



(12) 发明专利

(10) 授权公告号 CN 116830582 B

(45) 授权公告日 2025. 02. 14

(21) 申请号 202280014441.X

(22) 申请日 2022.06.01

(65) 同一申请的已公布的文献号
申请公布号 CN 116830582 A

(43) 申请公布日 2023.09.29

(30) 优先权数据
63/300,433 2022.01.18 US
17/824,168 2022.05.25 US

(85) PCT国际申请进入国家阶段日
2023.08.10

(86) PCT国际申请的申请数据
PCT/US2022/031737 2022.06.01

(87) PCT国际申请的公布数据
W02023/140883 EN 2023.07.27

(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 赵亮 赵欣 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
专利代理人 崔晓岚 张颖玲

(51) Int.Cl.
H04N 19/513 (2006.01)
H04N 19/105 (2006.01)
H04N 19/61 (2006.01)
H04N 19/139 (2006.01)

(56) 对比文件
CN 116686289 A, 2023.09.01

审查员 李若云

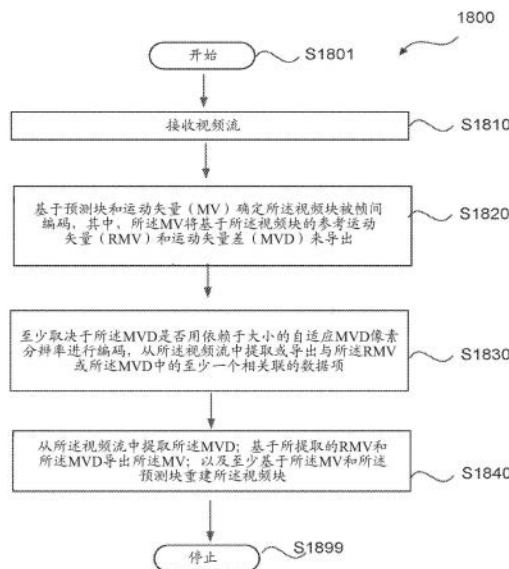
权利要求书2页 说明书38页 附图20页

(54) 发明名称

对视频流的视频块进行解码、编码的方法、
视频解码设备及视频编码设备

(57) 摘要

本公开根据在视频编码和解码中是否采用依赖于相关的自适应运动矢量差像素分辨率来写入运动矢量或运动矢量差。公开了一种用于对视频流的视频块进行处理的示例方法。所述方法可以包括：接收所述视频流；基于预测块和运动矢量(MV)确定所述视频块被帧间编码，其中，所述MV将基于所述视频块的参考运动矢量(RMV)和运动矢量差(MVD)来导出；以至少取决于所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码的方式，从所述视频流中提取或导出与所述RMV或所述MVD中的至少一个相关联的数据项；从所述视频流中提取所述MVD；基于所提取的RMV和所述MVD导出所述MV；以及至少基于所述MV和所述预测块重构所述视频块。



1. 一种用于对视频流的视频块进行处理的方法,其特征在于,包括:
 - 接收所述视频流;
 - 基于预测块和运动矢量MV确定所述视频块被帧间编码,其中,所述MV将基于所述视频块的参考运动矢量RMV和运动矢量差MVD来导出;
 - 以取决于所述MVD是用依赖于大小的自适应MVD像素分辨率进行编码的方式,并且当所述视频块以单参考模式预测得到时,确定RMV索引范围N,N是1或2,所述单参考模式是使用一个参考块来生成所述视频块的预测块的模式;以及
 - 当N=1时,所述视频流中不存在所述RMV索引,且所述RMV索引是响应于确定N=1而导出的;以及
 - 基于所述RMV索引范围解析所述视频流,以提取所述视频块的RMV索引,RMV索引1至N映射到由动态参考列表DRL标识的多个有序的候选RMV中的前N个候选RMV;
 - 从所述视频流中提取所述MVD;
 - 基于所提取的所述RMV索引和所述MVD导出所述MV;以及
 - 基于所述MV和所述预测块重建所述视频块。
2. 根据权利要求1所述的方法,其特征在于,N以序列级、帧级、片级、标题级或超级块级的语法元素被写入所述视频流,所述方法还包括从所述视频流中提取N。
3. 根据权利要求1所述的方法,其特征在于,
 - 从所述视频流中提取所述RMV索引;以及
 - 将所述RMV索引写入所述视频流的上下文取决于所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码。
4. 根据权利要求3所述的方法,其特征在于,当所述MVD用依赖于大小的自适应MVD像素分辨率进行编码时,使用第一上下文将所述RMV写入所述视频流中,而当所述MVD没有用依赖于大小的自适应MVD像素分辨率进行编码时,使用与所述第一上下文不同的第二上下文将所述RMV写入所述视频流中。
5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:响应于所述视频块用依赖于大小的自适应MVD像素分辨率进行编码,并且当所述视频块以单参考模式预测时,从所述视频流中提取指示是否采用重叠块运动补偿OBMC或扭曲运动的信息项。
6. 根据权利要求1所述的方法,其特征在于,所述方法还包括:响应于所述视频块用依赖于大小的自适应MVD像素分辨率进行编码,并且当所述视频块以单参考模式预测时,用于所述视频块的混合帧间和帧内预测的复合帧间-帧内模式被使用。
7. 根据权利要求1所述的方法,其特征在于,用于写入与所述MVD相关的至少一个语法元素的上下文导出取决于所述视频块是否用依赖于大小的自适应MVD像素分辨率进行编码。
8. 根据权利要求7所述的方法,其特征在于,与所述MVD相关的所述至少一个语法元素包括以下至少一个:
 - 第一MVD语法元素,用于指示所述MVD的哪些分量是非零的;
 - 第二MVD语法元素,用于指定所述MVD的符号;
 - 第三MVD语法元素,用于指定所述MVD的大小范围;
 - 第四MVD语法元素,用于指定所述MVD的大小范围内的整数大小偏移;或

第五MVD语法元素,用于指定所述MVD的像素分辨率。

9.根据权利要求8所述的方法,其特征在于,当所述视频块用依赖于大小的自适应MVD像素分辨率进行编码时,导出第一上下文以解码与所述MVD相关的所述至少一个语法元素,而当所述视频块没有用依赖于大小的自适应MVD像素分辨率进行编码时,导出与所述第一上下文不同的第二上下文以解码与所述MVD相关的所述至少一个语法元素。

10.一种用于对视频流的视频块进行编码的方法,其特征在于,包括:

接收所述视频流;

基于预测块和运动矢量MV设置所述视频块被帧间编码,其中,所述MV将基于所述视频块的参考运动矢量RMV和运动矢量差MVD来导出;

以取决于所述MVD是用依赖于大小的自适应MVD像素分辨率进行编码的方式,并且当所述视频块以单参考模式预测得到时,设置RMV索引范围N,N是1或2,所述单参考模式是使用一个参考块来生成所述视频块的预测块的模式;以及

N=1时,所述视频流中不存在所述RMV索引,且所述RMV索引是响应于确定N=1而导出的;以及

基于所述RMV索引范围解析所述视频流,以提取所述视频块的RMV索引,RMV索引1至N映射到由动态参考列表DRL标识的多个有序的候选RMV中的前N个候选RMV;

从所述视频流中提取所述MVD;

基于所提取的所述RMV索引和所述MVD导出所述MV;以及

基于所述MV和所述预测块执行所述视频块的编码处理。

11.一种视频解码设备,其特征在于,包括处理器和用于存储计算机指令的存储器,所述处理器在执行所述计算机指令时,被配置为使得所述视频解码设备执行权利要求1至9中任一项所述的方法。

12.一种视频编码设备,其特征在于,包括处理器和用于存储计算机指令的存储器,所述处理器在执行所述计算机指令时,被配置为使得所述视频编码设备执行权利要求10所述的方法。

13.一种处理视频流的方法,其特征在于,所述视频流基于权利要求1至9中任一项所述的方法进行解码,或者基于权利要求10所述的方法产生。

对视频流的视频块进行解码、编码的方法、视频解码设备及视频编码设备

[0001] 援引并入

[0002] 本申请基于并要求于2022年5月25日提交的美国非临时申请号为No.17/824,168, 名称为“Interdependence Between Adaptive Resolution of Motion Vector Difference and Signaling/Derivation of Motion Vector-Related Parameters”的优先权权益,该申请基于并要求于2022年1月18日提交的美国临时专利申请号为No.63/300,433,名称为“Improvement for Adaptive MVD resolution”的优先权权益。在先申请的全部内容以引用的方式并入本文中。

技术领域

[0003] 本公开总体上涉及视频编码及解码,尤其涉及用于在视频块的帧间预测中提供运动矢量差的自适应分辨率的方法和系统。

背景技术

[0004] 本文所提供的背景技术说明以对本公开的内容作一般性说明为目的。发明人的某些工作(即已在此背景技术部分中作出描述的工作)以及说明书中关于某些尚未成为申请日之前的现有技术的内容,无论是以明确或隐含的方式均不被视为相对于本公开的现有技术。

[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 1920×1080 的亮度样本及相关的全采样或子采样的色度样本的空间大小。该一系列图片可具有例如每秒60幅图片或每秒60帧的固定或可变的图片速率(替代地,称为帧率)。未压缩的视频对流式传输或数据处理具有特定比特率要求。例如,具有 1920×1080 的像素分辨率、60帧/秒的帧率、以及每个颜色通道每个像素8位的4:2:0的色度子采样的视频需要接近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。压缩可有助于减小上述带宽和/或存储空间需求,在某些情况下可减小两个数量级或大于两个数量级。可以采用无损压缩和有损压缩,以及它们的组合。无损压缩指的是可通过解码过程从已压缩的原始信号中重建原始信号的精确副本的技术。有损压缩指的是如下编码/解码过程,在该编码/解码过程中,在编码期间未完全保留原始视频信息,以及在解码期间未完全恢复原始视频信息。当使用有损压缩时,已重建的信号可能与原始信号不同,尽管丢失一些信息,但是原始信号和已重建的信号之间的失真足够小,以使已重建的信号可用于预期的应用。在视频的情况下,在许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,某些消费视频流式应用的用户相比电影或电视广播应用的用户来说可以容忍更高的失真。可以选择或调节可通过特定的编码算法达到的压缩率,以反映各种失真容忍度:更高的可容忍的失真通常允许编码算法产生更高的损失和更高的压缩率。

[0007] 视频编码器和解码器可利用来自多个类别和步骤的技术,这些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。当所有样本块都以帧内模式编码时,该图片可称为帧内图片。帧内图片及其派生物(例如,独立解码器刷新图片)可用于重置解码器状态,因此可用作已编码视频码流和视频会话中的第一张图片,或者用作静止图像。然后,帧内预测之后的块的样本可进行变换而处于频域中,且可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] 例如从诸如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,一些更新的视频压缩技术包括基于例如在空间相邻的编码和/或解码期间获得的、按照解码顺序在帧内编码或解码的数据块之前的周围样本数据和/或元数据来尝试对块进行编码/解码的技术。此类技术此后称为“帧内预测”技术。应注意,至少在一些情况下,帧内预测仅使用来自正在重建的当前图片的参考数据,而不使用来自其它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可使用不止一种这样的技术时,使用中的技术可称为帧内预测模式。可以在特定的编解码器中提供一种或多种帧内预测模式。在一些情况下,模式可具有子模式,和/或可与各种参数相关联,且用于视频块的模式/子模式信息和帧内编码参数可单独编码或共同包含在模式码字中。对于给定模式、子模式和/或参数组合使用哪个码字可能会对通过帧内预测的编码效率增益产生影响,且用于将码字转换为码流的熵编码技术同样对其也可以产生影响。

[0011] H.264引入了某种帧内预测模式,并在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,可使用已经变成可用的相邻样本值来形成预测块。例如,相邻样本的特定集的可用值可沿着一些方向和/或线复制到预测块中。对使用方向的参考可以在码流中进行编码,或者可对其本身进行预测。对所使用方向的参考可以编码在码流中,或者可以对其本身进行预测。

[0012] 参考图1A,在右下方描绘了H.265的33种可能的帧内预测方向(对应于H.265中指定的35个帧内模式的33个角模式)中指定的9个预测方向的子集。箭头收敛的点(101)表示正在被预测的样本。箭头表示相邻样本用于预测101处的样本所沿的方向。例如,箭头(102)指示从在右上方、与水平方向成45度角的一个或多个相邻样本预测样本(101)。类似地,箭头(103)指示从在样本(101)的左下方、与水平方向成22.5度角的一个或多个相邻样本预测样本(101)。

[0013] 仍然参考图1,在左上方描绘了 4×4 个样本的正方形块(104)(用粗体虚线表示)。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 4×4 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中,使用与正在重建的块相邻的预测样本。

[0014] 块104的帧内图片预测可通过根据用信号通知的预测方向从相邻样本复制参考样本值来开始。例如,假设已编码视频码流包括信令(signaling),该信令针对该块104指示箭头(102)的预测方向,即从在右上方、与水平方向成45度角的一个或多个预测本来预测样本。在这种情况下,从同一个参考样本R05预测样本S41、S32、S23和S14。然后,根据参考样本R08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。

[0016] 随着视频编码技术继续发展,可能的方向的数量增加。例如,在H.264(2003年)中,九个不同的方向可用于帧内预测。在H.265(2013年)中,增加到33个方向,而在本公开时,JEM/VVC/BMS可支持多达65个方向。已进行实验研究来帮助识别最合适的帧内预测方向,且熵编码中的一些技术可用于以少量比特对那些最合适的方向进行编码,对于方向,接受一定的比特代价。此外,有时可以从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示出了示意图(180),其描绘了根据JEM的65个帧内预测方向,以说明随着时间的推移,开发出来的各种编码技术中的预测方向的数量增加。

[0018] 已编码视频码流中表示帧内预测方向的比特映射到预测方向的方式可能因视频编码技术而不同;例如,其范围可以从预测方向简单直接映射到帧内预测模式,映射到码字,映射到涉及最可能模式的复杂自适应方案,以及类似技术。然而,在所有情况下,对于帧内预测,可存在某些方向,这些方向与某些其它方向相比,在统计上在视频内容中出现的可能性较小。由于视频压缩的目标是减少冗余,因此在一种设计良好的视频编码技术中,那些不太可能的方向相比可能出现的方向将由更多的位数表示。

[0019] 帧间图片预测或帧间预测可基于运动补偿。在运动补偿中,来自先前重建的图片或其部分(参考图片)的样本数据在沿着由运动矢量(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)。在某些情况下,参考图片可以与当前正在重建的图片相同。MV可具有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,可根据其它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其它区域相关的、且按解码顺序在当前MV之前的其它MV来预测适用于样本数据的某个区域的当前MV。这样做可通过消除相关MV中的冗余,而大大减少对MV进行编码所需的数据的总量,从而增加压缩效率。MV预测可有效地工作,例如,由于在对从相机获得的输入视频信号(称为自然视频)进行编码时,存在以下统计可能性:在视频序列中,比适用单个MV的区域更大的区域沿着相似的方向移动,因此,在某些情况下,可使用从相邻区域的MV导出的相似运动矢量来预测该更大的区域。这使得用于给定区域的实际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,该MV可以用比直接对MV进行编码(而非根据相邻MV来预测MV)时使用的位数更少的位数来表示。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,本文描述的是下文称为“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。另外,可从一个或多个参考图片相关联的元数据中导出该MV,而非对MV直接编码。例如,使用关联于A0、A1和B0、B1、B2(分别对应202到206)五个周围样本中的任一样本的MV,(按解码次序)从最近的参考图片的元数据中导出该MV。在H.265中,MV预测可使用来自相邻块正在使用的相同参考图片的预测值。

发明内容

[0023] 本公开总体上涉及视频编码,特别涉及用于基于在帧间预测中运动向量差是否采用依赖于大小的自适应分辨率来写入各种运动矢量或运动矢量差相关的语法的方法和系统。

[0024] 在示例实现中,公开了一种用于处理视频流的视频块的方法。所述方法可以包括:接收所述视频流;基于预测块和运动矢量(MV)确定所述视频块被帧间编码,其中,所述MV将基于所述视频块的参考运动矢量(reference motion vector, RMV)和运动矢量差(motion vector difference, MVD)来导出;以至少取决于所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码的方式,从所述视频流中提取或导出与所述RMV或所述MVD中的至少一个相关联的数据项;从所述视频流中提取所述MVD;基于所提取的RMV和所述MVD导出所述MV;以及至少基于所述MV和所述预测块重建所述视频块。

[0025] 在以上实现中,所述数据项可以包括与所述RMV或所述MVD中的至少一个相关联的语法元素。

[0026] 在以上任何一个实现中,语法元素可以包括RMV。

[0027] 在以上的任何一个实现中,所述数据项可以包括映射到动态参考列表(Dynamic Reference List, DRL)中的所述视频块的RMV索引,所述DRL被用于识别多个有序的候选RMV。

[0028] 在以上任何一个实现中,其中,至少取决于所述视频块的MVD是否用依赖于大小的自适应MVD像素分辨率进行编码,提取所述数据项可以包括:至少取决于所述视频块的MVD是否用依赖于大小的自适应MVD像素分辨率进行编码,确定RMV索引范围N,N是正整数;以及基于所述RMV索引范围解析所述视频流以提取所述视频块的RMV索引。

[0029] 在以上任何一个实现中, RMV索引1至N可以映射到所述DRL中的预定位置集合。

[0030] 在以上任何一个实现中,所述RMV索引1至N可以映射到由所述DRL标识的多个有序的候选RMV中的前N个候选RMV。

[0031] 在以上任何一个实现中,N可以是1或2。

[0032] 在以上任何一个实现中,N可以被写入所述视频流,所述方法还包括从所述视频流中提取N。

[0033] 在以上任何一个实现中,N可以以序列级、帧级、片级、标题级或超级块级的语法元素被写入。

[0034] 在以上任何一个实现中, $N=1$; 以及所述视频流中可以不存在所述RMV索引, 且所述RMV索引是响应于确定 $N=1$ 而导出的。

[0035] 在以上任何一个实现中, 所述提取或导出所述RMV索引的方式除了取决于所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码之外, 还可以取决于所述视频块是否以单参考模式预测。

[0036] 在以上任何一个实现中, 可以从所述视频流中提取所述RMV索引; 以及将所述RMV索引写入所述视频流的上下文可以取决于所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码。

[0037] 在以上任何一个实现中, 当所述MVD用依赖于大小的自适应MVD像素分辨率进行编码时, 可以使用第一上下文将所述RMV写入所述视频流中, 而当所述MVD没有用依赖于大小的自适应MVD像素分辨率进行编码时, 可以使用与所述第一上下文不同的第二上下文将所述RMV写入所述视频流中。

[0038] 在以上任何一个实现中, 该方法还可以包括: 响应于所述视频块用依赖于大小的自适应MVD像素分辨率进行编码, 并且当所述视频块以单参考模式预测时, 从所述视频流中提取指示是否采用重叠块运动补偿 (Overlapped Block Motion Compensation, OBMC) 或扭曲运动的信息项。

[0039] 在以上任何一个实现中, 该方法还可以包括: 响应于所述视频块用依赖于大小的自适应MVD像素分辨率进行编码, 并且当所述视频块以单参考模式预测时, 从所述视频流中提取指示是否采用复合帧间-帧内预测模式的信息项。

[0040] 在以上任何一个实现中, 用于写入与所述MVD相关的至少一个语法元素的上下文导出可以取决于所述视频块是否用依赖于大小的自适应MVD像素分辨率进行编码。

[0041] 在以上任何一个实现中, 与所述MVD相关的所述至少一个语法元素包括以下至少一个: 第一MVD语法元素, 用于指示所述MVD的哪些分量是非零的; 第二MVD语法元素, 用于指定所述MVD的符号; 第三MVD语法元素, 用于指定所述MVD的大小范围; 第四MVD语法元素, 用于指定所述MVD的大小范围内的整数大小偏移; 或第五MVD语法元素, 用于指定所述MVD的像素分辨率。

[0042] 在以上任何一个实现中, 当所述视频块用依赖于大小的自适应MVD像素分辨率进行编码时, 可以导出第一上下文以解码与所述MVD相关的所述至少一个语法元素, 而当所述视频块没有用依赖于大小的自适应MVD像素分辨率进行编码时, 可以导出与所述第一上下文不同的第二上下文以解码与所述MVD相关的所述至少一个语法元素。

[0043] 在另一实现中, 公开了一种用于对视频流的视频块进行解码的方法。所述方法包括: 接收所述视频流; 基于预测块和运动矢量 (MV) 确定所述视频块被帧间编码, 其中, 所述MV将基于所述视频块的参考运动矢量 (RMV) 和运动矢量差 (MVD) 来导出; 对映射到动态参考列表 (DRL) 中的所述视频块的RMV索引进行提取, 所述DRL用于识别多个有序的候选RMV; 以及基于所述RMV索引的值确定所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码。

[0044] 在以上实现中, 该方法还包括: 当所述RMV索引的值指示由所述DRL标识的所述多个有序的候选RMV中的前 N 个RMV候选之一时, 从所述视频流中提取标志, N 是正整数; 基于所述标志确定所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码; 以及当所述RMV

索引的值指示所述多个有序的候选RMV中没有所述前N个RMV候选时,确定所述MVD不是用依赖于大小的自适应MVD像素分辨率进行编码的。

[0045] 在以上任何一个实现中,N可以被预定义为1或2。在以上任何一个实现中,N被单独地写入所述视频流中。在以上任何一个实现中,N可以以序列级、帧级、片级、标题级或超级块级的语法元素被写入。

[0046] 本公开的各个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一方法实现的电路。

[0047] 本公开的各个方面还提供了一种非暂时性计算机可读介质,用于存储指令,当该指令被计算机执行用于视频解码和/或编码时,使得计算机执行上述用于视频解码和/或编码的方法。

附图说明

[0048] 通过下面的详细描述和附图,所公开的主题的进一步特征、性质和各种优点将更加明显,在附图中:

[0049] 图1A示出了帧内预测方向模式的示例性子集的示意图。

[0050] 图1B示出了示例性的帧内预测方向的图示。

[0051] 图2示出了一个示例中的用于运动矢量预测的当前块及其周围空间合并的候选块的示意图。

[0052] 图3示出了根据一个示例实施例的通信系统(300)的简化框图的示意图。

[0053] 图4示出了根据一个示例实施例的通信系统(400)的简化框图的示意图。

[0054] 图5示出了根据一个示例实施例的视频解码器的简化框图的示意图。

[0055] 图6示出了根据一个示例实施例的视频编码器的简化框图的示意图。

[0056] 图7示出了根据另一示例实施例的视频编码器的框图。

[0057] 图8示出了根据另一示例实施例的视频解码器的框图。

[0058] 图9示出了根据本公开的示例实施例的编码块划分的方案。

[0059] 图10示出了根据本公开的示例实施例的编码块划分的另一方案。

[0060] 图11示出了根据本公开的示例实施例的编码块划分的另一方案。

[0061] 图12示出了根据一个示例划分方案将基本块划分成编码块的示例。

[0062] 图13示出了示例性三元划分方案。

[0063] 图14示出了示例性四叉树二叉树编码块划分方案。

[0064] 图15示出了根据本公开的示例实施例的用于将编码块划分成多个变换块的方案以及这些变换块的编码顺序。

[0065] 图16示出了根据本公开的示例实施例的用于将编码块划分成多个变换块的另一方案以及这些变换块的编码顺序。

[0066] 图17示出了根据本公开的示例实施例的用于将编码块划分成多个变换块的另一方案。

[0067] 图18示出了根据本公开的示例实施例的方法的流程图。

[0068] 图19示出了根据本公开的示例实施例的方法的另一流程图。

[0069] 图20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

[0070] 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。同样地,本文中使用的短语“在一个实施方式中”或“在一些实施方式中”不一定指相同的实施方式,本文中使用的短语“在另一个实施方式中”或“在其他实施方式中”不一定指不同的实施方式。例如,所要求保护的主体包括全部或部分示例性实施例/实施方式的组合。

[0071] 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、结构或特性,或可用于以复数意义描述特征、结构或特性的组合。类似地,诸如“一个”、“一个”或“该”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,属于“基于”或“由...确定”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。图3示出了根据本公开的一个实施例的通信系统(300)的简化框图。通信系统(300)包括多个终端设备,该终端设备可通过例如网络(350)彼此通信。举例来说,通信系统(300)包括通过网络(350)互连的第一终端设备对(310)和(320)。在图3的示例中,第一终端设备对(310)和(320)可执行单向数据传输。例如,终端设备(310)可以对视频数据(例如,由终端设备(310)采集的视频图片流的数据)进行编码,以通过网络(350)传输到另一终端设备(320)。已编码的视频数据以一个或多个已编码视频码流形式传输。终端设备(320)可以从网络(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,以及根据恢复的视频数据显示视频图片。单向数据传输可以在媒体服务等应用中实现。

[0072] 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设备对(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,在一个示例中,终端设备(330)和(340)中的每个终端设备可以对视频数据(例如,由终端设备采集的视频图片流的数据)进行编码,以通过网络(350)传输到终端设备(330)和(340)中的另一终端设备。终端设备(330)和(340)中的每个终端设备还可接收由终端设备(330)和(340)中的另一终端设备传输的已编码视频数据,且可以对已编码视频数据进行解码以恢复视频图片,且可根据所恢复的视频数据在可访问的显示设备上显示视频图片。

[0073] 在图3的示例中,终端设备(310)、终端设备(320)、终端设备(330)和终端设备(340)可实现为服务器、个人计算机和智能电话,但是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台式计算机、膝上型计算机、平板电脑、媒体播放器、可穿戴式计算机、专用视频会议设备、和/或类似物上实现。网络(350)表示在终端设备(310)、终端设备(320)、终端设备(330)和终端设备(340)之间传送已编码视频数据的任何数量或类型的网络,包括例如有线(连线的)和/或无线通信网络。通信网络(350)可以在电路交换信道、分组交换信道、和/或其它类型的信道中交换数据。代表性的网络包括电信网络、局域网、广域网

和/或互联网。出于本讨论的目的,除非在本文中明确说明,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。

[0074] 作为用于所公开的主题的应用的示例,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。所公开的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TV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。

[0075] 视频流式传输系统可包括视频采集子系统(413),视频采集子系统(413)可包括例如数码相机的视频源(401),视频源(401)创建未压缩的视频图片流或图像(402)。在一个示例中,视频图片流(402)包括由视频源401的数码相机记录的样本。相较于已编码的视频数据(404)(或已编码的视频码流),视频图片流(402)被描绘为粗线以强调高数据量的视频图片流,视频图片流(302)可由电子设备(420)处理,该电子设备(320)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)(或已编码视频码流(404))可存储在流式传输服务器(405)上以供将来使用,或者直接存储于下游视频设备(未示出)。一个或多个流式客户端子系统,例如图4中的客户端子系统(406)和客户端子系统(408)可以访问流式服务器(405)以检索编码视频数据(404)的副本(407)和副本(409)。客户端子系统(406)可包括例如电子设备(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,且产生可以在显示器(412)(例如,显示屏)或其它呈现设备(未描绘)上呈现的未压缩的输出视频图片流(411)。视频解码器410可配置成执行本公开所描述的各种功能中的一些或全部功能。在一些流式传输系统中,可根据某些视频编码/压缩标准对已编码的视频数据(404)、视频数据(407)和视频数据(409)(例如视频码流)进行编码。该些标准的示例包括ITU-T H.265。在实施例中,正在开发的视频编码标准非正式地称为下一代视频编码(Versatile Video Coding,VVC)。所公开的主题可用于VVC的上下文中,且可用于其它视频编码标准。

[0076] 应注意,电子设备(420)和电子设备(430)可包括其它组件(未示出)。举例来说,电子设备(420)可包括视频解码器(未示出),且电子设备(430)还可包括视频编码器(未示出)。

[0077] 在下文中,图5示出了根据本公开的任意实施例的视频解码器(510)的框图。视频解码器(510)可设置在电子设备(530)中。电子设备(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(410)。

[0078] 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,一次可以对一个已编码视频序列进行解码,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。每个视频序列可与多个视频帧或图像相关联。可以从信道(501)接收已编码视频序列,信道(501)可以是硬件/软件链路,其通向存储已编码视频数据的存储设备或者发送已编码视频数据的流式传输源。接收器(531)可接收可转发到它们各自的处理电路(未描绘)的已编码视频数据和其它数据,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)和熵解码器/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)可实现为视频解码器(510)的一部分。在其它应用中,缓冲存储器(515)可位于视频解码器(510)的外部且与视频解码器(510)分开(未描绘)。而在另一些其它应用中,视频解码器(510)的外部设置缓冲存储器(未描绘)以例如防止网络抖动,且在视频解码器(510)的内部可设置另一附加的缓冲存储器(515)以例如处理播出定时。而当接收器(531)从具有足够带宽和可控性的存储/转发设备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。为了在诸如互联网等业务分组网络上使用,可能需要足够大小的缓冲存储器(515),缓冲存储器(515)的大小可相对较大。这种缓冲存储器可实现为具有自适应大小,且可至少部分地在操作系统或视频解码器(510)外部的类似元件(未描绘)中实现。

[0079] 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,以及用于控制诸如显示器(512)(例如,显示屏)的呈现设备的潜在信息,该呈现设备可以是或者不是电子设备(530)的整体部分,但是可耦接到电子设备(530),如图5所示。用于呈现设备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)参数集片段(未描绘)的形式。解析器(520)可以对由解析器(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(Huffman)编码、具有或不具有上下文敏感度的算术编码等。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。解析器(520)还可以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。

[0080] 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)。

[0081] 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的处理或功能单元。涉及哪些单元以及涉及方式可由解析器(520)通过从已编码视频序列解析的子群控制信息来控制。为了简单起见,未描绘解析器(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。

[0082] 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现方式中,这些功能单元中的许多功能单元彼此紧密交互且可至少部分地彼此集成。然而,出于清楚地描述所公开的主题的各种功能的目的,在本公开的下文中采用在概念上细分成多个功能单元。

[0083] 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)可以从解析器(520)接收作为符号(521)的量化变换系数以及控制信息,包括指示要使用哪种类型的逆变换、块大小、量化因子/参数、量化缩放矩阵等的信息。缩放器/逆变换单元(551)可输出包括样本值的块,样本值可输入到聚合器(555)中。

[0084] 在一些情况下,缩放器/逆变换(551)的输出样本可属于帧内编码块;即:不使用来

自先前重建的图像的预测信息,但是可使用来自当前图片的先前重建部分的预测信息的块。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)可使用已重建且存储在当前图片缓冲器(558)中的周围块信息来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。在一些实现方式中,聚合器(555)可基于每个样本,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。

[0085] 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和潜在运动补偿块。在这种情况下,运动补偿预测单元(553)可访问参考图片存储器(557)以提取用于帧间图片预测的样本。在根据属于块的符号(521)对所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的输出可称为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)从参考图片存储器(557)内的地址提取预测样本可受到运动矢量控制,且该运动矢量可以以符号(521)的形式提供给运动补偿预测单元(553)使用,符号(521)可具有例如X分量、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,且还可与运动矢量预测机制等相关联。

[0086] 聚合器(555)的输出样本可在环路滤波器单元(556)中被各种环路滤波技术采用。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(也称作已编码视频码流)中的参数,且该参数作为来自解析器(520)的符号(521)可用于环路滤波器单元(556)。然而,在其他实施例中,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,以及响应于先前重建且经过环路滤波的样本值。可以以各种顺序包括多种类型的环路滤波器,作为环路滤波器单元556的一部分,如将在下文进一步详细描述。

[0087] 环路滤波器单元(556)的输出可以是样本流,该样本流可输出到呈现设备(512)以及存储在参考图片存储器(557)中以用于未来的帧间图片预测。

[0088] 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的帧间图片预测。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。

[0089] 视频解码器(510)可根据诸如ITU-T H.265建议书的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上,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。具体而言,配置文件可以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。为了符合标准,已编码视频序列的复杂度可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)规范和已编码视频序列中用信号表示的HRD缓冲器管理的元数据来进一步限定。

[0090] 在一些示例性实施例中,接收器(531)可连同已编码视频一起接收附加(冗余)数

据。该附加数据可以是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。附加数据可呈例如时间、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。

[0091] 图6是根据本申请公开的一个示例性实施例的视频编码器(603)的框图。视频编码器(603)可包括在电子设备(620)中。电子设备(620)可进一步包括传输器(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(403)。

[0092] 视频编码器(603)可以从视频源(601)(并非图6的示例中的电子设备(620)的一部分)接收视频样本,视频源(601)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)可实现为电子设备(620)的一部分。

[0093] 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.....)和任何合适的采样结构(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设备。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。视频数据可作为多个单独的图片或图像来提供,当按顺序观看时,这些图片或图像被赋予运动。图片本身可构建为空间像素阵列,其中取决于所使用的采样结构、色彩空间等,每个像素可包括一个或多个样本。本领域的普通技术人员可容易地理解像素和样本之间的关系。下文侧重于描述样本。

[0094] 根据一些示例性实施例,视频编码器(603)可实时地或在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并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,且控制所述的其它功能单元。为了简单起见,图中未描绘耦接。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 λ 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)可用于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)。

[0095] 在一些示例性实施例中,视频编码器(603)可配置成在编码环路中进行操作。作为过于简化的描述,在一个示例中,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来创建符号,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)重建符号以用类似于(远程)解码器创建样本数据的方式创建样本数据,即使嵌入式解码器633通过不具有熵编码的源编码器630来处理已编码视频流(因为在所公开的主题中考虑的视频压缩技术中,熵编码中的符号与已编码视频码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)用于改善编码质量。

[0096] “本地”解码器(633)的操作可与例如已在上文结合图5详细描述视频解码器(510)的“远程”解码器相同。然而,另外简要参考图5,由于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/解码成已编码视频序列,因此包括缓冲存储器(515)和解析器(520)的视频解码器(510)的熵解码部分可能无法完全在本地解码器(633)中、在编码器中实现。

[0097] 此时可以观察到,除了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,也必定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,所公开的主题有时侧重于解码器操作,解码器操作与编码器的解码部分协作。因此,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。下面只对编码器的某些领域或方面进行了更详细的描述。

[0098] 在操作期间,在一些示例性实现方式中,源编码器(630)可执行运动补偿预测编码,通过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。以这种方式,编码引擎(632)对输入图片的像素块和参考图片的像素块之间的颜色通道中的差异(或残差)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”可互换使用。

[0099] 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)的操作可为有损过程。当已编码的视频数据可以在视频解码器(图6中未示出)上解码时,重建的视频序列通常可能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高速缓存(634)中。以这种方式,视频编码器(603)可以在本地存储已重建参考图片的副本,该副本与将由远端(远程)视频解码器获得的已重建参考图片具有共同内容(不存在传输误差)。

[0100] 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,根据预测器(635)获得的搜索结果,可确定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。

[0101] 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。

[0102] 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)根据诸如霍夫曼编码、可变长度编码、算术编码等的技术来对各种功能单元生成的符号进行无损压缩,从而将该符号转换成已编码视频序列。

[0103] 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的视频数据的存储设备的硬件/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)。

[0104] 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:

[0105] 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。本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。

[0106] 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。

[0107] 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。

[0108] 源图片通常可以在空间上细分成多个样本编码块(例如,4×4、8×8、4×8或16×16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,其它(已编码)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编码,或该块可参考同一图片的已经编码的块来进行预测编码(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域预测进行预测编码。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域预测进行预测编码。出于其它目的,源图片或中间处理后的图片可细分成其它类型的块。编码块和其它类型的块的划分可遵循或者可以不遵循相同的方式,如下文进一步详细描述。

[0109] 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用视频编码技术或标准指定的语法。

[0110] 在一种示例性实施例中,传输器(640)可在传输已编码的视频时传输附加数据。源编码器(630)可以包括这样的数据作为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。

[0111] 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(通常缩写为帧内预测)利用给定图片中的空间相关性,而帧间预测利用图片之间的(时间或其他)相关性。例如,可以将正在编码/解码的特定图片分成块,正在编码/解码的特定图片被称为当前图片。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,可通过被称为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。

[0112] 在一些示例性实施例中,双向预测技术可用于帧间图片预测。根据这种双向预测技术,使用两个参考图片,例如按解码次序在视频中的当前图片之前(但是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共同预测该块。

[0113] 此外,合并模式技术可用于帧间图片预测以改善编码效率。

[0114] 根据本公开的一些示例性实施例,诸如帧间图片预测和帧内图片预测的预测以块为单位来执行。例如,将视频图片序列中的图片分成编码树单元(coding tree unit,CTU)以用于压缩,图片中的CTU可具有相同大小,例如64×64像素、32×32像素或16×16像素。通

常,CTU可包括三个并行的编码树块(coding tree block,CTB),这三个并行的编码树块是一个亮度CTB和两个色度CTB。可以将每个CTU递归地以二叉树划分成一个或多个编码单元(CU)。例如,可以将 64×64 像素的CTU划分成一个 64×64 像素的CU,或4个 32×32 像素的CU。一个或多个 32×32 块中的每个块可进一步划分成4个 16×16 像素的CU。在一些示例性实施例中,可以在编码期间分析每个CU以确定各种预测类型中、用于CU的预测类型,各种预测类型例如是帧间预测类型或帧内预测类型。根据时间和/或空间可预测性,可以将CU划分成一个或多个预测单元(PU)。通常,每个PU包括亮度预测块(prediction block,PB)和两个色度PB。在实施例中,编码(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划分成PU(或者不同颜色通道的PB)。例如,亮度或色度PB可包括针对样本的值(例如,亮度值)的矩阵,所述的样本例如是 8×8 像素、 16×16 像素、 8×16 像素、 16×8 样本等。

[0115] 图7示出了根据本公开的另一示例性实施例的视频编码器(703)的图。视频编码器(703)用于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。示例性视频编码器(703)可用于代替图4的示例中的视频编码器(403)。

[0116] 例如,视频编码器(703)接收用于处理块的样本值的矩阵,该处理块例如是 8×8 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对处理块进行编码。当确定在帧内模式中对处理块进行编码时,视频编码器(703)可使用帧内预测技术以将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对处理块进行编码时,视频编码器(703)可分别使用帧间预测或双向预测技术以将处理块编码到已编码图片中。在一些示例性实施例中,合并模式可用作帧间图片预测子模式,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。在一些其它示例性实施例中,可存在适用于主题块的运动矢量分量。因此,视频编码器(703)可包括未在图7中明确示出的组件,例如用于确定处理块的预测模式的模式决策模块。

[0117] 在图7的示例中,视频编码器(703)包括如图7的示例性布置所示的耦接在一起的帧间编码器(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)。

[0118] 帧间编码器(730)配置成接收当前块(例如,处理块)的样本、将该块与参考图片中的一个或多个参考块(例如,按显示次序的先前图片和后来图片中的块)进行比较、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来计算帧间预测结果(例如,已预测块)。在一些示例中,参考图片是基于已编码视频信息,使用解码单元633解码的已解码参考图片,解码单元633嵌入在图6的示例性编码器620中(其示出为图7的残差解码器728,如下文进一步详细描述)。

[0119] 帧内编码器(722)配置成接收当前块(例如,处理块)的样本、将该块与同一图片中已编码的块进行比较、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)。

[0120] 通用控制器(721)可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。在一个示例中,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给开关(726)。例如,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并将帧内预测信息包括在码流中;以及当块的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并将帧间预测信息包括在码流中。

[0121] 残差计算器(723)可配置成计算所接收的块与从帧内编码器(722)或帧间编码器(730)选择的块预测结果之间的差(残差数据)。残差编码器(724)可配置成对残差数据进行编码来生成变换系数。例如,残差编码器(724)可配置成将残差数据从空间域变换到频域,以生成变换系数。变换系数接着经由量化处理以获得量化的变换系数。在各种示例性实施例中,视频编码器(703)还包括残差解码器(728)。残差解码器(728)用于执行逆变换,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,且已解码图片可以在存储器电路(未示出)中缓冲并用作参考图片。

[0122] 熵编码器(725)可配置成将码流格式化以包括已编码块且执行熵编码。熵编码器(725)配置成将各种信息包括在码流中。例如,熵编码器(725)可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码流中。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,可以不存在残差信息。

[0123] 图8示出了根据本公开的另一实施例的示例性视频解码器(810)的图。视频解码器(810)用于接收作为已编码视频序列的一部分的已编码图像,且对该已编码图像进行解码以生成重建的图片。在一个示例中,视频解码器(810)可用于代替图4的示例中的视频解码器(410)。

[0124] 在图8的示例中,视频解码器(810)包括如图8的示例性布置所示的耦接在一起的熵解码器(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)。

[0125] 熵解码器(871)可用于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对块进行编码的模式(例如,帧内模式、帧间模式、双向预测模式、合并子模式或另一子模式)、可识别分别供帧内解码器(872)或帧间解码器(880)使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)、呈例如量化变换系数形式的残差信息等。在一个示例中,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给帧间解码器(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给帧内解码器(872)。残差信息可经由逆量化并提供到残差解码器(873)。

[0126] 帧间解码器(880)可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。

[0127] 帧内解码器(872)可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。

[0128] 残差解码器 (873) 可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) 还可使用某些控制信息 (用以包括量化器参数 (Quantizer Parameter, QP)), 该信息可由熵解码器 (871) 提供 (未描绘数据路径, 因为这仅仅是低数据量控制信息)。

[0129] 重建模块 (874) 可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 (可由帧间预测模块或帧内预测模块输出, 视情况而定) 以形成已重建块, 已重建块形成已重建图片的一部分, 已重建图片作为已重建视频的一部分。应注意, 还可执行诸如去块操作等其它合适的操作来改善视觉质量。

[0130] 应注意, 可使用任何合适的技术来实施视频编码器 (403)、视频编码器 (603) 和视频编码器 (703) 以及视频解码器 (410)、视频解码器 (510) 和视频解码器 (810)。在一些示例性实施例中, 可使用一个或多个集成电路来实现视频编码器 (403)、视频编码器 (603) 和视频编码器 (703) 以及视频解码器 (410)、视频解码器 (510) 和视频解码器 (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)、视频编码器 (603) 和视频编码器 (603) 以及视频解码器 (410)、视频解码器 (510) 和视频解码器 (810)。

[0131] 转到用于编码和解码的块划分, 一般划分可以从基本块开始, 并且可以遵循预定义的规则集、特定模式、划分树或任何划分结构或方案。划分可以是分层的和递归的。在按照下面描述的任何示例划分过程或其他过程或其组合将基本块分割或划分之后, 可以获得最终的一组分区或编码块。这些分区中的每一个都可以处于划分层次结构中的不同划分级别之一, 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划分实现, 每个生成的CB可以是任何允许的大小和划分级别。这样的分区被称为编码块, 因为它们可以形成单元, 可以对这些单元进行一些基本的编码/解码, 并且可以对编码/解码参数进行优化、确定和在已编码视频码流中写入。最终分区中的最高或最深级别表示树的编码块划分结构的深度。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 (coding block tree, CBT)。

[0132] 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)。一个CTU中不同颜色通道的划分模式或结构可能相同, 也可能不同。

[0133] 在一些实现中, 用于亮度和色度通道的划分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划分树结构以及要使用的实际编码划分树结构可以取决于被编码的切片是P切片、B切片还是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划分树结构或编码划分树结构模式, 而对于P切片或B切片, 亮度和色度通道可以共享相同的编码划分树方案。当应用单独的编码划分树结构或模式时, 亮度通道可以通过一个编码划分树结构被分成CBs, 色度通道可以通过另一个编码划分树结构被分成色度CBs。

[0134] 在一些示例性实现中, 可以将预定的划分模式应用于基本块。如图9所示, 示例性的4路划分树可以从第一预定义级别 (例如, 64×64 块级别或其他大小, 作为基本块大小) 开始, 并且基本块可以向下分层划分到预定义的最低级别 (例如, 4×4 级别)。例如, 基本块可以有由902、904、906和908指示的四个预定义的划分选项或模式, 其中指定为R的分区被允

许用于递归划分,使得如图9中指示的相同划分选项可以以较低的规模重复,直到最低级别(例如,4×4级别)。在一些实现中,可以对图9的划分方案应用附加限制。在图9的实现中,可以被允许矩形划分(例如,1:2/2:1的矩形划分),但是这些矩形划分不允许是递归的,而正方形划分允许是递归的。如果需要的话,按照图9的递归划分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分割深度。例如,根节点或根块(例如64×64个块)的编码树深度可以设置为0,并且在根块按照图9被进一步分割一次之后,编码树深度增加1。对于上述方案,从64×64的基本块到4×4的最小分区的最大或最深级别将是4(从级别0开始)。这种划分方案可以应用于一个或多个颜色通道。可以按照图9的方案独立地划分每个颜色通道(例如,可以针对每个分层级别的每个颜色通道独立地确定预定义模式中的划分模式或选项)。可选地,两个或更多个颜色通道可以共享图9的相同分层模式树(例如,可以在每个分层级别为两个或更多个颜色通道选择预定义模式中的相同划分模式或选项)。

[0135] 图10示出了允许递归划分形成划分树的另一示例性预定义划分模式。如图10所示,可以预定义一个示例性10路划分结构或模式。根块可以从预定义的级别开始(例如,从128×128级别或64×64级别的基本块开始)。图10的示例划分结构包括各种2:1/1:2和4:1/1:4的矩形划分。图10的第二行中指示的1002、1004、1006和1008的3个子划分的划分类型可以被称为“T型”分区。“T型”分区1002、1004、1006和1008可以分别被称为左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的矩形划分中的任何一个。可以进一步定义编码树深度以指示来自根节点或根块的分割深度。例如,根节点或根块(例如128×128块)的编码树深度可以设置为0,并且在根块按照图10被进一步分割一次之后,编码树深度增加1。在一些实现中,只有1010中的全正方形划分可以被允许按照图10的模式递归划分到划分树的下一级。换句话说,对于T型模式1002、1004、1006和1008内的正方形划分,可以不允许递归划分。如果需要的话,按照图10的递归划分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。在一些实现中,在使用8×8级别以下的分区时,可以增加更多的灵活性。例如,在某些情况下可以使用2×2色度帧间预测。

[0136] 在用于编码块划分的一些其它示例性实现中,四叉树结构可用于将基本块或中间块分割成四叉树分区。这种四叉树分割可以分层地和递归地应用于任何正方形分区。基本块或中间块或分区是否被进一步四叉树分割可适应于基本块或中间块或分区的各种局部特征。可以进一步调整在图片边界处的四叉树划分。例如,可以在图片边界处执行隐式四叉树分割,使得块将保持四叉树分割,直到大小适合图片边界。

[0137] 在一些其他示例实现中,可以使用来自基本块的分层二叉划分。对于这样的方案,基本块或中间级块可以被划分成两个分区。二叉划分可以是水平的,也可以是垂直的。例如,水平二叉划分可以将基本块或中间块分割成相等的右划分和左划分。类似地,垂直二叉划分可以将基本块或中间块分割为相等的上划分和下划分。这种二叉划分可以是分层的和递归的。可以在每个基本块或中间块上决定是否应该继续二叉划分方案,并决定如果该方案继续下去,应该使用水平还是垂直二叉划分。在一些实现中,进一步的划分可能会在预定义的最低分区大小(在一个或两个维度中)停止。可选地,一旦达到来自基本块的预定义划分级别或深度,则可以停止进一步的划分。在一些实现中,分区的纵横比可能会受到限制。例如,分区的纵横比可以不小于1:4(或大于4:1)。因此,纵横比为4:1的垂直条形分区,只能

进一步被垂直二叉划分为上下两个分区,每个分区的纵横比为2:1。

[0138] 在又一些其它示例中,如图13所示,三叉划分方案可用于划分基本块或任何中间块。三叉图案可以如图13的1302所示以垂直实现,或者如图13的1304所示以水平实现。虽然图13中的示例性分割比(垂直地或水平地)被示出为1:2:1,但也可以预定义其他比例。在一些实现中,可以预定义两个或多个不同的比率。这种三叉划分方案可以用于补充四叉树或二叉划分结构,因为这种三叉树划分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划分,因此该对象被划分成单独的分区。在一些实现中,示例三叉树划分的宽度和高度始终是2的幂,以避免额外的变换。

[0139] 上述划分方案可以在不同的划分级别上以任何方式组合。作为一个示例,可以组合上述四叉树和二叉划分方案以将基本块划分为四叉树-二叉树(quadtrees-binary-trees, QTBT)结构。在这种方案中,基本块或中间块/划分可以是四叉树分割或二叉分割,如果指定的话,这取决于一组预定义的条件。图14示出了一个特定的例子。在图14的示例中,一个基本块首先被四叉树分成四个分区,如1402、1404、1406和1408所示。此后,每个生成的分区在下一级要么被四叉树分割成四个进一步的分区(例如1408),要么被二叉分割成两个进一步的分区(例如水平地或垂直地,例如1402或1406,两者都是对称的),要么不被分割(例如1404)。对于正方形分区,可以递归地允许二叉或四叉树分割,如1410的整体划分模式示例和1420中的相应树结构/表示所示,其中实线表示四叉树分割,虚线表示二叉树分割。可以将标志用于每个二叉分割节点(非叶二叉分区),以指示二叉分割是水平的还是垂直的。例如,如1420所示,与1410的划分结构一致,标志“0”可以表示水平二叉分割,标志“1”可以表示垂直二叉分割。对于四叉树分割划分,不需要指示分割类型,因为四叉树分割总是水平和垂直地分割块或分区,以产生大小相等的4个子块/分区。在一些实现中,标志“1”可以表示水平二叉分割,标志“0”可以表示垂直二叉分割。

[0140] 在QTBT的一些示例实现中,四叉树和二叉分割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:

[0141] -CTU size:四叉树的根节点大小(基本块的大小)

[0142] -MinQTSIZE:允许的最小四叉树叶节点大小

[0143] -MaxBTSIZE:允许的最大二叉树根节点大小

[0144] -MaxBTDepth:允许的最大二叉树深度

[0145] -MaxBTSIZE:允许的最小二叉树叶节点大小

[0146] 在QTBT划分结构的一些示例实现中,CTU大小可以被设置为 128×128 个亮度样本和两个对应的 64×64 个色度样本块(当考虑并使用示例色度子采样时),MinQTSIZE可以被设置为 16×16 ,MaxBTSIZE可以被设置为 64×64 ,MinBTSIZE(对于宽度和高度)可以被设置为 4×4 ,MaxBTDepth可以被设置为4。四叉树划分可以首先应用于CTU以生成四叉树叶节点。四叉树叶节点的大小可以从其最小允许大小 16×16 (即MinQTSIZE)到 128×128 (即CTU大小)。如果一个节点是 128×128 ,它将不会首先被二叉树分割,因为它的大小超过了MaxBTSIZE(即 64×64)。否则,不超过MaxBTSIZE的节点可以由二叉树进行划分。在图14的例子中,基本块是 128×128 。根据预定义的规则集,基本块只能进行四叉树分割。基本块的划分深度为0。得到的四个分区中的每一个都是 64×64 ,不超过MaxBTSIZE,可以在级别1进一步进行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分割。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分割。类似地, 当二叉树节点的高度等于MinBTSIZE时, 不再考虑进一步的垂直分割。

[0147] 在一些示例实现中, 上述QTBT方案可以被配置为支持亮度和色度具有相同QTBT结构或分离QTBT结构的灵活性。例如, 对于P切片和B切片, 一个CTU中的亮度和色度CTB可以共用相同的QTBT结构。然而, 对于I切片, 亮度CTB可以通过QTBT结构划分成CBs, 色度CTB可以通过另一QTBT结构划分成色度CBs。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切片或B切片中的CU可以由所有三个颜色分量的编码块组成。

[0148] 在一些其他实现中, QTBT方案可以用上述三叉方案进行补充。这种实现可以被称为多类型树 (multi-type-tree, MTT) 结构。例如, 除了节点的二叉分割之外, 可以选择图13的三叉划分模式之一。在一些实现中, 只有正方形节点可以进行三叉分割。可以使用附加标志来指示三叉划分是水平的还是垂直的。

[0149] 两级或多级树的设计, 如QTBT实现和通过三叉分割进行补充的QTBT实现, 可能主要是为了复杂性降低。理论上, 遍历树的复杂度是 T^D , 其中T表示分割类型的数量, D是树的深度。可以通过使用多种类型(T) 同时减少深度(D) 来进行折衷。

[0150] 在一些实现中, 可以对CB进行进一步划分。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进一步划分成多个预测块。换句话说, CB可以进一步划分为不同的子分区, 在这些子分区中可以做出单独的预测决策/配置。并行地, 为了描绘执行视频数据的变换或逆变换的水平, 可将CB进一步划分成多个变换块 (transform block, TB)。将CB划分为PBs和TBs的方案可以相同, 也可以不相同。例如, 每个划分方案可以使用其自己的过程来执行, 例如, 基于视频数据的各种特征。在一些示例实现中, PB和TB的划分方案可能是独立的。在其他一些示例实现中, PB和TB的划分方案和边界可能相互关联。例如, 在一些实现中, TB可以在PB划分之后进行划分, 特别地, 每个PB在编码块划分之后确定, 然后可以进一步划分为一个或多个TB。例如, 在一些实现中, 一个PB可以分成一个、两个、四个或其他数量的TB。

[0151] 在一些实现中, 为了将基本块划分为编码块并进一步划分为预测块和/或变换块, 亮度通道和色度通道可以进行不同地处理。例如, 在一些实现中, 对于亮度通道, 可以允许将编码块划分为预测块和/或变换块, 而对于色度通道, 可以不允许将编码块划分为预测块和/或变换块。因此, 在这样的实现中, 亮度块的变换和/或预测可以仅在编码块级别上执行。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允许划分成比色度通道更小的变换和/或预测块。对于又一示例, 将编码块划分成变换块和/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, 可以允许将亮度通道的编码块划分成比色度通道更深的变换块和/或预测块。对于特定示例, 亮度编码块可以被划分成多个大小的变换块, 这些变换块可以用递归划分表示, 递归划分最多可达2级, 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 4×4 到 64×64 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。

[0152] 在将编码块划分为PBs的一些示例实现中, PB划分的深度、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。

[0153] 将编码块(或预测块)划分为变换块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分割和预定模式分割,并且附加考虑在编码块或预测块的边界处的变换块。通常,得到的变换块可以处于不同的分割级别,可以不具有相同的大小,且可以不需要为正方形形状(例如,它们可以是具有一些允许的大小和纵横比的矩形)。下面结合图15、图16和图17更详细地描述其他示例。

[0154] 然而,在一些其他实现中,经由上述任何划分方案获得的CBs可作用于预测和/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,不执行进一步分割。例如,根据上述QTBT方案获得的CBs可以直接用作执行预测的单元。具体地,这种QTBT结构消除了多种划分类型的概念,即,它消除了CU、PU和TU的分离,且针对如上所述的CU/CB划分形状提供了更大的灵活性。在这种QTBT块结构中,CU/CB可以具有正方形或矩形形状。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划分。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。

[0155] 上述各种CB划分方案以及将CB进一步划分为PBs和/或TBs(不包括PB/TB划分)可以以任何方式组合。以下具体实现作为非限制性示例提供。

[0156] 下面描述编码块和变换块划分的具体示例实现。在这样的示例实现中,可以使用递归四叉树分割或上述预定义的分割模式(例如图9和图10中的模式)将基本块分割成编码块。在每个级别,是否应该继续对特定分区进行进一步的四叉树分割可以由本地视频数据特征来确定。所得到的CBs可以处于各种四叉树分割级别,并且具有各种大小。关于是否使用图片间(时间)或图片内(空间)预测来编码图片区域的决定可以在CB级别(或CU级别,对于所有三色通道)做出。每个CB可以根据预定义的PB分割类型进一步分割成一个、两个、四个或其他数量的PBs。在一个PB内,可以应用相同的预测处理,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分割类型的预测过程获得剩余块之后,可以根据类似于CB的编码树的另一四叉树结构将CB划分成TBs。在该特定实现中,CB或TB可以是但不限于正方形。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,而对于帧内预测,PB可以仅是正方形。编码块可以被分成例如四个正方形TB。每个TB可以被进一步递归地分割(使用四叉树分割)成更小的TB,称为剩余四叉树(Residual Quadtree, RQT)。

[0157] 下面进一步描述用于将基本块划分为CBs、PBs和/或TBs的另一示例实现。例如,可以使用采用二叉和三叉分割结构的内嵌多类型树的四叉树(例如,QTBT或如上所述的具有三叉分割的QTBT),而不是使用诸如图9或图10中示出的多个划分单元类型。CB、PB和TB的分离(即将CB划分为PBs和/或TBs,将PBs划分为TBs)可能会被放弃,除非CBs的大小对于最大变换长度来说太大,这样的CBs可能需要进一步分割。该示例划分方案可以被设计成支持CB划分形状的更大灵活性,使得预测和变换都可以在CB级别上执行,而无需进一步划分。在这样的编码树结构中,CB可以具有正方形或矩形形状。具体地,编码树块(CTB)可以首先通过四叉树结构划分。然后,可以通过内嵌多类型树结构进一步划分四叉树叶节点。图11中示出了使用二叉或三叉分割的内嵌多类型树结构的示例。具体地,图11的示例性多类型树结构包括四种分割类型,分别称为垂直二叉分割(SPLIT_BT_VER)(1102)、水平二叉分割(SPLIT_BT_HOR)(1104)、垂直三叉分割(SPLIT_TT_VER)(1106)和水平三叉分割(SPLIT_TT_HOR)(1108)。然后,CBs对应于多类型树的叶子。在该示例实现中,除非CB对于最大变换长度来说太大,否则该分割既用于预测又用于变换处理,而无需任何进一步的划分。这意味着,在大

多数情况下,在具有内嵌多类型树编码块结构的四叉树中,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,会出现例外。在一些实现中,除了二叉或三叉分割之外,图11的内嵌模式还可以包括四叉树分割。

[0158] 图12示出了对一个基本块进行块划分(包括四叉树、二叉树和三叉分割选项)的内嵌多类型树编码块结构的四叉树的具体示例。更详细地,图12示出了基本块1200被四叉树分割成四个正方形分区1202、1204、1206和1208。对于每一个四叉树分割分区,决定进一步使用图11的多类型树结构和四叉树进行进一步分割。在图12的示例中,分区1204没有被进一步分割。分区1202和分区1208各自采用另一四叉树分割。对于分区1202,二级四叉树分割的左上、右上、左下和右下分区分别采用四叉树、图11的水平二叉分割1104、非分割和图11的水平三叉分割1108的三级分割。分区1208采用另一种四叉树分割,二级四叉树分割的左上、右上、左下、右下分区分别采用图11的垂直三叉分割1106、不分割、不分割、图11的水平二叉分割1104的三级分割。分别根据图11的水平二叉分割1104和水平三叉分割1108进一步分割1208的第三级左上划分的两个子分区。分区1206采用继图11的垂直二叉分割1102之后的第二级分割模式,分成两个分区,再按照图11的水平三叉分割1108和垂直二叉分割1102进行第三级分割。根据图11的水平二叉分割1104,进一步将第四级分割应用于其中一个。

[0159] 对于上面的特定示例,最大亮度变换大小可以是 64×64 ,并且支持的最大色度变换大小可以不同于亮度,例如 32×32 。即使图12中的上述示例CBs通常不被进一步分割成更小的PBs和/或TBs,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以在水平和/或垂直方向上自动分割,以满足该方向上的变换尺寸限制。

[0160] 在上述用于将基本块划分为CBs的具体示例中,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切片和B切片,一个CTU中的亮度和色度CTB可以共用相同的编码树结构。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。当采用单独的块树结构时,可以用一种编码树结构将色度CTB划分为色度CBs,用另一种编码树结构将色度CTB划分为色度CBs。这意味着I切片中的CU可能由一个亮度分量的编码块或两个色度分量的编码块组成,而P切片或B切片中的CU总是由所有三种颜色分量的编码块组成,除非视频是单色的。

[0161] 当编码块被进一步划分为多个变换块时,其中的变换块可以按照各种顺序或扫描方式在码流中排序。下面将进一步详细描述将编码块或预测块划分成变换块以及变换块的编码顺序的示例实现。在一些示例实现中,如上所述,变换划分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,变换块大小范围从 4×4 到 64×64 不等。在一些实现中,如果编码块小于或等于 64×64 ,则变换块划分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两者可分别被隐式地分割成 $\min(W, 64) \times \min(H, 64)$ 和 $\min(W, 32) \times \min(H, 32)$ 变换块的倍数。

[0162] 在变换块划分的一些示例实现中,对于帧内编码块和帧间编码块,编码块可以进一步划分成多个变换块,其划分深度可达预定数量的级别(例如2个级别)。变换块划分深度和大小可以相关联。对于一些示例实现,从当前深度的变换块大小到下一深度的变换块大小的映射如下表1所示。

[0163] 表1变换块划分大小设置

	当前深度的变换块大小	下一深度的变换块大小
	TX_4×4	TX_4×4
	TX_8×8	TX_4×4
	TX_16×16	TX_8×8
	TX_32×32	TX_16×16
	TX_64×64	TX_32×32
	TX_4×8	TX_4×4
	TX_8×4	TX_4×4
[0164]	TX_8×16	TX_8×8
	TX_16×8	TX_8×48
	TX_16×32	TX_16×16
	TX_32×16	TX_16×16
	TX_32×64	TX_32×32
	TX_64×32	TX_32×32
	TX_4×16	TX_4×8
	TX_16×4	TX_8×4
	TX_8×32	TX_8×16
	TX_32×8	TX_16×8
[0165]	TX_16×64	TX_16×32
	TX_64×16	TX_32×16

[0166] 基于表1的示例映射,对于1:1的正方形块,下一级变换分割可以创建四个1:1的正方形子变换块。变换块划分可以在例如4×4处停止。因此,当前深度为4×4的变换块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,下一级变换分割可以创建两个1:1的正方形子变换块,而对于1:4/4:1的非正方形块,下一级变换分割可以创建两个1:2/2:1的子变换块。

[0167] 在一些示例中,对于帧内编码块的亮度分量,可以相对于变换块划分施加额外的限制。例如,对于变换划分的每一级,所有子变换块可以被限制为具有相等的大小。例如,对于32×16编码块,1级变换分割创建两个16×16子变换块,2级变换分割创建八个8×8子变换块。换句话说,第二级分割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中示出了按照表1将帧内编码的正方形块划分成变换块的示例,以及由箭头示出的编码顺序。具体地,1502示出了正方形编码块。在1504中示出了根据表1将编码块进行第一级划分而分成4个大小相等的变换块,这4个变换块的编码顺序由箭头指示。在1506中示出了根据表1将所有第一级的大小相等的块进行第二级划分而分成16个大小相等的变换块,这16个变换块的编码顺序由箭头指示。

[0168] 在一些示例实现中,对于帧间编码块的亮度分量,可能不适用上述对帧内编码的限制。例如,在第一级变换分割之后,任何一个子变换块可以被进一步独立地分割为一个以上的级别。因此,得到的变换块可以大小相同,也可以大小不相同。图16中示出了将帧间编

码块分割成具有其编码顺序的多个变换块的示例。在图16的示例中,根据表1,帧间编码块1602被分成两个级别的变换块。在第一级,帧间编码块被分割成大小相等的四个变换块。然后,四个变换块中只有一个(不是全部)被进一步分割成四个子变换块,从而产生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。

[0169] 在一些示例实现中,对于一个或多个色度分量,可以应用对变换块的一些附加限制。例如,对于一个或多个色度分量,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 8×8 。

[0170] 在其他一些示例实现中,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分割成 $\min(W, 64) \times \min(H, 64)$ 和 $\min(W, 32) \times \min(H, 32)$ 个变换单元的倍数。这里,在本公开中,“ $\min(a, b)$ ”可以返回a和b之间的较小值。

[0171] 图17进一步示出了用于将编码块或预测块划分成多个变换块的另一替代示例方案。如图17所示,可以根据编码块的变换类型对编码块应用一组预定义的划分类型,而不是使用递归变换划分。在图17所示的具体示例中,可以应用6种示例划分类型中的一种来将编码块分割成各种数量的变换块。这种生成变换块划分的方案可以应用于编码块或预测块。

[0172] 更详细地,图17的划分方案为任何给定的变换类型提供了多达6个示例划分类型(变换类型指的是例如主变换的类型,例如ADST等)。在该方案中,可以基于例如率失真成本给每个编码块或预测块分配一个变换划分类型。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该编码块或预测块的变换划分类型。特定的变换划分类型可以对应于变换块分割大小和模式,如图17中所示的6个变换划分类型所示。可以预定义各种变换类型和各种变换划分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划分类型:

[0173] • PARTITION_NONE(不划分):分配一个等于块大小的变换大小。

[0174] • PARTITION_SPLIT(分割划分):分配一个变换大小,其宽度为块大小宽度的二分之一,高度为块大小高度的二分之一。

[0175] • PARTITION_HORZ(水平划分):分配一个变换大小,其宽度与块大小的宽度相同,高度为块大小高度的二分之一。

[0176] • PARTITION_VERT(垂直划分):分配一个变换大小,其宽度为块大小宽度的二分之一,高度与块大小的高度相同。

[0177] • PARTITION_HORZ4(水平4划分):分配一个变换大小,其宽度与块大小的宽度相同,高度为块大小高度的四分之一。

[0178] • PARTITION_VERT4(垂直4划分):分配一个变换大小,其宽度为块大小宽度的四分之一,高度与块大小的高度相同。

[0179] 在上面的例子中,如图17所示的变换划分类型针对划分的变换块均包含统一的变换大小。这只是一个例子,不具有限制性。在一些其他实现中,可以对特定划分类型(或模式)中的划分的变换块使用混合的变换块大小。

[0180] 根据上述任何划分方案获得的PBs(或CBs,当没有被进一步划分成预测块时也称为PBs)可以成为用于经由帧内或帧间预测进行编码的单个块。对于当前PB的帧间预测,可以生成当前块和预测块之间的残差,对该残差进行编码,并将其包括在已编码码流中。

[0181] 可以例如以单参考模式或复合参考模式实现帧间预测。在一些实现中,可以首先在当前块(或更高级别)的码流中包含跳过标志(skip flag),用于指示当前块是否被帧间编码并且是否不被跳过。如果当前块被帧间编码,则可以在码流中进一步包含另一个标志作为一个信号,以指示是否使用单参考模式或复合参考模式来预测当前块。对于单参考模式,可以使用一个参考块来生成当前块的预测块。对于复合参考模式,可以使用两个或多个参考块,例如通过加权平均来生成预测块。复合参考模式可以称为不止一个参考模式、两个参考模式或多参考模式。可以使用一个或多个参考帧索引并另外使用指示参考块和当前块之间的位置(例如,水平和垂直像素)偏移的一个或多个对应运动矢量来识别一个或多个参考块。例如,当前块的帧间预测块可以从由参考帧中的一个运动矢量标识为单参考模式中的预测块的单参考块生成,而对于复合参考模式,预测块可以通过对由两个参考帧索引和两个相应的运动矢量指示的两个参考帧中的两个参考块进行加权平均来生成。可以对运动矢量进行编码并以各种方式将其包含在码流中。

[0182] 在一些实现中,编码或解码系统可以具有解码图像缓冲器(decoded picture buffer,DPB)。一些图像/图片可以被保存在DPB中等待被显示(在解码系统中),并且DPB中的一些图像/图片可以被用作参考帧以实现帧间预测(在解码系统或编码系统中)。在一些实现中,DPB中的参考帧可以被标记为正在编码或解码的当前图像的短期参考或长期参考。例如,短期参考帧可以包括用于执行如下操作的帧,用于对当前帧中的块进行帧间预测或以解码顺序对当前帧最接近的预定数量(例如,2个)的后续视频帧中的块进行帧间预测。长期参考帧可以包括DPB中的帧,这些帧可以用于预测在解码顺序上远离当前帧的超过预定数量数量的帧中的图像块。关于短期参考帧和长期参考帧的这种标签的信息可以被称为参考图片集(Reference Picture Set,RPS),并且该信息可以被添加到已编码码流中的每个帧的报头。已编码视频流中的每一帧可以由图像顺序计数器(Picture Order Counter,POC)标识,该图像顺序计数器根据播放序列以绝对方式或与从例如i帧开始的图像组相关的顺序进行编号。

[0183] 在一些示例实现中,可以基于RPS中的信息形成包含用于帧间预测的短期和长期参考帧的标识的一个或多个参考图片列表。例如,可以针对单向帧间预测形成单个图片参考列表,表示为L0参考(或参考列表0),双向帧间预测可形成两个图片参考列表,两个预测方向各记为L0(或参考列表0)和L1(或参考列表1)。包括在L0和L1列表中的参考帧可以以各种预定方式排序。L0和L1列表的长度可以被写入视频码流。当复合预测模式中用于通过加权平均生成预测块的多个参考帧位于待预测块的同一侧时,单向帧间预测可以是以单参考模式,或者是以复合参考模式。双向帧间预测可以仅是以复合模式,这是因为双向帧间预测涉及至少两个参考块。

[0184] 在一些实现中,可以实现用于帧间预测的合并模式(merge mode,MM)。通常,对于合并模式,当前PB的单参考预测中的运动矢量或复合参考预测中的一个或多个运动矢量可以从其他运动矢量导出,而不是独立地计算和写入。例如,在编码系统中,当前PB的当前运动矢量用当前运动矢量和其他一个或多个已经编码的运动矢量(称为参考运动矢量)之间的差来表示。这种运动矢量的差、而不是当前运动矢量的全部可以被编码并包括在码流中,并且这种差可以链接到参考运动矢量。相应地,在解码系统中,对应于当前PB的运动矢量可以基于解码的运动矢量差和与其链接的解码的参考运动矢量来导出。作为一般合并模式

(MM) 帧间预测的一种特定形式,这种基于运动矢量差的帧间预测可以被称为具有运动矢量差的合并模式(Merge Mode with Motion Vector Difference,MMVD)。因此,一般的MM或具体的MMVD可以被实现为利用与不同PB相关联的运动矢量之间的相关性来提高编码效率。例如,相邻的PB可以具有相似的运动矢量,因此,MVD可以很小,并且可以有效地编码。对于另一个示例,对于空间中类似定位/位置的块,运动矢量可以在时间上(在帧之间)相关。

[0185] 在一些示例实现中,在编码处理期间,可以在码流中包括MM标志,用于指示当前PB是否处于合并模式。另外地或可选地,在编码过程中可以将MMVD标志包括在并写入码流中以指示当前PB是否处于MMVD模式。可以以PB级、CB级、CU级、CTB级、CTU级、切片级、图像级等来提供MM和/或MMVD标志或指示符。对于特定示例,对于当前CU可以包括MM标志和MMVD标志,并且该MMVD标志可以紧接在跳过标志和MM标志之后被写入,以指定MMVD模式是否用于当前CU。

[0186] 在MMVD的一些示例实现中,针对被预测的块可以形成用于运动矢量预测的参考运动矢量(reference motion vector,RMV) 候选列表或MV预测候选列表。RMV候选列表可以包含预定数量(例如2个)的MV预测候选块,其运动矢量可以用于预测当前运动矢量。RMV候选块可以包括从同一帧的相邻块和/或临时块(例如,当前帧的前一帧或后一帧中相同位置的块)中选择的块。这些选项表示相对于当前块在空间或时间位置上的块,这些块可能具有与当前块相似或相同的运动矢量。可以预先确定MV预测器候选列表的大小。例如,该列表中可以包含两个或多个候选者。要位于RMV候选列表上,候选块例如可能需要具有与当前块相同的一个参考帧(或多个参考帧),必须存在(例如,当当前块靠近帧的边缘时,需要执行边界检查),并且必须在编码过程中已经被编码,和/或在解码过程中已经被解码。在一些实现中,如果合并候选列表可用并且满足上述条件,则可以首先使用空间上相邻的块(按照特定的预定义顺序扫描)填充,然后如果列表中仍有空间可用,则可以使用临时块填充。例如,这些相邻的RMV候选块可以从当前块的左侧块和顶部块中选择。可以以不同级别(序列、图片、帧、切片、超级块等)动态形成RMV预测候选列表作为动态参考列表(Dynamic Reference List,DRL)。DRL可以被写入码流。

[0187] 在一些实现中,可以将用作预测当前块的运动矢量的参考运动矢量的实际MV预测候选者写入码流。在RMV候选列表包含两个候选者的情况下,可以使用称为合并候选标志的一位标志来指示对参考合并候选者的选择。对于以复合模式预测的当前块,使用MV预测器预测的多个运动矢量中的每一个可以与来自合并候选列表的参考运动矢量相关联。编码器可以确定RMV候选中的哪一个能更好地预测当前编码块,并将该选择作为索引写入DRL中。

[0188] 在MMVD的一些示例实现中,在选择RMV候选者并将其用作待预测的运动矢量的基本运动矢量预测器之后,可以在编码系统中计算运动矢量差(MVD或增量(delta)MV,表示待预测的运动矢量和参考候选运动矢量之间的差)。这种MVD可以包括表示MV差的大小和MV差的方向的信息,这两者可以被写入码流。运动差大小和运动差方向可以以各种方式被写入码流。

[0189] 在MMVD的一些示例实现中,距离索引可用于指定运动矢量差的大小信息,并指示一组预定义偏移量中的一个,该一组预定义偏移量表示相对于起点(参考运动矢量)的预定义运动矢量差。然后,可以根据所述信号指示的索引的MV偏移添加到起始(参考)运动矢量的水平分量或垂直分量。参考运动矢量的水平分量或垂直分量是否应该偏移,可以由MVD

的方向信息确定。表2中指定了距离索引和预定义偏移量之间的预定义关系示例。

[0190] 表2距离索引和预定义MV偏移的示例关系

	距离索引	0	1	2	3	4	5	6	7
[0191]	偏移量 (以亮度样本为单位)	1/4	1/2	1	2	4	8	16	32

[0192] 在MMVD的一些示例性实现中,方向索引可以被进一步写入码流并用于表示MVD相对于参考运动矢量的方向。在一些实现中,方向可以被限定为水平方向和垂直方向中的任一个。2位方向索引的示例如表3所示。在表3的示例中,MVD的解释可以根据起始/参考MV的信息而变化。例如,当起始/参考MV对应于单预测块或对应于双预测块,其中两个参考帧列表都指向当前图片的同一侧时(即,两个参考图片的POC都大于当前图片的POC,或者都小于当前图片的POC),表3中的符号可以指定添加到起始/参考MV的MV偏移的符号(方向)。当起始/参考MV对应于当前图片不同侧的两个参考图片的双预测块时(即,一个参考图片的POC大于当前图片的POC,而另一个参考图片的POC小于当前图片的POC),并且图片参考列表0中的参考POC与当前帧之间的差值大于图片参考列表1中的参考POC与当前帧之间的差值时,表3中的符号可以指定添加到与图片参考列表0中的参考图片相对应的参考MV的MV偏移的符号,并且与图片参考列表1中的参考图片相对应的MV的偏移的符号可以具有相反的值(对偏移而言具有相反的符号)。否则,如果图片参考列表1中的参考POC与当前帧之间的差值大于图片参考列表0中的参考POC与当前帧之间的差值,则表3中的符号可以指定添加到与图片参考列表1相关联的参考MV的MV偏移的符号,并且与图片参考列表0相关联的参考MV的偏移的符号具有相反的值。

[0193] 表3由方向索引指定的MV偏移的符号的示例实现

[0194]	方向IDX	00	01	10	11
	x轴(水平)	+	-	N/A	N/A
	y轴(垂直)	N/A	N/A	+	-

[0195] 在一些示例实现中,MVD可以根据POCs在每个方向上的差值进行缩放调整。如果两个列表中POC的差值相同,则不需要调整。否则,如果参考列表0中的POC差值大于参考列表1中的POC差值,则对参考列表1的MVD进行调整。如果参考列表1的POC差大于列表0中的POC差值,则可以以相同的方式调整列表0的MVD。如果起始MV是单预测的,则将MVD添加到可用MV或参考MV中。

[0196] 在用于双向复合预测的MVD编码和写入的一些示例实现中,除了对两个MVD单独编码和写入之外,还可以实现对称MVD编码,使得只有一个MVD需要写入,而另一个MVD可以根据写入的MVD导出。在这种实现中,包括列表-0和列表-1两者的参考图片索引的运动信息被写入码流。然而,仅将与例如参考列表-0相关联的MVD写入,而与参考列表-1相关联的MVD不被写入而是被导出。具体地,在切片级,可以在码流中包括标志,称为“mvd_11_zero_flag”,用于指示参考列表-1是否未被写入码流。如果该标志是1,指示参考列表-1等于零(因此没有写入),则称为“BiDirPredFlag”的双向预测标志可以被设置为0,这意味着没有双向预测。否则,如果mvd_11_zero_flag为零,如果列表-0中的最近参考图片和列表-1中的最近参

考图片形成向前和向后的参考图片对或向后和向前的参考图片对,则BiDirPredFlag可以被设置为1,并且列表-0和列表-1的参考图片都是短期参考图片。否则,BiDirPredFlag设置为0。BiDirPredFlag为1可以表示对称模式标志被附加地写入码流。当BiDirPredFlag为1时,解码器可以从码流中提取对称模式标志。例如,对称模式标志可以在CU级上被写入(如果需要的话),并且它可以指示对称MVD编码模式是否正被用于相应的CU。当对称模式标志为1时,表示使用对称MVD编码模式,并且仅列表-0和列表-1(称为“mvp_l0_flag”和“mvp_l1_flag”)两者的参考图片索引及与列表-0相关联的MVD(称为“MVD0”)被写入,并且另一个运动矢量差“MVD1”将被导出而不是被写入。例如,MVD1可以导出为-MVD0。因此,在示例性对称MVD模式中仅有一个MVD被写入码流。在MV预测的一些其它示例性实现中,对于单参考模式和复合参考模式MV预测,可使用协调方案来实现通用合并模式MMVD和一些其它类型的MV预测。可以使用各种语法元素来表示对当前块的MV进行预测的方式。

[0197] 例如,对于单参考模式,以下MV预测模式可以被写入码流:

[0198] NEARMV-直接使用动态参考列表(Dynamic Reference List,DRL)索引指示的列表中的运动矢量预测器(motion vector predictor,MVP)之一,而不使用任何MVD。

[0199] NEWMV-使用由DRL索引写入的列表中的运动矢量预测器(MVPs)之一作为参考,并对MVP应用一个增量(例如,使用MVD)。

[0200] GLOBALMV-使用基于帧级全局运动参数的运动矢量。

[0201] 同样地,对于使用与待预测的两个MV对应的两个参考帧的复合参考帧间预测模式,以下MV预测模式可以被写入码流:

[0202] NEAR_NEARMV-对于要预测的两个MV中的每一个,使用由DRL索引写入的列表中的运动矢量预测器(MVP)之一,而不使用MVD。

[0203] NEAR_NEWMV-为了预测两个运动矢量中的第一个运动矢量,使用由DRL索引写入的列表中的运动矢量预测器(MVP)中的一个作为参考MV,而不使用MVD;为了预测两个运动矢量中的第二个运动矢量,使用由DRL索引写入的列表中的运动矢量预测器(MVP)中的一个作为参考MV,并结合另外写入的增量MV(MVD)。

[0204] NEW_NEARMV-为了预测两个运动矢量中的第二个运动矢量,使用由DRL索引写入的列表中的运动矢量预测器(MVP)中的一个作为参考MV,而不使用MVD;为了预测两个运动矢量中的第一个运动矢量,使用由DRL索引写入的列表中的运动矢量预测器(MVP)中的一个作为参考MV,并结合另外写入的增量MV(MVD)。

[0205] NEW_NEWMV-使用由DRL索引写入的列表中的运动矢量预测器(MVP)中的一个作为参考MV,并与另外写入的增量MV结合使用,以预测两个MV中的每一个。

[0206] GLOBAL_GLOBALMV-根据帧级全局运动参数使用每个参考的MV。

[0207] 因此,上述术语“NEAR”是指使用参考MV而不使用MVD作为一般合并模式的MV预测,而术语“NEW”是指涉及使用参考MV并用写入的MVD对其进行偏移作为MMVD模式的MV预测。对于复合帧间预测,上述的参考基本运动矢量和运动矢量增量两者在两个参考之间通常可以不同或独立,即使它们可以相关,并且可以利用这种相关性来减少写入两个运动矢量增量所需的信息量。在这种情况下,可以实现两个MVD的联合写入并指示在码流中。

[0208] 上面的动态参考列表(dynamic reference list,DRL)可以用于保存一组有索引的运动矢量,这一组运动矢量被动态保存并被认为是候选运动矢量预测器。

[0209] 在一些示例实现中,可以允许预定义的MVD的分辨率。例如,可以允许1/8像素的运动矢量精度(或准确度)。以上在各种MV预测模式中描述的MVD可以被构建并以各种方式写入码流。在一些实现中,可以使用各种语法元素来表示参考帧列表0或列表1中的上述运动矢量差。

[0210] 例如,称为“mv_joint”的语法元素可以指定与其相关联的运动矢量差的哪些分量是非零的。对于MVD,这是针对所有非零分量进行联合写入。例如,mv_joint具有以下值。

[0211] 0可表示沿水平或垂直方向不存在非零MVD;

[0212] 1可表示仅在水平方向上存在非零MVD;

[0213] 2可表示仅在垂直方向上存在非零MVD;

[0214] 3可表示在水平和垂直方向上都存在非零MVD。

[0215] 当MVD信号的“mv_joint”语法元素表示不存在非零MVD分量时,则没有进一步的MVD信息可被写入。如果“mv_joint”语法表示存在一个或两个非零分量,则可进一步针对每个非零MVD分量写入附加的语法元素,如下所述。

[0216] 例如,称为“mv_sign”的语法元素可用于附加地指定对应的运动矢量差分量是正的还是负的。

[0217] 对于另一示例,称为“mv_class”的语法元素可用于针对对应的非零MVD分量指定在预定义类别集合中运动矢量差的类别。例如,运动矢量差的预定义类别可用于将运动矢量差的连续大小空间划分成不重叠的范围,每个范围对应一个MVD类别。因此,写入的MVD类别指示相应MVD分量的大小范围。在以下表4所示的示例实现中,较高的类别对应于具有较大的大小范围的运动矢量差。在表4中,符号(n,m)用于表示大于n个像素且小于或等于m个像素的运动矢量差的范围。

[0218] 表4运动矢量差的大小类别

[0219]

MV类别	MVD的大小
MV_CLASS_0	(0, 2]
MV_CLASS_1	(2, 4]
MV_CLASS_2	(4, 8]
MV_CLASS_3	(8, 16]
MV_CLASS_4	(16, 32]
MV_CLASS_5	(32, 64]
MV_CLASS_6	(64, 128]
MV_CLASS_7	(128, 256]
MV_CLASS_8	(256, 512]
MV_CLASS_9	(512, 1024]
MV_CLASS_10	(1024, 2048]

[0220] 在一些其它示例中,称为“mv_bit”的语法元素可进一步用于指定非零运动矢量差分量与相应写入的MV类别大小范围的起始大小之间的偏移的整数部分。“mv_bit”中用于写入每个MVD类别的整个范围所需的位数可以随着MV类别的函数而变化。例如,在表4的实现中,MV_CLASS_0和MV_CLASS_1可能只需要单个比特来表示从MVD为0开始的整数像素偏移量为1或2。在表4的示例实现中,每个更高的MV_CLASS可能需要比前一个MV_CLASS逐渐增加一

个“mv_bit”位。

[0221] 在一些其它示例中,称为“mv_fr”的语法元素可进一步用于针对对应的非零MVD分量指定运动矢量差的前2个小数位,而称为“mv_hp”的语法元素可用于针对对应的非零MVD分量指定运动矢量差的第三小数位(高分辨率位)。两位“mv_fr”实际上提供了1/4像素的MVD分辨率,而“mv_hp”位可以进一步提供1/8像素的分辨率。在一些其它实现中,可以使用多个“mv_hp”位来提供比1/8像素更精细的MVD像素分辨率。在一些示例实现中,可以在一个或多个不同级别将附加标志写入码流,以指示是否支持1/8像素或更高的MVD分辨率。如果MVD分辨率没有应用于特定的编码单元,那么上面用于对应的不支持的MVD分辨率的语法元素可能不会被写入码流。

[0222] 在上面的示例实现中,小数分辨率可能与不同类别的MVD无关。换句话说,不管运动矢量差的大小如何,可以使用预定义数量的“mv_fr”和“mv_hp”位来将非零MVD分量的小数MVD写入,从而提供类似的用于运动矢量分辨率的选项。

[0223] 然而,在一些其他示例实现中,可以对各种MVD大小类别中运动矢量差的分辨率进行区分。具体地,对于较高MVD类别中大小较大的MVD,高分辨率MVD可能不会在压缩效率方面提供统计学意义上的显著改善。因此,对于更大的MVD大小范围,可以用递减的分辨率(整数像素分辨率或小数像素分辨率)对MVD进行编码,所述更大的MVD大小范围对应于更高的MVD大小类别。同样,对于通常较大的MVD值,可以用递减的分辨率(整数像素分辨率或小数像素分辨率)对MVD进行编码。这种依赖于MVD类别或依赖于MVD大小的MVD分辨率通常可称为自适应MVD分辨率、依赖于幅度的自适应MVD分辨率或依赖于大小的MVD分辨率。术语“分辨率”可进一步称为“像素分辨率”。自适应MVD分辨率可以在以下示例实现中描述的各种情况下实现,以实现总体上更好的压缩效率。特别地,由于统计观察到以非适应的方式类似于大小较小或低级别MVD的MVD分辨率对大小较大或高级别MVD的MVD分辨率进行处理可能不会显著增加大小较大或高级别MVD的块的帧间预测的残差编码效率,因此通过侧重于较低精度MVD而减少的信令的位数可能大于由于这种较低精度MVD而导致帧间预测的残差编码所需的附加位数。换句话说,对于大小较大或高级别的MVD,使用较高的MVD分辨率可能不会比使用较低的MVD分辨率产生太多的编码增益。

[0224] 在一些一般示例实现中,MVD的像素分辨率或精度可以随着MVD类别的增加而降低或不增加。降低MVD的像素分辨率对应于更粗的MVD(或从一个MVD级别到下一个MVD级别的更大步长)。在一些实现中,MVD像素分辨率和MVD类别之间的对应关系可以被指定、预定义或预配置,因此可以不需要被写入已编码码流。

[0225] 在一些示例实现中,表3的MV类别可以各自与不同的MVD像素分辨率相关联。

[0226] 在一些示例实现中,每个MVD类别可以与单个允许的分辨率相关联。在一些其它实现中,一个或多个MVD类别可以与两个或更多个可选的MVD像素分辨率相关联。因此,在将具有这种MVD类别的当前MVD分量写入码流中之后,接着是指示为当前MVD分量选择哪个可选的像素分辨率的附加写入。

[0227] 在一些示例实现中,自适应允许的MVD像素分辨率可以包括但不限于1/64像素、1/32像素、1/16像素、1/8像素、1-4像素、1/2像素、1像素、2像素、4像素……(按照分辨率的降序顺序)。因此,每个升序MVD类别可以以非升序方式与这些分辨率中的一个相关联。在一些实现中,MVD类别可以与两个或更多个分辨率相关联,并且较高的分辨率可以低于或等于先

前MVD类别的较低分辨率。例如,如果表4的MV_CLASS_3可以与可选的1像素和2像素分辨率相关联,则表4的MV_CLASS_4可以相关联的最高分辨率将是2像素。在一些其他实现中,MV类别的最高允许分辨率可能高于先前(较低)MV类别的最低允许分辨率。然而,升序的MV类别的允许的平均分辨率可能只是非升序的。

[0228] 在一些实现中,当允许小数像素分辨率高于1/8像素时,“mv_fr”和“mv_hp”信令可以相应地扩展到总共大于3个小数位。

[0229] 在一些示例实现中,小数像素分辨率可以仅被允许用于低于或等于阈值MVD类别的MVD类别。例如,小数像素分辨率可以仅允许用于MVD-CLASS 0,而不允许用于表4的所有其他MV类别。同样,小数像素分辨率可以仅允许用于低于或等于表4的任何一个其他MV类别的MVD类别。对于高于阈值MVD类别的其他MVD类别,仅允许MVD的整数像素分辨率。以这种方式,对于用高于或等于阈值MVD类别的MVD类别写入的MVD,可以不需要写入诸如“mv_fr”和/或“mv_hp”位中的一个或多个的小数分辨率信令。对于分辨率低于1像素的MVD类,可以进一步减少“mv_bit”信令中的位数。例如,对于表4中的MV_CLASS_5,MVD像素偏移的范围是(32, 64],因此需要5位来写入1像素分辨率的整个范围。然而,如果MV_CLASS_5与2像素MVD分辨率(低于1像素分辨率)相关联,则“mv_bit”可能需要4位而不是5位,并且在将“mv_class”写为“MV_CLASS_5”之后“mv_fr”和“mv_hp”都不需要被写入。

[0230] 在一些示例实现中,具有低于阈值整数像素值的整数值的MVD可仅允许小数像素分辨率。例如,对于小于5像素的MVD,可能只允许小数像素分辨率。对应于该示例,对于表4的MV_CLASS_0和MV_CLASS_1可以允许小数分辨率,而对于所有其它MV类别则不允许小数分辨率。再例如,对于小于7像素的MVD,可能只允许小数像素分辨率。与该示例相对应,对于表4中的MV_CLASS_0和MV_CLASS_1(范围小于5像素)可以允许小数分辨率,而对于MV_CLASS_3及更高的(范围大于5像素)则不允许小数分辨率。对于像素范围包含5个像素的属于MV_CLASS_2的MVD,根据“mv_bit”值,MVD的小数像素分辨率可以被允许或不被允许。如果“mv_bit”值被写入为1或2(使得写入的MVD的整数部分是5或6,计算为MV_CLASS_2的像素范围的开始,偏移量为1或2,由“mv_bit”表示),则可以允许小数像素分辨率。否则,如果“mv_bit”值被写入为3或4(使得写入的MVD的整数部分是7或8),则小数像素分辨率可能不被允许。

[0231] 在一些其他实现中,对于等于或高于阈值MV类别的MV类别,可以仅允许单个MVD值。例如,这样的阈值MV类别可以是MV_CLASS2。因此,MV_CLASS_2及以上可仅被允许具有单个MVD值并且没有小数像素分辨率。这些MV类别的单个允许MVD值可以是预定义的。在一些示例中,允许的单个值可以是表4中这些MV类别的各自范围的较高的端值。例如,MV_CLASS_2到MV_CLASS_10可以高于或等于MV_CLASS2的阈值类别,并且这些类别的单个允许MVD值可以分别预定义为8、16、32、64、128、256、512、1024和2048。在其他一些示例中,允许的单个值可能是表4中这些MV类别各自范围的中间值。例如,MV_CLASS_2到MV_CLASS_10可以高于类别阈值,并且这些类别的单个允许MVD值可以分别预定义为3、6、12、24、48、96、192、384、768和1536。范围内的任何其他值也可以被定义为各个MVD类别的单个允许分辨率。

[0232] 在上述实现中,在写入的“mv_class”等于或高于预定义的MVD等级阈值时,仅“mv_class”信令就足以确定MVD值。然后使用“mv_class”和“mv_sign”来确定MVD的大小和方向。

[0233] 因此,当MVD仅针对一个参考帧(来自参考帧列表0或列表1,但不是两者)写入时,或者针对两个参考帧联合写入时,MVD的精度(或分辨率)可以取决于表3中相关联的运动矢

量差的类别和/或MVD的大小。

[0234] 在一些其它实现中,MVD的像素分辨率或精度可以随着MVD大小的增加而降低或不增加。例如,像素分辨率可以取决于MVD大小的整数部分。在一些实现中,仅对于小于或等于振幅阈值的MVD大小,可以允许小数像素分辨率。对于解码器,可以首先从码流中提取MVD大小的整数部分。然后可以确定像素分辨率,并且然后可以决定码流中是否存在任何小数MVD并且需要被解析(例如,如果对于特定提取的MVD整数大小不允许小数像素分辨率,那么需要提取的码流中可能不包含小数MVD位)。上述与依赖于MVD类别的自适应MVD像素分辨率相关的示例实现适用于依赖于MVD大小的自适应MVD像素分辨率。对于特定示例,高于或包含大小阈值的MVD类别可能只被允许有一个预定义值。

[0235] 上面的各种示例实现用于单参考模式。这些实现也适用于MMVD下复合预测中的NEW_NEARMV、NEAR_NEWMV和/或NEW_NEWMV模式示例。这些实现通常适用于任何MVD的自适应分辨率。

[0236] 当采用MVD的自适应(更具体地为大小自适应)像素分辨率时,与编码块的MV和MVD相关的各参数可以是相互依赖的。被认为与MV或MVD相关的参数广义上指那些可以影响如何选择和检测RMV、如何写入、计算或导出RMV和MVD的信息项。这些MV或MVD相关的参数可包括但不限于:

[0237] • 上述用于标识对当前编码块的MV进行预测的有序的候选参考运动矢量(RMV)列表的动态参考列表(DRL)。例如,DRL可以标识可能与当前块具有相似的运动矢量的一组空间或时间相邻块位置。这些位置对应于可用作预测当前运动矢量的候选的RMVs。编码器可以从这些候选RMVs中选择与当前编码块的RMV最匹配的RMV,然后使用该RMV并导出相应的MVD。所选择的RMV例如可以由DRL中的相应位置或索引来表示或标识。

[0238] • 对应于为当前编码块所选择的候选RMV的DRL索引。

[0239] • 对采用自适应MVD像素分辨率的指示。

[0240] • MVD信息,包括但不限于如上所述的mv_joint、mv_sign、mv_class、mv_bit、mv_fr、mv_hp。

[0241] • 对使用诸如重叠块运动补偿模式的运动补偿模式的指示。

[0242] • 对使用诸如扭曲运动模式的提前运动补偿模式的指示。

[0243] 因为这些信息项或参数可以是相互依赖的,特别是当使用自适应MVD像素分辨率时,因此无论是被写入还是被导出的,它们被写入的顺序、用于写入的语法的数量、用于编码/写入这些语法元素的上下文的导出都可以在设计更有效的编码-解码方案时被考虑在内。

[0244] 在一些示例实现中,与MV或MVD相关的一个或多个参数是被写入视频流中还是从其他写入的信息导出可以取决于是否应用自适应MVD分辨率。可替换地或附加地,如果在视频流中写入这样的特定参数,则其写入的方式可以取决于是否应用自适应MVD分辨率。

[0245] 例如,当应用自适应MVD像素分辨率时,可以指定在编码器中仅允许使用DRL中的前N个RMV候选。换句话说,可能需要编码器从以自适应MVD像素分辨率进行帧间编码的当前编码块的DRL中选择前N个条目中的一个。在这样的实现中,如果不应用自适应MVD像素分辨率,编码器可以自由地从整个DRL列表中选择用于当前MV的预测。这种实现的基础可以基于这样的观察,即在DRL中具有较低索引(对应于更接近的空间相邻块)的RMV候选在统计上更

有可能更好地预测当前块的运动矢量。以这种方式,针对使用自适应MVD像素分辨率的块而写入码流的DRL索引可以需要更少数量的位,并且可以例如使用单独的语法元素用于写入。

[0246] 这里,N是一个正数。N可以小于DRL的整个索引空间。在一些具体实现中,N可以是1或2,这意味着当当前块使用自适应MVD像素分辨率时,预测器MV或RMV总是DRL中的第一RMV候选、或者第一或第二RMV候选。

[0247] N的值可以是预定义的,或者可以写入码流中。例如,N可以以各种编码级别被写入,包括但不限于序列级、帧级、片级、标题级或超级块级。N的值应用于在写入的级别内采用自适应MVD像素分辨率的各种块。

[0248] 在一些实现中,当N值被指定或写入为1时,则对于采用自适应MVD像素分辨率的块,不需要将DRL索引写入码流。基于DRL提取RMV的索引将在解码器处自动导出为1,指的是DRL中的第一RMV候选。例如,当N值被指定或写入为2时,可以使用单个比特来写入编码块的DRL索引。

[0249] 在一些其它实现中,以上描述的用于写入或导出DRL索引的方式可仅在帧间预测是单参考模式时适用。换句话说,当应用自适应MVD像素分辨率时,并且当帧间预测是单参考模式而不是复合参考模式时,可以指定或写入仅允许编码器使用DRL中的前N个MVP候选。在这种条件下,DRL索引的范围的导出或写入可以类似于以上描述的导出或写入,例如,N可以是有限的,且可以被写入或预定义。当帧间预测模式是复合参考模式时,或者当帧间预测模式是单参考模式但没有采用自适应MVD像素分辨率时,编码器可以从对应于DRL的全索引范围的候选RMV中选择当前编码块的RMV。

[0250] 因此,针对各编码块写入码流的DRL索引可以在范围(或比特位数)上变化。对于采用自适应MVD像素分辨率的编码块,如果这些编码块被写入,它们的DRL索引可能是从1到N,而对于不采用自适应MVD像素分辨率的编码块,它们的DRL索引可能是从1到DRL的全索引范围。因此,在这些不同情况下DRL索引可以遵循不同的概率模型,因此根据是否采用自适应MVD像素分辨率,可以以不同方式采用/导出写入的DRL索引的熵编码上下文。换句话说,当采用自适应MVD像素分辨率时,可以采用一组上下文来写入DRL索引。否则,可以采用另一组上下文来写入DRL索引。在解码器处用于解码DRL索引的上下文的导出将相应地取决于是否采用自适应MVD像素分辨率。

[0251] 在一些示例实现中,诸如OBMC模式和/或扭曲运动等运动补偿模式的写入可以取决于是否采用自适应MVD像素分辨率。例如,用于是否使用OBMC模式和/或扭曲运动的标志可以仅当对当前编码块采用自适应MVD像素分辨率时才被写入。否则,这样的标志不被写入(例如,解码器可以认为不采用OBMC模式和/或扭曲运动)。

[0252] 在一些示例实现中,上述OBMC模式和/或扭曲运动的写入也可以以当前编码块的帧间预测模式是单参考模式而不是复合参考模式为条件。换句话说,只有当当前编码块与单个参考帧相关联并且采用自适应MVD像素分辨率时,才可以写入OBMC模式和/或扭曲运动标志。

[0253] 在一些其他示例实现中,自适应MVD像素分辨率可仅与单参考帧间预测模式一起使用。因此,当自适应MVD像素分辨率用于通过写入或通过导出确定的当前编码块时,这表明帧间预测不是复合参考模式。在这种情况下,不需要在码流中写入标志来指示当前块是使用单参考模式还是复合参考模式。

[0254] 在一些其他示例实现中,是否使用用于指示是否为复合帧间-帧内模式的标志可以取决于是否应用自适应MVD像素分辨率以及当前块是否为单参考模式。例如,可能仅在单参考模式中并且仅当帧间预测基于采用自适应MVD像素分辨率时,用于当前块的混合帧间和帧内预测的复合帧间-帧内模式才被潜在地使用。因此,不需要标志来写入是否使用复合帧间-帧内模式,并且当当前块不是通过单参考模式预测的或者不依赖于自适应MVD像素分辨率时,认为不使用复合帧间-帧内模式。

[0255] 在一些其他示例实现中,用于写入其他MVD相关语法的上下文导出(context derivation)可以取决于是否应用自适应MVD像素分辨率。如以上所描述的,这些MVD相关语法同样可以包括mv_joint、mv_class、mv_bit、mv_fr、mv_hp等。例如,mv_joint和/或mv_class可以在统计上与自适应MVD像素分辨率相关,并且可以根据是否应用自适应MVD像素分辨率遵循不同的概率模型。具体地,如果应用自适应MVD分辨率,则可以使用/导出一个上下文来写入mv_joint(或mv_class)。否则,可能会使用/导出另一个或多个不同的上下文来写入mv_joint(或mv_class)。mv_joint和/或mv_class对采用自适应MVD像素分辨率的上下文依赖关系仅是一个示例。其它MVD相关的语法元素也可以与取决于是否应用自适应MVD像素分辨率的上下文导出相关联。

[0256] 在上面的各种示例实现中,假设在当前编码块中是否采用自适应MVD像素分辨率可以先从码流中提取(如所写入的那样)或导出,然后再确定其他一些MV或MVD相关的信息项或参数,如DRL索引等。在一些其他示例实现中,相反,可以在确定自适应MVD像素分辨率是否被用于当前编码块之前写入或导出此类其他信息项。由于这些其他信息项和是否应该使用自适应MVD像素分辨率之间的相关性或相互关系,因此是否使用自适应MVD像素分辨率的写入或导出可以取决于已经从码流提取或导出的这些其他信息项。

[0257] 例如,在一些特定实现中,是否应用自适应MVD分辨率可以取决于DRL索引的值。在这样的实现中,用于确定当前编码块的RMV的DRL索引可以首先被写入码流中(或者以一些其他方式从码流中导出)。当用于当前编码块的DRL索引被写入为在1到N范围内的值时,表明可以在有限的DRL索引范围内选择DRL中的RMV候选。在这种情况下,编码器可以进一步将标志包括在码流中,以写入当前编码块是否采用自适应MVD像素分辨率。然而,如果写入的DRL索引在范围1-N之外,则这可能指示当前编码块未采用自适应MVD像素分辨率,并且因此不需要在码流中包括指示是否采用自适应MVD像素分辨率的标志。在这种情况下,解码器将通过导出来简单地确定未使用自适应MVD像素分辨率。这里,N是一个小于DRL的全索引范围的正数。例如,N可以是1或2。在一些实现中,N可以是预定义的。在其他一些示例实现中,N可以被写入。例如,N可以以各种写入级别被写入,例如序列级、帧级、片级、标题级或超级块级之一。

[0258] 图18示出了示例方法的流程图1800,该方法遵循上述用于自适应MVD分辨率及其写入的实现原理。示例解码方法流程从S1801开始。在S1810中,接收视频流。在S1820中,基于预测块和运动矢量(MV)确定所述视频块被帧间编码,其中,所述MV将基于所述视频块的参考运动矢量(RMV)和运动矢量差(MVD)来导出。在S1830中,至少取决于所述MVD是否依赖于大小的自适应MVD像素分辨率进行编码,从所述视频流中提取或导出与所述RMV或所述MVD中的至少一个相关联的数据项。在S1840中,从所述视频流中提取所述MVD;基于所提取的RMV和所述MVD导出所述MV;以及至少基于所述MV和所述预测块重建所述视频块。示例

方法流程在S1899结束。

[0259] 图19示出了另一示例方法的流程图1900,该方案遵循上述用于自适应MVD分辨率及其写入的实现原理。示例解码方法流程从S1901开始。在S1910中,接收视频流。在S1920中,基于预测块和运动矢量(MV)确定所述视频块被帧间编码,其中,所述MV将基于所述视频块的参考运动矢量(RMV)和运动矢量差(MVD)来导出。在S1930中,对映射到动态参考列表(DRL)中的所述视频块的RMV索引进行提取,所述DRL用于识别多个有序的候选RMV。在S1940中,基于所述RMV索引的值确定所述MVD是否用依赖于大小的自适应MVD像素分辨率进行编码。示例方法流程在S1999结束。

[0260] 在本公开的实施例和实施例中,任何步骤和/或操作可以根据需要以任何数量或顺序组合或安排。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施方式可以单独使用或以任何顺序组合使用。此外,每个方法(或实施例)、编码器和解码器可以通过处理电路(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。在以下各项中,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,或面积大小(宽度*高度),或块的纵横比(宽度:高度,或高度:宽度)。

[0261] 可以将上述技术实现为计算机软件,该计算机软件使用计算机可读指令,并且物理存储在一个或多个计算机可读介质中。例如,图20示出了适合于实施所公开的主题的某些实施例的计算机系统(2000)。

[0262] 可以使用任何合适的机器代码或计算机语言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码、微码等执行。

[0263] 指令可以在各种类型的计算机或其组件上执行,例如包括个人计算机、平板计算机、服务器、智能电话、游戏设备、物联网设备等。

[0264] 图20示出的用于计算机系统(2000)的组件在本质上是示例性的,并不旨在对实现本公开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(2000)的示例性实施例中所示的组件中的任何一个组件或组件的组合有关的任何依赖或要求。

[0265] 计算机系统(2000)可以包括某些人机接口输入设备。此类人机接口输入设备可以响应于一个或多个人类用户通过例如下述的输入:触觉输入(例如:击键、划动,数据手套移动)、音频输入(例如:语音、拍手)、视觉输入(例如:手势)、嗅觉输入(未描绘出)。人机接口设备还可以用于捕获不一定与人的意识输入直接相关的某些媒介,例如音频(例如:语音、音乐、环境声音)、图像(例如:扫描的图像、从静止图像相机获取摄影图像)、视频(例如二维视频、包括立体视频的三维视频)等。

[0266] 输入人机接口装置可以包括下述中的一项或多项(每种中仅示出一个):键盘(2001)、鼠标(2002)、触控板(2003)、触摸屏(2010)、数据手套(未示出)、操纵杆(2005)、麦克风(2006)、扫描仪(2007)、相机(2008)。

[0267] 计算机系统(2000)也可以包括某些人机接口输出设备。这样的人机接口输出设备可以例如通过触觉输出、声音、光和气味/味道来刺激一个或多个人类用户的感官。此类人机接口输出设备可以包括触觉输出设备(例如触摸屏(2010)、数据手套(未示出)或操纵杆(2005)的触觉反馈,但是也可以是不作为输入设备的触觉反馈设备)、音频输出设备(例如:扬声器(2009)、耳机(未示出))、视觉输出设备(例如包括CRT屏幕、LCD屏幕、等离子屏幕、OLED屏幕的屏幕(2010),每种屏幕有或没有触摸屏输入功能,每种屏幕都有或没有触觉反馈功能-其中的一些屏幕能够通过诸如立体图像输出、虚拟现实眼镜(未描绘出)、全息显示器和烟箱(未描绘出)以及打印机(未描绘出)之类的设备来输出二维视觉输出或超过三维输出。

[0268] 计算机系统(2000)也可以包括人类可访问存储设备及其关联介质:例如包括具有CD/DVD等介质(2021)的CD/DVD ROM/RW(2020)的光学介质、指状驱动器(2022)、可拆卸硬盘驱动器或固态驱动器(2023)、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的设备(未示出)等。

[0269] 本领域技术人员还应理解,结合当前公开的主题使用的所术语“计算机可读介质”不涵盖传输介质、载波或其他暂时性信号。

[0270] 计算机系统(2000)还可以包括到一个或多个通信网络(2055)的接口(2054)。网络可以例如是无线网络、有线网络、光网络。网络可以进一步地是本地网络、广域网络、城域网络、车辆和工业网络、实时网络、延迟容忍网络等。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CAN Bus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(2049)的外部网络接口适配器(例如计算机系统(2000)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(2000)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(2000)可以使用这些网络中的任何一个网络与其他实体通信。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设备的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。

[0271] 上述人机接口设备、人机可访问的存储设备和网络接口可以附接到计算机系统(2000)的内核(2040)。

[0272] 内核(2040)可以包括一个或多个中央处理单元(CPU)(2041)、图形处理单元(GPU)(2042)、以现场可编程门区域(FPGA)形式的专用可编程处理单元(2043)、用于某些任务的硬件加速器(2044)、图形适配器(2050)等。这些设备以及只读存储器(ROM)(2045)、随机存取存储器(2046)、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(2047)可以通过系统总线(2048)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(2048),以能够通过附加的CPU、GPU等进行扩展。外围设备可以直接连接到内核的系统总线(2048)或通过外围总线(2049)连接到内核的系统总线(1848)。在一个示例中,屏幕(2010)可以连接到图形适配器(2050)。外围总线的体系结构包括PCI、USB等。

[0273] CPU(2041)、GPU(2042)、FPGA(2043)和加速器(2044)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(2045)或RAM(2046)中。过

渡数据也可以存储在RAM (2046) 中,而永久数据可以存储在例如内部大容量存储器 (2047) 中。可以通过使用高速缓冲来进行对任何存储设备的快速存储及检索,该高速缓冲可以与下述紧密关联:一个或多个CPU (2041)、GPU (2042)、大容量存储 (2047)、ROM (2045)、RAM (2046) 等。

[0274] 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。

[0275] 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(2000),特别是内核(2040)的计算机系统可以提供功能。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(2040)的存储器,例如内核内部大容量存储器(2047)或ROM(2045)。可以将实施本公开的各种实施例的软件存储在此类设备中并由内核(2040)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设备或芯片。软件可以使得内核(2040),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中的数据结构(2046)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(2044))中的逻辑中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包括两者。本公开包括硬件和软件的任何适当组合。

[0276] 尽管本公开已经描述了一些示例性实施例,但存在落入本公开范围内的改变、置换和各种替代等价物。因此,应当理解,本领域技术人员将能够设计出许多虽然未在本文中明确示出或描述,但体现了本公开的原理,因此落入本公开的精神和范围内的系统和方法。

[0277] 附录A:缩略语

[0278] JEM:联合探索模型

[0279] VVC:下一代视频编码

[0280] BMS:基准集

[0281] MV:运动矢量

[0282] HEVC:高效视频编码

[0283] SEI:补充增强信息

[0284] VUI:视频可用性信息

[0285] GOPs:图片群组

[0286] TUs:变换单元

[0287] PUs:预测单元

[0288] CTUs:编码树单元

[0289] CTBs:编码树块

[0290] PBs:预测块

[0291] HRD:假定参考解码器

- [0292] SNR:信噪比
- [0293] CPUs:中央处理单元
- [0294] GPUs:图形处理单元
- [0295] CRT:阴极射线管
- [0296] LCD:液晶显示器
- [0297] OLED:有机发光二极管
- [0298] CD:光盘
- [0299] DVD:数字视频光盘
- [0300] ROM:只读存储器
- [0301] RAM:随机存取存储器
- [0302] ASIC:专用集成电路
- [0303] PLD:可编程逻辑器件
- [0304] LAN:局域网
- [0305] GSM:全球移动通信系统
- [0306] LTE:长期演进
- [0307] CANBus:控制器局域网络总线
- [0308] USB:通用串行总线
- [0309] PCI:外围设备互连
- [0310] FPGA:现场可编程门区域
- [0311] SSD:固态驱动器
- [0312] IC:集成电路
- [0313] HDR:高动态范围
- [0314] SDR:标准动态范围
- [0315] JVET:联合视频勘探队
- [0316] MPM:最可能模式
- [0317] WAIP:广角帧内预测
- [0318] CU:编码单元
- [0319] PU:预测单元
- [0320] TU:变换单元
- [0321] CTU:编码树单元
- [0322] PDPC:位置相关预测组合
- [0323] ISP:帧内子块划分
- [0324] SPS:序列参数设置
- [0325] PPS:图片参数集
- [0326] APS:自适应参数集
- [0327] VPS:视频参数集
- [0328] DPS:解码参数集
- [0329] ALF:自适应环路滤波器
- [0330] SAO:采样自适应偏移CC-ALF:跨分量的自适应环路滤波器CDEF:约束定向增强滤

波器

- [0331] CCS0:跨分量样本偏移
- [0332] LS0:局部采样偏移
- [0333] LR:环路恢复滤波器
- [0334] AV1:开放媒体联盟视频1
- [0335] AV2:开放媒体联盟视频2
- [0336] MVD:运动矢量差
- [0337] CfL:从亮度预测色度
- [0338] SDT:半解耦树
- [0339] SDP:半解耦划分
- [0340] SST:半分离树
- [0341] SB:超级块
- [0342] IBC(或IntraBC):块内复制
- [0343] CDF:累积密度函数
- [0344] SCC:屏幕内容编码
- [0345] GBI:广义双预测
- [0346] BCW:CU级加权双向预测
- [0347] CIIP:组合帧内帧间预测
- [0348] POC:图片序列号
- [0349] RPS:参考图片集
- [0350] DPB:解码图片缓冲器
- [0351] MMVD:运动矢量差分合并模式

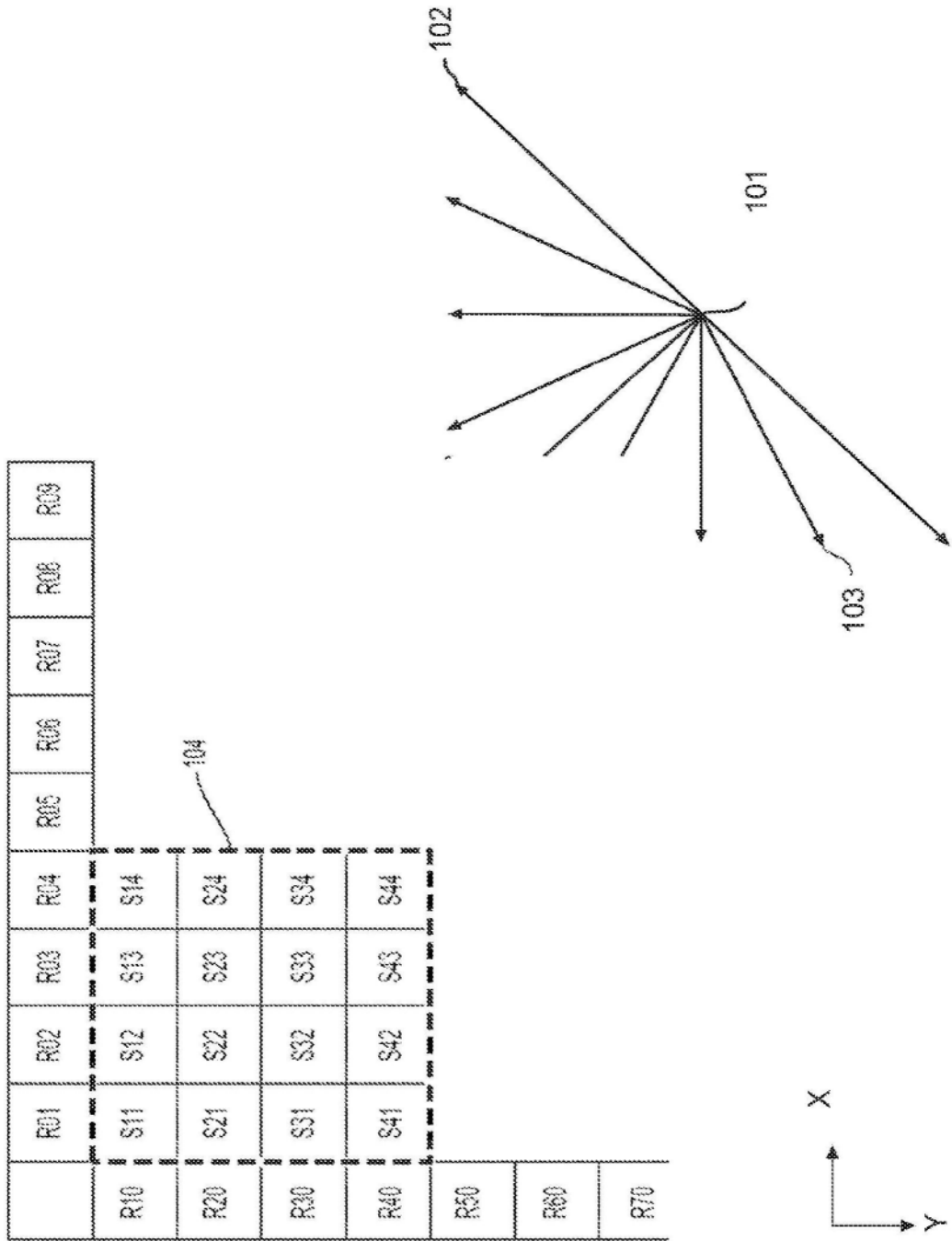


图1A

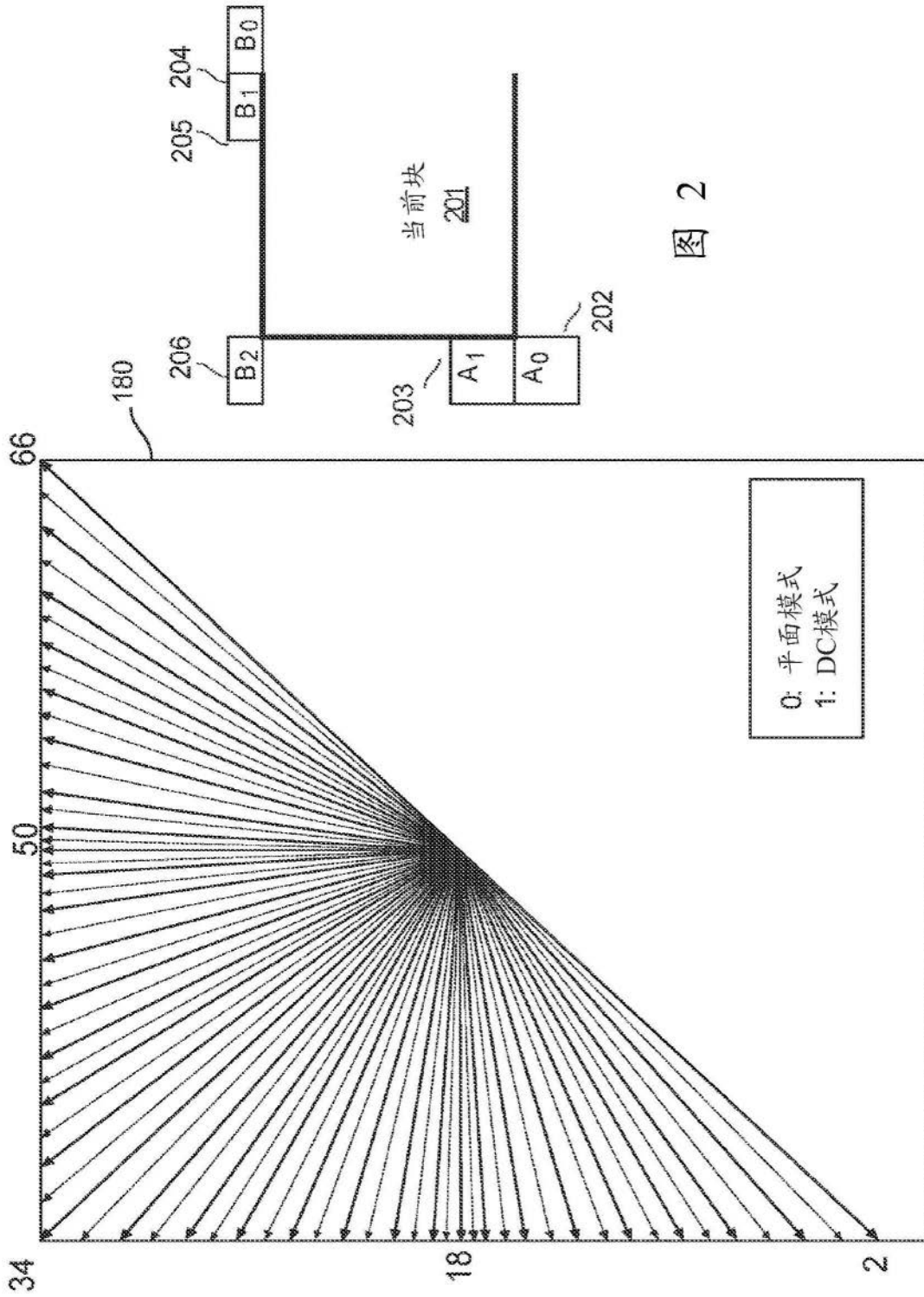


图 2

图 1B

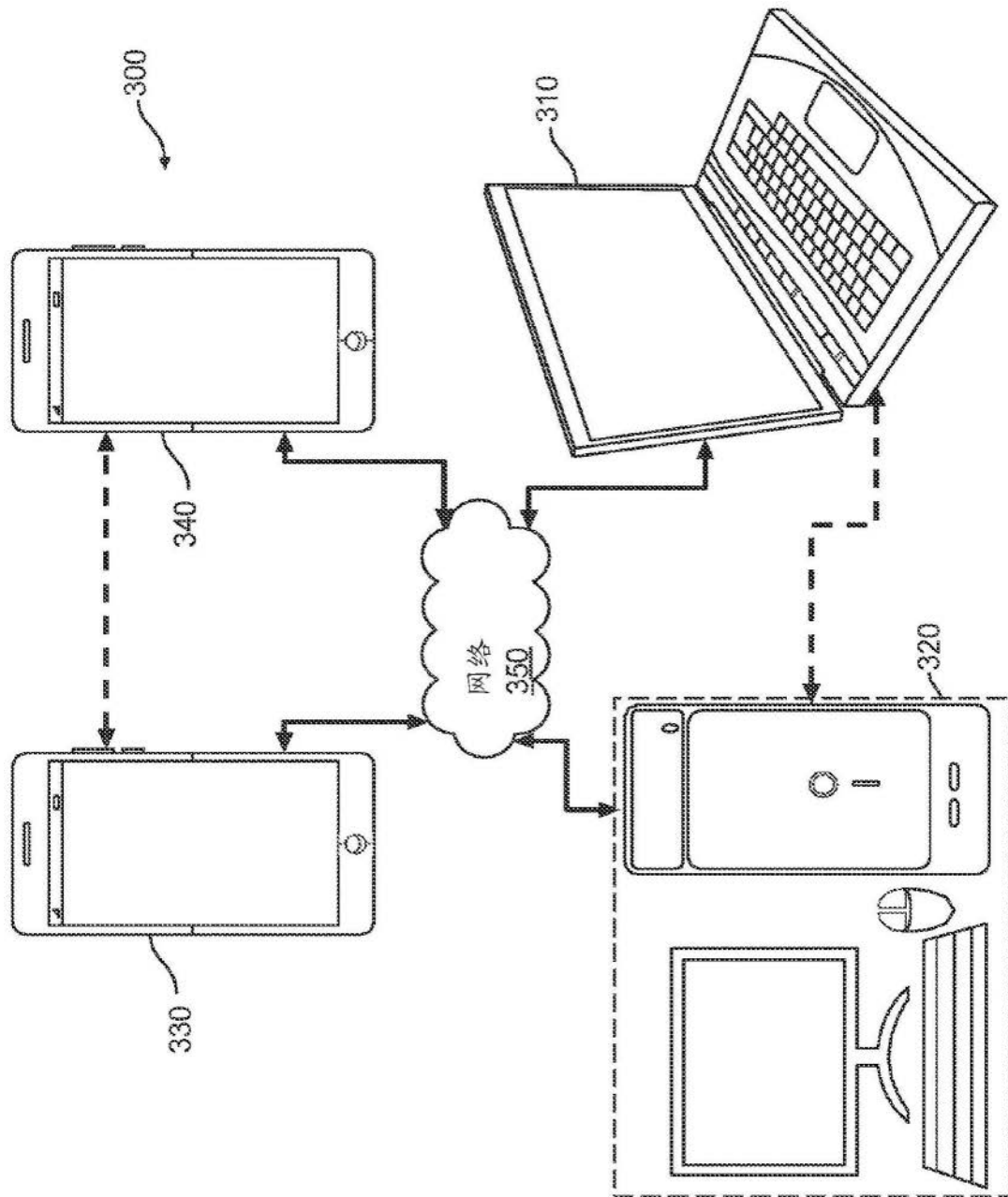


图3

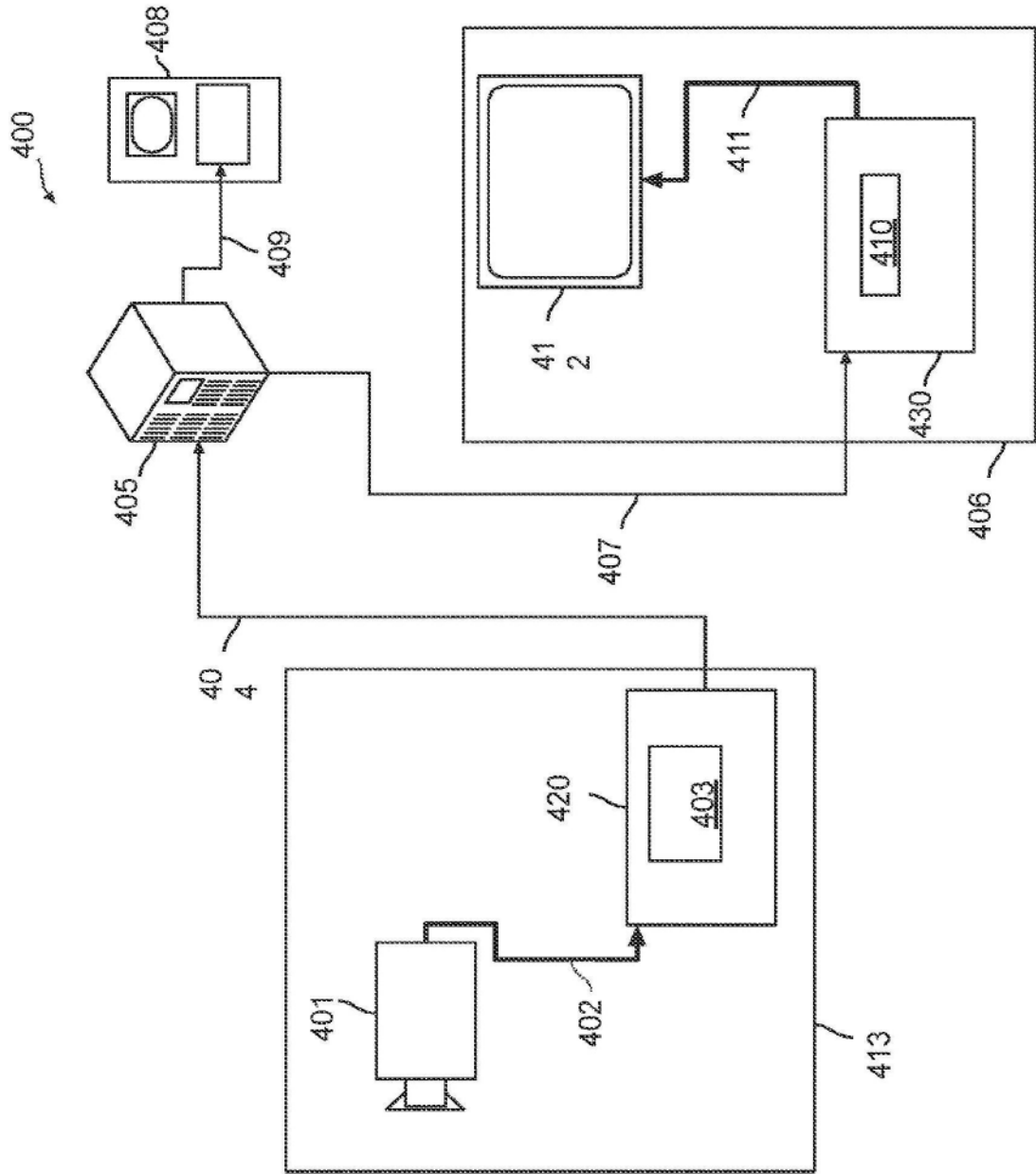


图4

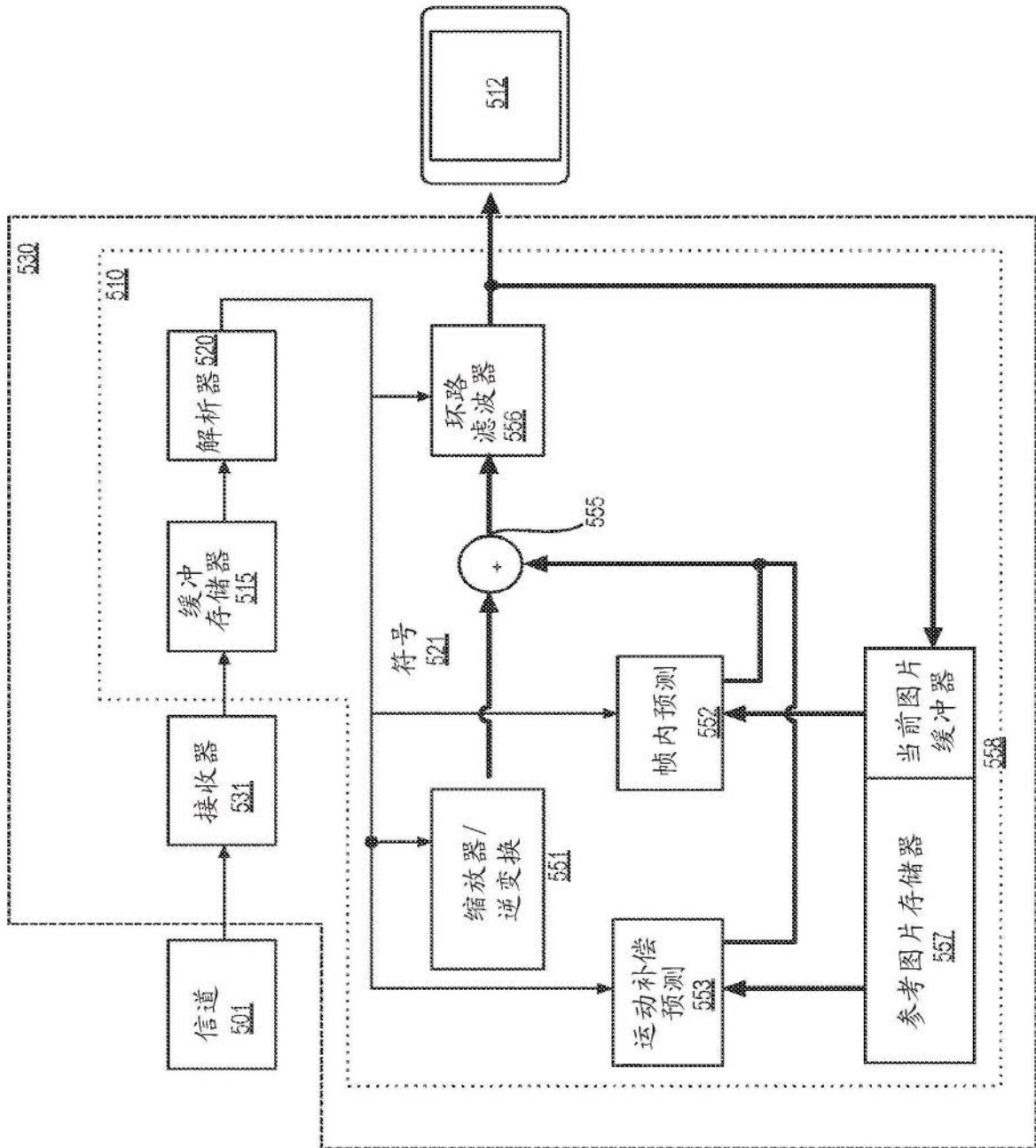


图5

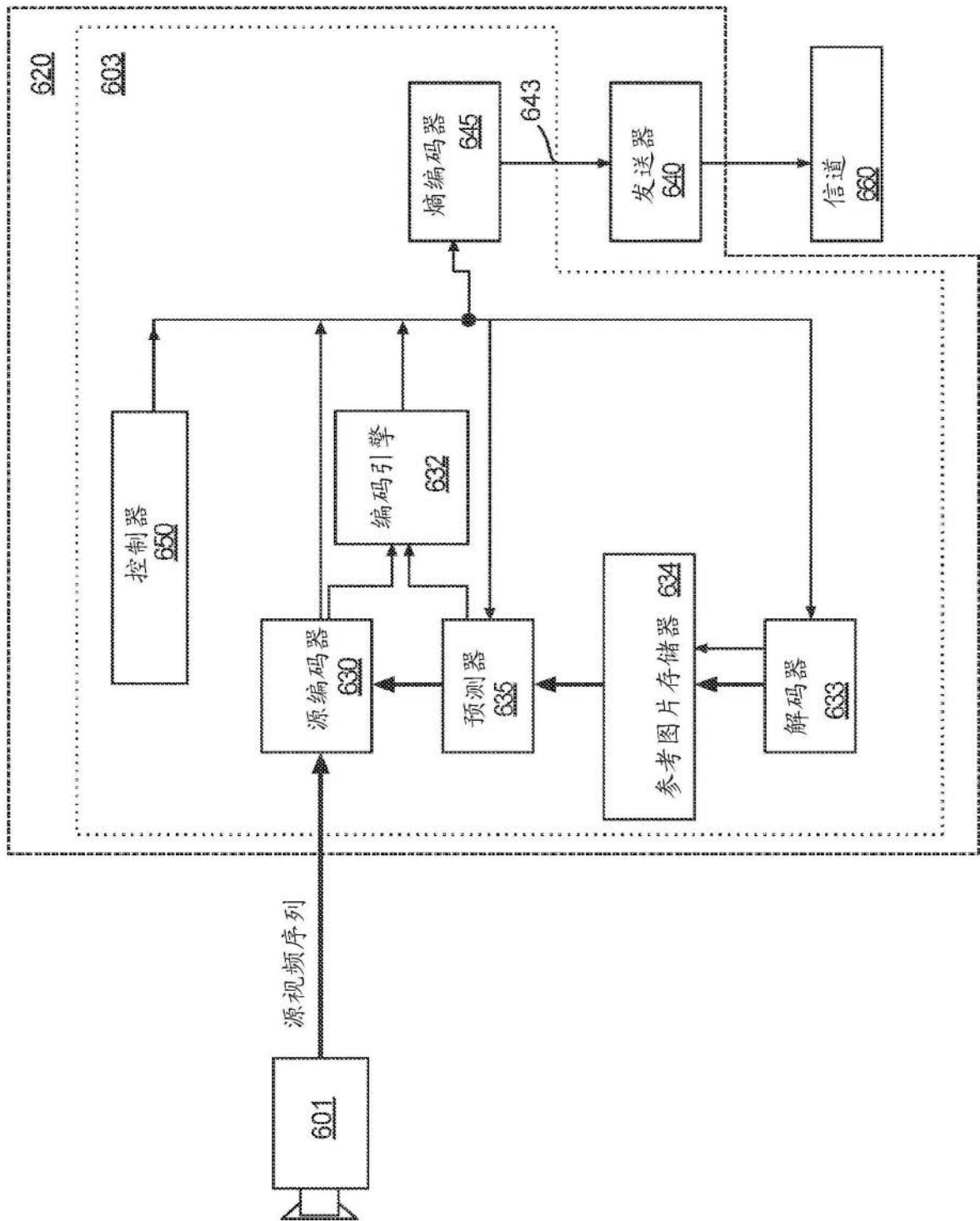


图6

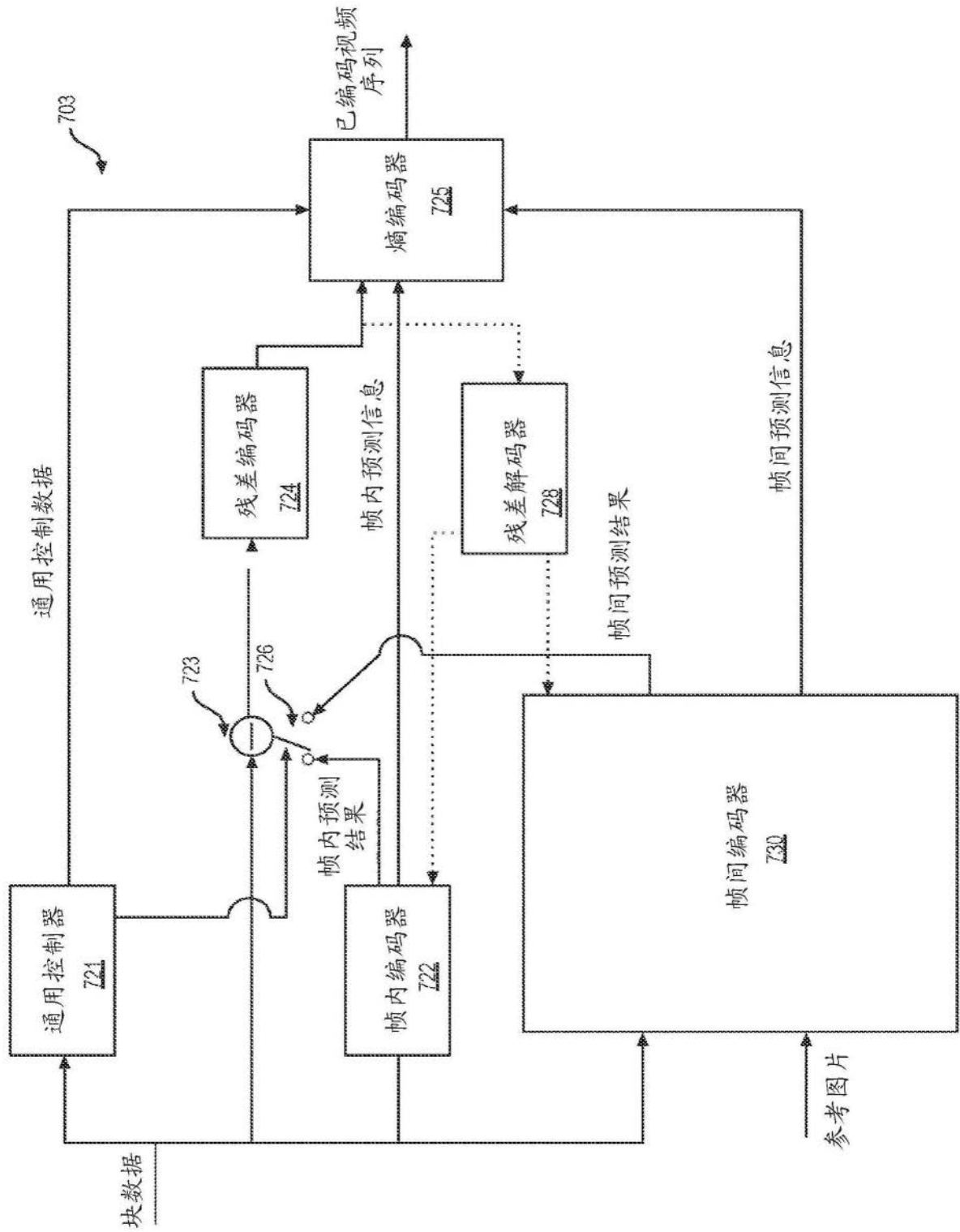


图7

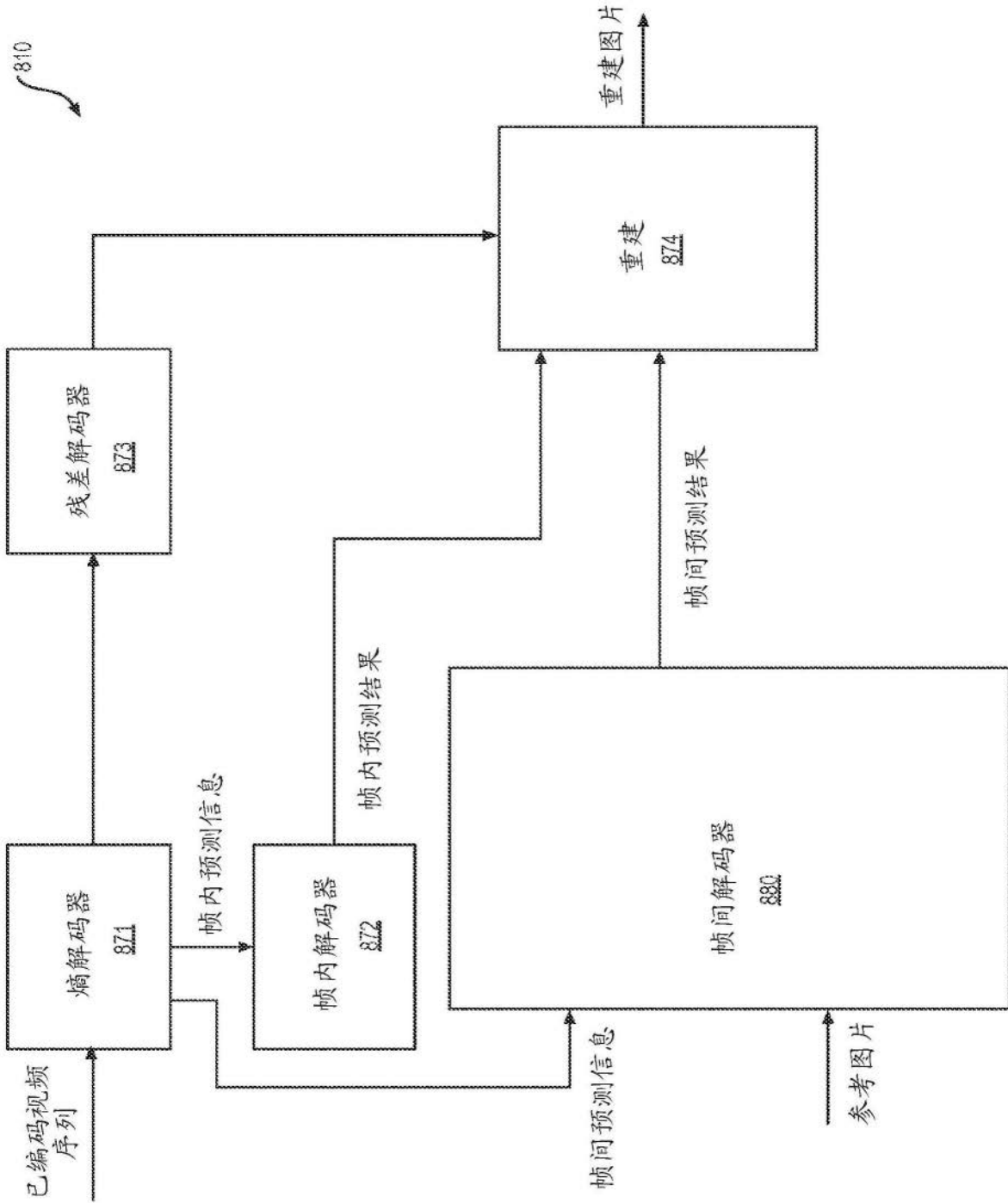


图8

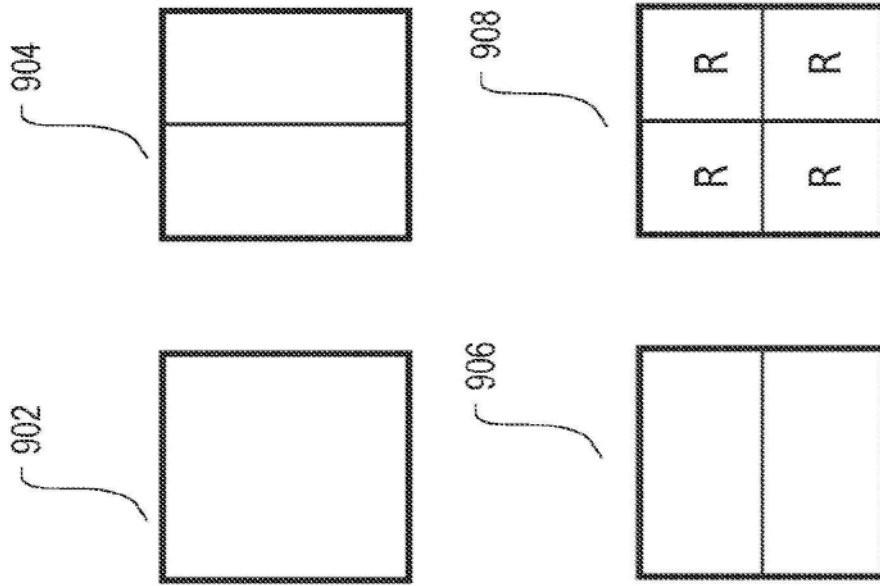


图9

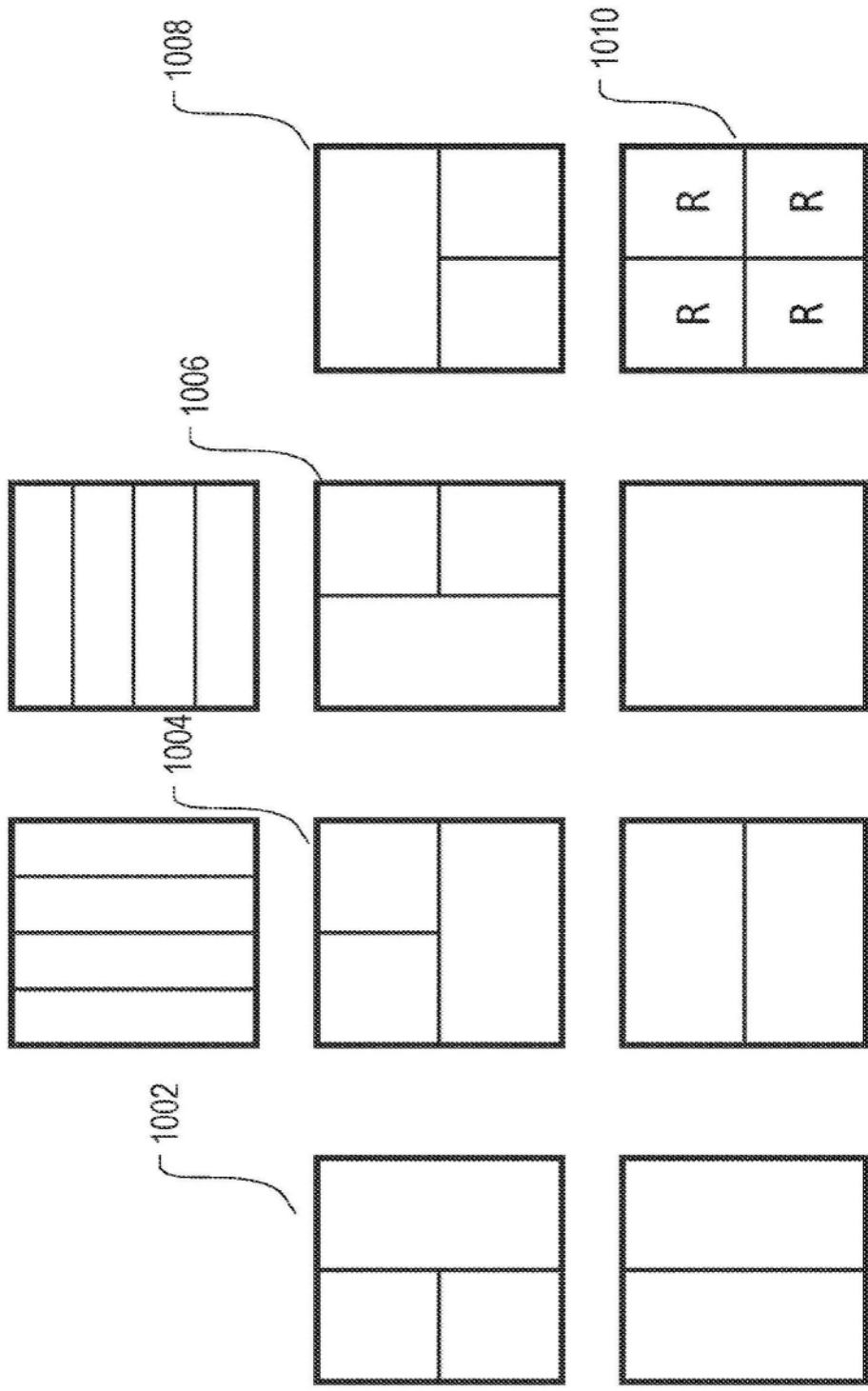


图10

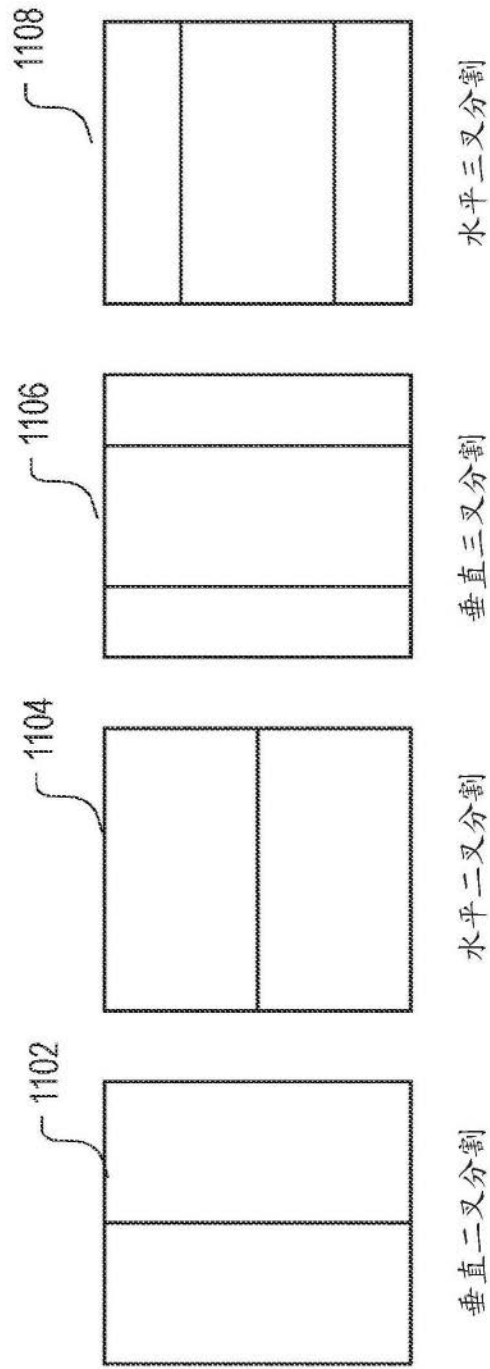


图11

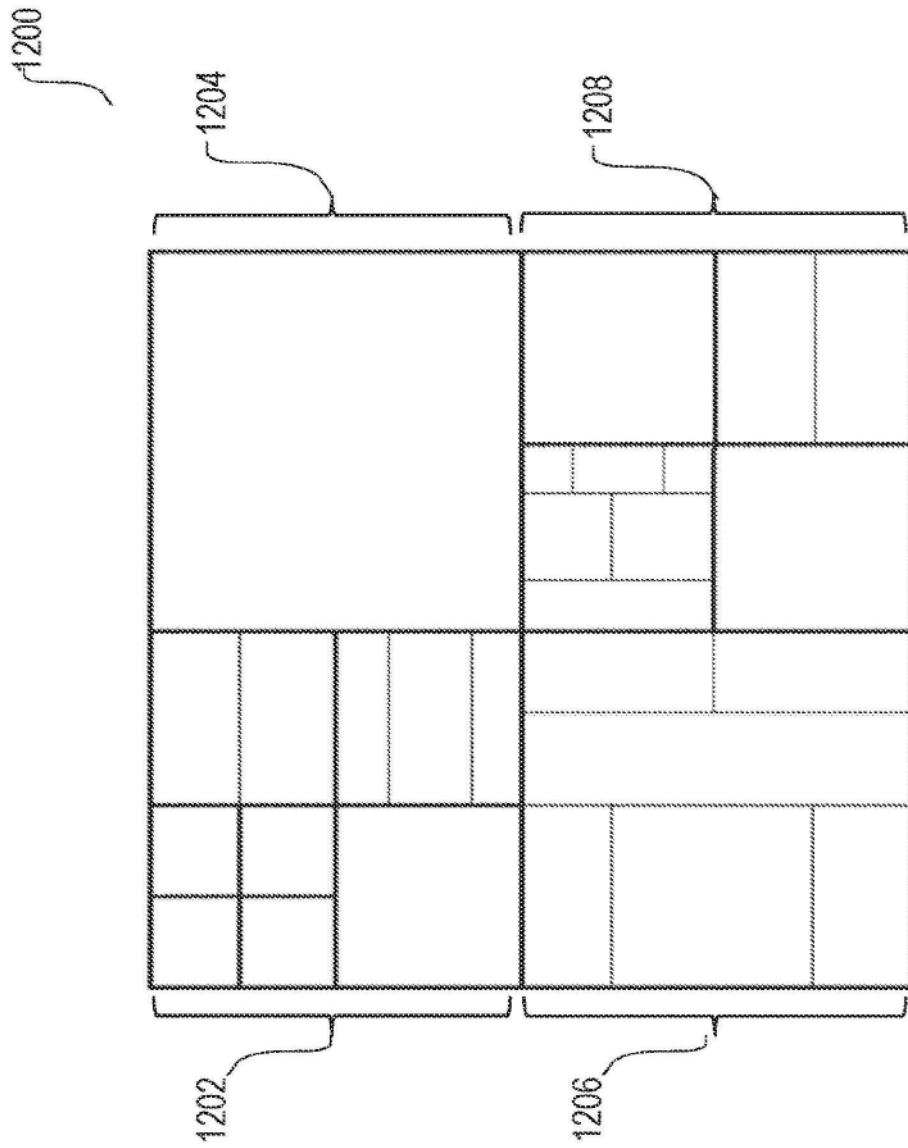


图12

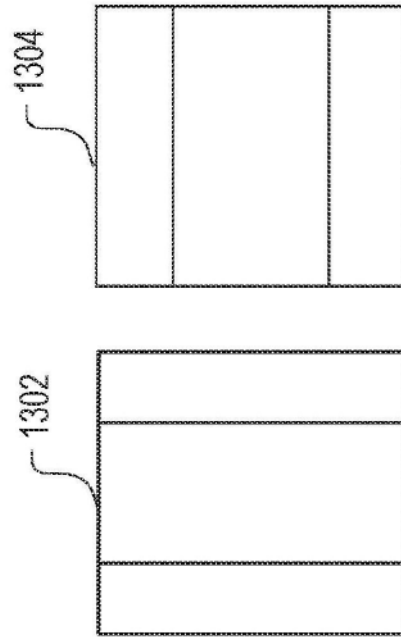


图13

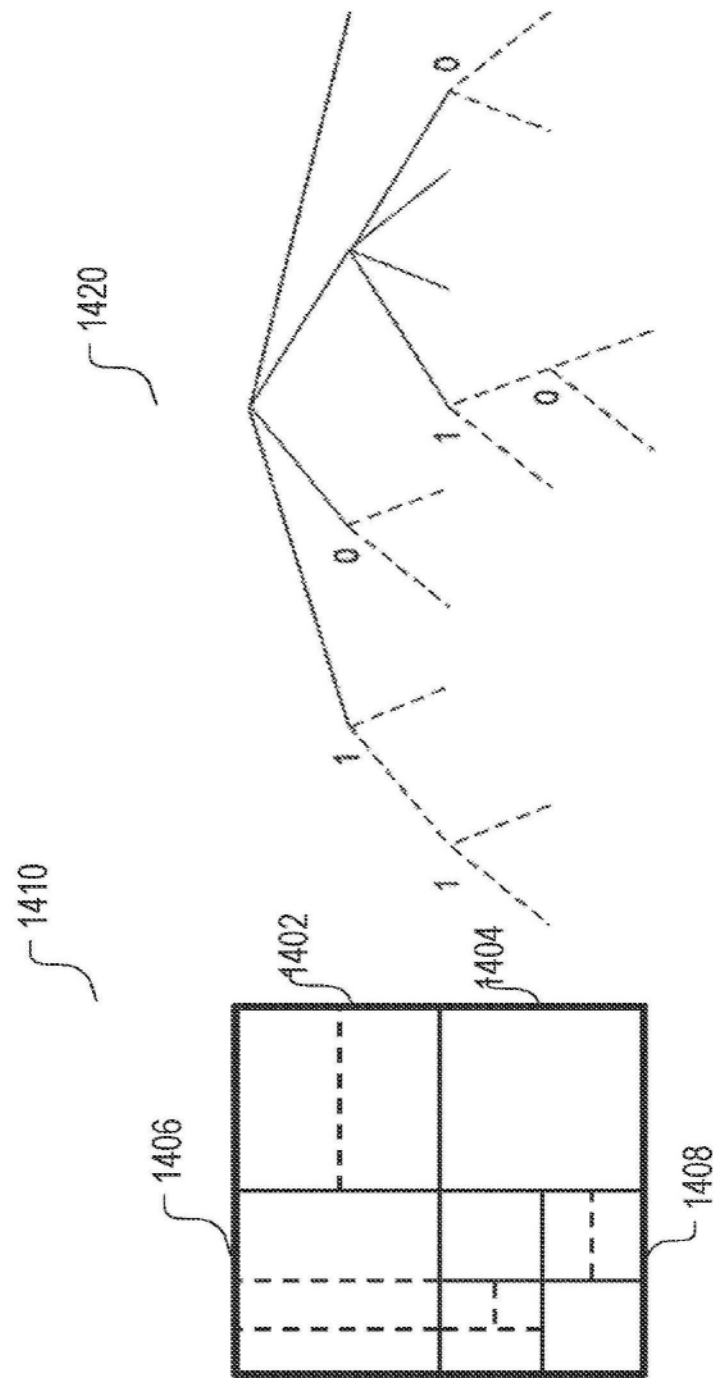


图14

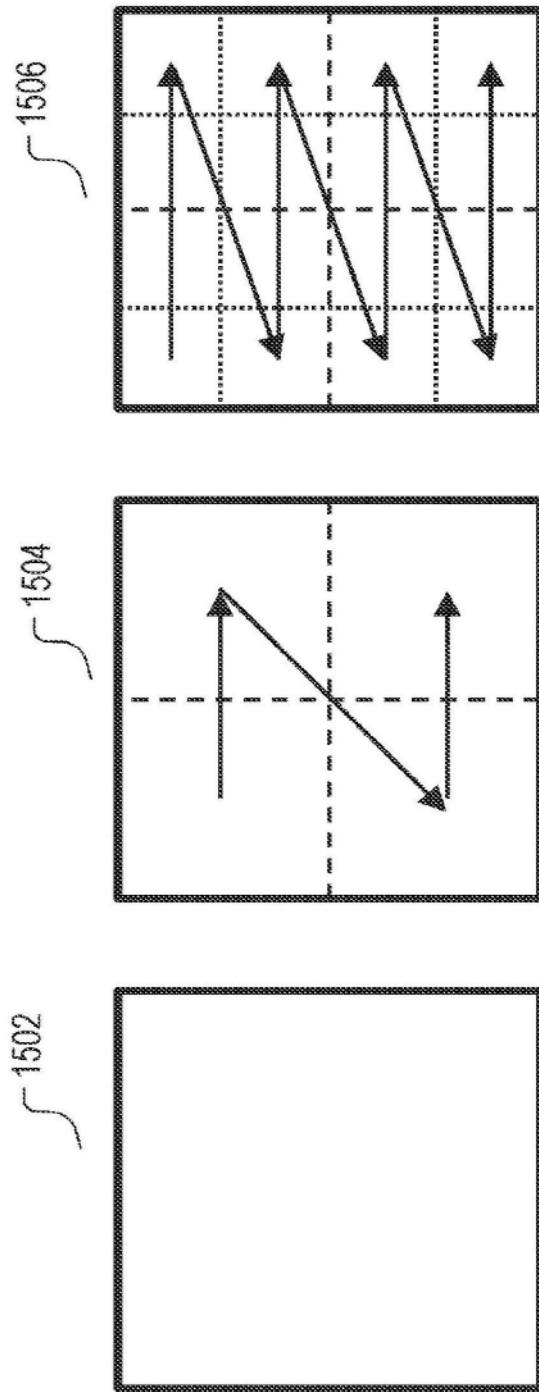


图15

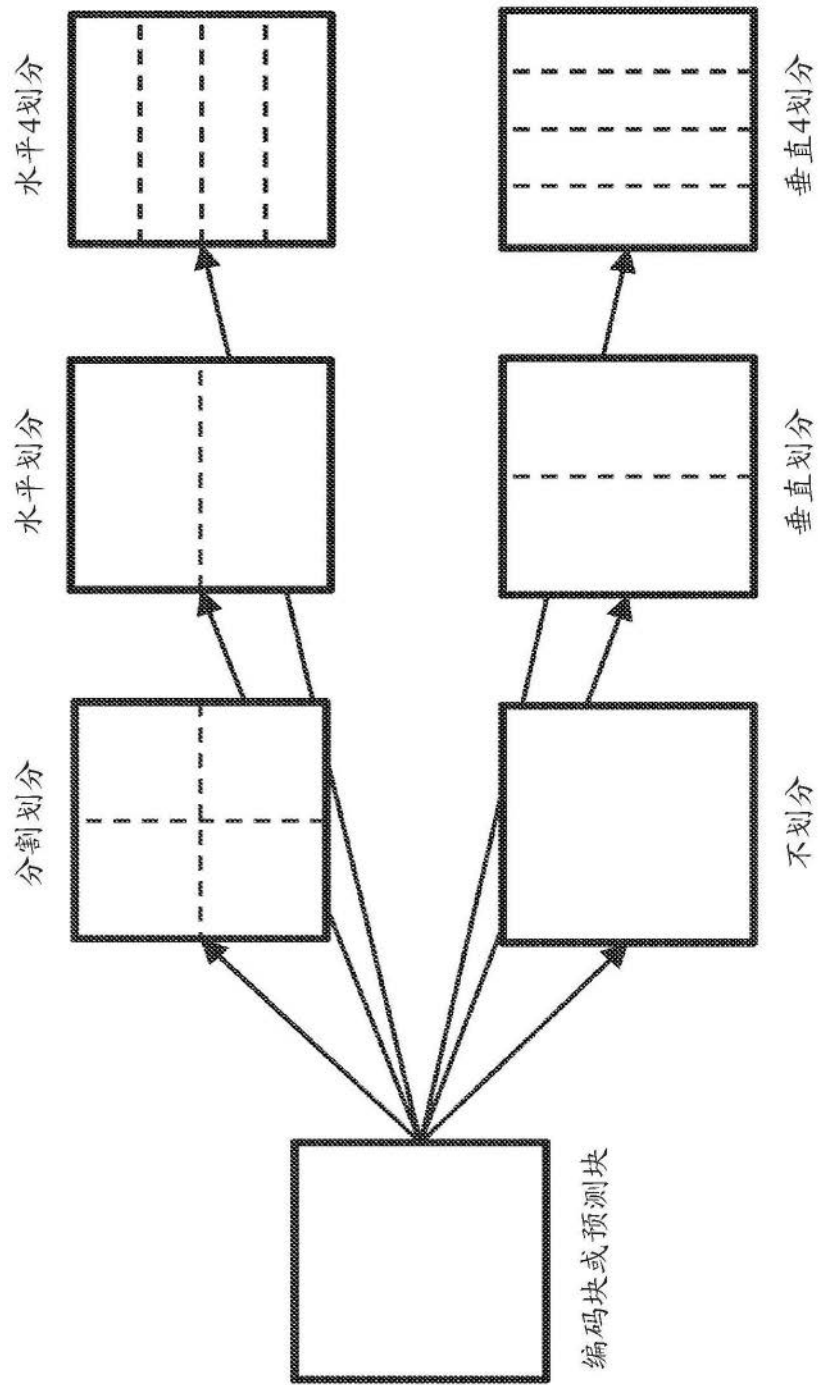


图17

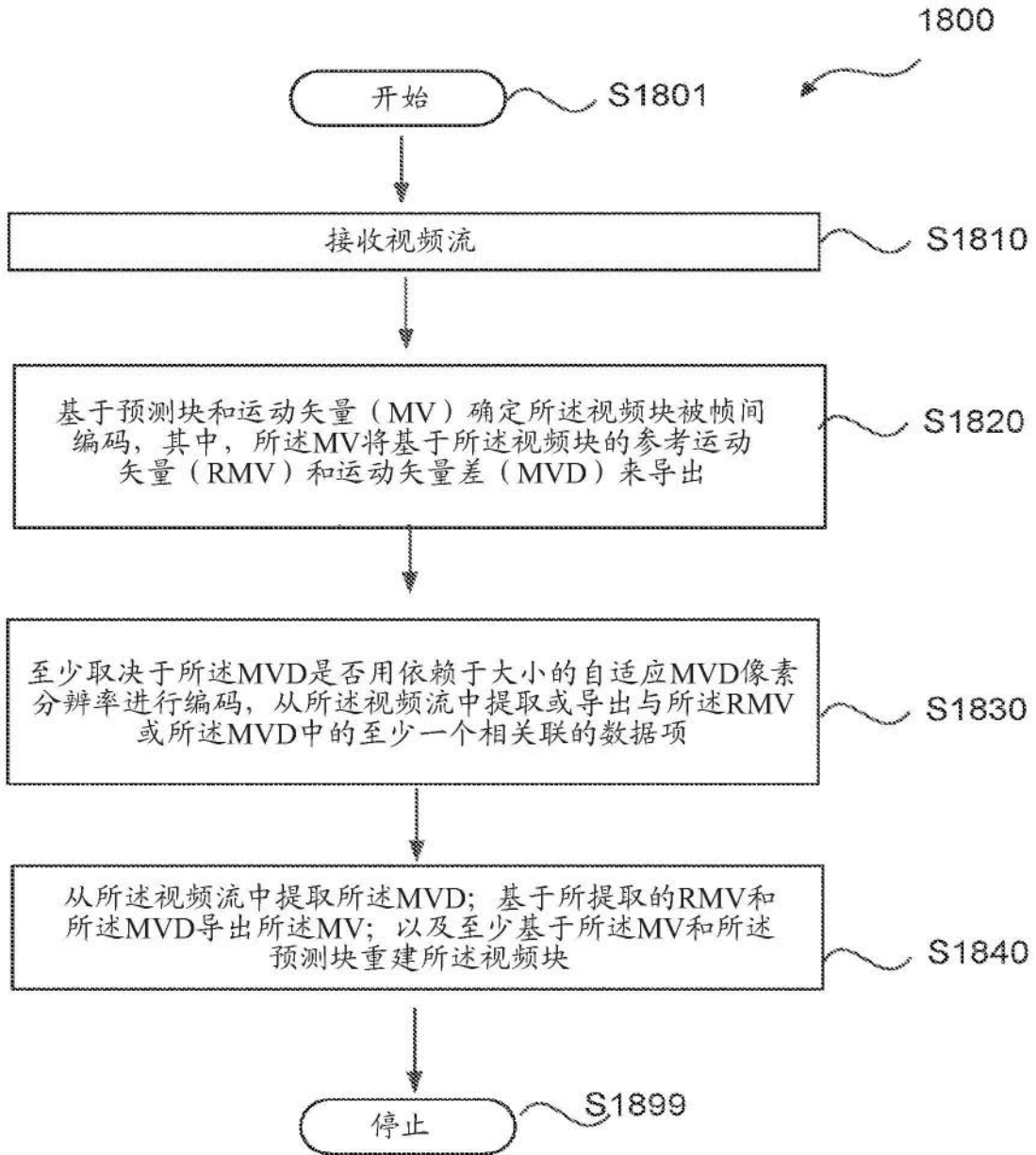


图18

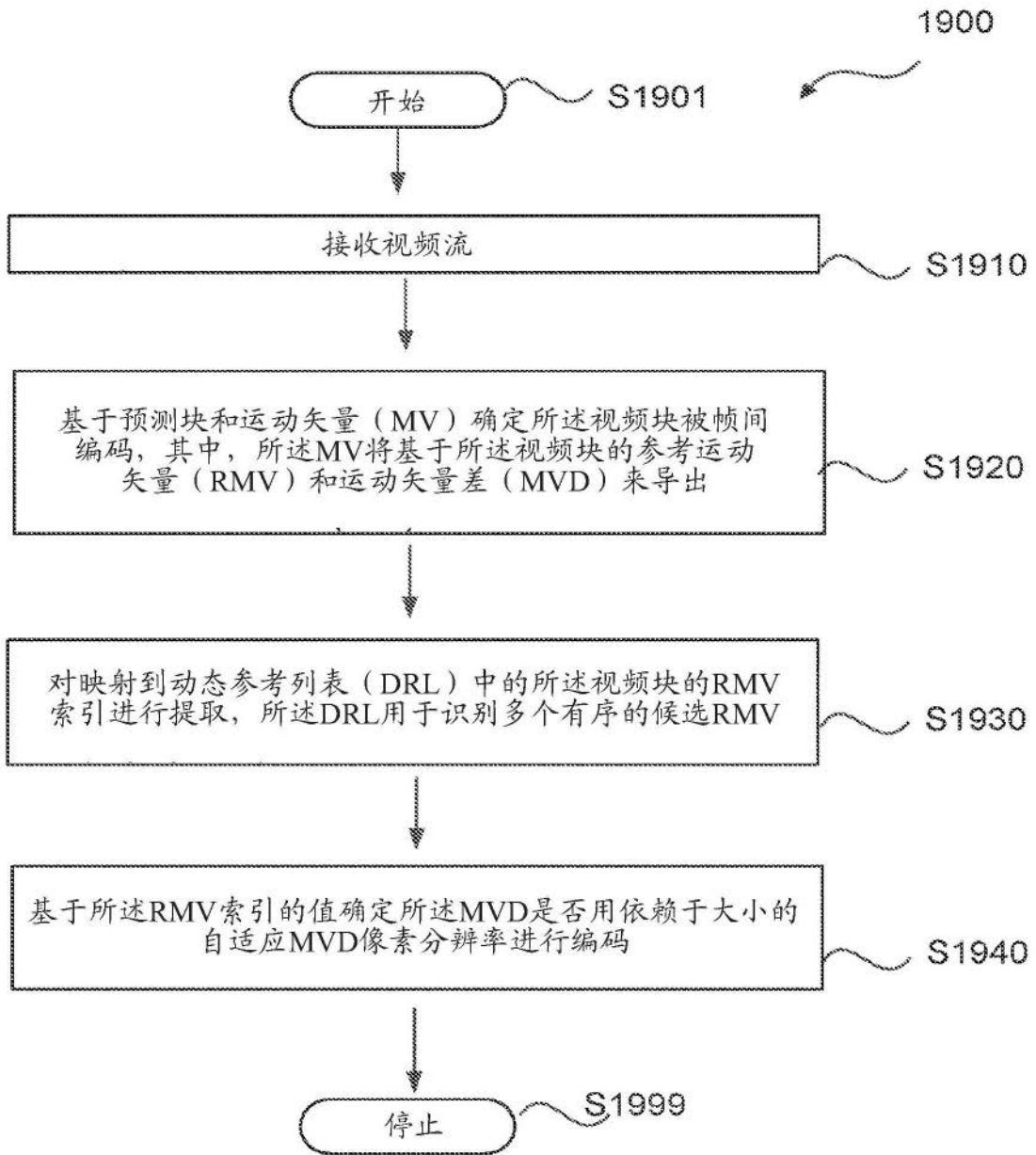


图19

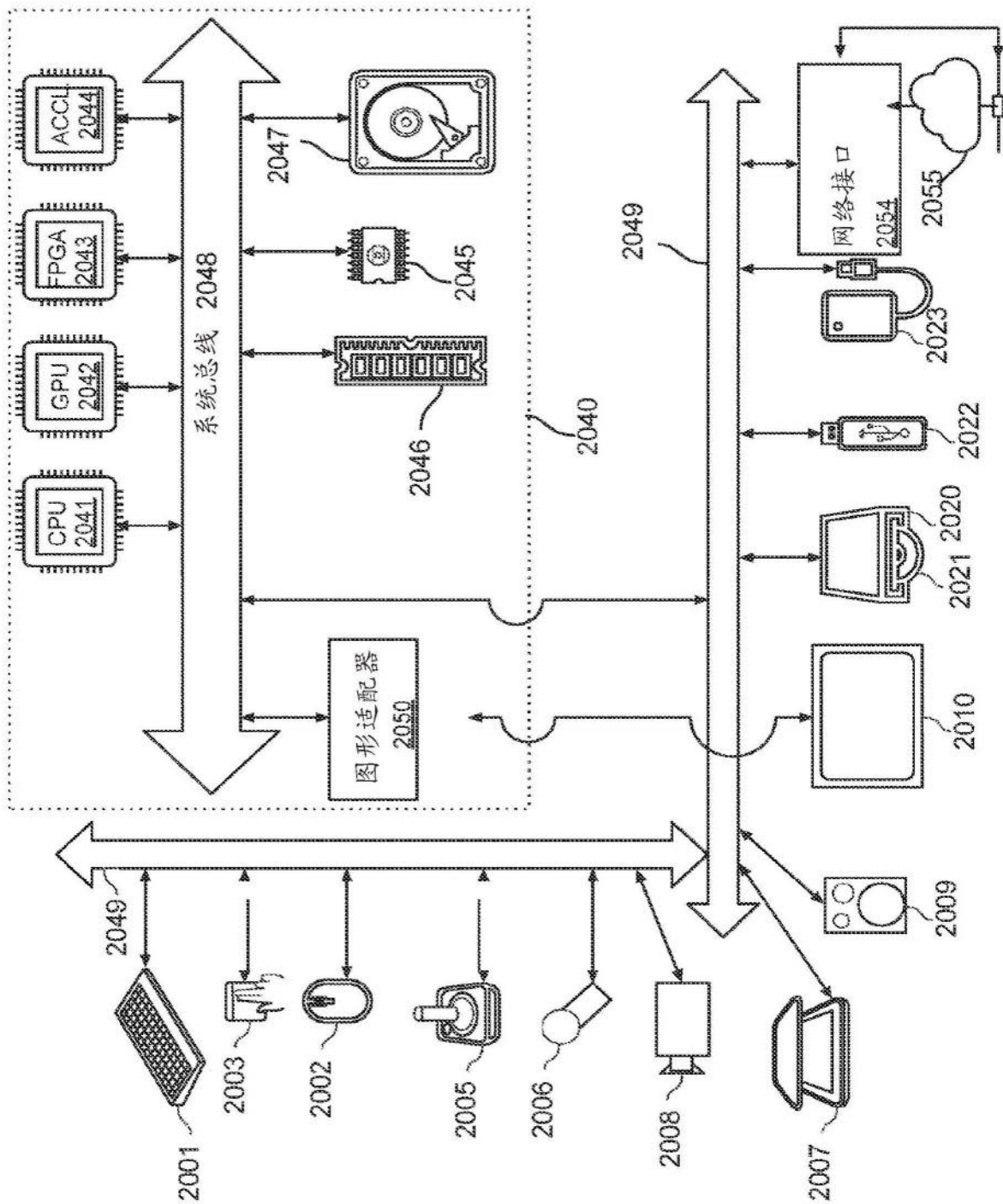


图20