



(12) 发明专利申请

(10) 申请公布号 CN 117597922 A

(43) 申请公布日 2024. 02. 23

(21) 申请号 202280046183.3

(22) 申请日 2022.06.28

(30) 优先权数据

63/215,957 2021.06.28 US

(85) PCT国际申请进入国家阶段日

2023.12.28

(86) PCT国际申请的申请数据

PCT/US2022/035375 2022.06.28

(87) PCT国际申请的公布数据

W02023/278489 EN 2023.01.05

(71) 申请人 北京达佳互联信息技术有限公司

地址 100085 北京市海淀区西二旗西路16

号院12号楼8层101

(72) 发明人 修晓宇 陈伟 郭哲玮 朱弘正

闫宁 陈漪纹 王祥林 于冰

(74) 专利代理机构 北京永新同创知识产权代理

有限公司 11376

专利代理师 刘柳 孙雪怡

(51) Int.Cl.

H04N 19/109 (2006.01)

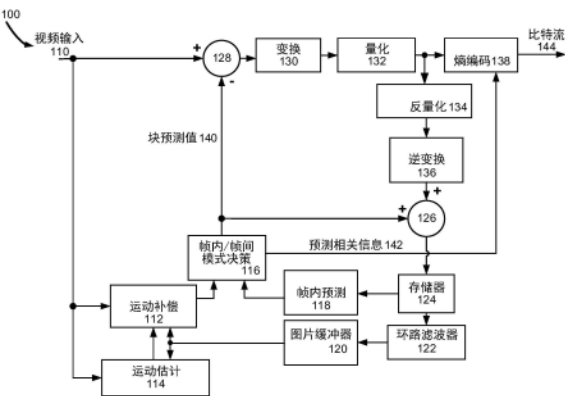
权利要求书2页 说明书40页 附图9页

(54) 发明名称

用于利用运动矢量细化的几何分区模式的方法和设备

(57) 摘要

一种用于在GPM中对视频块进行解码的方法,包括:将视频块分割成两个几何分区;通过添加常规合并候选来构建单向运动矢量(MV);响应于确定候选列表未满,通过将根据常规合并候选列表的双向预测MV得到的额外的单向MV添加到候选列表来构建第一更新的候选列表;响应于确定第一更新的候选列表未满,通过将成对平均候选添加到第一更新的候选列表来构建第二更新的候选列表;响应于确定第二更新的候选列表未满,将零单向MV周期性地添加到第二更新的候选列表,直到达到最大长度;并且分别生成用于每个几何分区的单向MV。



1. 一种用于在几何分区模式 (GPM) 中对视频块进行解码的方法, 包括:
 - 将所述视频块分割成第一几何分区和第二几何分区;
 - 通过添加多个常规合并候选来构建所述GPM的单向运动矢量 (MV) 候选列表;
 - 响应于确定所述单向MV候选列表未满, 通过将根据常规合并候选列表的一个或多个双向预测MV得到的一个或多个额外的单向MV添加到所述单向MV候选列表来构建第一更新的单向MV候选列表;
 - 响应于确定所述第一更新的单向MV候选列表未满, 通过将一个或多个成对平均候选添加到所述第一更新的单向MV候选列表来构建第二更新的单向MV候选列表;
 - 响应于确定所述第二更新的单向MV候选列表未满, 将零单向MV周期性地添加到所述第二更新的单向MV候选列表, 直到达到最大长度; 以及
 - 生成用于所述第一几何分区的单向MV和用于所述第二几何分区的单向MV。
2. 根据权利要求1所述的方法, 还包括:
 - 通过获得第一参考图片列表中的具有奇数合并索引的一个或多个MV候选和第二参考图片列表中的具有偶数合并索引的一个或多个MV来得到所述一个或多个额外的单向MV。
3. 根据权利要求2所述的方法, 其中, 所述第一参考图片列表是参考图片L0, 并且所述第二参考图片列表是L1。
4. 根据权利要求2所述的方法, 其中, 所述第一参考图片列表是参考图片L1, 并且所述第二参考图片列表是参考图片L0。
5. 根据权利要求1所述的方法, 其中, 将所述一个或多个成对平均候选添加到所述第一更新的单向MV候选列表还包括:
 - 获得第一参考图片列表或第二参考图片列表中的前两个单向MV候选; 以及
 - 响应于确定所述前两个单向MV候选指示相同的参考图片, 通过对所述前两个单向MV候选进行平均来获得成对平均候选。
6. 根据权利要求5所述的方法, 还包括:
 - 响应于确定所述前两个单向MV候选指示不同的参考图片, 通过借助对所述前两个单向MV候选进行平均来确定所述成对平均候选的幅度并且将所述第一单向MV候选的所述参考图片确定为所述成对平均候选的参考图片, 来获得所述成对平均候选。
7. 根据权利要求5所述的方法, 还包括:
 - 响应于确定所述前两个单向MV候选指示不同的参考图片, 通过借助对所述前两个单向MV候选进行平均来确定所述成对平均候选的幅度并且将所述第二单向MV候选的所述参考图片确定为所述成对平均候选的参考图片, 来获得所述成对平均候选。
8. 根据权利要求1所述的方法, 还包括:
 - 从所述单向MV候选列表中移除冗余的候选。
9. 根据权利要求8所述的方法, 还包括:
 - 响应于确定额外的单向MV等于所述单向MV候选列表中的候选, 跳过将所述额外的单向MV添加到所述单向MV候选列表。
10. 根据权利要求9所述的方法, 还包括:
 - 响应于确定所述额外的单向MV与所述单向MV候选列表中的所述候选之间的差小于MV门限, 确定所述额外的单向MV等于所述单向MV候选列表中的所述候选, 其中, 所述MV门限是

固定的门限或基于所述视频块的块大小的变量。

11. 根据权利要求5所述的方法,还包括:

响应于确定成对平均候选等于所述第一更新的单向MV候选列表中的候选,跳过将所述成对平均候选添加到所述第一更新的单向MV候选列表。

12. 根据权利要求11所述的方法,还包括:

响应于确定所述成对平均候选与所述第一更新的单向MV候选列表中的所述候选之间的差小于MV门限,确定所述成对平均候选等于所述候选,其中,所述MV门限是固定的门限或基于所述视频块的块大小的变量。

13. 一种用于视频编解码的装置,包括:

一个或多个处理器;以及

非暂时性计算机可读存储介质,其被配置为存储由所述一个或多个处理器可执行的指令;其中,所述一个或多个处理器在执行所述指令时被配置为执行根据权利要求1-12中的任一项所述的方法。

14. 一种非暂时性计算机可读存储介质,其存储计算机可执行指令,所述计算机可执行指令在由一个或多个计算机处理器执行时,使所述一个或多个计算机处理器执行根据权利要求1-12中的任一项所述的方法。

用于利用运动矢量细化的几何分区模式的方法和设备

[0001] 相关申请的交叉引用

[0002] 本申请基于并且要求于2021年6月28日提交的临时申请No.63/215,957的优先权,该临时申请的公开内容通过引用完整地并入本文,用于所有的目的。

技术领域

[0003] 本公开内容涉及视频编解码和压缩。更具体地,本公开内容涉及有关提高几何分区模式(GPM)(也被称为角度加权预测(AWP)模式)的编解码效率的方法和装置。

背景技术

[0004] 各种视频编解码技术可以被用于压缩视频数据。视频编解码是根据一个或多个视频编解码标准执行的。例如,如今,一些公知的视频编解码标准包括通用视频编解码(VVC)、高效视频编解码(HEVC,也被称为H.265或MPEG-H第2部分)和高级视频编解码(AVC,也被称为H.264或MPEG-4第10部分),它们是由ISO/IEC MPEG和ITU-T VECG联合开发的。AOMedia视频1(AV1)是由开放媒体联盟(AOM)开发的,作为其先前标准VP9的后继标准。音视频编解码(AVS)(其是指数字音频和数字视频压缩标准)是由中国音频和视频编解码标准工作组开发的另一个视频压缩标准系列。现有的视频编解码标准中的大多数视频编解码标准是建立在著名的混合视频编解码框架之上的,即,使用基于块的预测方法(例如,帧间预测、帧内预测)来减少视频图像或序列中存在的冗余,并使用变换编码来压缩预测误差的能量。视频编解码技术的一个重要目标是将视频数据压缩成使用较低比特率的形式,同时避免或最小化视频质量的下降。

发明内容

[0005] 本公开内容提供了用于视频编解码的方法和装置以及非暂时性计算机可读存储介质。

[0006] 根据本公开内容的第一方面,提供了一种用于在GPM中对视频块进行解码的方法。该方法可以包括:将视频块分割成第一几何分区和第二几何分区。该方法可以包括:通过添加多个常规合并候选来构建GPM的单向运动矢量(MV)候选列表。该方法可以包括:响应于确定单向MV候选列表未满,通过将根据常规合并候选列表中的一个或多个双向预测MV得到的一个或多个额外的单向MV添加到单向MV候选列表来构建第一更新的单向MV候选列表。

[0007] 该方法可以包括:响应于确定第一更新的单向MV候选列表未满,通过将一个或多个成对平均候选添加到第一更新的单向MV候选列表来构建第二更新的单向MV候选列表。该方法还可以包括:响应于确定第二更新的单向MV候选列表未满,将零单向MV周期性地添加到第二更新的单向MV候选列表,直到达到最大长度。该方法还可以包括:生成用于第一几何分区的单向MV和用于第二几何分区的单向MV。

[0008] 根据本公开内容的第二方面,提供了一种用于视频解码的装置。该装置可以包括一个或多个处理器以及非暂时性计算机可读存储介质。非暂时性计算机可读存储介质被配

置为存储由一个或多个处理器可执行的指令。一个或多个处理器在执行指令时被配置为执行第一方面中的方法。

[0009] 根据本公开内容的第三方面,提供了一种非暂时性计算机可读存储介质。该非暂时性计算机可读存储介质可以存储计算机可执行指令,该计算机可执行指令在由一个或多个计算机处理器执行时,使一个或多个计算机处理器执行第一方面中的方法。

附图说明

[0010] 被并入本说明书中并且构成本说明书的一部分的附图示出了与本公开内容相一致的示例,并且与本描述一起用于解释本公开内容的原理。

[0011] 图1是根据本公开内容的示例的编码器的框图。

[0012] 图2是根据本公开内容的示例的解码器的框图。

[0013] 图3A是示出根据本公开内容的示例的多类型树结构中的块分区的示意图。

[0014] 图3B是示出根据本公开内容的示例的多类型树结构中的块分区的示意图。

[0015] 图3C是示出根据本公开内容的示例的多类型树结构中的块分区的示意图。

[0016] 图3D是示出根据本公开内容的示例的多类型树结构中的块分区的示意图。

[0017] 图3E是示出根据本公开内容的示例的多类型树结构中的块分区的示意图。

[0018] 图4是根据本公开内容的示例的允许的几何分区 (GPM) 分区的图示。

[0019] 图5是示出根据本公开内容的示例的单向预测运动矢量选择的表。

[0020] 图6A是根据本公开内容的示例的运动矢量差 (MMVD) 模式的图示。

[0021] 图6B是根据本公开内容的示例的MMVD模式的图示。

[0022] 图7是根据本公开内容的示例的模板匹配 (TM) 算法的图示。

[0023] 图8是根据本公开内容的示例的在GPM中对视频块进行解码的方法。

[0024] 图9是示出根据本公开内容的示例的与用户接口耦合的计算环境的示意图。

[0025] 图10是示出根据本公开内容的一些示例的用于对视频块进行编码和解码的系统的框图。

具体实施方式

[0026] 现在将详细参照实施例,在附图中示出了实施例的示例。以下描述参考附图,其中,除非另有表示,否则不同附图中的相同的数字表示相同或相似的元素。在实施例的以下描述中阐述的实现方式并不表示与本公开内容一致的所有实现方式。相反,它们仅仅是与如所附权利要求中所记载的与本公开内容相关的方面一致的装置和方法的示例。

[0027] 本公开内容中使用的术语仅用于描述特定实施例的目的,并且不旨在限制本公开内容。如在本公开内容和所附权利要求中使用的,除非上下文另有明确指示,否则单数形式“一(a)”、“一个(an)”和“所述(the)”旨在也包括复数形式。还应理解的是,本文使用的术语“和/或”旨在表示并且包括所列出的相关联的项目中的一个或多个项目的任何或所有可能的组合。

[0028] 应当理解的是,尽管在本文中可以使用术语“第一”、“第二”、“第三”等来描述各种信息,但是该信息不应受这些术语的限制。这些术语仅用于将一类信息与另一类信息区分开来。例如,在不脱离本公开内容的范围的情况下,第一信息可以称为第二信息;并且类似

地,第二信息也可以称为第一信息。如本文中所使用的,根据上下文,术语“如果”可以理解为意指“当……时”、或“在……后”、或“响应于判断”。

[0029] 第一代AVS标准包括中国国家标准“信息技术,高级音视频编解码第2部分:视频”(被称为AVS1)和“信息技术,高级音视频编解码第16部分:广播电视视频”(被称为AVS+)。与MPEG-2标准相比,它可以在相同的感知质量下提供大约50%的比特率节省。AVS1标准视频部分于2006年2月颁布,作为中国国家标准。第二代AVS标准包括中国国家标准“信息技术,高效多媒体编解码”的系列(被称为AVS2),其主要针对于额外的HD TV节目的传输。AVS2的编解码效率是AVS+的编解码效率的两倍。在2016年5月,AVS2被发布作为中国国家标准。同时,电气和电子工程师协会(IEEE)将AVS2标准视频部分提交为一项国际标准予以应用。AVS3标准是针对UHD视频应用的新一代视频编解码标准,其旨在超越最新国际标准HEVC的编解码效率。2019年3月,在第68次AVS会议上完成了AVS3-P2基线,与HEVC标准相比,它提供了大约30%的比特率节省。目前,存在一个被称为高性能模型(HPM)的参考软件,其由AVS小组维护,以论证AVS3标准的参考实现方式。

[0030] 与HEVC一样,AVS3标准建立在基于块的混合视频编解码框架之上。

[0031] 图10是示出根据本公开内容的一些实现方式的用于并行地对视频块进行编码和解码的示例性系统10的框图。如图10中所示,系统10包括源设备12,源设备12生成并编码稍后将由目标设备14进行解码的视频数据。源设备12和目标设备14可以包括各种各样的电子设备中的任何电子设备,包括台式计算机或膝上型计算机、平板计算机、智能电话、机顶盒、数字电视、相机、显示器设备、数字媒体播放器、视频游戏机、视频流式传输设备等。在一些实现方式中,源设备12和目标设备14被配备有无线通信能力。

[0032] 在一些实现方式中,目标设备14可以经由链路16接收待被解码的经编码视频数据。链路16可以包括能够将经编码视频数据从源设备12移动到目标设备14的任何类型的通信介质或设备。在一个示例中,链路16可以包括使源设备12能够实时地将经编码视频数据直接发送到目标设备14的通信介质。经编码视频数据可以根据通信标准(例如无线通信协议)被调制,并且被发送到目标设备14。通信介质可以包括任何无线或有线通信介质,例如,射频(RF)频谱或者一个或多个物理传输线。通信介质可以形成基于分组的网络(例如,局域网、广域网或诸如互联网的全球网络)的一部分。通信介质可以包括路由器、交换机、基站或可以有益于促进从源设备12到目标设备14的通信的任何其他设备。

[0033] 在一些其它实现方式中,可以将经编码视频数据从输出接口22发送到存储设备32。随后,可以由目标设备14经由输入接口28访问存储设备32中的经编码视频数据。存储设备32可以包括各种分布式或本地访问的数据存储介质中的任何数据存储介质,例如,硬盘驱动器、蓝光光盘、数字多功能光盘(DVD)、压缩光盘只读存储器(CD-ROM)、闪存、易失性或非易失性存储器、或者用于存储经编码视频数据的任何其它合适的数字存储介质。在另一示例中,存储设备32可以对应于文件服务器或可以保持由源设备12生成的经编码视频数据的另一中间存储设备。目标设备14可以经由流式传输或下载从存储设备32访问被存储的视频数据。文件服务器可以是能够存储经编码视频数据并且将经编码视频数据发送到目标设备14的任何类型的计算机。示例性文件服务器包括网络服务器(例如,用于网站)、文件传输协议(FTP)服务器、网络附加存储(NAS)设备或本地磁盘驱动器。目标设备14可以通过适合于访问被存储在文件服务器上的经编码视频数据的任何标准数据连接来访问经编码视频

数据,标准数据连接包括无线信道(例如,无线保真(Wi-Fi)连接)、有线连接(例如,数字订户线(DSL)、电缆调制解调器等)或这两者的组合。经编码视频数据从存储设备32的传输可以是流式传输、下载传输或这两者的组合。

[0034] 如图10中所示,源设备12包括视频源18、视频编码器20和输出接口22。视频源18可以包括诸如以下项的源或这样的源的组合:视频捕获设备(例如,摄像机)、包含先前捕获的视频的视频存档、用于从视频内容提供者接收视频的视频馈入接口、和/或用于生成作为源视频的计算机图形数据的计算机图形系统。作为一个示例,如果视频源18是安全监控系统的摄像机,则源设备12和目标设备14可以形成相机电话或视频电话。然而,本申请中所描述的实现方式通常可以适用于视频编解码,并且可以被应用于无线和/或有线应用。

[0035] 可以由视频编码器20对经捕获、预先捕获或计算机生成的视频进行编码。可以经由源设备12的输出接口22将经编码视频数据直接发送到目标设备14。还可以(或替代地)将经编码视频数据存储到存储设备32上以便稍后由目标设备14或其它设备访问,以用于解码和/或回放。输出接口22可以进一步包括调制解调器和/或发射机。

[0036] 目标设备14包括输入接口28、视频解码器30和显示器设备34。输入接口28可以包括接收机和/或调制解调器,并且通过链路16接收经编码视频数据。通过链路16发送的或在存储设备32上提供的经编码视频数据可以包括由视频编码器20生成的各种语法元素以供视频解码器30在对视频数据进行解码时使用。这样的语法元素可以被包括在通信介质上发送的、被存储在存储介质上的或被存储在文件服务器上的经编码视频数据内。

[0037] 在一些实现方式中,目标设备14可以包括显示器设备34,显示器设备34可以是集成显示器设备和被配置为与目标设备14通信的外部显示器设备。显示器设备34向用户显示经解码视频数据,并且可以包括各种显示器设备中的任何显示器设备,诸如液晶显示器(LCD)、等离子显示器、有机发光二极管(OLED)显示器或另一类型的显示器设备。

[0038] 视频编码器20和视频解码器30可以根据专有标准或行业标准(例如,VVC、HEVC、MPEG-4的第10部分、AVC)或这样的标准的扩展进行操作。应当理解的是,本申请不限于特定的视频编码/解码标准,并且可以适用于其它视频编码/解码标准。通常认为源设备12的视频编码器20可以被配置为根据这些当前标准或未来标准中的任何标准对视频数据进行编码。类似地,通常还认为目标设备14的视频解码器30可以被配置为根据这些当前标准或未来标准中的任何标准对视频数据进行解码。

[0039] 视频编码器20和视频解码器30可以各自被实现为各种合适的编码器和/或解码器电路中的任何电路,例如,一个或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、分立逻辑、软件、硬件、固件或其任何组合。当部分地以软件实现时,电子设备可以将用于软件的指令存储于合适的非暂时性计算机可读介质中,并且使用一个或多个处理器执行硬件中的指令以执行本公开内容中所公开的视频编码/解码操作。视频编码器20和视频解码器30中的每一个可以被包括在一个或多个编码器或解码器中,编码器或解码器中的任一者可以被集成为相应设备中的组合式编码器/解码器(CODEC)的一部分。

[0040] 图1示出用于VVC的基于块的视频编码器的一般示意图。具体地,图1示出典型的编码器100。编码器100可以是如图10中所示的视频编码器20。编码器100具有视频输入110、运动补偿112、运动估计114、帧内/帧间模式决策116、块预测值140、加法器128、变换130、量化

132、预测相关信息142、帧内预测118、图片缓冲器120、反量化134、逆变换136、加法器126、存储器124、环路滤波器122、熵编码138和比特流144。

[0041] 在编码器100中,视频帧被分割成多个视频块,以用于处理。对于每个给定的视频块,基于帧间预测方法或帧内预测方法形成预测。

[0042] 表示当前视频块(视频输入110的一部分)与其预测值(块预测值140的一部分)之间的差的预测残差从加法器128被发送到变换130。然后,变换系数从变换130被发送到量化132,以用于熵减小。然后,经量化系数被馈送到熵编码138,以生成经压缩视频比特流。如图1中所示,来自帧内/帧间模式决策116的预测相关信息142(例如,视频块分区信息、运动矢量(MV)、参考图片索引和帧内预测模式)也被馈送通过熵编码138,并且被保存到经压缩比特流144中。经压缩比特流144包括视频比特流。

[0043] 在编码器100中,还需要解码器相关电路系统,以便重建用于预测目的的像素。首先,通过反量化134和逆变换136重建预测残差。该经重建预测残差与块预测值140组合,以为当前视频块生成未经滤波的重建像素。

[0044] 空间预测(或“帧内预测”)使用来自在与当前视频块相同的视频帧中的已编码的相邻块的样点(其被称为参考样点)的像素来预测当前视频块。

[0045] 时间预测(也被称为“帧间预测”)使用来自已编码的视频图片的经重建像素来预测当前视频块。时间预测减少了视频信号中固有的时间冗余。通常通过一个或多个MV发信号通知针对给定编码单元(CU)或编码块的时间预测信号,MV指示当前CU与其时间参考之间的运动的量和方向。此外,如果支持多个参考图片,则另外发送一个参考图片索引,该参考图片索引用于标识时间预测信号来自参考图片存储中的哪个参考图片。

[0046] 运动估计114接收视频输入110和来自图片缓冲器120的信号,并且将运动估计信号输出到运动补偿112。运动补偿112接收视频输入110、来自图片缓冲器120的信号以及来自运动估计114的运动估计信号,并且将运动补偿信号输出到帧内/帧间模式决策116。

[0047] 在执行空间和/或时间预测之后,编码器100中的帧内/帧间模式决策116例如基于率失真优化方法来选择最佳预测模式。然后,从当前视频块中减去块预测值140,并且使用变换130和量化132对得到的预测残差进行去相关。得到的经量化残差系数由反量化134进行反量化并由逆变换136进行逆变换,以形成经重建残差,然后将经重建残差添加回预测块以形成CU的经重建信号。此外,在经重建CU被放入图片缓冲器120的参考图片存储中并用于对未来的视频块进行编码之前,可以将环路滤波122(例如,去块滤波器、样点自适应偏移(SAO)和/或自适应环路滤波器(ALF))应用于经重建CU上。为了形成输出视频比特流144,编码模式(帧间或帧内)、预测模式信息、运动信息和经量化残差系数均被发送到熵编码单元138,以被进一步压缩和打包以形成比特流。

[0048] 图1给出通用的基于块的混合视频编码系统的框图。输入视频信号被逐块(被称为编码单元(CU))处理。与仅基于四叉树对块进行分割的HEVC不同,在AVS3中,一个编码树单元(CTU)被划分成CU,以适应基于四叉树/二叉树/扩展四叉树的不同局部特性。另外,移除了HEVC中多分区单元类型的概念,即,CU、预测单元(PU)和变换单元(TU)的分离不存在于AVS3中;相反,每个CU始终被用作预测和变换两者的基本单元,而不进行进一步分区。在AVS3的树分区结构中,首先基于四叉树结构对一个CTU进行分割。然后,每个四叉树叶节点可以基于二元和扩展四叉树结构被进一步分割。

[0049] 如图3A、3B、3C、3D和3E中所示,存在五种划分类型,四元分区、水平二元分区、垂直二元分区、水平扩展四叉树分区和垂直扩展四叉树分区。

[0050] 图3A示出说明根据本公开内容的多类型树结构中的块四元分区的示意图。

[0051] 图3B示出说明根据本公开内容的多类型树结构中的块垂直二元分区的示意图。

[0052] 图3C示出说明根据本公开内容的多类型树结构中的块水平二元分区的示意图。

[0053] 图3D示出说明根据本公开内容的多类型树结构中的块垂直三元分区的示意图。

[0054] 图3E示出说明根据本公开内容的多类型树结构中的块水平三元分区的示意图。

[0055] 在图1中,可以执行空间预测和/或时间预测。空间预测(或“帧内预测”)使用来自同一视频图片/条带中的已编码的相邻块的样点(其被称为参考样点)的像素来预测当前视频块。空间预测减少了视频信号中固有的空间冗余。时间预测(也被称为“帧间预测”或“运动补偿预测”)使用来自已编码的视频图片的经重建像素来预测当前视频块。时间预测减少了视频信号中固有的时间冗余。通常通过一个或多个运动矢量(MV)发信号通知针对给定CU的时间预测信号,MV指示当前CU与其时间参考之间的运动的量和方向。而且,如果支持多个参考图片,则另外发送一个参考图片索引,该参考图片索引用于标识时间预测信号来自参考图片存储中的哪个参考图片。在空间和/或时间预测之后,编码器中的模式决策块例如基于率失真优化方法来选择最佳预测模式。然后,从当前视频块中减去预测块;并且使用变换对预测残差进行去相关,并且然后进行量化。经量化残差系数被反量化和逆变换,以形成经重建残差,然后将经重建残差添加回预测块以形成CU的经重建信号。此外,在经重建CU被放入参考图片存储中并被用作参考以对未来的视频块进行编码之前,可以将环路滤波(例如,去块滤波器、样点自适应偏移(SAO)和/或自适应环路滤波器(ALF))应用于经重建CU上。为了形成输出视频比特流,编码模式(帧间或帧内)、预测模式信息、运动信息和量化残差系数均被发送到熵编码单元,以被进一步压缩和打包。

[0056] 图2示出用于VVC的视频解码器的一般框图。具体地,图2示出典型的编码器200框图。基于块的视频解码器200可以是如图10中所示的视频解码器30。解码器200具有比特流210、熵解码212、反量化214、逆变换216、加法器218、帧内/帧间模式选择220、帧内预测222、存储器230、环路滤波器228、运动补偿224、图片缓冲器226、预测相关信息234和视频输出232。

[0057] 解码器200类似于位于图1的编码器100中的重建相关部分。在解码器200中,传入的视频比特流210首先通过熵解码212进行解码,以得到经量化系数级和预测相关信息。然后通过反量化214和逆变换216处理经量化系数级,以获得经重建预测残差。在帧内/帧间模式选择器220中实现的块预测值机制被配置为基于经解码的预测信息来执行帧内预测222或运动补偿224。通过使用求和器218将来自逆变换216的经重建预测残差与由块预测值机制生成的预测输出相加来获得一组未经滤波的重建像素。

[0058] 经重建块在其被存储在用作参考图片存储的图片缓冲器226中之前可以进一步经过环路滤波器228。图片缓冲器226中的经重建视频可以被发送以驱动显示设备,以及被用于预测未来的视频块。在其中环路滤波器228被开启的情况下,对这些经重建像素执行滤波操作以得到最终经重建视频输出232。

[0059] 图2给出基于块的视频解码器的一般框图。视频比特流首先在熵解码单元处进行熵解码。编码模式和预测信息被发送到空间预测单元(如果是帧内编码的话)或时间预测单

元(如果是帧间编码的话),以形成预测块。残差变换系数被发送到反量化单元和逆变换单元以重建残差块。然后将预测块和残差块相加在一起。经重建块在其被存储在参考图片存储中之前可以进一步经过环路滤波。然后将参考图片存储中的经重建视频发送出去以供显示,并且被用于预测未来的视频块。

[0060] 本公开内容的重点在于提高在VVC和AVS3标准两者中使用的几何分区模式(GPM)的编解码性能。在AVS3中,该工具也被称为角度加权预测(AWP),其遵循GPM的相同设计精神,但是在某些设计细节中有一些细微差别。为了促进本公开内容的描述,在下文中,将VVC标准中的现有GPM设计用作示例来解释GPM/AWP工具的主要方面。同时,考虑到被称为被应用在VVC和AVS3标准两者中的利用运动矢量差的合并模式(MMVD)的另一现有帧间预测技术与本公开内容中所提出的技术密切相关,因此也对其进行了简要回顾。之后,识别了当前GPM/AWP设计的一些缺点。最后,详细地提供了所提出的方法。请注意的是,虽然贯穿本公开内容,VVC标准中的现有GPM标准被用作示例,但是对于现代视频编解码技术的领域的技术人员而言,所提出的技术也可以被应用于其它GPM/AWP设计或者采用相同或相似设计精神的其它编解码工具。

[0061] 几何分区模式(GPM)

[0062] 在VVC中,针对帧间预测支持几何分区模式。通过一个CU级别标志,来发信号通知作为一种特殊合并模式的几何分区模式。在当前的GPM设计中,GPM模式对于具有宽度和高度两者不小于8且不大于64(不包括 8×64 和 64×8)的每个可能的CU大小总共支持64个分区。

[0063] 如图4中所示(下文提供的描述),当使用该模式时,CU被几何定位的直线划分成两个部分。划分线的位置在数学上是根据特定分区的角度和偏移参数得到的。CU中的几何分区的每个部分是使用其自己的运动进行帧间预测的;针对每个分区,只允许单向预测,即,每个部分具有一个运动矢量和一个参考索引。应用单向预测运动约束,以确保其与传统的双向预测相同,针对每个CU只需要两个运动补偿预测。如果将几何分区模式用于当前CU,则进一步发信号通知指示几何分区的分区模式(角度和偏移)的几何分区索引和两个合并索引(针对每个分区一个合并索引)。在序列级别上,显式地发信号通知最大GPM候选大小的数量。

[0064] 图4示出允许的GPM分区,其中,每个图片中的划分具有一个相同的划分方向。

[0065] 单向预测候选列表构建

[0066] 为了得到针对一个几何分区的单向预测运动矢量,首先根据常规合并候选列表生成过程直接得到一个单向预测候选列表。将n表示为几何单向预测候选列表中的单向预测运动的索引。第n个合并候选的LX运动矢量(其中X等于n的奇偶校验)被作用于几何分区模式的第n个单向预测运动矢量。

[0067] 在图5中,这些运动矢量用“x”标记(在下文描述)。假使第n个扩展合并候选的对应LX运动矢量不存在,取而代之地使用相同候选的L(1-X)运动矢量作为用于几何分区模式的单向预测运动矢量。

[0068] 图5示出来自用于GPM的合并候选列表的运动矢量的单向预测运动矢量选择。

[0069] 沿几何分区边缘的混合

[0070] 在使用其自己的运动获得每个几何分区之后,将混合应用于两个单向预测信号,

以得到几何分区边缘周围的样点。用于CU的每个位置的混合权重是基于从每个单独的样点位置到对应的分区边缘的距离得到的。

[0071] GPM信令设计

[0072] 根据当前GPM设计,对GPM的使用是通过在CU级别上发信号通知一个标志来指示的。仅在当前CU是通过合并模式或跳过模式编码时才发信号通知标志。具体地,当标志等于一时,其指示当前CU是通过GPM预测的。否则(标志等于零),CU是通过另一合并模式(例如,常规合并模式、利用运动矢量差的合并模式、组合的帧间和帧内预测等)编码的。当针对当前CU启用GPM时,进一步发信号通知一个语法元素,即,merge_gpm_partition_idx,以指示所应用的几何分区模式(其指定将CU划分成两个分区(如图4中所示)的直线的方向及其与CU中心的偏移)。之后,发信号通知两个语法元素merge_gpm_idx0和merge_gpm_idx1,以指示被用于第一GPM分区和第二GPM分区的单向预测合并候选的索引。更具体地,这两个语法元素被用于从如在部分“单向预测合并列表构建”中所描述的单向预测合并列表确定两个GPM分区的单向MV。根据当前GPM设计,为了使两个单向MV更加不同,两个索引不能是相同的。基于这样的先验知识,首先发信号通知第一GPM分区的单向预测合并索引,并且将其用作预测值来减小第二GPM分区的单向预测合并索引的信令开销。详细地,如果第二单向预测合并索引小于第一单向预测合并索引,则直接发信号通知其原始值。否则(第二单向预测合并索引大于第一单向预测合并索引),在发信号通知给比特流之前,将其值减去一。在解码器侧,首先对第一单向预测合并索引进行解码。然而,对于第二单向预测合并索引的解码,如果所解析的值小于第一单向预测合并索引,则将第二单向预测合并索引设置为等于所解析的值;否则(所解析的值等于或大于第一单向预测合并索引),将第二单向预测合并索引设置为等于所解析的值加一。表1示出在当前VVC规范中被用于GPM模式的现有语法元素。

[0073] 表1VVC规范的合并数据语法表中的现有GPM语法元素

[0074]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	merge_gpm_idx0[x0][y0]	ae(v)
	if(MaxNumGpmMergeCand > 2)	
	merge_gpm_idx1[x0][y0]	ae(v)
	}	
		
	}	

[0075] 另一方面,在当前GPM设计中,将截断的一元编码用于两个单向预测合并索引(即,merge_gpm_idx0和merge_gpm_idx1)的二值化。另外,因为这两个单向预测合并索引不能是相同的,所以将不同的最大值用于截断这两个单向预测合并索引的码字,对于merge_gpm_idx0和merge_gpm_idx1,不同的最大值分别被设置为等于MaxGPMMergeCand-1和MaxGPMMergeCand-2。MaxGPMMergeCand是单向预测合并列表中的候选的数量。在解码器侧,当接收的merge_gpm_idx1的值等于或大于merge_gpm_idx0的值时,考虑到merge_gpm_idx0和merge_gpm_idx1的值不能是相同的,merge_gpm_idx1的值将会增加1。

[0076] 当应用GPM/AWP模式时,应用两种不同的二值化方法来将语法merge_gpm_

partition_idx转化成一串二进制比特。具体地,分别通过VVC和AVS3标准中的固定长度编码和截断二进制编码对语法元素进行二值化。同时,对于AVS3中的AWP模式,不同的最大值被用于语法元素的值的二值化。具体地,在AVS3中,允许的GPM/AWP分区模式的数量是56(即,merge_gpm_partition_idx的最大值是55),而在VVC中,该值被增加到64(即,merge_gpm_partition_idx的最大值是63)。

[0077] 利用运动矢量差的合并模式(MMVD)

[0078] 除了根据其空间/时间邻居得到一个当前块的运动信息的传统合并模式外,在VVC和AVS标准两者中引入MMVD/UMVE模式,作为一种特殊的合并模式。具体地,在VVC和AVS3两者中,在编码块级别上,通过一个MMVD标志发信号通知该模式。在MMVD模式中,选择合并列表中的针对常规合并模式的前两个候选,作为用于MMVD的两个基本合并候选。在选择并发信号通知一个基本合并候选之后,发信号通知额外的语法元素,以指示被添加到所选择的合并候选的运动的运动矢量差(MVD)。MMVD语法元素包括用于选择基本合并候选的合并候选标志、用于指定MVD幅度的距离索引以及用于指示MVD方向的方向索引。

[0079] 在现有MMVD设计中,距离索引指定MVD幅度,其是基于与起点的一组预定义的偏移来定义的。如图6A和6B中所示,该偏移被添加到起始MV的水平或垂直分量(即,所选择的基本合并候选的MV)。

[0080] 图6A示出用于L0参考的MMVD模式。图6B示出用于L1参考的MMVD模式。

[0081] 表2示出分别在AVS3中应用的MVD偏移。

[0082] 表2 AVS3中使用的MVD偏移

[0083]	距离IDX	0	1	2	3	4
	偏移(以亮度样点为单位)	1/4	1/2	1	2	4

[0084] 如表3中所示,方向索引被用于指定发信号通知的MVD的符号。应注意的是,MVD符号的含义可以根据起始MV而变化。当起始MV是单向预测MV或双向预测MV(其中MV指向其POC均大于当前图片的POC,或者两者小于当前图片的POC的两个参考图片)时,发信号通知的符号是与起始MV相加的MVD的符号。当起始MV是指向两个参考图片(其中一个图片的POC大于当前图片并且另一图片的POC小于当前图片)的双向预测MV时,将发信号通知的符号应用到L0 MVD,并且将发信号通知的符号的相反值应用到L1 MVD。

[0085] 表3如由方向索引指定的MVD符号

[0086]	方向IDX	00	01	10	11
	x轴	+	-	N/A	N/A
	y轴	N/A	N/A	+	-

[0087] 用于常规帧间模式的运动信令

[0088] 与HEVC标准类似,除了合并/跳过模式外,VVC和AVS3两者允许一个帧间CU显式地在比特流中指定其运动信息。总体而言,VVC和AVS3两者中的运动信息信令与在HEVC标准中的运动信息信令保持相同。具体地,首先发信号通知一个帧间预测语法(即,inter_pred_idc),以指示预测信号来自列表L0、L1还是两者。对于每个使用的参考列表,通过发信号通知针对对应参考列表的一个参考图片索引ref_idx_lx(x=0,1)来标识对应参考图片,并且对应MV由被用于选择MV预测值(MVP),接着是在目标MV与所选择的MVP之间的其运动矢量差(MVD)的一个MVP索引mvp_lx_flag(x=0,1)来表示。另外,在VVC标准中,在条带级别上,发

信号通知一个控制标志mvd_l1_zero_flag。当mvd_l1_zero_flag等于0时,在比特流中发信号通知L1 MVD;否则(当mvd_l1_zero_flag标志等于1时),不发信号通知L1 MVD,并且在编码器和解码器处,其值始终被推断为零。

[0089] 利用CU级别权重的双向预测

[0090] 在VVC和AVS3之前的先前标准中,当不应用加权预测(WP)时,通过对从两个参考图片获得的单向预测信号进行平均来生成双向预测信号。在VVC中,引入一种工具编码(即,利用CU级别权重的双向预测(BCW)),以提高双向预测的效率。具体地,代替简单的平均,通过允许对两个预测信号的加权平均来扩展BCW中的双向预测,如被描述为:

[0091]
$$P'(i, j) = ((8-w) \cdot P_0(i, j) + w \cdot P_1(i, j) + 4) >> 3$$

[0092] 在VVC中,当当前图片是一个低延迟图片时,允许从一组预定义的权重值 $w \in \{-2, 3, 4, 5, 10\}$ 中选择一个BCW编码块的权重,并且为4的权重表示其中两个单向预测信号被相等加权的传统双向预测情况。对于低延迟,仅允许3个权重 $w \in \{3, 4, 5\}$ 。一般而言,虽然在WP与BCW之间存在一些设计相似性,但是这两种编码工具针对解决不同粒度的照度改变问题。然而,因为WP与BCW之间的交互可能潜在地使VVC设计复杂化,所以这两种工具不允许被同时启用。具体地,当针对一个条带启用WP时,则不发信号通知针对该条带中的所有双向预测CU的BCW权重,并且BCW权重被推断为4(即,应用相等的权重)。

[0093] 模板匹配

[0094] 模板匹配(TM)是解码器侧MV推导方法,其通过找到由当前CU的顶部和左侧相邻经重建样点组成的一个模板与参考图片中的参考块(即,与模板相同大小)之间的最佳匹配来细化当前CU的运动信息。如图7中所示,将在 $[-8, +8]$ 图像元素(pe1)搜索范围内,在当前CU的初始运动矢量周围搜索一个MV。最佳匹配可以被定义为在当前模板与参考模板之间实现最低匹配成本(例如,绝对差之和(SAD)、绝对变换差之和(SATD)等)的MV。存在两种不同的方式来应用TM模式用于帧间编解码:

[0095] 在AMVP模式中,基于模板匹配差来确定MVP候选,以拾取在当前块模板与参考块模板之间达到最小差的MVP候选,并且然后TM仅针对该特定MVP候选执行MV细化。TM对该MVP候选进行细化,通过使用迭代菱形搜索,从在 $[-8, +8]$ 图像元素搜索范围内的完整图像元素MVD精度(或者对于4图像元素AMVR模式的4图像元素)开始。可以通过使用具有完整图像元素MVD精度(或针对4图像元素AMVR模式的4图像元素)、继续地接着半图像元素和四分之一图像元素MVD精度(取决于如下面的表13中指定的AMVP模式)的交叉搜索,来进一步细化AMVR候选。该搜索过程确保MVP候选仍然保持与在TM过程之后由AMVR模式指示的相同的MV精度。

[0096] 表14

搜索模式	AMVR 模式				合并模式	
	4 图像元素	完整图像元素	半图像元素	四分之一图像元素	AltIF=0	AltIF=1
4 图像元素菱形搜索	v					
4 图像元素交叉搜索	v					
完整图像元素菱形搜索		v	v	v	v	v
完整图像元素交叉搜索		v	v	v	v	v
半图像元素交叉搜索			v	v	v	v
四分之一图像元素交叉搜索				v	v	
1/8 图像元素交叉搜索					v	

[0097] 在合并模式中,将类似的搜索方法应用于由合并索引指示的合并候选。如上表中所示,TM可以一直向下执行到1/8图像元素MVD精度或跳过超过半图像元素MVD精度的那些MVD精度,这取决于是否根据合并运动信息使用替代内插滤波器(当AMVR是半图像元素模式时使用)。

[0099] 如上所提及的,从常规合并候选直接获得用于生成两个GPM分区的预测样点的单向运动。在空间/时间相邻块的MV之间不存在强相关的情况下,从合并候选得到的单向MV可能不足够准确以捕获每个GPM分区的真实运动。运动估计能够提供更准确的运动,然而更准确的运动出现的代价是由于可以被应用在现有单向MV之上的任意运动细化所导致的不可忽视的信令开销。另一方面,在VVC和AVS3标准两者中利用MVMD模式,MVMD模式已经被证明是降低MVD信令开销的一种高效信令机制。因此,其也可能有益于将GPM与MMVD模式进行组合。这样的组合通过提供更准确的MV来捕获每个GPM分区的单独运动,能够潜在地提高GPM工具的总体编解码效率。

[0100] 如前所讨论的,在VVC和AVS3标准两者中,GPM模式仅被应用于合并/跳过模式。考虑到所有非合并帧间CU不能从GPM的灵活的非矩形分区获益,这样的设计在编解码效率方面可能不是最优的。另一方面,由于与上文提及的相同的原因,根据常规合并/跳过模式得到的单向预测运动候选并非总是精确的以捕获两个几何分区的真实运动。基于这样的分析,通过GPM模式到非合并帧间模式(即,在比特流中显式地发信号通知其运动信息的CU)的合理扩展,可以预期额外的编码增益。然而,MV准确性的提高出现的代价是增加的信令开销。因此,为了将GPM模式高效地应用于显式帧间模式,重要的是识别能够最小化信令成本同时为两个几何分区提供更准确的MV的一种有效信令方案。

[0101] 所提出的方法

[0102] 在本公开内容中,提出了用于通过将进一步运动细化应用在现有的单向MV(其被

应用于每个GPM分区)之上来进一步提高GPM的编解码效率的方法。所提出的方法被称为利用运动矢量细化的几何分区模式(GPM-MVR)。另外,在所提出的方案中,以现有的MMVD设计的相似方式,即,基于运动细化的一组预定义的MVD幅度和方向,发信号通知运动细化。

[0103] 在本公开内容的另一方面中,提供了用于将GPM模式扩展到显式帧间模式的解决方案。为了便于描述,这些方案被称为利用显式运动信令的几何分区模式(GPM-EMS)。具体地,为了实现与常规帧间模式的更佳的协调,在所提出的GPM-EMS方案中采用现有的运动信令机制(即,MVP加MVD),以指定两个几何分区的对应单向MV。

[0104] 利用单独的运动矢量细化的几何分区模式

[0105] 为了提高GPM的编解码效率,在本部分中,提出了一种利用单独的运动矢量细化的改进的几何分区模式。具体地,考虑到GPM分区,所提出的方法首先使用现有的语法merge_gpm_idx0和merge_gpm_idx1来从现有的单向预测合并候选列表中识别用于两个GPM分区的单向MV,并且将它们用作基本MV。在确定了这两个基本MV之后,引入两组新语法元素,以指定分开地被应用在两个GPM分区的基本MV之上的运动细化的值。具体地,首先发信号通知两个标志(即,gpm_mvr_partIdx0_enable_flag和gpm_mvr_partIdx1_enable_flag),以指示是否将GPM-MVR分别应用于第一GPM分区和第二GPM分区。当一个GPM分区的标志等于一时,以MMVD方式发信号通知被应用于该分区的基本MV的MVR的对应值,即,用于指定MVR的幅度的一个距离索引(如由语法元素gpm_mvr_partIdx0_distance_idx和gpm_mvr_partIdx1_distance_idx指示的)和用于指定MVR的方向的一个方向索引(如由语法元素gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx1_distance_idx指示的)。表4示出由所提出的GPM-MVR方法引入的语法元素。

[0106] 表4所提出的将单独的MVR用于两个GPM分区的GPM-MVR方法的语法元素(方法一)

[0107]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
	gpm_mvr_partIdx0_enable_flag[x0][y0]	ae(v)
	if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx0_directoin_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx0_distance_idx[x0][y0]	ae(v)
[0108]	}	
	if(merge_gpm_idx0 != merge_gpm_idx1	
	gpm_mvr_partIdx0_enable_flag)	
	gpm_mvr_partIdx1_enable_flag[x0][y0]	ae(v)
	if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx1_direction_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx1_distance_idx[x0][y0]	ae(v)
	}	
	}	
		
	}	

[0109] 基于如表4中所示的所提出的语法元素,在解码器处,被用于生成每个GPM分区的单向预测样点的最终MV等于发信号通知的运动矢量细化与对应基本MV之和。实际上,可以预定义不同组的MVR幅度和方向,并且将其应用于所提出的GPM-MVR方案,这能够在运动矢量准确性与信令开销之间提供各种折衷。在一个具体的示例中,提出将在VVC标准中使用的八个MVD偏移(即,1/4图像元素、1/2图像元素、1图像元素、2图像元素、4图像元素、8图像元素、16图像元素和32图像元素)和四个MVD方向(即,+/-x轴和y轴)重新用于所提出的GPM-MVR方案。在另一示例中,将在AVS3标准中使用的现有的五个MVD偏移{1/4图像元素、1/2图像元素、1图像元素、2图像元素和4图像元素}和四个MVD方向(即,+/-x轴和y轴)应用在所提出的GPM-MVR方案中。

[0110] 如在部分“GPM信令设计”中讨论的,因为被用于两个GPM分区的单向MV不能是相同的,所以将迫使两个单向预测合并索引不同的一个约束应用在现有的GPM设计中。然而,在所提出的GPM-MVR方案中,在现有的GPM单向MV之上应用进一步的运动细化。因此,即使在两个GPM分区的基本MV是相同的时,只要两个运动矢量细化的值是不相同的,用于预测两个分区的最终单向MV可能仍然是不同的。基于以上考虑,当应用所提出的GPM-MVR方案时,移除该约束(其限制两个单向预测合并索引是不同的)。另外,因为两个单向预测合并索引被允许是相同的,所以相同的最大值MaxGPMMergeCand-1被用于merge_gpm_idx0和merge_gpm_idx1两者的二值化,其中,MaxGPMMergeCand是单向预测合并列表中的候选的数量。

[0111] 如上文所分析的,当两个GPM分区的单向预测合并索引(即,merge_gpm_idx0和merge_gpm_idx1)相同时,两个运动矢量细化的值不能是相同的,以确保用于两个分区的最终MV是不同的。基于这样的条件,在本公开内容的一个实施例,提出了一种信令冗余移除方法,其在两个GPM分区的单向预测合并索引是相同的(即,merge_gpm_idx0等于merge_gpm_idx1)时,使用第一GPM分区的MVR来降低第二GPM分区的MVR的信令开销。在一个示例中,应用以下信令条件:

[0112] 首先,当标志gpm_mvr_partIdx0_enable_flag等于0(即,不将GPM-MVR应用于第一GPM分区)时,不发信号通知gpm_mvr_partIdx1_enable_flag的标志,但是该标志被推断为1(即,将GPM-MVR应用于第二GPM分区)。

[0113] 其次,当标志gpm_mvr_partIdx0_enable_flag和gpm_mvr_partIdx1_enable_flag两者等于1(即,将GPM-MVR应用于两个GPM分区)并且gpm_mvr_partIdx0_direction_idx等于gpm_mvr_partIdx1_direction_idx(即,两个GPM分区的MVR具有相同的方向)时,第一GPM分区的MVR的幅度(即,gpm_mvr_partIdx0_distance_idx)被用于预测第二GPM分区的MVR的幅度(即,gpm_mvr_partIdx1_distance_idx)。具体地,如果gpm_mvr_partIdx1_distance_idx小于gpm_mvr_partIdx0_distance_idx,则直接发信号通知其原始值。否则(gpm_mvr_partIdx1_distance_idx大于gpm_mvr_partIdx0_distance_idx),在发信号通知给比特流之前,将其值减去一。在解码器侧,为了对gpm_mvr_partIdx1_distance_idx的值进行解码,如果所解析的值小于gpm_mvr_partIdx0_distance_idx,则将gpm_mvr_partIdx1_distance_idx设置为等于所解析的值;否则(所解析的值等于或大于gpm_mvr_partIdx0_distance_idx),将gpm_mvr_partIdx1_distance_idx设置为等于所解析的值加一。在这样的情况下,为了进一步降低开销,可以将不同的最大值MaxGPMVRDistance-1和MaxGPMVRDistance-2用于gpm_mvr_partIdx0_distance_idx和gpm_mvr_partIdx1_

distance_idx的二值化,其中,MaxGPMVRDistance是针对运动矢量细化的允许的幅度的数量。

[0114] 在另一实施例中,提出了将信令顺序切换为gpm_mvr_partIdx0_direction_idx/gpm_mvr_partIdx1_direction_idx,然后gpm_mvr_partIdx0_distance_idx/gpm_mvr_partIdx1_distance_idx,使得在MVR幅度前面,发信号通知MVR幅度。通过这种方式,遵循与上述相同的逻辑,编码器/解码器可以使用第一GPM分区的MVR方向来调节第二GPM分区的MVR方向的信令。在另一实施例中,提出了首先发信号通知第二GPM分区的MVR幅度和方向,并且将其用于调节第二GPM分区的MVR幅度和方向的信令。

[0115] 在另一实施例中,提出在发信号通知现有的GPM语法元素之前,发信号通知GPM-MVR相关的语法元素。具体地,在这样的设计中,首先发信号通知两个标志(gpm_mvr_partIdx0_enable_flag和gpm_mvr_partIdx1_enable_flag),以指示是否将GPM-MVR分别应用于第一GPM分区和第二GPM分区。当一个GPM分区的标志等于一时,距离索引(如由语法元素gpm_mvr_partIdx0_distance_idx和gpm_mvr_partIdx1_distance_idx指示的)和方向索引(如由语法元素gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx1_distance_idx指示的),以指定MVR的方向。之后,发信号通知现有的语法merge_gpm_idx0和merge_gpm_idx1,以标识用于两个GPM分区的单向MV,即,基本MV。表5示出所提出的GPM-MVR信令方案。

[0116] 表5所提出的将单独的MVR用于两个GPM分区的GPM-MVR方法的语法元素(方法二)

	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx0_enable_flag[x0][y0]	ae(v)
	if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx0_directoin_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx0_distance_idx[x0][y0]	ae(v)
	}	
[0117]	gpm_mvr_partIdx1_enable_flag[x0][y0]	ae(v)
	if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx1_direction_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx1_distance_idx[x0][y0]	ae(v)
	}	
	}	
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
		
	}	

[0118] 与表4中的信令方法相似,当应用表5中的GPM-MVR信令方法以确保被用于两个GPM分区的预测的得到的MV不相同,可以应用某些条件。具体地,提出以下条件来根据被应用到第一GPM分区和第二GPM分区的MVR的值来约束单向预测合并索引merge_gpm_idx0和merge_gpm_idx1的信令。

[0119] 首先,当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于0(即,针对两个GPM分区禁用GPM-MVR)时,merge_gpm_idx0和merge_gpm_idx1

的值不能是相同的。

[0120] 第二,当`gpm_mvr_partIdx0_enable_flag`等于1(即,针对第一GPM分区启用GPM-MVR)并且`gpm_mvr_partIdx1_enable_flag`等于0(即,针对第二GPM分区禁用GPM-MVR)时,允许`merge_gpm_idx0`的值和`merge_gpm_idx1`的值是相同的。

[0121] 第三,当`gpm_mvr_partIdx0_enable_flag`等于0(即,针对第一GPM分区禁用GPM-MVR)并且`gpm_mvr_partIdx1_enable_flag`等于1(即,针对第二GPM分区启用GPM-MVR)时,允许`merge_gpm_idx0`的值和`merge_gpm_idx1`的值是相同的。

[0122] 第四,当`gpm_mvr_partIdx0_enable_flag`的值和`gpm_mvr_partIdx1_enable_flag`的值两者等于1(即,针对两个GPM分区启用GPM-MVR)时,关于是否允许`merge_gpm_idx0`的值和`merge_gpm_idx1`的值相同的确定取决于被应用于两个GPM分区的MVR(如由`gpm_mvr_partIdx0_direction_idx`和`gpm_mvr_partIdx0_distance_idx`、以及`gpm_mvr_partIdx1_direction_idx`和`gpm_mvr_partIdx1_distance_idx`指示的)的值。如果两个MVR的值相等,则不允许`merge_gpm_idx0`和`merge_gpm_idx1`是相同的。否则(两个MVR的值不相等),则允许`merge_gpm_idx0`的值和`merge_gpm_idx1`的值是相同的。

[0123] 在以上四种情况下,当不允许`merge_gpm_idx0`的值和`merge_gpm_idx1`的值是相同的时,一个分区的索引值可以被用于另一分区的索引值的预测值。在一种方法中,提出首先发信号通知`merge_gpm_idx0`,并且使用其值来预测`merge_gpm_idx1`。具体地,在编码器处,当`merge_gpm_idx1`大于`merge_gpm_idx0`时,被发送到解码器的`merge_gpm_idx1`的值被减小1。在解码器处,当接收的`merge_gpm_idx1`的值等于或大于接收的`merge_gpm_idx0`的值时,`merge_gpm_idx1`的值被增加1。在另一种方法中,提出首先发信号通知`merge_gpm_idx1`,并且使用其值来预测`merge_gpm_idx0`。因此,在这样的情况下,在编码器处,当`merge_gpm_idx0`大于`merge_gpm_idx1`时,被发送到解码器的`merge_gpm_idx0`的值被减小1。在解码器处,当接收的`merge_gpm_idx0`的值等于或大于接收的`merge_gpm_idx1`的值时,`merge_gpm_idx0`的值被增加1。另外,与现有的GPM信令设计类似,不同的最大值`MaxGPMMergeCand-1`和`MaxGPMMergeCand-2`可以根据信令顺序分别被用于第一索引值和第二索引值的二值化。另一方面,当因为`merge_gpm_idx0`和`merge_gpm_idx1`的值之间不存在相关性,所以允许这两个索引值是相同的时,将同一最大值`MaxGPMMergeCand-1`用于两个索引值的二值化。

[0124] 在以上的方法中,为了降低信令成本,可以将不同的最大值应用于`merge_gpm_idx0`和`merge_gpm_idx1`的二值化。对对应最大值的选择取决于经解码的MVR值(如由`gpm_mvr_partIdx0_enable`、`gpm_mvr_partIdx1_enable`、`gpm_mvr_partIdx0_direction_idx`、`gpm_mvr_partIdx1_direction_idx`、`gpm_mvr_partIdx0_distance_idx`和`gpm_mvr_partIdx1_distance_idx`指示的)。这样的设计在不同的GPM语法元素之间引入了不期望的解析依赖,这可能自始至终影响总体解析。为了解决这样的问题,在一个实施例,提出始终用一个相同的最大值(例如,`MaxGPMMergeCand-1`)来解析`merge_gpm_idx0`的值和`merge_gpm_idx1`的值。当使用这样的方法时,可以使用一个比特流一致性约束来防止两个GPM分区的两个经解码的MV是相同的。在另一种方法中,其还可以移除这样的非同一性约束,使得允许两个GPM分区的经解码的MV是相同的。另一方面,当应用这样的方法(即,为`merge_gpm_idx0`和`merge_gpm_idx1`使用相同的最大值)时,在`merge_gpm_idx0/merge_gpm_idx1`与其它GPM-MVR语法元素之间不存在解析依赖。因此,发信号通知那些语法元素的顺序不再重要。

在一个示例中,提出将发信号通知merge_gpm_idx0/merge_gpm_idx1移动到发信号通知gpm_mvr_partIdx0_enable、gpm_mvr_partIdx1_enable、gpm_mvr_partIdx0_direction_idx、gpm_mvr_partIdx1_direction_idx、gpm_mvr_partIdx0_distance_idx和gpm_mvr_partIdx1_distance_idx之前。

[0125] 利用对称运动矢量细化的几何分区模式

[0126] 对于以上讨论的GPM-MVR方法,发信号通知两个单独的MVR值,其中一个MVR值被应用以改进仅一个GPM分区的基本MV。通过允许针对每个GPM分区的独立的运动细化,这样的方法关于预测准确性的提高可能是高效的。然而,考虑到两个不同组的GPM-MVR语法元素需要从编码器发送到解码器,这样的灵活的运动细化出现的代价是增加了信令开销。为了降低信令开销,在本部分中,提出了一种利用对称运动矢量细化的几何分区模式。具体地,在该方法中,针对一个GPM CU发信号通知一个单个MVR值,并且根据与两个GPM分区相关联的当前图片和参考图片的图片顺序计数(POC)值之间的对称关系,将该单个MVR值用于两个GPM分区两者。表6示出当应用所提出的方法时的语法元素。

[0127] 表6所提出的将对称MVR用于两个GPM分区的GPM-MVR方法的语法元素(方法一)

[0128]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
	gpm_mvr_enable_flag[x0][y0]	ae(v)
	if(gpm_mvr_enable_flag[x0][y0]) {	
	gpm_mvr_direction_idx[x0][y0]	ae(v)
	gpm_mvr_distance_idx[x0][y0]	ae(v)
	}	
		
	}	

[0129] 如表6中所示,在(基于merge_gpm_idx0和merge_gpm_idx1)选择了两个GPM分区的基本MV之后,发信号通知一个标志gpm_mvr_enable_flag,以指示是否将GPM-MVR模式应用于当前GPM CU。当标志等于一时,其指示应用运动细化,以增强两个GPM分区的基本MV。否则(当标志等于零时),其指示运动细化不被应用于两个分区两者。如果启用GPM-MVR模式,进一步发信号通知额外的语法元素,以通过方向索引gpm_mvr_direction_idx和幅度索引gpm_mvr_distance_idx来指定所应用的MVR的值。另外,与MMVD模式类似,MVR符号的含义可以根据GPM分区的当前图片和两个参考图片的POC之间的关系而变化。具体地,当两个参考图片的POC两者大于或小于当前图片的POC时,发信号通知的符号是与两个基本MV相加的MVR的符号。否则(当一个参考图片的POC大于当前图片,而另一参考图片的POC小于当前图片时),将发信号通知的符号应用于第一GPM分区的MVR,并且将相反符号应用于第二GPM分区。在表6中,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0130] 在另一实施例中,提出发信号通知两个不同的标志,以分开地控制针对两个GPM分区的GPM-MVR模式的分开地启用/禁用。然而,当启用GPM-MVR模式时,基于语法元素gpm_mvr_direction_idx和gpm_mvr_distance_idx来发信号通知仅一个MVR。表7中示出这样的

信令方法的对应语法表。

[0131] 表7所提出的将对称MVR用于两个GPM分区的GPM-MVR方法的语法元素 (方法二)

[0132]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
[0133]	gpm_mvr_partIdx0_enable_flag[x0][y0]	ae(v)
	if(merge_gpm_idx0 != merge_gpm_idx1 gpm_mvr_partIdx0_enable_flag)	
	gpm_mvr_partIdx1_enable_flag[x0][y0]	ae(v)
	if(gpm_mvr_partIdx0_enable_flag[x0][y0] gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
	gpm_mvr_direction_idx[x0][y0]	ae(v)
	gpm_mvr_distance_idx[x0][y0]	ae(v)
	}	
		
	}	

[0134] 当应用表7中的信令方法时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。然而,为了确保得到的被应用于两个GPM分区的MV不冗余,当标志gpm_mvr_partIdx0_enable_flag等于0(即,不将GPM-MVR应用于第一GPM分区)时,不发信号通知标志gpm_mvr_partIdx1_enable_flag,但是该标志被推断为1(即,将GPM-MVR应用于第二GPM分区)。

[0135] 针对GPM-MVR的允许的MVR的适配

[0136] 在以上讨论的GPM-MVR方法中,一个固定组的MVR值被用于编码器和解码器两者处的一个视频序列中的GPM CU。这样的设计对于具有高分辨率或具有剧烈运动的视频内容而言是次最优的。在这些情况下,MV往往大得多,使得固定的MVR值对于捕获那些块的真实运动来说可能不是最优的。为了进一步提高GPM-MVR模式的编解码性能,在本公开内容中提出支持允许由GPM-MVR模式在各个编码级别(例如,序列级别、图片/条带图片、编码块组级别等)上进行选择的MVR值的适配。例如,可以根据不同视频序列的特定运动特性,离线得到多个MVR组以及对应码字。编码器可以选择最佳MVR组,并且将所选择的组的对应索引发信号通知给解码器。

[0137] 在本公开内容的特定实施例中,除了包括八个偏移幅度(即,1/4图像元素、1/2图像元素、1图像元素、2图像元素、4图像元素、8图像元素、16图像元素和32图像元素)和四个MVR方向(即,+/-x轴和y轴)的默认MVR偏移外,针对GPM-MVR模式,提出在下表中定义的另外的MVR偏移。

[0138] 表15

[0139]	距离IDX	0	1	2	3	4	5	6	7	8
	偏移(以亮度样点为单位)	1/4	1/2	1	2	3	4	6	8	16

[0140] 表16

[0141]	方向IDX	000	001	010	011	100	101	110	111
	x轴	+1	-1	0	0	+1/2	-1/2	-1/2	+1/2
	y轴	0	0	+1	-1	+1/2	-1/2	+1/2	-1/2

[0142] 在上表15和表16中,x轴和y轴上的值+1/2和-1/2指示水平方向和垂直方向的对角线方向(+45°和-45°)。如表15和表16中所示,与现有的MVR偏移组相比,第二MVR偏移组引入两个新的偏移幅度(即,3图像元素和6图像元素)和四个偏移方向(45°、135°、225°和315°)。新增加的MVR偏移使第二MVR偏移组更适合对具有复杂运动的视频块进行编码。另外,为了实现两个MVR偏移组之间的自适应切换,提出一个控制标志来在某个编码级别(例如,序列、图片、条带、CTU和编码块等)上发信号通知,以指示选择哪组MVR偏移用于在编解码级别下应用的GPM-MVR模式。假设在图片级别上执行所提出的适配,下表17示出在图片头部发信号通知的对应语法元素。

[0143] 表17

[0144]	picture_header_structure() {	描述符
		
	if(sps_mmvd_fullpel_only_enabled_flag)	
	ph_mmvd_fullpel_only_flag	u(1)
	presenceFlag = 0	
	if(!pps_rpl_info_in_ph_flag) /* 该条件有意地不合并到下一条件中,以避免 RplIdx[i]不具有指定值的可能解释。 */	
	presenceFlag = 1	
	else if(num_ref_entries[1][RplIdx[1]] > 0)	
	presenceFlag = 1	
	if(presenceFlag) {	
[0145]	ph_mvd_l1_zero_flag	u(1)
	if(sps_bdof_control_present_in_ph_flag)	
	ph_bdof_disabled_flag	u(1)
	if(sps_dmv_r_control_present_in_ph_flag)	
	ph_dmv_r_disabled_flag	u(1)
	}	
	if(!pps_rpl_info_in_ph_flag num_ref_entries[1][RplIdx[1]] > 0)	
	ph_gpm_mvr_offset_set_flag	u(1)
	if(sps_prof_control_present_in_ph_flag)	
	ph_prof_disabled_flag	u(1)
		
	}	

[0146] 在上表17中,新标志ph_gpm_mvr_offset_set_flag被用于指示被用于图片的对应GPM MVR偏移的选择。当标志等于0时,其意味着在图片中将默认MVR偏移(即,1/4图像元素、1/2图像元素、1图像元素、2图像元素、4图像元素、8图像元素、16图像元素和32图像元素的幅度以及四个MVR方向+/-x轴和y轴)应用于GPM-MVR模式。否则,当标志等于1时,其意味着在图片中将第二MVR偏移(即,1/4图像元素、1/2图像元素、1图像元素、2图像元素、3图像元素、4图像元素、6图像元素、8图像元素、16图像元素的幅度以及八个MVR方向+/-x轴、y轴和45°、135°、225°和315°)应用于GPM-MVR模式。

[0147] 为了发信号通知MVR偏移,可以应用不同的方法。首先,考虑到MVR方向通常在统计上均匀地分布,因此提出使用固定长度的码字来对MVR方向进行二值化。以默认MVR偏移为例,总共存在四个方向,并且00、01、10和11的码字可以被用于表示四个方向。另一方面,因为MVR偏移幅度可以具有适应于视频内容的特定运动特性的不同分布,所以提出使用可变长度的码字来对MVR幅度进行二值化。下表18示出一个特定码字表,其可以被用于默认MVR偏移组和第二MVR偏移组的MVR幅度的二值化。

[0148] 表18

默认 MVR 偏移组		第二 MVR 偏移组	
MVR 偏移	二值化	MVR 偏移	二值化
1/4 图像元素	001	1/4 图像元素	001
1/2 图像元素	1	1/2 图像元素	1
1 图像元素	01	1 图像元素	01
2 图像元素	0001	2 图像元素	0001
4 图像元素	00001	3 图像元素	00001
8 图像元素	000001	4 图像元素	000001
16 图像元素	0000001	6 图像元素	0000001
32 图像元素	0000000	8 图像元素	00000001
		16 图像元素	00000000

[0151] 在其他实施例中,也可以应用不同的固定长度的可变码字来对默认MVR偏移组和第二MVR偏移组的MVR偏移幅度进行二值化,例如,以上码字表中的二进制位“0”和“1”可以被交换以用于适应于上下文自适应二进制算术编码(CABAC)引擎的各种0/1统计量。

[0152] 在一个具体的示例中,提供了两个不同的码字表来对MVR幅度的值进行二值化。以下各表示出在第一码字表和第二码字表中应用的默认MVR偏移组和第二MVR偏移组的对应码字。表19示出在第一码字表中的MVR偏移幅度的码字。表20示出在第二码字表中的MVR偏移幅度的码字。

[0153] 表19

默认 MVR 偏移组		第二 MVR 偏移组	
MVR 偏移	二值化	MVR 偏移	二值化
1/4 图像元素	1	1/4 图像元素	1
1/2 图像元素	10	1/2 图像元素	10
1 图像元素	110	1 图像元素	110
2 图像元素	1110	2 图像元素	1110
4 图像元素	11110	3 图像元素	11110
8 图像元素	111110	4 图像元素	111110
16 图像元素	1111110	6 图像元素	1111110
32 图像元素	1111111	8 图像元素	11111110
		16 图像元素	11111111

[0155] 表20

[0156]

默认 MVR 偏移组		第二 MVR 偏移组	
MVR 偏移	二值化	MVR 偏移	二值化
1/4 图像元素	111110	1/4 图像元素	111110
1/2 图像元素	1	1/2 图像元素	1
1 图像元素	10	1 图像元素	10
2 图像元素	110	2 图像元素	110
4 图像元素	1110	3 图像元素	1110
8 图像元素	11110	4 图像元素	11110
16 图像元素	1111110	6 图像元素	1111110
32 图像元素	1111111	8 图像元素	11111110
		16 图像元素	11111111

[0157] 为了实现两个码字表之间的自适应切换,提出一个指示标志来在某个编码级别(例如,序列、图片、条带、CTU和编码块等)上发信号通知,以指定在编码级别下将哪个码字表用于对MVR幅度进行二值化。假设在图片级别上执行所提出的适配,下表21示出在图片头部发信号通知的对应语法元素,其中新增的语法元素为粗斜体。

[0158] 表21

[0159]

picture_header_structure() {	描述符
.....	
if(sps_mmvd_fullpel_only_enabled_flag)	
ph_mmvd_fullpel_only_flag	u(1)

[0160]

presenceFlag = 0	
if(!pps_rpl_info_in_ph_flag) /*该条件有意地不合并到下一条件中, 以避免 RplIdx[i]不具有指定值的可能解释。*/	
presenceFlag = 1	
else if(num_ref_entries[1][RplIdx[1]] > 0)	
presenceFlag = 1	
if(presenceFlag) {	
ph_mvd_l1_zero_flag	u(1)
if(sps_bdof_control_present_in_ph_flag)	
ph_bdof_disabled_flag	u(1)
if(sps_dmvr_control_present_in_ph_flag)	
ph_dmvr_disabled_flag	u(1)
}	
if(!pps_rpl_info_in_ph_flag num_ref_entries[1][RplIdx[1]] > 0) {	
ph_gpm_mvr_offset_set_flag	u(1)
ph_gpm_mvr_step_codeword_flag	u(1)
}	
if(sps_prof_control_present_in_ph_flag)	
ph_prof_disabled_flag	u(1)
.....	
}	

[0161] 在以上语法表中,新标志ph_gpm_mvr_step_codeword_flag被用于指示被用于图片的MVR幅度的二值化的对应码字表的选择。当标志等于0时,其指示第一码字表被应用于图片;否则(即,标志等于1),其指示第二码字表被应用于图片。

[0162] 在另一实施例中,提出在对整个视频序列的编码/解码期间,始终将一个码字表用于对MVR偏移幅度进行二值化。在一个示例中,提出始终将第一码字表用于MVR幅度的二值化。在另一示例中,提出始终将第二码字表用于MVR幅度的二值化。在另一方法中,提出将一个固定的码字表(例如,第二码字表)用于所有MVR幅度的二值化。

[0163] 在其它方法中,可以应用一种基于统计量的二值化方法,以自适应地设计在不发信号通知的情况下在运行中用于MVR偏移幅度的最优码字。被用于确定最优码字的统计量可以是,但不限于在数个先前经编码的图片、条带和/或编码块上收集MVR偏移幅度的概率分布。可以在各个频率级别上重新确定/更新码字。例如,可以在每次在GPM-MVR模式中对CU进行编码时完成更新。在另一示例中,可以在每次存在数个CU(例如,8或16个)在GPM-MVR模式中被编码时对该更新进行重新确定和/或更新。

[0164] 在其它方法中,代替重新设计一个新组的码字,所提出的基于统计量的方法还可以被用于基于相同组的码字来对MVR幅度值进行重新排序,从而为较多使用的幅度指派较短的码字,并且为较少使用的幅度指派较长的码字。以下表为例,假设在图片级别上收集统计量,列“使用”指示在先前经编码的图片中由GPM-MVR编码块使用的不同MVR偏移幅度的对应百分比。根据“使用”列中的值,使用相同的二值化方法(即,截断一元码字),编码器/解码器可以基于MVR幅度值的使用对其进行排序;之后,编码器/解码器可以为最频繁使用的MVR幅度(即,1图像元素)指派最短的码字(即,“1”),并且为第二最频繁使用的MVR幅度(即,1/2图像元素)指派第二最短码字(即,“01”),……,并且为两个最少使用的MVR幅度(即,16图像元素和32图像元素)指派最长的码字(即,“0000001”和“0000000”)。如此,通过这样的重排序方案,同一组码字可以被自由地重新排序,以适应MVR幅度的统计分布的动态改变。

MVR偏移	使用	二值化
1/4图像元素	15%	001
1/2图像元素	20%	01
1图像元素	30%	1
2图像元素	10%	0001
4图像元素	9%	00001
8图像元素	6%	000001
16图像元素	5%	0000001
32图像元素	5%	0000000

[0165] 用于GPM-MVR率失真优化的编码器加速逻辑

[0166] 对于所提出的GPM-MVR方案,为了确定用于每个GPM分区的最优MVR,编码器可能需要对每个GPM分区的率失真成本测试多次,每次改变被应用的MVR值。这可能显著地增加GPM模式的编码复杂度。为了解决编码复杂度问题,在本部分中提出以下快速编码逻辑。

[0167] 首先,由于在VVC和AVS3中应用的四叉树/二叉树/三叉树块分区结构,可以在率失真优化(RDO)过程期间校验一个相同的编码块,每个通过一个不同的分区路径划分。在当前VTM/HPM编码器实现方式中,每当通过不同的块分区组合获得一个相同的CU时,总是测试GPM和GPM-MVR模式以及其它帧间和帧内编码模式。一般而言,对于不同的分区路径,一个CU的仅相邻块可能是不同的,然而,这应当对一个CU将选择的最优编码模式具有相对轻微的影响。基于这样的考虑,为了减少被应用的GPM RDO的总数,提出存储是否在首次校验一个

CU的RD成本时选择GPM模式的决策。之后,当通过RDO过程再次校验相同的CU(通过另一分区路径)时,只要针对该CU首次选择了GPM,就校验GPM(包括GPM-MVR)的RD成本。在不选择GPM用于一个CU的初始RD校验的情况下,当通过另一分区路径实现相同的CU时,仅测试GPM(没有GPM-MVR)。在另一方法中,当不选择GPM用于一个CU的初始RD校验时,在通过另一分区路径实现相同的CU时,不测试GPM和GPM-MVR两者。

[0169] 第二,为了减少用于GPM-MVR模式的GPM分区的数量,提出当首次校验一个CU的RD成本时,维持没有最小RD成本的前M个GPM分区模式。之后,当通过RDO过程再次校验相同的CU(通过另一分区路径)时,对于GPM-MVR模式,仅测试该M个GPM分区模式。

[0170] 第三,为了减少针对初始RDO过程而测试的GPM分区的数量,对于每个GPM分区,提出在将不同的单向预测合并候选用于两个GPM分区时,首先计算绝对差之和(SAD)值。然后,对于一个特定分区模式之下的每个GPM分区,选择具有最小SAD值的最佳单向预测合并候选,并且计算分区模式的对应SAD值,其等于用于两个GPM分区的最佳单向预测合并候选的SAD值之和。然而,对于以下RD过程,针对GPM-MVR模式,测试针对先前步骤的具有最佳SAD值的前N个分区模式。

[0171] 利用显式运动信令的几何分区

[0172] 在本部分中,提出多种方法来将GPM模式扩展到常规帧间模式的双向预测,其中将GPM模式的两个单向MV从编码器显式地发信号通知给解码器。

[0173] 在第一种解决方案(解决方案一)中,提出充分重新使用双向预测的现有运动信令以发信号通知GPM模式的两个单向MV。表8示出所提出的方案的修改的语法表,其中新增的语法元素为粗斜体。如表8中所示,在该解决方案中,充分重新使用发信号通知L0和L1运动信息的所有现有语法元素,以分别指示两个GPM分区的单向MV。另外,假设L0 MV是始终与第一GPM分区相关联的,并且L1 MV是始终与第二GPM分区相关联的。另一方面,在表8中,在GPM标志(即,gpm_flag)前面发信号通知帧间预测语法,即,inter_pred_idc,使得inter_pred_idc的值可以被用于调节gpm_flag的存在。具体地,在inter_pred_idc等于PRED_BI(即,双向预测)并且inter_affine_flag和sym_mvd_flag两者等于0(即,CU既不是通过仿射模式也不是通过SMVD模式编码的)时,仅需要发信号通知标志gpm_flag。当不发信号通知标志gpm_flag时,其值总是被推断为0(即,禁用GPM模式)。当gpm_flag是1时,进一步发信号通知另一语法元素gpm_partition_idx,以指示针对当前CU的(从总计64个GPM分区中)所选择的GPM模式。

[0174] 表8用于解决方案一的运动信令的修改的语法表(选项一)

[0175]

if(sh_slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag [x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1 [x0][y0]	ae(v)
if(ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	

[0176]	MvdCpL1[x0][y0][2][0] = 0	
	MvdCpL1[x0][y0][2][1] = 0	
	} else {	
	if(sym_mvd_flag[x0][y0]) {	
	MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
	MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
	} else	
	mvd_coding(x0, y0, 1, 0)	
	if(MotionModelIdc[x0][y0] > 0)	
	mvd_coding(x0, y0, 1, 1)	
	if(MotionModelIdc[x0][y0] > 1)	
	mvd_coding(x0, y0, 1, 2)	
	}	
	mvp_l1_flag[x0][y0]	ae(v)
	} else {	
	MvdL1[x0][y0][0] = 0	
	MvdL1[x0][y0][1] = 0	
	}	
	<i>if(inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && !sym_mvd_flag[x0][y0] && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) && cbWidth < 128 && cbHeight < 128) {</i>	
	<i>gpm_flag[x0][y0]</i>	ae(v)
	<i>if(gpm_partition_flag[x0][y0]) {</i>	
	<i>gpm_partition_idx[x0][y0]</i>	ae(v)
	<i>}</i>	

[0177] 在另一方法中,提出将发信号通知标志gpm_flag放置在其它帧间信令语法元素前面,使得gpm_flag的值可以被用于确定是否需要存在其它帧间语法元素。表9示出当在应用这样的方法时的对应语法表,其中新增的语法元素为粗斜体。可以看出,在表9中首先发信号通知gpm_flag。当gpm_flag等于1时,inter_pred_idc、inter_affine_flag和sym_mvd_flag的对应信令可以被旁路。取而代之地,三个语法元素的对应值可以分别被推断为PRED_BI、0和0。

[0178] 表9用于解决方案一的运动信令的修改的语法表(选项二)

[0179]	<i>if(sh_slice_type == B && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) && cbWidth < 128 && cbHeight < 128) {</i>	
	<i>gpm_flag[x0][y0]</i>	ae(v)
	<i>if(gpm_partition_flag[x0][y0]) {</i>	
	<i>gpm_partition_idx[x0][y0]</i>	ae(v)
	<i>}</i>	
	<i>if(sh_slice_type == B && !gpm_flag[x0][y0])</i>	
	<i>inter_pred_idc[x0][y0]</i>	ae(v)

[0180]

if(sps_affine_enabled_flag && !gpm_flag[x0][y0] && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !gpm_flag[x0][y0] && !inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag [x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1 [x0][y0]	ae(v)
if(ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else	

[0181]	<code>mvd_coding(x0, y0, 1, 0)</code>	
	<code>if(MotionModelIdx[x0][y0] > 0)</code>	
	<code>mvd_coding(x0, y0, 1, 1)</code>	
	<code>if(MotionModelIdx[x0][y0] > 1)</code>	
	<code>mvd_coding(x0, y0, 1, 2)</code>	
	<code>}</code>	
	<code>mvp_l1_flag[x0][y0]</code>	<code>ae(v)</code>
	<code>} else {</code>	
	<code>MvdL1[x0][y0][0] = 0</code>	
	<code>MvdL1[x0][y0][1] = 0</code>	
	<code>}</code>	

[0182] 在表8和表9两者中, SMVD模式不能与GPM模式组合。在另一示例中, 提出在通过GPM模式对当前CU进行编码时允许SMVD模式。当允许这样的组合时, 通过遵循SMVD的相同设计, 假设两个GPM分区的MVD是对称的, 使得仅需要发信号通知第一GPM分区的MVD, 并且第二GPM分区的MVD始终与第一MVD对称。当应用这样的方法时, 可以移除sym_mvd_flag关于gpm_flag的对应信令条件。

[0183] 如上所示, 在第一种解决方案中, 其始终假设将L0 MV用于第一GPM分区, 并且将L1 MV用于第二GPM分区。就该方法禁止两个GPM分区的MV来自一个相同的预测列表 (L0或L1) 的意义而言, 这样的设计可能不是最优的。为了解决这样的问题, 提出了一种利用如在表10中所示的信令设计的替代GPM-EMS方案 (解决方案二)。在表10中, 新增的语法元素为粗斜体。如表10中所示, 首先发信号通知标志gpm_flag。当标志等于1 (即, 启用GPM) 时, 发信号通知语法gpm_partition_idx, 以指定所选择的GPM模式。然后, 发信号通知一个额外的标志gpm_pred_dir_flag0, 以指示第一GPM分区的MV所来自的对应预测列表。当标志gpm_pred_dir_flag0等于1时, 其指示第一GPM分区的MV来自L1; 否则 (标志等于0), 其指示第一GPM分区的MV来自L0。之后, 利用现有的语法元素ref_idx_l0、mvp_l0_flag和mvd_coding() 来发信号通知第一GPM分区的参考图片索引、MVP索引和MVD的值。另一方面, 与第一分区类似, 引入另一语法元素gpm_pred_dir_flag1, 以选择第二GPM分区的对应预测列表, 随后将现有的语法元素ref_idx_l1、mvp_l1_flag和mvd_coding() 用于得到第二GPM分区的MV。

[0184] 表10用于解决方案二的运动信令的修改的语法表

[0185]	<code>if(sh_slice_type == B && cbWidth >= 8 && cbHeight >= 8</code>	
	<code>&& cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) &&</code>	
	<code>cbWidth < 128 && cbHeight < 128) {</code>	
	<code><i>gpm_flag[x0][y0]</i></code>	<code>ae(v)</code>
	<code>if(<i>gpm_partition_idx[x0][y0]</i>) {</code>	
	<code><i>gpm_partition_idx[x0][y0]</i></code>	<code>ae(v)</code>
	<code><i>gpm_pred_dir_flag0[x0][y0]</i></code>	<code>ae(v)</code>
	<code><i>ref_idx_l0[x0][y0]</i></code>	<code>ae(v)</code>
	<code>mvd_coding(x0, y0, 0, 0)</code>	
	<code><i>mvp_l0_flag[x0][y0]</i></code>	<code>ae(v)</code>
	<code><i>gpm_pred_dir_flag1[x0][y0]</i></code>	<code>ae(v)</code>
	<code><i>ref_idx_l1[x0][y0]</i></code>	<code>ae(v)</code>
	<code>mvd_coding(x0, y0, 0, 0)</code>	
	<code><i>mvp_l1_flag[x0][y0]</i></code>	<code>ae(v)</code>
	<code>} else {</code>	

[0186]

if(sh_slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag [x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1 [x0][y0]	ae(v)
if(ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	

[0187]	} else	
	mvd_coding(x0, y0, 1, 0)	
	if(MotionModelIdx[x0][y0] > 0)	
	mvd_coding(x0, y0, 1, 1)	
	if(MotionModelIdx[x0][y0] > 1)	
	mvd_coding(x0, y0, 1, 2)	
	}	
	mvp_l1_flag[x0][y0]	ae(v)
	} else {	
	MvdL1[x0][y0][0] = 0	
	MvdL1[x0][y0][1] = 0	
	}	
	}	

[0188] 最后,应当提到的是,考虑到GPM模式由两个单向预测分区组成(除划分边缘上的混合样点外),VVC和AVS3中的专门被设计用于双向预测的一些现有编码工具(例如,双向光流),当针对一个帧间CU启用所提出的GPM-EMS方案时,可以自动地旁路解码器侧运动矢量细化(DMVR)和利用CU权重的双向预测(BCW)。例如,当针对一个CU启用所提出的GPM-EMS中的一个时,考虑到BCW不能被应用于GPM模式,不需要针对该CU进一步发信号通知对应BCW权重,以减小信令开销。

[0189] GPM-MVR与GPM-EMS的组合

[0190] 在本部分中,提出将GPM-MVR与GPM-EMS进行组合,以用于具有几何分区的一个CU。具体地,不同于其中仅可以应用基于合并的运动信令或显式信令中的一项来发信号通知两个GPM分区的单向预测MV的GPM-MVR或GPM-EMS,在所提出的方案中,其允许1)使用基于GPM-MVR的运动信令的一个分区和使用基于GPM-EMS的运动信令的另一分区;或2)使用基于GPM-MVR的运动信令的两个分区;或3)使用基于GPM-EMS的运动信令的两个分区。使用表4中的GPM-MVR信令和表10中的GPM-EMS,表11示出在组合了所提出的GPM-MVR和GPM-EMS之后的对应语法表。在表11中,新增的语法元素为粗斜体。如表11中所示,分别针对分区#1和#2引入了两个额外的语法元素gpm_merge_flag0和gpm_merge_flag1,其指定对应分区使用基于GPM-MVR的合并信令或基于GPM-EMS的显式信令。当标志是一时,其意味着针对将通过merge_gpm_idxX、gpm_mvr_partIdxX_enabled_flag、gpm_mvr_partIdxX_direction_idx和gpm_mvr_partIdxX_distance_idx发信号通知其GPM单向预测运动的分区,启用基于GPM-MVR的信令,其中,X=0、1。否则,如果标志是零,其意味着将使用语法元素gpm_pred_dir_flagX、ref_idx_lX、mvp_lX_flag和mvd_lX通过GPM-EMS方式显式地发信号通知分区的单向预测运动,其中,X=0、1。

[0191] 表11用于利用GPM-MVR和GPM-EMS的组的GPM模式的所提出的语法表

[0192]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	<i>gpm_partition_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_merge_flag0[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_merge_flag0[x0][y0]</i>) {	
	<i>merge_gpm_idx0[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx0_directoin_idx[x0][y0]</i>	<i>ae(v)</i>
[0193]	<i>gpm_mvr_partIdx0_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	}else{	
	<i>gpm_pred_dir_flag0[x0][y0]</i>	<i>ae(v)</i>
	<i>ref_idx_l0[x0][y0]</i>	<i>ae(v)</i>
	<i>mvd_coding(x0, y0, 0, 0)</i>	
	<i>mvp_l0_flag[x0][y0]</i>	<i>ae(v)</i>
	}	
	<i>gpm_merge_flag1[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_merge_flag1[x0][y0]</i>) {	
	<i>merge_gpm_idx1[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx1_directoin_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx1_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	}else{	
	<i>gpm_pred_dir_flag1[x0][y0]</i>	<i>ae(v)</i>
	<i>ref_idx_l1[x0][y0]</i>	<i>ae(v)</i>
	<i>mvd_coding(x0, y0, 0, 0)</i>	
	<i>mvp_l1_flag[x0][y0]</i>	<i>ae(v)</i>
	}	
		
	}	

[0194] GPM-MVR与模板匹配的组合

[0195] 在本部分中,提供了不同的解决方案来将GPM-MVR与模板匹配进行组合。

[0196] 在方法一中,当在GPM模式中对一个CU进行编码时,提出针对两个GPM分区发信号通知两个单独的标志,每个标志指示是否通过模板匹配来进一步细化对应分区的单向运动。当启用该标志时,使用当前CU的左侧和顶部相邻经重建样点生成模板;然后,将通过遵循与部分“模板匹配”中引入的相同的过程最小化模板与其参考样点之间的差来细化分区的单向运动。否则(当禁用该标志时),不将模板匹配应用于分区,可以进一步应用GPM-MVR。使用表5中的GPM-MVR信令方法作为示例,表12示出当将GPM-MVR与模板匹配组合时的对应语法表。在表12中,新增的语法元素为粗斜体。

[0197] 表12将GPM-MVR与模板匹配进行组合的所提出的方法的语法元素(方法一)

[0198]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	gpm_tm_enable_flag0[x0][y0]	ae(v)
	if(!gpm_tm_enable_flag0) {	
	gpm_mvr_partIdx0_enable_flag[x0][y0]	ae(v)
	}	
[0199]	if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx0_directoin_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx0_distance_idx[x0][y0]	ae(v)
	}	
	gpm_tm_enable_flag1[x0][y0]	ae(v)
	if(!gpm_tm_enable_flag1) {	
	gpm_mvr_partIdx1_enable_flag[x0][y0]	ae(v)
	}	
	if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx1_direction_idx[x0][y0]	ae(v)
	gpm_mvr_partIdx1_distance_idx[x0][y0]	ae(v)
	}	
	}	
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
		
	}	

[0200] 如表12中所示,在所提出的方案中,首先发信号通知两个额外的标志gpm_tm_enable_flag0和gpm_tm_enable_flag1,以指示是否分别针对两个GPM分区对运动进行细化。当标志是一时,其指示应用TM以细化一个分区的单向MV。当标志是零时,进一步发信号通知一个标志(gpm_mvr_partIdx0_enable_flag或gpm_mvr_partIdx1_enable_flag),以指示是否将GPM-MVR分别应用于GPM分区。当一个GPM分区的标志等于一时,发信号通知距离索引(如由语法元素gpm_mvr_partIdx0_distance_idx和gpm_mvr_partIdx1_distance_idx指示的)和方向索引(如由语法元素gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx1_distance_idx指示的),以指定MVR的方向。之后,用信号通知现有语法merge_gpm_idx0和merge_gpm_idx1,以标识针对两个GPM分区的单向MV。同时,与被应用于表5的信令条件类似,可以应用以下条件以确保得到的被用于两个GPM分区的预测的MV不相同。

[0201] 首先,当gpm_tm_enable_flag0的值和gpm_tm_enable_flag1的值两者等于1(即,针对两个GPM分区启用TM)时,merge_gpm_idx0的值和merge_gpm_idx1的值不能是相同的。

[0202] 第二,当gpm_tm_enable_flag0和gpm_tm_enable_flag1中的一项是一,并且另一项是零时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0203] 否则,即,gpm_tm_enable_flag0和gpm_tm_enable_flag1两者等于一:首先,当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于0(即,针对两个GPM分区禁用GPM-MVR)时,merge_gpm_idx0的值和merge_gpm_idx1的值不能

是相同的；第二，当gpm_mvr_partIdx0_enable_flag等于1（即，针对第一GPM分区启用GPM-MVR）并且gpm_mvr_partIdx1_enable_flag等于0（即，针对第二GPM分区禁用GPM-MVR）时，允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的；第三，当gpm_mvr_partIdx0_enable_flag等于0（即，针对第一GPM分区禁用GPM-MVR）并且gpm_mvr_partIdx1_enable_flag等于1（即，针对第二GPM分区启用GPM-MVR）时，允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的；第四，当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于1（即，针对两个GPM分区启用GPM-MVR）时，关于是否允许merge_gpm_idx0的值和merge_gpm_idx1的值相同的确定取决于被应用于两个GPM分区的MVR的值（如由gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx0_distance_idx、以及gpm_mvr_partIdx1_direction_idx和gpm_mvr_partIdx1_distance_idx指示的）。如果两个MVR的值相等，则不允许merge_gpm_idx0和merge_gpm_idx1是相同的。否则（两个MVR的值不相等），则允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0204] 在以上方法一中，将TM和MVR排他地应用于GPM。在这样的方案中，禁止进一步将MVR应用于TM模式的经细化的MV之上。因此，为了为GPM进一步提供更多的MV候选，提出了方法二来实现MVR偏移在TM细化的MV之上的应用。表13示出当将GPM-MVR与模板匹配组合时的对应语法表。在表13中，新增的语法元素为粗斜体。

[0205] 表13将GPM-MVR与模板匹配进行组合的所提出的方法的语法元素（方法二）

	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	<i>gpm_tm_enable_flag0[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx0_directoin_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx0_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
[0206]	<i>gpm_tm_enable_flag1[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx1_direction_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx1_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	}	
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
		
	}	

[0207] 如表13中所示，不同于表12，移除gpm_mvr_partIdx0_enable_flag和gpm_mvr_partIdx1_enable_flag关于gpm_tm_enable_flag0和gpm_tm_enable_flag1的信令条件。使得无论是否应用TM来细化一个GPM分区的单向运动，始终允许将MV细化应用于GPM分区的MV。与以前类似，应当应用以下条件以确保得到的两个GPM分区的MV不相同。

[0208] 首先,当gpm_tm_enable_flag0和gpm_tm_enable_flag1中的一项是一并且另一项是零时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0209] 否则,即,gpm_tm_enable_flag0和gpm_tm_enable_flag1两者等于一,或两个标志等于零:首先,当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于0(即,针对两个GPM分区禁用GPM-MVR)时,merge_gpm_idx0的值和merge_gpm_idx1的值不能是相同的;第二,当gpm_mvr_partIdx0_enable_flag等于1(即,针对第一GPM分区启用GPM-MVR)并且gpm_mvr_partIdx1_enable_flag等于0(即,针对第二GPM分区禁用GPM-MVR)时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的;第三,当gpm_mvr_partIdx0_enable_flag等于0(即,针对第一GPM分区禁用GPM-MVR)并且gpm_mvr_partIdx1_enable_flag等于1(即,针对第二GPM分区启用GPM-MVR)时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的;第四,当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于1(即,针对两个GPM分区启用GPM-MVR)时,关于是否允许merge_gpm_idx0的值和merge_gpm_idx1的值相同的确定取决于被应用于两个GPM分区的MVR的值(如由gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx0_distance_idx、以及gpm_mvr_partIdx1_direction_idx和gpm_mvr_partIdx1_distance_idx指示的)。如果两个MVR的值相等,则不允许merge_gpm_idx0和merge_gpm_idx1是相同的。否则(两个MVR的值不相等),则允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0210] 在以上两种方法中,需要发信号通知两个单独的标志,以指示是否将TM应用于每个GPM分区。由于额外的开销,尤其是在低比特率下,增加的信令可能降低总体编解码效率。为了降低信令开销,代替引入额外的信令,提出了方法三来将基于TM的单向MV插入到GPM模式的单向MV候选列表中。遵循与在部分“模板匹配”中所描述的相同的TM过程并且使用GPM的原始单向MV作为初始MV来生成基于TM的单向MV。通过这样的方案,不需要将额外的控制标志从编码器进一步发信号通知给解码器。取而代之地,解码器可以通过从比特流接收的对应合并索引(即,merge_gpm_idx0和merge_gpm_idx1)来识别是否通过TM对一个MV进行细化。可能存在不同的方法来安排常规GPM MV候选(即,非TM)和基于TM的MV候选。在一种方法中,提出将基于TM的MV候选放置在MV候选列表的开头,随后是非基于TM的MV候选。在另一方法中,提出首先将非基于TM的MV候选放置在开头,随后是基于TM的候选。在另一方法中,提出以交织的方式放置基于TM的MV候选和非基于TM的MV候选。例如,可以放置前N个非基于TM的候选;然后是所有的基于TM的候选;最后是其余的非基于TM的候选。在另一示例中,可以放置前N个基于TM的候选;然后是所有的非基于TM的候选;最后是其余的基于TM的候选。在另一示例中,提出一个接一个地放置非基于TM的候选和基于TM的候选,即,一个非基于TM的候选、一个基于TM的候选等。

[0211] 在方法一中,在GPM-MVR标志之前发信号通知两个GPM模板标志。具体地,在这样的设计中,可以通过首先发信号通知等于零的一个分区的GPM模板标志来仅针对一个给定的GPM分区启用GPM-MVR。尽管可以使用适当的上下文模型来对GPM模板标志进行编码,但是其将会对GPM-MVR模式招致信令损失。为了解决这样的问题,在本公开内容的一个实施例中,提出在发信号通知GPM-TM模式之前,首先发信号通知GPM-MVR模式。具体地,在该方法中,针对每个GPM分区,首先发信号通知GPM-MVR标志,以指示是否将GPM-MVR应用于该分区。当标志等于一时,进一步发信号通知MVR语法元素gpm_mvr_partIdx0_distance_idx/gpm_mvr_

partIdx1_distance_idx和gpm_mvr_partIdx0_dierction_idx/gpm_mvr_partIdx1_direction_idx,以指定分区的MVR幅度和方向的对应值。否则,当分区的GPM-MVR标志等于假时,将发信号通知GPM-TM标志,以指示是否应用GPM-TM模式(使用左侧和顶部相邻的经重建样点来细化分区的MV)。表22示出当应用以上信令方法时的对应语法表,其中新增的语法元素为粗斜体。

[0212] 表22

[0213]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	<i>ae(v)</i>
	gpm_mvr_partIdx0_enable_flag[x0][y0]	<i>ae(v)</i>
	if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {	
	gpm_mvr_partIdx0_directoin_idx[x0][y0]	<i>ae(v)</i>
	gpm_mvr_partIdx0_distance_idx[x0][y0]	<i>ae(v)</i>
	} else {	
	gpm_tm_enable_flag0[x0][y0]	<i>ae(v)</i>
	}	
	gpm_mvr_partIdx1_enable_flag[x0][y0]	<i>ae(v)</i>
	if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {	
[0214]	gpm_mvr_partIdx1_direction_idx[x0][y0]	<i>ae(v)</i>
	gpm_mvr_partIdx1_distance_idx[x0][y0]	<i>ae(v)</i>
	} else {	
	gpm_tm_enable_flag1[x0][y0]	<i>ae(v)</i>
	}	
	}	
	merge_gpm_idx0[x0][y0]	<i>ae(v)</i>
	merge_gpm_idx1[x0][y0]	<i>ae(v)</i>
		
	}	

[0215] 此外,为了移除GPM合并索引之间的信令冗余,应当应用以下条件。

[0216] 首先,当gpm_tm_enable_flag0和gpm_tm_enable_flag1中的一项是一并且另一项是零时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0217] 第二,当gpm_tm_enable_flag0和gpm_tm_enable_flag1两者等于一时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0218] 在另一示例中,当gpm_tm_enable_flag0的值和gpm_tm_enable_flag1的值两者等于1时,不允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0219] 第三,当gpm_tm_enable_flag0和gpm_tm_enable_flag1两者等于零时,应用不同的条件。当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于0(即,针对两个GPM分区禁用GPM-MVR)时,merge_gpm_idx0的值和merge_gpm_idx1的值不能是相同的。当gpm_mvr_partIdx0_enable_flag等于1(即,针对第一GPM分区启用GPM-MVR)并且gpm_mvr_partIdx1_enable_flag等于0(即,针对第二GPM分区禁用GPM-MVR)时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。当gpm_mvr_partIdx0_enable_

flag等于0(即,针对第一GPM分区禁用GPM-MVR)并且gpm_mvr_partIdx1_enable_flag等于1(即,针对第二GPM分区启用GPM-MVR)时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于1(即,针对两个GPM分区启用GPM-MVR)时,关于是否允许merge_gpm_idx0的值和merge_gpm_idx1的值相同的确定取决于被应用于两个GPM分区的MVR的值(如由gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx0_distance_idx、以及gpm_mvr_partIdx1_direction_idx和gpm_mvr_partIdx1_distance_idx指示的)。如果两个MVR的值相等,则不允许merge_gpm_idx0和merge_gpm_idx1是相同的。否则(两个MVR的值不相等),则允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0220] 在另一方法中,代替使用两个单独的GPM-TM标志,提出一个单个标志来联合地控制针对两个GPM分区的模板匹配的启用/禁用。当该标志是真时,其意味着需要通过模板匹配方案,基于模板(即,左侧和顶部相邻的经重建样点)与其对应的参考样点之间的差的最小化来细化两个GPM分区的两个单向MV。具体地,与方法四类似,首先针对一个GPM CU发信号通知两个GPM-MVR标志,以指示是否将GPM-MVR应用于一个特定的GPM分区。当每个分区的GPM-MVR标志等于真时,在以下针对分区,进一步发信号通知MVR幅度和MVR方向。此外,当两个GPM分区的GPM-MVR标志两者等于假时,将进一步发信号通知GPM-TM标志,以指示是否将GPM-TM应用于两个GPM分区。表23示出在应用这样的设计时的对应语法表,其中新增的语法元素为粗斜体。

[0221] 表23

[0222]	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
[0223]	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	ae(v)
	<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	<i>if(gpm_mvr_partIdx0_enable_flag[x0][y0]) {</i>	
	<i>gpm_mvr_partIdx0_directoin_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx0_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	<i>if(gpm_mvr_partIdx1_enable_flag[x0][y0]) {</i>	
	<i>gpm_mvr_partIdx1_direction_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx1_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	<i>if(!gpm_mvr_partIdx0_enable_flag[x0][y0]</i>	
	<i>&& !gpm_mvr_partIdx1_enable_flag[x0][y0]) {</i>	
	<i>gpm_tm_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	}	
	merge_gpm_idx0[x0][y0]	ae(v)
	merge_gpm_idx1[x0][y0]	ae(v)
		
	}	

[0224] 在另一实施例中,提出在发信号通知两个GPM-MVR标志之前,针对两个GPM分区,发信号通知GPM-TM标志。对应的,GPM-TM的值可以用于调节两个GPM-MVR标志的存在,使得仅在GPM-TM标志的值等于零(即,不将GPM-TM应用于两个GPM分区)时发信号通知GPM-MVR标志。表24示出在应用这样的信令方案时的GPM模式的对应语法表,其中新增的语法元素为粗斜体。

[0225] 表24

	merge_data(x0, y0, cbWidth, cbHeight, chType) {	描述符
		
	if(!ciip_flag[x0][y0]) {	
	merge_gpm_partition_idx[x0][y0]	<i>ae(v)</i>
	<i>gpm_tm_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	if(<i>gpm_tm_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>	<i>ae(v)</i>
[0226]	}	
	if(<i>gpm_mvr_partIdx0_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx0_directoin_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx0_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	if(<i>gpm_tm_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>	<i>ae(v)</i>
	}	
	if(<i>gpm_mvr_partIdx1_enable_flag[x0][y0]</i>) {	
	<i>gpm_mvr_partIdx1_direction_idx[x0][y0]</i>	<i>ae(v)</i>
	<i>gpm_mvr_partIdx1_distance_idx[x0][y0]</i>	<i>ae(v)</i>
	}	
	merge_gpm_idx0[x0][y0]	<i>ae(v)</i>
[0227]	merge_gpm_idx1[x0][y0]	<i>ae(v)</i>
		
	}	

[0228] 另外,对于两种方法,为了移除GPM合并索引之间的信令冗余,应当应用以下条件。

[0229] 首先,当gpm_tm_enable_flag是一时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0230] 第二,当gpm_tm_enable_flag等于零时,可以应用不同的条件。例如,当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者等于0(即,针对两个GPM分区禁用GPM-MVR)时,merge_gpm_idx0的值和merge_gpm_idx1的值不能是相同的。此外,当gpm_mvr_partIdx0_enable_flag等于1(即,针对第一GPM分区启用GPM-MVR)并且gpm_mvr_partIdx1_enable_flag等于0(即,针对第二GPM分区禁用GPM-MVR)时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。此外,当gpm_mvr_partIdx0_enable_flag等于0(即,针对第一GPM分区禁用GPM-MVR)并且gpm_mvr_partIdx1_enable_flag等于1(即,针对第二GPM分区启用GPM-MVR)时,允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。此外,当gpm_mvr_partIdx0_enable_flag的值和gpm_mvr_partIdx1_enable_flag的值两者

等于1 (即,针对两个GPM分区启用GPM-MVR)时,关于是否允许merge_gpm_idx0的值和merge_gpm_idx1的值相同的确定取决于被应用于两个GPM分区的MVR的值 (如由gpm_mvr_partIdx0_direction_idx和gpm_mvr_partIdx0_distance_idx、以及gpm_mvr_partIdx1_direction_idx和gpm_mvr_partIdx1_distance_idx指示的)。如果两个MVR的值相等,则不允许merge_gpm_idx0和merge_gpm_idx1是相同的。否则 (两个MVR的值不相等),则允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0231] 在另一示例中,当gpm_tm_enable_flag等于一时,不允许merge_gpm_idx0的值和merge_gpm_idx1的值是相同的。

[0232] 当将模板匹配方案应用于GPM模式时,通过进行计算量大的运动估计以识别用于每个GPM分区的最优单向MV,额外的复杂度对于编码器和解码器两者而言变得是必要的。这样的不可忽视的复杂度增加可能使GPM模式对于某些低端编码器或某些视频应用 (例如,需要请求低视频延迟的直播视频流式传输、视频会议和视频游戏) 而言是不可行的。基于这样的考虑,提出在某一高编码级别 (例如,序列级别、图片/条带级别、编码块组级别等) 上增加一个控制标志,以在该级别下自适应地启用/禁用针对CU的GPM-TM模式。假设在图片级别上执行所提出的适配,表25示出在图片头部发信号通知的对应语法元素,其中新增的语法元素为粗斜体。

[0233] 表25

[0234]	picture_header_structure() {	描述符
		
	if(sps_mmvd_fullpel_only_enabled_flag)	
	ph_mmvd_fullpel_only_flag	u(1)
	presenceFlag = 0	
	if(!pps_rpl_info_in_ph_flag)/* 该条件有意地不合并到下一条件中,以避免RplsIdx[i]不具有指定值的可能解释。*/	
	presenceFlag = 1	
	else if(num_ref_entries[1][RplsIdx[1]] > 0)	
	presenceFlag = 1	
	}	
[0235]	if(presenceFlag) {	
	ph_mvd_l1_zero_flag	u(1)
	if(sps_bdof_control_present_in_ph_flag)	
	ph_bdof_disabled_flag	u(1)
	if(sps_dmvr_control_present_in_ph_flag)	
	ph_dmvr_disabled_flag	u(1)
	}	
	if(!pps_rpl_info_in_ph_flag num_ref_entries[1][RplsIdx[1]] > 0) {	
	if(sps_dmvd_enable_flag) {	
	ph_gpm_tm_enable_flag	u(1)
	}	
	}	
	if(sps_prof_control_present_in_ph_flag)	
	ph_prof_disabled_flag	u(1)
		
	}	

[0236] 在以上语法表25中,标志sps_dmvd_enable_flag是序列级别控制标志,其指示是否针对编码视频序列启用模板匹配,并且ph_gpm_tm_enable_flag是所提出的GPM-TM控制标志,其被用于指示是否可以将GPM-TM应用于图片内部的CU。

[0237] 利用运动矢量修剪的GPM候选列表构建

[0238] 如在介绍中所讨论的,为了获得两个几何分区的MV,首先根据常规合并候选列表生成过程直接得到一个单向预测候选列表。考虑到对每个GPM MV的预测方向的选择是基于对应的合并索引的奇偶校验,两个几何分区的MV可以是相同的,这并非明显地合乎情理,因为CU的几何分区不能提供超过非分区情况的任何额外的益处。为了避免这样的冗余,提出在生成一个GPM CU的单向预测MV候选列表时应用运动矢量修剪,使得在并且仅在其不与列表中的现有候选中的任何候选相同时,仅可以将一个MV添加到列表中。在另一方案中,进一步提出一个MV门限,以在比较两个MV时应用。具体地,通过这样的方法,当两个MV的差(分别在水平方向和垂直方向上)小于一个MV门限时,两个MV被视为是相同的;否则(在一个方向上的MV差大于或等于MV门限),两个MV被视为是不相同的。在一种方法中,提出将一个固定的MV门限用于所有的块大小。在另一方法中,提出基于编码块的大小来确定MV门限的值,使得将较大的MV门限用于较大的CU,而将较小的MV门限用于小CU。在一些示例中,当块中的样点的数量 $N < 64$ 时,MV门限的值被设置为1/4图像元素;并且当 $64 \leq N < 256$ 时,MV门限的值被设置为1/2图像元素;并且当 $N \geq 256$ 时,MV门限的值被设置为1图像元素。

[0239] 改进的单向MV候选列表构建

[0240] 提出了一种改进的候选列表构建方法,以根据常规合并模式的MV候选列表得到单向MV候选。

[0241] 首先,获得基于奇偶校验的单向MV。与现有的GPM设计相同,首先根据常规合并候选列表生成过程得到多个单向MV。例如,如图5中所示,n被表示为GPM MV候选列表中的单向运动的索引。第n个合并候选的LX运动矢量(其中X等于n的奇偶校验)被用作候选列表中的第n个单向MV。在第n个扩展合并候选的LX运动矢量不存在的情况下,将取而代之地选择相同候选的L(1-X)MV。

[0242] 在一些示例中,获得基于反奇偶校验的单向MV。当单向MV候选列表未满足时,根据常规合并候选列表的双向预测MV得到的额外的单向MV被进一步添加到单向MV候选列表中。具体地,对于常规合并候选列表中的具有合并索引n的每个双向预测MV,双向预测MV的L(1-X)MV被进一步添加到单向MV候选列表中,其中,X等于n的奇偶校验。

[0243] 在一些示例中,获得成对平均单向MV。当单向MV候选列表未满足时,通过对现有的单向MV候选列表中的一个参考图片列表(L0和L1)中的前两个单向候选进行平均来将一个或多个成对平均候选添加到列表中。在一些示例中,在基于反奇偶校验的单向MV被获得并且被添加到单向MV候选列表之后,获得成对平均单向MV。在一些其它示例中,在获得基于反奇偶校验的单向MV并且将基于反奇偶校验的单向MV添加到单向MV候选列表之前,可以获得成对平均单向MV。

[0244] 例如,对于一个给定的参考图片列表L0/L1,编码器/解码器可以假设第一MV候选被定义为p0Cand,并且第二L0 MV候选被定义为p1Cand。如果两个MV候选指向相同的参考图片,则通过对指向相同参考的两个MV候选进行平均来生成一个成对平均候选;否则,当两个MV候选指向不同的参考图片时,通过对p0Cand和p1Cand进行平均来计算成对MV的幅度,并

且选择第一MV候选的参考图片作为得到的成对平均MV的参考图片。在另一示例中,当两个MV候选指向不同的参考图片时,选择第二MV候选的参考图片作为得到的成对平均MV的参考图片。

[0245] 此外,获得零单向MV。在单向MV候选列表仍然未满足的情况下,将零单向MV(其指向当前图片的L0/L1参考图片列表中的不同的参考图片)周期性地添加到单向MV候选列表中,直到达到列表的最大长度。

[0246] 另外,在以上单向MV候选列表生成方案中,可以进一步应用MV修剪过程,以从列表中移除冗余的MV候选。在一个或多个实施例中,应用默认MV修剪方法,使得在并且仅在其不与列表中的现有候选中的任何候选相同时,仅可以将一个MV添加到列表中。在一些其它实施例中,如在部分“利用运动矢量修剪的GPM候选列表构建”中提出了替代MV修剪方法,其中被用于确定两个MV是否相同的MV门限取决于当前CU的块大小。

[0247] 图9示出与用户接口960耦合的计算环境(或计算设备)910。计算环境910可以是数据处理服务器的一部分。在一些实施例中,计算设备910可以执行如根据本公开内容的各种示例的前文所描述的各种方法或过程(例如,编码/解码方法或过程)中的任何方法或过程。计算环境910可以包括处理器920、存储器940和I/O接口950。

[0248] 处理器920通常控制计算环境910的整体操作,例如,与显示、数据采集、数据通信和图像处理相关联的操作。处理器920可以包括用于执行指令以执行上述方法中的全部或一些步骤的一个或多个处理器。此外,处理器920可以包括促进处理器920与其它组件之间的交互的一个或多个模块。处理器可以是中央处理单元(CPU)、微处理器、单片机、GPU等。

[0249] 存储器940被配置为存储各种类型的数据以支持计算环境910的操作。存储器940可以包括预定软件942。这样的数据的示例包括用于在计算环境910上操作的任何应用或方法的指令、视频数据集、图像数据等。存储器940可以通过使用任何类型的易失性或非易失性存储器设备或其组合来实现,例如,静态随机存取存储器(SRAM)、电可擦除可编程只读存储器(EEPROM)、可擦除可编程只读存储器(EPROM)、可编程只读存储器(PROM)、只读存储器(ROM)、磁存储器、闪存、磁盘或光盘。

[0250] I/O接口950提供处理器920与外围接口模块(例如,键盘、点击轮、按钮等)之间的接口。按钮可以包括但不限于主页按钮、开始扫描按钮和停止扫描按钮。I/O接口950可以与编码器和解码器耦合。

[0251] 在一些实施例中,还提供了一种包括例如被包括在存储器940中的多个程序的非暂时性计算机可读存储介质,所述多个程序由计算环境910中的处理器920可执行以用于执行上述方法。例如,非暂时性计算机可读存储介质可以是ROM、RAM、CD-ROM、磁带、软盘、光学数据存储设备等。

[0252] 非暂时性计算机可读存储介质具有被存储在其中的用于由具有一个或多个处理器的计算设备执行的多个程序,其中,所述多个程序在由一个或多个处理器执行时,使计算设备执行上述用于运动预测的方法。

[0253] 在一些实施例中,计算环境910可以利用用于执行上述方法的一个或多个专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、图形处理单元(GPU)、控制器、微控制器、微处理器或其它电子组件来实现。

[0254] 图8是示出根据本公开内容的示例的用于在GPM中对视频块进行解码的方法的流程图。

[0255] 在步骤801中,处理器920可以将视频块分割成第一几何分区和第二几何分区。

[0256] 在步骤802中,处理器920可以通过添加多个常规合并候选来构建GPM的单向MV候选列表。

[0257] 多个常规合并候选可以是根据常规合并候选列表生成过程得到的。例如,如图5中所示,第 n 个合并候选的LX运动矢量(其中X等于 n 的奇偶校验)被用作候选列表中的第 n 个单向MV。

[0258] 在步骤803中,处理器920可以通过响应于确定单向MV候选列表未满足来将根据常规合并候选列表的一个或多个双向预测MV得到的一个或多个额外的单向MV添加到单向MV候选列表,来构建第一更新的单向MV候选列表。

[0259] 在一些示例中,可以通过获得第一参考图片列表中的具有奇数合并索引的一个或多个MV候选和第二参考图片列表中的具有偶数合并索引的一个或多个MV来得到一个或多个额外的单向MV。例如,第一参考图片列表可以是L0,并且第二参考图片列表可以是L1。额外的单向MV可以包括L0中的具有奇数合并索引的候选和L1中的具有偶数合并索引的候选。也就是说,对于常规合并候选列表中的具有合并索引 n 的每个双向预测MV,双向预测MV的 $L(1-X)$ MV被进一步添加到单向MV候选列表中,其中,X等于 n 的奇偶校验。在一些示例中,第一参考图片列表可以是L1,并且第二参考图片列表可以是L0。

[0260] 在一些示例中,如在步骤804中所示,处理器920可以通过将一个或多个成对平均候选添加到第一更新的单向MV候选列表来构建第二更新的单向MV候选列表。

[0261] 例如,处理器920可以获得第一参考图片列表或第二参考图片列表中的前两个单向MV候选,并且响应于确定前两个单向MV候选指示相同的参考图片,通过对前两个单向MV候选进行平均来获得成对平均候选。

[0262] 在一些示例中,可以响应于确定前两个单向MV候选指示不同的参考图片,通过借助对前两个单向MV候选进行平均来确定成对平均候选的幅度并且将第一单向MV候选的参考图片确定为成对平均候选的参考图片来获得成对平均候选。

[0263] 在一些示例中,可以响应于确定前两个单向MV候选指示不同的参考图片,通过借助对前两个单向MV候选进行平均来确定成对平均候选的幅度并且将第二单向MV候选的参考图片确定为成对平均候选的参考图片来获得成对平均候选。

[0264] 在一些示例中,如在步骤805中所示,处理器920可以响应于确定第二更新的单向MV候选列表未满足,将零单向MV周期性地添加到第二更新的单向MV候选列表,直到达到最大长度。

[0265] 在一些示例中,当构建单向MV候选列表时,可以从单向MV候选列表中移除冗余的候选。

[0266] 例如,处理器920可以响应于确定额外的单向MV等于单向MV候选列表中的候选来跳过将额外的单向MV添加到单向MV候选列表。

[0267] 在一些示例中,MV门限被用于确定额外的单向MV是否等于单向MV候选列表中的候选。例如,响应于确定额外的单向MV与单向MV候选列表中的候选之间的差小于MV门限,确定额外的单向MV等于单向MV候选列表中的候选,其中,MV门限是固定的门限或基于视频块的

块大小的变量。

[0268] 此外,处理器920可以响应于确定成对平均候选等于第一更新的单向MV候选列表中的候选来跳过将成对平均候选添加到第一更新的单向MV候选列表。在一些示例中,响应于确定成对平均候选与第一更新的单向MV候选列表中的候选之间的差小于MV门限,确定成对平均候选等于该候选,其中,MV门限是固定的门限或基于视频块的块大小的变量。

[0269] 在步骤806中,处理器920可以生成针对第一几何分区的单向MV和针对第二几何分区的单向MV。

[0270] 在一些示例中,提供了一种用于在GPM中对视频块进行解码的装置。该装置包括处理器920和被配置为存储由处理器可执行的指令的存储器940;其中,处理器在执行指令时被配置为执行如图8中所示的方法。

[0271] 在一些其它示例中,提供了一种其中存储有指令的非暂时性计算机可读存储介质。当指令由处理器920执行时,指令使处理器执行如图8中所示的方法。

[0272] 根据对这里公开的公开内容的说明书和实践的考虑,本公开内容的其它示例对于本领域技术人员而言将是显而易见的。本申请旨在涵盖本公开内容的任何变型、使用或适配,这些变型、使用或适配遵循本公开内容的一般原理并且包括与本公开内容的在本领域的已知实践或惯例的范围内的偏离。旨在将本说明书和示例仅视为示例性的。

[0273] 将理解的是,本公开内容不限于上述和在附图中示出的确切示例,并且可以在不脱离其范围的情况下进行各种修改和改变。

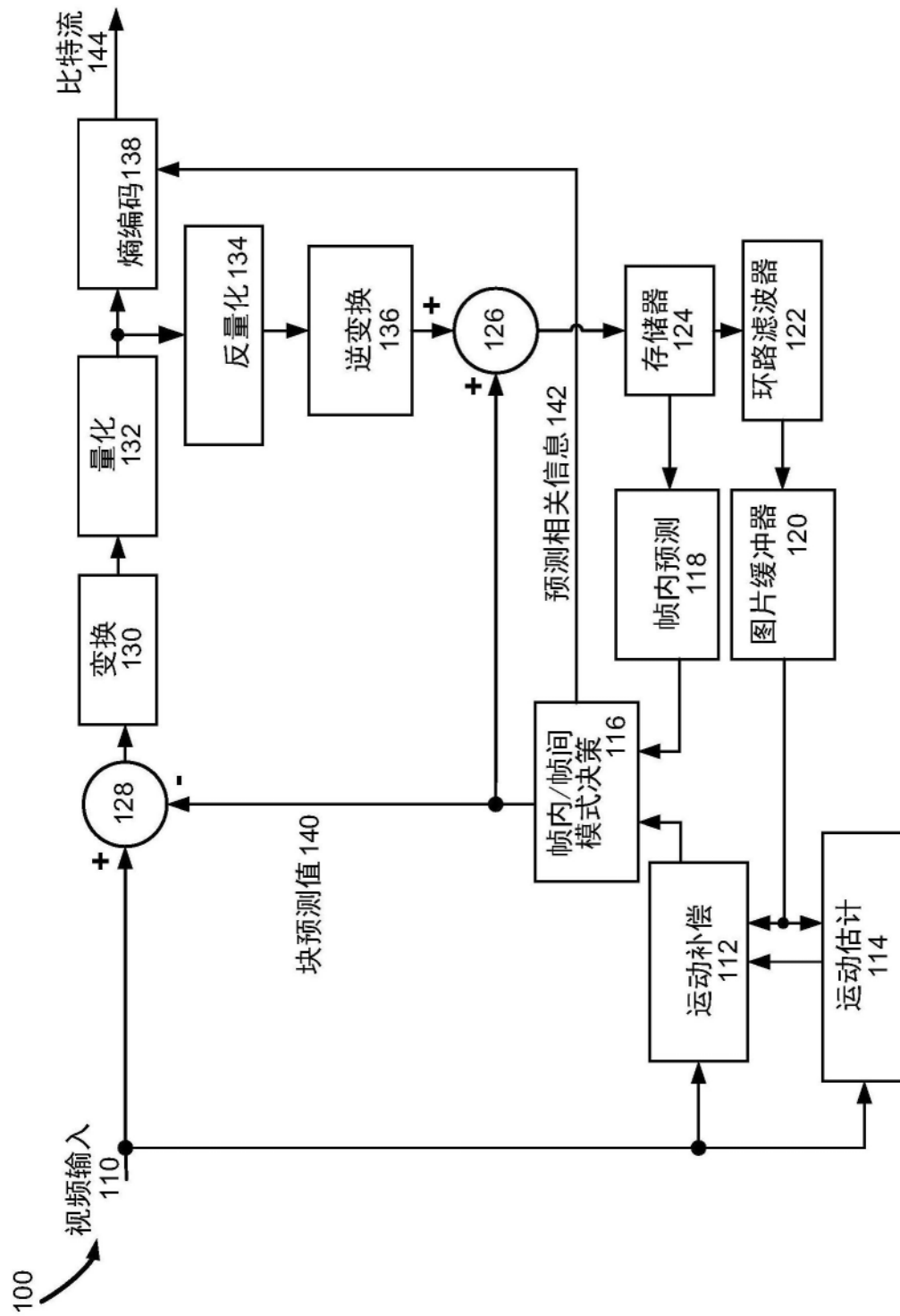


图1

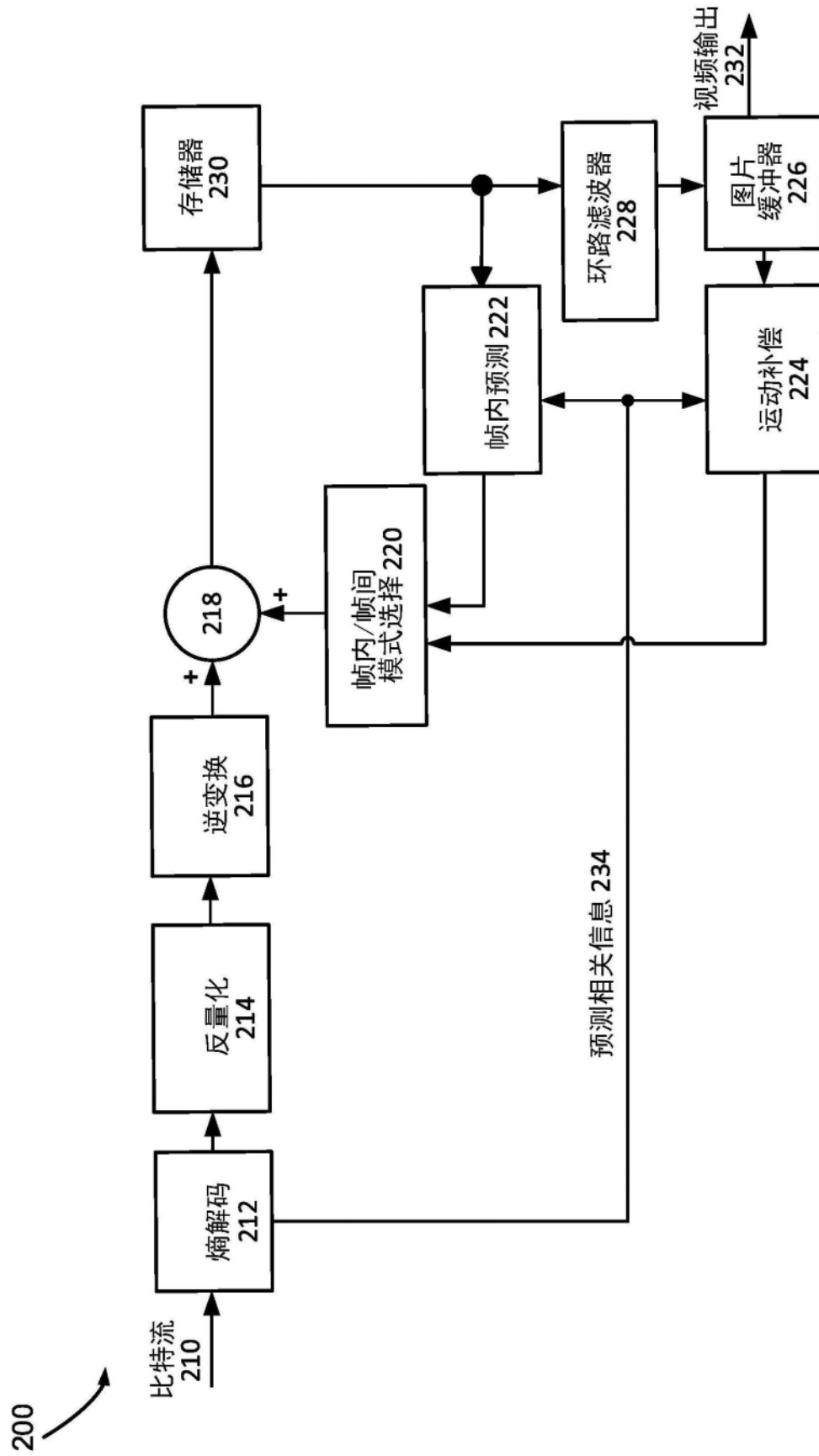


图2

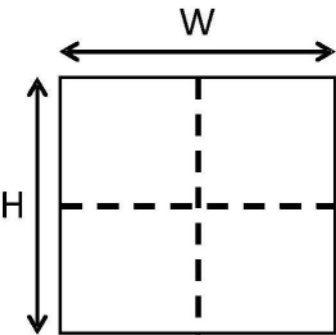


图3A

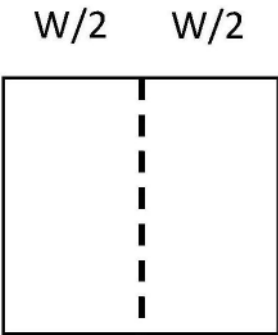


图3B

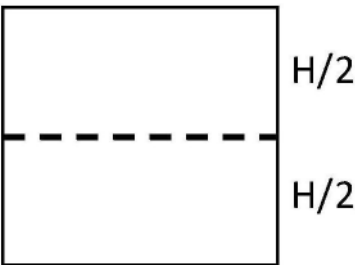


图3C

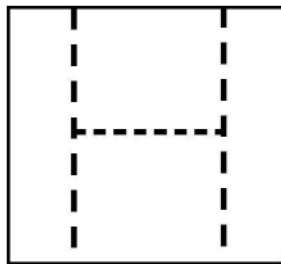
$W/4$ $W/2$ $W/4$ 

图3D

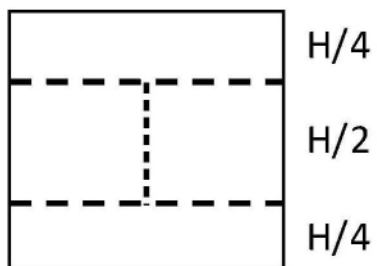


图3E

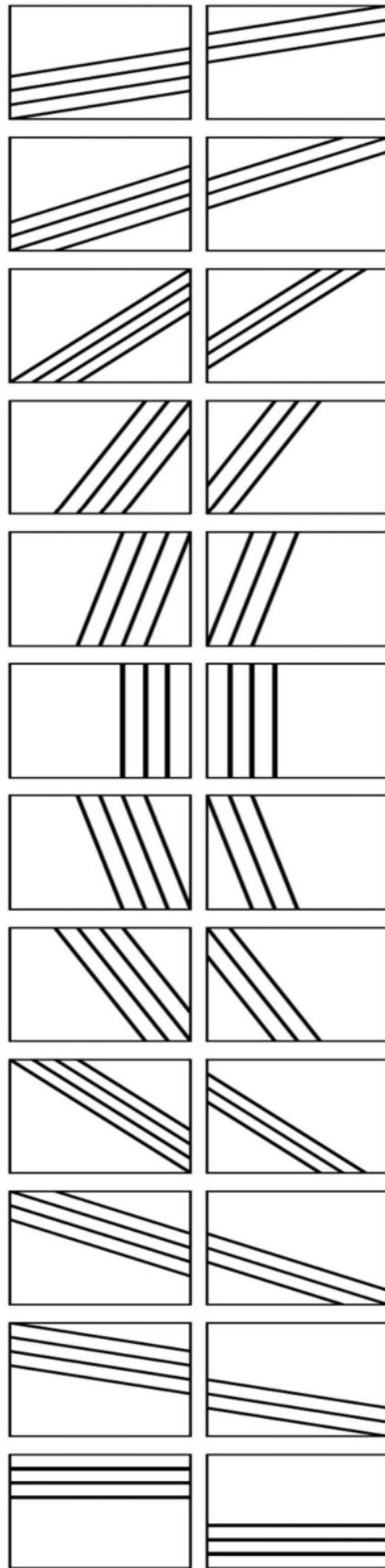


图4

	L0 MV	L1 MV
合并索引0	X	
合并索引1		X
合并索引2	X	
合并索引3		X
合并索引4	X	

图5

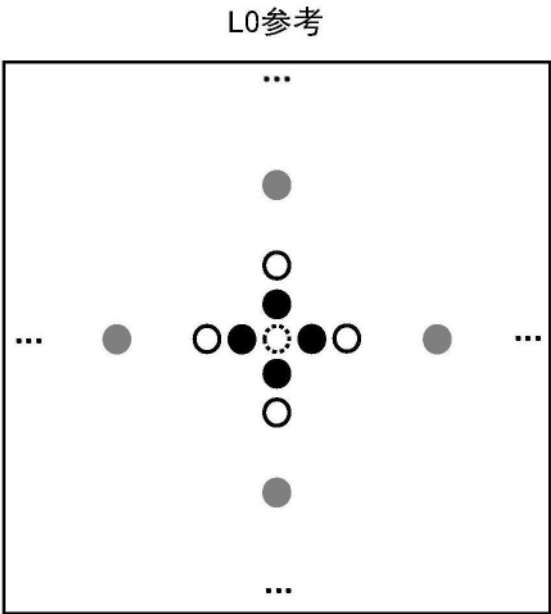


图6A

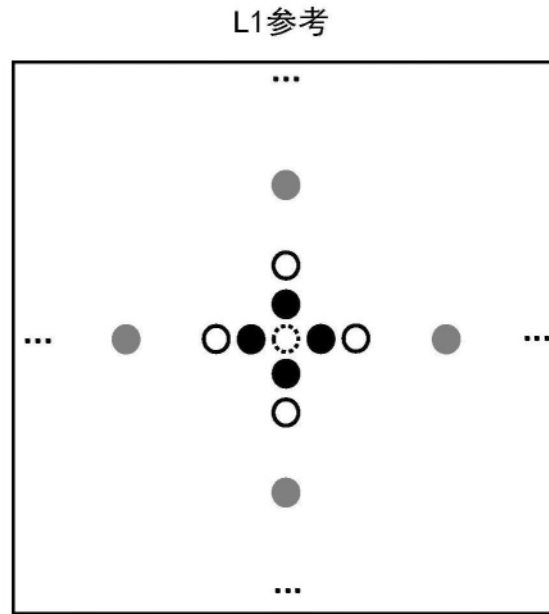


图6B

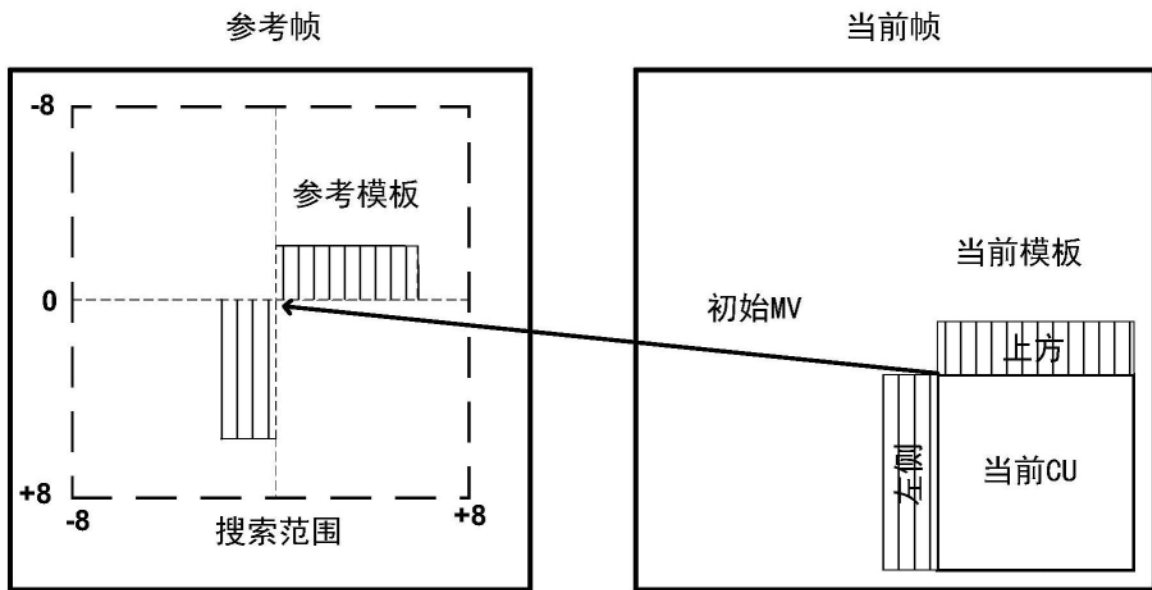


图7

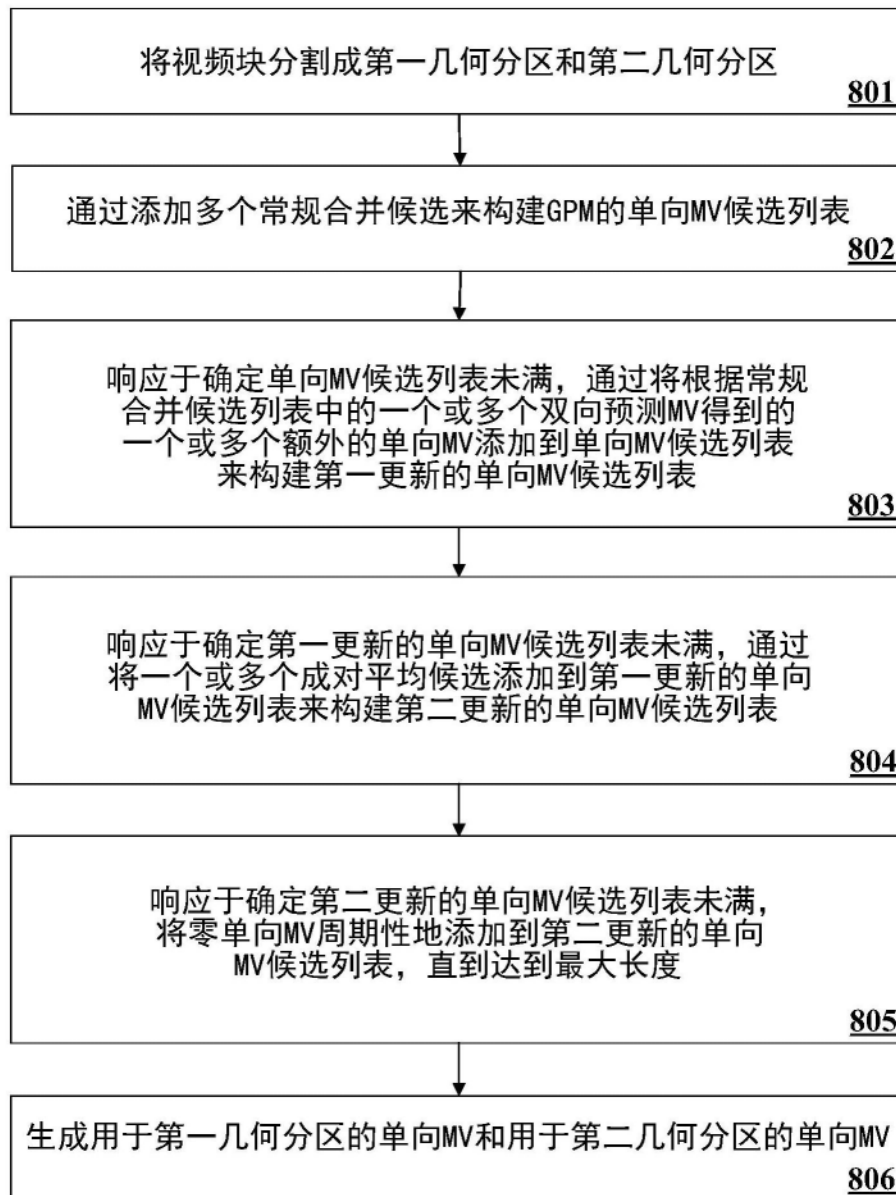


图8

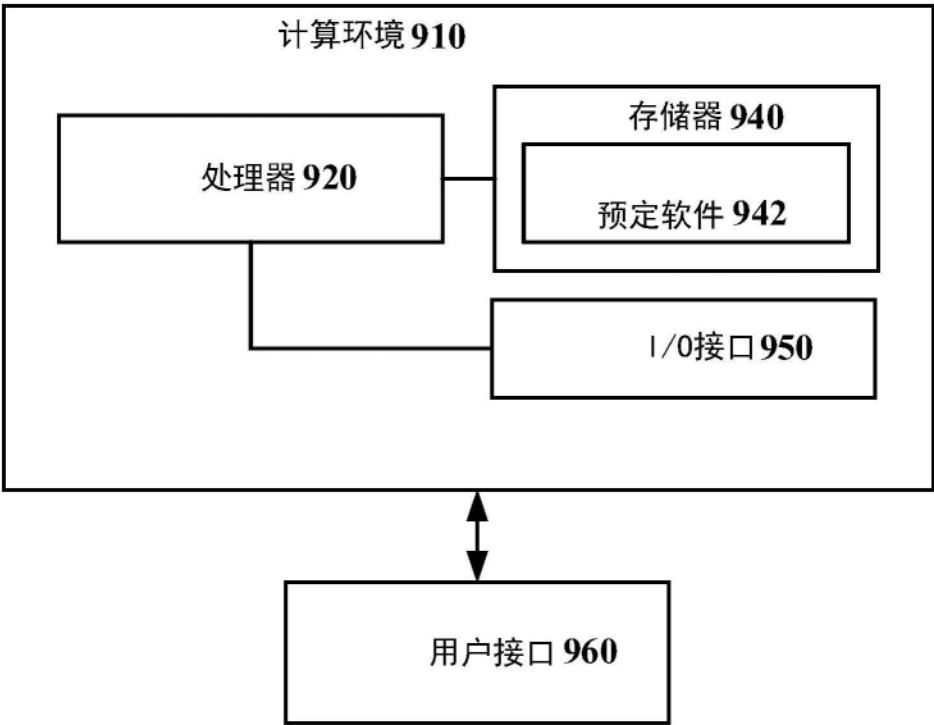


图9

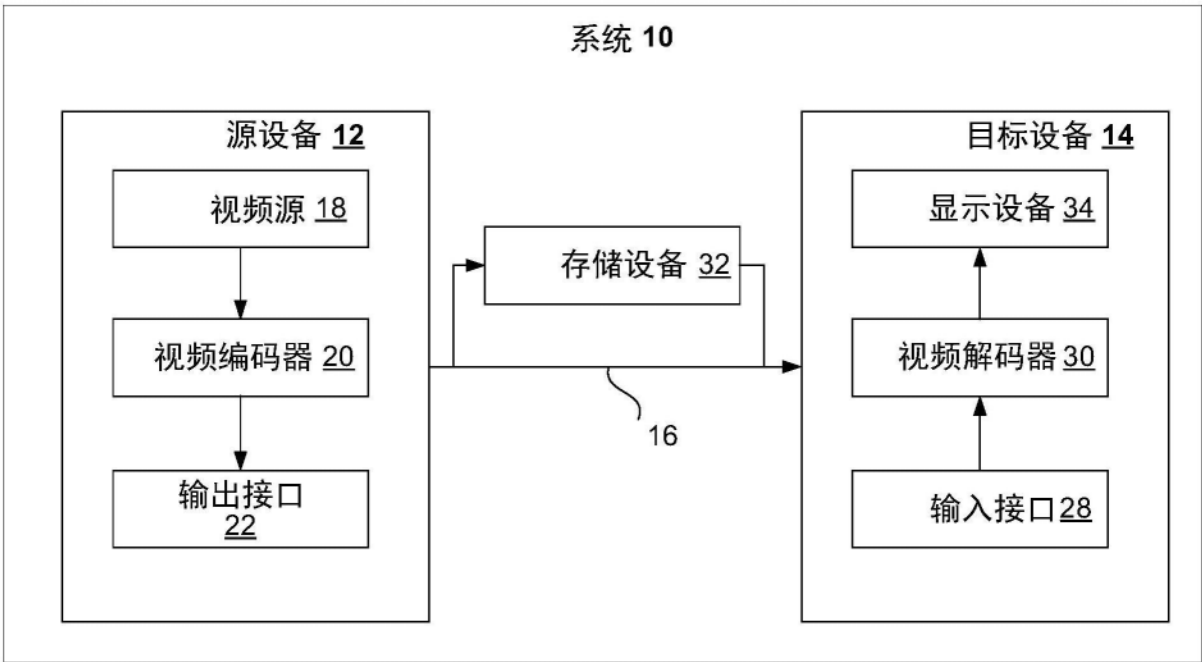


图10