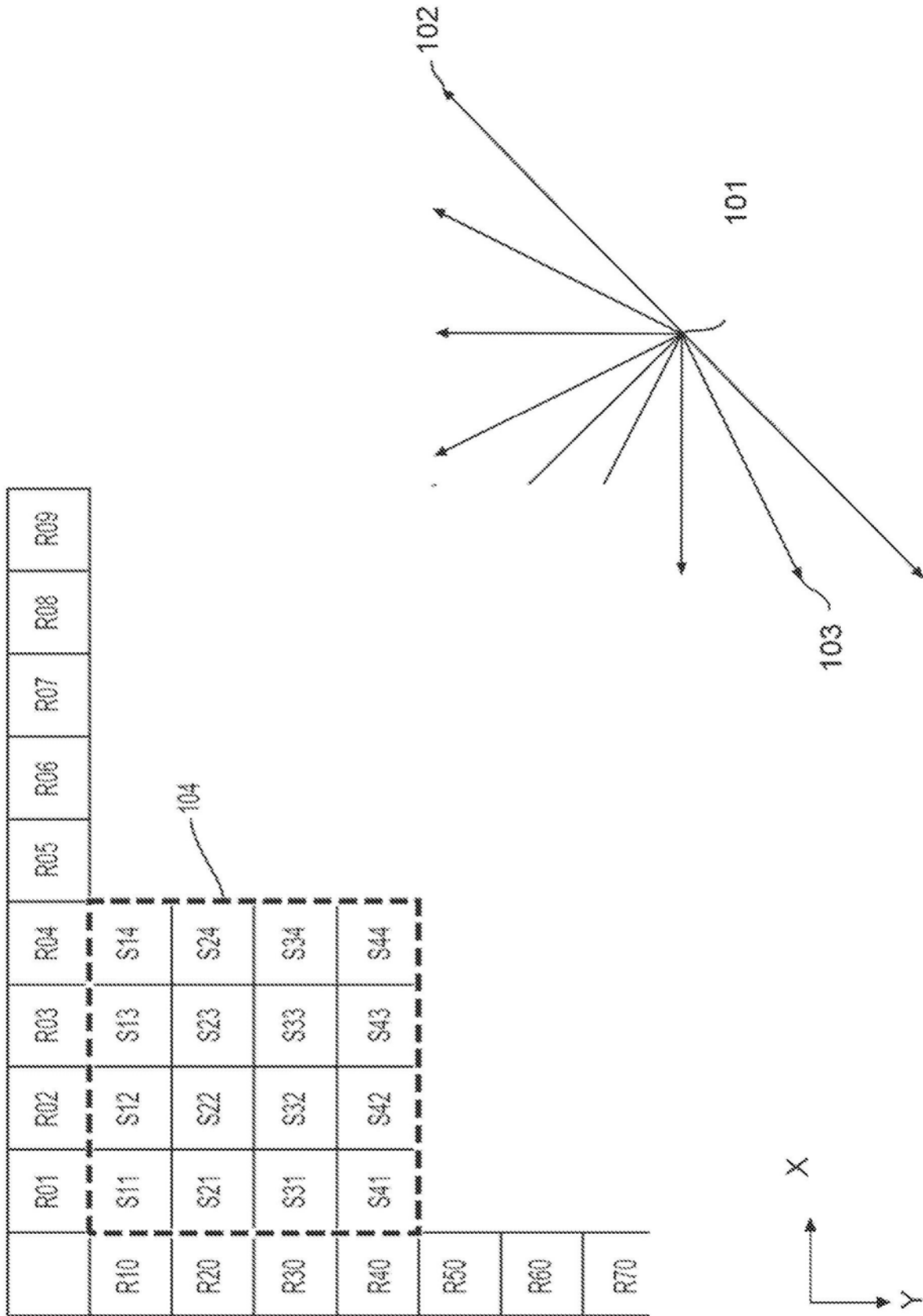


图1A

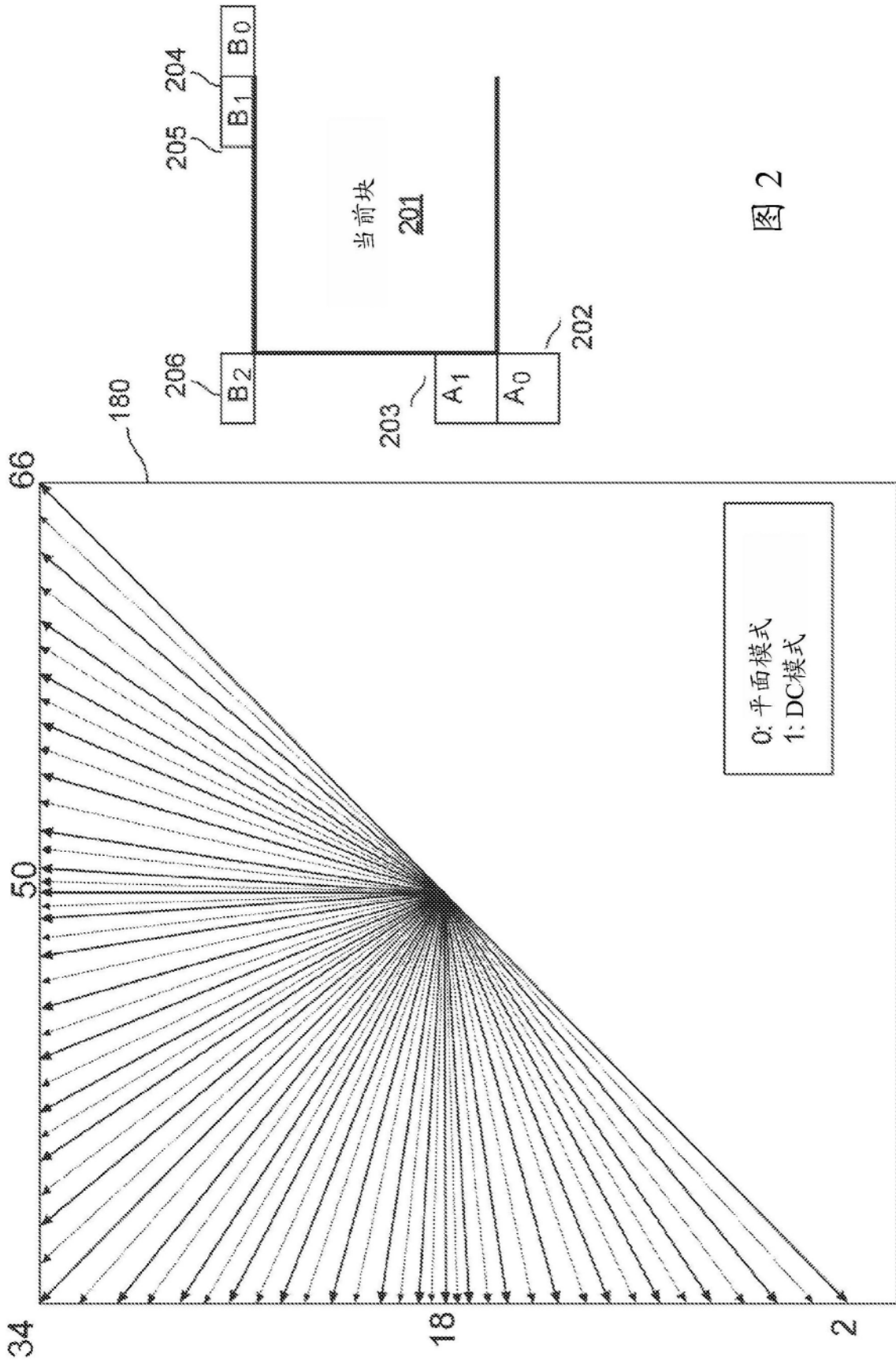


图 1B

图 2

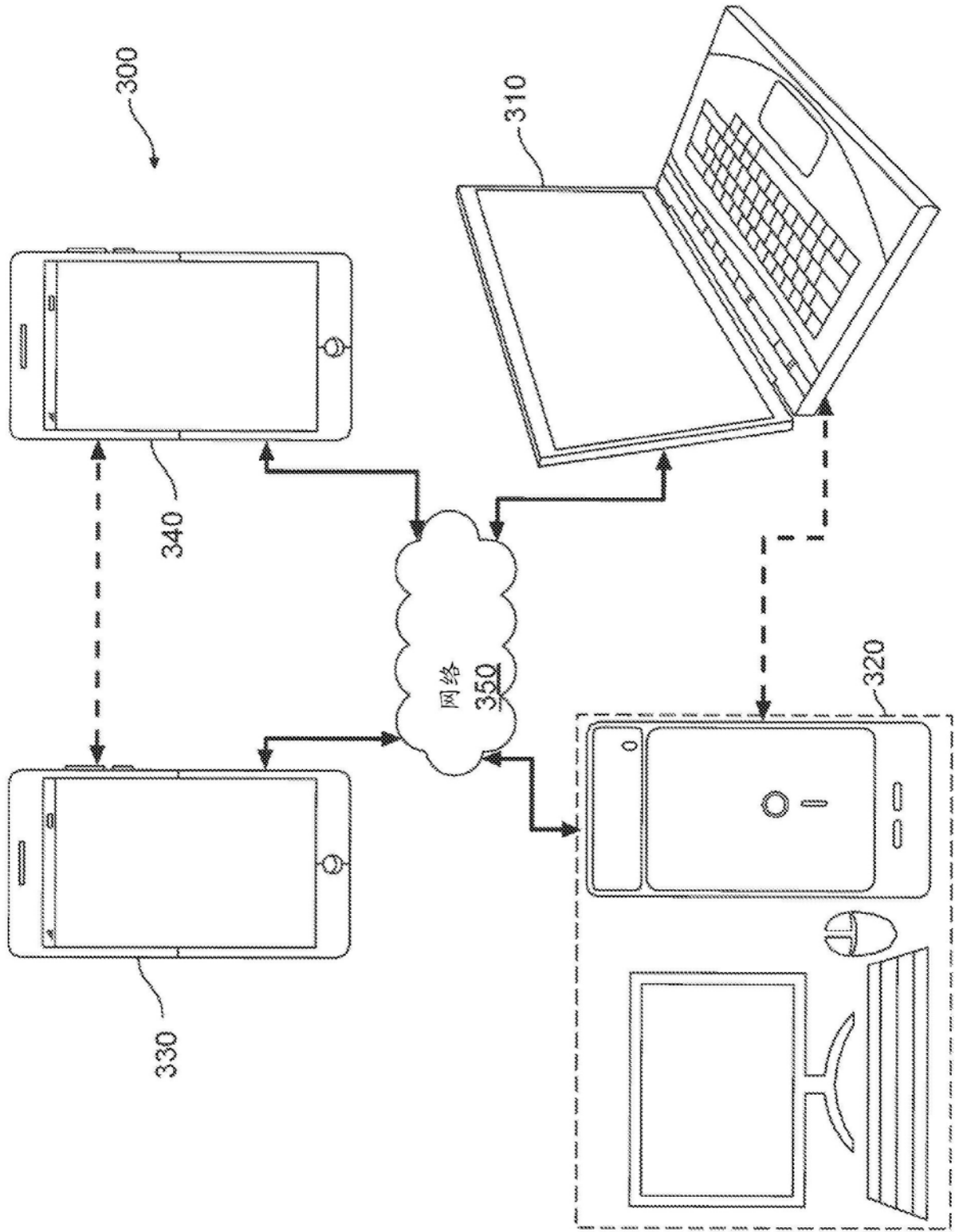


图3

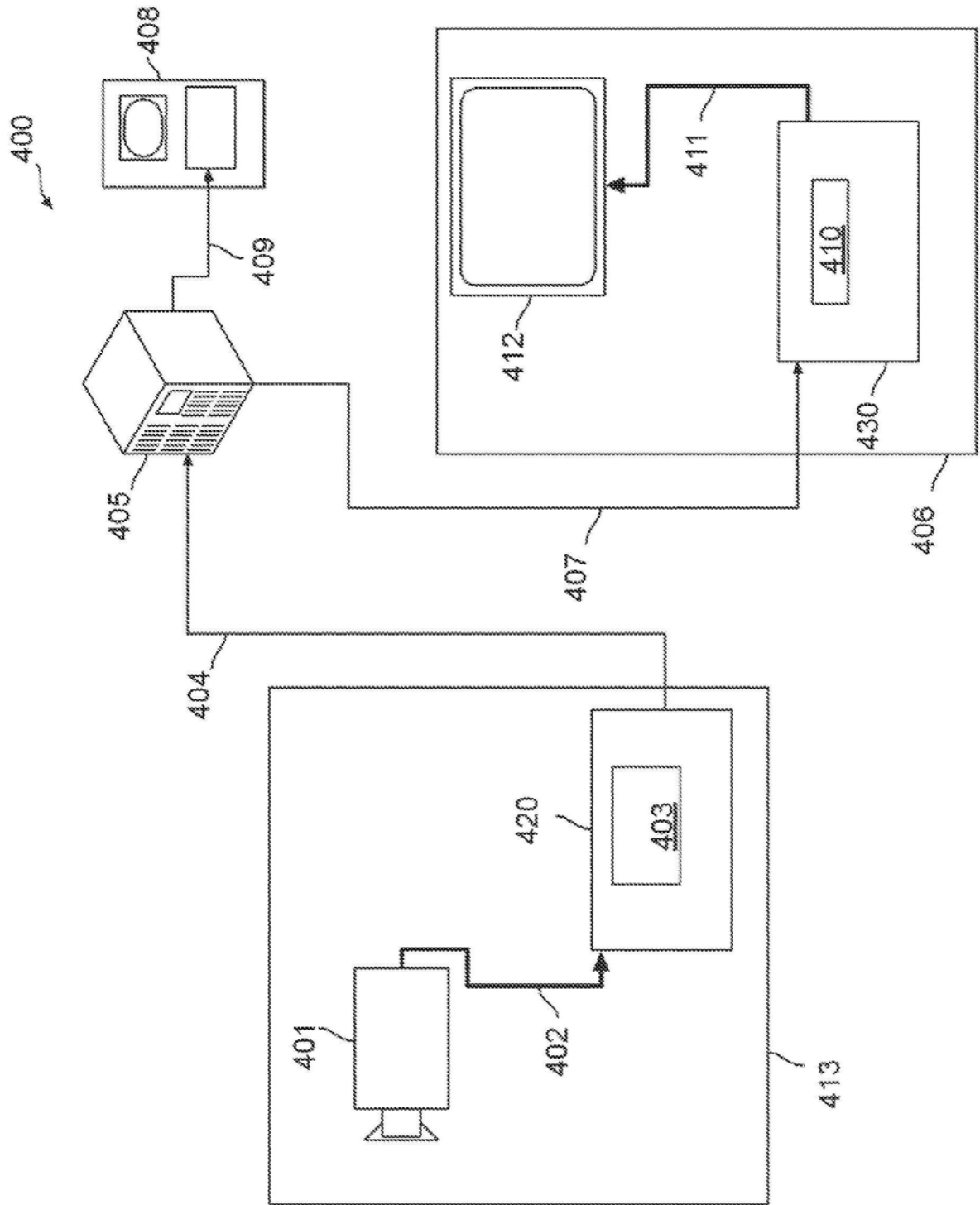


图4

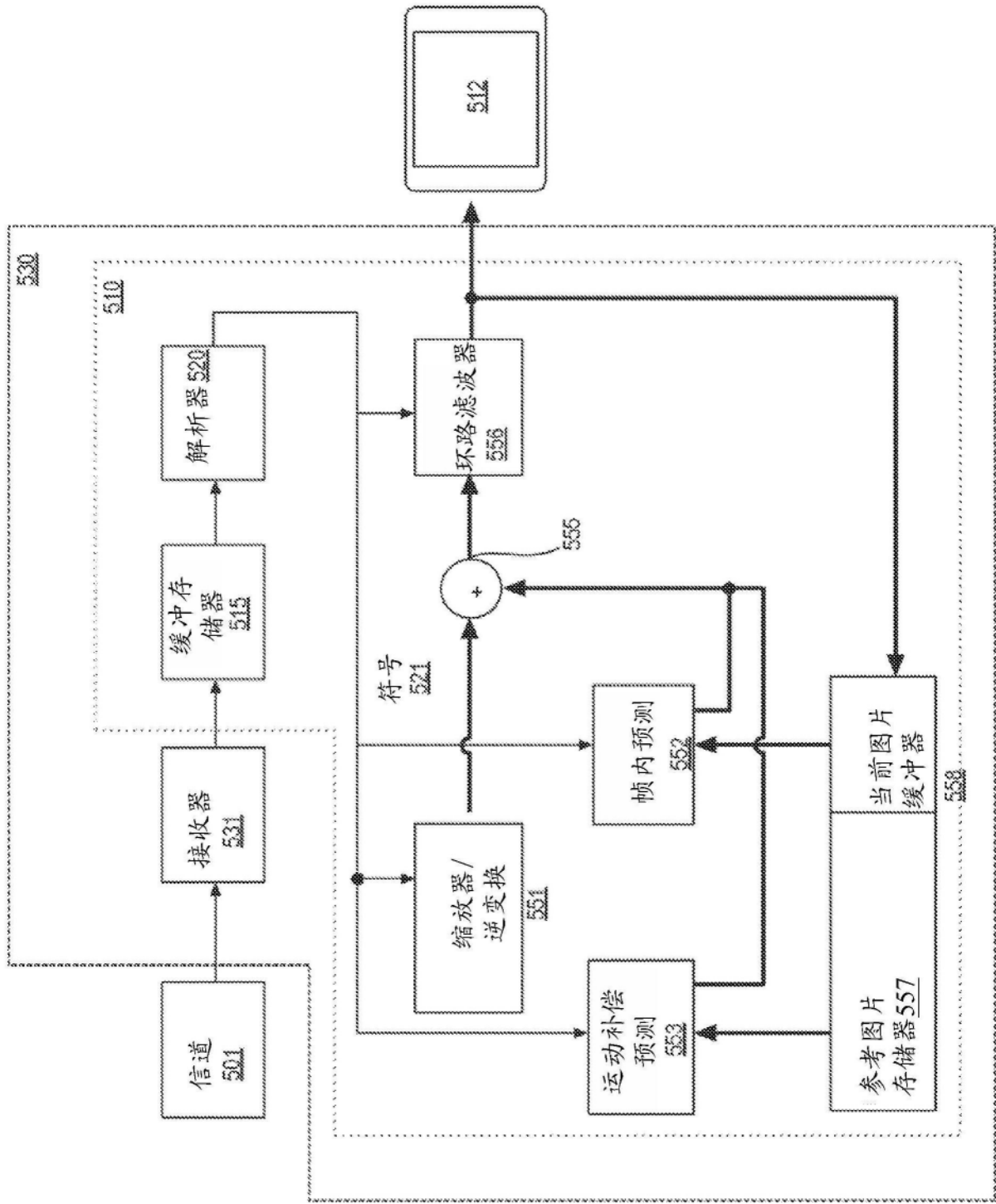


图5

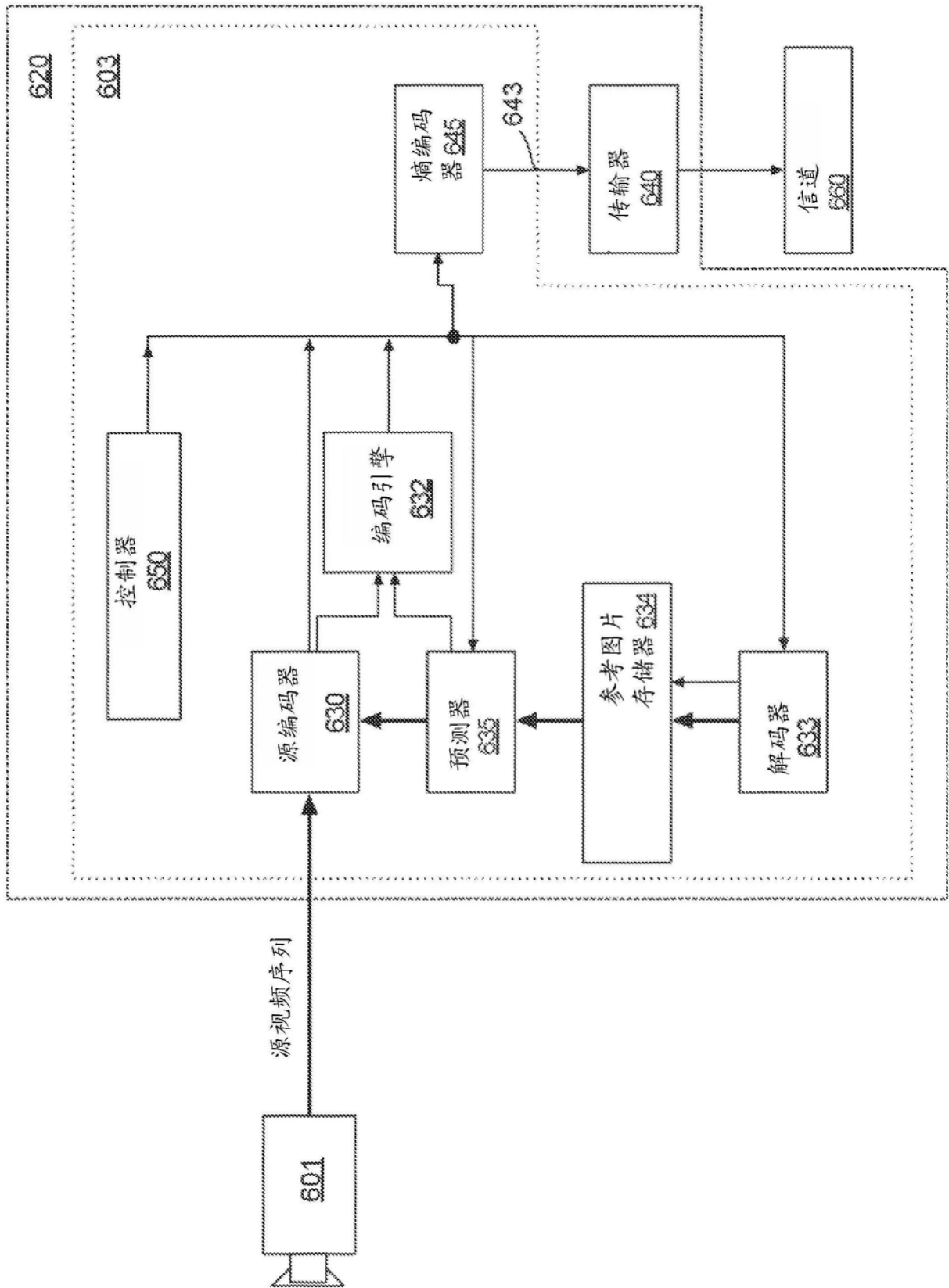


图6

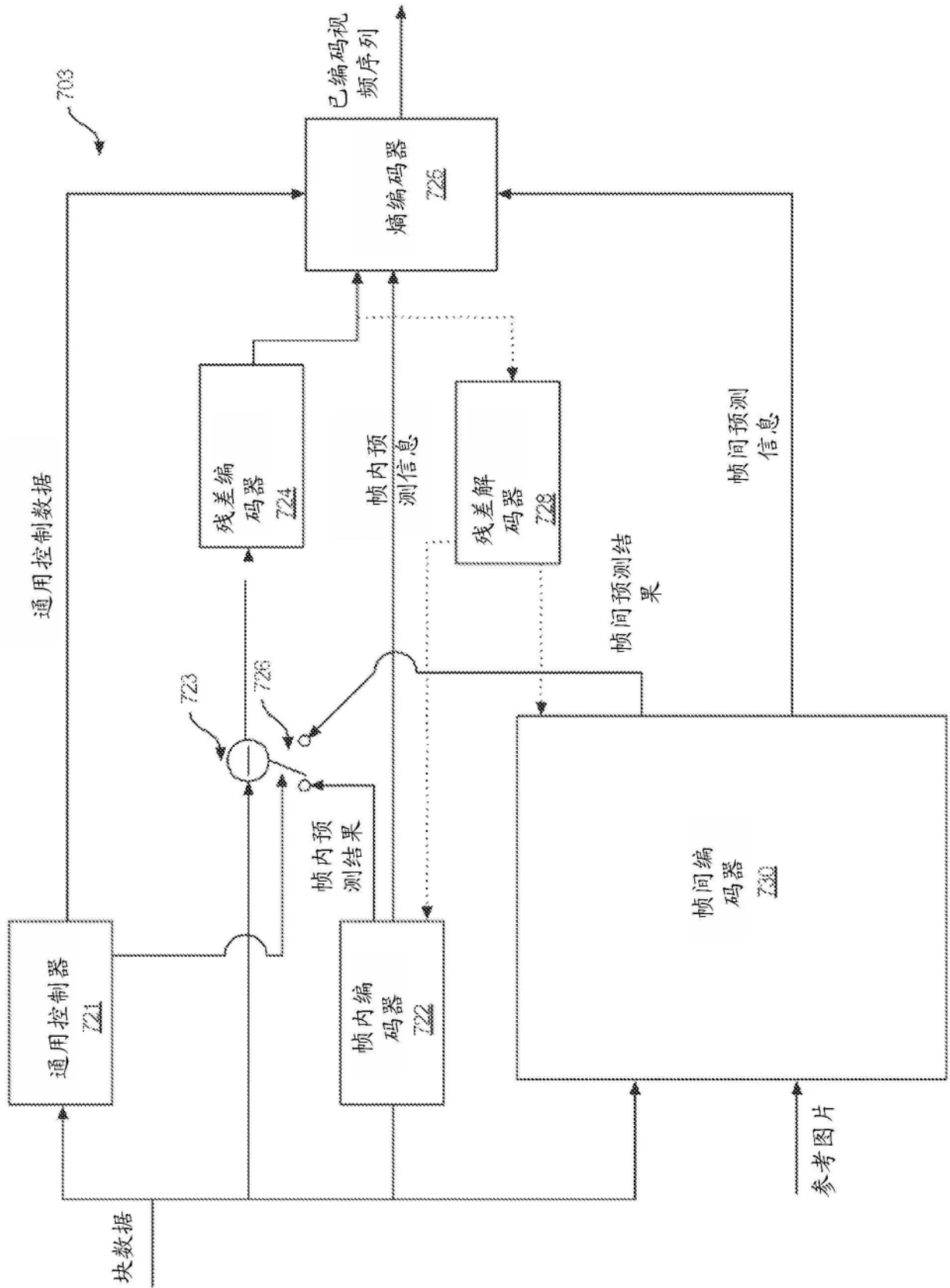


图7



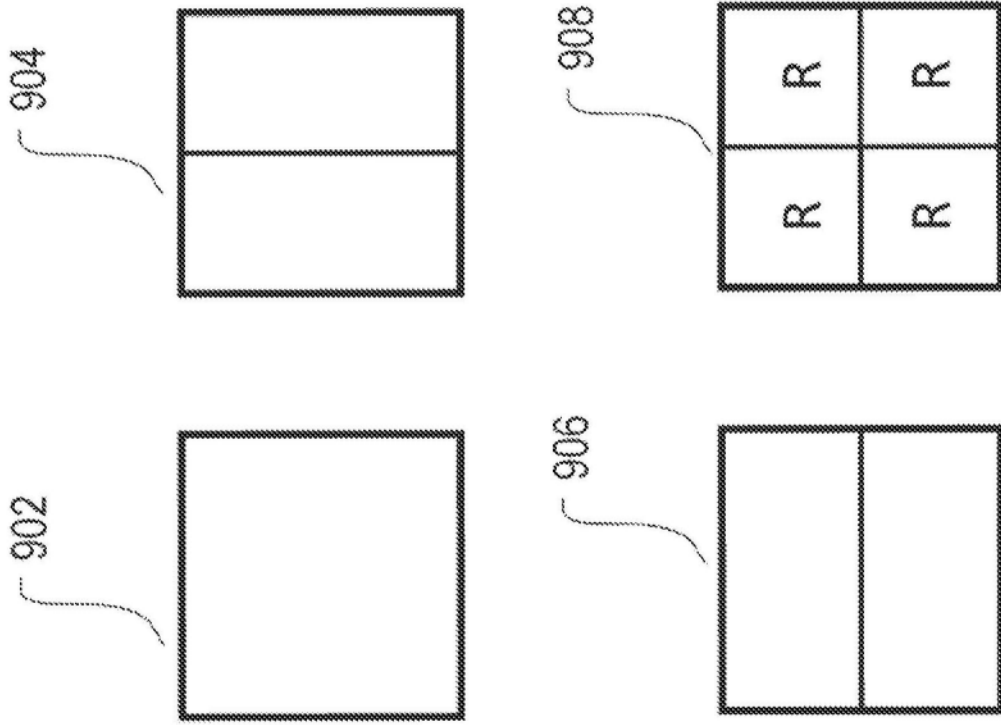


图9

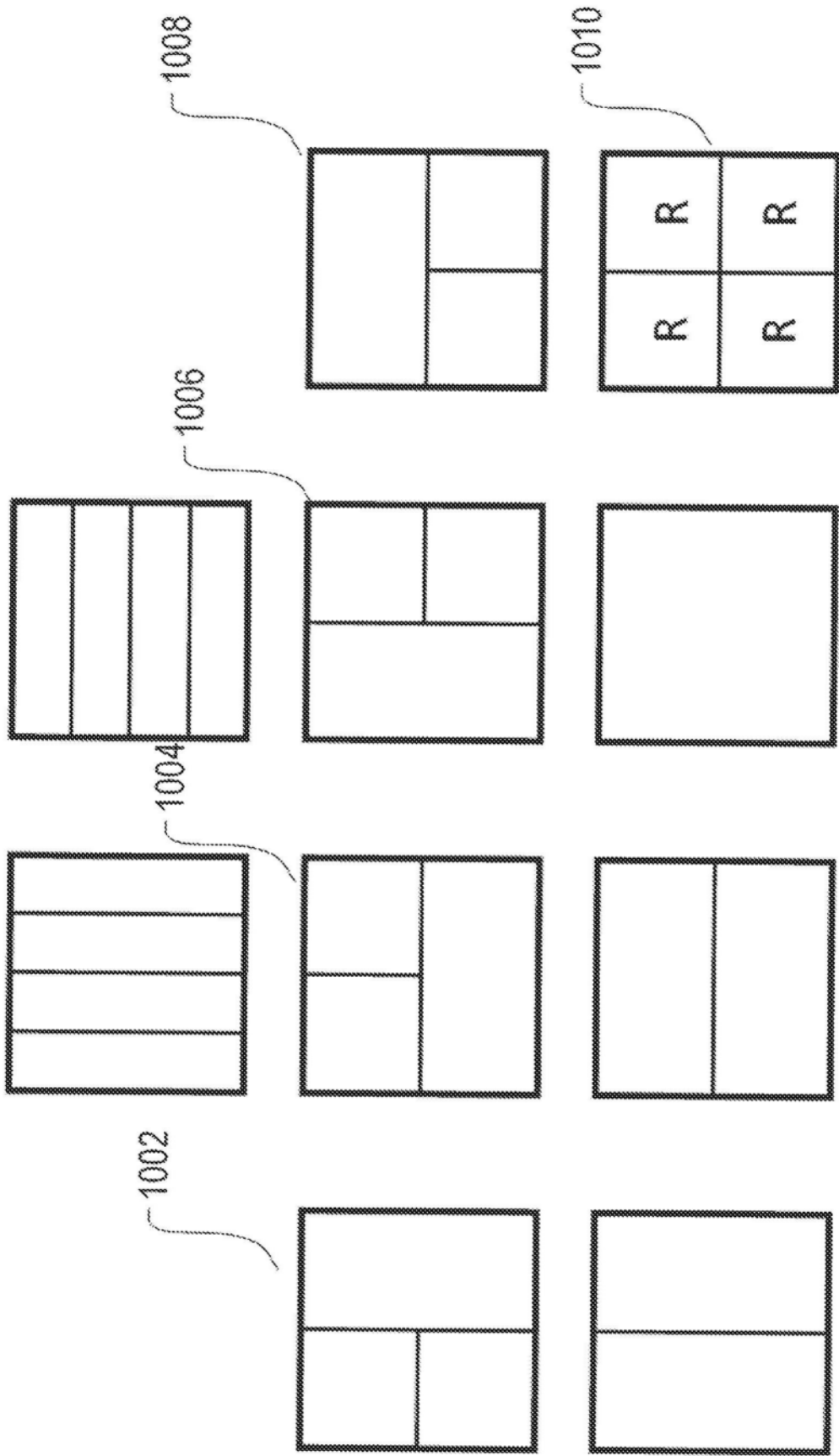


图10

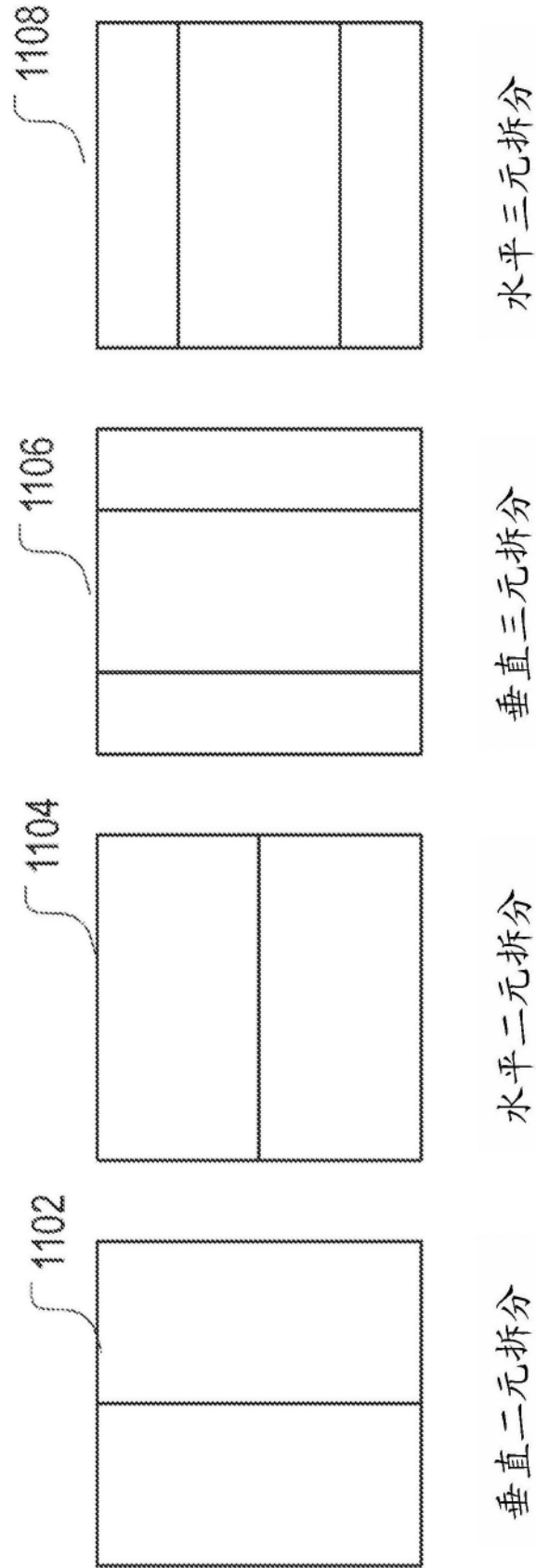


图11

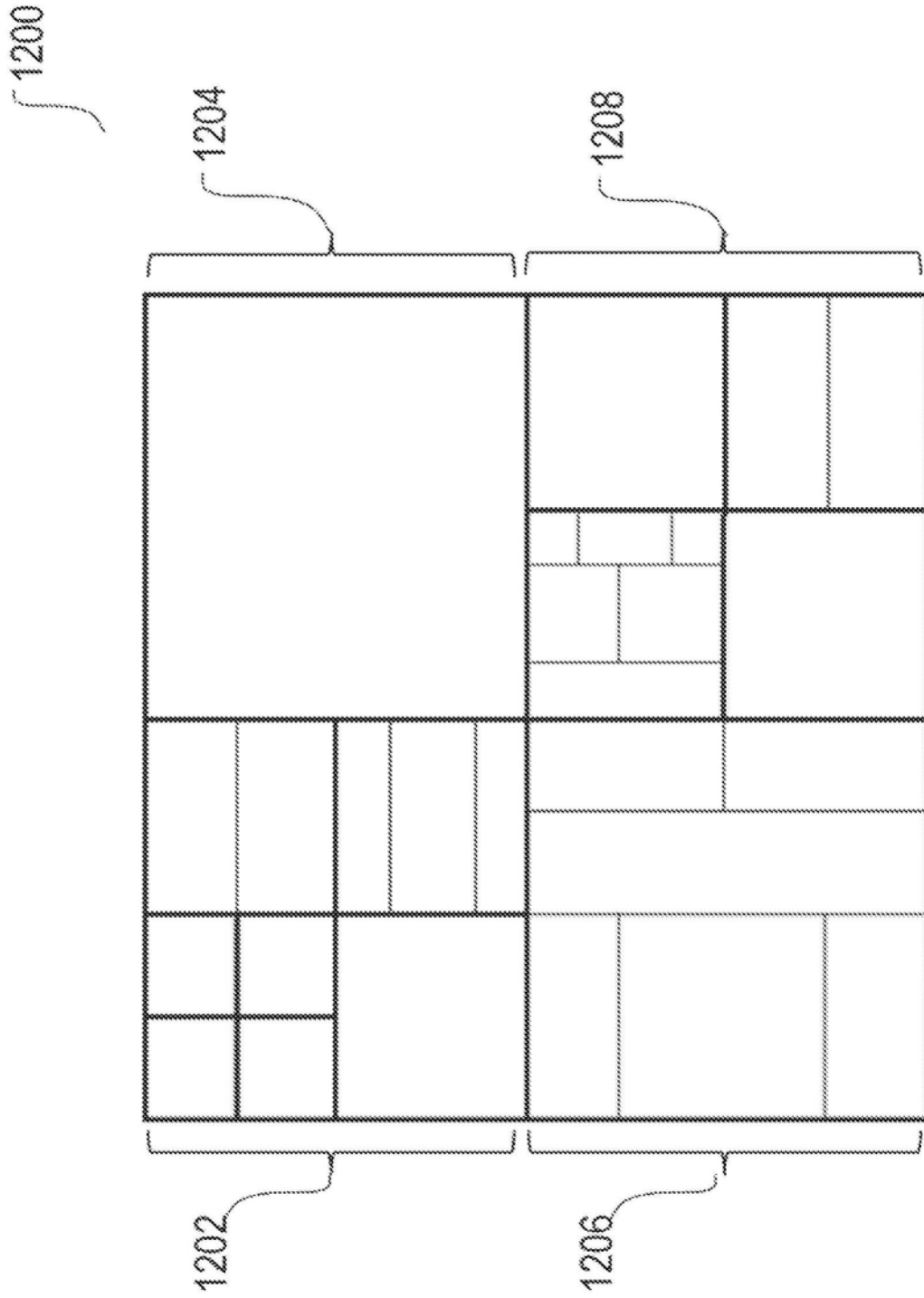


图12

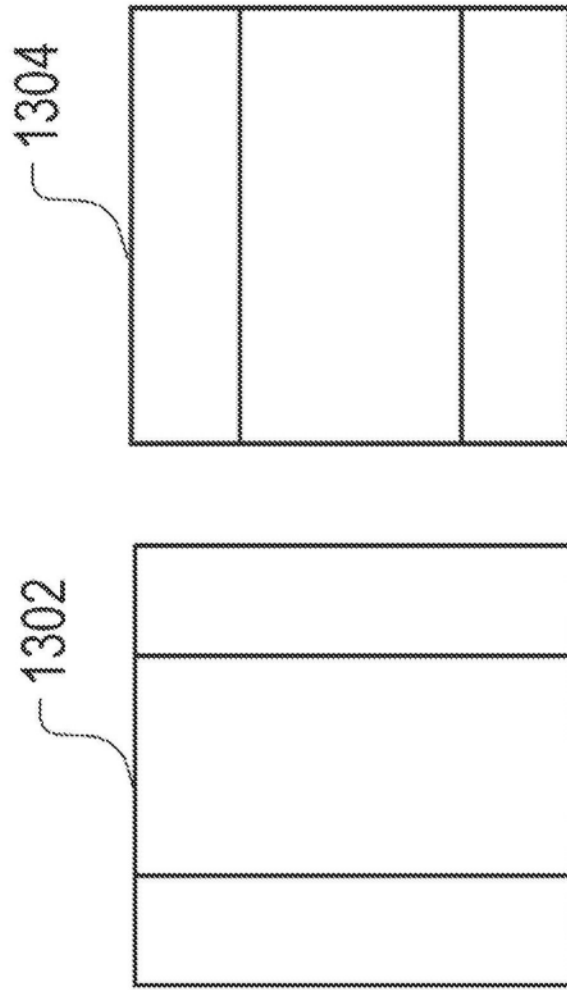


图13

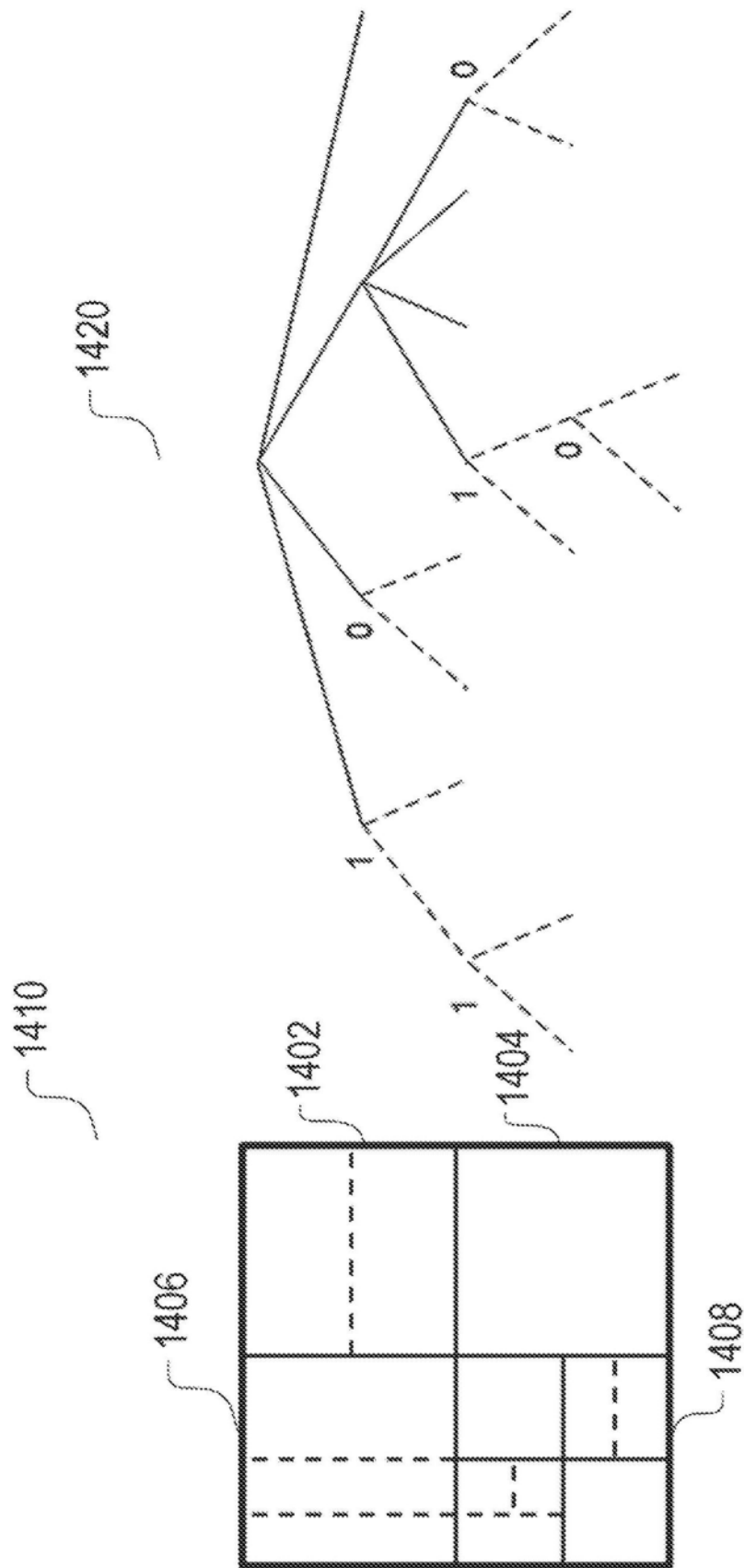


图14

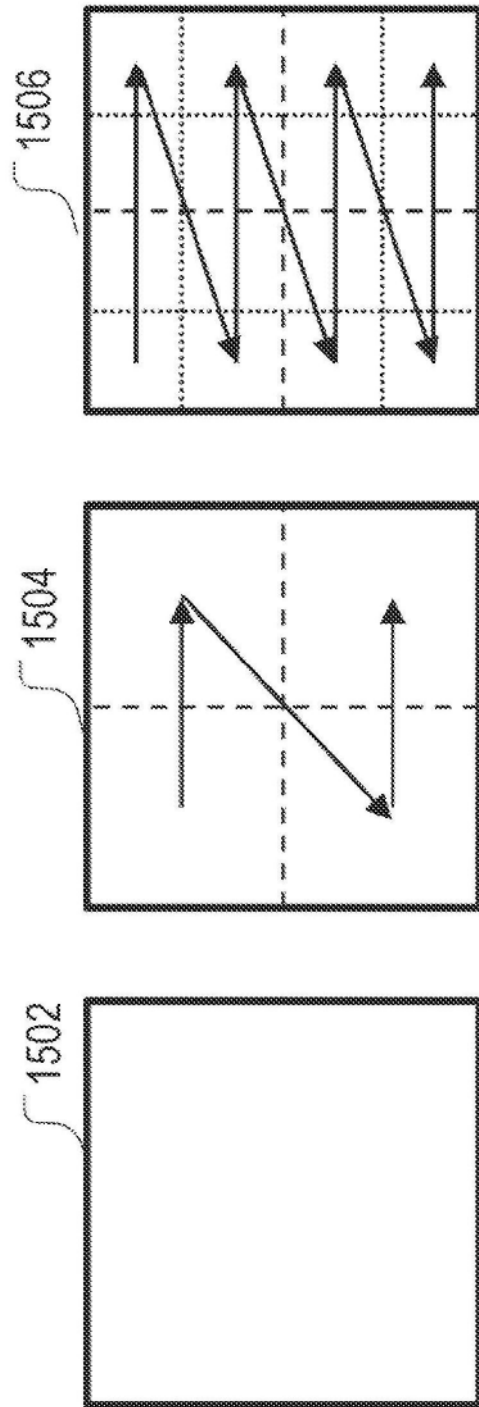


图15



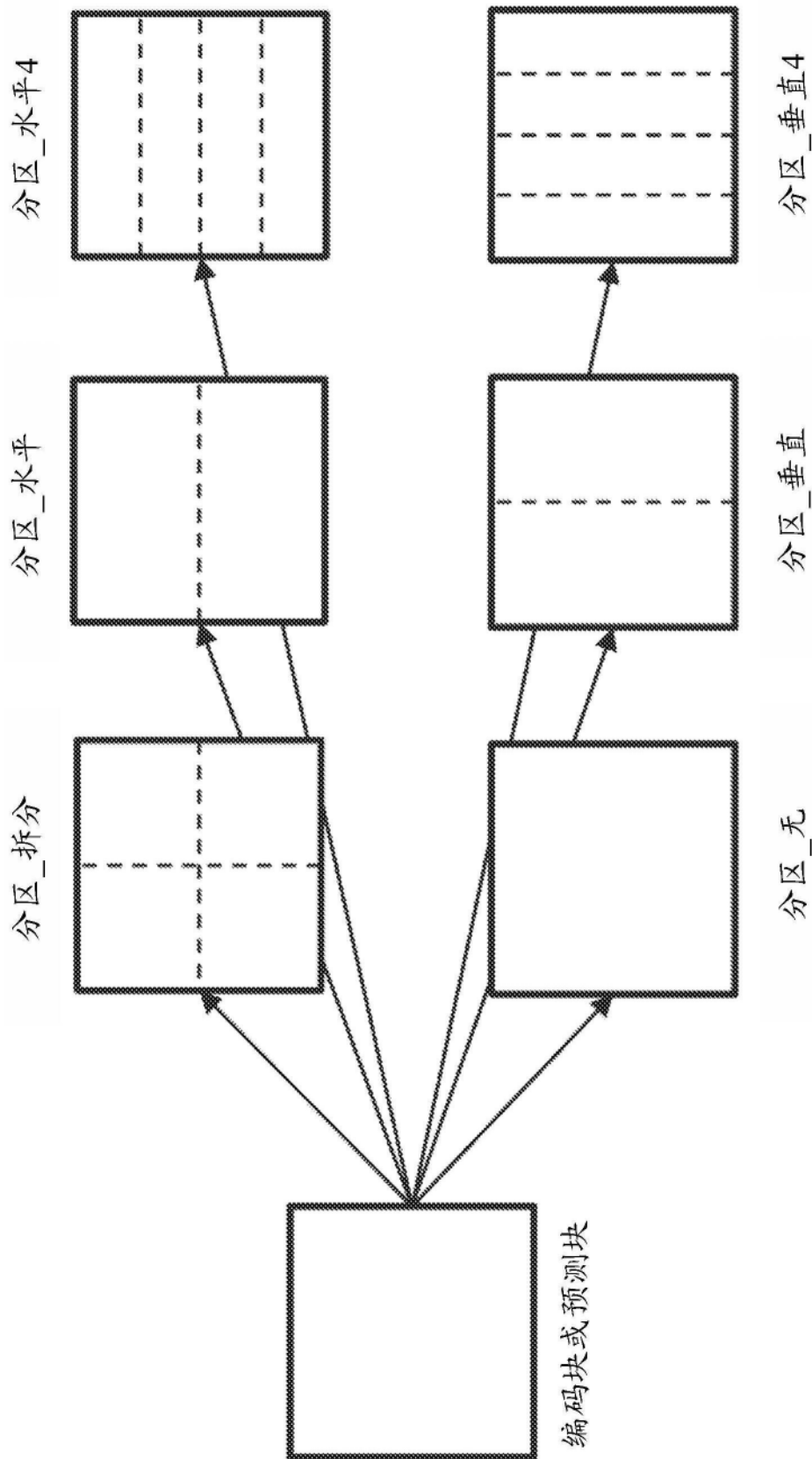


图17

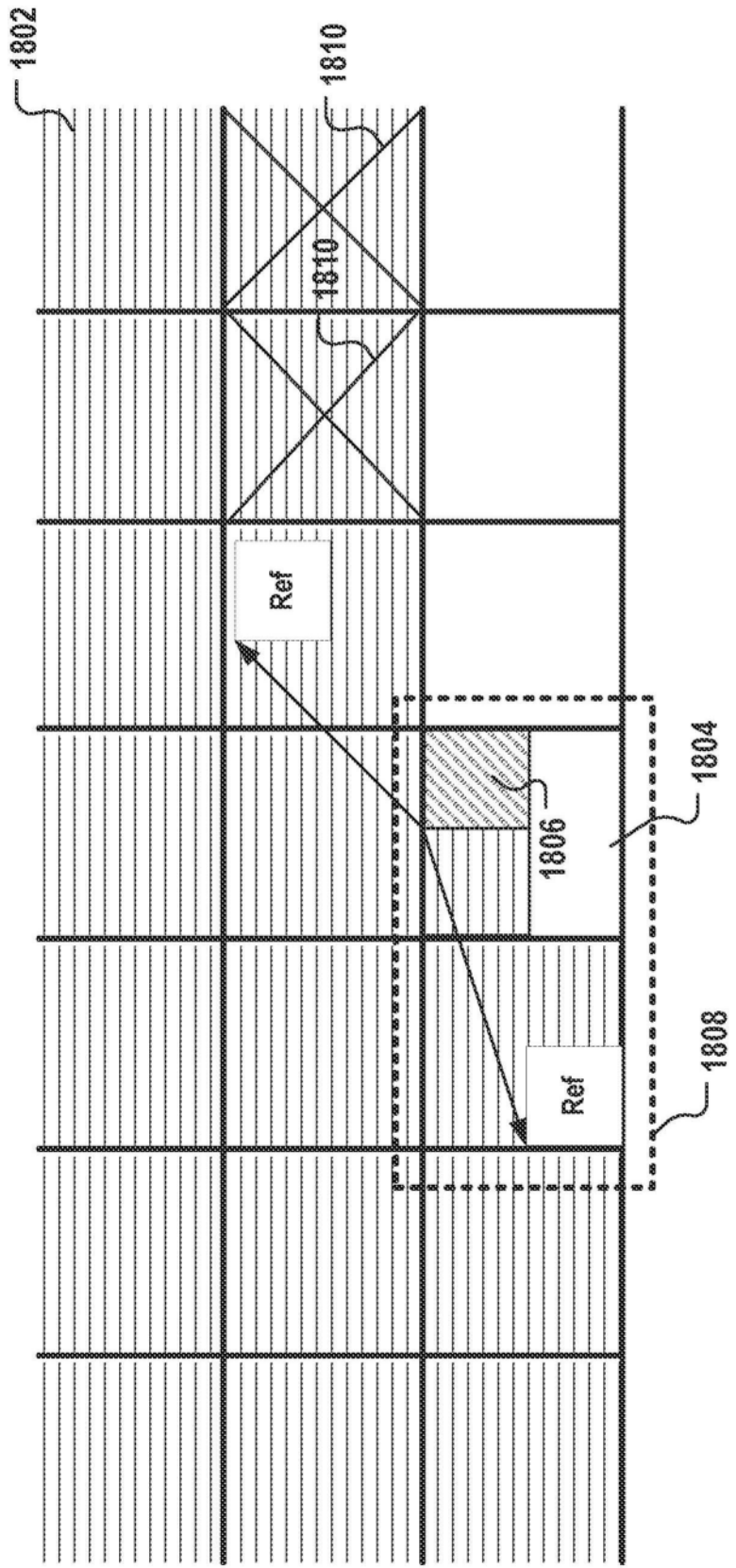


图18

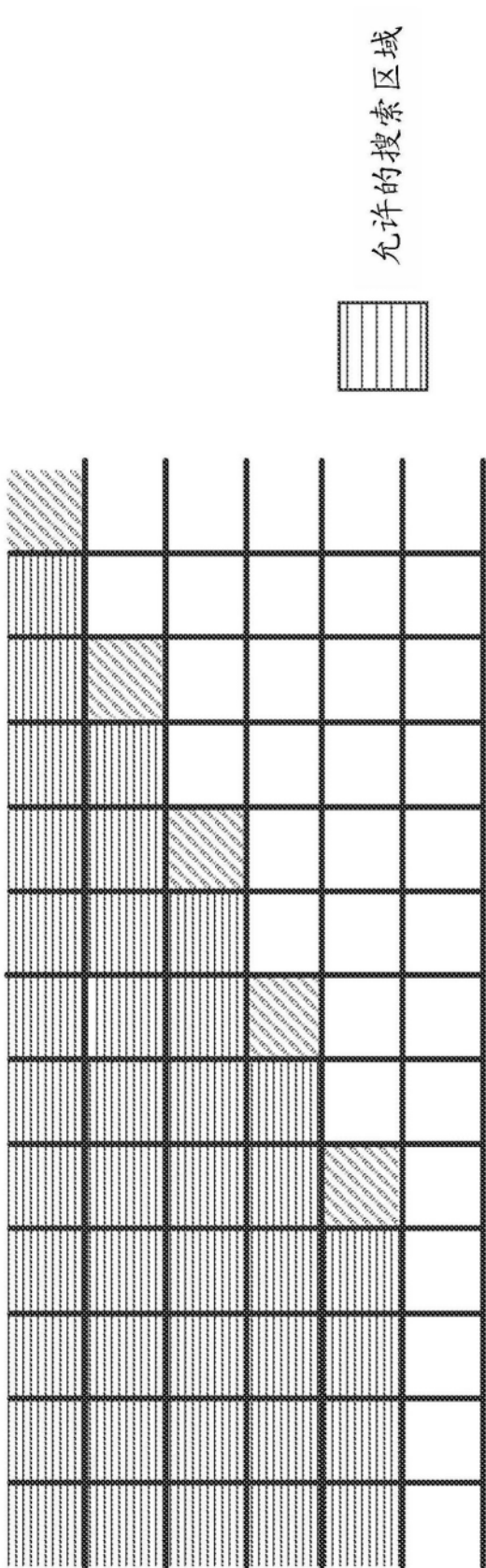


图 19

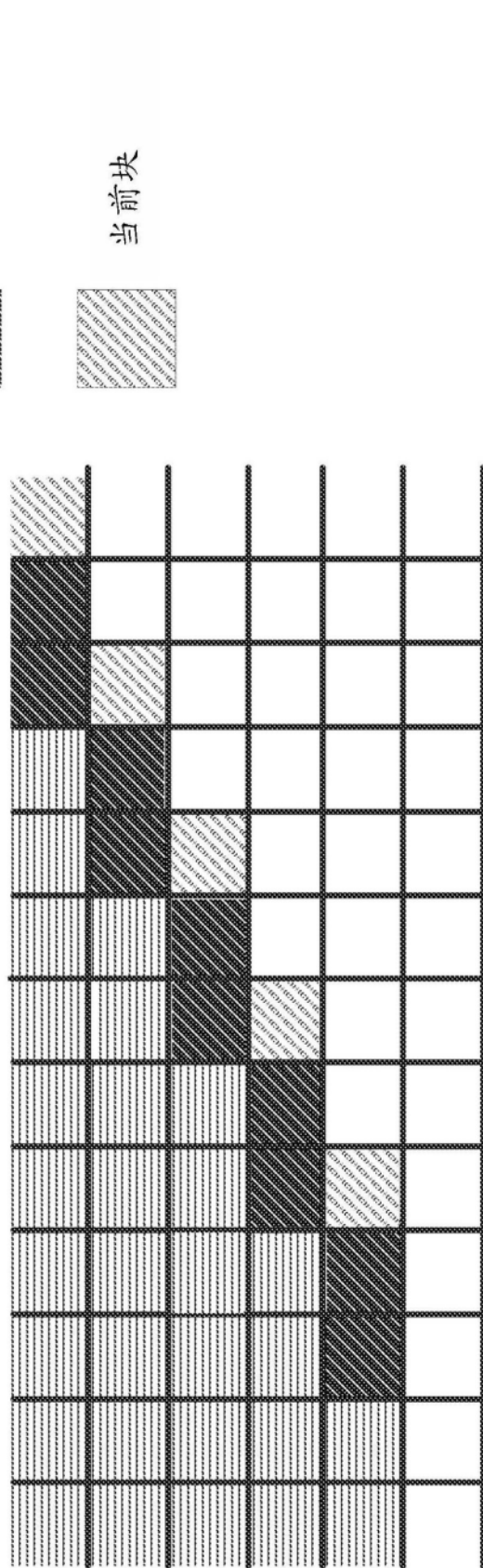


图 20

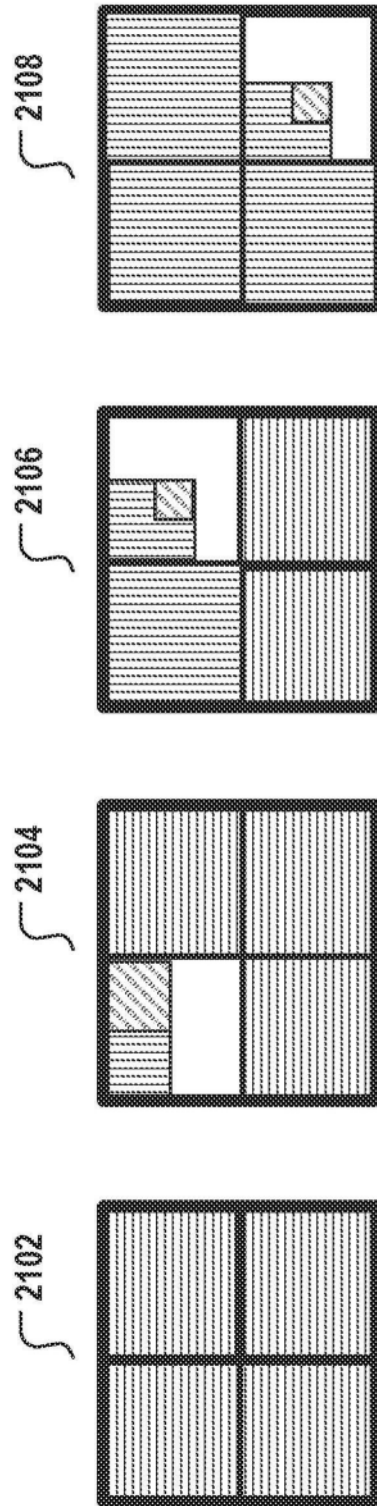


图21

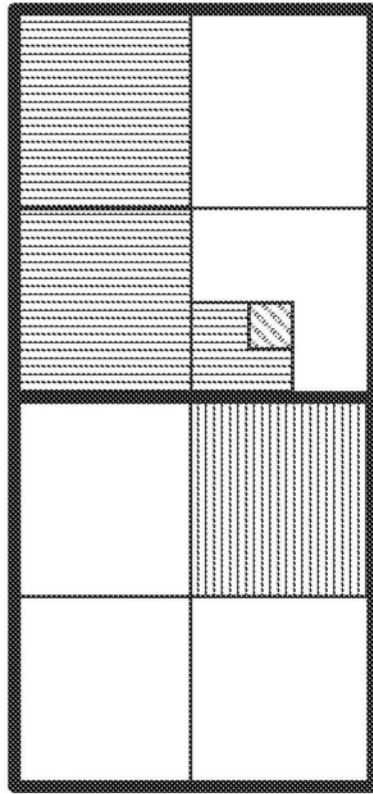


图22

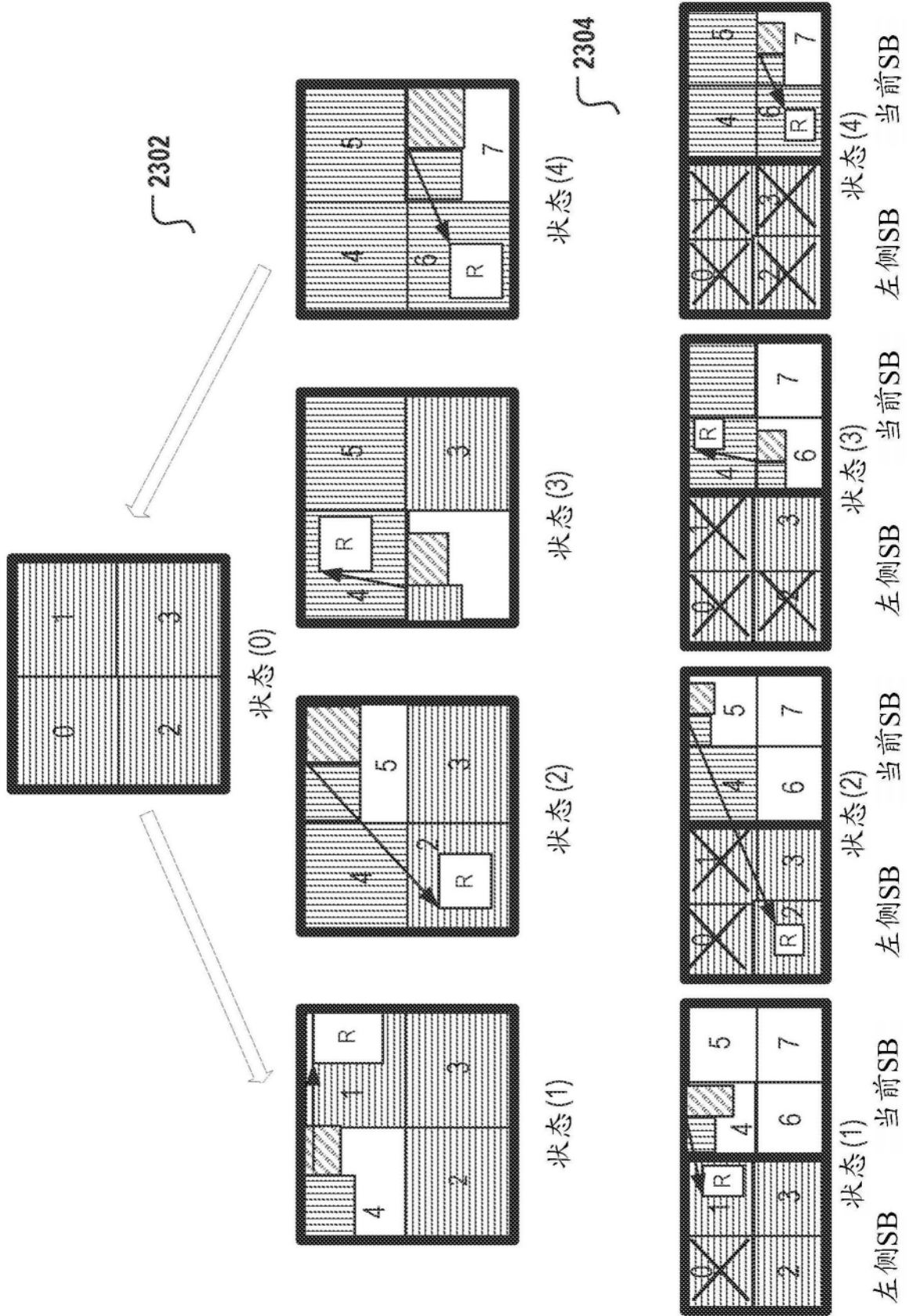
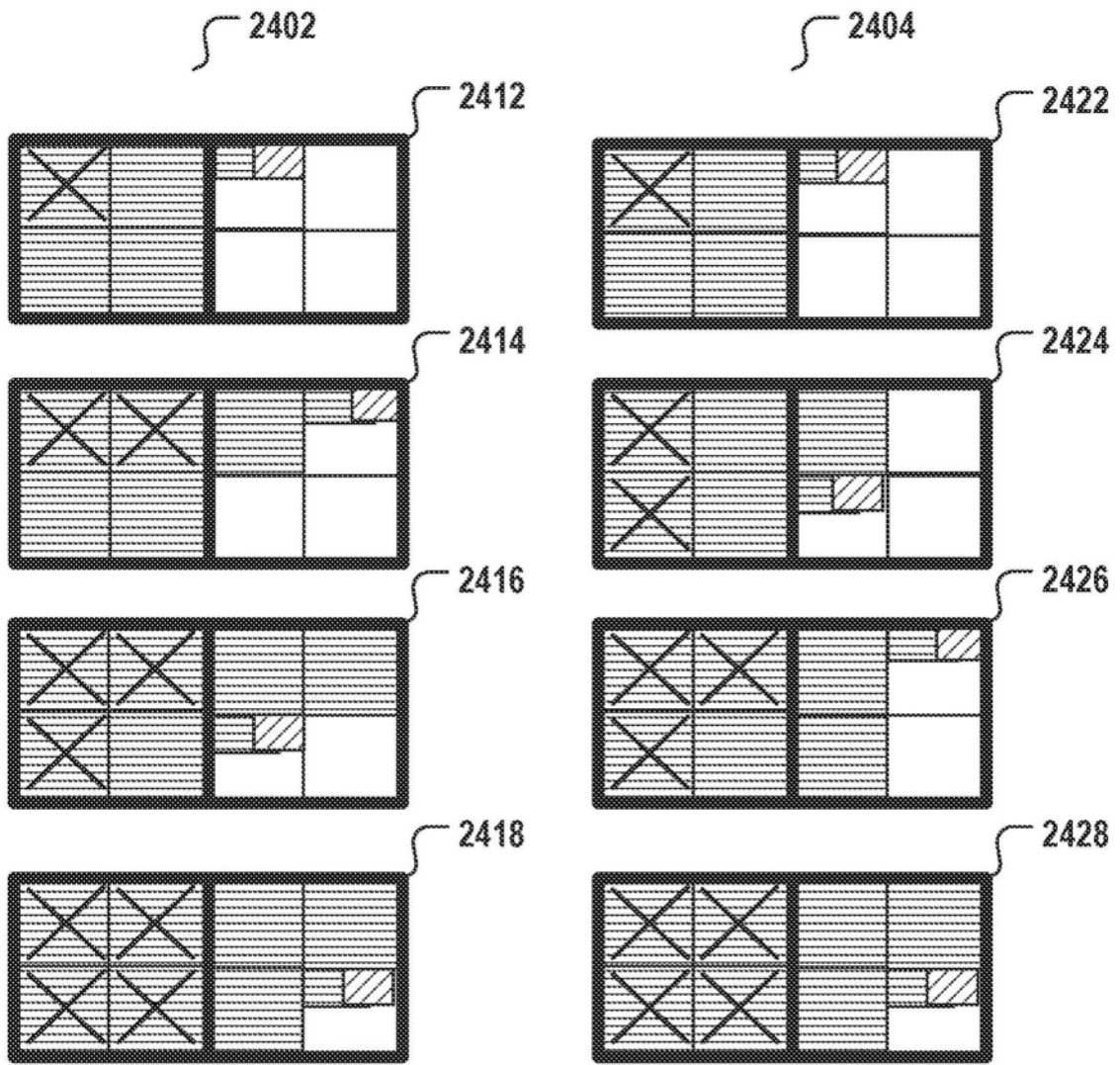


图23



水平拆分

垂直拆分

图24

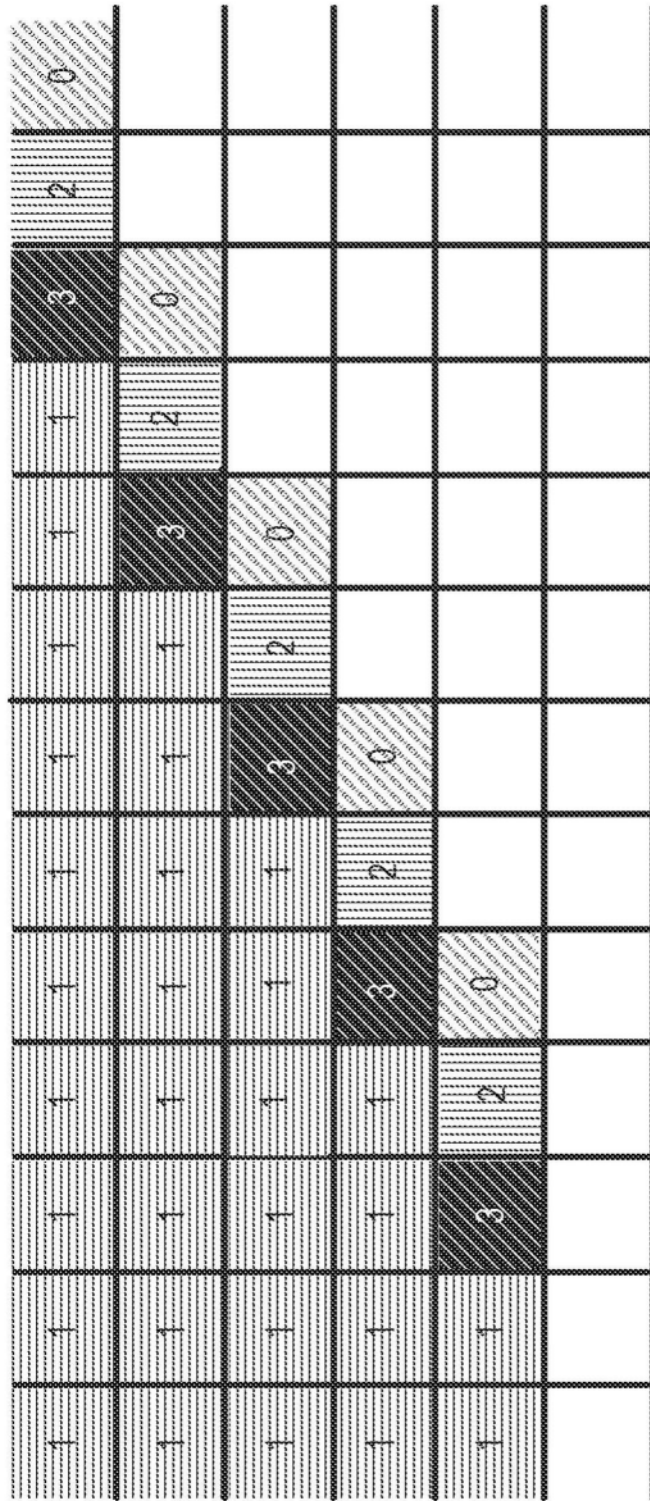


图25

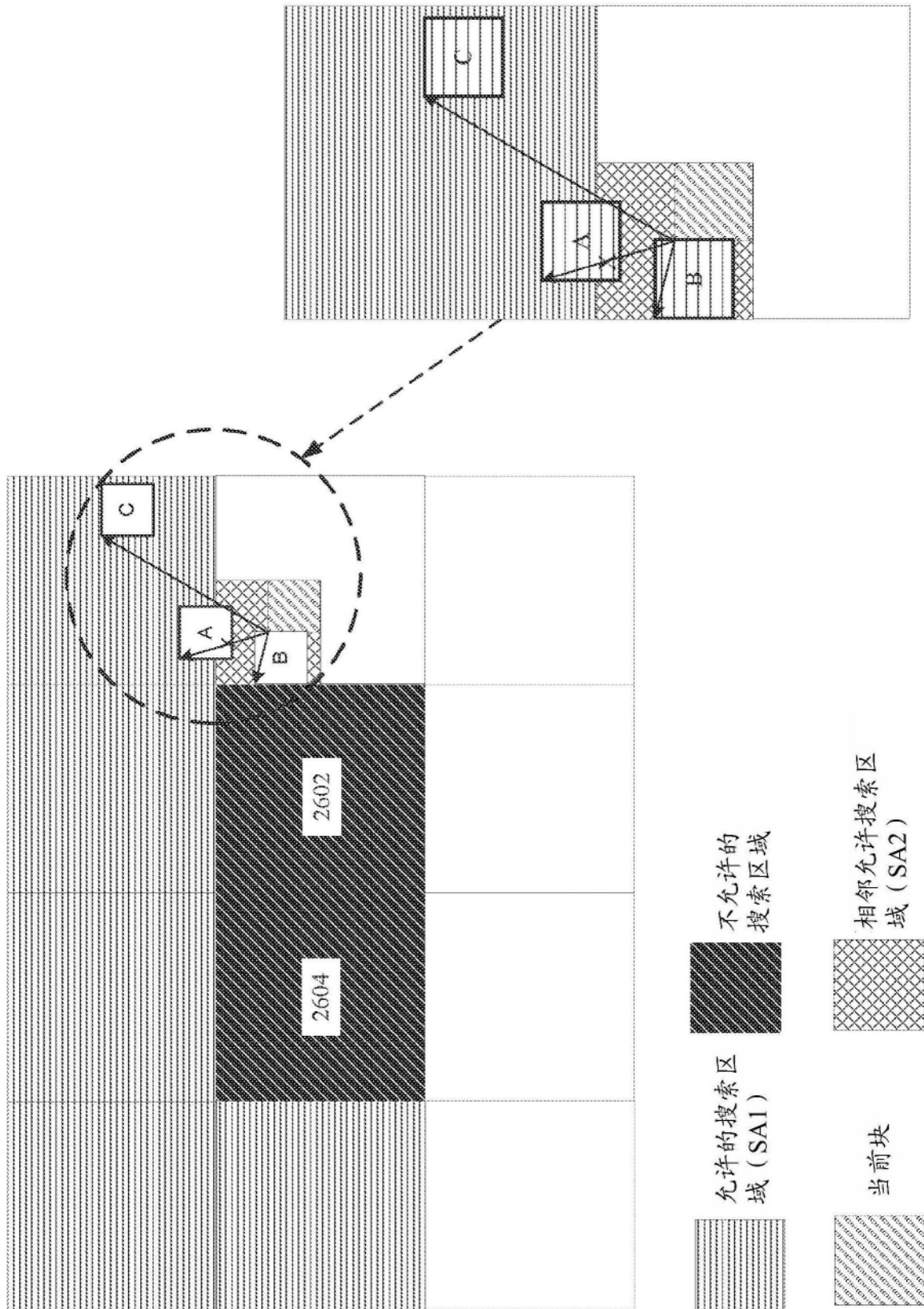


图26

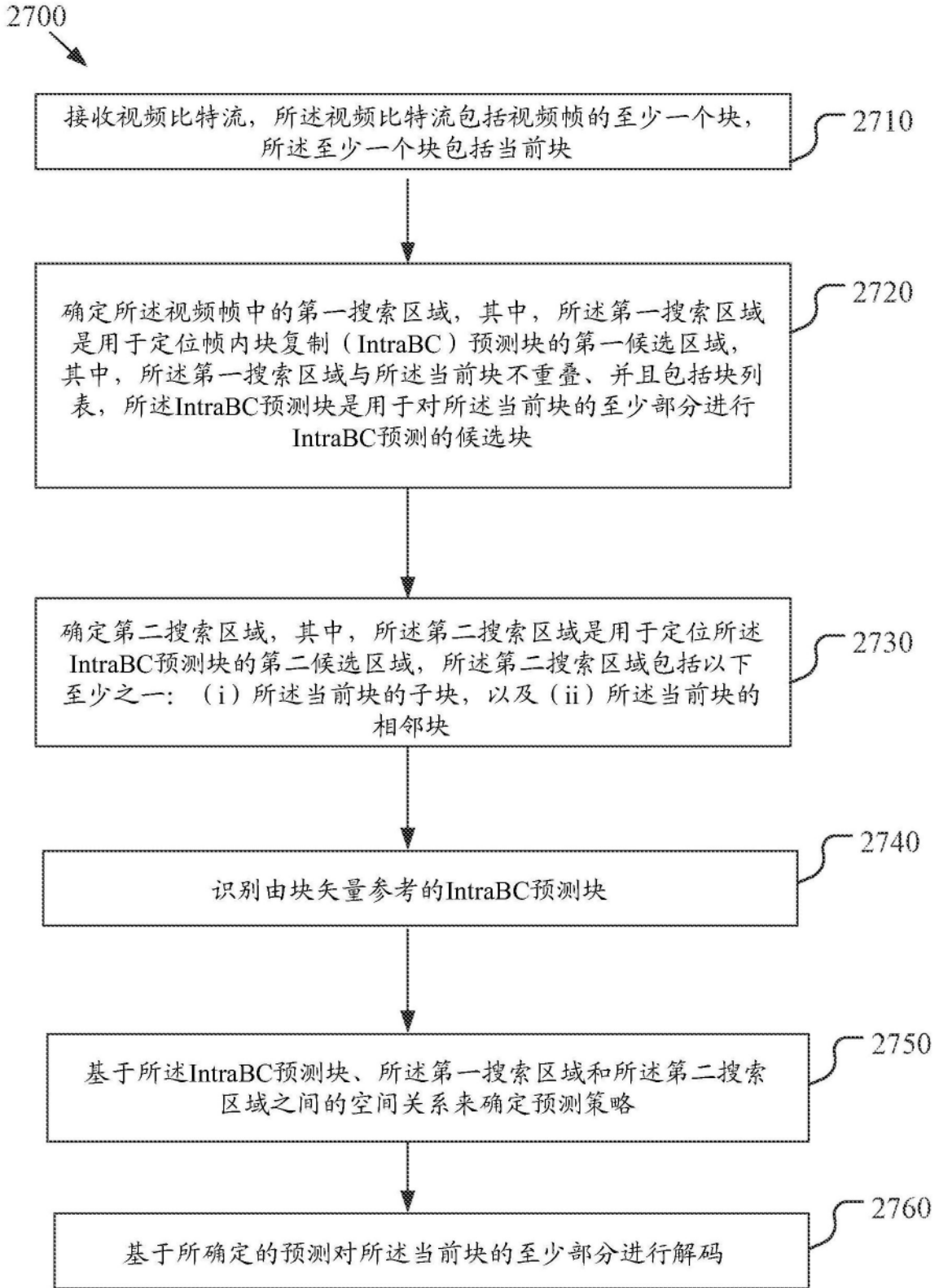


图27

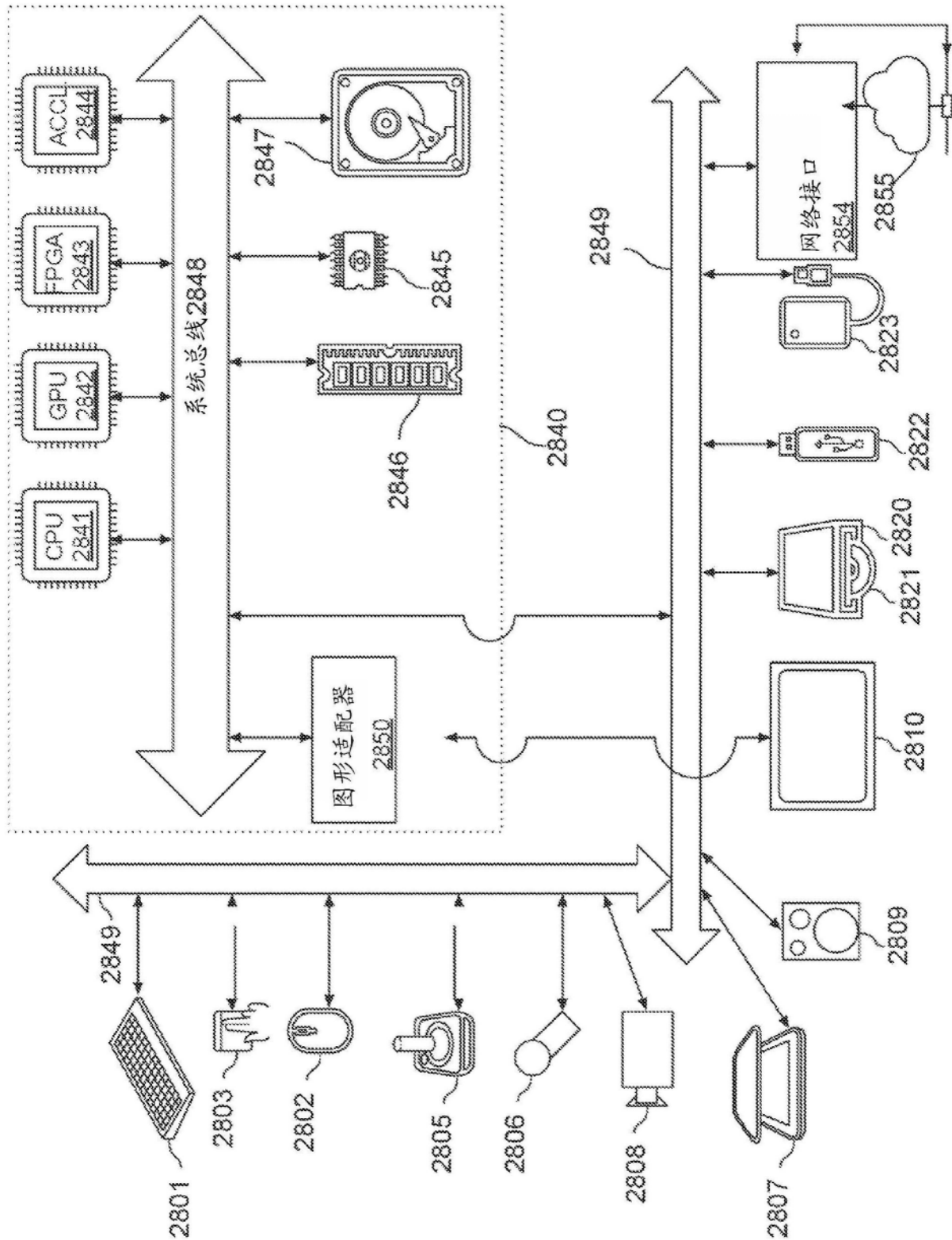


图28