



(21) 申请号 202380027887.0

(22) 申请日 2023.03.14

(30) 优先权数据

63/319,763 2022.03.14 US

(85) PCT国际申请进入国家阶段日

2024.09.14

(86) PCT国际申请的申请数据

PCT/US2023/015232 2023.03.14

(87) PCT国际申请的公布数据

W02023/177695 EN 2023.09.21

(71) 申请人 北京达佳互联信息技术有限公司

地址 100085 北京市海淀区上地西路6号1
幢1层101D1-7

(72) 发明人 陈漪纹 闫宁 修晓宇 郭哲玮

朱弘正 陈伟 王祥林 于冰

(74) 专利代理机构 北京律智知识产权代理有限公司 11438

专利代理师 李玉锁

(51) Int.Cl.

H04N 19/159 (2006.01)

H04N 19/105 (2006.01)

H04N 19/573 (2006.01)

H04N 19/176 (2006.01)

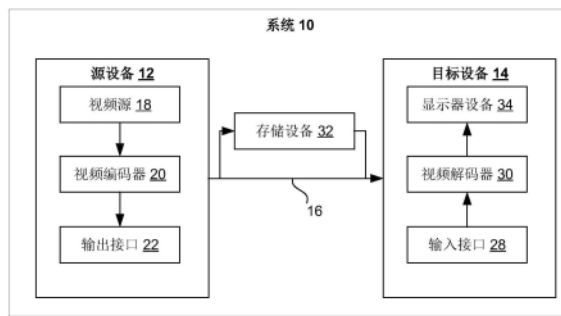
权利要求书3页 说明书23页 附图12页

(54) 发明名称

视频编解码中的帧间预测

(57) 摘要

提供了视频解码和编码方法、装置和非暂态存储介质。在一种解码方法中,解码器确定当前图片或当前图片的条带启用了高级运动矢量预测AMVP-合并AM模式。此外,解码器针对第一参考图片列表LX获得当前图片中当前块的AMVP运动矢量MV作为LX MV,其中X为0或1。此外,解码器针对第二参考图片列表L(1-X)获得有效合并MV候选列表,其中有效合并MV候选列表包括具有L(1-X) MV的合并候选。此外,解码器在有效合并MV候选列表中选择AM模式的L(1-X)MV。



1. 一种视频解码方法,包括:

由解码器确定针对当前图片或当前图片的条带启用高级运动矢量预测AMVP-合并AM模式;

由解码器针对第一参考图片列表LX获取所述当前图片中当前块的AMVP运动矢量MV作为LX MV,其中X为0或1;

由解码器针对第二参考图片列表L(1-X)获取有效合并MV候选列表,其中所述有效合并MV候选列表包括具有L(1-X)MV的合并候选;以及

由解码器在所述有效合并MV候选列表中选择所述AM模式的L(1-X)MV。

2. 根据权利要求1所述的方法,其中,无论合并候选的L(1-X)MV的参考图片是否与LX MV的参考图片处于相反方向,具有L(1-X)MV的所述合并候选都被确定为有效。

3. 根据权利要求1所述的方法,其中,在所有有效合并MV候选中,所述AM模式的L(1-X)MV被隐式地选择为与每个候选相关联的L(1-X)合并预测块和LX AMVP预测块之间具有最小匹配成本的MV。

4. 根据权利要求1所述的方法,其中,在确定只有一个参考图片能够用于LX MV时,不在比特流中发信号通知参考图片索引;或者在确定有多于一个的参考图片能够用于LX MV时,在比特流中发信号通知参考图片索引。

5. 根据权利要求1所述的方法,其中,在确定模板匹配功能已启用时,有条件地对具有较高模板匹配成本的所述LX MV或所述AM模式的L(1-X)MV应用模板匹配MV细化。

6. 根据权利要求5所述的方法,其中,在所有参考图片具有在所述当前图片之前的显示顺序,或者至少一些参考图片具有在所述当前图片之后的显示顺序的条件下,对具有较高模板匹配成本的所述LX MV或所述AM模式的L(1-X)MV应用所述模板匹配MV细化。

7. 根据权利要求1所述的方法,其中,无论模板匹配功能是否已启用,均不对所述AM模式的L(1-X)MV或所述LX MV应用模板匹配MV细化。

8. 根据权利要求1所述的方法,其中所述AM模式的L(1-X)MV被隐式地选择为具有所述LX MV和所述L(1-X)MV之间最小差的MV。

9. 根据权利要求1所述的方法,进一步包括:

在确定所述LX MV的参考图片和所述AM模式的所述L(1-X)MV的参考图片不同时,获得所述L(1-X)MV的缩放MV。

10. 根据权利要求9所述的方法,进一步包括:

基于缩放比率获得所述缩放MV,所述缩放比率是基于所述当前图片与所述LX MV和所述AM模式的所述L(1-X)MV的参考图片之间的时间距离计算的。

11. 根据权利要求1所述的方法,其中所述L(1-X)MV在比特流中显式地发信号通知,以指示为所述AM模式选择了哪个合并候选。

12. 根据权利要求1所述的方法,其中,在确定至少一些参考图片具有在所述当前图片之后的显示顺序时,具有L(1-X)MV的合并候选被视为无效,其中所述合并候选的L(1-X)MV的参考图片与所述LX MV的参考图片处于同一方向。

13. 根据权利要求1所述的方法,其中,在确定所有参考图片具有在所述当前图片之前的显示顺序时,在所有有效的合并MV候选中,仅选择所述AM模式的一个L(1-X)MV,作为具有与每个候选相关联的L(1-X)合并预测块和LX AMVP预测块之间的最小匹配成本的MV,并且

不在比特流中发信号通知合并MV索引;并且在确定至少一些参考图片具有在所述当前图片之后的显示顺序时,在比特流中发信号通知合并MV索引以指示所述AM模式的L(1-X)MV。

14. 根据权利要求1所述的方法,其中,在确定“ph_mvd_l1_zero_flag”已启用并且针对所述当前图片或所述当前图片的条带已启用所述AM模式时,不在比特流中发信号通知指示常规双向预测模式的语法元素。

15. 一种视频编码方法,包括:

由编码器确定当前图片或当前图片的条带已启用高级运动矢量预测AMVP-合并AM模式;

由编码器针对第一参考图片列表LX获取所述当前图片中当前块的AMVP运动矢量MV作为LX MV,其中X为0或1;

由编码器针对第二参考图片列表L(1-X)获取有效合并MV候选列表,其中有效合并MV候选列表包括具有L(1-X)MV的合并候选;以及

由编码器在有效合并MV候选列表中选择所述AM模式的L(1-X)MV。

16. 根据权利要求15所述的方法,其中,无论合并候选的L(1-X)MV的参考图片是否与LX MV的参考图片处于相反方向,具有L(1-X)MV的所述合并候选都被确定为有效。

17. 根据权利要求15所述的方法,其中,在所有有效合并MV候选中,所述AM模式的L(1-X)MV被隐式地选择为在与每个候选相关联的L(1-X)合并预测块和LX AMVP预测块之间具有最小匹配成本的MV。

18. 根据权利要求15所述的方法,其中,在确定只有一个参考图片能够用于LX MV时,不在比特流中发信号通知参考图片索引;或者在确定多于一个参考图片能够用于LX MV时,在比特流中发信号通知参考图片索引。

19. 如权利要求15所述的方法,其中在确定模板匹配功能已启用时,有条件地对具有较高模板匹配成本的LX MV或所述AM模式的L(1-X)MV应用模板匹配MV细化。

20. 如权利要求19所述的方法,其中在所有参考图片具有在所述当前图片之前的显示顺序,或者至少一些参考图片具有在所述当前图片之后的显示顺序的条件下,对具有较高模板匹配成本的LX MV或所述AM模式的L(1-X)MV应用所述模板匹配MV细化。

21. 如权利要求15所述的方法,其中无论模板匹配功能是否已启用,均不对所述AM模式的L(1-X)MV或LX MV应用模板匹配MV细化。

22. 根据权利要求15所述的方法,其中所述AM模式的L(1-X)MV被隐式地选择为LX MV和L(1-X)MV之间具有最小差的MV。

23. 根据权利要求15所述的方法,还包括:

在确定所述LX MV的参考图片和所述AM模式的L(1-X)MV的参考图片不同时,获得所述L(1-X)MV的缩放MV。

24. 根据权利要求23所述的方法,还包括:

基于缩放比率来获得所述缩放MV,所述缩放比率基于所述当前图片与所述LX MV和所述AM模式的L(1-X)MV的所述参考图片之间的时间距离计算。

25. 根据权利要求15所述的方法,其中所述L(1-X)MV在比特流中显式地用信号通知以指示为所述AM模式选择了哪个合并候选。

26. 根据权利要求15所述的方法,其中在确定至少一些参考图片具有在所述当前图片

之后的显示顺序时,具有L (1-X) MV的合并候选被视为无效,其中所述合并候选的L (1-X) MV的参考图片与所述LX MV的参考图片处于同一方向。

27. 根据权利要求15所述的方法,其中,在确定所有参考图片具有在所述当前图片之前的显示顺序时,在所有有效合并MV候选中,仅选择所述AM模式的一个L (1-X) MV,作为与每个候选相关联的L (1-X) 合并预测块和LX AMVP预测块之间具有最小匹配成本的MV,并且不在比特流中发信号通知合并MV索引;并且在确定至少一些参考图片具有在所述当前图片之后的显示顺序时,在比特流中发信号通知合并MV索引以指示所述AM模式的L (1-X) MV。

28. 根据权利要求15所述的方法,其中,在确定“ph_mvd_l1_zero_flag”已启用并且所述当前图片或所述当前图片的条带已启用所述AM模式时,不在比特流中发信号通知指示常规双向预测模式的语法元素。

29. 一种视频解码装置,包括:

一个或多个处理器;和

存储器,耦合到所述一个或多个处理器并被配置为存储可由所述一个或多个处理器执行的指令,

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至14中任一项所述的方法。

30. 一种视频编码装置,包括:

一个或多个处理器;和

存储器,耦合到所述一个或多个处理器并被配置为存储可由所述一个或多个处理器执行的指令,

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15至28中任一项所述的方法。

31. 一种非暂态计算机可读存储介质,用于存储计算机可执行指令,当所述计算机可执行指令由一个或多个计算机处理器执行时,使得所述一个或多个计算机处理器接收比特流,并基于所述比特流执行根据权利要求1至14中任一项所述的方法。

32. 一种非暂态计算机可读存储介质,用于存储计算机可执行指令,当所述计算机可执行指令由一个或多个计算机处理器执行时,使得所述一个或多个计算机处理器执行根据权利要求15至28中任一项所述的方法,以将所述当前块编码为比特流,并传输所述比特流。

视频编解码中的帧间预测

[0001] 相关申请的交叉引用

[0002] 本申请基于2022年3月14日提交的题为“视频编解码中的帧间预测”的美国临时申请第63/319,763号,并要求该临时申请的优先权,其全部内容以引用的方式并入本文以供参考。

技术领域

[0003] 本公开涉及视频编解码和压缩,并且具体地但不限于,视频编解码中的帧间预测的方法和装置。

背景技术

[0004] 可以使用各种视频编解码技术来压缩视频数据。根据一个或多个视频编解码标准进行视频编解码。例如,目前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,也称为H.265或MPEG-H Part2)和高级视频编解码(AVC,也称为H.264或MPEG-4Part 10),它们是由ISO/IEC MPEG和ITU-T VCEG联合开发的。AOMedia Video 1(AV1)是由开放媒体联盟(AOM)开发的,作为其前身标准VP9的后继者。音视频编解码(AVS)是指数字音频和数字视频压缩标准,是由中国音视频编解码标准工作组制定的另一个视频压缩标准系列。现有的大多数视频编解码标准都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,并使用变换编解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。

[0005] HEVC标准的第一个版本于2013年10月完成,与上一代视频编解码标准H.264/MPEG AVC相比,它提供了大约50%的比特率节省或等效的感知质量。尽管HEVC标准比其前身提供了显著的编解码改进,但有证据表明,使用额外的编解码工具可以实现优于HEVC的编解码效率。基于此,VCEG和MPEG都开始了未来视频编解码标准化的新编解码技术的探索工作。2015年10月,ITU-T VCEG和ISO/IEC MPEG成立了一个联合视频探索小组(JVET),开始对能够大幅提高编解码效率的先进技术进行深入研究。JVET通过在HEVC测试模型(HM)之上集成几个附加编解码工具来维护一个称为联合探索测试模型(JEM)的参考软件。

[0006] 2017年10月,ITU-T和ISO/IEC发布了关于具有超越HEVC的能力的视频压缩的联合提案征集(CfP)。2018年4月,第10次JVET会议收到并评估了23份CfP响应,结果表明压缩效率比HEVC提高了约40%。基于此类评估结果,JVET启动了一个新项目,以开发新一代视频编解码标准,该标准被称为通用视频编解码(VVC)。同月,建立了一个参考软件代码库,称为VVC测试模型(VTM),用于演示VVC标准的参考实现。

[0007] 此外,ITU-T VCEG(Q6/16)和ISO/IEC MPEG(JTC 1/SC 29/WG 11)正在研究未来视频编解码技术标准化的潜在需求,该技术的压缩能力将大大超过当前的VVC标准。这种未来的标准化行动可以采取VVC的额外扩展或全新标准的形式。联合视频探索小组(JVET)正在开展这项探索活动,以评估他们在该领域的专家提出的压缩技术设计。第一次探索实验

(EE) 于2021年1月6日至15日在JVET会议上建立,该探索软件模型被命名为增强压缩模型(ECM),ECM版本2(ECM2)于2021年8月发布。

发明内容

[0008] 本公开提供了有关视频编解码中的帧间预测的技术示例,更具体地,提供了有关ECM中的双边匹配AMVP合并模式的技术示例。

[0009] 根据本公开的第一方面,提供了一种视频解码方法。在视频解码方法中,解码器可以确定针对当前图片或当前图片的条带启用高级运动矢量预测(AMVP)-合并(AM)模式。另外,解码器可以针对第一参考图片列表L_X获取所述当前图片中当前块的AMVP运动矢量(MV)作为L_X MV,其中X为0或1。此外,解码器可以针对第二参考图片列表L(1-X)获取有效合并MV候选列表,其中所述有效合并MV候选列表包括具有L(1-X)MV的合并候选。进一步,解码器可以在所述有效合并MV候选列表中选择所述AM模式的L(1-X)MV。

[0010] 根据本公开的第二方面,提供了一种视频编码方法。在视频编码方法中,编码器可以确定当前图片或当前图片的条带已启用高级运动矢量预测(AMVP)-合并(AM)模式。另外,编码器可以针对第一参考图片列表L_X获取所述当前图片中当前块的AMVP运动矢量(MV)作为L_X MV,其中X为0或1。此外,编码器可以针对第二参考图片列表L(1-X)获取有效合并MV候选列表,其中有效合并MV候选列表包括具有L(1-X)MV的合并候选。进一步,编码器可以在有效合并MV候选列表中选择所述AM模式的L(1-X)MV。

[0011] 根据本公开的第三方面,提供了一种视频解码装置。所述装置包括:一个或多个处理器;和存储器,耦合到所述一个或多个处理器并被配置为存储可由所述一个或多个处理器执行的指令。此外,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以上第一方面所述的方法。

[0012] 根据本公开的第二方面,提供了一种视频编码装置。所述装置包括:一个或多个处理器;和存储器,耦合到所述一个或多个处理器并被配置为存储可由所述一个或多个处理器执行的指令。此外,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以上第二方面所述的方法。

[0013] 根据本公开的第五方面,提供了一种非暂态计算机可读存储介质,用于存储计算机可执行指令,当所述计算机可执行指令由一个或多个计算机处理器执行时,使得所述一个或多个计算机处理器接收比特流,并基于所述比特流执行根据第一方面所述的方法。

[0014] 根据本公开的第六方面,提供了一种非暂态计算机可读存储介质,用于存储计算机可执行指令,当所述计算机可执行指令由一个或多个计算机处理器执行时,使得所述一个或多个计算机处理器执行根据第二方面所述的方法,以将所述当前块编码为比特流,并传输所述比特流。

附图说明

[0015] 将参考附图中所示的具体示例对本公开的示例进行更具体的描述。鉴于这些图仅描绘了一些示例,因此不被视为限制范围,将通过使用附图以额外的特异性和细节描述和解释这些示例。

[0016] 图1A是示出根据本公开的一些示例的用于编码和解码视频块的系统的框图。

- [0017] 图1B是示出根据本公开的一些示例的编码器的框图。
- [0018] 图1C至图1F是示出根据本公开的一些示例如何将帧递归地分割为不同尺寸和形状的多个视频块的框图。
- [0019] 图1G是示出根据本公开的一些示例的示例性视频编码器的框图。
- [0020] 图2A是根据本公开的一些示例的解码器的框图。
- [0021] 图2B是示出根据本公开的一些示例的示例性视频解码器的框图。
- [0022] 图3A是示出根据本公开的一些示例的多类型树结构中的块分区的图。
- [0023] 图3B是示出根据本公开的一些示例的多类型树结构中的块分区的图。
- [0024] 图3C是示出根据本公开的一些示例的多类型树结构中的块分区的图。
- [0025] 图3D是示出根据本公开的一些示例的多类型树结构中的块分区的图。
- [0026] 图3E是示出根据本公开的一些示例的多类型树结构中的块分区的图。
- [0027] 图4A示出了根据本公开的一些示例的空间合并候选的位置。
- [0028] 图4B示出了根据本公开的一些示例的用于空间合并候选的冗余校验的候选对。
- [0029] 图4C示出了根据本公开的一些示例的时间合并候选的运动矢量缩放。
- [0030] 图4D示出了根据本公开的一些示例的时间合并候选 C_0 和 C_1 的候选位置。
- [0031] 图5示出了根据本公开的一些示例的MMVD搜索点。
- [0032] 图6示出了根据本公开的一些示例的计算环境。
- [0033] 图7是示出根据本公开的一些示例的视频解码方法的流程图。
- [0034] 图8是示出根据本公开的一些示例的、与图7所示的视频解码方法相对应的视频编码方法的流程图。

具体实施方式

[0035] 现在将详细参照具体实施方式,在附图中示出了具体实施方式的示例。在以下详细描述中,阐述了大量非限制性具体细节以便帮助理解本文呈现的主题。但是在不脱离权利要求的范围的情况下,可以使用各种替代方案,并且可以在没有这些具体细节的情况下实践主题。例如,本文呈现的主题可以在具有数字视频能力的许多类型的电子设备上实现。

[0036] 本公开中使用的术语仅用于描述具体实施例,并不旨在限制本公开。本公开和所附权利要求中的单数形式的“一个”、“所述”和“该”也旨在包括复数形式,除非在整个本公开中明确表示其他含义。还应理解,本公开中使用的术语“和/或”指代并包括列出的多个相关项的一个或任何或所有可能的组合。

[0037] 在本说明书中,对“一个实施例”、“实施例”、“示例”、“一些实施例”、“一些示例”或类似语言的引用意味着所描述的特定特征、结构或特性包括在至少一个实施例或示例中。与一个或一些实施例相关的特征、结构、元素或特性也适用于其他实施例,除非另有明确规定。

[0038] 在整个公开中,术语“第一”、“第二”、“第三”等均仅用作对相关元素(例如,装置、组件、组合物、步骤等)的命名,而不暗示任何空间或时间顺序,除非另有明确规定。例如,“第一装置”和“第二装置”可以指两个单独形成的装置,或同一装置的两个部分、组件或操作状态,并且可以任意命名。

[0039] 术语“模块”、“子模块”、“电路”、“子电路”、“电路系统”、“子电路系统”、“单元”或

“子单元”可以包括存储器(共享、专用或组),其存储可由一个或多个处理器执行的代码或指令。模块可以包括一个或多个具有或不具有存储的代码或指令的电路。模块或电路可以包括一个或多个直接或间接连接的组件。这些组件可以物理地连接到彼此或彼此相邻,或可以不物理地连接到彼此或不彼此相邻。

[0040] 如本文所用,术语“如果”或“当”可以根据上下文理解为“在…时”或“响应于…”。如果这些术语出现在权利要求中,则可能不表示相关限制或特征是有条件的或可选的。例如,方法可以包括以下步骤:i)当条件X存在时或如果条件X存在,执行功能或动作X',以及ii)当条件Y存在时或如果条件Y存在,执行功能或动作Y'。该方法可以实现为既具有执行功能或动作X'的能力,又具有执行功能或动作Y'的能力。因此,功能X'和Y'可以在该方法的多次执行中在不同的时间执行。

[0041] 单元或模块可以通过纯软件、纯硬件或硬件和软件的组合来实现。在纯软件实现中,例如,单元或模块可以包括功能相关的代码块或软件组件,它们直接或间接地链接在一起,以执行特定的功能。

[0042] 图1A是示出了根据本公开的一些实施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1A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、视频流传输设备等。在一些实施方式中,源设备12和目标设备14配备有无线通信能力。

[0043] 在一些实施方式中,目标设备14可以经由链路16接收待解码的编码视频数据。链路16可以包括能够将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能够实时地将编码视频数据直接发送到目标设备14的通信介质。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)频谱或一个或更多个物理传输线。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)的一部分。通信介质可以包括路由器、交换机、基站或可以有利于促进从源设备12到目标设备14的通信的任何其他装置。

[0044] 在其他一些实施方式中,可以将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、数字通用盘(DVD)、压缩盘只读存储器(CD-ROM)、闪存存储器、易失性或非易失性存储器、或者用于存储编码视频数据的任何其他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可以经由流传输或下载从存储设备32访问存储的视频数据。文件服务器可以是能够存储编码视频数据并且将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、网络附属存储(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存储在文件服务器上的编码视频数据的任何标准数据连接来访问编码视频数据,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)或无线信道和有线连接两者的组合。编码视频数据从存储设备32的传输可以是流传输、下载传输或流传输和下载传输两者的组合。

[0045] 如图1A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此类源的组合: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施方式通常可以适用于视频编解码,并且可以应用于无线和/或有线应用。

[0046] 可以由视频编码器20对捕获、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将编码视频数据直接发送到目标设备14。还可以(或备选地)将编码视频数据存储到存储设备32上以便稍后被目标设备14或其他设备访问,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/或发送器。

[0047] 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可以包括接收器和/或调制解调器,并且通过链路16接收编码视频数据。通过链路16传送或在存储设备32上提供的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。此类语法元素可以被包括在通信介质上发送的、存储在存储介质上的或存储在文件服务器上的编码视频数据内。

[0048] 在一些实施方式中,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。

[0049] 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)或此类标准的扩展进行操作。应当理解,本申请不限于特定的视频编码/解码标准,并且可以适用于其他视频编码/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,还通常认为目标设备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。

[0050] 视频编码器20和视频解码器30可以分别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,例如一个或更多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、分立逻辑器件、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态计算机可读介质中,并且使用一个或更多个处理器执行硬件中的指令以执行本公开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更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。

[0051] 与HEVC一样,VVC建立在基于块的混合视频编解码框架之上。图1B是示出根据本公开的一些实施方式的基于块的视频编码器的框图。在编码器100中,输入视频信号被逐块处理,称为编码单元(CU)。编码器100可以是如图1A所示的视频编码器20。在VTM-1.0中,CU可以高达128x128像素。然而,与仅基于二叉树对块进行分割的HEVC不同,在VVC中,一个编码

树单元 (CTU) 被拆分成CU,以适应基于四叉树/二叉树/三叉树的不同局部特征。此外,HEVC中的多分割单元类型的概念被删除,即,VVC中不再存在CU、预测单元 (PU) 和变换单元 (TU) 的分离;而是始终以每个CU作为预测和变换的基本单元,无需进一步分割。在多类型树结构中,一个CTU首先由四叉树结构分割。然后,每个四叉树叶节点可以进一步由二叉树结构和三叉树结构分割。

[0052] 图3A至图3E是示出根据本公开的一些实施方式的多类型树拆分模式的示意图。图3A至图3E分别示出了五种拆分类型,包括四元分割(图3A)、垂直二元分割(图3B)、水平二元分割(图3C)、垂直扩展三元分割(图3D)和水平扩展三元分割(图3E)。

[0053] 对于每个给定的视频块,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/片段中已编码的相邻块的样本(称为参考样本)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(也称为“帧间预测”或“运动补偿预测”)使用来自已编码视频图片的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。给定CU的时间预测信号通常由一个或多个运动矢量(MV)发出信号,这些运动矢量指示当前CU与其时间参考之间的运动量和方向。此外,如果支持多个参考图片,则还会发送一个参考图片索引,用于识别时间预测信号来自参考图片存储中的哪个参考图片。

[0054] 在空间和/或时间预测之后,编码器100中的帧内/帧间模式决策电路121选择最佳预测模式,例如基于速率失真优化方法。然后从当前视频块中减去块预测值120;并使用变换电路102和量化电路104对所得预测残差进行去相关。所得量化残差系数由反量化电路116进行反量化,并由逆变换电路118进行逆变换以形成重建残差,然后将其加回到预测块以形成CU的重建信号。此外,在将重建CU放入图片缓冲器117的参考图片存储器中并用于编码未来视频块之前,可以对重建CU应用环路滤波115,例如去块滤波器、样本自适应偏移(SAO)和/或自适应环路滤波器(ALF)。为了形成输出视频比特流114,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106以进一步压缩和打包以形成比特流。

[0055] 例如,在AVC、HEVC以及VVC的当前版本中,去块滤波器是可用的。在HEVC中,定义了一个称为SAO的附加环路滤波器以进一步提高编码效率。在VVC标准的当前版本中,正在积极研究另一个称为ALF的环路滤波器,并且它很有可能被纳入最终标准。

[0056] 这些环路滤波器操作是可选的。执行这些操作有助于提高编码效率和视觉质量。它们也可以作为编码器100做出的决定而关闭,以节省计算复杂性。

[0057] 应当注意,帧内预测通常基于未滤波的重建像素,而帧间预测则基于滤波的重建像素(如果编码器100开启了这些滤波选项)。

[0058] 图2A是示出基于块的视频解码器200的框图,其可与许多视频编解码标准结合使用。该解码器200类似于图1B的编码器100中的重建相关部分。基于块的视频解码器200可以是如图1A所示的视频解码器30。在解码器200中,首先通过熵解码202对传入的视频比特流201进行解码,以得出量化系数级别和预测相关信息。然后通过反量化204和逆变换206对量化系数级别进行处理,以获得重建的预测残差。在帧内/帧间模式选择器212中实现的块预测值机制被配置为基于解码的预测信息执行帧内预测208或运动补偿210。通过使用加法器214将来自逆变换206的重建预测残差与块预测值机制生成的预测输出相加,获得一组未过

滤的重建像素。

[0059] 在重建块存储在用作参考图片存储器的图片缓冲器213中之前,重建块可以进一步经过环路滤波器209。可以发送图片缓冲器213中的重建视频以驱动显示器设备,以及用于预测未来的视频块。在环路滤波器209开启的情况下,对这些重建像素执行过滤操作以得出最终的重建视频输出222。

[0060] 图1G是示出根据本申请中描述的一些实施方式的另一示例性视频编码器20的框图。视频编码器20可以对视频帧内的视频块执行帧内预测编码和帧间预测编码。帧内预测编码依赖于空间预测以减少或移除给定视频帧或图片内的视频数据中的空间冗余。帧间预测编码依赖于时间预测以减少或移除视频序列的邻近视频帧或图片内的视频数据中的时间冗余。应当注意的是,在视频编解码领域中,术语“帧”可以用作术语“图像”或“图片”的同义词。

[0061] 如图1G所示,视频编码器20包括视频数据存储单元40、预测处理单元41、解码图片缓冲器(DPB)64、加法器50、变换处理单元52、量化单元54和熵编码单元56。预测处理单元41进一步包括运动估计单元42、运动补偿单元44、分割单元45、帧内预测处理单元46和帧内块复制(BC)单元48。在一些实施方式中,视频编码器20还包括用于视频块重建的反量化单元58、逆变换处理单元60和加法器62。诸如去块滤波器的环路滤波器63可以位于加法器62与DPB 64之间以对块边界进行滤波以从重建视频去除块效应。除了去块滤波器之外,还可以使用另一环路滤波器(例如样本自适应偏移(SAO)滤波器和/或自适应环路滤波器(ALF))来对加法器62的输出进行滤波。在一些示例中,可以省略环路滤波器,并且解码视频块可以直接由加法器62提供给DPB 64。视频编码器20可以采取固定或可编程硬件单元的形式,或者可以分散在所说明的固定或可编程硬件单元中的一个或多个中。

[0062] 视频数据存储单元40可以存储将由视频编码器20的组件编码的视频数据。可例如从图1A所示的视频源18获得视频数据存储单元40中的视频数据。DPB 64是存储供视频编码器20(例如,以帧内或帧间预测编码模式)在对视频数据进行编码时使用的参考视频数据(例如,参考帧或图片)的缓冲器。视频数据存储单元40和DPB 64可以由各种存储器设备中的任何存储器设备形成。在各种示例中,视频数据存储单元40可以与视频编码器20的其他组件一起在芯片上,或相对于那些组件在芯片外。

[0063] 如图1G所示,在接收到视频数据之后,预测处理单元41内的分割单元45将视频数据分割为视频块。此分割还可以包括根据与视频数据相关联的预定义的拆分结构(例如二叉树(QT)结构)将视频帧分割为条带、瓦片(tile)(例如,视频块的集合)或其他更大的编码单元(CU)。视频帧是或可以被视为具有样本值的二维样本阵列或矩阵。阵列中的样本也可以被称为像素或图像元素(pixel)。阵列或图片的水平和垂直方向(或轴)上的样本的数量定义了视频帧的尺寸和/或分辨率。例如,可以通过使用QT分割将视频帧划分为多个视频块。视频块再次是或可以被视为具有样本值的二维样本阵列或矩阵,但是其维度小于视频帧的维度。视频块的水平和垂直方向(或轴)上的样本的数量定义了视频块的尺寸。通过例如迭代地使用QT分割、二叉树(BT)分割或二叉树(TT)分割或其任意组合,可以将视频块进一步分割为一个或多个块分区或子块(其可以再次形成块)。应注意,本文所使用的术语“块”或“视频块”可以是帧或图片的一部分,尤其是矩形(正方形或非正方形)部分。参考例如HEVC和VVC,块或视频块可以是或对应于编码树单元(CTU)、CU、预测单元(PU)或变换单元

(TU)和/或可以是或对应于相应的块(例如编码树块(CTB)、编码块(CB)、预测块(PB)或变换块(TB))和/或子块。

[0064] 预测处理单元41可以基于误差结果(例如,编码速率和失真等级)为当前视频块选择多个可行预测编码模式中的一个,例如多个帧内预测编码模式中的一个或多个帧间预测编码模式中的一个。预测处理单元41可以将所得的帧内预测编码块或帧间预测编码块提供给加法器50以生成残差块,并且提供给加法器62以重建编码块以用于随后作为参考帧的一部分使用。预测处理单元41还将语法元素(例如运动矢量、帧内模式指示符、分割信息和其他此类语法信息)提供给熵编码单元56。

[0065] 为了选择用于当前视频块的合适的帧内预测编码模式,预测处理单元41内的帧内预测处理单元46可以与和待编码的当前块在同一帧中的一个或多个邻近块相关地执行当前视频块的帧内预测编码以提供空间预测。预测处理单元41内的运动估计单元42和运动补偿单元44与一个或多个参考帧中的一个或多个预测块相关地执行当前视频块的帧间预测编码以提供时间预测。视频编码器20可以执行多个编码遍次,例如,以为视频数据的每个块选择合适的编码模式。

[0066] 在一些实施方式中,运动估计单元42通过根据视频帧序列内的预定模式生成运动矢量来确定用于当前视频帧的帧间预测模式,运动矢量指示当前视频帧内的视频块相对于参考视频帧内的预测块的位移。由运动估计单元42执行的运动估计是生成运动矢量的过程,该运动矢量估计针对视频块的运动。例如,运动矢量可以指示当前视频帧或图片内的视频块相对于与当前帧内正被编码的当前块相关的参考帧内的预测块的位移。预定模式可以将序列中的视频帧指定为P帧或B帧。帧内BC单元48可以以与由运动估计单元42确定用于帧间预测的运动矢量类似的方式确定用于帧内BC编码的矢量(例如,块矢量),或可以利用运动估计单元42确定块矢量。

[0067] 在像素差方面,针对视频块的预测块可以是或可以对应于被认为与待编码视频块紧密匹配的参考帧的块或参考块,像素差可以由绝对差总和(SAD)、平方差总和(SSD)或其他差度量确定。在一些实施方式中,视频编码器20可以计算用于DPB 64中存储的参考帧的子整数像素位置的值。例如,视频编码器20可以对参考帧的四分之一像素位置、八分之一像素位置或其他分数像素位置的值进行内插。因此,运动估计单元42可以相对于全像素位置和分数像素位置执行运动搜索并且输出具有分数像素精度的运动矢量。

[0068] 运动估计单元42通过以下方式来计算针对帧间预测编码帧中的视频块的运动矢量:将视频块的位置与从第一参考帧列表(列表0)或第二参考帧列表(列表1)选择的参考帧的预测块的位置进行比较,第一参考帧列表和第二参考帧列表中的每一个参考帧列表标识DPB 64中存储的一个或多个参考帧。运动估计单元42将计算出的运动矢量发送到运动补偿单元44,然后发送到熵编码单元56。

[0069] 由运动补偿单元44执行的运动补偿可以涉及基于由运动估计单元42确定的运动矢量获取或生成预测块。在接收到针对当前视频块的运动矢量之后,运动补偿单元44可以在参考帧列表中的一个参考帧列表中定位运动矢量所指向的预测块,从DPB 64取回预测块,并且将预测块转发到加法器50。然后,加法器50通过从正被编码的当前视频块的像素值减去由运动补偿单元44提供的预测块的像素值来形成像素差值的残差视频块。形成残差视频块的像素差值可以包括亮度分量差或色度分量差或两者。运动补偿单元44还可以生成与

视频帧的视频块相关联的语法元素以供视频解码器30在对视频帧的视频块进行解码时使用。语法元素可以包括例如定义用于识别预测块的运动矢量的语法元素、指示预测模式的任何标志、或本文描述的任何其他语法信息。应注意,运动估计单元42和运动补偿单元44可以高度集成,但出于概念目的而单独说明。

[0070] 在一些实施方式中,帧内BC单元48可以以与上文结合运动估计单元42和运动补偿单元44所描述的方式类似的方式生成矢量并获取预测块,但是这些预测块在与正被编码的当前块相同的帧中,并且这些矢量被称为块矢量而非运动矢量。具体地,帧内BC单元48可以确定将用于对当前块进行编码的帧内预测模式。在一些示例中,帧内BC单元48可以例如在单独的编码遍次期间使用各种帧内预测模式来对当前块进行编码,并且通过率失真分析来测试它们的性能。接下来,帧内BC单元48可以在各种测试的帧内预测模式中选择合适的帧内预测模式来使用并相应地生成帧内模式指示符。例如,帧内BC单元48可以使用率失真分析针对各种测试的帧内预测模式计算率失真值,并且在测试的模式中选择具有最佳率失真特性的帧内预测模式作为合适的帧内预测模式来使用。率失真分析大体上确定编码块与被编码以产生编码块的原始未编码块之间的失真(或误差)量、以及用于产生编码块的比特率(即,比特数量)。帧内BC单元48可以根据针对各种编码块的失真和速率计算比率,以确定哪个帧内预测模式展现针对块的最佳率失真值。

[0071] 在其他示例中,帧内BC单元48可以全部或部分地使用运动估计单元42和运动补偿单元44来执行根据本文描述的実施方式的用于帧内BC预测的此类功能。在任一情况下,对于帧内块复制,在像素差方面,预测块可以是被认为与待编码的块紧密匹配的块,像素差可以由SAD、SSD或其他差度量确定,并且识别预测块可以包括计算针对子整数像素位置的值。

[0072] 无论预测块是来自根据帧内预测的同一帧还是来自根据帧间预测的不同帧,视频编码器20可以通过从正被编码的当前视频块的像素值减去预测块的像素值来形成像素差值,从而形成残差视频块。形成残差视频块的像素差值可以包括亮度分量差和色度分量差两者。

[0073] 作为如上文所描述的由运动估计单元42和运动补偿单元44执行的帧间预测或由帧内BC单元48执行的帧内块复制预测的替代方案,帧内预测处理单元46可以对当前视频块进行帧内预测。具体地,帧内预测处理单元46可以确定帧内预测模式以用于对当前块进行编码。为此,帧内预测处理单元46可以例如在单独的编码遍次期间使用各种帧内预测模式来对当前块进行编码,并且帧内预测处理单元46(或在一些示例中,模式选择单元)可以从测试的帧内预测模式中选择合适的帧内预测模式来使用。帧内预测处理单元46可以将指示针对块选择的帧内预测模式的信息提供给熵编码单元56。熵编码单元56可以将指示选择的帧内预测模式的信息编码到比特流中。

[0074] 在预测处理单元41经由帧间预测或帧内预测确定针对当前视频块的预测块之后,加法器50通过从当前视频块减去预测块来形成残差视频块。残差块中的残差视频数据可以被包括在一个或更多个TU中并且提供给变换处理单元52。变换处理单元52使用变换(例如离散余弦变换(DCT)或概念上类似的变换)将残差视频数据变换为残差变换系数。

[0075] 变换处理单元52可以将所得变换系数发送到量化单元54。量化单元54对变换系数进行量化以进一步减小比特率。量化过程还可以减小与系数中的一些或全部相关联的比特深度。可以通过调整量化参数来修改量化程度。在一些示例中,量化单元54可以随后对包括

量化的变换系数的矩阵执行扫描。备选地,熵编码单元56可以执行扫描。

[0076] 在量化之后,熵编码单元56使用例如上下文自适应可变长度编码(CAVLC)、上下文自适应二进制算术编码(CABAC)、基于语法的上下文自适应二进制算术编码(SBAC)、概率区间分割熵(PIPE)编码或另一熵编码方法或技术,将量化的变换系数熵编码成视频比特流。然后,可以将编码的比特流发送到如图1A所示的视频解码器30,或存档于如图1A所示的存储设备32中,以供稍后发送到视频解码器30或由视频解码器30取回。熵编码单元56还可以对用于正被编码的当前视频帧的运动矢量和其他语法元素进行熵编码。

[0077] 反量化单元58和逆变换处理单元60分别应用反量化和逆变换以在像素域中重建残差视频块以用于生成用于预测其他视频块的参考块。如上文指出的,运动补偿单元44可以从存储在DPB 64中的帧的一个或多个参考块生成运动补偿预测块。运动补偿单元44还可以将一个或多个内插滤波器应用于预测块以计算子整数像素值以用于在运动估计时使用。

[0078] 加法器62将重建的残差块与由运动补偿单元44产生的运动补偿预测块相加来产生参考块以存储在DPB 64中。然后,参考块可以由帧内BC单元48、运动估计单元42和运动补偿单元44用作预测块以对后续视频帧中的另一视频块进行帧间预测。

[0079] 图2B是示出了根据本申请的一些实施方式的示例性视频解码器30的框图。视频解码器30包括视频数据存储器79、熵解码单元80、预测处理单元81、反量化单元86、逆变换处理单元88、加法器90和DPB 92。预测处理单元81进一步包括运动补偿单元82、帧内预测单元84和帧内BC单元85。视频解码器30可以执行与上文结合图1G关于视频编码器20所描述的编码过程基本互逆的解码过程。例如,运动补偿单元82可以基于从熵解码单元80接收的运动矢量生成预测数据,而帧内预测单元84可以基于从熵解码单元80接收的帧内预测模式指示符生成预测数据。

[0080] 在一些示例中,视频解码器30的单元可以被分派任务以执行本申请的实施方式。此外,在一些示例中,本公开的实施方式可以分散在视频解码器30的单元中的一个或多个单元中。例如,帧内BC单元85可以单独地或与视频解码器30的其他单元(例如运动补偿单元82、帧内预测单元84和熵解码单元80)组合地执行本申请的实施方式。在一些示例中,视频解码器30可以不包括帧内BC单元85,并且帧内BC单元85的功能可以由预测处理单元81的其他组件(例如运动补偿单元82)执行。

[0081] 视频数据存储器79可以存储将由视频解码器30的其他组件进行解码的视频数据,例如编码视频比特流。可以例如从存储设备32、从本地视频源(例如相机)、经由视频数据的有线或无线网络通信、或通过访问物理数据存储介质(例如,闪存驱动器或硬盘)获得存储在视频数据存储器79中的视频数据。视频数据存储器79可以包括存储来自编码视频比特流的编码视频数据的编码图片缓冲器(CPB)。视频解码器30的DPB 92存储参考视频数据以供视频解码器30(例如,以帧内或帧间预测编码模式)在对视频数据进行解码时使用。视频数据存储器79和DPB 92可以由各种存储器设备中的任何存储器设备形成,例如动态随机存取存储器(DRAM)(包括同步DRAM(SDRAM))、磁阻式RAM(MRAM)、电阻式RAM(RRAM)或其他类型的存储器设备。出于说明性目的,视频数据存储器79和DPB 92在图2B中被描绘为视频解码器30的两个不同组件。但是对于本领域的技术人员将显而易见的是,视频数据存储器79和DPB 92可以由同一存储器设备或单独存储器设备提供。在一些示例中,视频数据存储器79可以

与视频解码器30的其他组件一起在芯片上,或相对于那些组件在芯片外。

[0082] 在解码过程期间,视频解码器30接收表示编码视频帧的视频块和相关联的语法元素的编码视频比特流。视频解码器30可以在视频帧级和/或视频块级接收语法元素。视频解码器30的熵解码单元80对比特流进行熵解码以生成量化系数、运动矢量或帧内预测模式指示符、以及其他语法元素。然后,熵解码单元80将运动矢量或帧内预测模式指示符、以及其他语法元素转发到预测处理单元81。

[0083] 当视频帧被编码为帧内预测编码(I)帧或用于其他类型的帧中的帧内编码预测块时,预测处理单元81的帧内预测单元84可以基于用信号传送的帧内预测模式和来自当前帧的先前解码块的参考数据来生成用于当前视频帧的视频块的预测数据。

[0084] 当视频帧被编码为帧间预测编码(即,B或P)帧时,预测处理单元81的运动补偿单元82基于从熵解码单元80接收的运动矢量和其他语法元素产生针对当前视频帧的视频块的一个或多个预测块。预测块中的每一个可以从参考帧列表中的一个参考帧列表内的参考帧产生。视频解码器30可以基于存储在DPB 92中的参考帧使用默认构建技术来构建参考帧列表,即,列表0和列表1。

[0085] 在一些示例中,当根据本文描述的帧内BC模式对视频块进行编码时,预测处理单元81的帧内BC单元85基于从熵解码单元80接收的块矢量和其他语法元素产生针对当前视频块的预测块。预测块可以在由视频编码器20定义的与当前视频块相同的图片的重建区域内。

[0086] 运动补偿单元82和/或帧内BC单元85通过解析运动矢量和其他语法元素来确定针对当前视频帧的视频块的预测信息,然后使用该预测信息产生针对正被解码的当前视频块的预测块。例如,运动补偿单元82使用接收到的语法元素中的一些语法元素来确定用于对视频帧的视频块进行编码的预测模式(例如,帧内预测或帧间预测)、帧间预测帧类型(例如,B或P)、用于针对帧的参考帧列表中的一个或多个的构建信息、用于帧的每个帧间预测编码视频块的运动矢量、用于帧的每个帧间预测编码视频块的帧间预测状态、以及用于对当前视频帧中的视频块进行解码的其他信息。

[0087] 类似地,帧内BC单元85可以使用接收到的语法元素中的一些语法元素,例如标志,以确定当前视频块是使用帧内BC模式预测的、帧的哪些视频块在重建区域内且应被存储在DPB 92中的构建信息、用于帧的每个帧内BC预测视频块的块矢量、用于帧的每个帧内BC预测视频块的帧内BC预测状态、以及用于对当前视频帧中的视频块进行解码的其他信息。

[0088] 运动补偿单元82还可以使用如由视频编码器20在对视频块进行编码期间使用的内插滤波器执行内插,以计算针对参考块的子整数像素的内插值。在这种情况下,运动补偿单元82可以根据接收到的语法元素确定由视频编码器20使用的内插滤波器,并且使用这些内插滤波器来产生预测块。

[0089] 反量化单元86使用由视频编码器20针对视频帧中的每个视频块计算出的用于确定量化程度的相同量化参数,对在比特流中提供且由熵解码单元80熵解码的量化的变换系数进行反量化。逆变换处理单元88将逆变换(例如,逆DCT、逆整数变换或概念上类似的逆变换过程)应用于变换系数,以便在像素域中重建残差块。

[0090] 在运动补偿单元82或帧内BC单元85基于矢量和和其他语法元素生成针对当前视频块的预测块之后,加法器90通过将来自逆变换处理单元88的残差块与由运动补偿单元82和

帧内BC单元85生成的对应预测块相加,来重建针对当前视频块的解码视频块。环路滤波器91(例如去块滤波器、SAO滤波器和/或ALF)可以位于加法器90与DPB 92之间以进一步处理解码视频块。在一些示例中,可以省略环路滤波器91,并且解码视频块可以直接由加法器90提供给DPB 92。然后,将给定帧中的解码视频块存储在DPB 92中,DPB 92存储用于接下来的视频块的后续运动补偿的参考帧。DPB 92或与DPB 92分离的存储器设备还可以存储解码视频以用于稍后呈现在显示器设备(例如,图1A的显示器设备34)上。

[0091] 在当前VVC标准中,当前编码块的运动信息是从合并候选索引指定的空间或时间邻近块复制而来,或者通过显式地用信号传送的运动估计而获得。

[0092] 在典型的视频编码过程中,视频序列通常包括帧或图片的有序集合。每一帧可以包括三个样本阵列,表示为SL、SCb和SCr。SL是亮度样本的二维阵列。SCb是Cb色度样本的二维阵列。SCr是Cr色度样本的二维阵列。在其他实例中,帧可以是单色的,因此仅包括亮度样本的一个二维阵列。

[0093] 如图1C所示,视频编码器20(或更具体地,视频编码器20的预测处理单元中的分割单元)通过首先将帧分割为CTU的集合来生成帧的编码表示。视频帧可以包括以光栅扫描顺序从左到右和从上到下连续排序的整数个CTU。每个CTU是最大的逻辑编码单元,并且由视频编码器20在序列参数集中用信号传送CTU的宽度和高度,使得视频序列中的所有CTU具有 128×128 、 64×64 、 32×32 和 16×16 之一的相同尺寸。但是应当注意,本申请不必限于特定尺寸。如图1D中所示,每个CTU可以包括亮度样本的一个CTB、色度样本的两个对应编码树块、以及用于对编码树块的样本进行编码的语法元素。语法元素描述编码像素块的不同类型的单元的性质以及可以如何在视频解码器30处重建视频序列,包括帧间预测或帧内预测、帧内预测模式、运动矢量和其他参数。在单色图片或具有三个单独颜色平面的图片中,CTU可以包括单个编码树块和用于对该编码树块的样本进行编码的语法元素。编码树块可以是 $N \times N$ 的样本块。

[0094] 为了实现更好的性能,视频编码器20可以对CTU的编码树块递归地执行树分割,例如二叉树分割、三叉树分割、四叉树分割或其组合,并且将CTU划分为较小的CU。如图1E中所描绘的,首先将 64×64 的CTU 400划分为四个较小的CU,每个CU具有 32×32 的块尺寸。在四个较小的CU中,将CU 410和CU 420分别划分为块尺寸为 16×16 的四个CU。将两个 16×16 的CU 430和CU 440分别进一步划分为块尺寸为 8×8 的四个CU。图1F描绘了示出如图1E中所描绘的CTU 400的分割过程的最终结果的四叉树数据结构,四叉树的每个叶节点与范围从 32×32 到 8×8 的相应尺寸的一个CU对应。类似于图1D中描绘的CTU,每个CU可以包括相同尺寸的帧的亮度样本的CB和色度样本的两个对应编码块、以及用于对编码块的样本进行编码的语法元素。在单色图片或具有三个单独颜色平面的图片中,CU可以包括单个编码块和用于对编码块的样本进行编码的语法结构。应注意,图1E和图1F中所描绘的四叉树分割仅用于说明性目的,并且一个CTU可以基于四叉树分割/三叉树分割/二叉树分割而被拆分为多个CU以适应于变化的局部特性。在多类型树结构中,一个CTU按照四叉树结构被分割,并且每个四叉树叶CU可以按照二叉和三叉树结构被进一步分割。如图3A至图3E所示,具有宽度W和高度H的编码块有五种可能的分割类型,即四元分割、垂直二元分割、水平二元分割、垂直扩展三元分割和水平扩展三元分割。

[0095] 在一些实施方式中,视频编码器20可以进一步将CU的编码块分割为一个或更多个

$M \times N$ PB。PB是被应用相同预测(帧间或帧内)的矩形(正方形或非正方形)样本块。CU的PU可以包括亮度样本的PB、色度样本的两个对应PB和用于对PB进行预测的语法元素。在单色图片或具有三个单独颜色平面的图片中,PU可以包括单个PB和用于对PB进行预测的语法结构。视频编码器20可以生成针对CU的每个PU的亮度PB、Cb PB和Cr PB的预测亮度块、预测Cb块和预测Cr块。

[0096] 视频编码器20可以使用帧内预测或帧间预测来生成针对PU的预测块。如果视频编码器20使用帧内预测来生成PU的预测块,则视频编码器20可以基于与PU相关联的帧的解码样本来生成PU的预测块。如果视频编码器20使用帧间预测来生成PU的预测块,则视频编码器20可以基于除与PU相关联的帧之外的一个或更多个帧的解码样本来生成PU的预测块。

[0097] 在视频编码器20生成针对CU的一个或更多个PU的预测亮度块、预测Cb块和预测Cr块之后,视频编码器20可以通过从CU的原始亮度编码块减去CU的预测亮度块来生成针对CU的亮度残差块,使得CU的亮度残差块中的每个样本指示CU的预测亮度块之一中的亮度样本与CU的原始亮度编码块中的对应样本之差。类似地,视频编码器20可以分别生成针对CU的Cb残差块和Cr残差块,使得CU的Cb残差块中的每个样本指示CU的预测Cb块之一中的Cb样本与CU的原始Cb编码块中的对应样本之差,并且CU的Cr残差块中的每个样本可以指示CU的预测Cr块之一中的Cr样本与CU的原始Cr编码块中的对应样本之差。

[0098] 此外,如图1E中所示,视频编码器20可以使用二叉树分割将CU的亮度残差块、Cb残差块和Cr残差块分别分解成一个或更多个亮度变换块、Cb变换块和Cr变换块。变换块是被应用相同变换的矩形(正方形或非正方形)样本块。CU的TU可以包括亮度样本的变换块、色度样本的两个对应变换块和用于对变换块样本进行变换的语法元素。因此,CU的每个TU可以与亮度变换块、Cb变换块和Cr变换块相关联。在一些示例中,与TU相关联的亮度变换块可以是CU的亮度残差块的子块。Cb变换块可以是CU的Cb残差块的子块。Cr变换块可以是CU的Cr残差块的子块。在单色图片或具有三个单独颜色平面的图片中,TU可以包括单个变换块和用于对该变换块的样本进行变换的语法结构。

[0099] 视频编码器20可以将一个或更多个变换应用于TU的亮度变换块以生成针对TU的亮度系数块。系数块可以是变换系数的二维阵列。变换系数可以是标量。视频编码器20可以将一个或更多个变换应用于TU的Cb变换块以生成针对TU的Cb系数块。视频编码器20可以将一个或更多个变换应用于TU的Cr变换块以生成针对TU的Cr系数块。

[0100] 在生成系数块(例如,亮度系数块、Cb系数块或Cr系数块)之后,视频编码器20可以对系数块进行量化。量化通常是指变换系数被量化以可能减少用于表示变换系数的数据量从而提供进一步压缩的过程。在视频编码器20对系数块进行量化之后,视频编码器20可以对指示量化的变换系数的语法元素进行熵编码。例如,视频编码器20可以对指示量化的变换系数的语法元素执行CABAC。最后,视频编码器20可以输出包括比特序列的比特流,比特序列形成编码帧和相关联数据的表示,比特流被保存于存储设备32中或被发送到目标设备14。

[0101] 在接收到由视频编码器20生成的比特流之后,视频解码器30可以解析比特流以从比特流获得语法元素。视频解码器30可以至少部分地基于从比特流获得的语法元素来对视频数据的帧进行重建。对视频数据进行重建的过程通常与由视频编码器20执行的编码过程互逆。例如,视频解码器30可以对与当前CU的TU相关联的系数块执行逆变换以重建与当前

CU的TU相关联的残差块。视频解码器30还通过将针对当前CU的PU的预测块的样本与当前CU的TU的变换块的对应样本相加,来重建当前CU的编码块。在重建针对帧的每个CU的编码块之后,视频解码器30可以重建帧。

[0102] 如上所述,视频编码主要使用两种模式(即,帧内部预测(或帧内预测)和帧之间预测(或帧间预测))来实现视频压缩。应注意,IBC可以被视为帧内预测或第三模式。在两种模式之间,由于使用运动矢量来根据参考视频块预测当前视频块,所以帧间预测比帧内预测对编码效率的贡献更大。

[0103] 但是随着不断改进的视频数据捕获技术和用于保留视频数据中的细节的更精细的视频块尺寸,表示用于当前帧的运动矢量所需的数据量也大幅增加。克服此挑战的一种方式受益于以下事实:不仅空间域和时间域两者中的一组邻近CU具有用于预测目的的相似视频数据,而且这些邻近CU之间的运动矢量也是相似的。因此,可以通过以下方式而使用空间邻近CU和/或时间同位CU的运动信息作为当前CU的运动信息(例如,运动矢量)的近似(其也被称为当前CU的“运动矢量预测值”(MVP)):探索它们的空间和时间相关性。

[0104] 代替如上文结合图1B所描述的将由运动估计单元确定的当前CU的实际运动矢量编码到视频比特流中,从当前CU的实际运动矢量减去当前CU的运动矢量预测值以产生针对当前CU的运动矢量差(MVD)。通过这样做,不需要将由运动估计单元针对帧的每个CU确定的运动矢量编码到视频比特流中,并且可以显著减少用于表示视频比特流中的运动信息的数据量。

[0105] 类似于在编码块的帧间预测期间选择参考帧中的预测块的过程,视频编码器20和视频解码器30两者需要采用一组规则,以用于使用与当前CU的空间邻近CU和/或时间同位CU相关联的那些潜在候选运动矢量来构建针对当前CU的运动矢量候选列表(也称为“合并列表”),然后从运动矢量候选列表选择一个成员作为针对当前CU的运动矢量预测值。通过这样做,不需要从视频编码器20向视频解码器30发送运动矢量候选列表本身,并且运动矢量候选列表内的所选运动矢量预测值的索引足以使视频编码器20和视频解码器30使用运动矢量候选列表内的相同运动矢量预测值来对当前CU进行编码和解码。

[0106] 对于每个帧间预测CU,运动参数(包括运动向量、参考图片索引和参考图片列表使用索引)以及VVC的新编码特征所需的附加信息可用于帧间预测样本生成。运动参数可以以显式或隐式方式用信号传送。当使用跳过模式对CU进行编码时,CU与一个PU相关联并且没有显著残差系数、没有编码的运动向量增量(运动向量差)或参考图片索引。指定合并模式,由此从邻近CU(包括空间和时间候选)以及VVC中引入的附加调度获得当前CU的运动参数。合并模式可以应用于任何帧间预测CU,而不仅仅是跳过模式。合并模式的替代方案是显式传输运动参数,其中运动向量、每个参考图片列表的对应参考图片索引和参考图片列表使用标志以及其他所需信息针对每个CU显式地用信号传送。

[0107] 扩展合并模式

[0108] 在一些实施例中,合并候选列表通过按顺序包括以下五种类型的候选来构建:

[0109] (1) 来自空间邻近CU的空间MVP;

[0110] (2) 来自同位CU的时间MVP;

[0111] (3) 来自先进先出(FIFO)表的基于历史的MVP;

[0112] (4) 成对平均MVP;和

[0113] (5) 零MV。

[0114] 在序列参数集头部中发信号通知合并列表的大小,并且合并列表的最大允许大小为6。对于以合并模式编码的每个CU,最佳合并候选的索引是使用截断一元二值化(TU)进行编码的。合并索引的第一个二进制位是使用上下文进行编码的,并且针对其他二进制位使用旁路编码。

[0115] 上面提供了得到每种合并候选类别的过程。在一些实施例中,可以支持对一定大小的区域内的所有CU并行地得到合并候选列表。

[0116] 得到空间候选

[0117] 在VVC中得到空间合并候选与HEVC中的相同,除了前两个合并候选的位置发生了交换。在图4A中所示的位置的候选中选择最多四个合并候选。得到的顺序为B0、A0、B1、A1和B2。仅当位置B0、A0、B1、A1的一个或多于一个CU不可用(例如,因为它属于另一个条带或瓦片)或被帧内编码时,才考虑位置B2。在添加位置A1的候选之后,剩余候选的添加要经过冗余校验,这确保具有相同运动信息的候选被排除在列表之外,从而提高编解码效率。为了降低计算复杂度,在上述冗余校验中并未考虑所有可能的候选对。取而代之地,只考虑图4B中使用箭头链接的对,并且仅当用于冗余校验的对应候选不具有相同的运动信息时,才将候选添加到列表中。图4B示出了用于空间合并候选的冗余校验的候选对。

[0118] 得到时间候选

[0119] 在此步骤中,仅将一个候选添加到列表中。具体地,在得到该时间合并候选时,基于属于同位参考图片的同位CU得到缩放运动矢量。在条带头部中显式地发信号通知用于得到同位CU的参考图片列表和参考索引。如图4C中的虚线所示,获得时间合并候选的缩放的运动矢量,其使用POC距离tb和td从同位CU的运动矢量缩放,其中tb被定义为当前图片的参考图片与当前图片之间的POC差异,而td被定义为同位图片的参考图片与同位图片之间的POC差异。时间合并候选的参考图片索引被设置为等于零。

[0120] 如图4D所示,在候选C0和C1之间选择时间候选的位置。如果位置C0处的CU不可用、被帧内编码或位于当前CTU行之外,则使用位置C1。否则,在得到时间合并候选中,使用位置C0。

[0121] 基于历史的合并候选获取

[0122] 在空间MVP和时间运动矢量预测(TMVP)之后,将基于历史的MVP(HMVP)合并候选添加到合并列表中。在该方法中,先前编码的块的运动信息存储在表中并用作当前CU的MVP。在编码/解码过程中维持具有多个HMVP候选的表。当遇到新的CTU行时,该表被重置(清空)。每当存在非子块帧间编码的CU时,将相关联的运动信息作为新的HMVP候选添加到表的最后一个条目。

[0123] HMVP表大小S可以设置为6,这表示最多可以将5个基于历史的MVP(HMVP)候选添加到表中。当将新的运动候选插入表中时,利用约束先进先出(FIFO)规则,其中首先应用冗余校验来查找表中是否存在相同的HMVP。如果找到,则从表中移除相同的HMVP,并将之后的所有HMVP候选向前移动,并将相同的HMVP插入到表的最后一个条目。

[0124] 可以在合并候选列表构建过程中使用HMVP候选。表中最新的几个HMVP候选被依次检查,并在TMVP候选之后被插入到候选列表中。相对于空间或时间合并候选,对HMVP候选应用冗余校验。

[0125] 为了减少冗余校验操作的次数,引入了以下简化。首先,分别相对于A1和B1空间候选,对表中的最后两个条目进行冗余校验。其次,一旦可用的合并候选总数达到最大允许合并候选减1,就终止根据HMVP构建合并候选列表的过程。

[0126] 得到成对平均合并候选

[0127] 通过对使用前两个合并候选对现有合并候选列表中的预定义候选对进行平均来生成成对平均候选。第一合并候选被定义为p0Cand,第二合并候选可以被定义为p1Cand。单独地针对每个参考列表,根据p0Cand和p1Cand的运动矢量的可用性计算平均运动矢量。如果两个运动矢量在一个列表中都可用,则即使这两个运动矢量指向不同的参考图片,也会对它们进行平均,并将其参考图片设置为p0Cand的参考图片;如果只有一个运动矢量可用,则直接使用该运动矢量;如果没有可用的运动矢量,则保持此列表无效。此外,如果p0Cand和p1Cand的半像素插值滤波器索引不同,则将其设置为0。

[0128] 当在添加成对平均合并候选之后合并列表未满足时,在合并列表的末尾插入零MVP,直到遇到最大合并候选数。

[0129] 高精度(1/16像素)运动补偿和运动矢量存储

[0130] 在VVC标准中,MV精度提高到1/16亮度样本,以提高慢动作视频的预测效率。这种更高的运动精度对于具有局部变化和非平移运动的视频内容(例如在仿射模式的情况下)特别有用。为了生成更高MV精度的分数位置样本,HEVC的8抽头亮度插值滤波器和4抽头色度插值滤波器被扩展为亮度16相和色度32相。此扩展滤波器组应用于除仿射模式下的CU之外的帧间编码的CU的MC过程。对于仿射模式,使用一组具有16相的6抽头亮度插值滤波器以降低计算复杂度并节省内存带宽。

[0131] 非仿射CU的显式地用信号传送运动矢量的最高精度为四分之一亮度样本。在某些帧间预测模式(例如仿射模式)中,运动矢量可以以1/16亮度样本精度用信号传送。在具有隐式推断的MV的所有帧间编码的CU中,以1/16亮度样本精度得到MV,并且以1/16样本精度执行运动补偿预测。在内部运动场存储方面,所有运动矢量均以1/16亮度样本精度存储。

[0132] 对于TMVP和SbTVMP使用的时间运动场存储,运动场压缩以 8×8 大小粒度执行,而HEVC中为 16×16 大小粒度。

[0133] 利用MVD的合并模式(MMVD)

[0134] 除了合并模式之外,其中隐式得到的运动信息直接用于当前CU的预测样本生成,还引入了利用运动矢量差的合并模式(MMVD)。在发送常规合并标志之后立即发信号通知MMVD标志以指定针对CU是否使用MMVD模式。在选择合并候选之后,通过发信号通知的MVD信息对其进行进一步细化。进一步的信息包括合并候选标志、指定运动幅度的索引和指示运动方向的索引。在MMVD模式下,选择合并列表中的前两个候选中的一个作为MV基础。发信号通知MMVD候选标志以指定在第一和第二合并候选之间使用哪一个。

[0135] 距离索引指定运动幅度信息并指示与起点的预定义偏移。如图5所示,偏移被添加到起始MV的水平分量或垂直分量。距离索引与预定义偏移量的关系如表1所示。

[0136] 表1距离索引与预定义偏移量的关系

	距离索引	0	1	2	3	4	5	6	7
[0137]	偏移（以亮度样本为单位）	1/4	1/2	1	2	4	8	16	32

[0138] 方向索引表示MVD相对于起始点的方向。方向索引可以表示如表2所示的四个方向。需要注意的是，MVD符号的含义可以根据起始MV的信息而变化。当起始MV是非预测MV或双向预测MV且两个列表都指向当前图片的同一侧（即，两个参考的POC都大于当前图片的POC，或都小于当前图片的POC）时，表2中的符号指定添加到起始MV的MV偏移量的符号。当起始MV为双向预测MV，且两个MV指向当前图片的不同侧（即，一个参考的POC大于当前图片的POC，而另一个参考的POC小于当前图片的POC），且列表0中的POC差大于列表1中的POC差时，表2中的符号指定添加到起始MV的列表0MV分量的MV偏移的符号，而列表1MV的符号具有相反的值。否则，如果列表1中的POC差大于列表0，则表2中的符号指定添加到起始MV的列表1MV分量的MV偏移的符号，而列表0MV的符号具有相反的值。

[0139] 根据每个方向上的POC差缩放MVD。如果两个列表中的POC差相同，则无需缩放。否则，如果列表0中的POC差大于列表1中的POC差，则通过将L0的POC差定义为td，将L1的POC差定义为tb，来缩放列表1的MVD，如图4C所示。如果L1的POC差大于L0，则以相同方式缩放列表0的MVD。如果起始MV是单预测的，则将MVD添加到可用MV中。

[0140] 表2方向索引指定的MV偏移的符号

[0141]	方向索引	00	01	10	11
	x轴	+	-	N/A	N/A
	y轴	N/A	N/A	+	-

[0142] 对称MVD编码

[0143] 除了正常的单向预测和双向预测模式MVD信令之外，还可以应用针对双向预测MVD信令的对称MVD模式。在对称MVD模式下，包括列表0和列表1的参考图片索引以及列表1的MVD的运动信息不是发信号通知的，而是推断出的。

[0144] 对称MVD模式的解码过程如下：

[0145] 在条带级，按如下方式得到变量BiDirPredFlag、RefIdxSymL0和RefIdxSymL1：

[0146] -如果mvd_11_zero_flag为1，则BiDirPredFlag被设置为等于0。

[0147] -否则，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参考图片的前后向对或参考图片的后前向对，则BiDirPredFlag被设置为1，并且列表-0和列表-1参考图片均为短期参考图片。否则，BiDirPredFlag被设置为0。

[0148] 在CU级，如果CU是双向预测编码且BiDirPredFlag等于1，则显示地发信号通知指示是否使用对称模式的对称模式标志。

[0149] 当对称模式标志为真时，仅显示地发信号通知mvp_10_flag、mvp_11_flag和MVD0。列表-0和列表-1的参考索引分别被设置为等于参考图片对。MVD1被设置为(-MVD0)。最终运动矢量如下公式所示。

$$[0150] \quad \begin{cases} (mvx_0, mvy_0) = (mvp_x_0 + mvd_x_0, mvp_y_0 + mvd_y_0) \\ (mvx_1, mvy_1) = (mvp_x_1 - mvd_x_0, mvp_y_1 - mvd_y_0) \end{cases} \quad (1)$$

[0151] 在编码器中,对称MVD运动估计从初始MV评估开始。一组初始MV候选包括从单预测搜索获得的MV、从双预测搜索获得的MV和来自AMVP列表的MV。具有最低率失真成本的MV被选为对称MVD运动搜索的初始MV。

[0152] 双向匹配AMVP合并模式

[0153] 在双向匹配AMVP合并模式(在本公开中也称为AM模式)中,双向预测值由一个方向的AMVP预测值和另一个方向的合并预测值组成。当所选合并预测值和AMVP预测值满足相对于当前图片至少有一个来自过去的参考图片和一个来自未来的参考图片的条件时,可以对编码块启用该模式。

[0154] AM模式的AMVP部分被发信号通知为常规单向AMVP,即发信号通知参考索引和MVD,并且如果使用模板匹配则它具有推断出的MVP索引,或者当模板匹配被禁用时发信号通知MVP索引。

[0155] 对于AMVP方向LX(X可以是0或1;L代表参考图片列表),另一个方向L(1-X)中的合并部分通过最小化AMVP预测值和合并预测值之间的双边匹配成本(即,对于AMVP和合并运动向量对)而隐式得出。对于合并候选列表中具有该另一个方向L(1-X)运动向量的每个合并候选,使用合并候选MV和AMVP MV计算双边匹配成本。然后根据双边匹配成本对合并候选进行排序。然后将合并索引发信号传送到比特流中,以指示在前2个排序候选中选择哪个合并候选。AM模式的L(1-X)mv合并候选列表的得出基本上遵循前面段落中描述的常规合并列表构造的类似过程。编码器或解码器搜索不同的合并候选(包括空间、时间、HMVP、对平均和零候选),并将不同合并候选的有效L(1-X)mv插入到AM模式的mv合并候选列表中。当满足以下所有标准时,L(1-X)mv被视为有效。否则,当不满足以下至少一个标准时,该L(1-X)mv候选被视为无效。标准是:

[0156] 1. 合并候选具有L(1-X)mv。例如,当X为0且合并候选是双向预测或L1单向预测时,该L(1-X)mv候选被视为有效。

[0157] 2. 合并候选的L(1-X)mv的参考图片必须与AMVP模式得到的mv的LX mv的参考图片方向相反。

[0158] 然后将双边匹配细化或模板匹配应用于以所选合并候选MV和AMVP MV为起点的编码块。具体地,当从两个参考图片到当前图片的距离相同时(该距离由显示顺序测量,例如VVC或HEVC中定义的图片顺序计数),将双边匹配MV细化应用于合并候选MV和AMVP MVP作为起点。否则,如果启用了模板匹配功能,则将模板匹配MV细化应用于具有更高模板匹配成本的合并预测值或AMVP预测值。

[0159] AM模式由标志指示。在当前VVC和HEVC中,由于当所有参考图片都来自过去(就显示顺序而言)时不允许AM模式,因此在此条件下不需要发信号通知该指示标志。此外,如果为CU启用了AM模式,则AMVP方向LX还由标志指示。但是当为该图片启用“ph_mv_l1_zero_flag”时(例如,ph_mv_l1_zero_flag=1),AMVP方向LX的指示不需要用信号通知,并且假定为L0(换句话说,合并方向为L1),因为当启用“ph_mv_l1_zero_flag”时,不会为双向AMVP编码的CU的L1 mv发信号通知mvd信号。

[0160] 此外,基于AMVP方向LX的参考图片和合并方向L(1-X)的参考图片是否在相反方向而有条件地在比特流中发信号通知参考图片索引,也就是说,一个参考图片在当前图片之前(例如,过去的编码图片),而另一个参考图片在当前图片之后(例如,未来的编码图片)。

[0161] 如上所述,ECM中现有的双边匹配AMVP-合并模式具有限制,即该模式可以启用到满足以下两个条件的编码块:所选合并预测值和AMVP预测值满足DMVR条件,其中相对于当前图片至少有一个来自过去的参考图片和一个来自未来的参考图片;并且从两个参考图片到当前图片的距离相同。

[0162] 在本公开中,提出了几种帧间预测方案来改变或消除双边匹配AMVP-合并模式的约束,以实现更高的编解码效率(例如,更好的BD-速率性能)或更低的复杂度。所提出的方案可以独立应用或组合应用。

[0163] 在一个实施例中,无论所有参考图片是否来自过去,都启用双边匹配AMVP-合并模式。

[0164] 为了构建AM模式的 $L(1-X)$ 的mv合并候选列表,编码器或解码器搜索不同的合并候选(包括空间、时间、HMVP、对平均和零候选),并将不同合并候选的有效 $L(1-X)$ mv插入AM模式的mv合并候选列表或有效合并候选列表中。当满足以下标准时, $L(1-X)$ mv被视为有效:

[0165] 合并候选具有 $L(1-X)$ mv。例如,当 X 为0且合并候选为双向预测或 $L1$ 单向预测时,该 $L(1-X)$ mv候选被视为有效。

[0166] 也就是说,当合并候选具有 $L(1-X)$ mv时,无论合并候选的 $L(1-X)$ MV的参考图片是否与 LX MV的参考图片处于相反方向, $L(1-X)$ 都被视为有效。

[0167] 然后,在所有 $L(1-X)$ 合并候选中,隐式地选择AM模式的 $L(1-X)$ 合并mv作为与每个候选相关联的 $L(1-X)$ 合并预测块与 LX AMVP预测块之间具有最小匹配成本的mv。匹配成本可以是绝对差之和(SAD)、平方差之和(SSD)或任何其他用于测量两个块之间相似性的方法。

[0168] 然后将双边匹配细化或模板匹配应用于以所选合并候选MV和AMVP MV为起点的编码块。具体地,当从两个参考图片到当前图片的距离相同时(该距离由显示顺序测量,例如VVC或HEVC中定义的图片顺序计数),将双边匹配MV细化应用于合并MV候选和AMVP MVP作为起点。否则,如果启用了模板匹配功能,则将模板匹配MV细化应用于具有更高模板匹配成本的合并预测值或AMVP预测值。

[0169] 此外,当只有一个参考图片可用于AMVP方向 LX 时(X 可以是0或1;并且 L 代表参考图片列表),可以省略发信号通知参考图片索引。否则,当多于一个参考图片可用于AMVP方向 LX 时,参考图片索引始终在用于AMVP方向 LX 的比特流中用信号传送。

[0170] 在另一实施例中,基于本公开中所示的AM模式,当启用模板匹配功能时,可以有条件地将模板匹配MV细化应用于具有较高模板匹配成本的合并预测值或AMVP预测值。在一个示例中,当当前图片的所有参考图片都来自过去(例如,所有参考图片的显示顺序都在当前图片之前)时,模板匹配MV细化仅应用于具有较高模板匹配成本的合并预测值或AMVP预测值。

[0171] 在另一个示例中,当当前图片的参考图片并非全部来自过去(例如,至少一些参考图片具有在当前图片之后的显示顺序)时,模板匹配MV细化仅应用于具有更高模板匹配成本的合并预测值或AMVP预测值。

[0172] 在另一个示例中,无论模板匹配功能是否启用,模板匹配MV细化都不会应用于合并预测值或AMVP预测值。

[0173] 在另一个实施例中,基于本公开中所示的AM模式,AM模式的 $L(1-X)$ 合并mv被隐式

地选择为具有 LX mv和 $L(1-X)$ mv之间最小差异的mv。值得注意的是,当 LX mv和 $L(1-X)$ mv之间的参考图片不同时,可能需要mv缩放。在一个示例中, LX mv和 $L(1-X)$ mv具有不同的参考图片。然后将 $L(1-X)$ mv缩放为指向 LX mv的相同图片的新mv。缩放比率是基于当前图片和两个参考图片之间的时间距离来计算的。 LX mv和缩放后的 $L(1-X)$ mv之间的差被计算为水平分量的绝对差加上垂直分量的绝对差。

[0174] 在另一实施例中,基于本公开中所示的AM模式,AM模式的 $L(1-X)$ 合并mv在比特流中显式地用信号通知,以指示为AM模式选择了哪个合并候选。可以使用本公开中所示的标准或ECM中的现有方法来生成AM模式的合并候选列表。此外,在生成合并列表之后,可以选择性地应用候选重新排序。可以基于AMVP预测值和合并预测值的匹配成本来执行重新排序。

[0175] 在另一实施例中,可以自适应地使用AM模式的不同方案。在一个示例中,当当前图片的所有参考图片均来自过去(例如,所有参考图片的显示顺序均在当前图片之前)时,使用所提出的AM模式(在先前的实施例中示出)。否则,使用ECM中的原始AM模式。

[0176] 在另一个示例中,当当前图片的所有参考图片并非均来自过去(例如,至少一些参考图片的显示顺序在当前图片之后)时,使用所提出的AM模式(在先前的实施例中示出)。否则,使用ECM中的原始AM模式。

[0177] 在另一个示例中,当当前图片的所有参考图片均来自过去时,仅将一个合并候选用于AM模式;并且基于预定义标准(例如,双边匹配误差)在多个合并候选中选择该合并候选。由于仅直接选择一个合并候选,因此无需将合并索引信号传送到比特流中。当当前图片的参考图片并非全部来自过去时,可以为AM模式选择多个合并候选;并且在这种情况下,将合并索引信号传送到比特流中以指示哪个合并候选用于AM模式。

[0178] 在另一个实施例中,当为该图片(或条带)启用“ph_mvd_l1_zero_flag”(例如,ph_mvd_l1_zero_flag=1)并且还为该图片(或条带)启用AM模式时,不允许常规双向帧间模式,并且无需将其关联的语法元素用信号传送到比特流中。这是因为列表1mvd等于零的常规双向帧间模式与AM模式的功能高度重叠。

[0179] 在VVC规范中,语法元素“inter_pred_idc”用于指示当前帧间块是以列表0单向预测、列表1单向预测还是双向预测进行编码的。并且“inter_pred_idc”的三个不同值指示三个不同的预测方向(例如0、1和2分别代表L0预测、L1预测和双向预测)。inter_pred_idc的二值化如下表3所示。cbWidth和cbHeight是当前块或CU的宽度和高度。

[0180] 表3inter_pred_idc的二值化

inter_pred_idc 的值	inter_pred_idc 的名称	二进制串	
		(cbWidth+cbHeight)>12	(cbWidth+cbHeight)= =12
0	PRED_L0	00	0
1	PRED_L1	01	1
2	PRED_BI	1	-

[0182] 在所提出的方案中,当为该图片(或条带)启用“ph_mvd_l1_zero_flag”和AM模式(例如ph_mvd_l1_zero_flag=1)时,用于指示双向预测的inter_pred_idc的值被省略(即,

不发信号通知),如下表4所示。相反,AM模式可用作常规双向预测模式的替代方案。

[0183] 表4inter_pred_idc的建议二值化

inter_pred_idc 的值	inter_pred_idc 的名称	二进制串	
		(cbWidth+cbHeight)>12	(cbWidth+cbHeight) == 12 或(为该图片(或条 带)启用 “ph_mvd_l1_zero_flag” 和 AM 模式)
0	PRED_L0	00	0
1	PRED_L1	01	1
2	PRED_BI	1	-

[0185] 图6示出了与用户接口650耦合的计算环境610。计算环境610可以是数据处理服务器的一部分。计算环境610包括处理器620、存储器630和输入/输出(I/O)接口640。

[0186] 处理器620通常控制计算环境610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620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620可以包括促进处理器620与其他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、图形处理单元(GPU)等。

[0187] 存储器630被配置为存储各种类型的数据以支持计算环境610的操作。存储器630可以包括预定软件632。这种数据的示例包括用于在计算环境610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630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、闪存存储器、磁盘或光盘。

[0188] I/O接口640提供处理器620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640可以与编码器和解码器耦合。

[0189] 在实施例中,还提供了一种包括例如存储器630中的多个程序的非暂态计算机可读存储介质,所述多个程序可以由计算环境610中的处理器620执行以用于执行上述方法。备选地,非暂态计算机可读存储介质中可以存储有由编码器(例如,图2A中的视频编码器20)使用例如上述编码方法产生以由解码器(例如,图3A中的视频解码器30)在解码视频数据时使用的包括编码的视频信息(例如,包括一个或多个语法元素的视频信息)的比特流或数据流。非暂态计算机可读存储介质可以是例如ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘、光学数据存储设备等。

[0190] 在实施例中,还提供了一种计算设备,该计算设备包括:一个或多个处理器(例如,处理器620);以及在其中存储了可以由一个或多个处理器执行的多个程序的非暂态计算机可读存储介质或存储器1630,其中一个或多个处理器在执行多个程序时被配置为执行上述

方法。

[0191] 在实施例中,还提供了一种包括例如存储器630中的多个程序的计算机程序产品,所述多个程序可以由计算环境610中的处理器620执行以用于执行上述方法。例如,计算机程序产品可以包括非暂态计算机可读存储介质。

[0192] 在实施例中,计算环境1610可以由用于执行上述方法的一个或更多个ASIC、DSP、数字信号处理设备(DSPD)、可编程逻辑设备(PLD)、FPGA、GPU、控制器、微控制器、微处理器或其他电子组件来实现。

[0193] 本公开的描述已经出于说明的目的被呈现,并且不旨在穷举或限于本公开。受益于在以上描述和相关联的附图中呈现的教导,许多修改、变化和备选实施方式对于本领域普通技术人员将是显而易见的。

[0194] 除非另有具体说明,否则根据本公开的方法的步骤顺序仅旨在是说明性的,并且根据本公开的方法的步骤不局限于上述具体描述的顺序,而是可以根据实际情况而改变。此外,根据本公开的方法的步骤中的至少一个步骤可以根据实际需要进行调整、合并或删减。

[0195] 选择和描述示例是为了解释本公开的原理,并且使本领域的其他技术人员能够理解本公开的各种实施方式,并且最好地利用基本原理和具有适合于预期的特定用途的各种修改的各种实施方式。因此,将理解,本公开的范围不限于所公开的实施方式的具体示例,并且修改和其他实施方式旨在被包括在本公开的范围之内。

[0196] 图7是示出根据本公开的示例的视频解码方法的流程图。

[0197] 在步骤701中,解码器侧的处理器620可以确定对当前图片或当前图片的条带启用高级运动矢量预测(AMVP)-合并(AM)模式。

[0198] 在步骤702中,对于第一参考图片列表LX,处理器620可以获得当前图片中当前块的AMVP运动矢量(MV)作为LX MV,其中X为0或1。

[0199] 在步骤703中,处理器620可以获得第二参考图片列表L(1-X)的有效合并MV候选列表,其中有效合并MV候选列表包括具有L(1-X)MV的合并候选。

[0200] 在步骤704中,处理器620可在有效合并MV候选列表中选择AM模式的L(1-X)MV。

[0201] 在一些示例中,无论合并候选的L(1-X)MV的参考图片是否与LX MV的参考图片处于相反方向,具有L(1-X)MV的合并候选都被视为有效。

[0202] 图8是示出与图7所示的视频解码方法相对应的视频编码方法的流程图。

[0203] 在步骤801,编码器侧的处理器620可确定对当前图片或当前图片的条带启用高级运动矢量预测(AMVP)-合并(AM)模式;

[0204] 在步骤802,处理器620可针对第一参考图片列表LX(其中X为0或1)获得当前图片中当前块的AMVP运动向量(MV)作为LX MV。

[0205] 在步骤803,处理器620可针对第二参考图片列表L(1-X)获得有效合并MV候选列表,其中有效合并MV候选列表包括具有L(1-X)MV的合并候选。

[0206] 在步骤804,处理器620可在有效合并MV候选列表中选择AM模式的L(1-X)MV。

[0207] 在一些示例中,无论合并候选的L(1-X)MV的参考图片是否与LX MV的参考图片处于相反方向,具有L(1-X)MV的合并候选都被视为有效。

[0208] 在一些示例中,提供了一种用于视频解码的装置。该装置包括处理器620和存储器

630,存储器630被配置为存储可由处理器执行的指令;其中,处理器在执行指令后被配置为执行如图7所示的方法。

[0209] 在一些示例中,提供了一种用于视频编码的装置。该装置包括处理器620和存储器630,存储器630被配置为存储可由处理器执行的指令;其中,处理器在执行指令后被配置为执行如图8所示的方法。

[0210] 在一些其他示例中,提供了一种非暂态计算机可读存储介质,其中存储有指令。例如,指令可以存储为预定软件632,或软件的一部分。当指令由处理器620执行时,指令使得处理器执行如图7至图8所示的任何方法。在一个示例中,多个程序可以由计算环境610中的处理器620执行以接收(例如,从图1G中的视频编码器20)包括编码视频信息(例如,表示编码视频帧的视频块,和/或相关联的一个或多个语法元素等)的比特流或数据流,并且还可以由计算环境610中的处理器620执行以根据接收的比特流或数据流执行上述解码方法。在另一个示例中,多个程序可由计算环境610中的处理器620执行以执行上述编解码方法,以将视频信息(例如,表示视频帧的视频块,和/或相关联的一个或多个语法元素等)编码为比特流或数据流,并且也可由计算环境610中的处理器620执行以传输比特流或数据流(例如,传输至图2B中的视频解码器30)。或者,非暂态计算机可读存储介质中可存储比特流或数据流,该比特流或数据流包括由编码器(例如,图1G中的视频编码器20)使用例如上述编码方法生成的编码视频信息(例如,表示编码视频帧的视频块,和/或相关联的一个或多个语法元素等),以供解码器(例如,图2B中的视频解码器30)在解码视频数据时使用。非暂态计算机可读存储介质可以是例如ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘、光学数据存储设备等。

[0211] 上述方法可以使用包括一个或多个电路的装置来实现,这些电路包括专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑设备(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件。该装置可以将电路与其他硬件或软件组件结合使用以执行上述方法。上述公开的每个模块、子模块、单元或子单元可以至少部分地使用一个或多个电路来实现。

[0212] 本领域技术人员将从本文公开的说明书和实践中清楚地了解本公开内容的其他示例。本申请旨在涵盖遵循其一般原则对本公开内容的任何变化、使用或调整,并包括本领域已知或惯常做法范围内的与本公开内容的偏离。说明书和示例仅应被视为示例性的。

[0213] 应当理解,本公开内容不限于上述描述和附图中所示的确切示例,并且可以在不脱离其范围的情况下进行各种修改和改变。

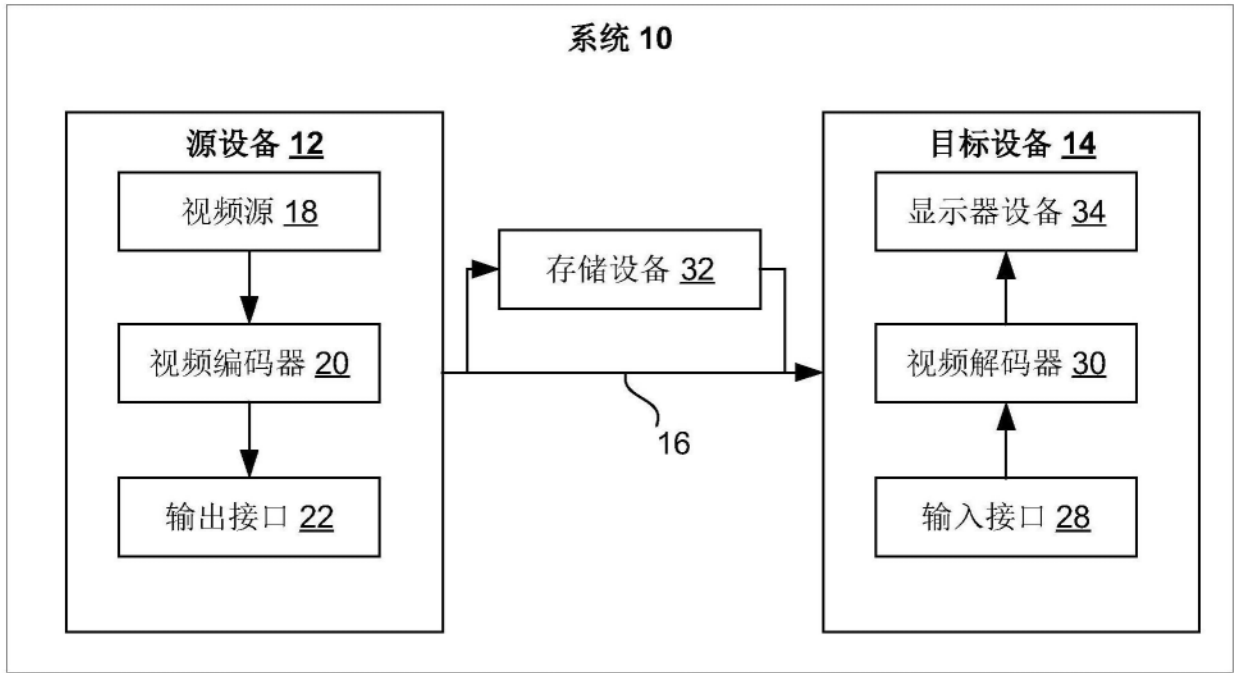


图1A

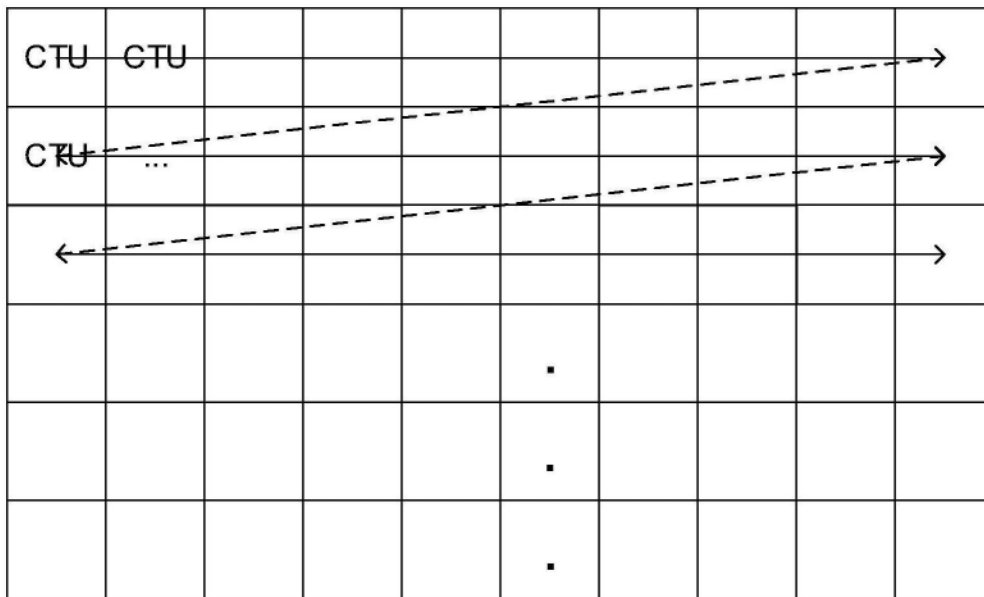


图1C

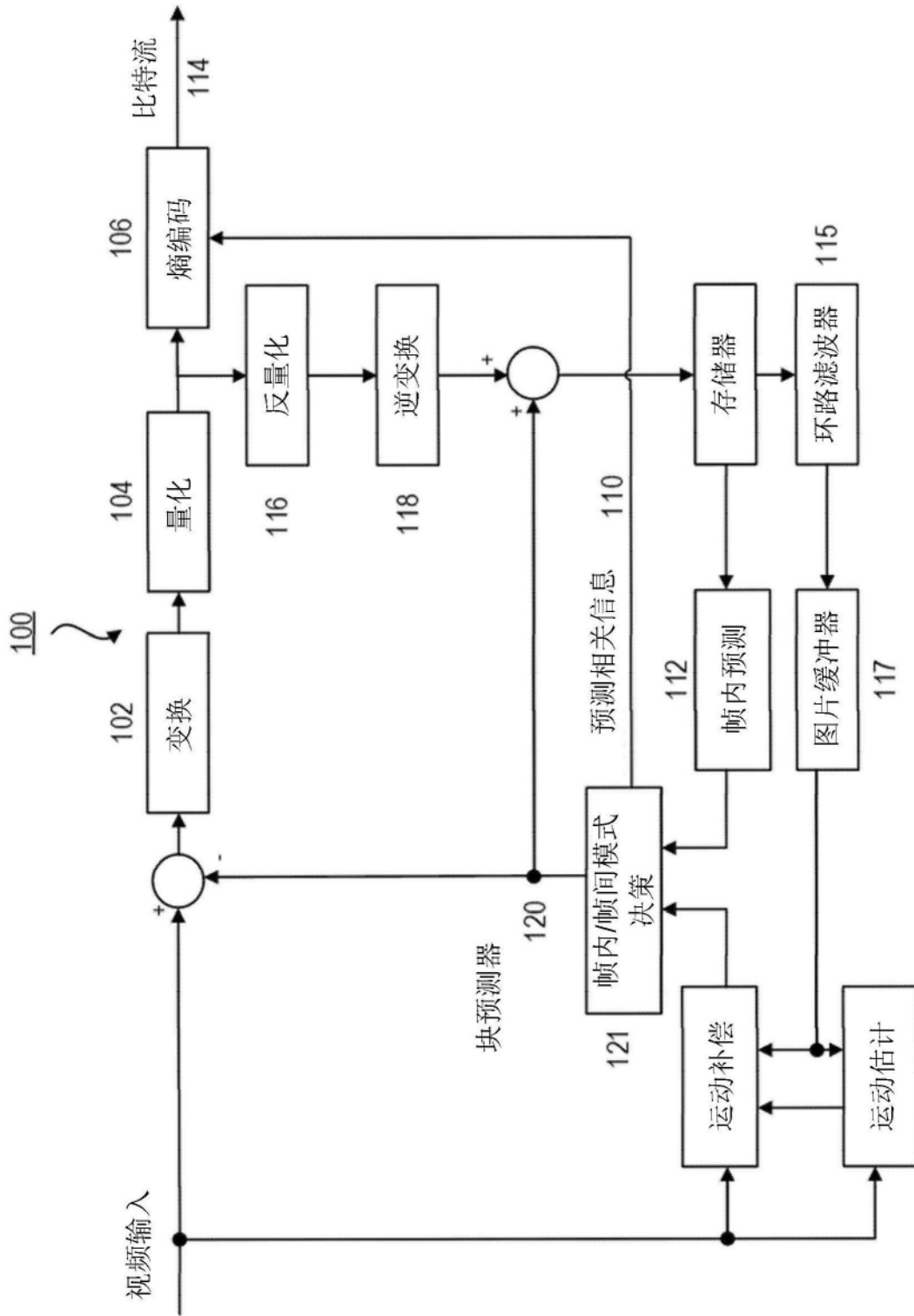


图1B

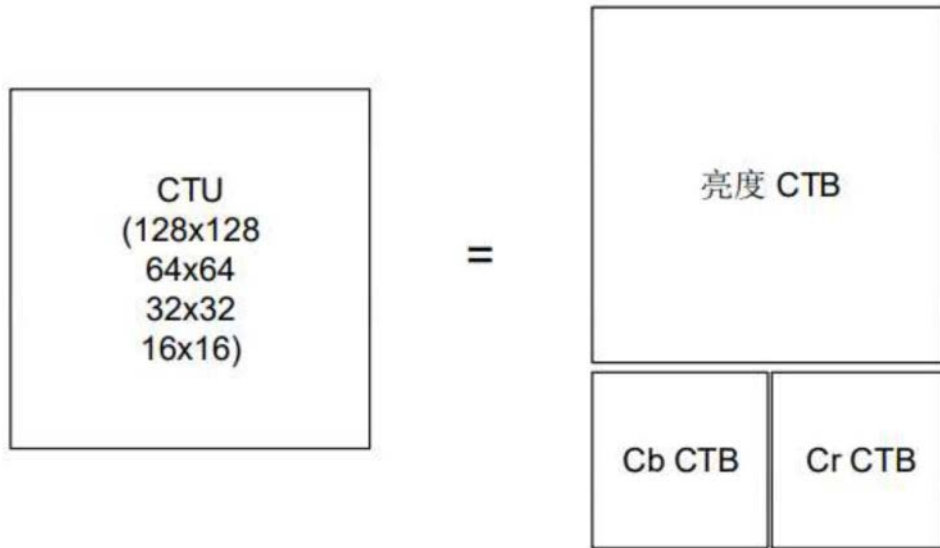


图1D

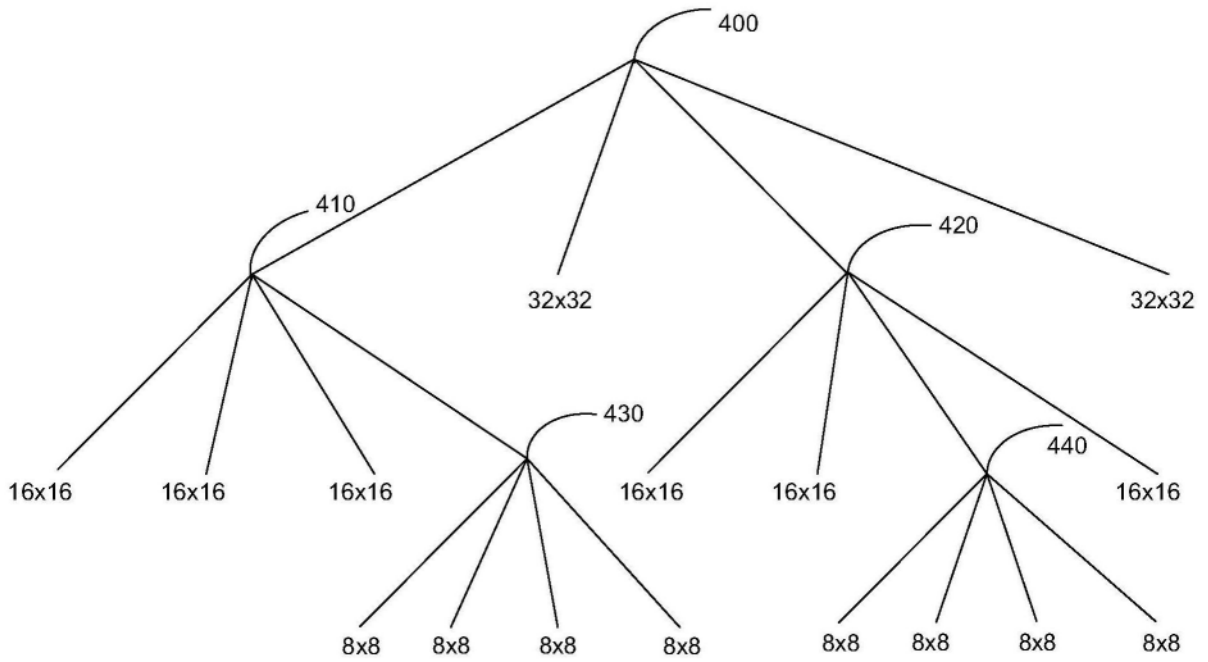


图1F

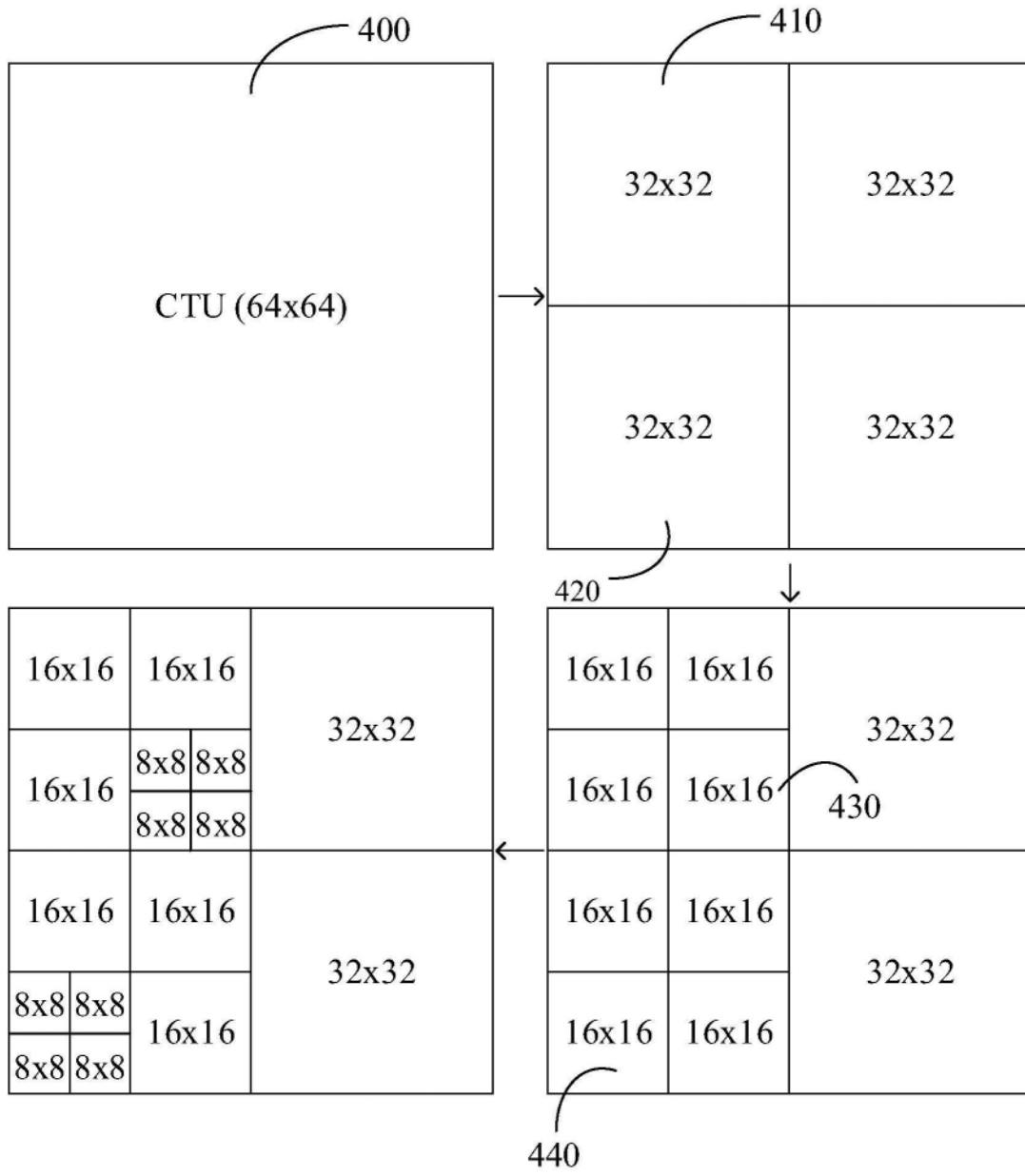


图1E

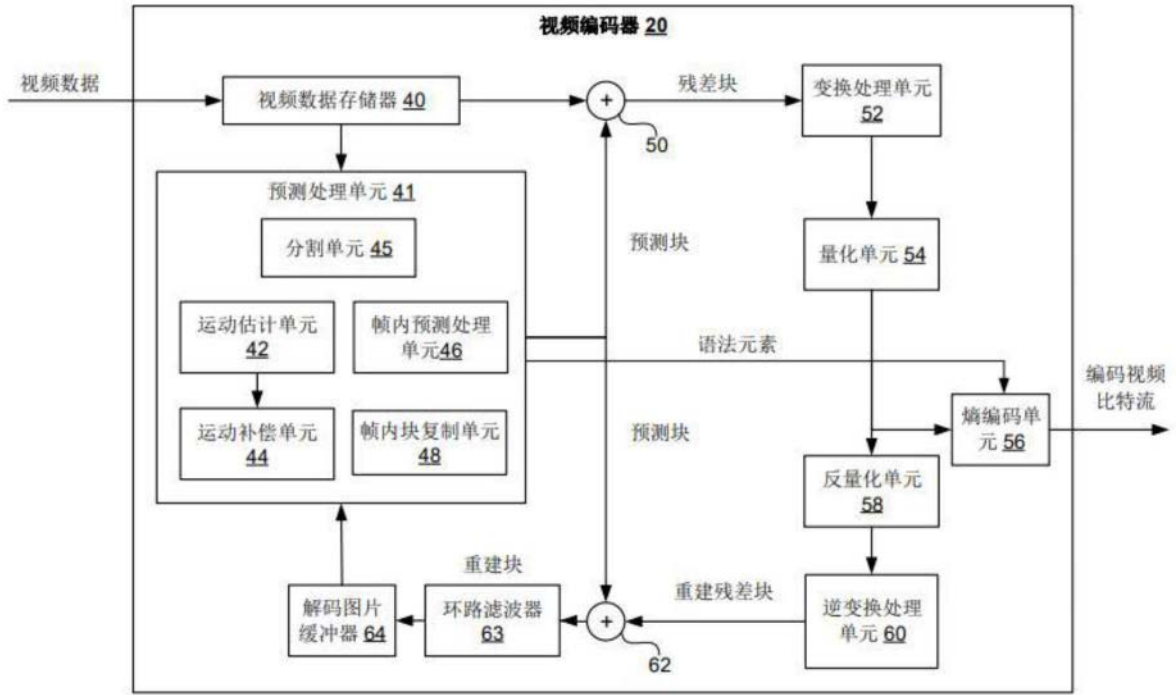


图1G

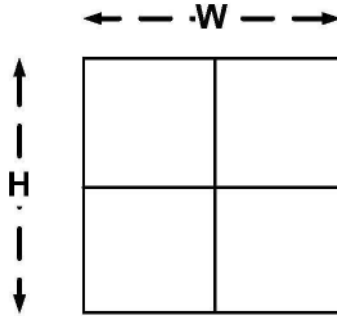


图3A

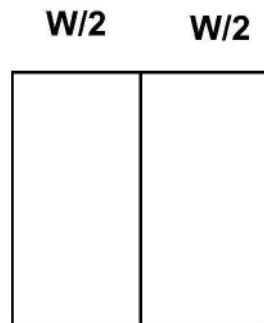


图3B



图3C

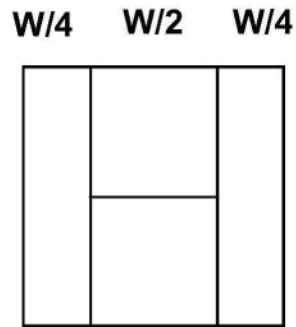


图3D

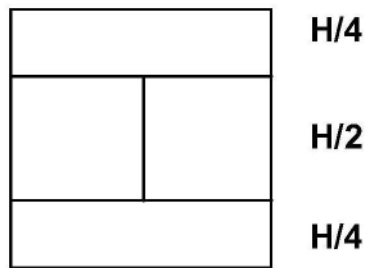


图3E

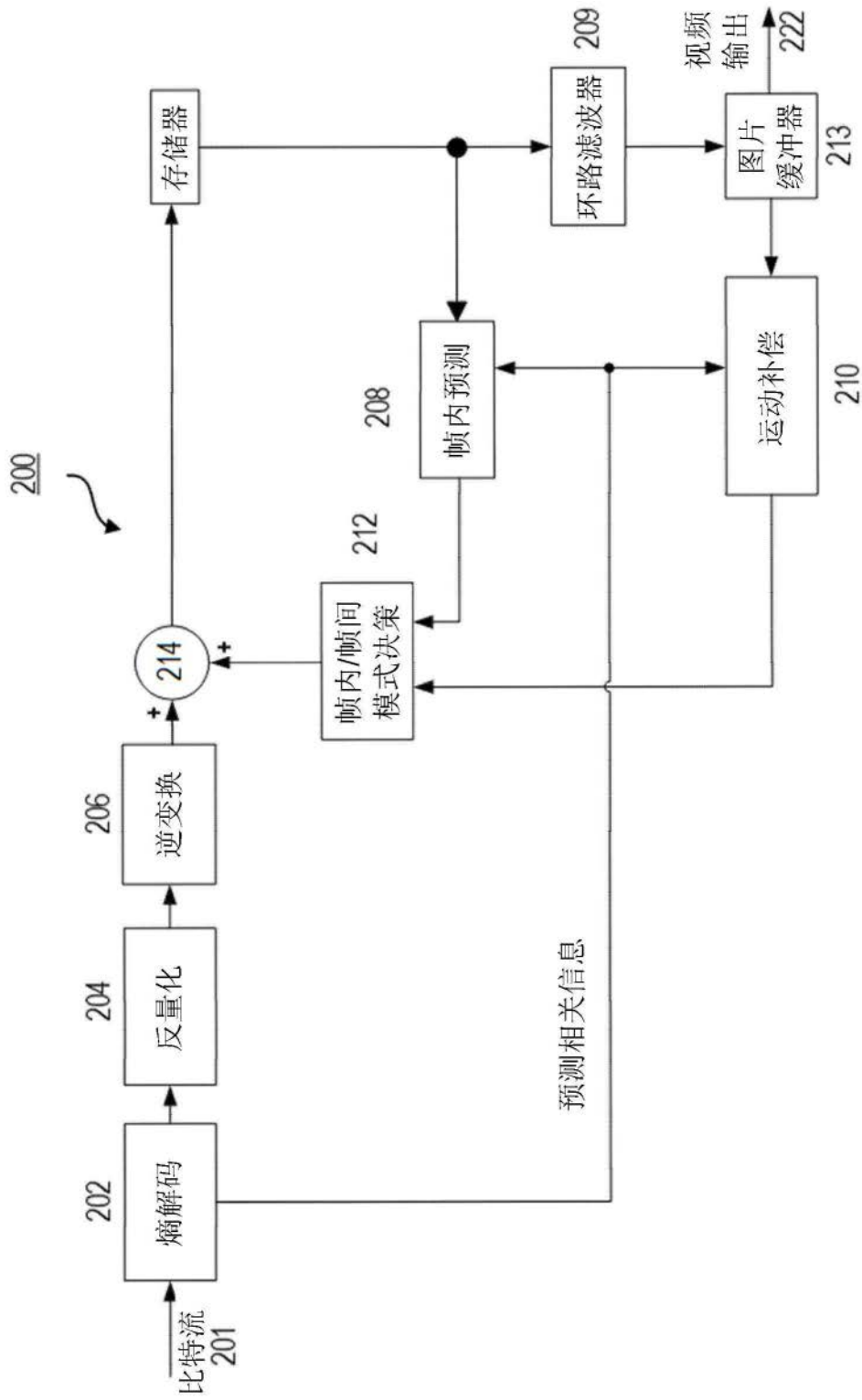


图2A

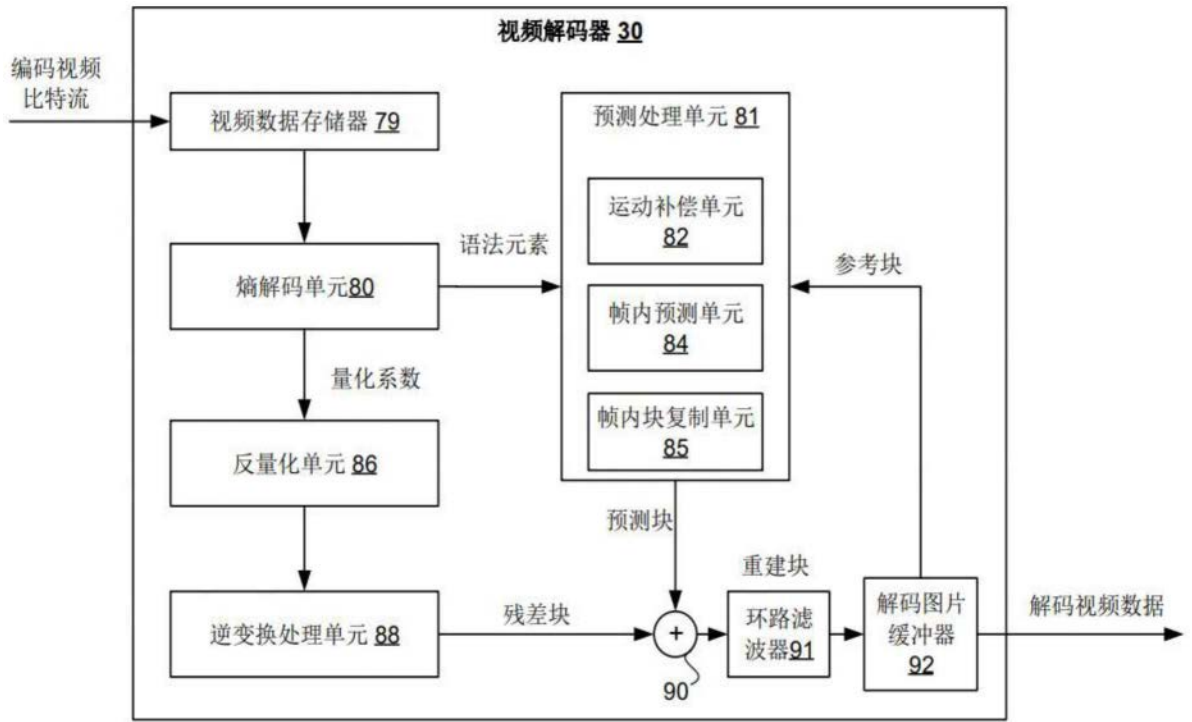


图2B

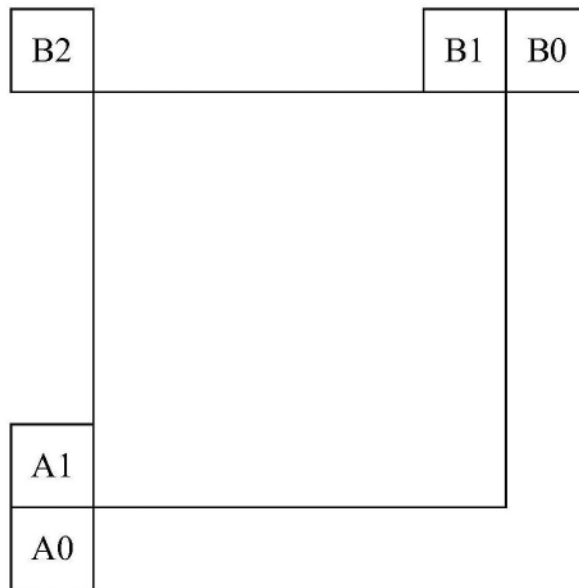


图4A

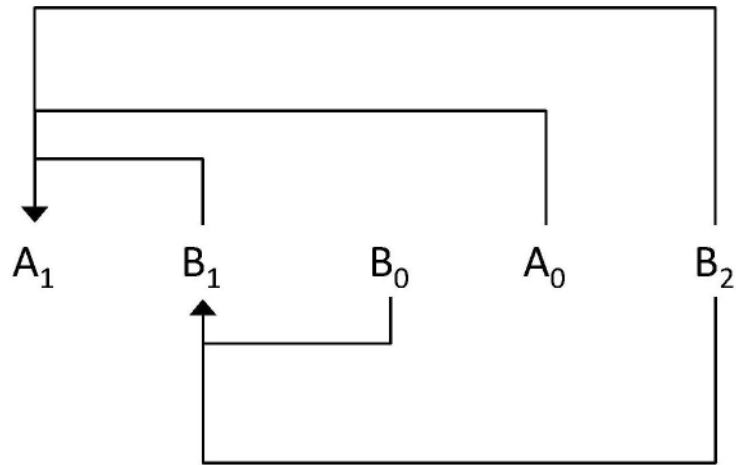


图4B

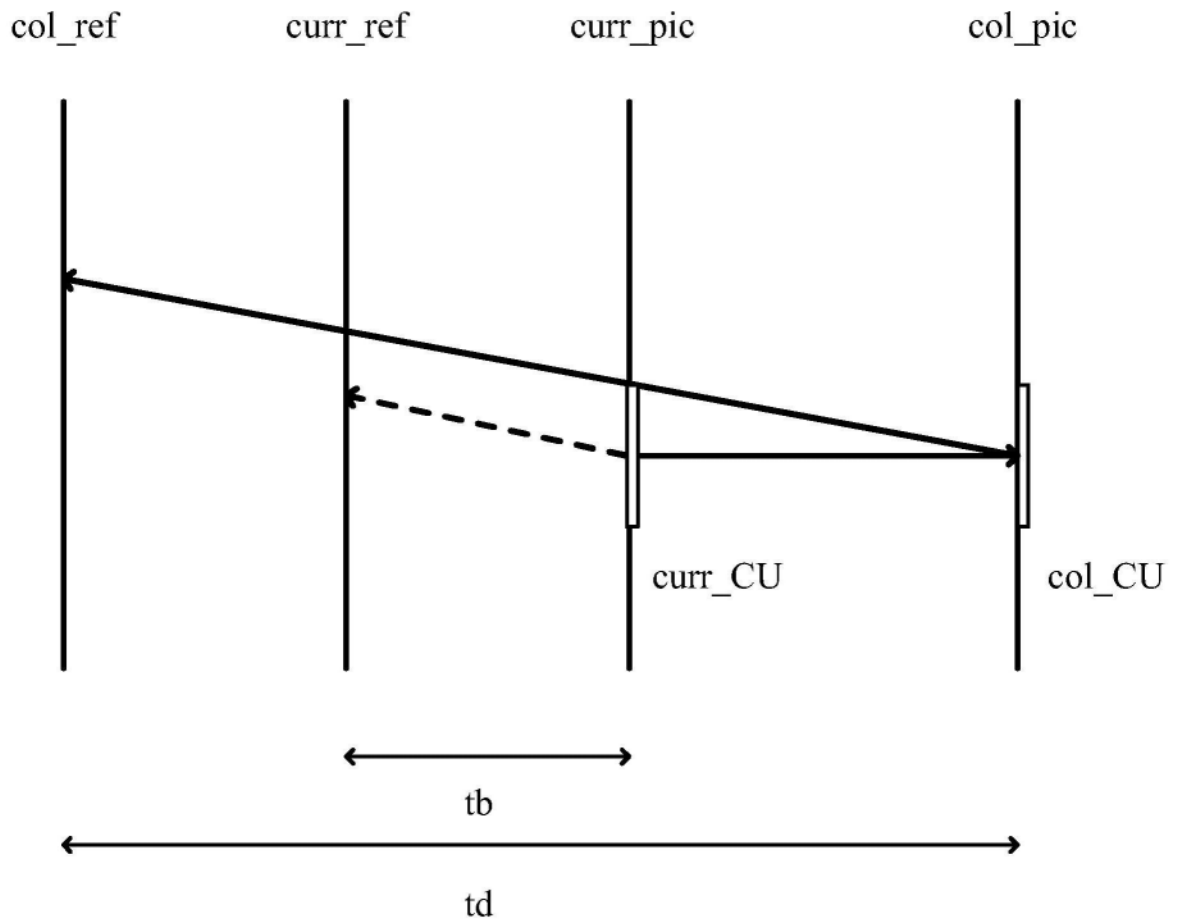


图4C

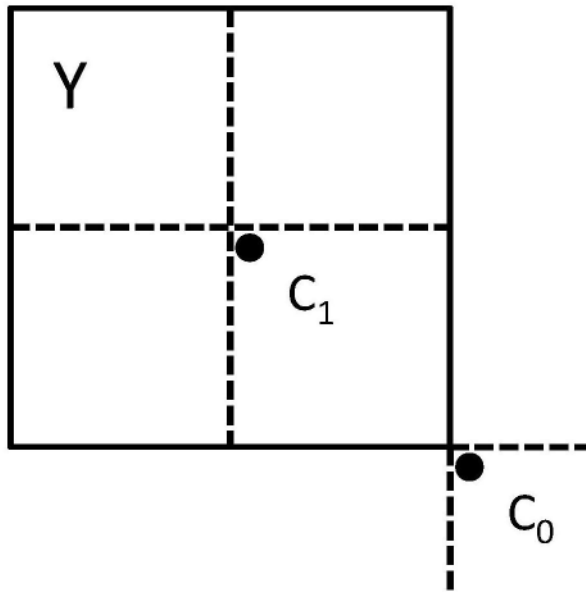
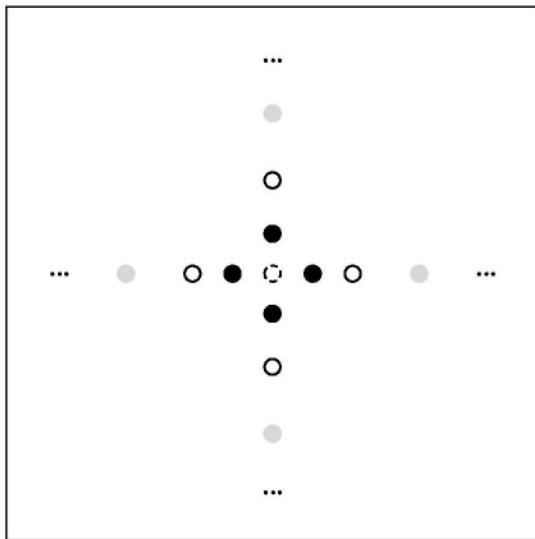


图4D

L0 参考



L1 参考

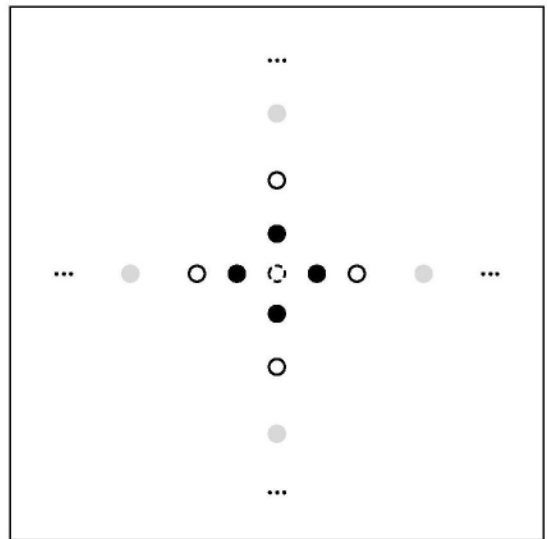


图5

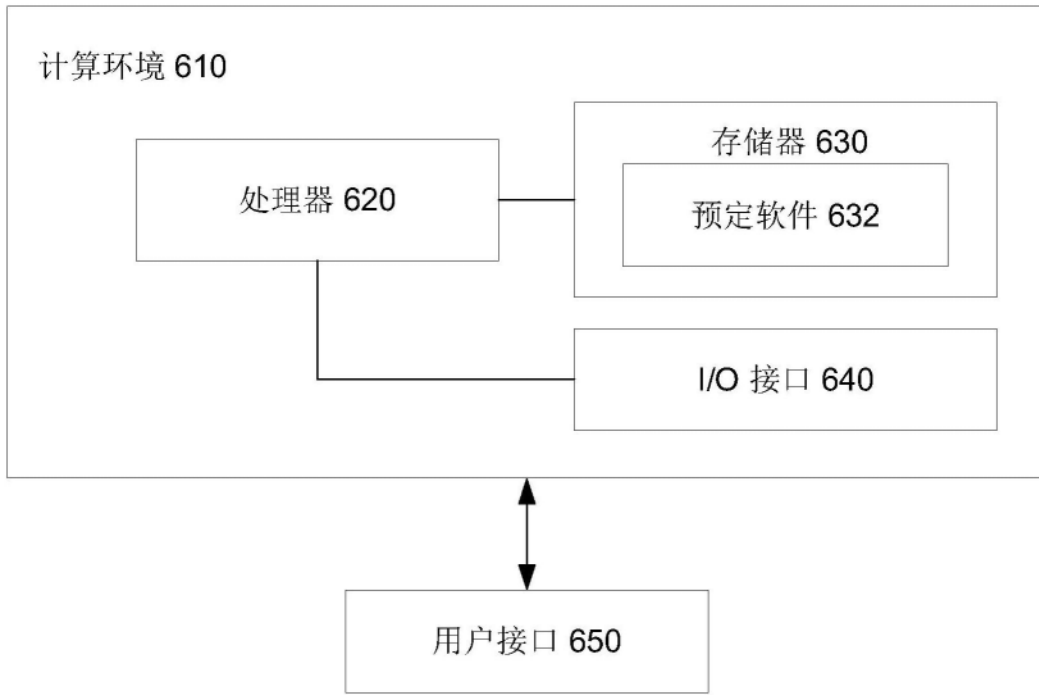


图6

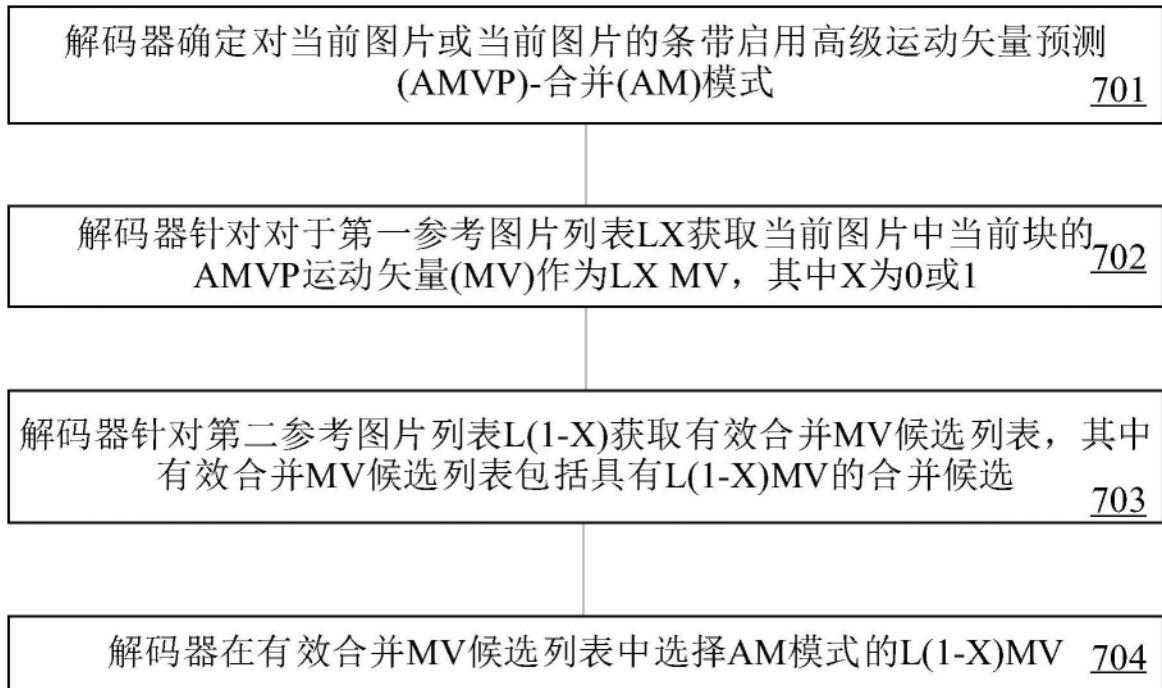


图7

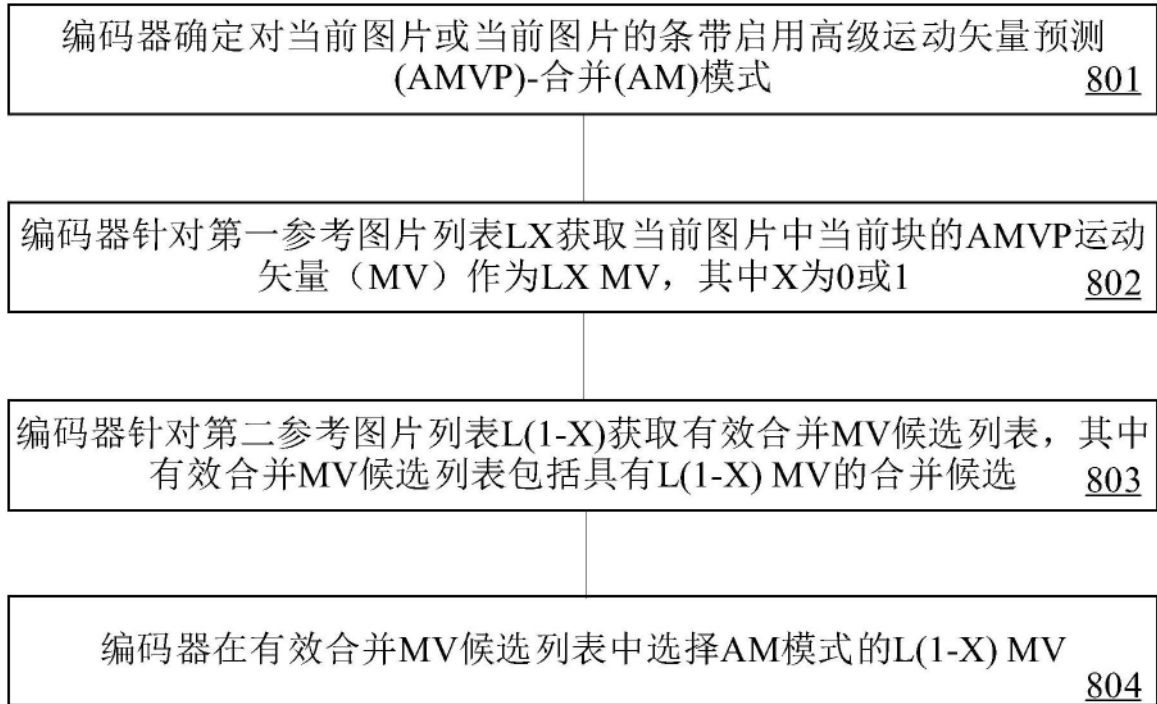


图8