



(10) 授权公告号 CN 116569547 B

(45) 授权公告日 2025. 04. 25

(21) 申请号 202280007734.5

(22) 申请日 2022.10.07

(65) 同一申请的已公布的文献号
申请公布号 CN 116569547 A

(43) 申请公布日 2023.08.08

(30) 优先权数据
63/257,856 2021.10.20 US
63/289,118 2021.12.13 US
17/960,206 2022.10.05 US

(85) PCT国际申请进入国家阶段日
2023.05.26

(86) PCT国际申请的申请数据
PCT/US2022/045994 2022.10.07

(87) PCT国际申请的公布数据
W02023/069257 EN 2023.04.27

(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
专利代理师 崔晓岚 张颖玲

(51) Int.Cl.
H04N 19/137 (2006.01)
H04N 19/513 (2006.01)
H04N 19/105 (2006.01)

(56) 对比文件
CN 116830581 A, 2023.09.29
审查员 师佳彬

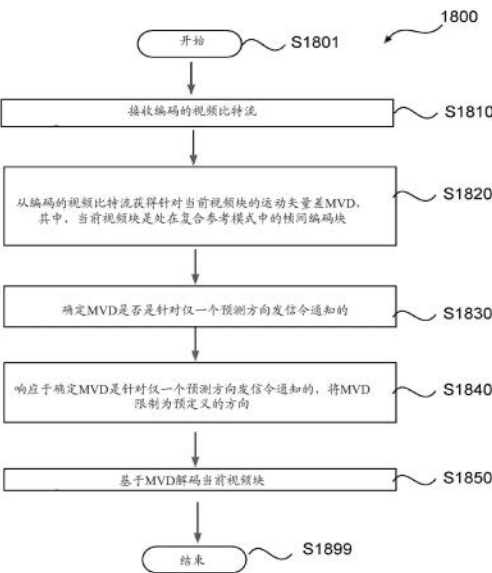
权利要求书2页 说明书37页 附图19页

(54) 发明名称

用于解码视频比特流的当前视频块的方法、装置及介质

(57) 摘要

本公开总体上涉及视频编码,尤其涉及用于提供用于限制运动矢量差(MVD)的信令方案的方法和系统。示例性的方法包括接收编码的视频比特流;从编码的视频比特流获得当前视频块的运动矢量差(MVD);确定MVD是否是针对仅一个预测方向发信令通知的;响应于确定MVD是针对仅一个预测方向发信令通知的,将MVD限制为预定义的方向;以及基于MVD解码当前视频块。



1. 一种用于解码视频比特流的当前视频块的方法,其特征在于,所述方法包括:

由设备接收编码的视频比特流;

由所述设备从所述编码的视频比特流获得当前视频块的运动矢量差MVD,其中,所述当前视频块是处在复合参考模式中的帧间编码块;在所述复合参考模式下,使用两个或多个参考块来通过加权平均生成预测块;

由所述设备确定所述MVD是否是针对仅一个预测方向发信令通知的;

响应于确定所述MVD是针对仅一个预测方向发信令通知的,由所述设备将所述MVD限制为预定义的方向,所述预定义的方向包括水平方向或垂直方向;所述响应于确定所述MVD是针对仅一个预测方向发信令通知的,由所述设备将所述MVD限制为预定义的方向,包括:在所述当前视频块处在NEW_NEARMV模式或NEAR_NEWMV模式时,将MVD限制到一个或多个确定的预定义方向;

由所述设备从所述编码的视频比特流获得指示水平方向和垂直方向是否具有非零MVD的标志;其中,当所述指示水平方向和垂直方向是否具有非零MVD的标志为第一标志时,所述第一标志的值为以下之一:指示所述水平方向和所述垂直方向均不具有非零MVD的第一标志值;指示所述水平方向和所述垂直方向中仅有一个方向具有非零MVD的第二标志值;或者,指示所述水平方向和所述垂直方向中仅有另一个方向具有非零MVD的第三标志值;

响应于所述当前视频块处在所述NEW_NEARMV模式中:响应于所述标志等于所述第二标志值,所述MVD对应于针对第一参考帧列表中的参考帧的垂直方向,或者,响应于所述标志等于所述第三标志值,所述MVD对应于针对所述第一参考帧列表中的所述参考帧的水平方向;

响应于所述当前视频块处在所述NEAR_NEWMV模式中:响应于所述标志等于所述第二标志值,所述MVD对应于针对第二参考帧列表中的参考帧的水平方向,或者,响应于所述标志等于所述第三标志值,所述MVD对应于针对所述第二参考帧列表中的所述参考帧的垂直方向;

由所述设备基于所述MVD解码所述当前视频块。

2. 根据权利要求1所述的方法,其特征在于,所述确定所述MVD是针对仅一个预测方向发信令通知的,包括:

所述MVD是在所述编码的视频比特流中针对在所述复合参考模式中的仅一个参考帧发信令通知的;或者

所述MVD是针对在所述复合参考模式中的两个参考帧联合发信令通知的。

3. 根据权利要求1所述的方法,其特征在于,当所述指示水平方向和垂直方向是否具有非零MVD的标志为第二标志时,所述第二标志的值为以下之一:

指示所述水平方向和所述垂直方向中仅有一个具有非零MVD的第四标志值;或者

指示所述水平方向和所述垂直方向中仅有另一个具有非零MVD的第五标志值。

4. 根据权利要求3所述的方法,其特征在于,所述方法还包括:

响应于所述当前视频块处在所述NEW_NEARMV模式中:

响应于所述标志等于所述第四标志值,所述MVD对应于针对第一参考帧列表中的参考帧的垂直方向,或者

响应于所述标志等于所述第五标志值,所述MVD对应于针对所述第一参考帧列表中的

所述参考帧的水平方向;以及

响应于所述当前视频块处在所述NEAR_NEWMV模式中:

响应于所述标志等于所述第四标志值,所述MVD对应于针对第二参考帧列表中的参考帧的水平方向,或者

响应于所述标志等于所述第五标志值,所述MVD对应于针对所述第二参考帧列表中的所述参考帧的垂直方向。

5. 根据权利要求1所述的方法,其特征在于,

所述标志是在高水平语法中发信令通知的,所述高水平语法包括以下至少一个:视频参数集VPS、图片参数集PPS、序列参数集SPS、自适应参数集APS、图片头部、帧头部、切片头部、瓦片头部或编码树单元CTU头部。

6. 根据权利要求1所述的方法,其特征在于,所述响应于确定所述MVD是针对仅一个预测方向发信令通知的,由所述设备将所述MVD限制为预定义的方向,包括:

响应于所述当前视频块处在NEW_NEARMV模式中或NEAR_NEWMV模式中,所述预定义的方向包括仅一个方向。

7. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

由所述设备基于针对所述当前视频块的模式确定水平方向和垂直方向是否具有非零MVD。

8. 根据权利要求7所述的方法,其特征在于,所述方法还包括:

响应于所述当前视频块处在NEW_NEARMV模式中,所述MVD对应于针对第一参考帧列表中的参考帧的水平方向,以及

响应于所述当前视频块处在NEAR_NEWMV模式中,所述MVD对应于针对第二参考帧列表中的参考帧的垂直方向。

9. 根据权利要求7所述的方法,其特征在于,所述方法还包括:

响应于所述当前视频块处在NEW_NEARMV模式中,所述MVD对应于针对第一参考帧列表中的参考帧的垂直方向,以及

响应于所述当前视频块处在NEAR_NEWMV模式中,所述MVD对应于针对第二参考帧列表中的参考帧的水平方向。

10. 一种用于解码视频比特流的当前视频块的装置,其特征在于,所述装置包括:

存储指令的存储器;以及

与所述存储器通信的处理器,其中,当所述处理器执行所述指令时,所述处理器被配置为使所述装置执行权利要求1至9中任一项所述的方法。

11. 一种存储指令的非暂态计算机可读存储介质,其特征在于,当所述指令由处理器执行时,所述指令被配置为使所述处理器执行权利要求1至9中任一项所述的方法。

12. 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9任一项所述的解码方法进行解码。

用于解码视频比特流的当前视频块的方法、装置及介质

[0001] 相关申请的交叉引用

[0002] 本申请基于并要求于2021年10月20日提交的第63/257856号美国临时专利申请的优先权的权益,其通过引用整体并入本文。本申请还基于并要求于2021年12月13日提交的第63/289118号美国临时专利申请的优先权的权益,其通过引用整体并入本文。本申请还基于并要求于2022年10月5日提交的第17/960206号美国非临时专利申请的优先权的权益,其通过引用整体并入本文。

技术领域

[0003] 本公开总体上涉及视频编码,尤其涉及用于提供用于限制运动矢量差(motion vector difference,MVD)的信令方案的方法和系统。

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。在背景技术部分以及本说明书的各个方面中所描述的目前已署名的发明人的工作所进行的程度,并不表明其在本申请提交时作为现有技术,且从未明示或暗示其被承认为本公开的现有技术。

[0005] 视频编码和解码可使用带有运动补偿的图片间预测来执行。未压缩的数字视频可包括一系列图片,每个图片具有例如为 1920×1080 的亮度样本及相关联的全色度样本或二次采样的色度样本的空间尺寸。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(或者称为帧率)。未压缩的视频具有用于流或数据处理的特定比特率要求。例如,像素分辨率为 1920×1080 、帧率为每秒60帧以及在每个颜色通道的每个像素为8位下色度二次采样为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输入未压缩视频信号中的冗余。压缩可有助于减小上述带宽和/或存储空间需求,在一些情况下可减小两个数量级或大于两个数量级。可采用无损压缩和有损压缩以及它们的组合。无损压缩指的是可以通过解码处理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同,但是原始信号和已重建的信号之间的失真足够小,以使已重建的信号虽然有一些信息丢失但可用于预期的应用。在视频的情况下,在许多应用中广泛采用有损压缩。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。可以选择或调整通过特定编码算法可实现的压缩比来反映各种失真容忍度:较高的可容忍失真通常允许产生较高损耗和较高压缩比的编码算法。

[0007] 视频编码器和解码器可利用来自多个宽泛类别和步骤的技术,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它数据的情况下表示样本值。在一些视频编解码器中,图片在空间上细分成样本块。当所有样本块都以帧内模式编码时,该图片可以被称为帧内图片。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以经受到频域的变换,并且可以在熵编码之前量化这样产生的变换系数。帧内预测表示一种使预变换域中的样本值最小化的技术。在一些情况下,变换之后的DC值越小,且AC系数越小,则在给定的量化步长下表示熵编码之后的块所需的位越少。

[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、按解码次序在帧内编码或解码的数据块之前的周围样本数据和/或元数据来进行尝试块的编码/解码的技术。在下文中这样的技术称为“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,而不使用来自其他参考图片的参考数据。

[0010] 帧内预测可以有許多不同形式。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,使用中的技术可以称为帧内预测模式。可以在特定的编解码器中提供一个或多个帧内预测模式。在一些情况下,模式可具有子模式和/或可以与各种参数关联,并且用于视频块的模式/子模式信息和帧内编码参数可以被单独地编码或集体地包括在模式码字中编码。给定的模式、子模式和/或参数组合使用哪个码字,可能会影响通过帧内预测的编码效率增益,因此可能会影响用于将码字转换成码流的熵编码技术。

[0011] H.264引入了某种帧内预测模式,该帧内预测模式在H.265中得到完善,且在诸如联合探索模型(JEM)、下一代视频编码(VVC)和基准集(BMS)的更新的编码技术中进一步得到完善。通常,对于帧内预测,可使用属于已经变成可用的相邻样本值来形成预测块。例如,可以将沿着特定方向和/或线的特定相邻样本组的可用值复制到预测器块中。对使用中的方向的参考可以在码流中进行编码,或者可以对其本身进行预测。

[0012] 参考图1A,在右下方描绘了在H.265的33种可能的帧内预测方向(对应于在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。箭头汇聚的点(101)表示正在被预测的样本。箭头表示用于从其预测在101处的样本的相邻样本的方向。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1A,在左上方描绘了 4×4 个样本的正方形块(104)(由粗体虚线指示)。正方形块(104)包括16个样本,每个样本用“S”、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是在Y维度上(从顶部开始)的第二个样本和在X维度上(从左侧开始)的第一个样本。类似地,样本S44是在Y和X维度上、块(104)中的第四个样本。由于块的尺寸为 4×4 个样本,因此S44位于右下角。还示出了遵循类似编号方案的示例性参考样本。参考样本用R、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,使用了与正在重建的块紧密相邻的预测样本。

[0014] 块104的帧内图片预测可通过根据用信号表示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,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样本来预测样本。在这种情

况下,从同一个参考样本R05预测样本S41、S32、S23和S14。然后从参考样本R08预测样本S44。

[0015] 在一些情况下,可例如通过插值来组合多个参考样本的值,以计算参考样本;尤其是当方向无法以45度均匀分开时。

[0016] 随着视频编码技术继续发展,可能的方向的数量增加。在H.264(2003年)中,例如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,以及在本公开时,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编码那些最合适的方向,对于方向,接受一定位的代价。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的在各种编码技术中的预测方向的数量增加。

[0018] 表示帧内预测方向的位向已编码视频码流中的预测方向映射的方式,可能因视频编码技术的不同而不同;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,可存在某些用于帧内预测的方向,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更可能的方向来说,将由更多的位数表示。

[0019] 图片间预测或帧间预测可以基于运动补偿。在运动补偿中,来自先前重建的图片或其部分(参考图片)的样本数据块在沿着由运动矢量(此后称为MV)指示的方向在空间上偏移之后,可以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(与时间维度类似)。

[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且按解码次序在当前MV之前的样本数据的其他区域相关的其它MV来预测适用于样本数据的某个区域的当前MV。这样做可大大减少通过依赖去除相关MV中的冗余对MV进行编码所需的数据量,从而提高压缩效率。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:比单个MV适用的区域更大的区域沿着在视频序列中相似的方向移动,因此在一些情况下,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。这使得被认为是用于给定区域的实际的MV与根据周围MV所预测的MV相似或相同。在熵编码之后,这样的MV又可以用比直接对MV进行编码而非从相邻的MV中预测时使用的位数更少的位数来表示。在其他情况下,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。在其它情况下,例如由于根据多个周围MV计算预测值时出现舍入误差,使得MV预测本身可以是有损的。

[0021] H.265/HEVC(ITU-T H.265建议书,“High Efficiency Video Coding(高效视频编码)”,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找到的样本,可根据已产生空间偏移的相同尺寸的先前块来预测该样本。可以从与一个或多个参考图片

相关联的元数据中导出MV,而非直接对该MV进行编码,例如使用与被标记为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相关联的MV,(按解码次序)从最近的参考图片中导出该MV。在H.265中,MV预测可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] 本公开描述了用于限制运动矢量差(MVD)的方法、装置和计算机可读存储介质的各种实施例。

[0024] 根据一方面,本公开的实施例提供了一种用于解码视频比特流的当前视频块的方法。该方法包括由设备接收编码的视频比特流。该设备包括存储指令的存储器和与存储器通信的处理器。该方法还包括由该设备从编码的视频比特流获得针对当前视频块的运动矢量差(MVD),其中,当前视频块是处在复合参考模式中的帧间编码块;该设备确定MVD是否是针对仅一个预测方向发信令通知的;响应于确定MVD是针对仅一个预测方向发信令通知的,该设备将MVD限制为预定义的方向;以及该设备基于MVD解码当前视频块。

[0025] 根据另一方面,本公开的实施例提供了一种用于解码视频比特流的当前视频块的装置。该装置包括存储指令的存储器和与存储器通信的处理器。当处理器执行指令时,处理器被配置为使装置执行用于视频解码和/或编码的上述方法。

[0026] 在另一方面,本公开的实施例提供了存储指令的非暂态计算机可读介质,当由计算机执行指令以进行视频解码和/或编码时,使计算机执行用于视频解码和/或编码的上述方法。

[0027] 在附图、说明书和权利要求中更详细地描述了上述的方面和其他方面及其实现方式。

附图说明

[0028] 根据下面的详细描述和附图,所公开的主体的其他特征、性质以及各种优点将更加明显,在附图中:

[0029] 图1A示出了帧内预测方向模式的示例性子集的示意性图示;

[0030] 图1B示出了示例性帧内预测方向的图示;

[0031] 图2示出了一个示例中的当前块和其周围的用于运动矢量预测的空间合并候选者的示意图;

[0032] 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;

[0033] 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;

[0034] 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;

[0035] 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;

[0036] 图7示出了根据另一个示例实施例的视频编码器的框图;

[0037] 图8示出了根据另一个示例实施例的视频解码器的框图;

[0038] 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;

[0039] 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;

[0040] 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;

- [0041] 图12示出了根据示例分区方案将基本块分区为编码块的示例；
- [0042] 图13示出了示例三进制分区方案；
- [0043] 图14示出了示例四叉树二叉树编码块分区方案；
- [0044] 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的方案和变换块的编码顺序；
- [0045] 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的另一方案和变换块的编码顺序；
- [0046] 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；
- [0047] 图18示出了根据本公开的示例实施例的方法的流程图；
- [0048] 图19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

[0049] 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声明的含义的细微含义。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,本文使用的短语“在一个实现方式中”或“在一些实现方式中”不一定指相同的实现方式,并且本文使用的短语“在另一个实现方式中”或“在其他实现方式中”不一定指不同的实现方式。例如,所要求保护的主题意在包括全部或部分示例性实施例/实现方式的组合。

[0050] 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,本文使用的诸如“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”如果用于关联列表,例如A、B或C,则意在表示A、B和C(本文用于包含的含义)以及A、B或C(本文用于排他的含义)。此外,本文使用的术语“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、结构或特点,或可用于以复数意义描述特征、结构或特点的组合。类似地,至少部分取决于上下文,诸如“一个(a)”、“一(an)”或“该(the)”的术语也可以被理解为表达单数用法或表达复数用法。此外,“基于”或“由...确定”的术语可被理解为不一定意在表达一组排他性因素,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,该多个终端设备可通过例如网络(350)彼此通信。例如,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)可以执行单向数据传输。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时频数据)进行编码,以通过网络(350)传输到另一终端设备(320)。已编码视频数据可以以一个或多个已编码视频码流的形式传输。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,以及根据所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。

[0051] 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(330)的视频数据)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,以及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。

[0052] 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。本公开的实施例可在台式计算机、膝上型计算机、平板电脑、媒体播放器、可穿戴计算机、专用视频会议设备上和/或类似物中实现。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/或其他类型信道中交换数据。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。

[0053] 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。

[0054] 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)可包括诸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。相较于已编码视频数据(404)(或已编码视频码流),被描绘为粗线以强调高数据量的视频图片流(402)可由电子设备(420)处理,电子设备(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的主题的各方面。相较于未压缩视频图片流(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)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(图中未示出)。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)上呈现的且未压缩的输出视频图片流(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,视频码流)进行编码。这些标准的示例包括ITU-T H.265建议书。在一个示例中,正在开发的视频编码标准非正式地称为下一代视频编码(VVC)。所公开的主题可用于VVC以及其他视频编码标准的上下文中。

[0055] 应注意,电子设备(420)和电子设备(430)可包括其它组件(未示出)。例如,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)。

[0056] 图5示出了根据下文本公开的任一个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)。

[0057] 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。每个视频序列可以与多个视频帧或图像关联。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通向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系统(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)可以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)可以实现为视频解码器(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)的外部或与视频解码器(510)分开(未描绘)。而在另一些其它应用中,在视频解码器(510)的外部可设置缓冲存储器(未描绘),其目的是例如防止网络抖动,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)以例如处理播放定时。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,可能不需要缓冲存储器(515),或可以将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,可能需要充足尺寸的缓冲存储器(515),其尺寸可以相对较大。这种缓冲存储器可以实现为具有自适应尺寸,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。

[0058] 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)之类的呈现设备的潜在信息,该显示器可以是也可以不是电子设备(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(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(GOP)、图片、图块、切片、宏块、编码单元(CU)、块、变换单元(TU)、预测单元(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。

[0059] 解析器(520)可以对从缓冲存储器(515)接收的视频序列执行熵解码/解析操作,从而创建符号(521)。

[0060] 取决于已编码视频图片或一部分已编码视频图片(例如:图片间和图片内,块间和块内)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。所涉及的单元以及它们如何涉及可由解析器(520)通过从已编码视频序列解析的子群控制信息来控制。为了简要起见,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。

[0061] 除了已提及的功能块之外,视频解码器(510)可以在概念上细分成如下文所描述的多个功能单元。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,下文公开中适用在概念上细分成多个功能单元。

[0062] 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、块尺寸、量化因子/参数、量化定标矩阵等信息。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。

[0063] 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:不使用来自先前重建的图像的预测信息,但是可使用来自当前图像的先前重建部分的预测信息的块。此类预测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)可以使用已经重建并存储在当前图片缓冲器(558)的周围块信息来生成尺寸和形状与正在重建的块相同的块。例如,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。

[0064] 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的块。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)以提取用于图片间预测的样本。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的输出可以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,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(移位)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,也可以与运动矢量预测机制关联等。

[0065] 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(还称为已编码视频码流)中且可作为来自解析器(520)的符号(521)而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对已编码图片或已编码视频序列的先前(按解码次序)部分进行解码期间获得的元信息,以及响应于先前重建且经过环路滤波的样本值。几种类型的环路滤波器可以以各种顺序作为环路滤波器单元556的一部分而被包括,将在下文进一步详细描述。

[0066] 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)中以用于将来的图片间预测。

[0067] 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图片间预测。例如,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)可变成参考图片存储器(557)的一部分,且可以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。

[0068] 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基础上,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。为了合规性,已编码视频序列的复杂度可以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,层级限制最大图片尺寸、最大帧率、最大重建采样率(以例如每秒兆(mega)个样本为单位进行测量)、最大参考图片尺寸等。在一些情况下,由层级设定的限制可通过假想参考解码器(HRD)规范和在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。

[0069] 在一些示例性实施例中,接收器(531)可以在接收已编码视频时接收附加(冗余)数据。附加数据可以被包括作为已编码视频序列的一部分。附加数据可由视频解码器(510)使用来对数据进行适当解码和/或更准确地重建原始视频数据。附加数据可采用例如时间、空间或信噪比(SNR)增强层、冗余切片、冗余图片、前向纠错码等形式。

[0070] 图6示出了根据本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)还可以包括传输器(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)。

[0071] 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)可以实现为电子设备(620)的一部分。

[0072] 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的位深(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。

[0073] 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并控制如下文所描述的其它功能单元。为了简要起见,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.....)、图片尺寸、图片群组(GOP)布局、最大运动矢量搜索范围等。控制器(650)可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)。

[0074] 在一些示例性实施例中,视频编码器(603)可以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)解码器的方式创建样本数据,即使嵌入式解码器633在没有熵编码的情况下处理由源编码器(630)编码的已编码视频流也可以创建(因为在所公开的主题中考虑的视频压缩技术中,符号与熵编码中已编码视频码流之

间的任何压缩可以是无损的)。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和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)还用于提高编码质量。

[0075] “本地”解码器(633)的操作可与诸如已在上文结合图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,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编码器的本地解码器(633)中实现。

[0076] 在这一点上可以进行的观察是,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,可以以相同或基本上相同的功能形式存在于对应的编码器中。因此,所公开的主题有时候侧重于解码器操作,其与编码器的解码部分关联。因为编码器技术与全面地描述的解码器技术互逆,可由此简化编码器技术的描述。在某些区域或方面,在下文提供对编码器更详细的描述。

[0077] 在操作期间,在一些示例的实现方式中,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”及其形容词形式的“残差”可以互换使用。

[0078] 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)的操作有利地可以是有损过程。当已编码视频数据可以在视频解码器(图6未示出)中被解码时,已重建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,且可使已重建参考图片存储在参考图片存储器(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)。

[0079] 预测器(635)可针对编码引擎(632)执行预测搜索。即,对于待编码的新图片,预测器(635)可以在参考图片存储器(634)中搜索可用作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,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。

[0080] 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。

[0081] 可以在熵编码器(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。

[0082] 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,通信信道(660)可以是通向可存储已编码视频数据的存储设备的硬件/软件链路。传输器(640)可以将来自视频编码器(603)的已编码视频数据与待传输的其它数据合并,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)。

[0083] 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)可以向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,通常可以将图片分配为以下任一种图片类型:

[0084] 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(“IDR”)图片。本领域的普通技术人员了解I图片的这些变体及其相应的应用和特征。

[0085] 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。

[0086] 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的元数据来用于重建单个块。

[0087] 源图片通常可以在空间上细分成多个样编码本块(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。例如,I图片的块可进行非预测性编码,或者I图片的块可参考同一图片的已编码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来进行预测性编码。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来进行预测性编码。出于其他目的,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,将在下面进一步详细描述。

[0088] 视频编码器(603)可根据诸如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性编码操作。已编码视频数据可相应地符合所使用的视频编码技术或标准指定的语法。

[0089] 在一些示例性实施例中,传输器(640)可以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、诸如冗余图片和切片的其它形式的冗余数据、SEI消息、VUI参数集片段等。

[0090] 所采集的视频可作为呈时间序列的多个源图片(视频图片)。图片内预测(通常简化为帧内预测)利用给定图片中的空间相关性,而图片间预测则利用图片之间的(时间或其它)相关性。例如,可将正在编码/解码的特定图片分成块,正在编码/解码的特定图片称为当前图片。在当前图片中的块在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,可通过称为运动矢量的矢量对当前图片中的块进行编码。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,运动矢量可具有识别参考图片的第三个维度。

[0091] 在一些示例性实施例中,双向预测技术可用于图片间预测。根据该双向预测技术,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

是过去或将来)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来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。

[0092] 此外,合并模式技术可用于图片间预测以改善编码效率。

[0093] 根据本公开的一些示例性实施例,诸如图片间预测和图片内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(CTU)以用于压缩,图片中的CTU可以具有相同尺寸,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(CTB),这三个编码树块是一个亮度CTB和两个色度CTB。可以将每个CTU递归地以四叉树划分为一个或多个编码单元(CU)。例如,可以将 64×64 像素的CTU划分为一个 64×64 像素的CU,或4个 32×32 像素的CU。一个或多个 32×32 块中的每一个可以被进一步划分为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,以在诸如帧间预测类型或帧内预测类型的各种预测类型中确定CU的预测类型。取决于时间和/或空间可预测性,可以将CU划分为一个或多个预测单元(PU)。通常,每个PU包括一个亮度预测块(PB)和两个色度PB。在一个实施例中,以预测块为单位执行编码(编码/解码)中的预测操作。可以以各种空间模式将CU划分为PU(或不同颜色通道的PB)。例如,一个亮度或色度PB可以包括诸如 8×8 像素、 16×16 像素、 8×16 像素、 16×8 像素等的样本的值(例如,亮度值)的矩阵。

[0094] 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)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。在一个示例中,视频编码器(703)可以用于代替图4的示例中的视频编码器(403)。

[0095] 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。当确定要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;以及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,合并模式可以是图片间预测的子模式,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。在一些其它示例性实施例中,可存在适用于主题块的运动矢量分量。相应地,视频编码器(703)可包括其它未在图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。

[0096] 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)。

[0097] 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,按显示顺序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,参考图片是使用嵌入在图6中的示例编码器620中的解码单元633(示出的为图7的残差解码

器728,将在下文进一步详细描述),基于已编码视频信息解码的已解码参考图片。

[0098] 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,并且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)。

[0099] 通用控制器(721)可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;以及当针对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。

[0100] 残差计算器(723)可被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)选择的用于块的预测结果之间的差(残差数据)。残差编码器(724)可被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)可被配置成将残差数据从空间域转换到频域,以生成变换系数。变换系数随后经受量化处理以获得量化变换系数。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)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。例如,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。适当地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。

[0101] 熵编码器(725)可被配置成将码流格式化以包括已编码块并执行熵编码。熵编码器(725)被配置成将各种信息包括在比特流中。例如,熵编码器(725)可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,可能不存在残差信息。

[0102] 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)配置成接收作为已编码视频序列的一部分的已编码图片,且对已编码图片进行解码以生成已重建图片。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)。

[0103] 在图8的示例中,视频解码器(810)包括如图8中的示例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)。

[0104] 熵解码器(871)可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)。残差信息可经受逆量化并提供给残差解码器 (873)。

[0105] 帧间解码器 (880) 可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。

[0106] 帧内解码器 (872) 可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。

[0107] 残差解码器 (873) 可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,以将残差从频域转换到空间域。残差解码器 (873) 还可能需某些控制信息(用以包括量化器参数(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)。

[0108] 重建模块 (874) 可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(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,已重建图片是已重建视频的一部分。应注意,可执行诸如去块操作等其它合适的操作来改善视觉质量。

[0109] 应注意,可使用任何合适的技术来实现视频编码器 (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现视频编码器 (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)。

[0110] 转到用于编码和解码的块分区,一般的分区可以从基本块开始,并可以遵循预定义的规则集、特定图案、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。在按照下面描述的任何示例分区过程或其他过程或其组合分开或分区基本块之后,可以获得最终的一组分区或编码块。这些分区中的每一个可以处于分区层次结构中的各种分区水平之一,并且可以具有各种形状。每个分区可以被称为编码块(coding block,CB)。对于下面进一步描述的各种示例分区实现方式,每个得到的CB可以是允许的尺寸和分区水平中的任何一个。这样的分区被称为编码块,因为它们可以形成单元,对于这些单元,可以做出一些基本的编码/解码决策,并且可以优化编码/解码参数、确定编码/解码参数和在已编码的视频比特流中发信令通知编码/解码参数。最终分区中的最高或最深水平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树结构可以称为编码块树(coding block tree,CBT)。

[0111] 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图案或结构可能相同,也可能不同。

[0112] 在一些实现方式中,用于亮度通道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,亮度通道和色度通道可以具有单独的编码树结构或图案。此外,亮度通道和色度通道是使用相同还是不同的编码分区树结构以及要使用的实际编码分区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分区树结构或编码分区树结构模式,而对于P切片或B切片,亮度通道和色度通道可以共享相同的编码分区树方案。当采用单独的编码分区树结构或模式时,亮度

通道可以被一个编码分区树结构分区为CB,色度信道可以被另一个编码分区树结构分区为色度CB。

[0113] 在一些示例实现方式中,预定义的分区分区图案可以应用于基本块。如图9所示,示例的4路分区树可以从第一预定义水平(例如,64×64块水平或其他尺寸,作为基本块尺寸)开始,且基本块可以分层地向下分区到预定义的最低水平(例如,4×4水平)。例如,基本块可以经受由902、904、906和908指示的四个预定义的分区分区选项或图案,指定为R的分区分区被允许用于递归分区分区,因为如图9所示的相同分区分区图案可以以较低的尺度重复,直至分区分区到最低水平(例如,4×4水平)。在一些实现方式中,额外的限制可以应用于图9的分区分区方案。在图9的实现方式中,可以允许矩形分区分区(例如,1:2/2:1矩形分区分区),但是不允许它们可以是递归的,而允许正方形分区分区是递归的。图9中的分区分区具有递归,如果需要,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从根节点或根块开始的划分深度。例如,根节点或根块(例如64×64块)的编码树深度可以被设置为0,并且在根块在根据图9被进一步划分一次之后,编码树深度增加1。对于上述方案,从64×64基本块到4×4的最小分区分区的最大或最深水平将是4(从水平0开始)。这种分区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(例如,对于每个分层水平的每个颜色通道,可以独立地确定预定义的图案中的分区分区图案或选项)独立进行分区分区。可选地,两个或更多个颜色通道可以共享图9中的相同分层图案树(例如,可以为每个分层水平的两个或更多个颜色通道选择预定义的图案中的相同分区分区图案或选项)。

[0114] 图10示出了允许递归分区分区形成分区分区树的另一个示例的预定义分区分区图案。如图10所示,可以预定义示例性的10路分区分区结构或图案。根块可以以预定义的水平开始(例如,从128×128水平或64×64水平的基本块开始)。图10的示例分区分区结构包括各种2:1/1:2和4:1/1:4矩形分区分区。在图10的第二行中由1002、1004、1006和1008指示的具有3个子分区分区的分区分区类型可以称为“T型”分区分区。“T型”分区分区1002、1004、1006和1008可以被称为左T型、上T型、右T型和下T型。在一些示例性实现方式中,图10的矩形分区分区中没有一个被允许进一步细分。可以进一步定义编码树深度以指示从根节点或根块开始的划分深度。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被进一步划分一次之后,编码树深度增加1。在一些实现方式中,只有1010中的全正方形分区分区可以被允许按照图10的图案递归分区分区到分区分区树的下一级。换句话说,对于T型图案1002、1004、1006和1008内的正方形分区分区,可以不允许递归分区分区。如果需要,按照图10的分区分区过程递归,生成最终一组编码块。这种方案可以应用于一个或多个颜色通道。在一些实现方式中,可以为低于8×8水平的分区分区的使用增加更多的灵活性。例如,在某些情况下可以使用2×2色度帧间预测。

[0115] 在用于编码块分区分区的一些其它示例实现方式中,四叉树结构可用于将基本块或中间块划分为四叉树分区分区。这种四叉树划分可以分层地和递归地应用于任何方形分区分区。是否进一步四叉树划分基本块或中间块或分区分区可针对基本块或中间块/分区分区的各种局部特征调整。可以进一步调整在图片边界处的四叉树分区分区。例如,可以在图片边界处执行隐式四叉树划分,使得块将持续进行四叉树划分,直到块的大小适合图片边界。

[0116] 在一些其他示例实现方式中,可以使用从基本块开始的分层二叉树分区分区。对于这样的方案,基本块或中间水平块可以被分区分区为两个分区分区。二叉树分区分区可以是水平的,也可以是垂直的。例如,水平二叉树分区分区可以将基本块或中间块划分为相等的右分区分区和左分区分区。同

样,垂直二叉树分区可以将基本块或中间块划分为相等的上分区和下分区。这种二叉树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做出是否应该继续二叉树分区方案的决定,并且如果该方案确实进一步继续,则决定应该使用水平二叉树分区还是垂直二叉树分区。在一些实现方式中,进一步的分区可以在预定义的最低分区尺寸处停止(在一个维度或两个维度中)。可选地,一旦达到来自基本块的预定义分区水平或深度,则可以停止进一步的分区。在一些实现方式中,分区的纵横比可以被约束。例如,分区的纵横比可以不小于1:4(或不大于4:1)。因此,具有4:1的垂直与水平纵横比的垂直条形分区可以仅进一步被垂直地二叉树分区为每个具有2:1的垂直与水平纵横比的上部分区和下部分区。

[0117] 在一些其他示例中,如图13所示,三叉树分区方案可以用于分区基本块或任何中间块。三叉树图案可以垂直实现,如图13的1302所示,或水平实现,如图13的1304所示。而图13中的示例划分比,不管是水平的还是垂直的,均示出为1:2:1,可以预定义其它比率。在一些实现方式中,可以预定义两个或更多个不同的比率。这种三叉树分区方案可以用于补充四叉树或二叉树分区结构,因为这种三叉树分区能够在一个连续的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进行划分,由此会将对象划分到不同的分区中。在一些实现方式中,示例的三叉树的分区的宽度和高度总是2的幂,以避免额外的变换。

[0118] 上述分区方案可以在不同的分区水平上以任何方式组合。作为一个示例,可以组合上述四叉树和二叉树分区方案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/分区可以是四叉树划分或二叉树划分,如果指定的话,服从一组预定义的条件。一个特定的示例如图14所示。在图14的示例中,基本块首先被四叉树划分成四个分区,如1402、1404、1406和1408所示。此后,所得到的分区中的每一个在下一水平或者被四叉树划分为四个进一步的分区(例如1408),或者被二叉树划分为两个进一步的分区(例如水平地或垂直地划分,例如1402或1406,两者都是对称的),或者是非划分的(例如1404)。对于方形分区,可以递归地允许二叉树或四叉树划分,如1410的整体示例性分区图案和1420中的相应树结构/表示所示,其中实线表示四叉树划分,虚线表示二叉树划分。可以为每个二叉树划分节点(非叶二叉树分区)使用标志来指示该二叉树划分是水平划分的还是垂直划分的。例如,如1420所示,与1410的划分结构一致,标志“0”可以表示水平二叉树划分,标志“1”可以表示垂直二叉树划分。对于四叉树划分分区,不需要指示划分类型,因为四叉树划分总是水平和垂直地划分块或分区,以产生尺寸相等的4个子块/子分区。在一些实现方式中,标志“1”可以表示水平二叉树划分,标志“0”可以表示垂直二叉树划分。

[0119] 在QTBT的一些示例实现方式中,四叉树和二叉树划分规则集可以由以下预定义的参数和与其关联的相应函数来表示:

[0120] -CTU尺寸:四叉树的根节点尺寸(基本块的尺寸)

[0121] -MinQTSIZE:允许的最小四叉树叶节点尺寸

[0122] -MaxBTSIZE:允许的最大二叉树根节点尺寸

[0123] -MaxBTDepth:允许的最大二叉树深度

[0124] -MinBTSIZE:允许的最小二叉树叶节点尺寸

[0125] 在QTBT分区结构的一些示例实现方式中,CTU尺寸可以被设置为具有两个对应的64×64色度样本块(当考虑并使用示例色度子采样时)的128×128亮度样本,MinQTSIZE可

以设置为 16×16 , MaxBTSIZE可以设置为 64×64 , MinBTSIZE (针对宽度和高度两者) 可以设置为 4×4 , 并且MaxBTDepth可以设置为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的尺寸 16×16 (即, MinQTSIZE) 至 128×128 (即CTU尺寸) 的尺寸。如果一个节点是 128×128 , 它将不会首先被二叉树划分, 因为它的尺寸超过了MaxBTSIZE (即, 64×64)。此外, 不超过MaxBTSIZE的节点可以用二叉树来分区。在图14的示例中, 基本块为 128×128 。根据预定义的规则集, 基本块只能进行四叉树划分。基本块的分区深度为0。得到的四个分区中的每一个都是 64×64 , 不超过MaxBTSIZE, 可以在水平1进一步进行四叉树或二叉树划分。这一过程仍会继续。当二叉树深度达到MaxBTDepth (即4) 时, 可以不考虑进一步划分。当二叉树节点的宽度等于MinBTSIZE (即4) 时, 可以不考虑进一步水平划分。类似地, 当二叉树节点的高度等于MinBTSIZE, 不考虑进一步垂直划分。

[0126] 在一些示例实现方式中, 上述QTBT方案可以被配置为支持亮度和色度具有相同QTBT结构或不同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可以共享相同的QTBT结构。然而, 对于I切片, 亮度CTB可以通过一种QTBT结构分区为亮度CB, 色度CTB可以通过另一种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, 而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。

[0127] 在一些其他实现方式中, QTBT方案可以采用上述三叉树方案来补充。这种实现方式可以被称为多类型树 (multi-type-tree, MTT) 结构。例如, 除了节点的二叉树划分之外, 还可以选择图13中的三叉树分区图案之一。在一些实现方式中, 只有方形节点可以经受三叉树划分。可以使用附加的标志来指示三叉树分区是水平的还是垂直的。

[0128] 例如QTBT实现方式和由三叉树划分补充的QTBT实现方式的双水平或多水平树的设计可能主要是通过复杂性降低来驱动。理论上, 遍历树的复杂度是 T^D , 其中T表示划分类型的数量, D是树的深度。可以在减小深度 (D) 的同时, 使用多种类型 (T) 来进行折衷。

[0129] 在一些实现方式中, 可以对CB进行进一步分区。例如, 为了在编码和解码过程期间进行帧内或帧间预测, 可将CB进一步分区为多个预测块 (PB)。换句话说, 可以将CB进一步划分为不同的子分区, 其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进一步分区为多个变换块 (TB)。将CB分区为PB和TB的方案可以是相同的, 也可以是不同的。例如, 每种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程序来执行。在一些示例实现方式中, PB和TB分区方案可以是独立的。在一些其他示例实现方式中, PB和TB分区方案和边界可以是相关的。在一些实现方式中, 例如, 可以在PB分区之后对TB进行分区, 并且具体地, 在确定编码块后续的分区之后, 每个PB然后可以被进一步分区为一个或多个TB。例如, 在一些实现方式中, PB可以被划分为一个、两个、四个或其他数量的TB。

[0130] 在一些实现方式中, 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, 可以不同地对待亮度通道和色度通道。例如, 在一些实现方式中, 对于亮度通道, 可以允许将编码块分区为预测块和/或变换块, 而对于色度通道, 可以不允许将编码块分区为预测块和/或变换块。在这样的实现方式中, 可以仅在编码块水平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尺寸可以不同, 例如, 可以允许亮度通道的编码块被分区为的变换块和/或预测块的尺寸比色度通道的编码块被分区为的变

换块和/或预测块的尺寸更小。对于又一示例,将编码块划分为变换块和/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,可以允许亮度通道的编码块被分区的变换块和/或预测块的深度比色度通道的编码块被分区的变换块和/或预测块的深度更深。对于特定示例,亮度编码块可以被分区为多个尺寸的变换块,这些变换块可以由向下最多低至2水平的递归分区来表示,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 4×4 到 64×64 的变换块尺寸。然而,对于色度块,可以仅允许为亮度块指定的最大可能的变换块。

[0131] 在用于将编码块分区为PB的一些示例实现方式中,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。

[0132] 将编码块(或预测块)分区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划分和预定义图案的划分,并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划分水平,可以不具有相同的尺寸,并且可以不需要在形状上是正方形的(例如,可以是具有一些允许的尺寸和纵横比的矩形)。下面结合图15、图16和图17更详细地描述进一步的示例。

[0133] 然而,在一些其它实现方式中,经由上述任何分区方案获得的CB可作用于预测和/或变换的基本编码块或最小编码块。换句话说,针对执行帧间预测/帧内预测的目的和/或变换的目的,不执行进一步的划分。例如,从上述QTBT方案获得的CB可以直接用作执行预测的单位。具体地,这样的QTBT结构消除了多种分区类型的概念,也即,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可以具有正方形或矩形形状。这种QTBT的叶节点用作预测和变换处理的单位,而无需任何进一步的分区。这意味着在这种示例的QTBT编码块结构中,CU、PU和TU具有相同的块尺寸。

[0134] 可以以任何方式组合上述各种CB分区方案和从CB到PB和/或TB的进一步分区(不包括PB/TB分区)。以下提供的特定实现方式是非限制性的示例。

[0135] 下面描述编码块和变换块分区的具体示例性实现方式。在这样的示例性实现方式中,可以使用递归的四叉树划分或上述预定义的划分图案(例如图9和图10中的那些)将基本块划分为编码块。在每个水平,可以由本地视频数据特性来确定是否应该继续对特定分区进行进一步的四叉树划分。所得到的CB可以处于不同的四叉树划分水平,并具有不同的尺寸。可以在CB水平(或CU水平,用于所有三种颜色信道)做出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策。每个CB可以根据预定义的PB划分类型进一步划分为一个、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处理,并且可以基于PB将相关信息发送到解码器。在应用基于PB划分类型的预测过程获得残差块之后,可以根据类似于CB的编码树的另一种四叉树结构将CB划分为TB。在这种特定实现方式中,CB或TB可以但不必限于方形。此外,在这种特定示例中,对于帧间预测,PB可以是正方形或矩形,对于帧内预测,PB可以仅是正方形。编码块可以被划分为例如四个方形TB。每个TB可以被进一步递归地划分(使用四叉树划分)成更小的TB,称为残差四叉树(Residual Quadtree, RQT)。

[0136] 下面进一步描述用于将基本块分区为CB、PB和/或TB的另一种示例性实现方式。例如,不使用如图9和图10所示的多个分区单元类型,而可以使用具有嵌套多类型树的四叉树,该嵌套多类型树使用二叉树和三叉树划分分区结构(例如,如上所述的QTBT或包含三叉

树划分的QTBT)。除非其尺寸对于最大变换长度来说太大的CB需要,这样的CB可能需要进一步划分,否则可以放弃CB、PB和TB的分离(即,放弃将CB分区为PB和/或TB,以及将PB分区为TB)。该示例性分区方案可以被设计为针对CB分区形状支持更大灵活性,使得预测和变换都可以在CB水平上执行,而无需进一步分区。在这种编码树结构中,CB可以具有正方形或矩形形状。具体地,编码树块(CTB)可以首先由二叉树结构进行分区。然后,可以通过嵌套多类型树结构进一步分区二叉树叶节点。图11示出了使用二叉树或三叉树划分的嵌套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划分类型,称为垂直二叉树划分(SPLIT_BT_VER)(1102)、水平二叉树划分(SPLIT_BT_HOR)(1104)、垂直三叉树划分(SPLIT_TT_VER)(1106)和水平三叉树划分(SPLIT_TT_HOR)(1108)。然后,CB对应于多类型树的叶。在该示例性实现方式中,除非CB对于最大变换长度来说太大,否则该分段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具有相同的块尺寸。当最大支持的变换长度小于CB的颜色分量的宽度或高度时,会出现异常。在一些实现方式中,除了二叉树或三叉树划分之外,图11的嵌套图案还可以包括四叉树划分。

[0137] 图12示出了具有用于一个基本块的块分区(包括四叉树、二叉树和三叉树划分选项)的嵌套多类型树编码块结构的四叉树的一个具体示例。更详细地说,图12示出了基本块1200被四叉树划分为四个正方形分区1202、1204、1206和1208。针对每个四叉树划分的分区做出进一步将图11的多类型树结构和四叉树用于进一步划分的决定。在图12的示例中,分区1204不被进一步划分。分区1202和1208各自采用另一种四叉树划分。对于分区1202,第二水平四叉树划分的左上、右上、左下和右下分区分别采用第三水平的四叉树划分、图11的水平二叉树划分1104、不划分和图11的水平三叉树划分1108。分区1208采用另一种四叉树划分,第二水平的四叉树划分的左上、右上、左下和右下分区分别采用图11的第三水平的垂直三叉树划分1106、不划分、不划分和图11的水平二叉树划分1104。分别根据图11的水平二叉树划分1104和水平三叉树划分1108进一步划分分区1208的第三水平左上分区的子分区中的两个子分区。分区1206采用根据图11的垂直二叉树划分1102的第二水平划分图案划分为两个分区,这两个分区根据图11的水平三叉树划分1108和垂直二叉树划分1102在第三水平进一步划分,根据图11的水平二叉树划分1104,进一步将第四水平划分应用于它们中的一个。

[0138] 对于上面的特定示例,最大亮度变换尺寸可以是 64×64 ,并且最大支持的色度变换尺寸可以不同于亮度,在例如 32×32 。尽管上面图12中的示例性CB通常不再进一步划分为更小的PB和/或TB,但是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,则可以在水平和/或垂直方向上自动划分亮度编码块或色度编码块以满足该方向上的变换尺寸限制。

[0139] 在上述用于将基本块划分为CB的具体示例中,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。例如,对于P切片和B切片,一个CTU中的亮度和色度CTB可以共享相同的编码树结构。对于I切片,例如,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,可以通过一种编码树结构将亮度CTB划分为亮度CB,而通过另一种编码树结构将色度CTB划分为色度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,而除非视频是单色的,否则P切片或B切片中的CU总是由所有三

个颜色分量的编码块组成。

[0140] 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按照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以及变换块的编码顺序的示例性实现方式将在下面进一步详细描述。在一些示例性实现方式中,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,其中变换块尺寸的范围从例如 4×4 到 64×64 。在一些实现方式中,如果编码块小于或等于 64×64 ,则变换块分区可以仅应用于亮度分量,使得对于色度块,变换块尺寸与编码块尺寸相同。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隐式地划分为 $\min(W, 64) \times \min(H, 64)$ 和 $\min(W, 32) \times \min(H, 32)$ 变换块的倍数。

[0141] 在变换块分区的一些示例性实现方式中,对于帧内编码块和帧间编码块,可以将编码块进一步分区为分区深度高达预定义数量的水平(例如,2水平)的多个变换块。变换块分区深度和尺寸可以关联。对于一些示例性实现方式,从当前深度的变换尺寸到下一个深度的变换尺寸的映射在下面的表1中示出。

[0142] 表1:变换分区尺寸设置

当前深度的变换尺寸	下一深度的变化尺寸
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
[0143] 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

[0144] 基于表1的示例性映射,对于1:1正方形块,下一水平的变换划分可以创建四个1:1正方形子变换块。例如,变换分区可以在 4×4 处停止。因此, 4×4 的当前深度为变换尺寸对应下一深度的相同 4×4 尺寸。在表1的示例中,对于1:2/2:1的非正方形块,下一水平的变换划分可以创建两个1:1的正方形子变换块,而对于1:4/4:1的非正方形块,下一水平的变换划分可以创建两个1:2/2:1的子变换块。

[0145] 在一些示例实现方式中,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的限制。例如,对于变换划分的每一水平,可以限制所有子变换块具有相等的尺寸。例如,对于 32×16 编码块,水平1的变换划分创建两个 16×16 的子变换块,水平2的变换划分创建八个 8×8 的子变换块。换句话说,水平2的划分必须应用于所有水平1的子块,以保持变

换单元尺寸相等。图15示出了帧内编码正方形块的遵循表1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划分成4个尺寸相等的变换块的第一水平划分,其中编码顺序由箭头指示。在1506中示出了根据表1将所有第一水平相等尺寸块划分为16个相等尺寸变换块的第二水平划分,其中编码顺序由箭头指示。

[0146] 在一些示例实现方式中,对于帧间编码块的亮度分量,可以不应用上述对帧内编码的限制。例如,在第一水平变换划分之后,子变换块中的任何一个可以被进一步独立地划分为一个以上的水平。因此,得到的变换块可以具有相同的尺寸,也可以不具有相同的尺寸。图16示出了将帧间编码块划分为变换块的示例划分连同变换块的编码顺序。在图16的示例中,根据表1将帧间编码块1602划分为两个水平的变换块。在第一水平,将帧间编码块划分为尺寸相等的四个变换块。然后,仅四个变换块中的一个(不是全部)被进一步划分为四个子变换块,得到具有两种不同水平的总共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。

[0147] 在一些示例实现方式中,对于色度分量,可以应用针对变换块的一些附加限制。例如,对于色度分量,变换块尺寸可以与编码块尺寸一样大,但不小于预定义的例如 8×8 的尺寸。

[0148] 在一些其它示例实现方式中,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划分为 $\min(W, 64) \times \min(H, 64)$ 和 $\min(W, 32) \times \min(H, 32)$ 变换单元。此处,在本公开中,“ $\min(a, b)$ ”可以返回a和b之间的较小值。

[0149] 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可选示例方案。如图17所示,替代使用递归变换分区,可以根据编码块的变换类型将预定义的一组分区类型应用于编码块。在图17所示的特定示例中,可以应用6个示例分区类型中的一个来将编码块划分为不同数量的变换块。这种生成变换块分区的方案可以应用于编码块或预测块。

[0150] 更详细地说,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(转换类型指的是例如主转换的类型,例如ADST等)。在该方案中,基于例如率失真成本,可以为每个编码块或预测块分配变换分区类型。在一个示例中,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变换块划分尺寸和图案,如图17示出的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:

[0151] • PARTITION_NONE:分配等于块尺寸的变换尺寸。

[0152] • PARTITION_SPLIT:分配宽度为块尺寸的宽度的1/2且高度为块尺寸的高度的1/2的变换尺寸。

[0153] • PARTITION_HORZ:分配宽度与块尺寸的宽度相同且高度为块尺寸的高度的1/2的变换尺寸。

[0154] • PARTITION_VERT:分配宽度为块尺寸的宽度的1/2且高度与块尺寸相同的变换尺寸。

[0155] • PARTITION_HORZ 4:分配宽度与块尺寸的宽度相同且高度为块尺寸的高度的1/4的变换尺寸。

[0156] • PARTITION_VERT 4:分配宽度为块尺寸的宽度的1/4且高度与块尺寸相同的变换尺寸。

[0157] 在上面的示例中,变换分区类型如图17所示,都包含用于分区的变换块的统一变换尺寸。这只是示例,而非限制。在一些其他实现方式中,混合的变换块尺寸可以用于特定分区类型(或图案)的分区变换块。

[0158] 从上述任何分区方案获得的PB(或CB,当没有被进一步分区为预测块时也称为PB)然后可以成为用于经由帧内预测或帧间预测进行编码的单独的块。对于当前PB的帧间预测,可以生成、编码当前块和预测块之间的残差,并将其包括在编码的比特流中。

[0159] 可以例如在单参考模式或复合参考模式中实现帧间预测。在一些实现方式中,跳过(skip)标志可以首先包括在当前块的比特流中(或者包括在更高的水平中),以指示是否帧间编码当前块以及是否不跳过当前块。如果帧间编码当前块,则另一个标志可以进一步包括在比特流中作为信令,以指示针对当前块的预测是使用单参考模式还是使用复合参考模式。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,可以使用两个或更多个参考块来通过例如加权平均来生成预测块。复合参考模式可以称为多于一个参考模式、两个参考模式或多个参考模式。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位置上(例如,水平像素上和垂直像素上)的偏移的一个或多个对应的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单参考块生成,该单参考块被参考帧中的一个运动矢量识别为单参考模式中的预测块,而对于复合参考模式,预测块可以根据由两个参考帧索引和两个对应的运动矢量指示的两个参考帧中的两个参考块的加权平均生成。可以以各种方式编码运动矢量并将其包括在比特流中。

[0160] 在一些实现方式中,编码或解码系统可以维持解码图片缓冲器(DPB)。一些图像/图片可以保存在DPB中等待被显示(在解码系统中),并且DPB中的一些图像/图片可以用作参考帧以使得能够进行帧间预测(在解码系统或编码系统中)。在一些实现方式中,可以将DPB中的参考帧标记为正在编码或解码的当前图片的短期参考或长期参考。例如,短期参考帧可以包括用于当前帧中的块的帧间预测的帧,或者在解码顺序中最接近当前帧的预定义数量(例如,2个)的后续视频帧中的块的帧间预测的帧。长期参考帧可以包括DPB中的帧,该DPB中的帧可用于预测在解码顺序中远离当前帧大于预定义数量的帧中的图像块。关于短期和长期参考帧的这种标签的信息可以称为参考图片集(Reference Picture Set,RPS),并且可以将其添加到编码的比特流中的每个帧的头部。编码视频流中的每个帧可以由图片顺序计数(Picture Order Counter,POC)来识别,该图片顺序计数根据回放序列以绝对值方式进行编号,或者与例如从I帧开始的图片组相关。

[0161] 在一些示例实现方式中,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辨别的一个或多个参考图片列表。例如,针对单向帧间预测可以形成单个图片参考列表,表示为L0参考(或参考列表0),而针对双向帧间预测可以形成两个图片参考列表,表示为针对两个预测方向中的每预测方向的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义的方式排序。L0和L1列表的长度可以在视频比特流中发信令通知。当用于在复合预测模式中通过加权平均生成预测块的多个参考在将要预测的块的同一侧时,单向帧间预测可以是单参考模式,或者是复合参考模式。双向帧间预测可以仅是复合参考模式,因为双向帧间预测涉及至少两个参考块。

[0162] 在一些实现方式中,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,而不是单独地计算并发信令通知。例如,在编码系统中,当前PB的当前运动矢量可以由当前运动的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。替代当前运动矢量的全部,运动矢量的这种差可以被编码并包括在比特流中,并且可以链接到参考运动矢量。相应地,在解码系统中,当前PB对应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,可以实现一般的MM或特定的MMVD以利用与不同PB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MMVD可以很小并且可以被有效地编码。对于另一个示例,对于在空间中类似地定位/定向的块,运动矢量可以在时间上(在帧之间)相关。

[0163] 在一些示例实现方式中,在编码过程期间,可以将MM标志包括在比特流中以指示当前PB是否处于合并模式。另外地或可选地,在编码过程期间,可以将MMVD标志包括在比特流中,并在被比特流中发信令通知以指示当前PB是否处于MMVD模式。可以在PB水平、CB水平、CU水平、CTB水平、CTU水平、切片水平、图片水平等提供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,并且可以紧接在跳过标志和MM标志之后发信令通知MMVD标志以明确指出当前CU是否使用MMVD模式。

[0164] 在MMVD的一些示例实现方式中,可以针对正在预测的块形成用于运动矢量预测的参考运动矢量(reference motion vector,RMV)或MV预测器候选者的列表。RMV候选者列表可包含预定义数目(例如2)的MV预测器候选块,其运动矢量可用于预测当前的运动矢量。RMV候选块可以包括从同一帧中的相邻块和/或时间块(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。这种选择方式表示相对于当前块的这些空间或时间位置上的可能具有与当前块相似或相同的运动矢量的块。可以预先确定MV预测器候选者列表的大小。例如,列表可能包含两个或更多候选者。为了出现在RMV候选者列表上,该候选块,例如可能需要与当前块具有相同的一个或多个参考帧,需要必须存在(例如,在当前块靠近该帧的边缘时,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。在一些实现方式中,可以首先在可用并满足上述条件的情况下用空间上相邻的块(以特定的预定义的顺序扫描)填充合并候选者的列表,然后如果合并的候选者列表中的空间仍然可用,则填充时间块。例如,相邻的RMV候选块可以从当前块的左侧块和顶部块中选择。RMV预测器候选者的列表可以在不同水平(序列、图片、帧、切片、超级块等)动态形成,作为动态参考列表(Dynamic Reference List,DRL)。可以在比特流中发信令通知DRL。

[0165] 在一些实现方式中,可以发信令通知用作用于预测当前块的运动矢量的参考运动矢量的实际的MV预测器候选者。在RMV候选者列表包含两个候选者的情况下,可以使用称为合并候选者标志的一比特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的候选者列表的参考运动矢量关联。编码器可以确定RMV候选者中的哪一个更接近预测当前编码块,并将该选择作为索引发送到DRL中。

[0166] 在MMVD的一些示例实现方式中,在选定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差的幅度和MV差的方向的信息,这两种信息都可以在比特流中发信令通知。可以以各种方式发信号通知运动矢量差的幅度和运动矢量差的方向。

[0167] 在MMVD的一些示例实现方式中,距离索引可用于指定运动矢量差的幅度信息,并指示表示相对于起点(参考运动矢量)的预定义的运动矢量差的一组预定义偏移中的一个。然后,可以根据发信令通知的索引的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的方向信息来确定。表2中指定了距离索引和预定义的偏移之间的示例性预定义关系。

[0168] 表2:距离索引和预定义的MV偏移之间的示例性关系

[0169]	距离索引	0	1	2	3	4	5	6	7
	偏移 (以亮度样本为单位)	1/4	1/2	1	2	4	8	16	32

[0170] 在MMVD的一些示例实现方式中,方向索引可以进一步被发信令通知并用于表示MVD相对于参考运动矢量的方向。在一些实现方式中,方向可以被限制为水平方向和垂直方向中的任一个。表3中示出了示例2比特方向索引。在表3的示例中,对MVD的理解可以根据起始/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两个参考帧列表都指向当前图片的同一侧的双预测块(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)中。当起始/参考MV对应于两个参考图片位于当前图片不同侧的双预测块(即,一个参考图片的POC大于当前图片的POC,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与当前帧之间的差大于图片参考列表1中的参考POC与当前帧之间的差时,表3中的符号可以指定添加到与图片参考列表0中的参考图片对应的参考MV的MV偏移的符号中,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(偏移的相反符号)。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关联的参考MV中的MV偏移的符号,并且与图片参考列表0关联的参考MV的偏移的符号具有相反的值。

[0171] 表3:由方向索引指定的MV偏移的符号的示例实现方式

[0172]	方向IDX	00	01	10	11
	X轴(水平)	+	-	N/A	N/A
	Y轴(垂直)	N/A	N/A	+	-

[0173] 在一些示例实现方式中,可以根据POC在每个方向上的差来缩放MVD。如果两个列表中POC的差相同,则不需要缩放。否则,如果参考列表0中的POC的差大于参考列表1中的POC的差,则缩放参考列表1的MVD。如果参考列表1中的POC的差大于列表0,则可以以相同的方式缩放列表0的MVD。如果起始MV是单向预测的,则添加MVD至可用的MV或参考MV。

[0174] 在用于双向复合预测的MVD编码和信令的一些示例实现方式中,除了单独编码和

发信令通知两个MVD之外或替代单独编码和发信令通知两个MVD,还可以实现对称MVD编码,使得只需要发信令通知一个MVD,而另一个MVD可以从发信令通知的MVD中导出。在这样的实现方式中,包括列列表0和列表1两者的参考图像索引的运动信息是发信令通知的。但是,只有与例如参考列表0关联的MVD是发信令通知的,而与参考列表1关联的MVD不是发信令通知的而是导出的。具体地,在切片水平,标志可以包括在比特流中,称为“mvd_11_zero_flag”,用于指示在比特流中是否发信令通知参考列表1。如果该标志为1,指示参考列表1等于零(因此没有发信令通知),则称为“BiDirPredFlag”的双向预测标志可以被设置为0,这意味着不存在双向预测。此外,如果mvd_11_zero_flag为0,如果列表0中最近的参考图片和列表1中最近的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1参考图片都是短期参考图片。否则,BiDirPredFlag设置为0。BiDirPredFlag为1可以指示在比特流中额外发信令通知对称模式标志。当BiDirPredFlag为1时,解码器可以从比特流中提取对称模式标志。例如,可以在CU水平上发信令通知(如果需要的话)对称模式标志,并且该对称模式标志可以指示对称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,指示使用对称MVD编码模式,且发信令通知了仅列表0和列表1(称为“mvp_10_flag”和“mvp_11_flag”)两者的参考图片索引以及与列表0关联的MVD(称为“MVD0”),而另一运动矢量差“MVD1”将被导出而不是发信令通知。例如,可以导出MVD1为-MVD0。因此,在示例的对称MVD模式中仅发信令通知一个MVD。在用于MV预测的一些其它示例实现方式中,协调方案可用于实现用于单参考模式和复合参考模式MV预测的一般合并模式、MMVD和一些其它类型的MV预测。可以使用各种语法元素来发信令通知预测当前块的MV的方式。

[0175] 例如,对于单参考模式,可以发信令通知以下MV预测模式:

[0176] NEARMV-在没有任何MVD的情况下直接使用DRL索引指示的列表中的运动矢量预测器(motion vector predictor,MVP)中的一者。

[0177] NEWMV-使用由DRL索引发信令通知的列表中的MVP中的一者作为参考,并对该MVP应用一个增量(例如,使用MVD)。

[0178] GLOBALMV-使用基于帧水平全局运动参数的运动矢量。

[0179] 同样地,对于使用对应待预测的两个MV的两个参考帧的复合参考帧间预测模式,可以发信令通知以下MV预测模式:

[0180] NEAR_NEARMV-针对要预测的两个MV中的每一个,在没有MVD的情况下使用由DRL索引发信令通知的列表中的MVP中的一者。

[0181] NEAR_NEWMV-为了预测两个运动矢量中的第一个,在没有MVD的情况下使用由DRL索引发信令通知的列表中的MVP中的一者作为参考MV;为了预测两个运动矢量中的第二个,使用由DRL索引发信令通知的列表中的MVP中的一者作为参考MV,并结合额外发信令通知的增量MV(MVD)。

[0182] NEW_NEARMV-为了预测两个运动矢量中的第二个,在没有MVD的情况下使用由DRL索引发信令通知的列表中的MVP中的一者作为参考MV;为了预测两个运动矢量中的第一个,使用由DRL索引发信令通知的列表中的MVP中的一者作为参考MV,并结合额外发信令通知的增量MV(MVD)。

[0183] NEW_NEWMV-使用由DRL索引发信令通知的列表中的MVP中的一者作为参考MV,并将

其与额外发信令通知的增量MV结合使用,以预测两个MV中的每一个。

[0184] GLOBAL_GLOBALMV-基于其帧水平全局运动参数使用来自每个参考的MV。

[0185] 由此,上面的术语“NEAR”指的是使用参考MV并如一般合并模式那样不使用MVD的MV预测,而术语“NEW”指的是涉及使用参考MV并如在MMVD模式中那样使用发信令通知的MVD来偏移该参考MV的MV预测。对于复合帧间预测,上述参考基本运动矢量和运动矢量增量两者在两个参考之间可以通常不同或独立,即使它们可以相关,并且可以利用这种相关性来减少发信令通知两个运动矢量增量所需的信息量。在这种情况下,可以在比特流中实现和指示两个MVD的联合信令。

[0186] 上面的动态参考列表(DRL)可以用于保存动态维护并被认为是候选运动矢量预测器的一组索引运动矢量。

[0187] 在一些示例实现方式中,可以允许用于MVD的预定义的分辨率。例如,可以允许1/8像素的运动矢量精度(或准确度)。可以以各种方式构造和发信令通知上述各种MV预测模式中的MVD。在一些实现方式中,可以使用各种语法元素来发信令通知参考帧列表0或列表1中的上述运动矢量差。

[0188] 例如,称为“mv_joint”的语法元素可以指定与其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联合用信令通知的。例如,mv_joint的值为

[0189] 0可以指示沿着水平或垂直方向不存在非零MVD;

[0190] 1可以指示仅沿水平方向存在非零MVD;

[0191] 2可以指示仅沿垂直方向存在非零MVD;

[0192] 3可指示沿水平方向和垂直方向均存在非零MVD。

[0193] 当用于MVD的“mv_joint”语法元素发信令通知不存在非零MVD分量时,则可以不发信令通知进一步的MVD信息。然而,如果“mv_joint”语法发信令通知存在一个或两个非零分量,则可以针对每个非零MVD分量进一步发信令通知附加语法元素,如下所述。

[0194] 例如,称为“mv_sign”的语法元素可用于附加指定对应的运动矢量差分量是正的还是负的。

[0195] 对于另一示例,称为“mv_class”的语法元素可用于指定对应非零MVD分量的预定义的一组类中的运动矢量差的类。例如,运动矢量差的预定义的类可用于将运动矢量差的连续幅度空间划分成不重叠的范围,每个范围对应于一种MVD类。因此,发信令通知的MVD类指示相应MVD分量的幅度范围。在下表4所示的示例性实现方式中,较高的类对应于具有较大幅度范围的运动矢量差。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。

[0196] 表4:用于运动矢量差的幅度类

[0197]

MV类	MVD的幅度
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]

[0198] 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地发信令通知的MV类幅度范围的起始幅度之间的偏移的整数部分。因此,mv_bit可以指示MVD的幅度或振幅。“mv_bit”中用于发信令通知每个MVD类的全范围所需的比特数可以作为MV类的函数而变化。例如,表4的实现方式中的MV_CLASS 0和MV_CLASS 1可能仅仅需要单个比特来指示从为0的起始MVD起1或2的整数像素偏移;表4的示例性实现方式中的每个更高的MV_CLASS会需要比先前的MV_CLASS逐步多一个“mv_bit”位。

[0199] 在一些其它示例中,称为“mv_fr”的语法元素可进一步用于指定对应非零MVD分量的运动矢量差的前2个小数位,而称为“mv_hp”的语法元素可用于指定对应非零MVD分量的运动矢量差的第3个小数位(高分辨率位)。两位“mv_fr”实质上提供1/4像素MVD分辨率,而“mv_hp”位可进一步提供1/8像素分辨率。在一些其它实现方式中,可以使用多于一个“mv_hp”位来提供比1/8像素更精细的MVD像素分辨率。在一些示例实现方式中,可以在各种水平中的一个或多个水平上发信令通知附加的标志,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,那么可能不会发信令通知上面针对对应的不支持的MVD分辨率的语法元素。

[0200] 在上面的一些示例实现方式中,小数的分辨率可以独立于不同的MVD类。换句话说,不管运动矢量差的幅度如何,可以使用用于发信令通知非零MVD分量的小数位MVD的预定义数量的“mv_fr”和“mv_hp”位来提供用于运动矢量分辨率的类似选项。

[0201] 然而,在一些其他示例实现方式中,可以区分各种MVD幅度类中的运动矢量差的分辨率。具体地,对于较高MVD类的大MVD幅度的高分辨率MVD可能不会在压缩效率方面提供统计上显著的改善。因此,对于较大的MVD幅度范围,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,所述较大的MVD幅度范围对应更高的MVD幅度类。同样地,通常对于较大的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)来编码MVD。这种依赖MVD类别或依赖MVD幅度的MVD分辨率通常可称为自适应MVD分辨率、依赖幅度的自适应MVD分辨率或依赖幅度的MVD分辨率。术语“分辨率”可进一步称为“像素分辨率”,自适应MVD分辨率可按如下示例实现方式所描述的各种方式实现,以取得总体上更好的压缩效率。特别地,由于统计观察到,以与非适应性方式的低幅度或低类MVD相似的水平处理大幅度或高类MVD的MVD分辨率可能不会显著增加具有大幅度或高类MVD的块的帧间预测残差编码效率,因此,旨在较低精度MVD的减少的信令位数可能大于作为这种较低精度MVD的结果的编码帧间预测残差所需的附加位数。换句话说,将较高的MVD分辨率用于大幅度或高类MVD可能不会比使用较低的MVD分辨率产生太多的编码增益。

[0202] 在一些一般示例实现方式中,MVD的像素分辨率或精度可以随着MVD类的增大而降低或不增加。降低MVD的像素分辨率对应更粗的MVD(或从一个MVD水平到下一个MVD水平的更大步长)。在一些实现方式中,可以指定、预定义或预配置MVD像素分辨率和MVD类之间的对应关系,因此可以不需要在编码的比特流中发信令通知该对应关系。

[0203] 在一些示例实现方式中,表3的MV类可以各自与不同的MVD像素分辨率关联。

[0204] 在一些示例实现方式中,每个MVD类可以与单个允许的分辨率关联。在一些其他实现方式中,一个或多个MVD类可以与两个或更多个可选MVD像素分辨率关联。因此,在比特流中的针对具有这种MVD类的当前MVD分量的信号之后可以是用于指示为当前MVD分量选择的可选像素分辨率的附加信令。

[0205] 在一些示例实现方式中,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)。因此,升序MVD类中的每个类可以以非升序方式与这些MVD像素分辨率之一关联。在一些实现方式中,MVD类可以与以上两个或更多个分辨率关联,并且较高的分辨率可以低于或等于前面MVD类的较低分辨率。例如,如果表4的MV_CLASS_3与可选的1像素和2像素分辨率相关联,则表4的MV_CLASS_4可以与之关联的最高分辨率将是2像素。在一些其他实现方式中,MV类的最高允许的分辨率可以高于前面(较低)MV类的最低允许分辨率。但是,针对升序MV类的允许的分辨率的平均可能只是非升序的。

[0206] 在一些实施方式中,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。

[0207] 在一些示例实现方式中,可仅允许小数像素分辨率用于低于或等于阈值MVD类的MVD类。例如,在表4中,可能只允许小数像素分辨率用于MVD-CLASS 0,而不允许用于所有其他MV类。同样,可能仅允许小数像素分辨率用于低于或等于表4的任何一个其他MV类的MVD类。对于高于阈值MVD类的其他MVD类,仅允许整数像素分辨率用于MVD。以这种方式,对于用高于或等于阈值MVD类的MVD类发信令通知的MVD,可以不需要发信令通知诸如“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位来发信令通知具有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像素分辨率的分辨率)关联,则“mv-bit”可能需要4比特而不是5比特,并且在发信令通知“mv_class”为MV-CLASS_5之后,不需要发信令通知“mv_fr”和“mv_hp”中的任何一个。

[0208] 在一些示例实现方式中,对于整数值低于阈值整数像素值的MVD,可以仅允许小数像素分辨率。例如,可能仅允许小数像素分辨率用于小于5像素的MVD。对应该示例,可以允许小数分辨率用于表4的MV_CLASS_0和MV_CLASS_1,而不允许用于所有其它MV类。对于另一个示例,可能仅允许小数像素分辨率用于小于7像素的MVD。对应该示例,可以允许小数分辨率用于表4的MV_CLASS_0和MV_CLASS_1(具有低于5像素的范围),而不允许用于MV_CLASS_3和更高的MV类(具有高于5像素的范围)。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,可以或可能允许小数像素分辨率用于MVD。如果发信令通知“m-bit”值为1或2(使得发信令通知的MVD的整数部分是5或6,计算为具有由“m-bit”指示的偏移1或2的MV_CLASS_2的像素范围的开始),则可以允许小数像素分辨率。否则,如果发信令通知“mv-bit”值为3或4(使得发信令通知的MVD的整数部分是7或8),则可以不允许小数像素分辨率。

[0209] 在一些其它实现方式中,对于等于或高于阈值MV类的MV类,可以仅允许单个MVD值。例如,这样的阈值MV类可以是MV_CLASS 2。因此,可以仅允许MV_CLASS_2及高于MV_

CLASS_2的MV类具有单个MVD值而不使用小数像素分辨率。这些MV类的单个允许的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类的相应范围的较高端值。例如,MV_CLASS_2到MV_CLASS_10可以高于或等于阈值类MV_CLASS_2,并且这些类的单个允许的MVD值可以分别预定义为8、16、32、64、128、256、512、1024和2048。在一些其他示例中,允许的单个值可以是针对表4中这些MV类的相应范围的中间值。例如,MV_CLASS_2到MV_CLASS_10可以高于类阈值,并且这些类的单个允许的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类的单个允许的分辨率。

[0210] 在上述实现方式中,只有在发信令通知的“mv_class”等于或高于预定义的MVD类阈值时,“mv_class”信令才足以确定MVD值。然后使用“mv_class”和“mv_sign”来确定MVD的幅度和方向。

[0211] 因此,当针对仅一个参考帧(来自参考帧列表0或列表1,但不是来自两者)发信令通知MVD时,或者针对两个参考帧联合发信令通知MVD时,MVD的精度(或分辨率)可取决于表3中运动矢量差的关联类和/或MVD的幅度。

[0212] 在一些其他实现方式中,MVD的像素分辨率或精度可以随着MVD幅度的增加而降低或不增加。例如,像素分辨率可以取决于MVD幅度的整数部分。在一些实现方式中,可以仅允许小数像素分辨率用于小于或等于幅度阈值的MVD幅度。对于解码器,可以首先从比特流中提取MVD幅度的整数部分。然后可以确定像素分辨率,并且然后可以做出关于任何小数MVD是否存在于比特流中并且是否需要被解析的判断(例如,如果对于特定提取的MVD整数幅度不允许使用小数像素分辨率,则在需要提取的比特流中可以不包括小数MVD位)。上面涉及MVD类相关的自适应MVD像素分辨率的示例实现方式应用于MVD幅度相关的自适应MVD像素分辨率。对于特定示例,可以允许高于或包含幅度阈值的MVD类仅具有一个预定义值。

[0213] 在一些实现方式中,MVD可以主要沿着一个或多个特定方向分布,即,MVD具有高概率沿着一个或多个特定方向,例如水平方向和/或垂直方向。例如,可以通过对MVD施加限制来利用该统计特性,以便提高编码/解码效率并增强帧间编码模式结构。对于非限制性的示例,当一个或多个特定方向包括水平方向和/或垂直方向时,可以沿着水平方向和/或垂直方向限制MVD。

[0214] 图18示出了遵循用于限制运动矢量差(MVD)的上述实现方式的基本原理的示例方法的流程图1800。示例性解码方法流程从1801开始,并且可以包括以下步骤的一部分或全部:在步骤1810中,接收编码的视频比特流;在步骤1820中,从编码的视频比特流获得当前视频块的运动矢量差MVD,其中,当前视频块是处在复合参考模式中的帧间编码块;在步骤1830中,确定MVD是否是针对仅一个预测方向发信令通知的;在步骤1840中,响应于确定MVD是针对仅一个预测方向发信令通知的,将MVD限制为预定义的方向;和/或在步骤1850中,基于MVD解码当前视频块。示例方法可以在步骤1899处停止。在一些实现方式中,在当前视频块是复合参考模式下的帧间编码块时,当前视频块是经由双向帧间预测的。

[0215] 在一些实现方式中,参考帧相对于当前帧的方向可以通过参考帧在显示顺序中是否在当前帧之前,或者参考帧在显示顺序中是否在当前帧之后来确定。

[0216] 在一些实现方式中,可以在复合参考模式中使用两个参考帧。当一个运动矢量对的两个参考帧的POC均大于或均小于当前帧的POC时,可以将两个参考帧的方向视为相同。

否则,当一个参考帧的POC大于当前帧的POC,而另一个参考帧的POC小于当前帧的POC时,可以将两个参考帧的方向视为不同。

[0217] 在一些实现方式中,可以为当前视频块(或当前帧)发信令通知仅一个MVD,可以包括以下情况中的至少一种:在复合参考模式中针对仅一个参考帧,在编码的视频比特流中发信令通知MVD;和/或在复合参考模式中针对两个参考帧联合发信令通知MVD。在上述一些情况下,可以将针对第一参考帧列表(即,参考帧列表0)或第二参考帧列表(即,参考帧列表1)发信令通知的MVD限制在一个或多个预定义的方向(例如,水平方向和/或垂直方向)。

[0218] 对于非限制性的示例,当针对仅一个预测方向发信令通知MVD时(例如,针对仅参考帧列表0或列表1,而不是两者,发信令通知MVD,或者针对两个参考帧列表联合发信令通知MVD),可以将参考帧列表0(或参考帧列表1)的MVD限制到某些预定义的方向。

[0219] 在一些实现方式中,当前视频块处在包括NEW_NEARMV模式或NEAR_NEWMV模式中的至少一个的复合参考模式中。在一些实现方式中,仅在当前视频块处在包括NEW_NEARMV模式或NEAR_NEWMV模式中的至少一个的复合参考模式时,将MVD限制到一个或多个确定的预定义方向。在一些实现方式中,在当前视频块不处在NEW_NEARMV模式或NEAR_NEWMV模式时,不将MVD限制到一个或多个确定的预定义方向。

[0220] 对于非限制性的示例,将MVD限制到确定的预定义的方向仅适用于NEW_NEARMV和/或NEAR_NEWMV模式。

[0221] 在一些实现方式中,响应于当前视频块处在NEW_NEARMV模式中或NEAR_NEWMV模式中,预定义的方向包括水平方向或垂直方向中的至少一个。

[0222] 对于非限制性的示例,对于NEW_NEARMV(或NEAR_NEWMV)模式,可将参考帧列表0(或参考帧列表1)的MVD限制为水平和/或垂直方向。

[0223] 在一些实现方式中,图18中的方法还可以包括从编码的视频比特流获得指示水平方向和垂直方向是否具有非零MVD的标志。在一些实现方式中,标志可以指语法或语法元素。对于非限制性的示例,为了发信令通知参考帧列表0(或参考帧列表1)中的针对NEW_NEARMV(或NEAR_NEWMV)模式的MVD,发信令通知一个标志(例如,mvd_direction_flag)以指示是否存在沿水平或垂直方向的非零MVD。

[0224] 在一些实现方式中,标志的值仅包括以下之一:指示水平方向和垂直方向均不具有非零MVD的第一标志值;指示水平方向和垂直方向中仅有一个方向具有非零MVD的第二标志值;指示水平方向和垂直方向中仅有另一个方向具有非零MVD的第三标志值。

[0225] 在一些实现方式中,响应于当前视频块处在NEW_NEARMV模式中:响应于标志的值等于第二标志值,MVD对应于针对第一参考帧列表中的参考帧的垂直方向,以及响应于标志的值等于第三标志值,MVD对应于针对第一参考帧列表中的参考帧的水平方向;并且响应于当前视频块处在NEAR_NEWMV模式中:响应于标志等于第二标志值,MVD对应于针对第二参考帧列表中的参考帧的水平方向,以及响应于标志等于第三标志值,MVD对应于针对第二参考帧列表中的参考帧的垂直方向。

[0226] 对于非限制性的示例,标志(例如,mvd_direction_flag)的值可以仅为0、1或2,其中0指示沿垂直或水平方向不存在非零MVD,1指示仅沿垂直(或水平)方向存在非零MVD,2指示仅沿水平(或垂直)方向存在非零MVD。

[0227] 在一些实现方式中,标志的值仅包括以下之一:指示水平方向和垂直方向中仅有

一个具有非零MVD的第一标志值;或者水平方向和垂直方向中仅有另一个具有非零MVD的第二标志值。

[0228] 在一些实现方式中,响应于当前视频块处在NEW_NEARMV模式中:响应于标志等于第一标志值,MVD对应于针对第一参考帧列表中的参考帧的垂直方向,以及响应于标志等于第二标志值,MVD对应于针对第一参考帧列表中的参考帧的水平方向;并且响应于当前视频块处在NEAR_NEWMV模式中:响应于标志等于第一标志值,MVD对应于针对第二参考帧列表中的参考帧的水平方向,以及响应于标志的值等于第二标志值,MVD对应于针对第二参考帧列表中的参考帧的垂直方向。

[0229] 对于非限制性的示例,标志(例如,mvd_direction_flag)的值可以仅为0或1,其中0指示仅沿垂直(或水平)方向存在非零MVD,1指示仅沿水平(或垂直)方向存在非零MVD。

[0230] 在一些实现方式中,在高水平语法中发信令通知标志,高水平语法包括以下至少一个:视频参数集VPS、图片参数集PPS、序列参数集SPS、自适应参数集APS、图片头部、帧头部、切片头部、瓦片头部或编码树单元CTU头部。对于非限制性的示例,在包括但不限于VPS、PPS、SPS、APS、帧头部或切片头部的高水平语法中发信令通知指示非零MVD是沿着水平方向还是沿着垂直方向的标志。

[0231] 在一些实现方式中,响应于当前视频块处在NEW_NEARMV模式中或NEAR_NEWMV模式中,预定义的方向仅包括一个方向。对于非限制性的示例,对于NEW_NEARMV(或NEAR_NEWMV)模式,可以将参考帧列表0(或参考帧列表1)的MVD限制到水平方向或垂直方向的仅一个方向。

[0232] 在一些实现方式中,方法1800还包括基于针对当前视频块的模式确定水平方向和垂直方向是否具有非零MVD。

[0233] 在一些实现方式中,响应于当前视频块处在NEW_NEARMV模式中,MVD对应于针对第一参考帧列表中的参考帧的水平方向,和/或响应于当前视频块处在NEAR_NEWMV模式中,MVD对应于针对第二参考帧列表中的参考帧的垂直方向。

[0234] 在一些实现方式中,响应于当前视频块处在NEW_NEARMV模式中,MVD对应于针对第一参考帧列表中的参考帧的垂直方向,和/或响应于当前视频块处在NEAR_NEWMV模式中,MVD对应于针对第二参考帧列表中的参考帧的水平方向。

[0235] 对于非限制性的示例,基于帧间预测模式推断指示是否存在沿水平或垂直方向的非零MVD的标志,例如,水平方向(或垂直方向)的MVD方向可以基于帧间预测模式作为预定义的(或默认)配置。对于非限制性的示例,对于NEW_NEARMV模式,可推断MVD的方向为(或默认为)水平(或垂直)方向;和/或对于NEAR_NEWMV模式,可推断MVD的方向为(或默认为)垂直(或水平)方向。

[0236] 在本公开的实施例和实现方式中,可以根据需要以任何数量或顺序组合或安排任何步骤和/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方式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态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(CU)。这里的术语“块”也可以用来指代变换块。在以下项目中,当提及块尺寸时,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)。

[0237] 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19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(1900)。

[0238] 可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经受汇编、编译、链接或类似的机制以创建包括指令的代码,指令可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,或者通过解释、微代码执行等执行。

[0239] 指令可以在各种类型的计算机或其组件上执行,计算机或其组件包括例如个人计算机、平板计算机、服务器、智能电话、游戏设备、物联网设备等。

[0240] 图19所示的计算机系统(1900)的组件本质上是示例性的,并不旨在对实施本公开的实施例的计算机软件的用途或功能的范围提出任何限制。组件的配置也不应解释为具有与计算机系统(1900)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相关的任何依赖或要求。

[0241] 计算机系统(1900)可包括某些人机接口输入设备。此类人机接口输入设备可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)。人机接口设备还可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:扫描图像、从静止图像相机获取的拍摄图像)、视频(例如,二维视频、包括立体视频的三维视频)。

[0242] 人机接口输入设备可包括下述的一项或多项(每种中仅示出一个):键盘(1901)、鼠标(1902)、触控板(1903)、触摸屏(1910)、数据手套(未示出)、操纵杆(1905)、麦克风(1906)、扫描仪(1907)、相机(1908)。

[0243] 计算机系统(1900)还可包括某些人机接口输出设备。此类人机接口输出设备可通过例如触觉输出、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包括触觉输出设备(例如,触摸屏(1910)的触觉反馈、数据手套(未示出)或操纵杆(1905),但还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(1909)、耳机(未描绘))、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(1910),每种屏幕具有或没有触摸屏输入功能,每种屏幕具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之类的装置、虚拟现实眼镜(未描绘)、全息显示器和烟箱(未描绘)以及打印机(未描绘)来输出二维视觉输出或超过三维的输出。

[0244] 计算机系统(1900)还可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(1921)的CD/DVD ROM/RW(1920)的光学介质、指状驱动器(1922)、可拆卸硬盘驱动器或固态驱动器(1923)、诸如磁带和软盘之类的传统磁性介质(未描绘)、诸如安全软件狗之类的基于专用ROM/ASIC/PLD的设备(未描绘)等。

[0245] 本领域技术人员还应该理解,结合当前公开的主题所使用的术语“计算机可读介质”不涵盖传输介质、载波或其它暂时性信号。

[0246] 计算机系统(1900)还可包括通向一个或多个通信网络(1955)的接口(1954)。网络

可例如是无线网络、有线网络、光网络。网络可进一步是本地网络、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CANBus的车辆和工业网络等。某些网络通常需要附接到某些通用数据端口或外围总线(1949)的外部网络接口适配器(例如,计算机系统(1900)的USB端口);如下所述,其它网络接口通常通过附接到系统总线而集成到计算机系统(1900)的内核中(例如,连接到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(1900)可使用这些网络中的任何网络与其它实体通信。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它计算机系统。如上所述,可以在那些网络和网络接口中的每一个上使用某些协议和协议栈。

[0247] 上述人机接口设备、人机可访问的存储设备和网络接口可附接到计算机系统(1900)的内核(1940)。

[0248] 内核(1940)可包括一个或多个中央处理单元(CPU)(1941)、图形处理单元(GPU)(1942)、现场可编程门区域(FPGA)(1943)形式的专用可编程处理单元、用于某些任务的硬件加速器(1944)、图形适配器(1950)等。这些设备以及只读存储器(ROM)(1945)、随机存取存储器(1946)、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(1947)可通过系统总线(1948)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(1948),以能够通过附加的CPU、GPU等进行扩展。外围设备可直接附接到内核的系统总线(1948)或通过外围总线(1949)附接到内核的系统总线(1948)。在一个示例中,屏幕(1910)可连接到图形适配器(1950)。外围总线的架构包括PCI、USB等。

[0249] CPU(1941)、GPU(1942)、FPGA(1943)和加速器(1944)可执行某些指令,这些指令可组合来构成上述计算机代码。该计算机代码可存储在ROM(1945)或RAM(1946)中。过渡数据可存储在RAM(1946)中,而永久数据可例如存储在内部大容量存储器(1947)中。可通过使用高速缓存来进行通向任何存储设备的快速存储及检索,该高速缓存可与下述紧密关联:一个或多个CPU(1941)、GPU(1942)、大容量存储(1947)、ROM(1945)、RAM(1946)等。

[0250] 计算机可读介质可以在其上具有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。

[0251] 作为非限制性示例,可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(1900),特别是内核(1940)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介绍的用户可访问的大容量存储相关联的介质,以及某些非暂时性内核(1940)的存储器,例如内核内部大容量存储器(1947)或ROM(1945)。实施本公开的各种实施例的软件可存储在此类设备中并由内核(1940)执行。根据特定需要,计算机可读介质可包括一个或多个存储设备或芯片。软件可使得内核(1940),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(1946)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。附加地或作为替代,可由于硬连线或以其它方式体现在电路(例如:加速器(1944))中的逻辑而使得计算机系统提供功能,该电路可替代软件或与软件一起

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合适的组合。

[0252] 虽然本公开已描述多个示例性实施例,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,但是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。

[0253] 附录A:首字母缩写词

[0254] JEM:联合探索模型

[0255] VVC:下一代视频编码

[0256] BMS:基准集

[0257] MV:运动矢量

[0258] HEVC:高效视频编码

[0259] SEI:辅助增强信息

[0260] VUI:视频可用性信息

[0261] GOP:图片群组

[0262] TU:变换单元

[0263] PU:预测单元

[0264] CTU:编码树单元

[0265] CTB:编码树块

[0266] PB:预测块

[0267] HRD:假想参考解码器

[0268] SNR:信噪比

[0269] CPU:中央处理单元

[0270] GPU:图形处理单元

[0271] CRT:阴极射线管

[0272] LCD:液晶显示器

[0273] OLED:有机发光二极管

[0274] CD:光盘

[0275] DVD:数字视频光盘

[0276] ROM:只读存储器

[0277] RAM:随机存取存储器

[0278] ASIC:专用集成电路

[0279] PLD:可编程逻辑器件

[0280] LAN:局域网

[0281] GSM:全球移动通信系统

[0282] LTE:长期演进

[0283] CANBus:控制器局域网络总线

[0284] USB:通用串行总线

- [0285] PCI:互连外围组件
- [0286] FPGA:现场可编程门区域
- [0287] SSD:固态驱动器
- [0288] IC:集成电路
- [0289] HDR:高动态范围
- [0290] SDR:标准动态范围
- [0291] JVET:联合视频探索组
- [0292] MPM:最可能模式
- [0293] WAIP:宽角度帧内预测
- [0294] CU:编码单元
- [0295] PU:预测单元
- [0296] TU:变换单元
- [0297] CTU:编码树单元
- [0298] PDPC:依赖位置的预测组合
- [0299] ISP:帧内子划分
- [0300] SPS:序列参数集
- [0301] PPS:图片参数集
- [0302] APS:自适应参数集
- [0303] VPS:视频参数集
- [0304] DPS:解码参数集
- [0305] ALF:自适应环路滤波器
- [0306] SAO:采样自适应偏移
- [0307] CC-ALF:跨分量的自适应环路滤波器
- [0308] CDEF:约束定向增强滤波器
- [0309] CCSO:跨分量的采样偏移
- [0310] LS0:局部采样偏移
- [0311] LR:环路重建滤波器
- [0312] AV1:A0媒体视频1
- [0313] AV2:A0媒体视频2
- [0314] MVD:运动矢量差
- [0315] CfL:从亮度分量来预测色度分量
- [0316] SDT:半解耦树
- [0317] SDP:半解耦分区
- [0318] SST:半分离树
- [0319] SB:超级块
- [0320] IBC(or IntroBC):块内拷贝
- [0321] CDF:累积密度函数
- [0322] SCC:屏幕内容编码
- [0323] GBI:全面双向预测

- [0324] BCW:CU水平加权双向预测
- [0325] CIIP:结合的帧内帧间预测
- [0326] POC:图片顺序计数
- [0327] RPS:参考图片集
- [0328] DPB:解码图片缓冲器
- [0329] MMVD:具有运动矢量差的合并模式
- [0330] MV:运动矢量
- [0331] MVP:运动矢量预测器

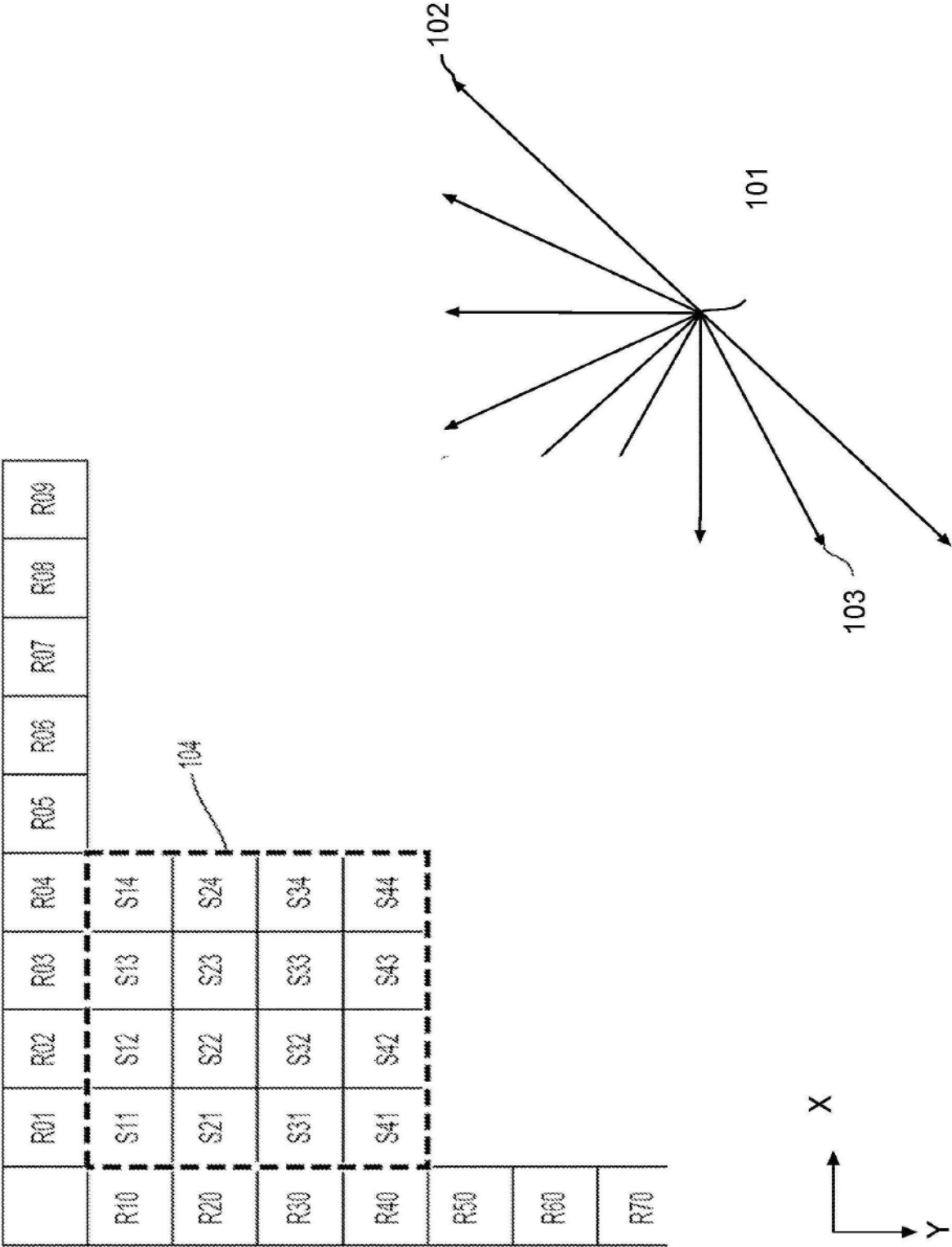


图1A

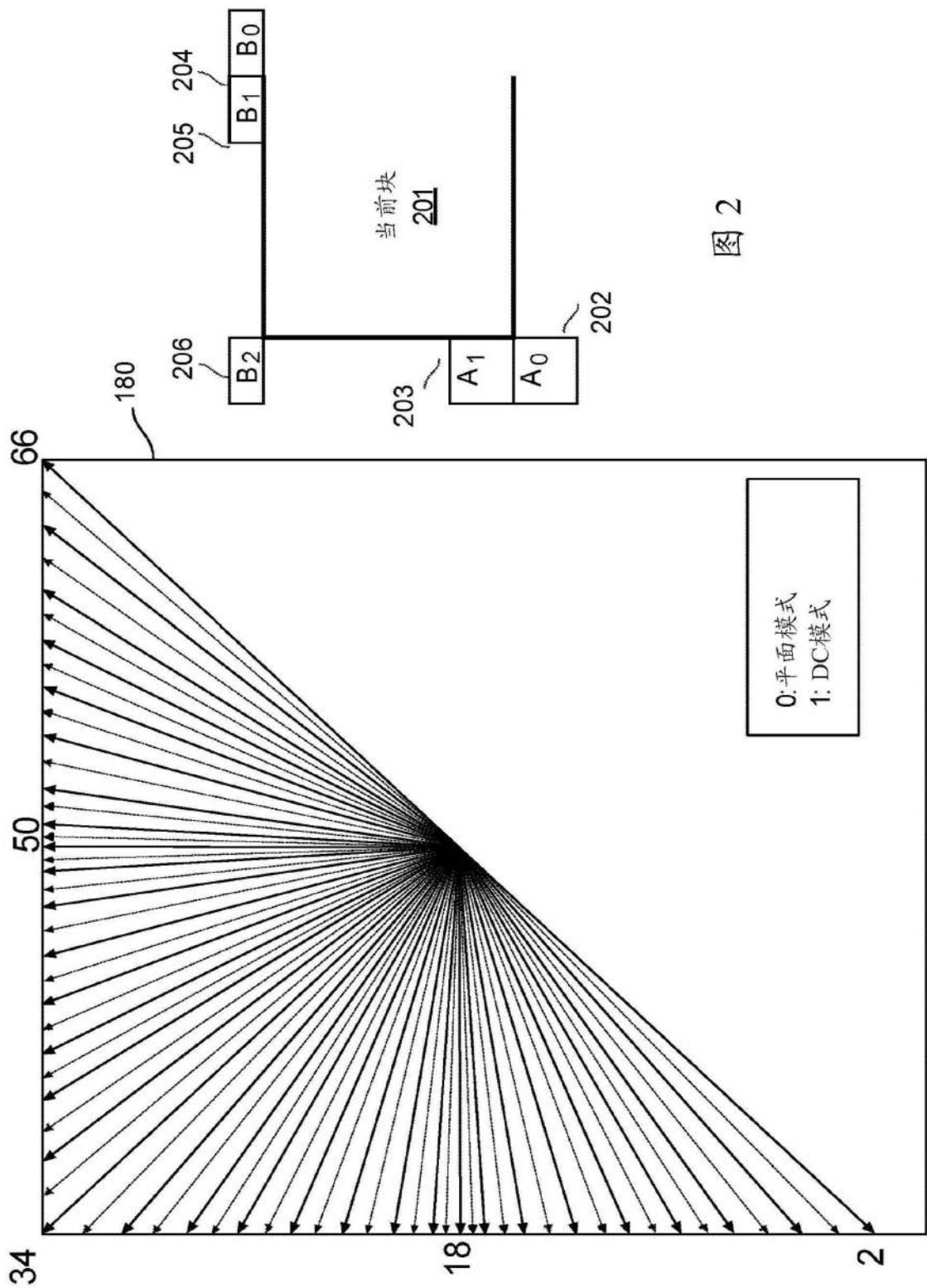


图 2

图 1B

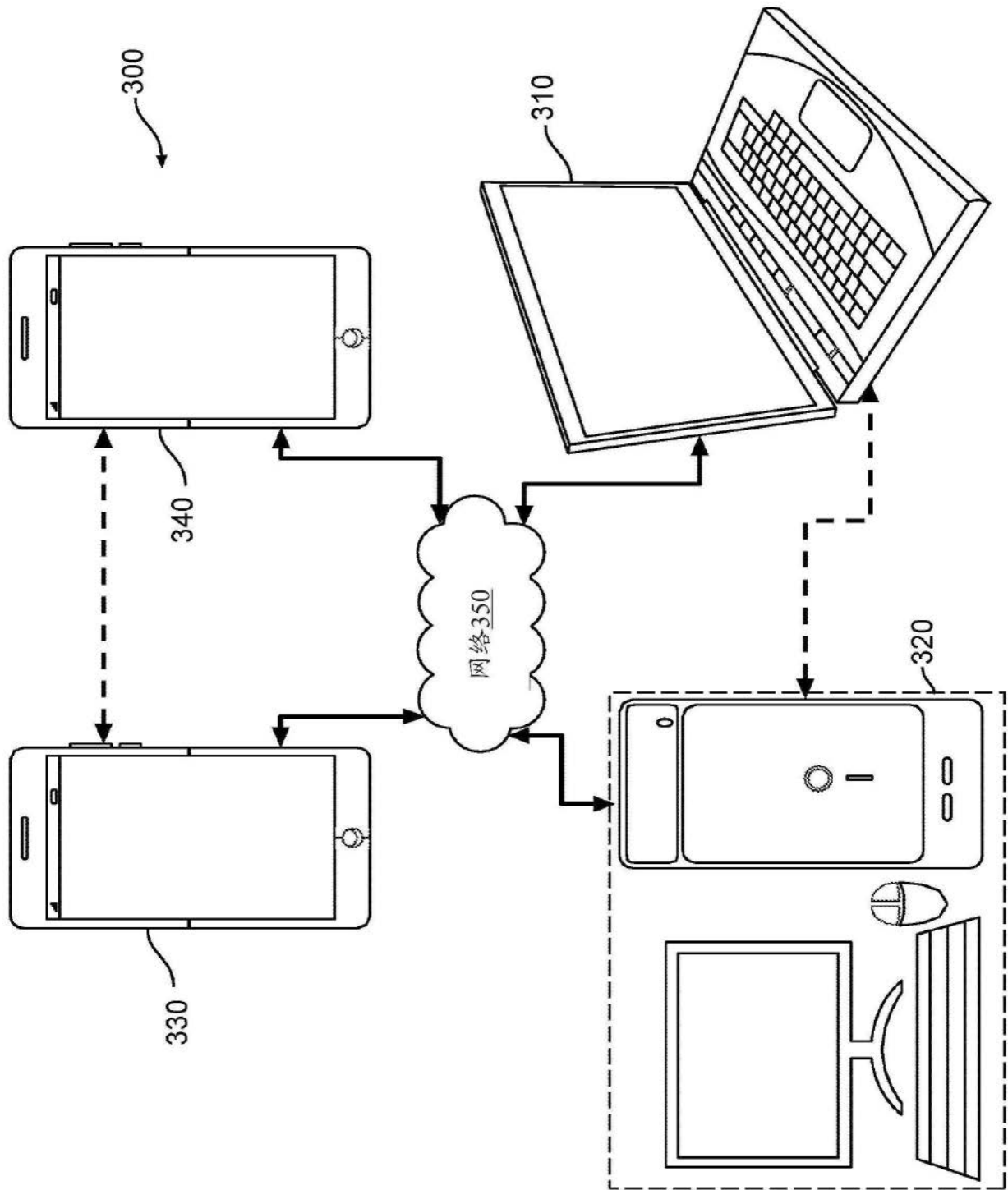


图3

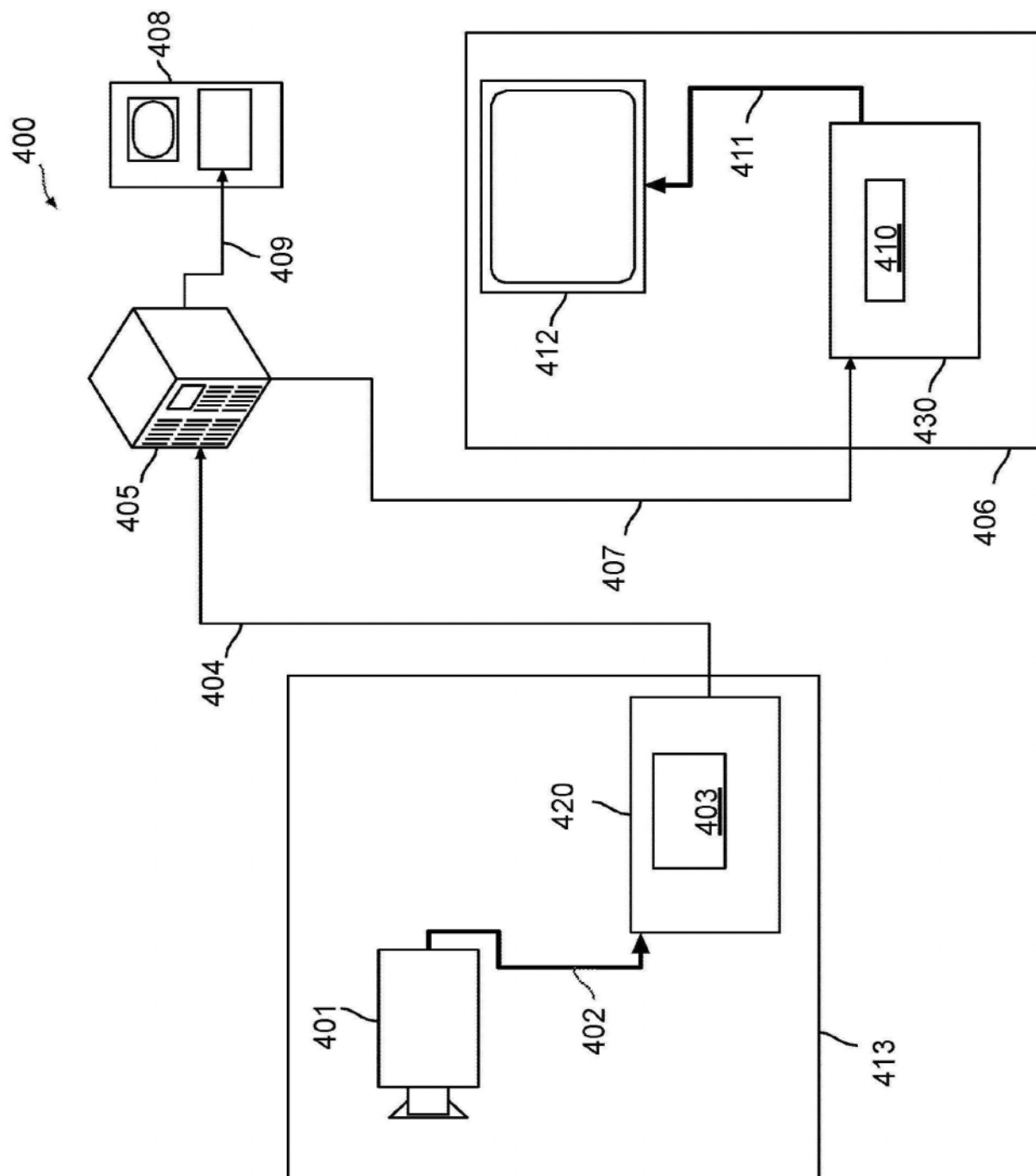


图4

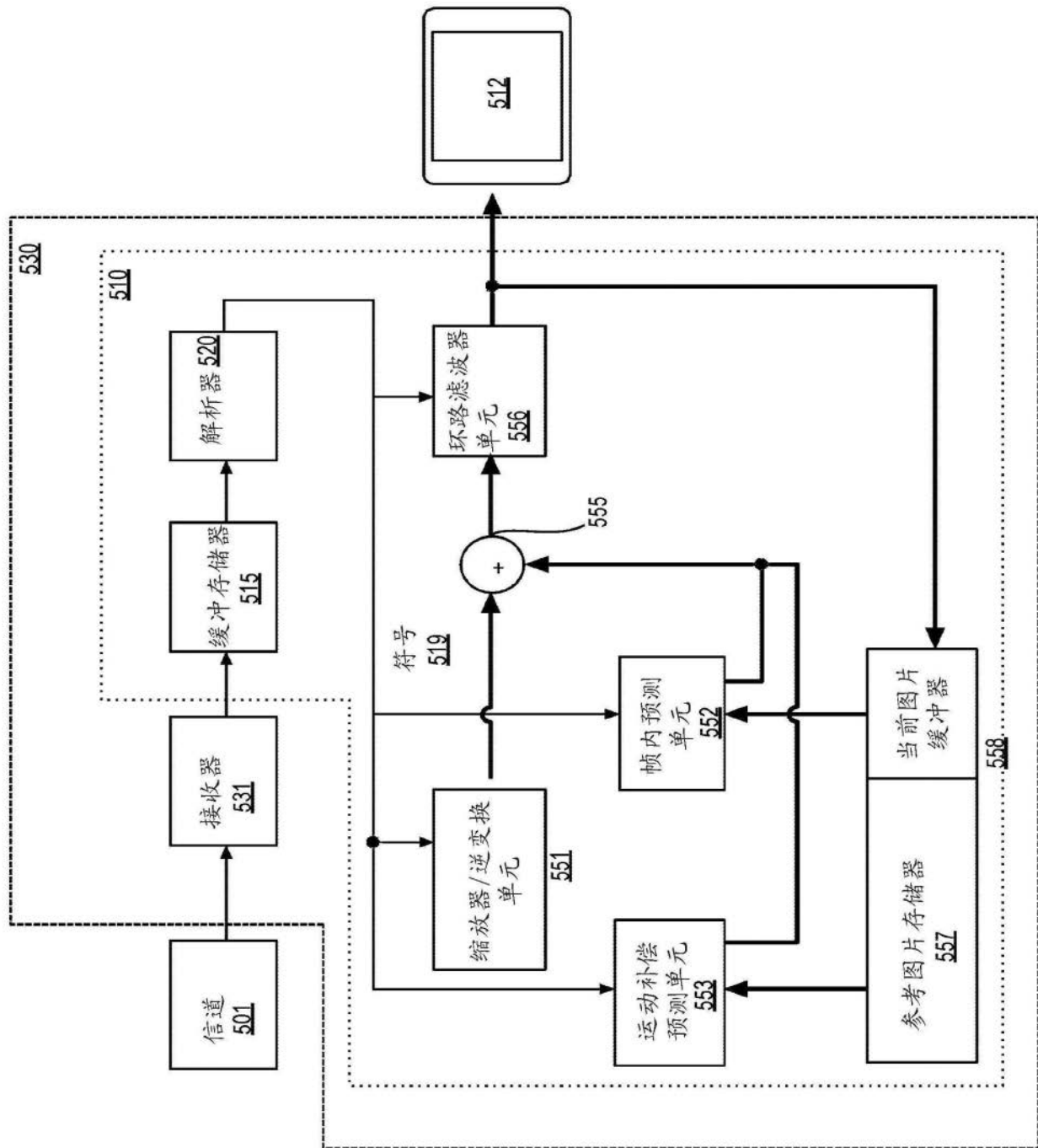


图5

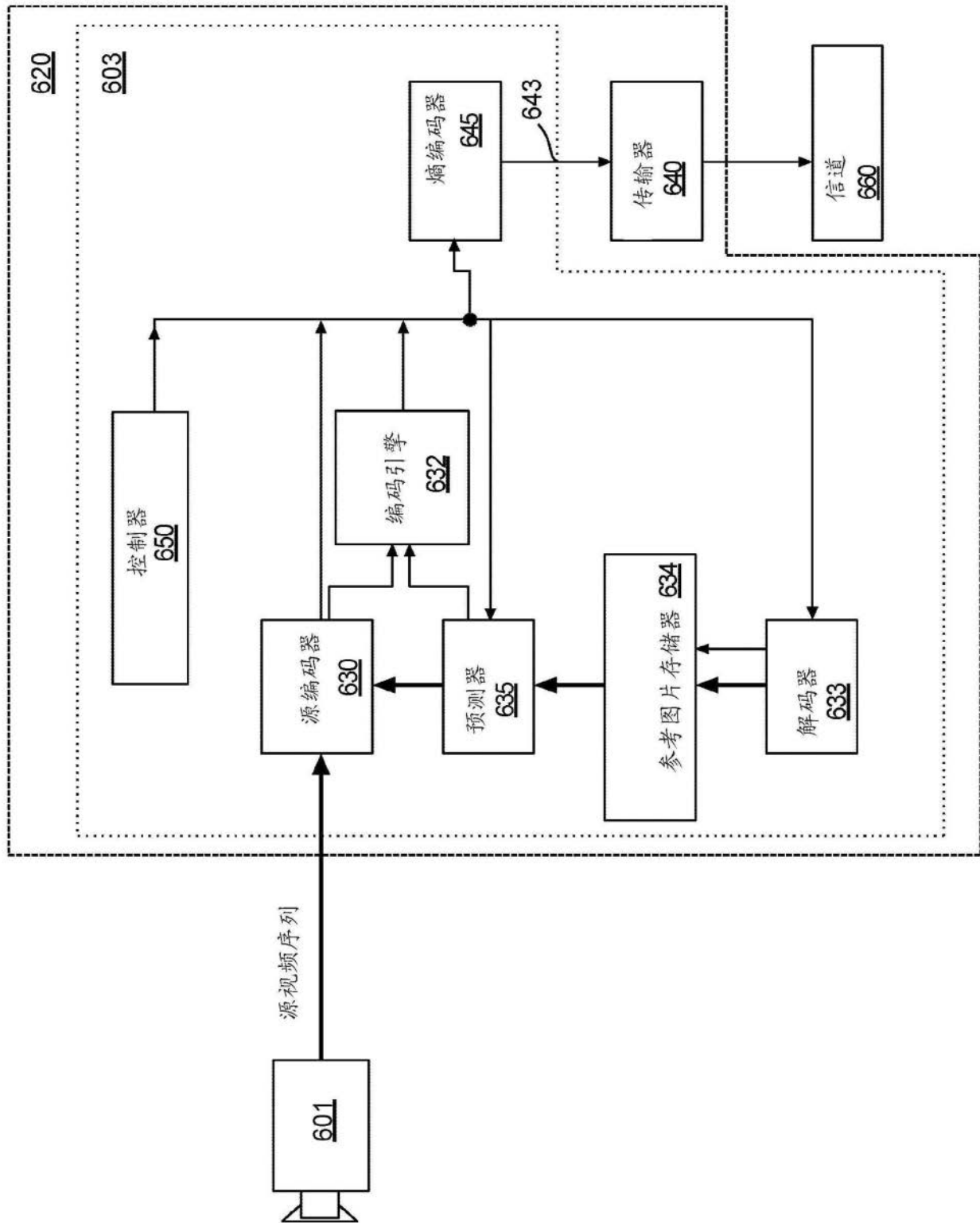


图6

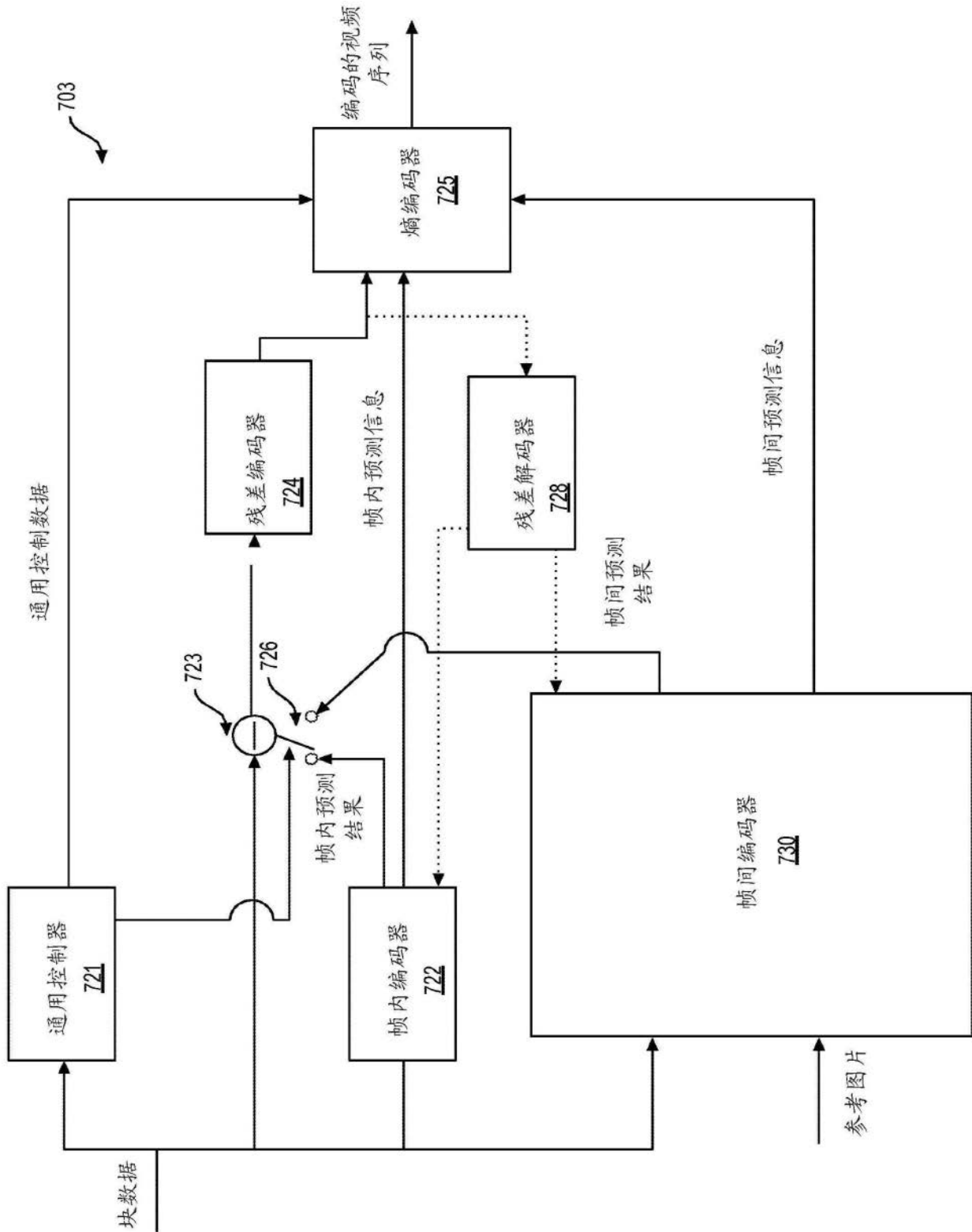


图7

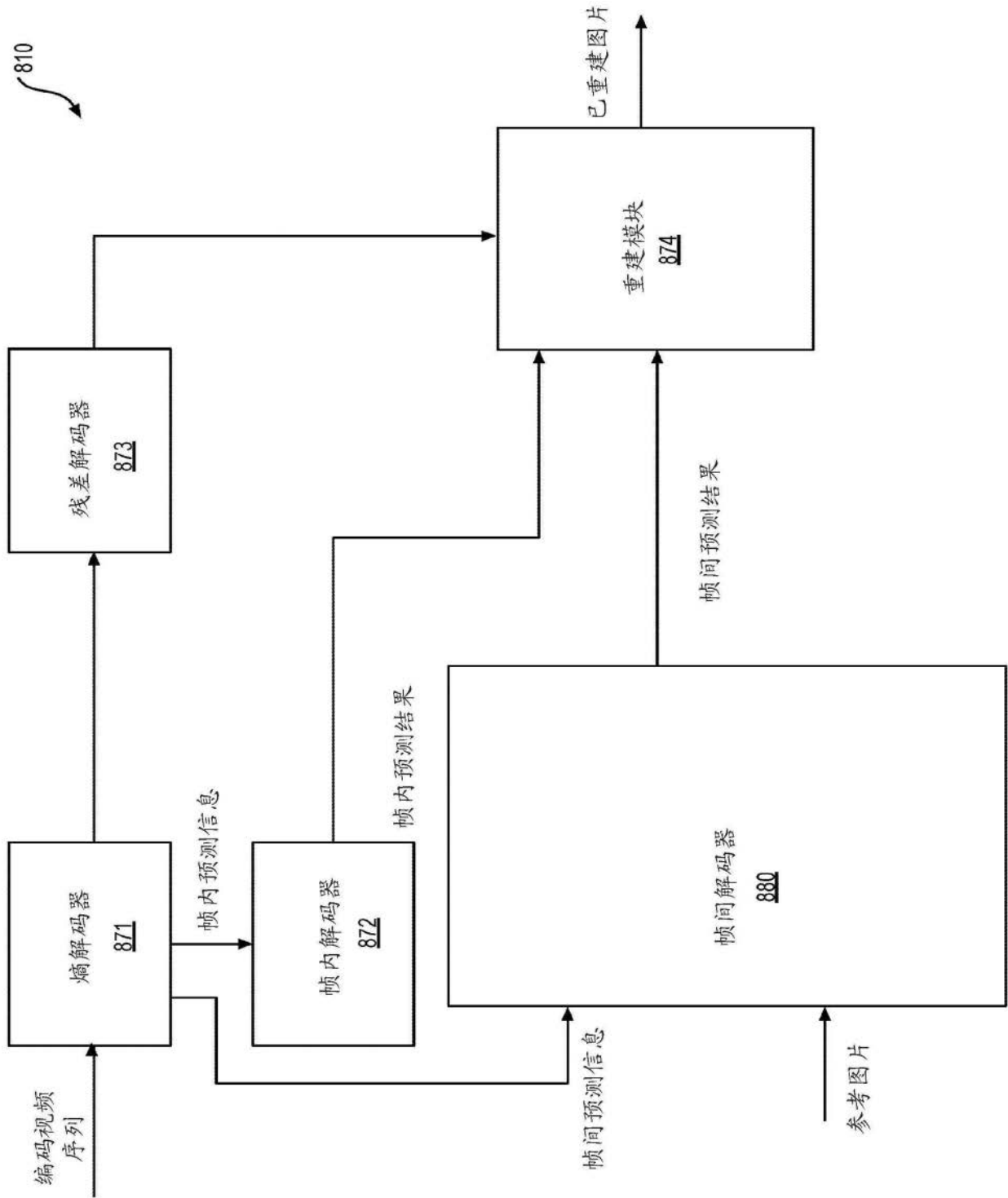


图8

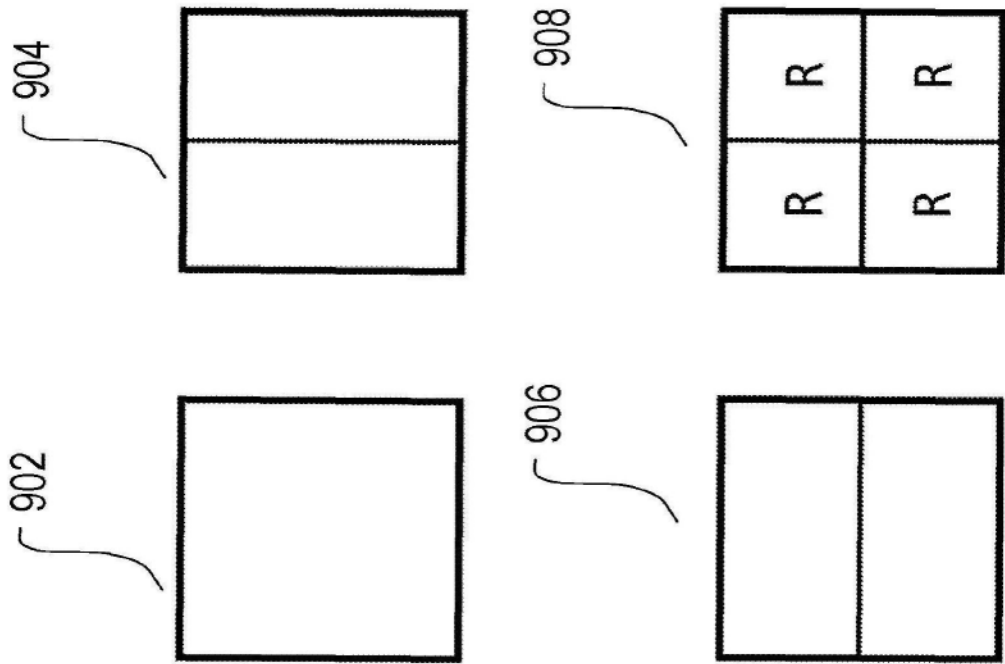


图9

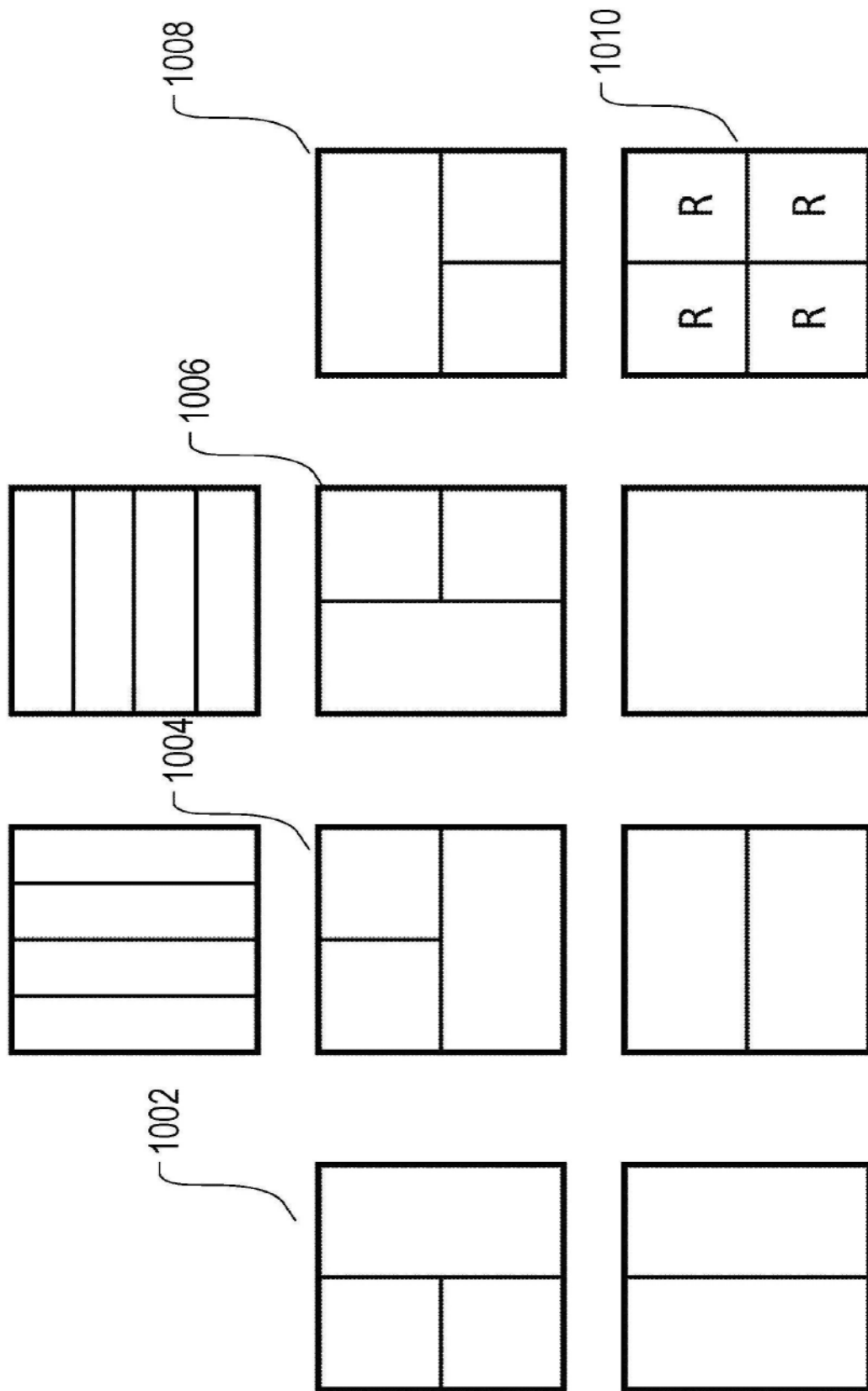


图10

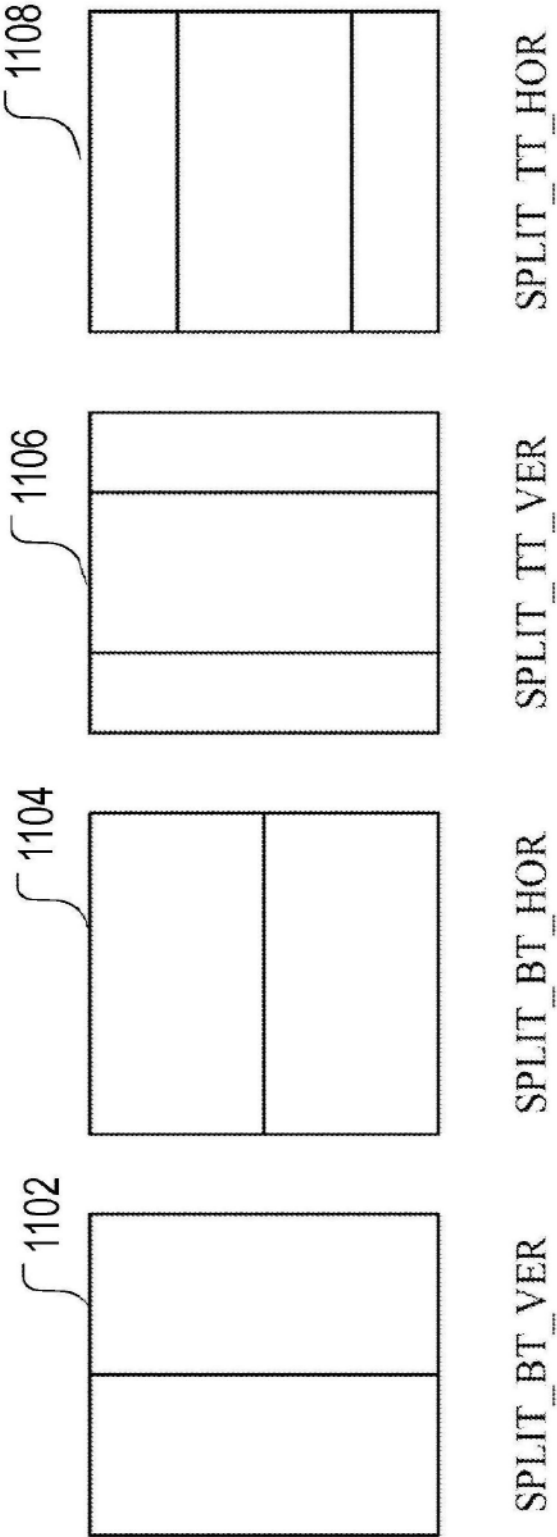


图11

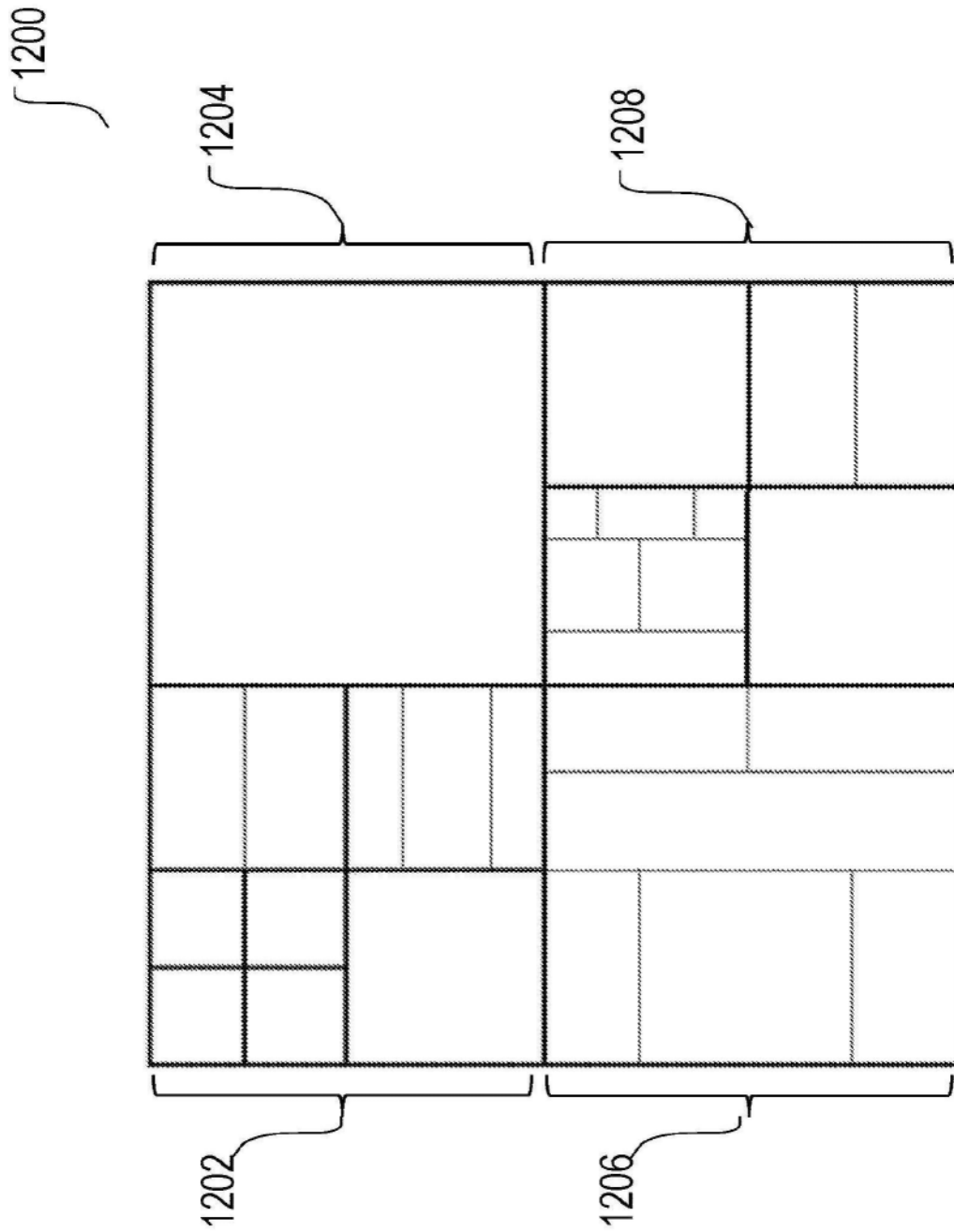


图12

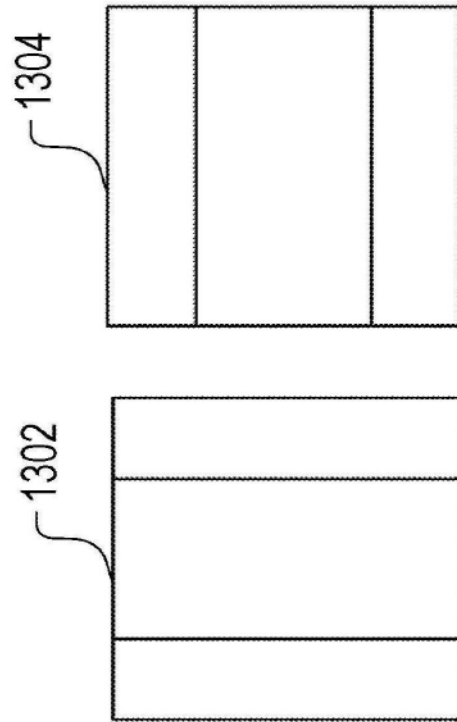


图13

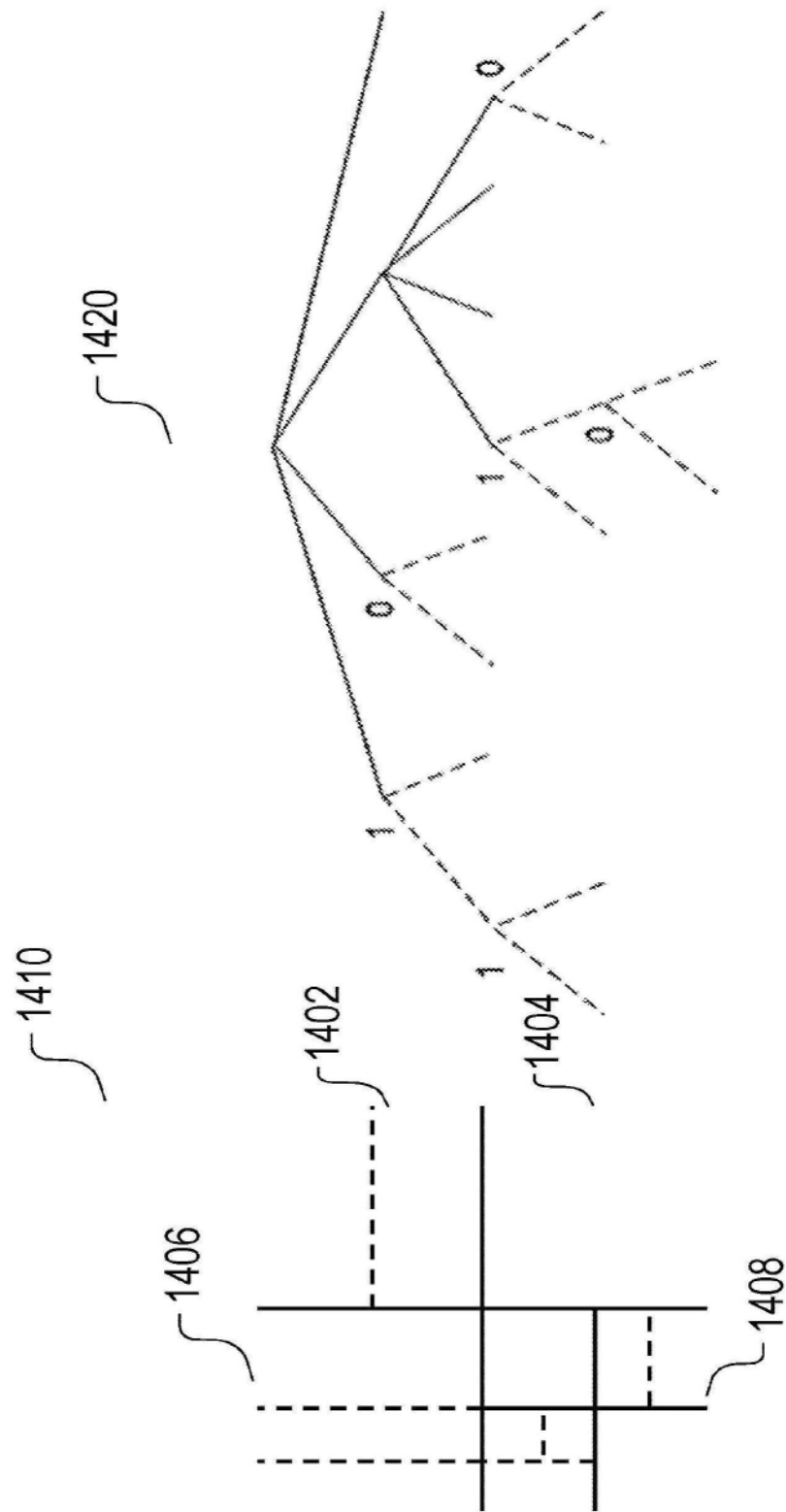
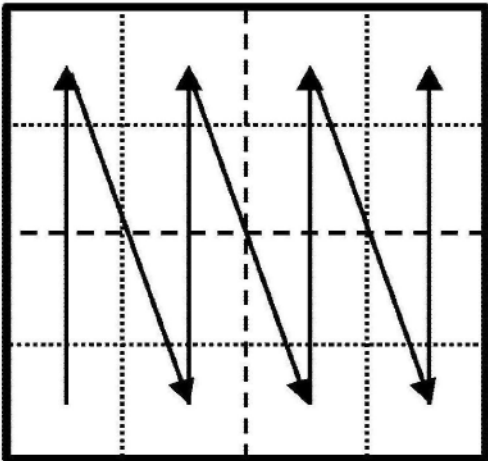
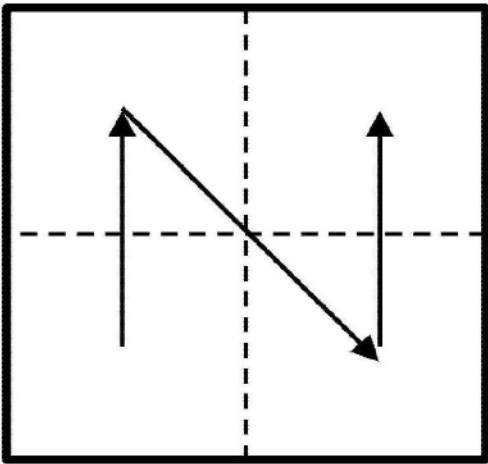


图14

1506



1504



1502



图15

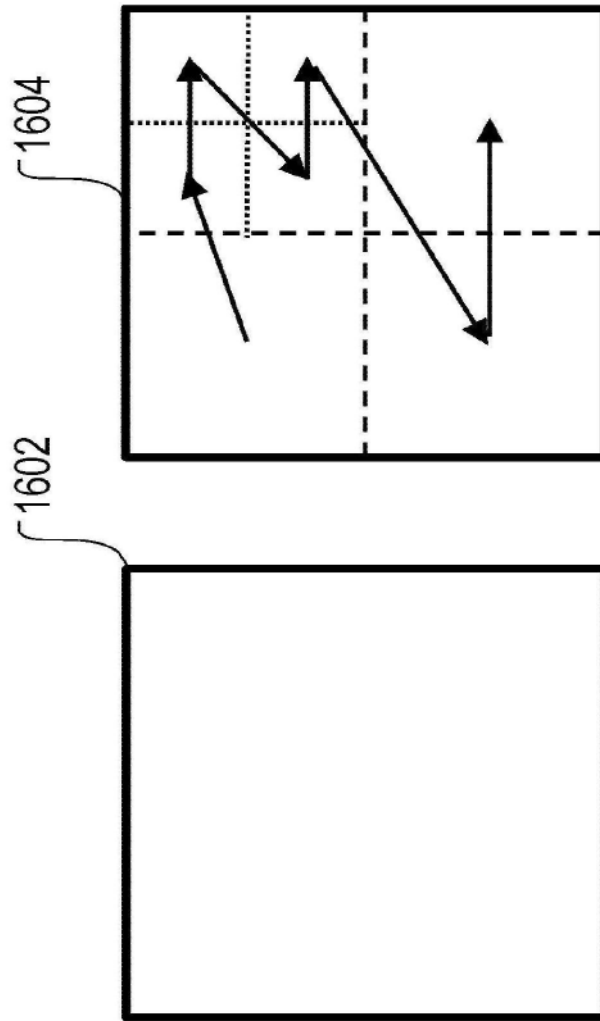


图16

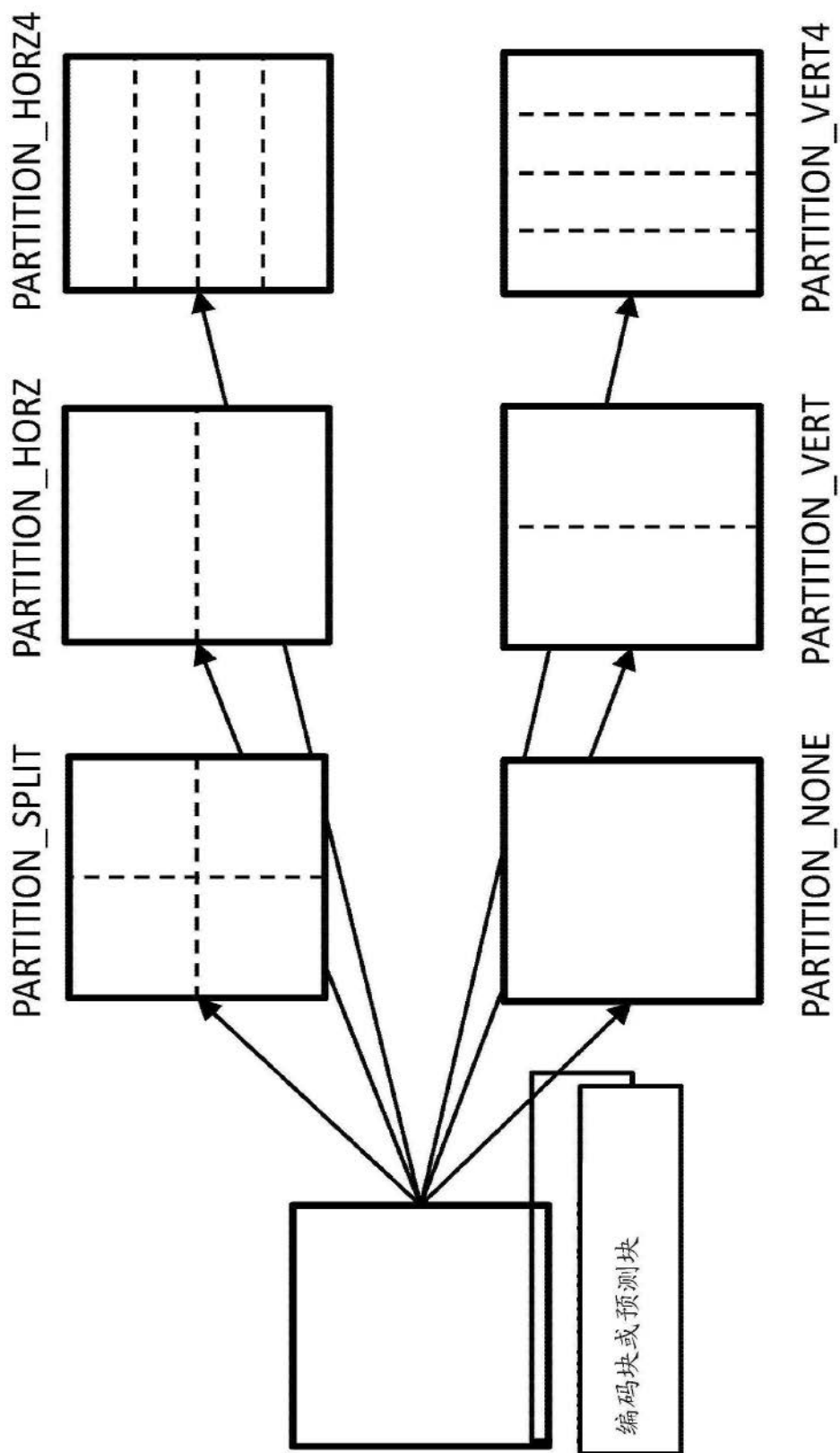


图17

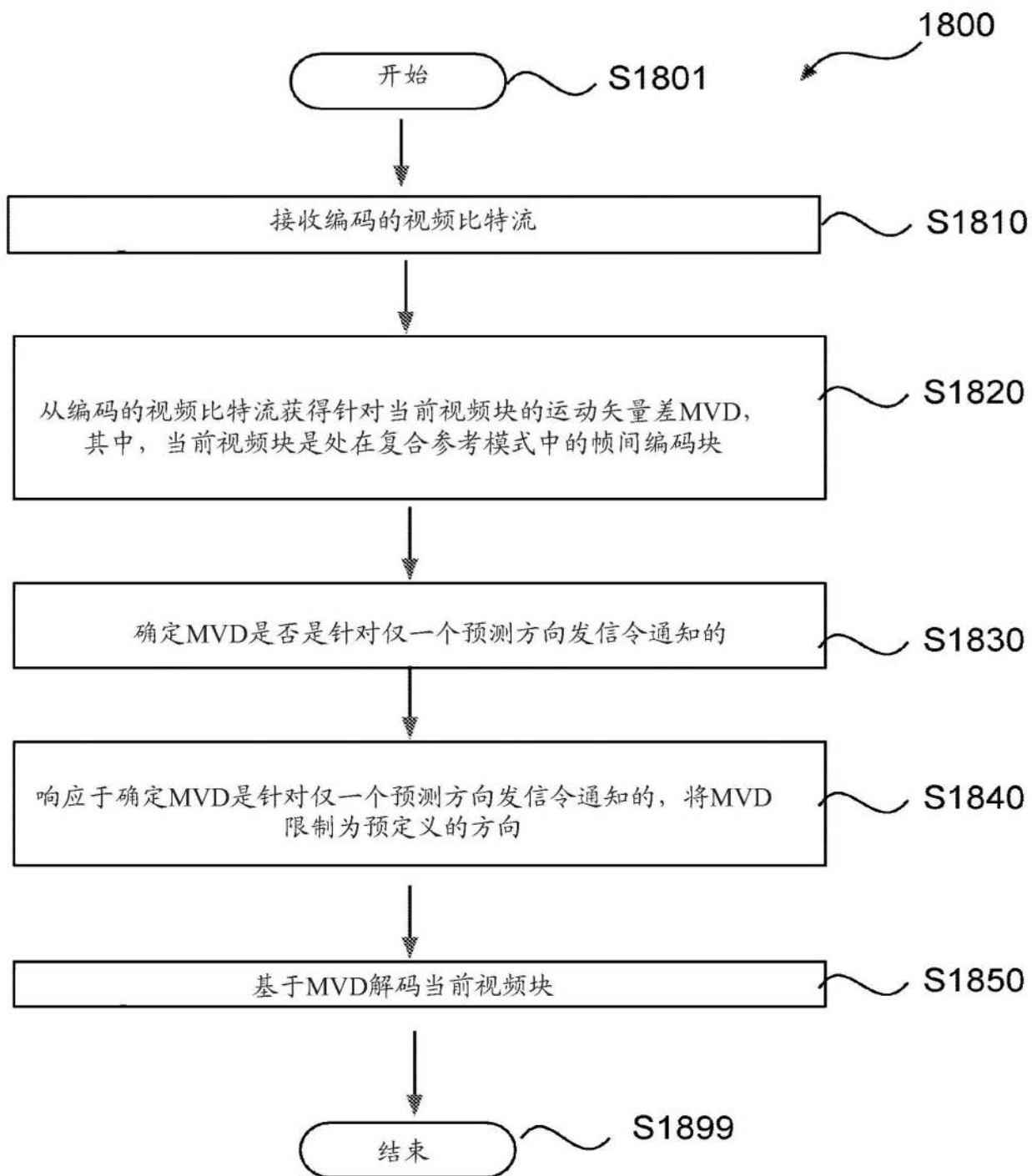


图18

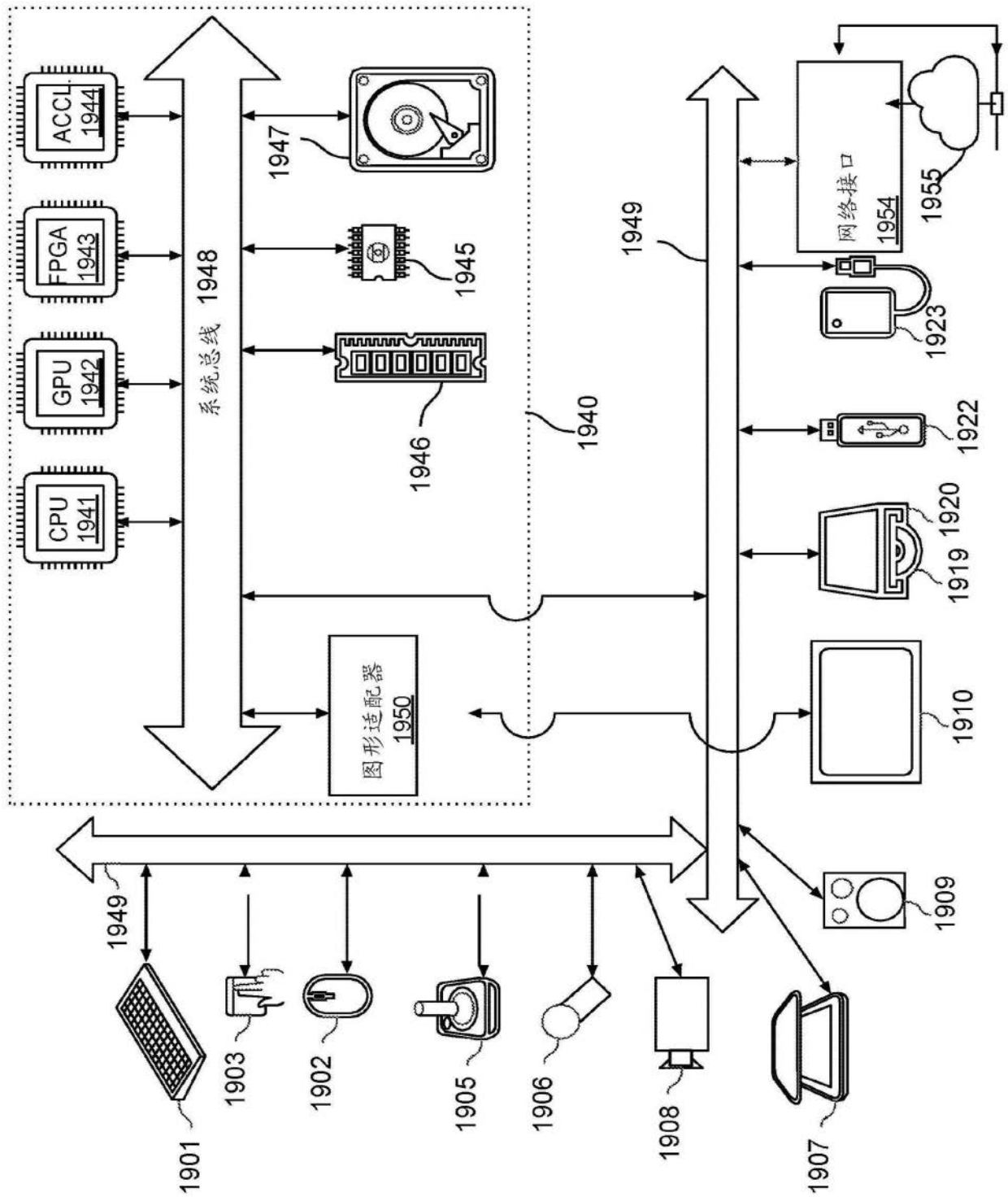


图19