



(12)发明专利

(10)授权公告号 CN 106797476 B

(45)授权公告日 2019.09.17

(21)申请号 201580053496.1

(72)发明人 庞超 王业奎

(22)申请日 2015.10.07

克里希纳坎斯·拉帕卡
瓦迪姆·谢廖金
马尔塔·卡切维奇

(65)同一申请的已公布的文献号

申请公布号 CN 106797476 A

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

(43)申请公布日 2017.05.31

代理人 宋献涛

(30)优先权数据

62/061,121 2014.10.07 US

(51)Int.Cl.

62/087,705 2014.12.04 US

H04N 19/52(2014.01)

62/111,568 2015.02.03 US

H04N 19/70(2014.01)

14/876,699 2015.10.06 US

(85)PCT国际申请进入国家阶段日

2017.03.31

(56)对比文件

(86)PCT国际申请的申请数据

PCT/US2015/054537 2015.10.07

CN 103339936 A,2013.10.02,

(87)PCT国际申请的公布数据

CN 103299621 A,2013.09.11,

W02016/057701 EN 2016.04.14

LI Bin.Non-SCCE1: Unification of
intra BC and inter modes.《Joint
Collaborative Team on Video Coding(JCT-
VC)》.2014,摘要、第2节、第7.4.3.2节、第
7.4.7.1节.

(73)专利权人 高通股份有限公司

审查员 楚丹丹

地址 美国加利福尼亚州

权利要求书4页 说明书57页 附图12页

(54)发明名称

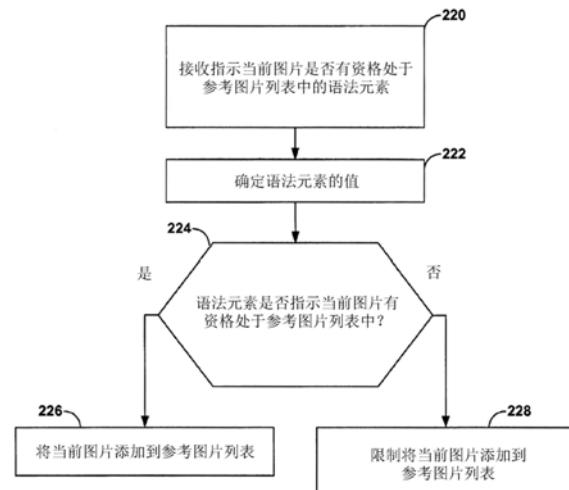
用于编码/解码视频数据的方法和视频解码

装置

(57)摘要

一般来说,本发明描述与用于块内复制模式及帧间模式的块向量译码相关的技术。在一个实例中,本发明涉及一种视频译码装置,其包括经配置以存储视频数据的存储器及一或多个处理器。所述视频译码装置经配置以确定用于译码当前视频块的参考图片并确定所述参考图片的图片次序计数POC值。响应于所述参考图片的所述POC值等于包含所述当前视频块的当前图片的POC值,所述视频译码装置设定语法元素的值以指示参考图片列表包含所述当前图片。否则,所述视频译码装置设定所述语法元素的值以指示所述参考图片列表不包含所述当前图片。

CN 106797476 B



1. 一种用于解码视频数据的方法,所述方法包括:

接收指示包含所述视频数据的第一视频块的第一图片是否有资格处于参考图片列表中的第一语法元素;

确定所述第一语法元素的值;

响应于所述第一语法元素指示所述第一图片有资格处于所述参考图片列表中:

将所述第一图片添加到所述参考图片列表;

确定所述第一视频块的运动向量具有整数像素精度;及

在将所述第一图片的多个运动向量预测符候选者中的每一运动向量预测符候选者用于候选者导出过程之前或在所述候选者导出过程完成之后,将所述每一运动向量预测符候选者转换为整数像素精度;

使用来自所述参考图片列表的图片的块和第一残余来解码所述第一视频块;

接收指示包含所述视频数据的第二视频块的第二图片是否有资格处于参考图片列表中的第二语法元素;

确定所述第二语法元素的值;

响应于所述第二语法元素指示所述第二图片没资格处于所述参考图片列表中,限制将所述第二图片放置于所述参考图片列表中;及

使用来自所述参考图片列表的图片的块和第二残余来解码所述第二视频块。

2. 根据权利要求1所述的方法,其中所述第一图片是包括P切片或B切片中的一者的帧内随机存取图片。

3. 根据权利要求2所述的方法,其中所述帧内随机存取图片进一步包括指定非空参考图片集的一或多个不同语法元素的集合。

4. 根据权利要求1所述的方法,其中用于解码所述第一图片的参考图片包含所述第一图片的经解码未滤波样本。

5. 根据权利要求1所述的方法,其中所述第一语法元素位于图片参数集或序列参数集中的一者中。

6. 根据权利要求1所述的方法,其进一步包括:

至少部分地基于与所述第一图片不同的图片确定时间运动向量预测符。

7. 根据权利要求1所述的方法,其中所述第一图片是清洁随机存取图片或断链存取图片中的一者,且其中所述方法进一步包括:

接收指示多个图片有资格用作所述第一图片的预测参考的第三语法元素,其中所述第三语法元素的值为1。

8. 根据权利要求1所述的方法,其进一步包括:

当所述参考图片列表仅包含所述第一图片时停用时间运动向量预测符。

9. 根据权利要求1所述的方法,其中所述第一视频块包括明度分量及色度分量,且其中所说明度分量或所述色度分量中的仅一者具有整数像素精度。

10. 根据权利要求1所述的方法,其中所述第一图片有资格被添加到所述参考图片列表,其中所述第一图片的每一预测样本包含在所述第一图片的图片边界内。

11. 根据权利要求1所述的方法,其进一步包括:

确定所述第一图片的相邻图片,其中所述相邻块是空间上相邻图片或时间上相邻图

片；

确定所述相邻图片中的预测性块,其中所述预测性块包含运动向量；

使用所述运动向量确定所述第一视频块的参考图片中的参考块,其中所述参考块包含所述运动预测信息；及

使用所述参考块的所述运动预测信息及所述第一残余解码所述第一视频块。

12. 一种编码视频数据的方法,所述方法包括:

确定用于编码所述视频数据的第一视频块的第一参考图片；

确定所述第一参考图片的图片次序计数POC值；

响应于所述第一参考图片的所述POC值等于包含所述第一视频块的第一图片的POC值：

设定语法元素的值以指示参考图片列表包含所述第一图片；

确定所述第一视频块的运动向量具有整数像素精度；及

在将所述第一图片的多个运动向量预测符候选者中的每一运动向量预测符候选者用于候选者导出过程之前或在所述候选者导出过程完成之后,将所述每一运动向量预测符候选者转换为整数像素精度；

使用来自所述参考图片列表的图片的块和第一残余来编码所述第一视频块；确定用于编码所述视频数据的第二视频块的第二参考图片；

确定用于所述第二参考图片的POC值；

响应于所述第二参考图片的所述POC值不同于第二图片的所述POC值,设定所述语法元素的所述值以指示所述参考图片列表不包含所述第二图片；及

使用来自所述参考图片列表的图片的块和第二残余来编码所述第二视频块。

13. 根据权利要求12所述的方法,其中所述第一图片是包括P切片或B切片中的一者的帧内随机存取图片。

14. 根据权利要求13所述的方法,其中所述帧内随机存取图片进一步包括指定非空参考图片集的一或多个不同语法元素的集合。

15. 根据权利要求12所述的方法,其中所述参考图片包含所述第一图片的经解码未滤波样本。

16. 根据权利要求12所述的方法,其中所述语法元素位于图片参数集或序列参数集中的一者中。

17. 根据权利要求12所述的方法,其进一步包括:

至少部分地基于与所述第一图片不同的图片确定时间运动向量预测符。

18. 根据权利要求12所述的方法,其中所述第一图片是清洁随机存取图片或断链存取图片中的一者,且其中所述方法进一步包括:

将指示多个图片有资格被用作所述第一图片的预测参考的第三语法元素的值设定为1。

19. 根据权利要求12所述的方法,其进一步包括:

当所述参考图片列表仅包含所述第一图片时停用时间运动向量预测符。

20. 根据权利要求12所述的方法,其中所述第一视频块包括明度分量及色度分量,且其中所说明度分量或所述色度分量中的仅一者具有整数像素精度。

21. 根据权利要求12所述的方法,其中所述参考图片的所述POC值等于所述第一图片的

所述POC值,其中所述第一图片的每一预测样本包含在所述第一图片的图片边界内。

22. 根据权利要求12所述的方法,其进一步包括:

确定所述第一图片的相邻图片,其中所述相邻块是空间上相邻图片或时间上相邻图片;

确定所述相邻图片中的预测性块,其中所述预测性块包含运动向量;

使用所述运动向量确定所述第一视频块的所述参考图片中的参考块,其中所述参考块包含所述运动预测信息;及

使用所述参考块的所述运动预测信息及所述第一残余编码所述第一视频块。

23. 一种视频解码装置,其包括:

存储器,其经配置以存储视频数据;及

一或多个处理器,其经配置以:

接收指示包含所述视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素;

确定所述语法元素的值;

响应于所述语法元素指示所述当前图片有资格处于所述参考图片列表中:

将所述当前图片添加到所述参考图片列表;

确定所述当前视频块的运动向量具有整数像素精度;及

在将所述当前图片的多个运动向量预测符候选者中的每一运动向量预测符候选者用于候选者导出过程之前或在所述候选者导出过程完成之后,将所述每一运动向量预测符候选者转换为整数像素精度;

响应于所述语法元素指示第二图片没资格处于所述参考图片列表中,限制将所述当前图片放置于所述参考图片列表中;及

使用来自所述参考图片列表的图片的块和残余来解码所述当前视频块。

24. 根据权利要求23所述的视频解码装置,其中所述一或多个处理器进一步经配置以:

当所述参考图片列表仅包含所述当前图片时停用时间运动向量预测符。

25. 根据权利要求23所述的视频解码装置,其中所述当前视频块包括明度分量及色度分量,且其中所述明度分量或所述色度分量中的仅一者具有整数像素精度。

26. 根据权利要求23所述的视频解码装置,其中所述一或多个处理器进一步经配置以:

确定所述当前图片的相邻图片,其中所述相邻块是空间上相邻图片或时间上相邻图片;

确定所述相邻图片中的预测性块,其中所述预测性块包含运动向量;

使用所述运动向量确定所述当前块的参考图片中的参考块,其中所述参考块包含运动预测信息;及

使用所述参考块的所述运动预测信息及残余解码所述当前视频块。

27. 一种存储指令的非暂时性计算机可读存储媒体,所述指令在由一或多个处理器执行时使得所述一或多个处理器进行以下操作:

接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素;

确定所述语法元素的值;

响应于所述语法元素指示所述当前图片有资格处于所述参考图片列表中；
将所述当前图片添加到所述当前图片的所述参考图片列表；
确定所述当前视频块的运动向量具有整数像素精度；及
在将所述当前图片的多个运动向量预测符候选者中的每一运动向量预测符候选者用于候选者导出过程之前或在所述候选者导出过程完成之后，将所述每一运动向量预测符候选者转换为整数像素精度；
响应于所述语法元素指示第二图片没资格处于所述参考图片列表中，限制将所述当前图片放置于所述参考图片列表中；及
使用来自所述参考图片列表的图片的块和残余来解码所述当前视频块。

用于编码/解码视频数据的方法和视频解码装置

[0001] 本发明涉及以下美国临时专利申请案,所述临时专利申请案中的每一者的全部内容以引用的方式并入本文中:

[0002] 2015年2月3日申请的美国临时专利申请案62/111,568;

[0003] 2014年12月4日申请的美国临时专利申请案62/087,705;及

[0004] 2014年10月7日申请的美国临时专利申请案62/061,121。

技术领域

[0005] 本发明涉及视频译码,并且更确切地说,涉及视频块的基于其它视频块的预测。

背景技术

[0006] 数字视频能力可并入到多种多样的装置中,包含数字电视、数字直播系统、无线广播系统、个人数字助理 (PDA)、膝上型或桌上型计算机、平板计算机、电子图书阅读器、数码相机、数字记录装置、数字媒体播放器、视频游戏装置、视频游戏控制台、蜂窝式或卫星无线电电话(所谓的“智能电话”)、视频电话会议装置、视频流式传输装置及其类似者。数字视频装置实施视频压缩技术,例如由MPEG-2、MPEG-4、ITU-T H.263或ITU-T H.264/MPEG-4第10部分高级视频译码 (AVC) 所定义的标准、目前正在开发的高效率视频译码 (HEVC) 标准及这些标准的扩展中所描述的视频压缩技术。视频装置可以通过实施这些视频压缩技术更有效地发射、接收、编码、解码及/或存储数字视频信息。

[0007] 视频压缩技术执行空间(图片内)预测及/或时间(图片间)预测来减少或去除视频序列中固有的冗余。对于基于块的视频译码来说,可以将视频切片(即,视频帧或视频帧的一部分)分割成视频块,视频块还可被称作树块、译码单元 (CU) 和/或译码节点。图片的经帧内译码 (I) 切片中的视频块是使用相对于同一图片中的相邻块中的参考样本的空间预测来编码。图片的经帧间译码 (P或B) 切片中的视频块可使用相对于同一图片中的相邻块中的参考样本的空间预测,或相对于其它参考图片中的参考样本的时间预测。图片可被称作帧,且参考图片可被称为参考帧。

[0008] 空间或时间预测产生待译码块的预测性块。残余数据表示待译码原始块与预测性块之间的像素差。经帧间译码块是根据指向形成预测性块的参考样本块的运动向量和指示经译码块与预测性块之间的差的残余数据来编码。经帧内译码块是根据帧内译码模式及残余数据来编码。为了进一步压缩,可将残余数据从像素域变换到变换域,从而产生残余变换系数,随后可以对所述残余变换系数进行量化。可扫描最初布置为二维阵列的经量化变换系数,以便产生变换系数的一维向量,且可应用熵译码以实现更多压缩。

发明内容

[0009] 一般来说,本发明描述与用于块内复制模式及帧间模式的块向量译码相关的技术。在各种实例中,本发明的技术可与屏幕内容译码结合使用。

[0010] 在一个实例中,本发明涉及一种编码视频块之方法,所述方法包括:确定用于译码

当前视频块的参考图片；确定参考图片的图片次序计数(POC)值；响应于参考图片的POC值等于包含当前视频块的当前图片的POC值，设定语法元素的值以指示参考图片列表包含当前图片；及响应于参考图片的POC值不同于当前图片的POC值，设定语法元素的值以指示参考图片列表不包含当前图片。

[0011] 在另一个实例中，本发明涉及一种视频编码装置，其包括经配置以存储视频数据的存储器；及经配置以进行以下操作的一或多个处理器：确定用于译码当前视频块的参考图片；确定参考图片的图片次序计数(POC)值；响应于参考图片的POC值等于包含当前视频块的当前图片的POC值，设定语法元素的值以指示参考图片列表包含当前图片；及响应于参考图片的POC值不同于当前图片的POC值，设定语法元素的值以指示参考图片列表不包含当前图片。

[0012] 在另一个实例中，本发明涉及一种用于视频编码的设备，所述设备包括：用于确定用于译码视频块的参考图片的装置；用于确定参考图片的图片次序计数(POC)值的装置；用于响应于参考图片的POC值等于包含当前视频块的当前图片的POC值，设定语法元素的值以指示参考图片列表包含当前图片的装置；及用于响应于参考图片的POC值不同于当前图片的POC值，设定语法元素的值以指示参考图片列表不包含当前图片的装置。

[0013] 在另一个实例中，本发明涉及一种存储指令的计算机可读存储媒体，所述指令在由一或多个处理器执行时使得一或多个处理器进行以下操作：确定用于译码当前视频块的参考图片；确定参考图片的图片次序计数(POC)值；响应于参考图片的POC值等于包含当前视频块的当前图片的POC值，设定语法元素的值以指示参考图片列表包含当前图片；及响应于参考图片的POC值不同于当前图片的POC值，设定语法元素的值以指示参考图片列表不包含当前图片。

[0014] 在另一个实例中，本发明涉及一种用于解码视频数据的方法，所述方法包括：接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素；确定语法元素的值；响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中，将当前图片添加到参考图片列表；及响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中，限制将当前图片放置于参考图片列表中。

[0015] 在另一个实例中，本发明涉及一种视频解码装置，其包括：经配置以存储视频数据的存储器；及经配置以进行以下操作的一或多个处理器：接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素；确定语法元素的值；响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中，将当前图片添加到参考图片列表中；及响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中，限制将当前图片放置于参考图片列表中。

[0016] 在另一个实例中，本发明涉及一种用于视频编码的设备，所述设备包括：用于接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素的装置；用于确定语法元素的值的装置；响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中，用于将当前图片添加到参考图片列表中的装置；及响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中，用于限制将当前图片放置于参考图片列表中的装置。

[0017] 在另一个实例中，本发明涉及一种存储指令的计算机可读存储媒体，所述指令在

由一或多个处理器执行时使得所述一或多个处理器进行以下操作：接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素；确定语法元素的值；响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中，将当前图片添加到参考图片列表中；及响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中，限制将当前图片放置于参考图片列表中。

[0018] 在附图和以下描述中阐述本发明的一或多个实例的细节。本发明的其它特征、目标和优点将从所述描述和图式以及权利要求书显而易见。

附图说明

- [0019] 图1是说明可利用本发明中所描述的技术的实例视频编码和解码系统的框图。
- [0020] 图2A到2C是说明用于视频数据的不同样本格式的概念图。
- [0021] 图3是说明根据4:2:0样本格式格式化的 16×16 译码单元的概念图。
- [0022] 图4是说明根据4:2:2样本格式格式化的 16×16 译码单元的概念图。
- [0023] 图5展示帧内运动补偿(IMC)模式的概念性说明。
- [0024] 图6展示用于合并模式和AMVP模式的空间相邻运动向量候选者的实例。
- [0025] 图7是由相邻块P及Q形成的四个像素长的垂直块边界的说明。
- [0026] 图8是说明实例空间块向量候选者的概念图。
- [0027] 图9是说明可实施本发明中所描述的技术的实例视频编码器的框图。
- [0028] 图10是说明可实施本发明中所描述的技术的实例视频解码器的框图。
- [0029] 图11是说明根据本文中所描述的一或多种技术的实例视频编码技术的流程图。
- [0030] 图12是说明根据本文中所描述的一或多种技术的实例视频解码技术的流程图。

具体实施方式

[0031] 包含最近开发的高效视频译码(HEVC)标准在内的各种视频译码标准包含用于视频块的预测性译码模式，其中基于已经译码的视频数据块来预测当前正译码的块。在帧内预测模式中，基于与当前块在相同图片中的一或多个先前经译码相邻块来预测当前块，而在帧间预测模式中，基于不同图片中的已经译码块来预测当前块。在帧间预测模式中，确定先前经译码帧的块用作预测性块的过程有时被称作运动估计，其通常由视频编码器执行，且识别及检索预测性块的过程有时被称作运动补偿，其由视频编码器及视频解码器两者执行。

[0032] 视频编码器通常通过使用多种译码情形来译码视频且识别产生所要速率失真折衷的译码情形来确定如何译码视频数据序列。当为特定视频块测试帧内预测译码情形时，视频编码器通常测试相邻像素行(即，正译码块的紧接上方的像素行)，且测试相邻像素列(即，正译码块的紧接左侧的像素列)。相比之下，当测试帧间预测情形时，视频编码器通常在大得多的搜索区域中识别候选预测性块，其中所述搜索区域对应于先前经译码的视频数据帧中的视频块。

[0033] 然而，已发现，对于某些类型的视频图像，诸如包含文字、符号或重复图案的视频图像，可通过使用帧内运动补偿(IMC)模式相对于帧内预测及帧间预测实现译码增益，所述帧内运动补偿模式有时也被称作块内复制(IBC)模式。在本发明中，术语IMC模式及IBC模式

是可互换的。举例来说,术语IMC模式最初由开发人员使用,但后来经修改为IBC模式。在IMC模式中,当在帧内预测模式中时,视频编码器搜索与经译码块处于相同帧或图片中的预测性块,但当在帧间预测模式中时,视频编码器对较宽搜索区域进行搜索而并不只是相邻行及列。

[0034] 在IMC模式中,对于识别与经预测性块处于相同帧或图片内的预测性块,视频编码器可确定偏移向量,有时也被称作运动向量或块向量。偏移向量包含(例如)x分量及y分量,其中x分量识别经预测视频块与预测性块之间的水平位移,且其中y分量识别经预测视频块与预测性块之间的垂直位移。视频编码器在经编码位流中传信所述所确定的偏移向量,使得视频解码器在解码经编码位流时可识别视频编码器所选定的预测性块。

[0035] 本发明提出有效地统一块内复制及帧间译码的技术。所提出的技术主要涉及(但不限于)屏幕内容译码,包含支持可能地高位深度(例如,超过8位)及不同色度取样格式(诸如4:4:4、4:2:2、4:2:0、4:0:0等)。

[0036] 图1是说明可利用本发明中所描述的技术的实例视频编码和解码系统的框图。如图1中所展示,系统10包含源装置12,其产生稍后由目的地装置14解码的经编码视频数据。源装置12及目的地装置14可包括多种多样的装置中的任一者,包含桌上型计算机、笔记型(即,膝上型)计算机、平板计算机、机顶盒、电话手持机(例如所谓的“智能”电话)、所谓的“智能”板、电视机、相机、显示装置、数字媒体播放器、视频游戏控制台、视频流式传输装置或类似者。在一些情况下,源装置12和目的地装置14可经装备以用于无线通信。

[0037] 目的地装置14可经由链路16接收待解码的经编码视频数据。链路16可包括能够将经编码视频数据从源装置12移动到目的地装置14的任何类型的媒体或装置。在一个实例中,链路16可包括使得源装置12能够实时地将经编码视频数据直接发射到目的地装置14的通信媒体。经编码视频数据可根据通信标准(例如,无线通信协议)来调制,并且发射到目的地装置14。通信媒体可包括任何无线或有线通信媒体,例如射频(RF)频谱或一或多个物理发射线。通信媒体可形成分组网络(例如,局域网、广域网或全球网络,例如因特网)的部分。通信媒体可包含路由器、交换机、基站或可用于促进从源装置12到目的地装置14的通信的任何其它设备。

[0038] 替代地,可将经编码数据从输出接口22输出到存储装置17。类似地,可以通过输入接口从存储装置17存取经编码数据。存储装置17可包含多种分布式或本地存取的数据存储媒体中的任一者,例如硬盘驱动器、蓝光光盘、DVD、CD-ROM、快闪存储器、易失性或非易失性存储器,或用于存储经编码视频数据的任何其它合适的数字存储媒体。在另一实例中,存储装置17可对应于文件服务器或可保持由源装置12产生的经编码视频的另一中间存储装置。目的地装置14可经由串流传输或下载从存储装置17存取所存储的视频数据。文件服务器可为能够存储经编码视频数据且将经编码视频数据发射到目的地装置14的任何类型的服务器。实例文件服务器包含万维网服务器(例如,用于网站)、FTP服务器、网络附接存储(NAS)装置或本地磁盘驱动器。目的地装置14可以通过任何标准数据连接(包含因特网连接)来存取经编码视频数据。此连接可包含适用于存取存储在文件服务器上的经编码视频数据的无线信道(例如,Wi-Fi连接)、有线连接(例如,DSL、电缆调制解调器等)或两者的组合。经编码视频数据从存储装置17的发射可为流式发射、下载发射或两者的组合。

[0039] 本发明的技术不一定限于无线应用或设定。所述技术可应用于支持多种多媒体应

用中的任一者的视频译码,例如空中电视广播、有线电视发射、卫星电视发射、流式视频发射(例如,经由因特网)、用于存储于数据存储媒体上的数字视频的编码、存储在数据存储媒体上的数字视频的解码,或其它应用。在一些实例中,系统10可经配置以支持单向或双向视频发射从而支持例如视频流式传输、视频重放、视频广播及/或视频电话的应用。

[0040] 在图1的实例中,源装置12包含视频源18、视频编码器20和输出接口22。在一些情况下,输出接口22可包含调制器/解调器(调制解调器)及/或发射器。在源装置12中,视频源18可包含例如视频俘获装置(例如,摄像机)、含有先前俘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及/或用于产生计算机图形数据作为源视频的计算机图形系统或此类来源的组合等来源。作为一个实例,如果视频源18为摄像机,那么源装置12和目的地装置14可形成所谓的相机电话或视频电话。然而,本发明中所描述的技术可大体适用于视频译码,且可应用于无线和/或有线应用。

[0041] 可由视频编码器20编码所俘获视频、预俘获视频或计算机产生的视频。可经由源装置12的输出接口22将经编码视频数据直接发射到目的地装置14。经编码视频数据还可(或替代地)存储到存储装置17上以供稍后由目的地装置14或其它装置存取以用于解码及/或重放。

[0042] 目的地装置14包含输入接口28、视频解码器30和显示装置32。在一些情况下,输入接口28可包含接收器及/或调制解调器。目的地装置14的输入接口28经由链路16接收经编码视频数据。经由链路16传达或在存储装置17上提供的经编码视频数据可包含由视频编码器20产生的多种语法元素以供由例如视频解码器30的视频解码器用于解码视频数据。此些语法元素可与在通信媒体上发射、存储在存储媒体上或存储在文件服务器中的经编码视频数据包含在一起。

[0043] 显示器装置32可与目的地装置14集成或在所述目的地装置外部。在一些实例中,目的地装置14可包含集成显示装置,且还经配置以与外部显示装置介接。在其它实例中,目的地装置14可为显示装置。一般来说,显示装置32将经解码视频数据显示给用户,且可包括多种显示装置中的任一者,例如液晶显示器(LCD)、等离子体显示器、有机发光二极管(OLED)显示器或另一类型的显示装置。

[0044] 视频编码器20和视频解码器30可根据例如高效视频译码(HEVC)的视频压缩标准来操作,并且可符合HEVC测试模型(HM)。被称作“HEVC工作草案10”或“HEVC WD10”的HEVC标准的工作草案描述于布洛斯(Bross)等人的“编者提出的对HEVC版本1的校正(Editors' proposed corrections to HEVC version 1)”中,ITU-T SG16 WP3和ISO/IEC JTC1/SC29/WG11的视频译码联合合作小组(JCT-VC),2013年4月,韩国仁川第13次会议。本发明中描述的技术也可根据当前正开发的HEVC标准的扩展来操作。替代地或另外,视频编码器20和视频解码器30可根据其它专有或业界标准来操作,所述标准为例如ITU-T H.264标准(替代地被称作MPEG-4第10部分,高级视频译码(AVC)),或此类标准的扩展。然而,本发明的技术不限于任何特定译码标准。视频压缩标准的其它实例包含ITU-T H.261、ISO/IEC MPEG-1视觉、ITU-T H.262或ISO/IEC MPEG-2视觉、ITU-T H.263、ISO/IEC MPEG-4视觉和ITU-T H.264(也被称为ISO/IEC MPEG-4 AVC),包含其可缩放视频译码(SVC)和多视图视频译码(MVC)扩展。

[0045] 最近,已由ITU-T视频译码专家组(VCEG)和ISO/IEC动画专家组(MPEG)的视频译码

联合协作小组 (JCT-VC) 完成新的视频译码标准 (即, 高效率视频译码 (HEVC)) 的设计。最新的HEVC草案说明书 (且下文中称为HEVC WD) 可从http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCTVC-01003-v2.zip获得。对HEVC的范围扩展 (即HEVC RExt) 也正由JCT-VC开发。范围扩展的最新工作草案 (WD) (在下文中被称作RExt WD7) 可从http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zip获得。

[0046] 在此文件中, JCTVC-Q1003中的HEVC规范文本常常被称作HEVC版本1。范围扩展规范可变成HEVC的版本2。然而, 在很大程度上, 就所提出的技术涉及 (例如) 运动向量预测来说, HEVC版本1及范围扩展规范为技术上类似的。因此, 每当我们提及基于HEVC版本1的变化时, 相同变化可适用于范围扩展规范, 且每当我们再使用HEVC版本1模块时, 我们实际上也再使用HEVC范围扩展模块 (具有相同子条款)。

[0047] 近年来, 开始研究用于带有运动的例如文本和图形的屏幕内容材料的新译码工具, 并且研究人员已经提出了改进屏幕内容的译码效率的技术。因为有迹象表明可通过用新颖的专用译码工具来开发屏幕内容的特性来获得译码效率的显著改进, 所以正发布征求计划书 (CfP), 其目标是可能开发高效视频译码 (HEVC) 标准的进一步扩展, 包含用于屏幕内容译码 (SCC) 的特定工具。邀请公司和组织来响应于此征求而提交建议。此CfP的使用情况和要求描述于MPEG文件N14174中。在第17次JCT-VC会议期间, 建立SCC测试模型 (SCM)。SCC的最新工作草案 (WD) 可从http://phenix.int-evry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R1005-v3.zip获得。

[0048] 尽管图1中未示出, 但在一些方面中, 视频编码器20和视频解码器30可各自与音频编码器和解码器整合, 且可包含适当的多路复用器-多路分用器 (MUX-DEMUX) 单元或其它硬件和软件, 以处置对共同数据流或单独数据流中的音频和视频两者的编码。在一些实例中, 如果适用的话, 多路复用器-多路分用器单元可符合ITU H.223多路复用器协议, 或例如用户数据报协议 (UDP) 等其它协议。

[0049] 视频编码器20和视频解码器30各自可实施为多种合适的编码器电路中的任一者, 例如一或多个微处理器、数字信号处理器 (DSP)、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA)、离散逻辑、软件、硬件、固件或其任何组合。当部分地用软件实施所述技术时, 装置可将用于所述软件的指令存储于合适的非暂时性计算机可读媒体中且使用一或多个处理器执行硬件中的指令以执行本发明的技术。视频编码器20和视频解码器30中的每一者可包含在一或多个编码器或解码器中, 所述编码器或解码器中的任一者可集成为相应装置中的组合编码器/解码器 (编解码器) 的部分。

[0050] JCT-VC开发了HEVC标准。所述HEVC标准化努力是基于被称作HEVC测试模型 (HM) 的视频译码装置的演进模型。HM根据 (例如) ITU-T H.264/AVC假设视频译码装置相对于现有装置的若干额外能力。举例来说, 虽然H.264提供九种帧内预测编码模式, 但HM可提供多达三十三种帧内预测编码模式。

[0051] 一般来说, HM的工作模型描述视频帧或图片可以划分成包含明度及色度样本两者的一连串树块或最大译码单元 (LCU)。树块具有与H.264标准的宏块类似的目的。切片包含按译码顺序的多个连续树块。视频帧或图片可以被分割成一或多个切片。每一树块可根据四叉树而分裂成多个译码单元 (CU)。举例来说, 作为四叉树的根节点的树块可分裂成四个

子节点,且每一子节点又可为父代节点且可分裂成另外四个子节点。最后的未经分裂的子节点(作为四叉树的叶节点)包括译码节点,即,经译码视频块。与经译码位流相关联的语法数据可定义树块可分裂的最大次数,且还可定义译码节点的最小大小。

[0052] 在HEVC中将CU定义为基本译码单元。在HEVC中,帧首先被划分成被称作CTU(译码树单元)的多个正方形单元。使CTU大小为 $2N \times 2N$ 。每一CTU可被划分成4个 $N \times N$ CU,且每一CU可进一步被划分成4个 $(N/2) \times (N/2)$ 单元。块分裂可以相同方式继续,直到其达到预定义的最大分裂等级或所允许的最小CU大小为止。CTU的大小、CTU进一步分裂成CU的等级以及CU的最小大小是在编码配置中定义的,且将发送到视频解码器30或可为视频编码器20和视频解码器30两者所已知。

[0053] CU包含译码节点以及与所述译码节点相关联的预测单元(PU)及变换单元(TU)。CU的大小对应于译码节点的大小并且形状必须是正方形。CU的大小可从 8×8 像素到具有最大 64×64 像素或更大的树块的大小变动。每一CU可含有一或多个PU及一或多个TU。举例来说,与CU相关联的语法数据可描述将CU分割成一或多个PU。分割模式可在CU被跳过还是经直接模式编码、经帧内预测模式编码或经帧间预测模式编码之间有所不同。PU可分割成非正方形形状。举例来说,与CU相关联的语法数据还可描述CU根据四叉树分割成一或多个TU。TU可为正方形或非正方形形状。

[0054] HEVC标准允许根据TU变换,TU可针对不同CU而有所不同。TU的大小通常是基于针对经分割LCU定义的给定CU内的PU的大小而设定,但是情况可能并不总是如此。TU通常与PU大小相同或小于PU。在一些实例中,对应于CU的残余样本可使用被称为“残余四叉树”(RQT)的四叉树结构而细分成较小单元。RQT的叶节点可被称作变换单元(TU)。可变换与TU相关联的像素差值以产生可经量化的变换系数。

[0055] 一般来说,PU包含与预测过程有关的数据。举例来说,当对PU进行帧内模式编码时,PU可包含描述用于PU的帧内预测模式的数据。作为另一实例,当对PU进行帧间模式编码时,PU可以包含定义PU的运动向量的数据。定义PU的运动向量的数据可描述(例如)运动向量的水平分量、运动向量的垂直分量、运动向量的分辨率(例如,四分之一像素精度或八分之一像素精度)、运动向量所指的参考图片,及/或运动向量的参考图片列表(例如,列表0、列表1或列表C)。

[0056] 一般来说,TU用于变换和量化过程。具有一或多个PU之给定CU还可包含一或多个变换单元(TU)。在预测之后,视频编码器20可计算对应于PU的残余值。残余值包括像素差值,所述像素差值可变换为变换系数、经量化并且使用TU进行扫描以产生串行化变换系数以用于熵译码。本发明通常使用术语“视频块”来指CU的译码节点。在一些特定情况下,本发明还可使用术语“视频块”来指包含译码节点以及PU及TU的树块,即,LCU或CU。

[0057] 视频序列通常包含一系列视频帧或图片。图片群组(GOP)通常包括一系列视频图片中的一或多者。GOP可包含GOP的标头、图片中的一或多者的标头或其它处的语法数据,所述语法数据描述GOP中所包含的多个图片。图片的每一切片可包含描述用于相应切片的编码模式的切片语法数据。视频编码器20通常对个别视频切片内的视频块进行操作以便对视频数据进行编码。视频块可对应于CU内的译码节点。视频块可具有固定或变化的大小,并且可根据指定译码标准具有不同大小。

[0058] 作为一实例,HM支持各种PU大小的预测。假定特定CU的大小为 $2N \times 2N$,那么HM支持

在 $2N \times 2N$ 或 $N \times N$ 的PU大小下的帧内预测,和在 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 或 $N \times N$ 的对称PU大小下的帧间预测。HM还支持用于在 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 及 $nR \times 2N$ 的PU大小下的帧间预测的不对称分割。在不对称分割中,不分割CU的一个方向,但是另一方向分割成25%及75%。CU的对应于25%分区的部分由“n”后面接着“上方”、“下方”、“左侧”或“右侧”的指示来指示。因此,举例来说,“ $2N \times nU$ ”是指水平地分割的 $2N \times 2N$ CU,其中顶部为 $2N \times 0.5N$ PU,而底部为 $2N \times 1.5N$ PU。

[0059] 在本发明中,“ $N \times N$ ”与“ $N \times N$ ”可互换地使用以指代就垂直维度和水平维度来说的视频块的像素尺寸,例如,16×16像素或16乘16像素。一般来说,16×16块将在垂直方向上具有16个像素($y=16$),且在水平方向上具有16个像素($x=16$)。同样地, $N \times N$ 块一般在垂直方向上具有N个像素,并且在水平方向上具有N个像素,其中N表示非负整数值。块中的像素可布置成行和列。此外,块未必需要在水平方向与垂直方向上具有相同数目的像素。举例来说,块可包括 $N \times M$ 个像素,其中M未必等于N。

[0060] 因此,根据HEVC,CU可包含一或多个预测单元(PU)及/或一或多个变换单元(TU)。本发明还使用术语“块”、“分区”或“部分”来指CU、PU或TU中的任一者。一般来说,“部分”可指视频帧的任何子集。此外,本发明通常使用术语“视频块”来指CU的译码节点。在一些特定情况下,本发明还可使用术语“视频块”来指包含译码节点以及PU及TU的树块,即,LCU或CU。因此,视频块可对应于CU内的译码节点,并且视频块可具有固定或变化的大小,并且可根据指定译码标准具有不同大小。

[0061] 还可被称作色度格式的视频取样格式可相对于CU中所包含的明度样本的数目定义CU中包含的色度样本的数目。取决于色度分量的视频取样格式,U和V分量的大小(就样本的数目来说)可与Y分量的大小相同或不同。在HEVC标准中,定义被称作chroma_format_idc的值以相对于明度分量指示色度分量的不同取样格式。在HEVC中,在SPS中传信chroma_format_idc。表1说明chroma_format_idc的值与相关联的色度格式之间的关系。

[0062]

chroma_format_idc	chroma format	SubWidthC	SubHeightC
0	单色	-	-
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

[0063] 表1:HEVC中定义的不同色度格式

[0064] 在表1中,变量SubWidthC及SubHeightC可用于指示用于明度分量的样本的数目与用于每一色度分量的样本的数目之间的水平及垂直取样速率比。在表1中所描述的色度格式中,两个色度分量具有相同的取样速率。因此,在4:2:0取样中,两个色度阵列中的每一者具有明度阵列的高度的一半和宽度的一半,而在4:2:2取样中,两个色度阵列中的每一者具有所说明度阵列的相同高度和宽度的一半。在4:4:4取样中,两个色度阵列中的每一者可具有与明度阵列相同的高度和宽度,或在一些情况下,三个色彩平面可全部分别处理为单色经取样图片。

[0065] 在表1的实例中,对于4:2:0格式,用于明度分量的取样速率为用于水平及垂直方向两者的色度分量的取样速率的两倍。因此,对于根据4:2:0格式格式化的译码单元,用于

明度分量的样本阵列的宽度及高度为用于色度分量的每一样本阵列的宽度及高度的两倍。类似地,对于根据4:2:2格式格式化的译码单元,用于明度分量的样本阵列的宽度为用于每一色度分量的样本阵列的宽度的两倍,但用于明度分量的样本阵列的高度等于用于每一色度分量的样本阵列的高度。对于根据4:4:4格式格式化的译码单元,用于明度分量的样本阵列具有与用于每一色度分量的样本阵列相同的宽度及高度。应注意,除YUV色彩空间之外,还可根据RGB空间颜色定义视频数据。以此方式,本文所描述的色度格式可适用于YUV或RGB色彩空间。通常对RGB色度格式进行取样以使得红色样本的数目、绿色样本的数目及蓝色样本的数目相等。因此,如本文所使用的术语“4:4:4色度格式”可指YUV色彩空间或RGB色彩空间,其中对于所有颜色分量,样本的数目相等。

[0066] 根据本发明的技术,视频编码器20可确定用于译码当前图片中的当前块的参考图片。举例来说,参考图片可包含具有用于译码当前视频块的运动预测信息的参考块。视频编码器20可进一步确定参考图片的图片次序计数(POC)值。响应于参考图片的POC值等于包含当前视频块的当前图片的POC值,视频编码器20可设定语法元素的值以指示参考图片列表包含当前图片。响应于参考图片的POC值不同于当前图片的POC值,视频编码器20可设定语法元素的值以指示参考图片列表不包含当前图片。

[0067] 在本发明的其它技术中,视频解码器30可接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素。视频解码器30可进一步确定语法元素的值。响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中,视频解码器30可将当前图片添加到参考图片列表。响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中,视频解码器30可限制将当前图片放置于参考图片列表中。

[0068] 图2A到2C是说明用于视频数据的不同样本格式的概念图。图2A是说明4:2:0样本格式的概念图。如图2A中所说明,对于4:2:0样本格式,色度分量的大小为明度分量的大小的四分之一。因此,对于根据4:2:0样本格式格式化的CU,对于色度分量的每个样本来说存在四个明度样本。图2B是说明4:2:2样本格式的概念图。如图2B中所说明,对于4:2:2样本格式,色度分量的大小为明度分量的大小的二分之一。因此,对于根据4:2:2样本格式格式化的CU,对于色度分量的每个样本来说存在两个明度样本。图2C是说明4:4:4样本格式的概念图。如图2C中所说明,对于4:4:4样本格式,色度分量的大小与明度分量的大小相同。因此,对于根据4:4:4样本格式格式化的CU,对于色度分量的每个样本存在一个明度样本。

[0069] 图3是说明根据4:2:0样本格式格式化的 16×16 译码单元的实例的概念图。图3说明色度样本相对于CU内的明度样本的相对位置。如上文所描述,CU通常是根据水平及垂直明度样本的数目来定义。因此,如图3中所说明,根据4:2:0样本格式格式化的 16×16 CU包含明度分量的 16×16 样本及每一色度分量的 8×8 样本。另外,如上文所描述,CU可分割成较小CU。举例来说,图3中所说明的CU可分割成四个 8×8 CU,其中每一 8×8 CU包含用于明度分量的 8×8 样本及用于每一色度分量的 4×4 样本。

[0070] 图4是说明根据4:2:2样本格式格式化的 16×16 译码单元的实例的概念图。图4说明色度样本相对于CU内的明度样本的相对位置。如上文所描述,CU通常是根据水平及垂直明度样本的数目来定义。因此,如图4中所说明,根据4:2:2样本格式格式化的 16×16 CU包含明度分量的 16×16 样本及每一色度分量的 8×16 样本。另外,如上文所描述,CU可分割成较小CU。举例来说,图4中所说明的CU可分割成四个 8×8 CU,其中每一CU包含用于明度分量的8

×8样本及用于每一色度分量的4×8样本。

[0071] 在使用CU的PU的帧内预测性或帧间预测性译码之后,视频编码器20可以计算CU的TU的残余数据。PU可包括空间域(也被称作像素域)中的像素数据,且TU在将变换应用到残余视频数据之后可包括变换域中的系数,所述变换(例如)为离散余弦变换(DCT)、整数变换、小波变换或概念上类似的变换。所述残余数据可对应于未经编码图片的像素与对应于PU的预测值之间的像素差。视频编码器20可形成包含用于CU的残余数据的TU,且接着变换TU以产生用于CU的变换系数。

[0072] 在用以产生变换系数的任何变换之后,视频编码器20可执行变换系数的量化。量化通常是指变换系数经量化以可能地减少用于表示系数的数据量从而提供进一步压缩的过程。量化过程可减小与系数中的一些或全部相关联的位深度。举例来说,可在量化期间将n位值舍入到m位值,其中n大于m。

[0073] 在一些实例中,视频编码器20可利用预定义扫描次序来扫描经量化的变换系数以产生可经熵编码的串行化向量。在其它实例中,视频编码器20可执行自适应扫描。在扫描经量化的变换系数以形成一维向量之后,视频编码器20可例如根据上下文自适应可变长度译码(CAVLC)、上下文自适应二进制算术译码(CABAC)、基于语法的上下文自适应二进制算术译码(SBAC)、概率区间分割熵(PIPE)译码或另一熵编码方法对一维向量进行熵编码。视频编码器20可熵编码与经编码视频数据相关联的语法元素以供视频解码器30在解码视频数据时使用。

[0074] 为了执行CABAC,视频编码器20可以向待发射的符号指派上下文模型内的上下文。所述上下文可(例如)与符号的相邻值是否为非零有关。为了执行CAVLC,视频编码器20可选择用于待发射的符号的可变长度码。VLC中的码字可经建构使得相对较短码对应于更有可能符号,而较长码对应于不太可能符号。以此方式,使用VLC可实现优于(例如)将相同长度的码字用于待发射的每一符号的位节省。概率确定可基于指派给符号的上下文。

[0075] 根据本发明的一个实例技术,视频解码器30可使用IMC模式对当前视频数据块进行解码。对于当前视频数据块,视频解码器30可确定用于传信偏移向量的分量的码字长度且基于所述码字长度对偏移向量进行译码。经译码的偏移向量的分量可为x分量或y分量,且用于传信一个分量的码字的长度可不同于用于传信x分量与y分量中的另一者的第二码字的长度。

[0076] 视频解码器30可(例如)通过基于用于针对当前视频数据块执行IMC的搜索区的大小确定码字的长度来确定用于传信偏移向量的分量的码字的长度。可(例如)基于当前块的像素与搜索区的顶边界之间的距离、当前块的像素与搜索区的左边界之间的距离、当前块的像素与搜索区的右边界之间的距离中的一或者确定搜索区的大小。

[0077] 另外或替代地,视频解码器30可基于包括当前块的译码树单元(CTU)的大小、译码树单元中的当前块的位置或视频数据帧中的当前块的位置中的一或者,基于当前块的大小确定用于传信偏移向量的分量的码字的长度。

[0078] 根据本发明的另一实例技术,视频解码器30可使用IMC模式对当前视频数据块进行解码。视频解码器30可针对当前视频数据块确定偏移向量(例如,视频编码器20针对其传信视频解码器30用以确定偏移向量的信息的当前块的明度分量的偏移向量),且响应于偏移向量指向子像素位置(例如,响应于偏移向量指向色度样本内的子像素位置),修改偏移

向量以产生用于针对当前块的色度分量定位参考块的经修改偏移向量。经修改偏移向量可(例如)指向整数像素位置或指向与子像素位置相比处于较低精度位置的像素位置。

[0079] 根据本发明的另一实例技术,视频解码器30可针对当前视频数据块确定最大CTU大小。视频解码器30可针对当前视频数据块确定用于IMC模式的最大CU大小。用于IMC模式的最大CU大小可小于最大CTU大小。视频解码器30可基于用于IMC模式的最大CU大小对当前视频数据块进行译码。基于用于IMC模式的最大CU大小译码当前视频数据块可(例如)包含响应于当前视频数据块的大小大于用于IMC模式的最大CU大小不在IMC模式中译码当前视频数据块或响应于当前视频数据块的大小小于或等于用于IMC模式的最大CU大小在IMC模式中译码当前视频数据块中的一或多或少者。可(例如)在经编码视频位流中传信或基于已经译码视频数据的统计数据确定用于IMC模式的最大CU大小。

[0080] 根据本发明的另一实例技术,视频解码器30可使用IMC模式对当前视频数据块进行译码。基于当前块的大小、当前块的位置及包括当前块的CTU的大小中的一或多或少者,视频解码器30可针对当前视频数据块确定用于译码偏移向量的译码方法且基于所确定的译码方法对所述偏移向量进行译码。用于译码偏移向量的译码方法可(例如)包含固定长度译码、可变长度译码、算术译码及基于上下文译码中的一者或其组合。当前块的位置可(例如)为CTU内的位置或视频数据帧内的位置。

[0081] 图5展示IMC模式的概念性说明。如上所述,IMC模式与IBC模式相同。视频编码器20和视频解码器30可(例如)经配置以使用IMC模式编码及解码视频数据块。例如远程桌面、远程游戏、无线显示器、汽车信息娱乐、云计算等许多应用正变成人的日常生活中的日常,且可通过使用IMC模式来改进当译码此类内容时的译码效率。图1的系统10可表示经配置以执行这些应用程序中的任一者的装置。这些应用程序中的视频内容通常是自然内容、文本、人造图形等的组合。在视频帧的文本和人造图形区中,常常存在重复图案(例如字符、图标、符号等)。如上文所介绍,IMC是如JCT-VC M0350中所报告的实现去除这类冗余且潜在地改进帧内译码效率的专用技术。如图5中所说明,对于使用IMC的译码单元(CU),从相同帧中的已经重建区域获得预测信号。最后,对指示从当前CU移位的预测信号的位置的偏移向量以及残余信号进行编码。

[0082] 举例来说,图5说明根据本发明,例如根据依据本发明的技术的帧内MC模式,用于根据来自同一图片内的预测性视频数据块的视频数据块的帧内预测的模式来预测当前图片103内的当前视频数据块102的实例技术。图5说明当前图片103内的预测性视频数据块104。视频译码器(例如,视频编码器20和/或视频解码器30)可依据根据本发明的技术的帧内MC模式使用预测性视频块104来预测当前视频块102。

[0083] 图5展示在IMC模式中经译码的当前块。当前块可(例如)为当前CU。可从搜索区108获得当前CU的预测性块。搜索区108包含与当前CU来自相同帧的已经译码块。举例来说,假定帧正以光栅扫描次序(即从左到右及从上到下)经译码,帧的已经译码块对应于在当前CU的左上方的块,如图5中所展示。在一些实例中,搜索区108可包含所述帧中的所有已经译码的块,而在其它实例中,搜索区可包含少于所有的已经译码的块。图5中的偏移向量(有时被称作运动向量或预测向量)识别当前CU的左上方像素与预测性块(图5中的经标记预测信号)的左上方像素之间的差。因此,通过在经编码视频位流中传信偏移向量,当当前CU在IMC模式中经译码时,视频解码器可识别当前CU的预测性块。

[0084] 图5是说明块内复制过程的实例的图式。视频编码器20可执行帧内BC过程以预测当前块。如上文所指出,帧内BC可以是去除图片内的冗余的专用过程。举例来说,对于使用帧内BC的译码单元(CU),视频编码器20或视频解码器30可从相同图片中的已经重建区域获得当前块。在一些情况下,视频编码器20或视频解码器30可分别编码或解码偏移或位移向量(也被称作运动向量),所述偏移或位移向量指示图片中的用于在从当前CU移位时产生当前块以及当前信号的所述块的位置。

[0085] 举例来说,图5说明帧内BC的实例,其已包括于RExt(即,HEVC的范围扩展,包含开发标准的WD5)中。在图5中,从当前图片/切片的已经解码块预测当前CU。换句话说,视频编码器20或视频解码器30可将用作参考图片的当前图片用于预测当前图片的当前CU。

[0086] 一般来说,许多应用(例如远程桌面、远程游戏、无线显示器、车用信息娱乐、云计算等)正变得常规。这些应用中的视频内容通常是自然内容、文本、人工图形等的组合。在文字和人工图形区中,重复图案(例如,字符、图标、符号等)通常存在。帧内BC是实现去除这类冗余且改进帧内译码效率的专用技术。如图5中所说明,对于使用帧内BC的译码单元(CU),从相同图片/切片中的已经重建区域获得预测信号。最后,对指示从当前CU移位的预测信号的位置的偏移向量(也被称为运动向量)以及残余信号进行编码。

[0087] 在一些实例中,视频编码器20可传信语法元素curr_pic_as_ref_enabled_flag以向视频解码器30指示指代PPS的图片是否可存在于图片本身的参考图片列表中。作为一个实例,视频编码器20可将curr_pic_as_ref_enabled_flag传信为等于1以指示指代PPS的图片可存在于图片本身的参考图片列表中。作为另一实例,视频编码器20可将curr_pic_as_ref_enabled_flag传信为等于0以指示指代PPS的图片不存在于图片本身的参考图片列表中。作为又一实例,视频编码器20可不传信curr_pic_as_ref_enabled_flag。在某一实例中,当不存在语法元素curr_pic_as_ref_enabled_flag时,视频解码器30可推断curr_pic_as_ref_enabled_flag的值等于0。在一些实例中,视频编码器20可将变量NumAddRefPic设定为等于(curr_pic_as_ref_enabled_flag?1:0)。在其它实例中,此语法元素可被另外或替代地添加到其它数据结构,例如,序列参数集(SPS)、视频参数集(VPS)、切片标头或类似者。

[0088] 在当前HEVC范围扩展标准中,使用HEVC v1中的运动向量差(MVD)译码方法来编码用于帧内BC的MV。然而,归因于帧内BC MV领域与常规帧间MV领域的不同特性,此译码方法并不有效。受此启发,在本发明中,提出若干MV译码方法。可分别或组合实施这些实例技术。举例来说,视频编码器20和视频解码器30可分别实施每一技术,或可组合实施一或多种技术。举例来说,本发明提出更高效地对MV进行译码的若干技术。

[0089] 参考MV的水平分量(即,MV_x)描述所述技术,然而所述技术可同等地可适用于MV的垂直分量(即,MV_y)。可对一维运动向量的分量以及二维运动向量的分量(即,水平分量和垂直分量)中的一者或两者使用所述技术。另外,所述技术将用于MV_x的二进制化串称为包含b0、b1、…、bn,其中bi是所述串中的第i个二进制数。这些技术可由编码器20的熵编码单元或由解码器30的熵解码单元实施。在其它实例中,这些技术可由编码器20的预测处理单元或由解码器30的预测处理单元实施。在一些实例中,这些技术可由视频编码器20和视频解码器30的单元的任何组合实施。举例来说,在视频编码器20和视频解码器30实施为集成电路(IC)或微控制器的实例中,所述技术可由IC或微控制器的一或多个单元实施。

[0090] 在本发明的第一实例中,第一二进制数b0可指示MV_x的值为非零。举例来说,在abs(MV_x) > 0的情况下b0可等于1,且在abs(MV_x) = 0的情况下b0可等于零(即,abs(MV_x) > 0, (b0=1), 否则, (b0=0))。可使用具有上下文的CABAC来编码或解码(在适当时)第一二进制数b0。在一些实例中,用于MV_x及MV_y的b0可具有单独上下文。在一些实例中,用于MV_x及MV_y的b0可共享相同上下文。在一些实例中,帧内MC的mv译码中的第i个二进制数可与帧间MC的mv译码中的第i个二进制数共享相同上下文。在一些实例中,帧内MC的mv译码及帧间MC的mv译码中的第i个二进制数可不共享上下文。

[0091] 后续二进制数b1b2…可表示abs(MV_x)-1的值。在一些实例中,可使用具有参数3的指数哥伦布码以旁路模来编码或解码(在适当时)后续二进制数b1b2…。在一些实例中,可使用指数哥伦布码的其它阶数,例如,1、2、4、5,且可使用其它代码,例如,莱斯哥伦布码。应注意,指数哥伦布码或莱斯哥伦布码的阶数可取决于CU的大小。在一些实例中,b1可表示abs(MV_x)是=1(即,b1=1)还是不=1(即,b1=0)。在一些实例中,可用旁路模式或用CABAC上下文来编码或解码(在适当时)b1。在此些实例中,b2b3…可表示abs(MV_x)-2的值且可使用具有参数3的指数哥伦布码以旁路模式来编码或解码(在适当时)。

[0092] 在一些实例中,最后一个二进制数可指示MV_x的正负号。在无任何上下文的情况下,可以旁路模式来编码或解码(在适当时)最后一个二进制数。在一些实例中,可使用具有一或多个上下文的CABAC来编码或解码(在适当时)带符号二进制数。在一些实例中,用于MV_x及MV_y的带符号二进制数可具有单独上下文。在一些实例中,用于MV_x及MV_y的带符号二进制数可共享相同上下文。

[0093] 在本发明的第二实例中,第一二进制数b0可指示后续b1b2…是否表示MV_x减去偏移(即,MV_x-offset_x)的值。举例来说,在后续b1b2…表示MV_x减去偏移的值的情况下,b0可等于一(即,b0=1),且在b1b2…不表示MV_x减去偏移的值的情况下,b0可等于零(即,b0=0)。在一些实例中,例如在b1b2…不表示MV_x减去偏移的值的情况下,b1b2…可表示MV_x的值。换句话说,第一二进制数b0可指示后续b1b2…是表示(MV_x-offset_x)的值(b0=1)还是表示(MV_x)的值(b0=0)。在一些实例中,偏移(即,offset_x)可对应于预定义值或可为CU的属性(例如,CU宽度)的函数。举例来说,偏移(offset_x)可等于CU的宽度乘以负一(即,offset_x是-CUWidth)。在一些实例中,MV_x及MV_y可具有不同偏移且每一分量具有单独b0。在一些实例中,MV_x及MV_y可共享b0,且MV_x及MV_y的偏移可相同或不同。在一些实例中,在无任何上下文的情况下,可以旁路模式来编码或解码(在适当时)b0。在一些实例中,可使用具有一或多个上下文的CABAC来编码或解码(在适当时)b0。在一些实例中,用于MV_x及MV_y的b0可具有单独上下文。在一些实例中,用于MV_x及MV_y的b0可共享相同上下文。

[0094] 在一些实例中,例如在b0=1的情况下,b1b2…可表示MV_x-offset_x。在一些实例中,例如在b0=0的情况下,b1b2…可表示MV_x。在一些实例中,上文在本发明的第一实例中上文所描述的译码方法可用于对b1b2…进行译码。换句话说,在b0之后,可用后续字符串b1b2…表示(MV_x-offset_x) (b0=1)或(MV_x) (b0=0),且上文在第一实例中所描述的方法可用作译码方法。在一些实例中,其它译码方法可用于对b1b2…进行编码或解码(在适当时)。举例来说,HEVC v1中的MVD译码方法可用于对进行b1b2…译码。

[0095] 在本发明的第三实例中,第一二进制数b0可指示后续b1b2…是否表示MV_x的绝对

值减去偏移(即, $(\text{abs}(\text{MV}_x) - \text{offset}_x)$)。举例来说,在后续b1b2…表示 MV_x 的绝对值减去偏移的情况下,b0可等于一(即,b0=1),且在后续b1b2…并不表示 MV_x 的绝对值减去偏移的情况下,b0可等于零(即,b0=0)。在一些实例中,例如在b1b2…并不表示 MV_x 的绝对值减去偏移的情况下,b1b2…可表示 MV_x 的绝对值。换句话说,第一二进制数b0可指示后续b1b2…是表示 $(\text{abs}(\text{MV}_x) - \text{offset}_x)$ 的值(b0=1)还是表示 $(\text{abs}(\text{MV}_x))$ 的值(b0=0)。在一些实例中,偏移(即,offset_x)可对应于预定义值或可为CU的属性的函数(例如,CU宽度的函数)。举例来说,偏移可等于CU的宽度乘以负一(即,offset_x=-CUWidth)。在一些实例中, MV_x 及 MV_y 可具有不同偏移且每一分量具有单独b0。在一些实例中, MV_x 及 MV_y 可共享b0,且 MV_x 及 MV_y 的偏移可相同或不同。在一些实例中,在无任何上下文的情况下,可以旁路模式来编码或解码(在适当时)b0。在一些实例中,可使用具有一或多个上下文的CABAC来编码或解码(在适当时)b0。在一些实例中,用于 MV_x 及 MV_y 的b0可具有单独上下文。在一些实例中,用于 MV_x 及 MV_y 的b0可共享相同上下文。

[0096] 在一些实例中,后续二进制数b1b2…可表示 $(\text{abs}(\text{MV}_x) - \text{offset}_x)$ 的值,例如在b0=1的情况下,或可表示 $(\text{abs}(\text{MV}_x))$ 的值,例如在b0=0的情况下。在一些实例中,可使用具有参数3的指数哥伦布码以旁路模式来编码或解码(在适当时)b1b2。在一些实例中,可使用指数哥伦布码的其它阶数,例如,1、2、4、5,且可使用其它代码,例如,莱斯哥伦布码。在一些实例中,指数哥伦布码或莱斯哥伦布码的阶数可取决于CU大小。

[0097] b1表示 $(\text{abs}(\text{MV}_x) - \text{offset}_x)$ (b0=1)=1或 $(\text{abs}(\text{MV}_x))$ (b0=0)的值是否=1可为有可能的(b1=0)。换句话说,如果b0=1,那么b1b2…可表示 $(\text{abs}(\text{MV}_x) - \text{offset}_x)$ 的值,并且,在此情况下(即,在b0=1的情况下),如果 $(\text{abs}(\text{MV}_x) - \text{offset}_x)$ 等于1,那么b1=1,且如果 $(\text{abs}(\text{MV}_x) - \text{offset}_x)$ 不等于1,那么b1=0。如果b0=0,那么b1b2…可表示 $(\text{abs}(\text{MV}_x))$ 的值,并且,在此情况下(即,在b0=0的情况下),如果 $(\text{abs}(\text{MV}_x))$ 等于1,那么b1=1,且如果 $(\text{abs}(\text{MV}_x))$ 不等于1,那么b1=0。

[0098] 在一些实例中,可用旁路模式或用CABAC上下文来编码或解码(在适当时)b1。在一些实例中,b2b3…可表示 $\text{abs}(\text{MV}_x) - 2$ 的值且可使用具有参数3的指数哥伦布码以旁路模式来编码或解码(在适当时)。在一些实例中,最后一个二进制数可指示 MV_x 的正负号,且所述二进制数可在无任何上下文的情况下以旁路模式编码或解码(在适当时)。在一些实例中,可使用具有一或多个上下文的CABAC来编码或解码(在适当时)带符号二进制数。在一些实例中,用于 MV_x 及 MV_y 的带符号二进制数可具有单独上下文。在一些实例中,用于 MV_x 及 MV_y 的带符号二进制数可共享相同上下文。

[0099] 现将描述HEVC中的运动信息、POC、CU结构及运动预测的方面。对于每一块,运动信息集合可为可用的。运动信息集合含有用于正向和后向预测方向的运动信息。在此,正向及后向预测方向是双向预测模式的两个预测方向,且术语“正向”及“后向”未必具有几何含义,而是其对应于当前图片的参考图片列表0(RefPicList0)及参考图片列表1(RefPicList1)。当仅一个参考图片列表可用于图片或切片时,仅RefPicList0可用,并且切片的每一块的运动信息始终是正向的。

[0100] 对于每个预测方向,运动信息必须含有参考索引和运动向量。在一些情况下,为简单起见,可以假定其具有相关联参考索引的方式来提起运动向量本身。参考索引用于识别当前参考图片列表(RefPicList0或RefPicList1)中的参考图片。运动向量具有水平和垂直

分量。

[0101] 图片次序计数 (POC) 广泛用于视频译码标准以识别图片的显示次序。尽管存在一个经译码视频序列内的两个图片可具有相同 POC 值的情况, 但其通常不在经译码视频序列内发生。当位流中存在多个经译码视频序列时, 具有相同 POC 值的图片可能在解码次序方面彼此更靠近。图片的 POC 值通常用于参考图片列表建构、参考图片集的导出 (如在 HEVC 中), 和运动向量按比例缩放。

[0102] 在 HEVC 中, 切片中的最大译码单元被称作译码树块 (CTB)。CTB 含有四叉树, 所述四叉树的节点为译码单元。CTB 的大小可介于 HEVC 主规范中的 16×16 到 64×64 的范围内 (尽管技术上可支持 8×8 CTB 大小)。尽管译码单元 (CU) 可与 CTB 大小相同, 但其可小至 8×8 大小。每一译码单元用一种模式译码。当 CU 经帧间译码时, CU 可进一步分割成两个预测单元 (PU), 或当进一步分割不适用时变为仅一个 PU。当两个 PU 存在于一个 CU 中时, 其可为一半大小的矩形或具有 CU 的 $1/4$ 或 $3/4$ 大小的两个矩形大小。

[0103] 当 CU 经帧间译码时, 针对每一 PU 存在一个运动信息集合。另外, 每一 PU 是以独特帧间预测模式译码以导出所述运动信息集合。在 HEVC 中, 最小 PU 大小为 8×4 和 4×8 。

[0104] 在 HEVC 标准中, 对于预测单元 (PU) 存在两个帧间预测模式, 分别称为合并 (跳过被视为合并的特殊状况) 和高级运动向量预测 (AMVP) 模式。在 AMVP 或合并模式中, 为多个运动向量预测符维持运动向量 (MV) 候选者列表。当前 PU 的运动向量 (以及合并模式中的参考索引) 是通过从 MV 候选者列表获取一个候选者而产生。

[0105] MV 候选者列表含有用于合并模式的多达 5 个候选者和用于 AMVP 模式的仅两个候选者。合并候选者可含有运动信息集合, 例如对应于两个参考图片列表 (列表 0 和列表 1) 的运动向量和参考索引。如果通过合并指数来识别合并候选者, 那么确定用于当前区块的预测的参考图片以及相关联的运动向量。然而, 在 AMVP 模式下, 对于从列表 0 或列表 1 起的每一潜在预测方向, 需要将参考索引连同对 MV 候选者列表的 MVP 索引一起明确地传信, 这是因为 AMVP 候选者仅含有运动向量。在 AMVP 模式中, 可进一步优化经预测运动向量。如可从上文看出, 合并候选者对应于整个运动信息集合, 而 AMVP 候选者仅含有用于特定预测方向的一个运动向量和参考索引。以类似方式从相同空间及时间相邻块导出用于两种模式的候选者。

[0106] 视频编码器 20 从一组先前经重建视频数据块选择用于预测当前视频块 102 的预测性视频块 104。视频编码器 20 通过反量化和反变换也包含在经编码视频位流中的视频数据且对所得残余块与用于预测经重建视频数据块的预测性块求和而重建视频数据块。在图 5 的实例中, 图片 103 内的既定区 108 也可被称作“既定区域”或“光栅区域”, 包含先前经重建视频块的集合。视频编码器 20 可以多种方式定义图片 103 内的既定区 108, 如下文更详细描述。视频编码器 20 可根据基于既定区 108 内的各种视频块预测和译码当前视频块 102 的相对效率和准确性的分析而从既定区 108 中的视频块当中选择预测性视频块 104 以预测当前视频块 102。

[0107] 视频编码器 20 确定表示预测性视频块 104 相对于当前视频块 102 的位置或位移的二维向量 106。为偏移向量的实例的二维向量 106 包含水平位移分量 112 及垂直位移分量 110, 所述水平位移分量及垂直位移分量分别表示预测性视频块 104 相对于当前视频块 102 的水平及垂直位移。视频编码器 20 可将识别或定义二维向量 106 (例如, 定义水平位移分量 112 及垂直位移分量 110) 的一个或多个语法元素包含在经编码视频位流中。视频解码器 30 可

解码所述一或多个语法元素以确定二维运动向量106,且使用所确定的向量来识别用于当前视频块102的预测性视频块104。

[0108] 在一些实例中,二维向量106的分辨率可为整数像素,例如,被限制为具有整数像素分辨率。在此些实例中,水平位移分量112和垂直位移分量110的分辨率将为整数像素。在此些实例中,视频编码器20和视频解码器30不需要内插预测性视频块104的像素值以确定用于当前视频块102的预测符。

[0109] 在其它实例中,水平位移分量112和垂直位移分量110中的一者或两者的分辨率可为子像素。举例来说,分量112和110中的一者可具有整数像素分辨率,而另一者具有子像素分辨率。在一些实例中,水平位移分量112和垂直位移分量110两者的分辨率可为子像素,但水平位移分量112和垂直位移分量110可具有不同分辨率。

[0110] 在一些实例中,视频译码器(例如,视频编码器20和/或视频解码器30)基于特定层级来适配水平位移分量112和垂直位移分量110的分辨率,例如块层级、切片层级或图片层级适配。举例来说,视频编码器20可在切片层级下传信旗标,例如,在切片标头中,所述旗标指示水平位移分量112及垂直位移分量110的分辨率是整数像素分辨率还是并非整数像素分辨率。如果所述旗标指示水平位移分量112和垂直位移分量110的分辨率不是整数像素分辨率,那么视频解码器30可推断分辨率是子像素分辨率。在一些实例中,不一定是旗标的一或多个语法元素可针对每一切片或其它视频数据单元经发射以指示水平位移分量112和/或垂直位移分量110的共同或个别分辨率。

[0111] 在又其它实例中,替代旗标或语法元素,视频编码器20可基于分辨率上下文信息而设定且视频解码器30可从分辨率上下文信息推断水平位移分量112和/或垂直位移分量110的分辨率。举例来说,分辨率上下文信息可包含色彩空间(例如,YUV、RGB或类似者)、特定色彩格式(例如,4:4:4、4:2:2、4:2:0或类似者)、帧大小、帧速率,或用于包含当前视频块102的图片或图片序列的量化参数(QP)。在至少一些实例中,视频译码器可基于与经先前译码帧或图片相关的信息来确定水平位移分量112和/或垂直位移分量110的分辨率。以此方式,水平位移分量112的分辨率和垂直位移分量110的分辨率可经预定义、传信、可从其它辅助信息(例如,分辨率上下文信息)推断,或可基于已经译码的帧。

[0112] 当前视频块102可为CU,或CU的PU。在一些实例中,视频译码器(例如,视频编码器20和/或视频解码器30)可将根据IMC预测的CU分裂为多个PU。在此些实例中,视频译码器可确定用于CU的PU中的每一者的相应(例如,不同)二维向量106。举例来说,视频译码器可将 $2N \times 2N$ CU分裂为两个 $2N \times N$ PU、两个 $N \times 2N$ PU,或四个 $N \times N$ PU。作为其它实例,视频译码器可将 $2N \times 2N$ CU分裂为 $((N/2) \times N + (3N/2) \times N)$ PU、 $((3N/2) \times N + (N/2) \times N)$ PU、 $(N \times (N/2) + N \times (3N/2))$ PU、 $(N \times (3N/2) + N \times (N/2))$ PU、四个 $(N/2) \times 2N$ PU或四个 $2N \times (N/2)$ PU。在一些实例中,视频译码器可使用 $2N \times 2N$ PU预测 $2N \times 2N$ CU。

[0113] 当前视频块102包含明度视频块(例如,明度分量)及对应于所说明度视频块的色度视频块(例如,色度分量)。在一些实例中,视频编码器20可仅将定义明度视频块的二维向量106的一或多个语法元素编码到经编码视频位流中。在此些实例中,视频解码器30可基于针对明度块传信的二维向量导出对应于所说明度块的一或多个色度块中的每一者的二维向量106。在本发明中所描述的技术中,在为一或多个色度块导出二维向量时,如果明度块的二维向量指向色度样本内的子像素位置,那么视频解码器30可修改所说明度块的二维向

量。

[0114] 取决于色彩格式,例如色彩取样格式或色度取样格式,视频译码器可相对于明度视频块下取样对应色度视频块。色彩格式4:4:4不包含下取样,意味着色度块在水平和垂直方向中包含与明度块数目相同的样本。色彩格式4:2:2在水平方向上下取样,意味着在色度块中在水平方向上相对于明度块存在二分之一多的样本。色彩格式4:2:0在水平和垂直方向中下取样,意味着在色度块中在水平和垂直方向中相对于明度块存在二分之一多的样本。

[0115] 在其中视频译码器基于对应明度块的向量106确定色度视频块的向量106的实例中,视频译码器可需要修改明度向量。举例来说,如果明度向量106具有整数分辨率,其中水平位移分量112和/或垂直位移分量110为奇数数目的像素,且色彩格式是4:2:2或4:2:0,那么经转换明度向量可不指向对应色度块中的整数像素位置。在此些实例中,视频译码器可按比例缩放明度向量以用作色度向量来预测对应色度块。

[0116] 图6展示可用于导出用于以合并模式或AMVP模式译码的特定PU(PU_0)的空间MV候选者的相邻块的实例。然而,用于从块产生候选者的技术针对合并和AMVP模式不同。举例来说,在合并模式中,可根据某一次序检查图6中所展示的五个空间MV候选者中的每一者的可用性。举例来说,HEVC使用次序: a_1, b_1, b_0, a_0, b_2 。

[0117] 在AMVP模式中,将相邻块划分成两个群组:由块 a_0 及 a_1 组成的左群组以及由块 b_0 、 b_1 及 b_2 组成的上群组,如图6中所展示。对于左群组,根据次序: $\{a_0, a_1\}$ 来检查可用性。对于上群组,根据次序: $\{b_0, b_1, b_2\}$ 来检查可用性。对于每一群组,指代与由经传信参考索引所指示的相同的参考图片的相邻块中的潜在候选者具有有待被选以形成所述群组的最终候选者的最高优先级。有可能所有相邻块均不含指向同一参考图片的运动向量。因此,如果无法找到此类候选者,那么将按比例缩放第一可用候选者以形成最终候选者,因此可补偿时间距离差异。

[0118] 现将论述用于HEVC中的合并/层级的色度译码及平行处理的运动向量。运动向量在用于色度运动补偿之前针对当前PU/CU的明度分量而导出,所述运动向量基于色度取样格式按比例缩放。

[0119] 在HEVC中,可将LCU划分成若干并行运动估计区(MER),且仅允许将属于与当前PU不同的MER的那些相邻PU包含于合并/跳过MVP列表建构过程中。在图片参数集中将MER的大小传信为 $log2_parallel_merge_level_minus2$ 。当MER大小大于 $N \times N$ 时,其中 $2N \times 2N$ 是最小CU大小,MER以如果空间相邻块在与当前PU相同的MER内部那么所述空间相邻块被视为不可用的方式起作用。

[0120] 块内复制(BC)已包含于当前SCC中。在上文图5中展示帧内BC的实例,其中当前CU/PU是从当前图片/切片的已经解码块预测的。应注意,预测信号经重建而无需环路内滤波(包含解块和样本自适应偏移(SAO))。

[0121] 在块补偿中,对于以帧内BC译码的明度分量或色度分量,通过整数块补偿完成块补偿。因此,不需要内插。块向量在整数层级下预测及传信。

[0122] 在当前SCC中,在每一CTB的开始处将块向量预测值设定成 $(-w, 0)$,其中w是CU的宽度。如果以帧内BC模式来译码,那么将此块向量预测符更新为最新经译码CU/PU中的所述一者。如果并未以帧内BC来译码CU/PU,那么块向量预测符保持不变。在块向量预测之后,使用

HEVC中的MV差(MVD)译码方法来编码块向量差。

[0123] 在CU及PU层级两者下启用当前帧内BC。对于PU层级帧内BC,针对全部CU大小支持 $2N \times N$ 和 $N \times 2N$ PU分区。另外,当CU是最小CU时,支持 $N \times N$ PU分割。

[0124] 根据本发明的技术,视频译码装置可将帧内BC块视为帧间块。在JCTVC-R0100中,提出帧内BC模式及帧间模式的统一。因此,当前图片可用作参考图片且被添加到参考列表中。一旦将当前图片添加到参考列表,按帧内BC块以帧间模式译码那样来处理所述块。在当前图片解码之前将当前图片标记为长期的,且在解码当前图片之后标记为短期的。当启用帧内BC时,针对I切片按照P切片的语法解析过程及解码过程进行。

[0125] 现有IBC技术可具有一些缺点。举例来说,当将帧内BC视为帧间时,常规切片类型传信将造成在解码过程中检查的额外状况。并且,对于时间运动向量预测符(TMVP)、受限帧内预测、帧内BC MV精度等等之间的相互作用来说可能存在一些问题。

[0126] 根据本发明的技术,为了实现帧内BC模式及帧间模式的更好统一,在可将帧内BC模式视为帧间模式且将当前图片添加到参考列表0((参考列表1或两者)的情况下,提出后续技术。尽管预测模式可以相同(MODE_INTER),但视频译码装置可通过检查由与块相关联的参考索引识别的参考图片是否为当前图片(即参考图片是否与当前图片具有相同POC值)来区分帧内BC块与常规帧间块。如果参考图片及当前图片具有相同POC值,那么当前块可被视为帧内BC块。否则,当前块可被视为常规帧间块。

[0127] 在统一帧内BC模式及帧间模式中,视频译码装置可实施许多其它变化。举例来说,IRAP图片(即,VCL NAL单元具有在BLA_W_LP到RSV_IRAP_VCL23范围内的nal_unit_type)可不含I切片。实际上,IRAP图片可包含P切片或B切片。在此些实例中,P切片的运动向量参考当前图片内的块。在其它实例中,B切片的双向预测运动向量都参考当前图片内的视频块。在IRAP图片含有B切片的情况下,视频译码装置可进一步传信待应用于B切片的不同运动向量的权重。举例来说,一个运动向量可具有25%的权重且第二运动向量可具有75%的权重。

[0128] 此外,IRAP图片可含有指定非空RPS的语法元素。举例来说,RPS可包含表示 Δ (delta) POC值或参考图片可能处于的POC值的范围的语法元素。根据HEVC标准,RPS可能含有其它语法元素,例如定义RPS的各种属性及可包含于RPS中的图片的种类的旗标。

[0129] 参考图片可包含当前图片的已经解码的未经滤波(SAO/解块)样本。此参考图片可包含于参考图片集中。当从此参考图片(当前图片)应用预测时,其可针对解码过程的剩余部分被视为帧间模式。在一个实例中,在切片标头、SPS、PPS、VPS中或其它处发射旗标,其指示当前图片是否用作用于帧间预测的参考图片。

[0130] 在一个实例中,当指代SPS的特定图片可包含于本身等于1的特定图片的参考图片列表中时,CSV中可能不存在指代SPS的I切片。在另一个实例中,当指示指代PPS的特定图片是否可包含于特定图片本身的参考图片列表中的语法元素等于1时(即,指示特定图片包含于参考图片列表中),图片中可能不存在指代PPS的I切片。实例可扩展到VPS或切片标头粒度层级。

[0131] 在一个实例中,当将当前图片用作参考图片时,如下修改变量NumPicTotalCurr(即,可用作用于预测当前图片的参考的图片的数目):

```

NumPicTotalCurr = 0
for( i = 0; i < NumNegativePics[ CurrRpsIdx ]; i++ )
    if( UsedByCurrPicS0[ CurrRpsIdx ][ i ] )
        NumPicTotalCurr++
for( i = 0; i < NumPositivePics[ CurrRpsIdx ]; i++ ) (7-54)
    if( UsedByCurrPicS1[ CurrRpsIdx ][ i ] )
        NumPicTotalCurr++
[0132] for( i = 0; i < num_long_term_sps + num_long_term_pics; i++ )
    if( UsedByCurrPicLt[ i ] )
        NumPicTotalCurr++
    if( curr_pic_as_ref_enabled_flag )
        NumPicTotalCurr++

```

[0133] 变量NumPicTotalCurr可取决于特定图片是否可包含于特定图片本身的参考图片列表中。

[0134] 语法元素curr_pic_as_ref_enabled_flag可经定义使得将curr_pic_as_ref_enabled_flag设定成等于1以指定指代SPS的特定图片可包含于特定图片本身的参考图片列表中。相反地,将curr_pic_as_ref_enabled_flag设定成等于0以指定指代SPS的特定图片从不包含于特定图片本身任何参考图片列表中。当语法元素本身不存在时,推断语法元素curr_pic_as_ref_enabled_flag的值等于0。此外,如果当前图片为BLA或CRA图片,那么NumPicTotalCurr的值将等于curr_pic_as_ref_enabled_flag。否则,当当前图片含有P切片或B切片时,NumPicTotalCurr的值将不等于curr_pic_as_ref_enabled_flag。此外,语法元素NumPicTotalCurr可小于或等于8+curr_pic_as_ref_enabled_flag。

[0135] 在一些实例中,当当前图片用作参考图片时,用于建构参考图片列表的导出过程经修改使得变量NumRpsCurrTempList0被设定成等于Max (num_ref_idx_10_active_minus1 +1, NumPicTotalCurr)。此外,如下建构列表RefPicListTemp0:

[0136]

rIdx = 0

while(rIdx < NumRpsCurrTempList0) {

for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0; rIdx++, i++)

RefPicListTemp0[rIdx] = RefPicSetStCurrBefore[i]

if(curr_pic_as_ref_enabled_flag)

RefPicListTemp0[rIdx++] = currPic

for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0; rIdx++, i++)

(8-8)

RefPicListTemp0[rIdx] = RefPicSetStCurrAfter[i]

for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++)

RefPicListTemp0[rIdx] = RefPicSetLtCurr[i]

}

[0137] 如下建构列表RefPicList0:

[0138] for (rIdx=0; rIdx<=num_ref_idx_10_active_minus1; rIdx++) (8-9)

[0139] RefPicList0[rIdx] = ref_pic_list_modification_flag_10?

[0140] RefPicListTemp0[list_entry_10[rIdx]] : RefPicListTemp0[rIdx]

[0141] 当切片是B切片时,将变量NumRpsCurrTempList1设定为等于Max (num_ref_idx_11_active_minus1+1, NumPicTotalCurr) 且如下建构列表RefPicListTemp1:

[0142]

rIdx = 0

while(rIdx < NumRpsCurrTempList1) {

for(i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList1; rIdx++, i++)

RefPicListTemp1[rIdx] = RefPicSetStCurrAfter[i]

if(curr_pic_as_ref_enabled_flag)

[0143]

RefPicListTemp1[rIdx++] = currPic

for(i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList1; rIdx++, i++) (8-10)

RefPicListTemp1[rIdx] = RefPicSetStCurrBefore[i]

for(i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList1; rIdx++, i++)

RefPicListTemp1[rIdx] = RefPicSetLtCurr[i]

}

[0144] 如上文所描述,对应于帧内BC的MV可经译码且以整数像素精度或子像素精度(例

如HEVC中的1/4像素精度)使用。一般来说,帧内BC MV精度可为与帧间模式中的MV精度不同的精度。在下文一些实例中,针对整数像素帧内BC MV精度提供所描述技术;然而,应理解,并非整数像素的其它精度可应用且应被视为经本发明的范围内。

[0145] 在一个实例中,可针对切片标头、SPS、PPS、VPS或其它处中的块发射旗标。旗标可指示所述范围内的当前帧内BC块、帧内BC块群组或所有帧内BC块是否可具有以整数像素精度或子像素精度(例如1/4、1/8像素及类似精度,(例如)如HEVC中所定义的)保持的MV。举例来说,范围可为块、块的群组、切片、图片或整个序列。

[0146] 在一个实例中,与帧内BC块相关联的MV可具有整数像素精度,即MV可经译码且与整数像素精度下的块相关联。在此些实例中,可透明地使用AMVP导出、合并候选者导出或相关过程的MVD译码而无需后续方法中的任一者的任何变化。首先,假定通过将当前图片添加到参考图片列表来启用帧内BC且将参考图片经标记为长期的,在用于帧内BC块的MV预测符导出过程期间可能不允许常规长期参考图片的使用以避免与MV按比例缩放相关的问题,确切地说其可导致低效MV候选者。相反地,假定通过将当前图片添加到参考列表来启用帧内BC且将此参考图片标记为长期参考图片,在用于帧内BC块的MV预测符导出过程期间可仍然允许常规长期参考图片。然而,根据HEVC设计,不管精度是否为整数像素精度,可直接地使用对应于长期参考图片的MV的值而无需MV候选者导出过程中的MV按比例缩放。

[0147] 替代地或另外,当帧内BC MV具有整数像素精度时,可以用于MV候选者导出过程(AMVP及合并模式)的空间相邻块或并置块的MV可指向常规长期参考图片。在此情况下,那些MV预测符候选者在用于候选者导出过程之前可转换为整数像素精度。并且,可在帧内BC块的MV预测符候选者导出过程中的精简过程之前或之后执行此精度转换过程。举例来说,转换可通过舍入过程完成,例如右移或经舍入右移。替代地,MV预测符候选者导出过程可在处于常规帧间模式中时保持不变,但将最终候选者列表中的MV转换为帧内BC准确度(整数像素)。类似地,在帧内BC MV用作用于常规帧间的预测符导出过程的候选者之前,其被转换为常规帧间的MV精度,例如通过左移。

[0148] 在另一个实例中,在帧内BC块的MV预测符导出过程期间,假定通过将当前图片添加到参考图片列表来启用帧内BC且将参考图片标记为长期的,如果空间相邻块或并置块的参考图片是常规长期图片(不是当前图片),那么所述空间相邻块或并置块被视为不可用是有可能的。类似地,在帧间块的MV预测符导出过程期间,其中参考图片为常规长期参考图片(不是当前图片),如果空间相邻块或并置块是帧内BC块,那么其被视为不可用是有可能的。替代地,在MV预测符导出过程期间,帧内BC块及帧间块在参考图片为常规长期参考图片的情况下被视为可用于彼此。

[0149] 在另一个实例中,当用于帧内BC模式的帧内BC MV经译码且与处于和常规帧间模式精度(例如HEVC中的1/4像素精度)相同的精度下的块相关联时,AMVP、合并或MVD译码相关过程可透明地使用而无需任何变化。替代地,MV可以整数像素精度译码但在帧内BC块MV可具有与常规帧间模式相同的精度的情况下经存储。因此,也可以整数像素精度译码MVD。在此情况下,在解码器侧,可将与常规帧间处于相同精度的所有存储MV导出为:

[0150] $MV = (MVP >> 2 + MVD) \ll 2,$

[0151] 其中MVP是帧间模式精度下的对应MV预测符,且MVD可具有整数像素准确度。其它转换机制或舍入是可适用的。对于帧内BC合并,MVD为零,且MVP为来自对应合并候选者的

MV。

[0152] 如果帧内BC MV经存储具有整数像素精度,那么帧内BC MV可转换为帧间模式MV精度。举例来说,帧内BC MV可经受左移,随后可使用常规帧间模式的整个解块模块而无需任何变化。可根据MV值导出解块滤波器强度,这是因为帧内BC及帧间模式可具有不同MV精度,对于更好的图片质量来说可能希望MV精度相等,例如帧内BC。

[0153] 如果以整数像素精度存储帧内BC MV,那么帧内BC MV可转换为常规帧间模式的MV精度,例如通过左移,随后可使用常规帧间模式的明度运动补偿模块而无需任何变化。对于色度分量,视频译码器可使用常规帧间模式的色度运动补偿模块而无需任何变化,或当色度取样不是4:4:4且分割模式不是 $2N \times 2N$ 时视频译码器可使用其它色度运动补偿方法。

[0154] 在以下情况下,可能不允许用于MV预测符导出及合并的TMVP。可以不同方式不允许TMVP。举例来说,可停用TMVP启用旗标,或与TMVP候选者相关联的合并指数或MVP指数可能不在位流中传信及用于预测。在另一个实例中,当通过将当前图片添加到参考列表来启用帧内BC时,如果参考列表仅含有当前图片,那么不允许用于MV预测符导出及合并的TMVP。此外,可能不发射TMVP启用旗标,推断所述旗标为0。在另一个实例中,当启用帧内BC及受限帧内预测两者时,假定用于帧内或帧内BC块的预测可仅从帧内块或帧内BC块起,MV预测符导出及合并过程不允许TMVP。

[0155] 帧内BC所允许的分割模式可能与常规帧间不同或相同。在一个实例中,帧内BC允许的分割模式及分割模式传信恰好与常规帧间相同,除了当CU大小是 8×8 时允许 $N \times N$ 分割模式以外,这意味着帧内BC块大小可为 4×4 。有可能的是,是否允许 4×4 帧内BC块取决于帧内BC的块向量精度。举例来说,仅当帧内BC的MV精度是整数像素时允许 4×4 帧内BC块。当允许 4×4 帧内BC块时,其可受限使得如果相同CU中存在任何 4×4 帧内BC块,那么所述 4×4 块应为帧内BC块。替代地,当允许 4×4 帧内BC块时,对于每一CU,其可包含 4×4 帧内BC块及 4×4 常规帧间块两者。在此情况下,其可限制 4×4 帧内BC块及 4×4 帧间块两者的MV精度为整数像素。在另一个实例中,帧内BC允许的分割模式及分割模式传信恰好与常规帧间相同。举例来说,可能不存在 4×4 帧内BC块。因此,可使用分割模式的解析及解码模块而无需任何变化。

[0156] 帧内BC的MV预测符导出可能与常规帧间不同或相同。在一个实例中,MV预测符导出与常规帧间相同。在另一个实例中,MV预测符导出将AMVP用于常规帧间模式,且将其它方法用于帧内BC模式。此外,帧内BC的MVD译码可能与常规帧间不同或相同。

[0157] 所存储帧内BC MV可具有与常规帧间预测相同的准确度,且可仅针对某些色彩分量进行到帧内BC准确度(例如整数像素)的MV转换。举例来说,明度分量可使用所存储的MV准确度(较高MV精度),且色度分量可将MV转换到整数像素准确度(较低MV精度),例如在色度运动补偿步骤期间。

[0158] 当特定图片包含于本身等于1的特定图片的参考图片列表中时,且当当前预测单元的refIdx指向图片本身时,随后位流一致性可要求:CU内的所有预测单元应具有相同refIdx。也就是说,当至少一个预测单元从CU中的当前图片开始参考时,所有预测单元将仅从当前图片开始参考。当特定图片包含于本身等于1的特定图片的参考图片列表中时,且当当前预测单元的refIdx指向图片本身时,随后位流一致性可要求:预测样本(来自当前图片)中的无一者将跨过图片边界。换句话说,来自当前图片的每一预测样本可在当前图片的

图片边界内。

[0159] 其可受限使得不能将当前图片用作用于当前图片本身的TMVP中的并置图片。替代地,将当前图片用作TMVP中的并置图片也是可能的。在此情况下,将TMVP设定成不可用,或可用的处理TMVP且用于TMVP的MV可用预定义的默认MV来设定,例如但限于: (-2w, 0)、(2w, 0)、(-w, 0)、(w, 0)、(0, -h)、(0, -2h)、(0, h)、(0, 2h)、(-8, 0)、(0, 8)、(0, 0),其中w及h是当前CU、PU或CTB的宽度及高度。

[0160] 当自适应MV分辨率在不同层级(例如,MV分量、PU/CU/CTU/切片/PPS/SPS/VPS)下用于常规帧间(例如JCTVC-0085)时,可使用整数像素精度或分数像素精度来译码MV。作为一个实例,在常规帧间模式的MV是以整数像素精度译码但以四分之一像素精度存储时。因此,也可以整数像素精度译码MVD。在此情况下,在解码器侧,可将具有四分之一像素精度的所有存储MV导出为:

[0161] $MV = (MVP >> 2 + MVD) \ll 2$,

[0162] 其中MVP是帧间模式精度下的对应MV预测符,且MVD可具有整数像素准确度。其它转换机制或舍入是可适用的。对于帧内BC合并,MVD为零,且MVP为来自对应合并候选者的MV。

[0163] 作为另一实例,当以整数像素精度译码常规帧间模式的MV时,MV可以整数像素精度存储,且MVD也可以整数像素精度译码。AMVP中的MVP候选者或合并中合并候选者的导出可与HEVC中相同,通过直接使用AMVP候选者或合并候选者的MV值而无需考虑这些MV值是以整数像素精度还是分数像素精度存储。在另一个实例中,AMVP中的四分之一像素精度MVP候选者或合并中的四分之一像素精度合并候选者在使用之前首先转换为整数像素精度。

[0164] BLA图片并不指代用于其译码的任何其它图片,且按解码次序可为位流中的第一图片或稍后在位流中呈现。每一BLA图片开始新的CVS,且对解码过程具有与IDR图片相同的作用。然而,BLA图片含有指定非空RPS的语法元素。当BLA图片的每一VCL NAL单元具有等于BLA_W_LP的nal_unit_type时,所述BLA图片可以具有相关联的RASL图片,所述RASL图片未由解码器输出且可能不可解码,因为所述RASL图片可能含有对不存在于所述位流中的图片的参考。当BLA图片的每一VCL NAL单元具有等于BLA_W_LP的nal_unit_type时,所述BLA图片还可以具有相关联的RADL图片,所述RADL图片被指定解码。当BLA图片的每一VCL NAL单元具有等于BLA_W_RADL的nal_unit_type时,所述BLA图片不具有相关联的RASL图片但可具有相关联的RADL图片。当BLA图片的每一VCL NAL单元具有等于BLA_N_LP的nal_unit_type时,所述BLA图片不具有任何相关联的前导图片。

[0165] CRA图片并不指代用于其译码的任何其它图片,且按解码次序可为位流中的第一图片,或稍后可在位流中呈现。CRA图片可以具有相关联的RADL或RASL图片。与BLA图片相同,CRA图片可含有指定非空RPS的语法元素。当CRA图片具有等于1的NoRas10OutputFlag时,解码器不输出相关联的RASL图片,因为所述RASL图片可能不可解码,因为所述RASL图片可能包含对不存在于所述位流中的图片的参考。

[0166] IDR图片并不指代用于其译码的任何其它图片,且按解码次序可为位流中的第一图片或稍后在位流中呈现。每一IDR图片为按解码次序的CVS的第一图片。当IDR图片的每一VCL NAL单元具有等于IDR_W_RADL的nal_unit_type时,其可具有相关联的RADL图片。当IDR图片的每一VCL NAL单元具有等于IDR_N_LP的nal_unit_type时,所述IDR图片不具有任何

相关联的前导图片。IDR图片不具有相关联的RASL图片。

[0167] 帧间预测可为以取决于一或多个参考图片的数据元素(例如样本值或运动向量)的方式导出的预测。帧内预测为从同一经解码切片的仅数据元素(例如样本值)导出而无需指代参考图片的预测。

[0168] IRAP图片并不指代用于其译码的任何其它图片,且可为BLA图片、CRA图片或IDR图片。按解码次序在位流中的第一图片必定是IRAP图片。假设必要参数集在其需要激活时可用,那么IRAP图片和按解码次序的所有随后非RASL图片可正确地解码,而无需执行按解码次序在IRAP图像前的任何图像的解码过程。位流中可能存在并不指代用于其译码的任何其它图片的并非IRAP图片的图片。参考图片含有可用于按解码次序的当前图片或随后图片的解码过程中的帧间预测的样本。

[0169] 参考图片集可为与图片相关联的一组参考图片,所述参考图片集由按解码次序在所述相关联图片之前的所有参考图片组成,所述参考图片可用于所述相关联图片或按解码次序在所述相关联图片之后的任何图片的帧间预测。

[0170] 到译码过程的输入可包含:当前明度译码块的左上方样本相对于当前图片的左上方明度样本的明度位置(x_{Cb}, y_{Cb})、指定当前明度译码块的大小的变量 n_{CbS} 、当前明度预测块的左上方样本相对于当前图片的左上方明度样本的明度位置(x_{Pb}, y_{Pb})、指定当前明度预测块的宽度及高度的两个变量 n_{PbW} 及 n_{PbH} 、指定当前译码单元内的当前预测单元的分割索引的变量 $partIdx$,及由相邻预测块相对于当前图片的左上方明度样本覆盖的明度位置(x_{NbY}, y_{NbY})。

[0171] 译码过程的输出为覆盖位置(x_{NbY}, y_{NbY})的相邻预测块的可用性,其表示为 $availableN$,所述输出导出如下:变量 $sameCb$ 指定当前明度预测块及相邻明度预测块是否覆盖同一亮度译码块。如果所有以下条件为TRUE,那么将 $sameCb$ 设定成等于TRUE: x_{Cb} 小于或相等 x_{NbY}, y_{Cb} 小于或相等 $y_{NbY}, (x_{Cb}+n_{CbS})$ 大于 x_{NbY} ,及 $(y_{Cb}+n_{CbS})$ 大于 y_{NbY} 。否则,将 $sameCb$ 设定成等于FALSE。

[0172] 如下导出相邻预测块可用性 $availableN$:如果 $sameCb$ 等于FALSE,那么如子条款6.4.1中所指定的用于z扫描次序块可用性的导出过程经调用且(x_{Curr}, y_{Curr})被设定成等于(x_{Pb}, y_{Pb})且明度位置(x_{NbY}, y_{NbY})作为输入,且将输出指派给 $availableN$ 。否则,如果所有以下条件为TRUE,那么将 $availableN$ 设定成等于FALSE: $(n_{PbW} < 1)$ 等于 n_{CbS} 、 $(n_{PbH} < 1)$ 等于 n_{CbS} 、 $partIdx$ 等于1、 $(y_{Cb}+n_{PbH})$ 小于或等于 y_{NbY} ,及 $(x_{Cb}+n_{PbW})$ 大于 x_{NbY} 。否则,将 $availableN$ 设定成等于TRUE。当 $availableN$ 等于TRUE、 $CuPredMode[x_{Pb}][y_{Pb}]$ 等于 $MODE_INTER$,且 $CuPredMode[x_{NbY}][y_{NbY}]$ 等于 $MODE_INTRA$ 时,将 $availableN$ 设定成等于FALSE。

[0173] 本文中所描述的各种语法元素的语法包含下表。通过这些表更新的HEVC标准的相关部分将提供于每一表的标头中。

[0174] 章节7.3.2.2.3

[0175]

sps_scc_extensions() {	描述符
curr_pic_as_ref_enabled_flag	u(1)
palette_mode_enabled_flag	u(1)
residual_adaptive_colour_transform_enabled_flag	u(1)
}	

[0176] 章节7.3.8.5

[0177]

coding_unit(x0, y0, log2CbSize) {	描述符
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(palette_mode_enabled_flag && ChromaArrayType == 3)	
palette_mode_flag[x0][y0]	ae(v)
if(palette_mode_flag[x0][y0])	
palette_coding(x0, y0, nCbS)	
else {	
if(CuPredMode[x0][y0] != MODE_INTRA	
log2CbSize == MinCbLog2SizeY)	
part_mode	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag &&	
log2CbSize >= Log2MinIpcmCbSizeY &&	

[0178]

log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag [x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample (x0, y0, log2CbSize)	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
prev_intra_luma_pred_flag [x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(prev_intra_luma_pred_flag [x0 + i][y0 + j])	
mpm_idx [x0 + i][y0 + j]	ae(v)
else	
rem_intra_luma_pred_mode [x0 + i][y0 + j]	ae(v)
if(ChromaArrayType == 3)	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_chroma_pred_mode [x0 + i][y0 + j]	ae(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else {	
if(PartMode == PART_2Nx2N)	
prediction_unit (x0, y0, nCbS, nCbS)	
else if(PartMode == PART_2NxN) {	
prediction_unit (x0, y0, nCbS, nCbS / 2)	
prediction_unit (x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
} else if(PartMode == PART_Nx2N) {	
prediction_unit (x0, y0, nCbS / 2, nCbS)	
prediction_unit (x0 + (nCbS / 2), y0, nCbS / 2, nCbS)	
} else if(PartMode == PART_2NxN) {	
prediction_unit (x0, y0, nCbS, nCbS / 4)	
prediction_unit (x0, y0 + (nCbS / 4), nCbS, nCbS * 3 / 4)	
} else if(PartMode == PART_2NxN) {	
prediction_unit (x0, y0, nCbS, nCbS * 3 / 4)	
prediction_unit (x0, y0 + (nCbS * 3 / 4), nCbS, nCbS / 4)	
} else if(PartMode == PART_nLx2N) {	
prediction_unit (x0, y0, nCbS / 4, nCbS)	
prediction_unit (x0 + (nCbS / 4), y0, nCbS * 3 / 4, nCbS)	
} else if(PartMode == PART_nRx2N) {	
prediction_unit (x0, y0, nCbS * 3 / 4, nCbS)	
prediction_unit (x0 + (nCbS * 3 / 4), y0, nCbS / 4, nCbS)	
} else { /* PART_NxN */	
prediction_unit (x0, y0, nCbS / 2, nCbS / 2)	
prediction_unit (x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
prediction_unit (x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
prediction_unit (x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	

[0179]

[0180] 章节7.3.8.9

[0181]

transform_tree(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx) {	描述符
if(log2TrafoSize <= MaxTbLog2SizeY &&	
log2TrafoSize > MinTbLog2SizeY &&	
trafoDepth < MaxTrafoDepth && !(IntraSplitFlag && (trafoDepth == 0)))	
split_transform_flag [x0][y0][trafoDepth]	ae(v)
if((log2TrafoSize > 2 && ChromaArrayType != 0) ChromaArrayType == 3) {	
if(trafoDepth == 0 cbf_cb[xBase][yBase][trafoDepth - 1]) {	
cbf_cb [x0][y0][trafoDepth]	ae(v)
if(ChromaArrayType == 2 &&	
(!split_transform_flag[x0][y0][trafoDepth] log2TrafoSize == 3))	
cbf_cb [x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth]	ae(v)
}	
if(trafoDepth == 0 cbf_cr[xBase][yBase][trafoDepth - 1]) {	
cbf_cr [x0][y0][trafoDepth]	ae(v)
if(ChromaArrayType == 2 &&	
(!split_transform_flag[x0][y0][trafoDepth] log2TrafoSize == 3))	
cbf_cr [x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth]	ae(v)
}	
}	
if(split_transform_flag[x0][y0][trafoDepth]) {	
x1 = x0 + (1 << (log2TrafoSize - 1))	
y1 = y0 + (1 << (log2TrafoSize - 1))	
transform_tree(x0, y0, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 0)	
transform_tree(x1, y0, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 1)	
transform_tree(x0, y1, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 2)	
transform_tree(x1, y1, x0, y0, log2TrafoSize - 1, trafoDepth + 1, 3)	
} else {	

[0182]

if(CuPredMode[x0][y0] == MODE_INTRA	
trafoDepth != 0 cbf_cb[x0][y0][trafoDepth]	
cbf_cr[x0][y0][trafoDepth] (ChromaArrayType == 2 &&	
(cbf_cb[x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth]	
cbf_cr[x0][y0 + (1 << (log2TrafoSize - 1))][trafoDepth])))	
cbf_luma [x0][y0][trafoDepth]	ae(v)
transform_unit(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx)	
}	
}	

[0183] 章节7.3.8.12

[0184]

描述符
transform_unit(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx) {
log2TrafoSizeC = Max(2, log2TrafoSize - (ChromaArrayType == 3 ? 0 : 1))
cbfDepthC = trafoDepth - (ChromaArrayType != 3 && log2TrafoSize == 2 ? 1 : 0)
xC = (ChromaArrayType != 3 && log2TrafoSize == 2) ? xBase : x0
yC = (ChromaArrayType != 3 && log2TrafoSize == 2) ? yBase : y0
cbfLuma = cbf_luma[x0][y0][trafoDepth]
cbfChroma =
cbf_cb[xC][yC][cbfDepthC]
cbf_cr[xC][yC][cbfDepthC]
(ChromaArrayType == 2 &&
(cbf_cb[xC][yC + (1 << log2TrafoSizeC)][cbfDepthC]
cbf_cr[xC][yC + (1 << log2TrafoSizeC)][cbfDepthC]))
if(cbfLuma cbfChroma) {
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {
cu_qp_delta_abs ae(v)
if(cu_qp_delta_abs)
cu_qp_delta_sign_flag ae(v)
}
if(cu_chroma_qp_offset_enabled_flag && cbfChroma &&
!cu_transquant_bypass_flag && !IsCuChromaQpOffsetCoded) {
cu_chroma_qp_offset_flag ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)
cu_chroma_qp_offset_idx ae(v)
}
if(cbfLuma)
residual_coding(x0, y0, log2TrafoSize, 0)
if(log2TrafoSize > 2 ChromaArrayType == 3) {
if(cross_component_prediction_enabled_flag && cbfLuma &&
(CuPredMode[x0][y0] == MODE_INTER
intra_chroma_pred_mode[x0][y0] == 4))
cross_comp_pred(x0, y0, 0)
for(tIdx = 0; tIdx < (ChromaArrayType == 2 ? 2 : 1); tIdx++)
if(cbf_cb[x0][y0 + (tIdx << log2TrafoSizeC)][trafoDepth])
residual_coding(x0, y0 + (tIdx << log2TrafoSizeC), log2TrafoSizeC, 1)
if(cross_component_prediction_enabled_flag && cbfLuma &&
(CuPredMode[x0][y0] == MODE_INTER
intra_chroma_pred_mode[x0][y0] == 4))
cross_comp_pred(x0, y0, 1)

[0185]

```

for( tIdx = 0; tIdx < ( ChromaArrayType == 2 ? 2 : 1 ); tIdx++ )
    if( cbf_cr[ x0 ][ y0 + ( tIdx << log2TrafoSizeC ) ][ trafoDepth ] )
        residual_coding( x0, y0 + ( tIdx << log2TrafoSizeC ), log2TrafoSizeC, 2 )
    } else if( blkIdx == 3 ) {
        for( tIdx = 0; tIdx < ( ChromaArrayType == 2 ? 2 : 1 ); tIdx++ )
            if( cbf_cb[ xBase ][ yBase + ( tIdx << log2TrafoSizeC ) ][ trafoDepth - 1 ] )
                residual_coding( xBase, yBase + ( tIdx << log2TrafoSizeC ), log2TrafoSize, 1 )
        for( tIdx = 0; tIdx < ( ChromaArrayType == 2 ? 2 : 1 ); tIdx++ )
            if( cbf_cr[ xBase ][ yBase + ( tIdx << log2TrafoSizeC ) ][ trafoDepth - 1 ] )
                residual_coding( xBase, yBase + ( tIdx << log2TrafoSizeC ), log2TrafoSize, 2 )
    }
}
}

```

[0186] 章节7.3.8.13

[0187]

描述符	residual_coding(x0, y0, log2TrafoSize, cIdx) {
	if(transform_skip_enabled_flag && !cu_transquant_bypass_flag && (log2TrafoSize <= Log2MaxTransformSkipSize))
ac(v)	transform_skip_flag[x0][y0][cIdx]
	if(CuPredMode[x0][y0] == MODE_INTER && explicit_rdpcm_enabled_flag && (transform_skip_flag[x0][y0][cIdx] cu_transquant_bypass_flag)) {
ae(v)	explicit_rdpcm_flag[x0][y0][cIdx]
	if(explicit_rdpcm_flag[x0][y0][cIdx])
ae(v)	explicit_rdpcm_dir_flag[x0][y0][cIdx]
	}
ae(v)	last_sig_coeff_x_prefix
ae(v)	last_sig_coeff_y_prefix
	if(last_sig_coeff_x_prefix > 3)
ae(v)	last_sig_coeff_x_suffix
	if(last_sig_coeff_y_prefix > 3)
ae(v)	last_sig_coeff_y_suffix
	lastScanPos = 16
	lastSubBlock = (1 << (log2TrafoSize - 2)) * (1 << (log2TrafoSize - 2)) - 1
	escapeDataPresent = 0
	do {
	if(lastScanPos == 0) {
	lastScanPos = 16
	lastSubBlock--
	}
	lastScanPos--
	xS = ScanOrder[log2TrafoSize - 2][scanIdx][lastSubBlock][0]
	yS = ScanOrder[log2TrafoSize - 2][scanIdx][lastSubBlock][1]
	xC = (xS << 2) + ScanOrder[2][scanIdx][lastScanPos][0]
	yC = (yS << 2) + ScanOrder[2][scanIdx][lastScanPos][1]
	} while((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))
	for(i = lastSubBlock; i >= 0; i--) {

[0188]

<pre> xS = ScanOrder[log2TrafoSize - 2][scanIdx][i][0] yS = ScanOrder[log2TrafoSize - 2][scanIdx][i][1] inferSbDcSigCoeffFlag = 0 if((i < lastSubBlock) && (i > 0)) { coded_sub_block_flag[xS][yS] ae(v) inferSbDcSigCoeffFlag = 1 } for(n = (i == lastSubBlock) ? lastScanPos - 1 : 15; n >= 0; n--) { xC = (xS << 2) + ScanOrder[2][scanIdx][n][0] yC = (yS << 2) + ScanOrder[2][scanIdx][n][1] if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag)) { sig_coeff_flag[xC][yC] ae(v) if(sig_coeff_flag[xC][yC]) inferSbDcSigCoeffFlag = 0 } } firstSigScanPos = 16 lastSigScanPos = -1 numGreater1Flag = 0 lastGreater1ScanPos = -1 for(n = 15; n >= 0; n--) { xC = (xS << 2) + ScanOrder[2][scanIdx][n][0] yC = (yS << 2) + ScanOrder[2][scanIdx][n][1] if(sig_coeff_flag[xC][yC]) { if(numGreater1Flag < 8) { coeff_abs_level_greater1_flag[n] ae(v) numGreater1Flag++ if(coeff_abs_level_greater1_flag[n] && lastGreater1ScanPos == -1) lastGreater1ScanPos = n else if(coeff_abs_level_greater1_flag[n]) escapeDataPresent = 1 } else escapeDataPresent = 1 if(lastSigScanPos == -1) lastSigScanPos = n firstSigScanPos = n } } if(cu_transquant_bypass_flag (CuPredMode[x0][y0] == MODE_INTRA && implicit_rdpcm_enabled_flag && transform_skip_flag[x0][y0][cIdx] && (predModeIntra == 10 predModeIntra == 26)) explicit_rdpcm_flag[x0][y0][cIdx]) signHidden = 0 else signHidden = (lastSigScanPos - firstSigScanPos > 3) if(lastGreater1ScanPos != -1) { coeff_abs_level_greater2_flag[lastGreater1ScanPos] ae(v) if(coeff_abs_level_greater2_flag[lastGreater1ScanPos]) escapeDataPresent = 1 } </pre>	
--	--

[0189]

```

for( n = 15; n  >=  0; n-- ) {
    XC = ( xS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
    yC = ( yS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
    if( sig_coeff_flag[ XC ][ yC ]  &&
        ( !sign_data_hiding_enabled_flag || !signHidden || ( n != firstSigScanPos ) ) )
        coeff_sign_flag[ n ]
    }
    numSigCoeff = 0
    sumAbsLevel = 0
    for( n = 15; n  >=  0; n-- ) {
        XC = ( xS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]
        yC = ( yS  <<  2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]
        if( sig_coeff_flag[ XC ][ yC ] ) {
            baseLevel = 1 + coeff_abs_level_greater1_flag[ n ] +
                        coeff_abs_level_greater2_flag[ n ]
            if( baseLevel == ( ( numSigCoeff < 8 ) ?
                ( ( n == lastGreater1ScanPos ) ? 3 : 2 ) : 1 ) )
                coeff_abs_level_remaining[ n ]
            TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ XC ][ yC ] =
                ( coeff_abs_level_remaining[ n ] + baseLevel ) * ( 1 - 2 *
            coeff_sign_flag[ n ] )
            if( sign_data_hiding_enabled_flag  &&  signHidden ) {
                sumAbsLevel += ( coeff_abs_level_remaining[ n ] + baseLevel )
                if( ( n == firstSigScanPos )  &&  ( ( sumAbsLevel % 2 )  ==  1 ) )
                    TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ XC ][ yC ] =
                        -TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ XC ][ yC ]
            }
            numSigCoeff++
        }
    }
}
}

```

[0190] 为了实施本文中所描述的技术,视频译码器可利用各种语法元素。等于1的语法元素transform_skip_rotation_enabled_flag指定针对使用变换跳过操作译码的帧内 4×4 块将旋转应用于残余数据块。等于0的transform_skip_rotation_enabled_flag指定不应使用此旋转。当不存在时,推断transform_skip_rotation_enabled_flag的值等于0。

[0191] 等于1的语法元素transform_skip_context_enabled_flag指定针对具有跳过变换的变换块将特定上下文用于sig_coeff_flag的解析。等于0的语法元素transform_skip_context_enabled_flag指定变换跳过存在与否或用于变换块的变换旁路不用于关于此旗标的上下文选择。当不存在时,推断transform skip context enabled flag的值等于0。

[0192] 等于1的语法元素 `implicit_rdpcm_enabled_flag` 指定使用变换旁路的块的残余修改过程可用于CVS中的帧内块。等于0的语法元素 `implicit_rdpcm_enabled_flag` 指定残余修改过程不用于CVS中的帧内块。当不存在时,推断 `implicit_rdpcm_enabled_flag` 的值等于0。

[0193] 等于1的语法元素explicit_rdpcm_enabled_flag指定使用变换旁路的块的残余修改过程可用于CVS中的帧间块。等于0的语法元素explicit_rdpcm_enabled_flag指定残余修改过程不用于CVS中的帧间块。当不存在时，推断explicit_rdpcm_enabled_flag的值

等于0。

[0194] 等于1的语法元素extended_precision_processing_flag指定将扩展动态范围用于系数解析及反变换处理。等于0的语法元素extended_precision_processing_flag指定不使用扩展动态范围。当不存在时,推断extended_precision_processing_flag的值等于0。

[0195] 如下导出变量CoeffMinY、CoeffMinC、CoeffMaxY及CoeffMaxC:

[0196] $\text{CoeffMinY} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15))$ (7-27)

[0197] $\text{CoeffMinC} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15))$ (7-28)

[0198] $\text{CoeffMaxY} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15)) - 1$ (7-29)

[0199] $\text{CoeffMaxC} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15)) - 1$ (7-30)

[0200] 等于1的语法元素intra_smoothing_disabled_flag指定相邻样本的滤波过程无条件停用于帧内预测。等于0的语法元素intra_smoothing_disabled_flag指定不停用相邻样本的滤波过程。当不存在时,推断intra_smoothing_disabled_flag的值等于0。

[0201] 等于1的语法元素high_precision_offsets_enabled_flag指定使用位深度相关的精度来传信经加权预测偏移值。等于0的语法元素high_precision_offsets_enabled_flag指定用相当于八位处理的精度来传信经加权预测偏移值。

[0202] 如下导出变量WpOffsetBdShiftY、WpOffsetBdShiftC、WpOffsetHalfRangeY及WpOffsetHalfRangeC:

[0203] $\text{WpOffsetBdShiftY} = \text{high_precision_offsets_enabled_flag} ? 0 : (\text{BitDepthY} - 8)$ (7-31)

[0204] $\text{WpOffsetBdShiftC} = \text{high_precision_offsets_enabled_flag} ? 0 : (\text{BitDepthC} - 8)$ (7-32)

[0205] $\text{WpOffsetHalfRangeY} = 1 \ll (\text{high_precision_offsets_enabled_flag} ? (\text{BitDepthY} - 1) : 7)$ (7-33)

[0206] $\text{WpOffsetHalfRangeC} = 1 \ll (\text{high_precision_offsets_enabled_flag} ? (\text{BitDepthC} - 1) : 7)$ (7-34)

[0207] 等于1的语法元素persistent_rice_adaptation_enabled_flag指定用于coeff_abs_level_remaining的二进制化的莱斯参数导出使用从先前子块累计的模式相关统计数据在每一子块的开始处初始化。等于0的语法元素persistent_rice_adaptation_enabled_flag指定无先前子块状态用于莱斯参数导出。当不存在时,推断persistent_rice_adaptation_enabled_flag的值等于0。

[0208] 等于1的语法元素cabac_bypass_alignment_enabled_flag指定在语法元素coeff_sign_flag[]及coeff_abs_level_remaining[]的旁路解码之前使用CABAC对准过程。等于0的语法元素cabac_bypass_alignment_enabled_flag指定在旁路解码之前不使用CABAC对准过程。当不存在时,推断cabac_bypass_alignment_enabled_flag的值等于0。

[0209] 等于1的语法元素curr_pic_as_ref_enabled_flag指定指代SPS的特定图片可包含于特定图片本身的参考图片列表中。等于0的语法元素curr_pic_as_ref_enabled_flag指定指代SPS的特定图片从不包含于特定图片本身的任何参考图片列表中。当不存在时,推断curr_pic_as_ref_enabled_flag的值等于0。当curr_pic_as_ref_enabled_flag等于1时,CVS中可能不存在指代SPS的I切片。

[0210] 等于1的语法元素palette_mode_enabled_flag指定调色板模式可用于帧内块。等于0的语法元素palette_mode_enabled_flag指定不应用调色板模式。当不存在时,推断palette_mode_enabled_flag的值等于0。

[0211] 等于1的语法元素residual_adaptive_colour_transform_enabled_flag指定在解码过程中自适应色彩变换可应用于残余。等于0的语法元素residual_adaptive_colour_transform_enabled_flag指定不将自适应色彩变换应用于残余。当不存在时,推断residual_adaptive_colour_transform_enabled_flag的值等于0。

[0212] constrained_intra_pred_flag可经定义使得等于0的constrained_intra_pred_flag指定帧内预测允许使用相邻译码块的残余数据及经解码样本,所述相邻译码块使用或未使用不是当前图片的参考图片经译码。等于1的constrained_intra_pred_flag指定受限帧内预测,在此情况下一般帧内预测过程仅使用来自在不使用并非当前图片的参考图片的情况下经译码的相邻译码块的残余数据及经解码样本。

[0213] 根据本文中所描述的技术,表可用于定义切片类型。下文展示这种表的实例。

[0214]

slice_type	slice_type的名称
0	B (B切片)
1	P (P切片)
2	I (I切片)

[0215] 当nal_unit_type具有包含性地介于BLA_W_LP到RSV_IRAP_VCL23的范围内的值时,即图片为IRAP图片,且curr_pic_as_ref_enabled_flag等于0,slice_type将等于2。当sps_max_dec_pic_buffering_minus1[TemporalId]等于0时,slice_type将等于2。

[0216] collocated_ref_idx指定用于时间运动向量预测的并置图片的参考索引。当slice_type等于P时或当slice_type等于B且collocated_from_10_flag等于1时,collocated_ref_idx指代列表0中的图片,且collocated_ref_idx的值将包含性地介于0到num_ref_idx_10_active_minus1的范围内。当slice_type等于B且collocated_from_10_flag等于0时,collocated_ref_idx指代列表1中的图片,且collocated_ref_idx的值将包含性地介于0到num_ref_idx_11_active_minus1的范围内。位流一致性可要求:由collocated_ref_idx指代的图片将对于经译码图片的所有切片相同且不应为当前图片本身。

[0217] 等于1的语法元素cu_skip_flag[x0][y0]指定对于当前译码单元,当解码P或B切片时,除合并候选者索引merge_idx[x0][y0]外,在cu_skip_flag[x0][y0]之后不再解析语法元素。等于0的语法元素cu_skip_flag[x0][y0]指定不跳过译码单元。阵列索引x0,y0指定所考虑译码块的左上方明度样本相对于图片的左上方明度样本的位置(x0,y0)。当cu_skip_flag[x0][y0]不存在时,推断其等于0。

[0218] 等于0的语法元素pred_mode_flag指定当前译码单元是以帧间预测模式译码。等于1的语法元素pred_mode_flag指定当前译码单元是以帧内预测模式译码。针对 $x=x_0..x_0+nCbS-1$ 及 $y=y_0..y_0+nCbS-1$ 如下导出变量CuPredMode[x][y]:如果pred_mode_flag等于0,那么将CuPredMode[x][y]设定成等于MODE_INTER。否则(pred_mode_flag等于1),将CuPredMode[x][y]设定成等于MODE_INTRA。

[0219] 语法元素part_mode指定当前译码单元的分割模式。part_mode的语义取决于CuPredMode[x0][y0]。从part_mode的值导出变量PartMode及IntraSplitFlag。如下限制part_mode的值:如果CuPredMode[x0][y0]等于MODE_INTRA,那么part_mode将等于0或1。否则(CuPredMode[x0][y0]等于MODE_INTER),以下适用:如果log2CbSize大于MinCbLog2SizeY且amp_enabled_flag等于1,那么part_mode将包含性地介于0到2的范围内或包含性地介于4到7的范围内。否则,如果log2CbSize大于MinCbLog2SizeY且amp_enabled_flag等于0,或log2CbSize等于3且curr_pic_as_ref_enabled_flag等于0,那么part_mode将包含性地介于0到2的范围内。否则(log2CbSize大于3且小于或等于MinCbLog2SizeY,或curr_pic_as_ref_enabled_flag等于1),part_mode的值将包含性地介于0到3的范围内。

[0220] 当part_mode不存在时,如下导出变量PartMode及IntraSplitFlag:将PartMode设定成等于PART_2N×2N。将IntraSplitFlag设定成等于0。当CuPredMode[x0][y0]等于MODE_INTER、PartMode等于PART_NxN且log2CbSize等于3时,位流一致性要求:当前CU中的所有块的参考索引将仅指代当前图片。

[0221] 语法元素ref_idx_11[x0][y0]与ref_idx_10具有相同语义,其中10及列表0分别由11及列表1替代。如下导出变量codeMvdFlag:

[0222] codeMvdFlag = (DiffPicOrderCnt(RefPicList0[ref_idx_10], currPic) != 0) ? 1 : 0

[0223] 语法元素mvd_sign_flag[compIdx]如下指定运动向量分量差的正负号:如果mvd_sign_flag[compIdx]等于0,那么对应运动向量分量差具有正值。否则(mvd_sign_flag[compIdx]等于1),对应运动向量分量差具有负值。当mvd_sign_flag[compIdx]不存在时,推断其等于0。如下导出compIdx=0..1的运动向量差1Mvd[compIdx]:

[0224] $1Mvd[compIdx] = abs_mvd_greater0_flag[compIdx] * (abs_mvd_minus2[compIdx] + 2) * (1 - 2 * mvd_sign_flag[compIdx])$ (7-70)

[0225] 变量MvdLX[x0][y0][compIdx] (其中X为0或1) 指定所使用的列表X向量分量与其预测之间的差。MvdLX[x0][y0][compIdx]的值将包含性地介于-215到215-1的范围内。阵列索引x0、y0指定所考虑预测块的左上方明度样本相对于图片的左上方明度样本的位置(x0, y0)。给水平运动向量分量差指派compIdx=0,且给垂直运动向量分量指派compIdx=1。如果refList等于0,那么针对compIdx=0..1将MvdL0[x0][y0][compIdx]设定成等于1Mvd[compIdx]。否则(refList等于1),针对compIdx=0..1将MvdL1[x0][y0][compIdx]设定成等于1Mvd[compIdx]。

[0226] 以下适用于NumPicTotalCurr的值:当当前图片为BLA或CRA图片时,NumPicTotalCurr的值将等于curr_pic_ref_enabled_flag。

[0227] 位流一致性可要求:当nal_unit_type具有包含性地介于BLA_W_LP到RSV_IRAP_

VCL23的范围内的值时,即图片为IRAP图片,且curr_pic_as_ref_enabled_flag等于1,RefPicList0将含有仅指代当前图片的一或多个条目,并且,当slice_type等于1时,RefPicList1将含有仅指代当前图片的一或多个条目。

[0228] 语法元素CuPredMode[xNbY][yNbY]可能不等于MODE_INTRA,在相邻明度位置(xNbY,yNbY)处的块的参考图片中的至少一者不是当前图片,且constrained_intra_pred_flag等于1。

[0229] 在变量predFlagLX、mvLX及refIdxLX中,在PRED_LX中,及在语法元素ref_idx_1X及MvdLX中,对于X由0或1替代,以下适用:

[0230] 1.如下导出变量refIdxLX及predFlagLX:

[0231] -如果inter_pred_idc[xPb][yPb]等于PRED_LX或PRED_BI,那么

[0232] refIdxLX=ref_idx_1X[xPb][yPb]

[0233] predFlagLX=1

[0234] -否则,如下指定变量refIdxLX及predFlagLX:

[0235] refIdxLX=-1

[0236] predFlagLX=0

[0237] 2.如下导出变量mvLX:

[0238] mvLX[0]=MvdLX[xPb][yPb][0]

[0239] mvLX[1]=MvdLX[xPb][yPb][1]

[0240] 3.当predFlagLX等于1时,子条款8.5.3.2.6中的明度运动向量预测的导出过程经调用且明度译码块位置(xCb,yCb)、译码块大小nCbS、明度预测块位置(xPb,yPb)、变量nPbW、nPbH、refIdxLX及分割索引partIdx作为输入,且输出为mvLX。

[0241] 4.当predFlagLX等于1且参考图片不是当前图片时,如下导出明度运动向量mvLX:

[0242] uLX[0]=(mvLX[0]+mvLX[0]+2¹⁶)%2¹⁶

[0243] mvLX[0]=(uLX[0]>=2¹⁵)?(uLX[0]-2¹⁶):uLX[0]

[0244] uLX[1]=(mvLX[1]+mvLX[1]+2¹⁶)%2¹⁶

[0245] mvLX[1]=(uLX[1]>=2¹⁵)?(uLX[1]-2¹⁶):uLX[1]

[0246] 注释-如上文所指定的mvLX[0]及mvLX[1]的所得值将始终包含性地介于-2¹⁵到2¹⁵-1的范围内。

[0247] 5.当predFlagLX等于1且参考图片是当前图片时,如下导出明度运动向量mvLX:

[0248] mvLX[0]=(mvLX[0]>>2+mvLX[0])<<2

[0249] mvLX[1]=(mvLX[0]>>2+mvLX[0])<<2

[0250] 位流一致性要求:当参考图片是当前图片时,明度运动向量mvLX将遵从以下约束:

[0251] (a)当用于z扫描次序块可用性的导出过程被调用且(xCurr,yCurr)被设定成等于(xCb,yCb)并且相邻明度位置(xNbY,yNbY)被设定成等于(xPb+(mvLX[0]>>2),(yPb+mvLX[1]>>2))作为输入时,输出将等于TRUE。

[0252] (b)当用于z扫描次序块可用性的导出过程被调用且(xCurr,yCurr)被设定成等于(xCb,yCb)并且相邻明度位置(xNbY,yNbY)被设定成等于(xPb+(mvLX[0]>>2)+nPbW-1,yPb+(mvLX[1]>>2)+nPbH-1)作为输入时,输出将等于TRUE。

[0253] (c)以下条件中的一者或两者将为真:

[0254] $-(\text{mvLX}[0] \gg 2) + \text{nPbW} + \text{xB1}$ 的值小于或等于0。

[0255] $-(\text{mvLX}[1] \gg 2) + \text{nPbH} + \text{yB1}$ 的值小于或等于0。

[0256] 当 ChromaArrayType 不等于0且 predFlagLX (其中X为0或1) 等于1时, 子条款 8.5.3.2.10 中的色度运动向量的导出过程被调用且明度译码块位置 (xCb, yCb)、明度预测块位置 (xPb, yPb) 及明度运动向量 mvLX 作为输入, 且输出为 mvCLX。

[0257] 如下导出变量 mvLXCol 及 availableFlagLXCol: 如果 slice_temporal_mvp_enabled_flag 等于0, 那么将 mvLXCol 的两个分量都设定成等于0且将 availableFlagLXCol 设定成等于0。否则, 如果参考图片是当前图片且 constrained_intra_pred_flag 等于1, 那么将 mvLXCol 的两个分量设定成等于0且将 availableFlagLXCol 设定成等于0。

[0258] 其它语法表包含以下。通过这些表更新的HEVC标准的相关部分将提供于每一表的标头中。

[0259] 章节9.3.2.2

[0260]

语法结构	语法元素	initType		
		0	1	2
sao()	sao_merge_left_flag	0	1	2
	sao_merge_up_flag	0	1	2
coding_quadtree()	sao_type_idx_luma	0	1	2
	sao_type_idx_chroma	0	1	2
coding_unit()	split_cu_flag[][]	0..2	3..5	6..8
	cu_transquant_bypass_flag	0	1	2
	cu_skip_flag		0..2	3..5
	palette_mode_flag[][]	0	1	2
	pred_mode_flag		0	1
	part_mode	0	1..4	5..8
	prev_intra_luma_pred_flag[][]	0	1	2
	intra_chroma_pred_mode[][]	0	1	2
	rqt_root_cbf		0	1
prediction_unit()	cu_residual_act_flag	0	1	2
	merge_flag[][]		0	1

[0261]

	merge_idx[][]		0	1
	inter_pred_idc[][]		0..4	5..9
	ref_idx_10[][], ref_idx_11[][]		0..1	2..3
	mvp_10_flag[][], mvp_11_flag[][]		0	1
transform_tree()	split_transform_flag[][][]	0..2	3..5	6..8
	cbf_luma[][][]	0..1	2..3	4..5
	cbf_cb[][][], cbf_cr[][][]	0..3 12	4..7 13	8..11 14
mvd_coding()	abs_mvd_greater0_flag[]		0	2
	abs_mvd_greater1_flag[]		1	3
transform_unit()	cu_qp_delta_abs	0..1	2..3	4..5
	cu_chroma_qp_offset_flag	0	1	2
	cu_chroma_qp_offset_idx	0	1	2
cross_comp_pred()	log2_res_scale_abs_plus1[]	0..7	8..15	16..23
	res_scale_sign_flag[]	0..1	2..3	4..5
residual_coding()	transform_skip_flag[][][0]	0	1	2
	transform_skip_flag[][][1]	3	4	5
	transform_skip_flag[][][2]			
	explicit_rdpcm_flag[][][0]		0	1
	explicit_rdpcm_flag[][][1]		2	3
	explicit_rdpcm_flag[][][2]			
	explicit_rdpcm_dir_flag[][][0]		0	1
	explicit_rdpcm_dir_flag[][][1]		2	3
	explicit_rdpcm_dir_flag[][][2]			
	last_sig_coeff_x_prefix	0..17	18..35	36..53
	last_sig_coeff_y_prefix	0..17	18..35	36..53
	coded_sub_block_flag[][]	0..3	4..7	8..11
	sig_coeff_flag[][]	0..41 126..127	42..83 128..129	84..125 130..131
	coeff_abs_level_greater1_flag[]	0..23	24..47	48..71
	coeff_abs_level_greater2_flag[]	0..5	6..11	12..17

[0262] 章节9.3.2.2

[0263]	初始化变量	mvp_10_flag 和 mvp_11_flag 的 ctxIdx	
		0	1
	initValue	168	168

[0264] 章节9.3.3.1

[0265]

语法结构	语法元素	二进位化
		输入参数
slice_segment_data()	end_of_slice_segment_flag	cMax = 1
	end_of_subset_one_bit	cMax = 1
sao()	sao_merge_left_flag	cMax = 1
	sao_merge_up_flag	cMax = 1
	sao_type_idx_luma	cMax = 2, cRiceParam = 0
	sao_type_idx_chroma	cMax = 2, cRiceParam = 0
	sao_offset_abs[][][][]	cMax = (1 << (Min(bitDepth, 10) - 5)) - 1, cRiceParam = 0

[0266] 章节9.3.3.1

[0267]

coding_quadtree()	sao_offset_sign[][][][]	cMax = 1
	sao_band_position[][][]	cMax = 31
	sao_eo_class_luma	cMax = 3
	sao_eo_class_chroma	cMax = 3
	split_cu_flag[][]	cMax = 1
	cu_transquant_bypass_flag	cMax = 1
	cu_skip_flag	cMax = 1
	palette_mode_flag	cMax = 1
	pred_mode_flag	cMax = 1
	part_mode	(xCb, yCb) = (x0, y0), log2CbSize
	pcm_flag[][]	cMax = 1
	prev_intra_luma_pred_flag[][]	cMax = 1
	mpm_idx[][]	cMax = 2, cRiceParam = 0
	rem_intra_luma_pred_mode[][]	cMax = 31
coding_unit()	intra_chroma_pred_mode[][]	-
	rqt_root_cbf	cMax = 1
	cu_residual_act_flag	cMax = 1
	previous_palette_entry_flag[]	cMax = 1
	palette_share_flag	cMax = 1
	palette_num_signalled_entries	cMax = 31, cRiceParam = 0
	palette_entries	cMax = cIdx == 0 ? ((1 << BitDepthY) - 1) : ((1 << BitDepthC) - 1)
	palette_transpose_flag	cMax = 1
	palette_escape_val	cIdx, qP
	escape_val_present_flag	cMax = 1
palette_coding()	palette_run_type_flag	cMax = 1
	palette_index	cMax = adjustedIndexMax
	palette_run	-
	palette_all_zeros_in_group	cMax = 1
	palette_last_group	cMax = 1
	merge_flag[][]	cMax = 1
	merge_idx[][]	cMax = MaxNumMergeCand - 1, cRiceParam = 0
	inter_pred_idc[x0][y0]	nPbW, nPbH
	ref_idx_l0[][]	cMax = num_ref_idx_l0_active_minus1, cRiceParam = 0
	mvp_l0_flag[][]	cMax = 1
prediction_unit()	ref_idx_l1[][]	cMax = num_ref_idx_l1_active_minus1, cRiceParam = 0
	mvp_l1_flag[][]	cMax = 1
transform_tree()	split_transform_flag[][][]	cMax = 1
	cbf_luma[][][]	cMax = 1
	cbf_cb[][][]	cMax = 1

[0268] 章节9.3.3.1

[0269]

	cbf_cr[][][]	cMax = 1
bvd_coding()	abs_bvd_greater0_flag[]	cMax = 1
	abs_bvd_minus1[]	-
	bvd_sign_flag[]	cMax = 1
	abs_mvd_greater0_flag[]	cMax = 1
mvd_coding()	abs_mvd_greater1_flag[]	cMax = 1
	abs_mvd_minus2[]	-
	mvd_sign_flag[]	cMax = 1
	cu_qp_delta_abs	-
transform_unit()	cu_qp_delta_sign_flag	cMax = 1
	cu_chroma_qp_offset_flag	cMax = 1
	cu_chroma_qp_offset_idx	cMax = chroma_qp_offset_list_len_minus1, cRiceParam = 0
	log2_res_scale_abs_plus1	cMax = 4, cRiceParam = 0
cross_comp_pred()	res_scale_sign_flag	cMax = 1
	transform_skip_flag[][][]	cMax = 1
	explicit_rdpcm_flag[][][]	cMax = 1
	explicit_rdpcm_dir_flag[][][]	cMax = 1
	last_sig_coeff_x_prefix	cMax = (log2TrafoSize << 1) - 1, cRiceParam = 0
	last_sig_coeff_y_prefix	cMax = (log2TrafoSize << 1) - 1, cRiceParam = 0
	last_sig_coeff_x_suffix	cMax = (1 << ((last_sig_coeff_x_prefix >> 1) - 1) - 1)
	last_sig_coeff_y_suffix	cMax = (1 << ((last_sig_coeff_y_prefix >> 1) - 1) - 1)
	coded_sub_block_flag[][]	cMax = 1
	sig_coeff_flag[][]	cMax = 1
	coeff_abs_level_greater1_flag[]	cMax = 1
	coeff_abs_level_greater2_flag[]	cMax = 1
	coeff_abs_level_remaining[]	当前子块扫描索引 i, baseLevel
	coeff_sign_flag[]	cMax = 1

[0270] 章节9.3.3.7

[0271]

CuPredMode [xCb][yCb]	part_mode	PartMode	二进制数串			
			log2CbSize > MinCbLog2SizeY		log2CbSize == MinCbLog2SizeY	
			!amp_enabled_flag	amp_enabled_flag	log2CbSize == 3 && !curr_pic_as_ ref_enabled_flag	log2CbSize > 3 curr_pic_as_ref_ enabled_flag
MODE_INTRA	0	PART_2Nx2N	-	-	1	1
	1	PART_NxN	-	-	0	0
MODE_INTER	0	PART_2Nx2N	1	1	1	1
	1	PART_2NxN	01	011	01	01
	2	PART_Nx2N	00	001	00	001
	3	PART_NxN	-	-	-	000
	4	PART_2NxN_U	-	0100	-	-
	5	PART_2NxN_D	-	0101	-	-
	6	PART_nLx2N	-	0000	-	-

[0272]

7	PART_nRx2N	-	0001	-	-
---	------------	---	------	---	---

[0273] 章节9.3.4.2.1

[0274] 将ctxInc指派给具有上下文经译码二进制数的语法元素

[0275]

语法元素	binIdx					
	0	1	2	3	4	≥ 5
end_of_slice_segment_flag	终止	na	na	na	na	na
end_of_subset_one_bit	终止	na	na	na	na	na
sao_merge_left_flag	0	na	na	na	na	na
sao_merge_up_flag	0	na	na	na	na	na
sao_type_idx_luma	0	旁路	na	na	na	na
sao_type_idx_chroma	0	旁路	na	na	na	na
sao_offset_abs[][][][]	旁路	旁路	旁路	旁路	旁路	na
sao_offset_sign[][][][]	旁路	na	na	na	na	na
sao_band_position[][][]	旁路	旁路	旁路	旁路	旁路	旁路
sao_eo_class_luma	旁路	旁路	旁路	na	na	na
sao_eo_class_chroma	旁路	旁路	旁路	na	na	na
split_cu_flag[][]	0,1,2 (子条款 9.3.4.2.2)	na	na	na	na	na
cu_transquant_bypass_flag	0	na	na	na	na	na
cu_skip_flag	0,1,2 (子条款 9.3.4.2.2)	na	na	na	na	na
pred_mode_flag	0	na	na	na	na	na
palette_mode_flag	0	na	na	na	Na	Na
palette_share_flag	0	na	na	na	Na	na
previous_palette_entry_flag	旁路	na	na	na	na	Na
palette_num_signalled_entries	旁路	旁路	旁路	旁路	旁路	旁路
palette_entries	旁路	旁路	旁路	旁路	旁路	旁路
palette_transpose_flag	0, 1 (子条款 9.3.4.2.2)	na	na	na	na	Na
palette_escape_val	旁路	旁路	旁路	旁路	旁路	旁路
escape_val_present_flag	旁路	na	na	na	na	na
palette_run_type_flag	0, 1 (子条款 9.3.4.2.2)	na	Na	na	Na	Na
palette_index	旁路	旁路	旁路	旁路	旁路	旁路
palette_run	0	1	2	旁路	旁路	旁路
palette_all_zeros_in_group	旁路	na	na	na	na	na
palette_last_group	旁路	na	na	na	na	na
part_mode						
log2CbSize == MinCbLog2SizeY	0	1	2	旁路	na	na
part_mode						
log2CbSize > MinCbLog2SizeY	0	1	3	旁路	na	na
pcm_flag[][]	终止	na	na	na	na	na
prev_intra_luma_pred_flag[][]	0	na	na	na	na	na

[0276] 将ctxInc指派给具有上下文经译码二进制数的语法元素

[0277]

语法元素	binIdx					
	0	1	2	3	4	≥ 5
mpm_idx[][]	旁路	旁路	na	na	na	na
rem_intra_luma_pred_mode[][]	旁路	旁路	旁路	旁路	旁路	na
intra_chroma_pred_mode[][]	0	旁路	旁路	na	na	na
rqt_root_cbf	0	na	na	na	na	na
cu_residual_act_flag	0	na	na	na	na	na
merge_flag[][]	0	na	na	na	na	na
merge_idx[][]	0	旁路	旁路	旁路	na	na
inter_pred_idc[x0][y0]	$(nPbW + nPbH) != 12 ? CtDepth[x0][y0] : 4$	4	na	na	na	na
ref_idx_10[][]	0	1	旁路	旁路	旁路	旁路
ref_idx_11[][]	0	1	旁路	旁路	旁路	旁路
mvp_10_flag[][]	0	na	na	na	na	na
mvp_11_flag[][]	0	na	na	na	na	na
split_transform_flag[][][]	$5 - \log2\text{TrafoSize}$	na	na	na	na	na
cbf_cb[][][]	trafoDepth	na	na	na	na	na
cbf_cr[][][]	trafoDepth	na	na	na	na	na
cbf_luma[][][]	trafoDepth == 0 ? 1 : 0	na	na	na	na	na
abs_mvd_greater0_flag[]	0	na	na	na	na	na
abs_mvd_greater1_flag[]	0	na	na	na	na	na
abs_mvd_minus2[]	旁路	旁路	旁路	旁路	旁路	旁路
mvd_sign_flag[]	旁路	na	na	na	na	na
cu_qp_delta_abs	0	1	1	1	1	旁路
cu_qp_delta_sign_flag	旁路	na	na	na	na	na
cu_chroma_qp_offset_flag	0	na	na	na	na	na
cu_chroma_qp_offset_idx	0	0	0	0	0	na
log2_res_scale_abs_plus1[c]	$4 * c + 0$	$4 * c + 1$	$4 * c + 2$	$4 * c + 3$	na	na
res_scale_sign_flag[c]	c	na	na	na	na	na
transform_skip_flag[][][]	0	na	na	na	na	na
explicit_rdpcm_flag[][][]	0	na	na	na	na	na
explicit_rdpcm_dir_flag[][][]	0	na	na	na	na	na
last_sig_coeff_x_prefix	0.17 (子条款 9.3.4.2.3)					
last_sig_coeff_y_prefix	0.17 (子条款 9.3.4.2.3)					
last_sig_coeff_x_suffix	旁路	旁路	旁路	旁路	旁路	旁路
last_sig_coeff_y_suffix	旁路	旁路	旁路	旁路	旁路	旁路
coded_sub_block_flag[][]	0.3 (子条款 9.3.4.2.4)	na	na	na	na	na
sig_coeff_flag[][]	0.43 (子条款 9.3.4.2.5)	na	na	na	na	na
coeff_abs_level_greater1_flag[]	0.23 (子条款 9.3.4.2.6)	na	na	na	na	na
coeff_abs_level_greater2_flag[]	0.5 (子条款 9.3.4.2.7)	na	na	na	na	na
coeff_abs_level_remaining[]	旁路	旁路	旁路	旁路	旁路	旁路

[0278] 将ctxInc指派给具有上下文经译码二进制数的语法元素

[0279]

语法元素	binIdx					
	0	1	2	3	4	≥ 5
coeff_sign_flag[]	旁路	na	na	na	na	na

[0280] 图6展示用于导出运动信息合并及AMVP模式的空间相邻运动向量候选者的实例。根据HEVC标准,当使用帧内预测或帧内预测时,运动信息可能可用于视频编码器20或视频解码器30。对于每一块,运动信息集合可为可用的。运动信息集合可含有用于正向和后向预测方向的运动信息。术语正向和后向预测方向的意思是指示与双向预测模式相关联的两个预测方向。“正向”和“后向”这些术语未必具有几何含义。实际上,“正向”和“后向”对应于当前图片的参考图片列表0 (RefPicList0) 和参考图片列表1 (RefPicList1)。当仅一个参考图片列表可用于图片或切片时,仅RefPicList0可用,并且切片的每一块的运动信息始终是正向的。

[0281] 对于每个预测方向,运动信息必须含有参考索引和运动向量。参考索引是参考图片列表中的特定图片的索引。在一些情况下,为简单起见,视频译码器可假设运动具有相关联的参考索引。参考索引用于识别当前参考图片列表 (RefPicList0或RefPicList1) 中的参考图片。运动向量具有水平和垂直分量。当CU经帧间译码时,针对每一PU存在一个运动信息集合。视频译码器使用与PU相关联的帧间预测模式以导出用于所述PU的运动信息集合。

[0282] 在HEVC标准中,对于一个预测单元 (PU) 存在两种帧间预测模式,分别是合并 (包含跳过模式,这是合并模式的一种特殊情况) 和高级运动向量预测 (AMVP) 模式。在AMVP或合并模式中,视频译码器保持多个运动向量 (MV) 预测符的运动向量候选者列表。视频译码器产生运动向量以及有待在通过从MV候选者列表中选择一个候选者而使用合并模式译码PU时使用的参考索引。

[0283] MV候选者列表含有用于合并模式的多达5个候选者和用于AMVP模式的仅两个候选者。合并候选者可含有运动信息集合,例如,对应于两个参考图片列表 (列表0及列表1) 及对应参考索引的运动向量。如果视频译码器通过合并索引识别合并候选者,那么视频译码器使用识别的参考图片预测当前块。视频译码器还确定用于当前块的相关联的运动向量。然而,当使用AMVP译码块时,对于从列表0或列表1起的每一潜在预测方向,视频译码器必须明确地传信每一参考索引以及MVP索引。MVP索引是MV候选者列表的索引,因为AMVP候选者仅含有一个运动向量。在AMVP模式中,视频译码器可以进一步优化预测的运动向量。

[0284] 因此,如上文所描述,视频译码器基于相同的空间和时间相邻块针对AMVP和合并模式两者类似地导出候选者。然而,合并和AMVP模式的不同之处在于,合并候选者对应于完整的运动信息集合,而AMVP候选者仅含有用于特定的预测方向的一个运动向量和一个参考索引。以类似方式从相同空间及时间相邻块导出用于两种模式的候选者。图6说明用于合并及AMVP模式的MV候选者。

[0285] 尽管从块产生候选者的方法对于合并模式和AMVP模式是不同的,但对于特定PU (PU_0),从图6中示出的相邻块导出空间MV候选者。

[0286] 在合并模式中,存在五个可能的空间MV候选者位置: $\{a_0, a_1, b_0, b_1, b_2\}$,如图6中所说明。视频译码器针对每一MV候选者位置确定在所述位置处的MV候选者是否为可用的。作为一实例,如果所述位置处的块尚未解码,候选者可能不是可用的。视频译码器按以下次序

检查MV候选项可用: $\{a_1, b_1, b_0, a_0, b_2\}$ 。

[0287] 在AVMP模式中,视频译码器(例如,视频编码器20或视频解码器30)将相邻块划分成两个群组:由块 a_0 及 a_1 组成的左群组以及由块 b_0 、 b_1 及 b_2 组成的上方相邻群组,如图6中所展示。对于左方相邻群组,视频译码器根据以下次序检查块的可用性: $\{a_0, a_1\}$ 。对于上方相邻群组,视频译码器根据以下次序检查上方相邻块的可用性: $\{b_0, b_1, b_2\}$ 。对于每一群组,指代与由经传信参考索引所指示的图片相同的参考图片的潜在候选者具有有待被视频译码器选择以形成所述群组的最终候选者的最高优先级。有可能的是没有相邻块含有指向相同参考图片的运动向量。在这种情况下,如果视频译码器无法找到指向与经传信索引所指示的图片相同的参考图片的运动向量,那么视频译码器选择第一可用候选者。视频译码器按比例缩放第一可用候选者以形成最终候选者。在这种情况下,按比例缩放候选运动向量能补偿时间距离差异。

[0288] 运动向量在用于色度运动补偿之前针对当前PU/CU的明度分量被导出,所述运动向量基于色度取样格式按比例缩放。

[0289] PU或CU可具有样本的明度块,以及样本的多个对应色度(色彩)块。在一些情况下,可能存在对应于每个色度样本的多个明度样本(被称作色度子取样)。当确定运动向量时,视频译码器首先针对PU或CU导出明度运动向量分量。视频译码器可以按比例缩放明度运动向量以基于色度子取样格式确定色度运动向量。

[0290] 此外,在HEVC中,视频译码器可以将LCU划分成并行运动估计区(MER)。当LCU是MER的一部分时,视频译码器仅允许将属于与当前PU不同的MER的相邻PU包含于合并/跳过MVP列表建构过程中。视频译码器对图片参数集的MER的大小进行译码。指示MER大小的语法元素是“log2_parallel_merge_level1_minus2”。

[0291] 在一些情况下,例如,当MER大小大于 $N \times N$ 样本时,且当最小可能的CU大小被设定成 $2N \times 2N$ 样本时,MER可相对于运动估计对空间上相邻块的可用性具有影响。举例来说,在前述情况下,如果空间上相邻块在与当前PU相同的MER内部,那么视频译码器将空间上相邻块视为不可用的。

[0292] 帧内块复制(BC)已包含于当前屏幕内容译码(SCC)中。在图5中展示帧内BC的实例,其中当前CU/PU是从当前图片/切片的已经解码块预测的。应注意,预测信号经重建而无需环路内滤波(包含解块和样本自适应偏移(SAO))。

[0293] 对于以帧内BC译码的明度分量或色度分量,通过整数块补偿完成块补偿;因此不需要内插。

[0294] 在合并模式中,在验证空间候选者之后,视频译码器去除两种类别的冗余。如果当前PU的候选位置将涉及同一CU内的第一PU,那么视频译码器排除所述位置,因为可通过CU来实现相同合并,而不分裂成若干预测分区。此外,视频译码器排除其中候选者恰好具有相同运动信息的任何冗余条目。在检查空间相邻候选者之后,视频译码器验证时间候选者。对于时间候选者,如果正好在参考图片的并置PU外部的右下方位置可用,那么视频译码器使用所述右下方位置。否则,视频译码器使用中心位置。视频译码器选择并置PU的方式与先前标准的方式类似,但HEVC通过发射索引以指定将哪一参考图片列表用于并置参考图片而允许更多灵活性。

[0295] 与时间候选者的使用有关的一个问题是存储参考图片的运动信息所需的存储器

的量。这是通过将用于存储时间运动候选者的粒度限制为 16×16 明度网格的分辨率来解决的,甚至在参考图片中的对应位置处使用较小PB结构时也是如此。另外,PPS层级旗标允许视频编码器停用时间候选者,这适用于具有易错发射的应用程序。视频编码器通过在切片标头中传信合并候选者(C)来指示C的最大数目。如果所发现的合并候选者(包含时间候选者)的数目大于C,那么仅视频译码器仅保留前C-1个空间候选者及时间候选者。

[0296] 否则,如果所识别的合并候选者数目小于C,那么视频译码器产生额外候选者直到数目等于C为止。这简化了剖析,并使其更稳健,因为剖析经译码数据的能力不取决于合并候选者可用性。

[0297] 对于B切片,视频译码器通过根据参考图片列表0和列表1的预定义次序挑选两个现有候选者来产生额外合并候选者。举例来说,首先产生的候选者使用列表0的第一合并候选者和列表1的第二合并候选者。HEVC将已经建构的合并候选者列表中的按以下次序的两者的总共12个预定义对指定为(0,1)、(1,0)、(0,2)、(2,0)、(1,2)、(2,1)、(0,3)、(3,0)、(1,3)、(3,1)、(2,3)和(3,2)。其中,在去除冗余条目之后,可包含至多五个候选者。当合并候选者的数目仍小于C时,改为使用默认合并候选者(包含默认运动向量和对应参考索引),其中使用与来自零到所述参考图片数目减一的参考索引相关联的零运动向量来填充合并候选者列表中的任何其余条目。

[0298] 在AMVP模式中,HEVC仅允许将数目低得多的候选者用于运动向量预测过程,这是因为视频编码器可发送经译码差来改变运动向量。此外,视频编码器需要执行运动估计,其为编码器中的计算上最昂贵操作中的一者,且通过允许少量候选者来降低复杂性。当相邻PU的参考索引不等于当前PU的参考索引时,视频译码器使用运动向量的经按比例缩放版本。根据当前图片与分别由相邻PU和当前PU的参考索引指示的参考图片之间的时间距离来按比例缩放相邻运动向量。

[0299] 当两个空间候选者具有相同运动向量分量时,排除一个冗余空间候选者。当运动向量预测符的数目不等于二且不明确地停用时间MV预测时,包括时间MV预测候选者。这意味着当两个空间候选者可用时,根本不使用时间候选者。最终,重复地包括为零运动向量的默认运动向量直到运动向量预测候选者的数目等于二为止,这保证运动向量预测符数目为二。因此,对于识别在AMVP模式的情况下使用哪一运动向量预测,仅经译码旗标是必需的。

[0300] 在运动预测期间,视频译码器导出当前PU/CU的明度分量的MV,例如,使用AMVP或合并模式。在视频译码器可将MV用于色度运动补偿之前,视频译码器基于PU/CU的色度取样格式来按比例缩放MV。

[0301] 图7是由相邻块P及Q形成的四个像素长的垂直块边界的说明。当Bs为正时,用以确定是否启用解块滤波器的准则可根据以下方程定义:

$$|P_{2,0}-2P_{1,0}+P_{0,0}|+|P_{2,3}-2P_{1,3}+P_{0,3}|+|Q_{2,0}-2Q_{1,0}+Q_{0,0}|+|Q_{2,3}-2Q_{1,3}+Q_{0,3}|<\beta \quad (1)$$

[0303] 用以确定普通及强解块滤波器的准则是(i=0,3):

$$|P_{2,i}-2P_{1,i}+P_{0,i}|+|Q_{2,i}-2Q_{1,i}+Q_{0,i}|<\beta/8 \quad (2)$$

$$|P_{3,i}-P_{0,i}|+|Q_{3,i}-Q_{0,i}|<\beta/8 \quad (3)$$

$$|P_{0,i}-Q_{0,i}|<2.5T_c \quad (4)$$

[0307] 可以类似方式处理水平块边界。

[0308] 图8是说明实例空间块向量候选者的概念图。将如图4中所展示的四个空间块向量

预测符候选者划分成：包含{a₂, a₁}的左群组及包含{b₂, b₁}的上群组。选择两个空间块向量预测符候选者，其中一者来自根据可用性检查次序{a₂, a₁}的左群组，且另一者来自根据可用性检查次序{b₂, b₁}的上群组。如果一个空间块向量预测符候选者不可用，那么改为使用(-2w, 0)。如果两个空间块向量预测符候选者不可用，那么改为使用(-2*w, 0)及(-w, 0)，其中w为CU宽度。应注意，当当前块变成4×4大小时，b₂及b₁将为相同块且a₂及a₁将为相同块。

[0309] 图9为说明可实施本发明中所描述的技术的实例视频编码器20的框图。视频编码器20可经配置以将视频输出到后处理实体27。后处理实体27既定表示可处理来自视频编码器20的经编码视频数据的视频实体的实例，例如MANE或拼接/编辑装置。在一些情况下，后处理实体27可为网络实体的实例。在一些视频编码系统中，后处理实体27和视频编码器20可为单独装置的各部分，而在其它情况下，关于后处理实体27所描述的功能性可由包括视频编码器20的相同装置执行。在某一实例中，后处理实体27是图1的存储装置17的实例。

[0310] 视频编码器20可在视频切片内执行视频块的帧内译码、帧间译码及IMC译码。帧内译码依赖于空间预测来减少或去除给定视频帧或图片内的视频中的空间冗余。帧间译码依赖于时间预测来减少或去除视频序列的邻近帧或图片内的视频中的时间冗余。帧内模式(I模式)可指代若干基于空间的压缩模式中的任一者。例如单向预测(P模式)或双向预测(B模式)的帧间模式可指若干基于时间的压缩模式中的任一者。如上文所描述的IMC译码模式可从视频数据帧去除空间冗余，但不同于传统帧内模式，IMC译码模式可用于在帧内的较大搜索区域中定位预测性块且指代具有偏移向量的预测性块，而不是依赖于帧内预测译码模式。

[0311] 在图9的实例中，视频编码器20包含视频数据存储器33、分割单元35、预测处理单元41、滤波器单元63、经解码图片缓冲器64、求和器50、变换处理单元52、量化处理单元54和熵编码单元56。预测单元41包含运动估计单元42、运动补偿单元44和帧内预测处理单元46。为了视频块重建，视频编码器20还包含反量化单元58、反变换处理单元60及求和器62。滤波器单元63既定表示一或多个环路滤波器，例如，解块滤波器、自适应环路滤波器(ALF)及样本自适应偏移(SAO)滤波器。尽管在图7中将滤波器单元63展示为环内滤波器，但在其它配置中，可将滤波器单元63实施为环路后滤波器。

[0312] 视频数据存储器33可存储待由视频编码器20的组件编码的视频数据。可(例如)从视频源18获得存储在视频数据存储器33中的视频数据。经解码图片缓冲器64可以是参考图片存储器，其存储用于由视频编码器20(例如)以帧内、帧间或IMC译码模式对视频数据进行编码的参考视频数据。视频数据存储器33和经解码图片缓冲器64可由多种存储器装置中的任一者形成，例如动态随机存取存储器(DRAM)，包含同步DRAM(SDRAM)、磁阻式RAM(MRAM)、电阻式RAM(RRAM)或其它类型的存储器装置。视频数据存储器33和经解码图片缓冲器64可由相同存储器装置或单独的存储器装置提供。在各种实例中，视频数据存储器33可与视频编码器20的其它组件一起位于芯片上，或相对于那些组件在芯片外。

[0313] 如图7中所展示，视频编码器20接收视频数据并将视频数据存储于视频数据存储器33中。分割单元35将所述数据分割成若干视频块。此分割还可包含分割成切片、图块或其它较大单元，以及例如根据LCU和CU的四叉树结构的视频块分割。视频编码器20大体上说明对待编码的视频切片内的视频块进行编码的组件。所述切片可划分成多个视频块(并且可能划分成被称作图块的视频块集合)。预测处理单元41可基于错误结果(例如，译码速率及

失真层级)针对当前视频块选择多个可能的译码模式中的一者,例如多个帧内译码模式中的一者、多个帧间译码模式中的一者,或多个IMC译码模式中的一者。预测处理单元41可将所得帧内、帧间或IMC经译码块提供到求和器50以产生残余块数据且提供到求和器62以重建用作参考图片的经编码块。

[0314] 预测处理单元41内的帧内预测处理单元46可相对于与待译码的当前块在相同帧或切片中的一或多个相邻块执行当前视频块的帧内预测性译码以提供空间压缩。预测处理单元41内的运动估计单元42及运动补偿单元44相对于一或多个参考图片中的一或多个预测性块执行当前视频块的帧间预测性译码以提供时间压缩。预测处理单元41内的运动估计单元42和运动补偿单元44还可相对于在同一图片中的一或多个预测性块执行当前视频块的IBC译码以提供空间压缩。

[0315] 运动估计单元42可经配置以根据视频序列的预定模式为视频切片确定帧间预测模式或IMC模式。预定模式可将序列中的视频切片标明为P切片、B切片或GPB切片。运动估计单元42和运动补偿单元44可高度集成,但出于概念目的而分别加以说明。由运动估计单元42执行的运动估计为产生运动向量的过程,所述过程估计视频块的运动。举例来说,运动向量可指示当前视频帧或图片内的视频块的PU相对于参考图片内的预测性块的位移。在IMC译码的情况下,运动向量,其在IMC中可被称为偏移向量,可指示当前视频帧或图片内的视频块的PU相对于当前视频帧内的预测性块的位移。

[0316] 预测性块为被发现在像素差方面与待译码的视频块的PU密切匹配的块,像素差可通过绝对差总和(SAD)、平方差总和(SSD)或其它差度量值来确定。在一些实例中,视频编码器20可计算存储在经解码图片缓冲器64中的参考图片的子整数像素位置的值。举例来说,视频编码器20可内插四分之一像素位置、八分之一像素位置或参考图片的其它分数像素位置的值。因此,运动估计单元42可相对于整数像素位置及分数像素位置执行运动搜索且输出具有分数像素精度的运动向量。

[0317] 运动估计单元42通过比较PU的位置与参考图片的预测性块的位置来计算用于经帧间译码切片中的视频块的PU的运动向量。参考图片可选自第一参考图片列表(列表0)或第二参考图片列表(列表1),所述列表中的每一者识别存储在经解码图片缓冲器64中的一或多个参考图片。运动估计单元42将计算出的运动向量发送到熵编码单元56及运动补偿单元44。

[0318] 根据本发明的一些技术,当使用IMC模式来译码视频块时,运动估计单元42可确定视频块的明度分量的运动向量或偏移向量,且基于所说明度分量的偏移向量来确定视频块的色度分量的偏移向量。在另一个实例中,当使用IMC模式译码视频块时,运动估计单元42可确定视频块的色度分量的运动向量或偏移向量,且基于色度分量的偏移向量确定视频块的明度分量的偏移向量。因此,视频编码器20可在位流中传信仅一个偏移向量,从所述偏移向量可确定视频块的色度和明度分量两者的偏移向量。

[0319] 由运动补偿单元44执行的运动补偿可以涉及基于通过运动估计(有可能对子像素精度执行内插)确定的运动向量提取或产生预测性块。内插滤波可从已知像素样本产生额外像素样本,从而潜在地增加可用于译码视频块的候选预测性块的数目。在接收当前视频块的PU的运动向量之后,运动补偿单元44可将预测性块定位于运动向量所指向的参考图片列表中的一者中,或在IMC译码的情况下,定位于经译码图片内。视频编码器20通过从正被

译码的当前视频块的像素值减去预测性块的像素值从而形成像素差值来形成残余视频块。像素差值形成用于所述块的残余数据，并且可包含明度和色度差分量两者。求和器50表示执行此减法运算的一或多个组件。运动补偿单元44还可产生与视频块和视频切片相关联的供视频解码器30在对视频切片的视频块进行解码时使用的语法元素。

[0320] 作为由运动估计单元42及运动补偿单元44执行的帧间预测及IMC(如上文所描述)的替代方案,帧内预测处理单元46可帧内预测当前块。具体来说,帧内预测处理单元46可确定用于对当前块编码的帧内预测模式。在一些实例中,帧内预测处理单元46可(例如)在单独的编码遍次期间使用各种帧内预测模式编码当前块,并且帧内预测处理单元46(或在一些实例中为模式选择单元40)可以从经测试模式中选择适当帧内预测模式来使用。举例来说,帧内预测处理单元46可以使用速率失真分析计算用于各种经测试帧内预测模式的速率失真值,并且从所述经测试模式当中选择具有最佳速率失真特性的帧内预测模式。速率失真分析大体上确定经编码块与经编码以产生所述经编码块的原始未编码块之间的失真(或误差)的量,以及用于产生经编码块的位速率(即,位数目)。帧内预测处理单元46可以从用于各种经编码块的失真及速率计算比率,以确定哪种帧内预测模式对于所述块来说展现最佳速率失真值。

[0321] 在任何情况下,在选择了用于块的帧内预测模式之后,帧内预测处理单元46可以将指示用于所述块的所选帧内预测模式的信息提供到熵编码单元56。熵编码单元56可根据本发明的技术编码指示所选帧内预测模式的信息。视频编码器20可将配置数据包含在所发射的位流中,所述配置数据可包含多个帧内预测模式索引表和多个经修改的帧内预测模式索引表(也称为码字映射表);对各种块的编码上下文的定义;以及对待用于所述上下文中的每一者的最可能帧内预测模式、帧内预测模式索引表及经修改的帧内预测模式索引表的指示。

[0322] 在预测处理单元41经由帧间预测、帧内预测抑或IMC产生当前视频块的预测性块之后,视频编码器20通过从当前视频块减去预测性块而形成残余视频块。残余块中的残余视频数据可包含在一或多个TU中并应用于变换处理单元52。变换处理单元52使用例如离散余弦变换(DCT)或概念上类似的变换等变换来将残余视频数据转换成残余变换系数。变换处理单元52可将残余视频数据从像素域转换到变换域,例如,频域。

[0323] 变换处理单元52可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降低位速率。量化过程可减小与系数中的一些或全部相关联的位深度。可通过调整量化参数来修改量化程度。在一些实例中,量化单元54可以随后执行包含经量化变换系数的矩阵的扫描。替代地,熵编码单元56可以执行所述扫描。

[0324] 在量化之后,熵编码单元56对经量化变换系数进行熵编码。举例来说,熵编码单元56可执行上下文自适应可变长度译码(CAVLC)、上下文自适应二进制算术译码(CABAC)、基于语法的上下文自适应二进制算术译码(SBAC)、概率区间分割熵(PIPE)译码或另一熵译码方法或技术。在由熵编码单元56进行熵编码之后,可将经编码位流发射到视频解码器30,或将经编码位流存档以供稍后发射或由视频解码器30检索。熵编码单元56还可对正被译码的当前视频切片的运动向量及其它语法元素进行熵编码。

[0325] 反量化单元58及反变换处理单元60分别应用反量化及反变换以在像素域中重构残余块,以供稍后用作参考图片的参考块。运动补偿单元44可通过将残余块添加到在参考

图片列表中的一者内的参考图片中的一者的预测性块来计算参考块。运动补偿单元44还可将一或多个内插滤波器应用于经重建的残余块以计算子整数像素值以用于运动估计。内插滤波可从已知像素样本产生额外像素样本,从而潜在地增加可用于译码视频块的候选预测性块的数目。求和器62将经重建的残余块添加到由运动补偿单元44产生的运动补偿预测块以产生参考块以用于存储在经解码图片缓冲器64中。参考块可由运动估计单元42和运动补偿单元44用作参考块以对后续视频帧或图片中的块进行帧间预测。

[0326] 根据本发明的技术,如上文所描述,视频编码器20可经配置以使用统一块内复制模式对视频数据的CU进行译码,其中使用统一块内复制译码块包含将当前图片添加到参考图片集,在不译码表示运动向量的精度的语法元素的情况下,如果参考图片集仅包含当前图片或如果用于块的参考图片是当前图片,且基于CU的运动向量对CU进行译码,那么确定CU的运动向量具有整数像素精度。

[0327] 在根据本发明的技术的另一实例中,视频编码器20可经配置以对指示CU的运动向量是具有整数像素精度还是子整数像素精度的语法元素进行译码。当语法元素指示运动向量具有整数像素精度时:视频编码器20可按比例缩放CU的运动向量以产生CU的色度样本的块的经按比例缩放的色度运动向量,如果所述经按比例缩放色度运动向量不具有整数像素精度,那么将所述经按比例缩放色度运动向量上舍入或下舍入为整数像素值,且基于经舍入的色度运动向量对CU进行译码。

[0328] 根据本文中所描述的一或多种技术,视频编码器20的熵编码单元56可经配置以执行本发明的技术。举例来说,视频编码器20的熵编码单元56可经配置以确定用于译码当前视频块的参考图片。视频编码器20的熵编码单元56可进一步确定参考图片的POC值。响应于参考图片的POC值等于包含当前视频块的当前图片的POC值,视频编码器20的编码单元56可设定语法元素的值以指示参考图片列表包含当前图片。在一些实例中,语法元素可位于PPS或SPS中的一者中。相反地,响应于参考图片的POC值不等于包含当前视频块的当前图片的POC值,视频编码器20的编码单元56可设定语法元素的值以指示参考图片列表并不包含当前图片。

[0329] 在一些实例中,当前图片是包括P切片或B切片中的一者的帧内随机存取图片(IRAP)。在此些实例中,IRAP图片可进一步包含指定非空参考图片集的一或多个不同语法元素的集合。在一些实例中,当前图片的当前视频块包含明度分量及色度分量。在此些实例中,明度或色度分量中的仅一者可具有整数像素精度。

[0330] 在一些实例中,参考图片可进一步包含当前图片的经解码未滤波样本。在各种个例中,参考图片可包含进一步包含运动预测信息的参考块。视频编码器20的熵编码单元56可将参考块的运动信息用于译码当前视频块。运动补偿单元44可(例如)编码以合并模式译码的块。如果当前块的图片有资格处于参考图片列表中,那么合并候选者列表中的合并候选者中的一或多者可包含指向与当前块处于同一图片中的参考块的运动向量(例如偏移向量)。

[0331] 在参考图片的POC值不等于当前图片的POC值的一些实例中,指代当前图片的SPS的CVS可包含一或多个I切片。

[0332] 在一些实例中,当前图片是清洁随机存取图片或断链存取图片中的一者。在此些实例中,视频编码器20的熵编码单元56可将指示可用作当前图片的预测参考的多个图片的

第二语法元素的值设定为1。

[0333] 在参考图片的POC值等于当前图片的POC值的一些实例中,当前视频块的MV可具有整数像素精度。此外,视频编码器20的熵编码单元56可在将当前图片的多个运动向量预测符候选者中的每一MV预测符候选者用于候选者导出过程之前或在候选者导出过程完成之后将所述每一运动向量预测符候选者转换为整数像素精度。在参考图片的POC值等于当前图片的POC值的其它实例中,当前图片的每一预测样本包含在当前图片的图片边界内。此外,如果参考图片列表仅包含当前图片,那么视频编码器20的熵编码单元56可停用TMVP。

[0334] 视频解码器30可经配置以执行根据本发明的技术的大体上互逆过程。

[0335] 图10是说明可实施本发明中所描述的技术的实例视频解码器的框图。在图10的实例中,视频解码器30包含视频数据存储器78、熵解码单元80、预测处理单元81、反量化单元86、反变换处理单元88、求和器90、滤波器单元91以及经解码图片缓冲器92。预测处理单元81包含运动补偿单元82及帧内预测处理单元84。在一些实例中,视频解码器30可执行大体上与相对于图9的视频编码器20描述的编码遍次互逆的解码遍次。

[0336] 在解码过程期间,视频解码器30从视频编码器20接收表示经编码视频切片的视频块和相关联语法元素的经编码视频位流。视频解码器30可从网络实体29接收视频数据,并将所述视频数据存储在视频数据存储器78中。视频数据存储器78可存储待由视频解码器30的组件解码的视频数据,例如经编码视频位流。存储在视频数据存储器78中的视频数据可(例如)从存储装置17、从例如相机等本地视频源、经由视频数据的有线或无线网络通信或者通过存取物理数据存储媒体而获得。视频数据存储器78可形成存储来自经编码视频位流的经编码视频数据的经译码图片缓冲器。因此,尽管图10中分开地展示,但视频数据存储器78及经解码图片缓冲器92可由相同存储器装置或单独存储器装置提供。视频数据存储器78和经解码图片缓冲器92可由多种存储器装置中的任一者形成,例如动态随机存取存储器(DRAM),包含同步DRAM(SDRAM)、磁阻式RAM(MRAM)、电阻式RAM(RRAM)或其它类型的存储器装置。在各种实例中,视频数据存储器78可与视频解码器30的其它组件一起在芯片上,或相对于那些组件在芯片外。

[0337] 网络实体29可(例如)为服务器、MANE、视频编辑器/拼接器,或经配置以实施上文所描述的技术中的一或者者的其它此类装置。网络实体29可包含或可不包含视频编码器,例如视频编码器20。在网络29将经编码视频位流发射到视频解码器30之前,可由网络实体29实施本发明中描述的技术中的一些。在一些视频解码系统中,网络实体29及视频解码器30可为单独装置的部分,而在其它情况下,关于网络实体29描述的功能性可由包括视频解码器30的相同装置执行。在一些情况下,网络实体29可为图1的存储装置17的实例。

[0338] 视频解码器30的熵解码单元80对位流进行熵解码以产生经量化系数、运动向量和其它语法元素。熵解码单元80将运动向量及其它语法元素转发到预测处理单元81。视频解码器30可在视频切片层级及/或视频块层级处接收语法元素。

[0339] 当视频切片经译码为经帧内译码(I)切片时,预测处理单元81的帧内预测处理单元84可基于经传信帧内预测模式和来自当前帧或图片的先前经解码块的数据而产生用于当前视频切片的视频块的预测数据。当将视频帧译码为经帧间译码(即,B或P)切片时,或当块经IMC译码时,预测处理单元81的运动补偿单元82基于从熵解码单元80接收到的运动向量和其它语法元素产生当前视频切片的视频块的预测性块。对于帧间预测,可从参考图片

列表中的一者内的参考图片中的一者产生预测性块。视频解码器30可以基于存储在经解码图片缓冲器92中的参考图片使用默认建构技术建构参考帧列表：列表0和列表1。对于IMC译码，可从与正预测的块相同的图片产生预测性块。

[0340] 运动补偿单元82通过解析运动向量和其它语法元素来确定用于当前视频切片的视频块的预测信息，并且使用所述预测信息产生用于正经解码的当前视频块的预测性块。举例来说，运动补偿单元82使用所接收到的语法元素中的一些确定用以译码视频切片的视频块的预测模式（例如，帧内预测或帧间预测）、帧间预测切片类型（例如，B切片或P切片）、切片的参考图片列表中的一或者者的建构信息、切片的每一经帧间编码视频块的运动向量、切片的每一经帧间译码视频块的帧间预测状态及用以解码当前视频切片中的视频块的其它信息。

[0341] 运动补偿单元82还可基于内插滤波器而执行内插。运动补偿单元82可使用由视频编码器20在视频块的编码期间使用的内插滤波器来计算参考块的子整数像素的内插值。在此情况下，运动补偿单元82可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，并使用所述内插滤波器来产生预测性块。

[0342] 根据本发明的一些技术，当使用IMC模式来译码视频块时，运动补偿单元82可确定视频块的明度分量的运动向量或偏移向量，且基于所说明度分量的运动向量来确定视频块的色度分量的运动向量。在另一个实例中，当使用IMC模式译码视频块时，运动补偿单元82可确定视频块的色度分量的运动向量或偏移向量，且基于色度分量的运动向量确定视频块的明度分量的运动向量。因此，视频解码器30可在位流中接收仅一个偏移向量，从所述偏移向量可确定视频块的色度和明度分量两者的偏移向量。

[0343] 当使用IMC模式来解码视频块时，运动补偿单元82可（例如）修改明度分量的运动向量（被称作IMC模式的偏移向量），以确定色度分量的偏移向量。运动补偿单元82可（例如）基于视频块的取样格式且基于偏移向量指向的子像素位置的精度，来修改明度块的块向量的x分量和y分量中的一者或两者。举例来说，如果使用4:2:2取样格式来译码视频块，那么运动补偿单元82可仅修改明度偏移向量的x分量，而不修改y分量，以确定色度分量的偏移向量。如从图4可见，在4:2:2取样格式中，色度块及明度块在垂直方向上具有相同数目个样本，因此使得可能不需要修改y分量。如果明度偏移向量在用于定位色度预测性块时指向无色度样本的位置（例如，在包含所述当前块的当前图片的色度样本中的子像素位置处），那么运动补偿单元82可仅修改明度偏移向量。如果明度偏移向量在用于定位色度预测性块时指向其中存在色度样本的位置，那么运动补偿单元82可不修改明度偏移向量。

[0344] 在另一实例中，如果使用4:2:0取样格式来译码视频块，那么运动补偿单元82可修改明度偏移向量的x分量和y分量中的任一者或两者，以确定色度分量的偏移向量。如从图3可见，在4:2:0取样格式中，色度块及明度块在垂直方向及水平方向上具有不同数目个样本。如果明度偏移向量在用于定位色度预测性块时指向无色度样本的位置（例如，在包含所述当前块的当前图片的色度样本中的子像素位置处），那么运动补偿单元82可仅修改明度偏移向量。如果明度偏移向量在用于定位色度预测性块时指向其中存在色度样本的位置，那么运动补偿单元82可不修改明度偏移向量。

[0345] 运动补偿单元82可修改明度偏移向量，以产生经修改运动向量（也被称作经修改偏移向量）。运动补偿单元82可修改在用以定位色度预测性块时指向子像素位置明度偏移

向量,使得用于色度块的经修改偏移向量指向较低分辨率子像素位置或指向整数像素位置。作为一个实例,可修改指向1/8像素位置的明度偏移向量以指向1/4像素位置,可修改指向1/4像素位置的明度块向量以指向1/2像素位置,等。在其它实例中,运动补偿单元82可修改明度偏移向量,使得经修改的偏移向量始终指向用于定位色度参考块的整数像素位置。修改明度偏移向量以指向较低分辨率子像素位置或指向整数像素位置可消除对一些内插滤波的需要,和/或降低任何所需的内插滤波的复杂度。

[0346] 参考图3及4且假定左上方样本定位于位置(0,0)处,视频块在奇数及偶数x位置两者和奇数及偶数y位置两者处都具有明度样本。在4:4:4取样格式中,视频块在奇数及偶数x位置两者及奇数及偶数y位置两者处还具有色度样本。因此,对于4:4:4取样格式,运动补偿单元可将相同偏移向量用于定位明度预测性块及色度预测性块两者。如图4中所展示,对于4:2:2取样格式,视频块在奇数及偶数y位置两者处但仅偶数x位置处具有色度样本。因此,对于4:2:2取样格式,如果明度偏移向量指向奇数x位置,那么运动补偿单元82可修改明度偏移向量的x分量以产生指向偶数x位置的经修改偏移向量,使得经修改偏移向量可用于为当前块的色度块定位参考色度块而无需内插。运动补偿单元82可(例如)通过上舍入或下舍入到最接近的偶数x位置来修改x分量,即改变x分量以使得其指向最接近左方x位置或最接近右方x位置。如果明度偏移向量已指向偶数x位置,那么可能不需要修改。

[0347] 如图3中所展示,对于4:2:0取样格式,视频块在仅偶数y位置处及仅偶数x位置处具有色度样本。因此,对于4:2:0取样格式,如果明度偏移向量指向奇数x位置或奇数y位置,那么运动补偿单元82可修改明度偏移向量的x分量或y分量以产生指向偶数x位置的经修改偏移向量,使得经修改偏移向量可用于为当前块的色度块定位参考色度块而无需内插。运动补偿单元82可(例如)通过上舍入或下舍入到最接近的偶数x位置来修改x分量,即改变x分量以使得其指向最接近左方x位置或最接近右方x位置。运动补偿单元82可(例如)通过上舍入或下舍入到最接近的偶数y位置来修改y分量,即改变y分量以使得其指向最接近上方y位置或最接近下方y位置。如果明度偏移向量已指向偶数x位置及偶数y位置,那么可能不需要修改。

[0348] 反量化单元86对提供于位流中且由熵解码单元80解码的经量化变换系数进行反量化(即,解量化)。反量化过程可包含使用由视频编码器20针对视频切片中的每一视频块计算的量化参数来确定应应用的量化的程度以及同样地反量化的程度。反变换处理单元88将反变换应用于变换系数,例如反DCT、反整数变换或概念上类似的反变换过程,以便产生像素域中的残余块。

[0349] 在运动补偿单元82基于运动向量和其它语法元素产生当前视频块的预测性块之后,视频解码器30通过将来自反变换处理单元88的残余块与由运动补偿单元82产生的对应预测性块求和来形成经解码视频块。求和器90表示执行此求和运算的一或多个组件。必要时,还可使用环路滤波器(在译码环路中或在译码环路之后)来使像素过渡平滑或者以其它方式改进视频质量。滤波器单元91既定表示一或多个环路滤波器,例如,解块滤波器、自适应环路滤波器(ALF)及样本自适应偏移(SAO)滤波器。尽管在图10中将滤波器单元91展示为环内滤波器,但在其它配置中,可将滤波器单元91实施为环路后滤波器。接着将给定帧或图片中的经解码视频块存储在经解码图片缓冲器92中,所述经解码图片缓冲器存储用于后续运动补偿的参考图片。经解码图片缓冲器92可为存储器的一部分,其还存储经解码视频,以

供稍后在显示装置(例如图1的显示装置32)上呈现,或可与此类存储器分开。

[0350] 图10的视频解码器30表示可经配置以接收语法元素(例如,上文所描述的curr_pic_as_ref_enabled_flag)的视频解码器的实例,所述语法元素指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表(例如列表0或列表1中的一者或两者)中。视频解码器30可确定语法元素的值(例如对于1位语法元素为1或0),且响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中,将当前图片添加到参考图片列表。响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中,视频解码器30可限制将当前图片放置于参考图片列表中。

[0351] 视频解码器30的熵解码单元80可接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素。视频解码器30的熵解码单元80可进一步确定语法元素的值。响应于语法元素指示视频数据的当前视频块有资格处于参考图片列表中,视频解码器30的运动补偿单元82可将当前图片添加到参考图片列表中。响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中,视频解码器30的运动补偿单元82可限制将当前图片放置于参考图片列表中。

[0352] 在一些实例中,当前图片是包括P切片或B切片中的一者的帧内随机存取图片(IRAP)。在此些实例中,IRAP图片可进一步包含指定非空参考图片集的一或多个不同语法元素的集合。在一些实例中,当前图片的当前视频块包含明度分量及色度分量。在此些实例中,明度或色度分量中的仅一者可具有整数像素精度。

[0353] 在一些实例中,参考图片可进一步包含当前图片的经解码未滤波样本。在各种个例中,参考图片可包含进一步包含运动预测信息的参考块。视频解码器30的运动补偿单元82可将参考块的运动信息用于译码当前视频块。运动补偿单元82可(例如)解码以合并模式译码的块。如果当前块的图片有资格处于参考图片列表中,那么合并候选者列表中的合并候选者中的一或者可包含指向与当前块处于同一图片中的参考块的运动向量(例如偏移向量)。

[0354] 在参考图片的POC值不等于当前图片的POC值的一些实例中,指代当前图片的SPS的CVS可包含一或多个I切片。

[0355] 在一些实例中,当前图片是清洁随机存取图片或断链存取图片中的一者。在此些实例中,视频解码器30的熵解码单元80可接收指示多个图片有资格用作当前图片的预测参考的第二语法元素,其中第二语法元素的值为1。

[0356] 在参考图片的POC值等于当前图片的POC值的一些实例中,当前视频块的MV可具有整数像素精度。此外,视频解码器30的熵解码单元80可在将当前图片的多个运动向量预测符候选者中的每一MV预测符候选者用于候选者导出过程之前或在候选者导出过程完成之后将所述每一运动向量预测符候选者转换为整数像素精度。在参考图片的POC值等于当前图片的POC值的其它实例中,当前图片的每一预测样本包含在当前图片的图片边界内。此外,如果参考图片列表仅包含当前图片,那么视频解码器30的熵解码单元80可停用TMVP。

[0357] 图11是说明根据本文中所描述的一或多种技术的实例视频编码技术的流程图。出于描述此流程图的目的,将对视频译码装置进行参考。应认识到,包括存储器及一或多个处理器的视频译码装置可包含视频编码器,例如图1及9的视频编码器20。此外,视频译码装置可将以下技术用于编码视频块。

[0358] 根据本文中所描述的一或多种技术,视频译码装置可经配置以执行本发明的技术。举例来说,视频译码装置可经配置以确定用于译码当前视频块的参考图片(200)。视频译码装置可进一步确定参考图片的POC值(202)。响应于参考图片的POC值等于包含当前视频块的当前图片的POC值(204的“是”分支),视频译码装置可设定语法元素的值以指示参考图片列表包含当前图片(206)。在一些实例中,语法元素可位于PPS或SPS中的一者中。相反地,响应于参考图片的POC值不等于包含当前视频块的当前图片的POC值(204的“否”分支),视频译码装置可设定语法元素的值以指示参考图片列表不包含当前图片(208)。

[0359] 在一些实例中,当前图片是包括P切片或B切片中的一者的帧内随机存取图片(IRAP)。在此些实例中,IRAP图片可进一步包含指定非空参考图片集的一或多个不同语法元素的集合。在一些实例中,当前图片的当前视频块包含明度分量及色度分量。在此些实例中,明度或色度分量中的仅一者可具有整数像素精度。

[0360] 在一些实例中,参考图片可进一步包含当前图片的经解码未滤波样本。在各种个例中,参考图片可包含进一步包含运动预测信息的参考块。视频译码装置可将参考块的运动信息用于译码当前视频块。

[0361] 在参考图片的POC值不等于当前图片的POC值的一些实例中,指代当前图片的SPS的CVS可包含一或多个I切片。

[0362] 在一些实例中,当前图片是清洁随机存取图片或断链存取图片中的一者。在此些实例中,视频译码装置可将指示可用作当前图片的预测参考的多个图片的第二语法元素的值设定为1。

[0363] 在参考图片的POC值等于当前图片的POC值的一些实例中,当前视频块的MV可具有整数像素精度。此外,视频译码装置可在将当前图片的多个运动向量预测符候选者中的每一MV预测符候选者用于候选者导出过程之前或在候选者导出过程完成之后将所述每一运动向量预测符候选者转换为整数像素精度。在参考图片的POC值等于当前图片的POC值的其它实例中,当前图片的每一预测样本包含在当前图片的图片边界内。

[0364] 在一些实例中,如果参考图片列表仅包含当前图片,那么视频译码装置可停用TMVP。在其它实例中,可对视频译码装置提出约束以使得视频译码装置可至少部分地基于不同于当前图片的图片确定TMVP。

[0365] 在一些实例中,视频译码装置可确定当前图片的相邻图片,其中相邻块为空间上相邻图片或时间上相邻图片。视频译码装置可进一步确定相邻图片中的预测性块,其中预测性块包含运动向量。视频译码装置可使用运动向量确定当前块的参考图片中的参考块,其中所述参考块包含运动预测信息。视频译码装置可随后使用参考块的运动预测信息及残余来编码当前视频块。

[0366] 图12是说明根据本文中所描述的一或多种技术的实例视频编码技术的流程图。出于描述此流程图的目的,将对视频译码装置进行参考。应认识到,包括存储器及一或多个处理器的视频译码装置可包含视频解码器,例如图1及图10的视频解码器30。此外,视频译码装置可将以下技术用于解码视频块。

[0367] 根据本文中所描述的一或多种技术,视频译码装置可接收指示包含视频数据的当前视频块的当前图片是否有资格处于参考图片列表中的语法元素(220)。视频译码装置可进一步确定语法元素的值(222)。响应于语法元素指示视频数据的当前视频块有资格处于

参考图片列表中(224的“是”分支),视频译码装置可将当前图片添加到参考图片列表(226)。响应于语法元素指示视频数据的当前视频块没资格处于参考图片列表中(224的“否”分支),视频译码装置可限制将当前图片放置于参考图片列表中(228)。

[0368] 在其它实例中,视频译码装置可经配置以执行本发明的技术。举例来说,视频译码装置可接收指示用于解码视频数据的当前视频块的参考图片的POC值是否与包含当前视频块的当前图片的POC值相同的语法元素。在一些实例中,语法元素可位于PPS或SPS中的一者中。视频译码装置可进一步确定语法元素的值。响应于语法元素指示参考图片的POC值等于当前图片的POC值,视频译码装置可将当前图片添加到参考图片列表。响应于语法元素指示参考图片的POC值不同于当前图片的POC值,视频译码装置可确定参考图片的POC值且将参考图片添加到参考图片列表。

[0369] 在一些实例中,当前图片是包括P切片或B切片中的一者的帧内随机存取图片(IRAP)。在此些实例中,IRAP图片可进一步包含指定非空参考图片集的一或多个不同语法元素的集合。在一些实例中,当前图片的当前视频块包含明度分量及色度分量。在此些实例中,明度或色度分量中的仅一者可具有整数像素精度。

[0370] 在一些实例中,参考图片可进一步包含当前图片的经解码未滤波样本。在各种个例中,参考图片可包含进一步包含运动预测信息的参考块。视频译码装置可将参考块的运动信息用于译码当前视频块。

[0371] 在参考图片的POC值不等于当前图片的POC值的一些实例中,指代当前图片的SPS的CVS可包含一或多个I切片。

[0372] 在一些实例中,当前图片是清洁随机存取图片或断链存取图片中的一者。在此些实例中,视频译码装置可接收指示多个图片有资格用作当前图片的预测参考的第二语法元素,其中所述第二语法元素的值为1。

[0373] 在参考图片的POC值等于当前图片的POC值的一些实例中,当前视频块的MV可具有整数像素精度。此外,视频译码装置可在将当前图片的多个运动向量预测符候选者中的每一MV预测符候选者用于候选者导出过程之前或在候选者导出过程完成之后将所述每一运动向量预测符候选者转换为整数像素精度。在参考图片的POC值等于当前图片的POC值的其它实例中,当前图片的每一预测样本包含在当前图片的图片边界内。

[0374] 在一些实例中,如果参考图片列表仅包含当前图片,那么视频译码装置可停用TMVP。在其它实例中,可对视频译码装置提出约束以使得视频译码装置可至少部分地基于不同于当前图片的图片确定TMVP。

[0375] 在一些实例中,视频译码装置可确定当前图片的相邻图片,其中相邻块为空间上相邻图片或时间上相邻图片。视频译码装置可进一步确定相邻图片中的预测性块,其中预测性块包含运动向量。视频译码装置可使用运动向量确定当前块的参考图片中的参考块,其中所述参考块包含运动预测信息。视频译码装置可随后使用参考块的运动预测信息及残余来解码当前视频块。

[0376] 在一或多个实例中,所描述的功能可实施在硬件、软件、固件或其任何组合中。如果用软件实施,则所述功能可以作为一或多个指令或代码在计算机可读媒体上存储或发射,并且由基于硬件的处理单元执行。计算机可读媒体可包括计算机可读存储媒体,其对应于有形媒体,例如数据存储媒体,或包括任何促进将计算机程序从一处传送到另一处的媒

体(例如,根据一种通信协议)的通信媒体。以此方式,计算机可读媒体通常可对应于(1)有形计算机可读存储媒体(其是非暂时的)或(2)通信媒体,例如信号或载波。数据存储媒体可以是可由一或多个计算机或一或多个处理器存取以检索用于实施本发明中描述的技术的指令、代码和/或数据结构的任何可用的媒体。计算机程序产品可包括计算机可读媒体。

[0377] 借助于实例而非限制,此类计算机可读存储媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置、快闪存储器或可以用来存储指令或数据结构的形式的期望程序代码并且可以由计算机存取的任何其它媒体。并且,任何连接可恰当地被称为计算机可读媒体。举例来说,如果使用同轴电缆、光纤缆线、双绞线、数字订户线(DSL)或例如红外线、无线电及微波等无线技术从网站、服务器或其它远程源传输指令,那么将同轴电缆、光纤缆线、双绞线、DSL或例如红外线、无线电及微波等无线技术包含在媒体的定义中。然而,应理解,所述计算机可读存储媒体及数据存储媒体并不包含连接、载波、信号或其它暂时媒体,而是实际上针对于非暂时性有形存储媒体。如本文所使用,磁盘及光盘包含压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软性磁盘及蓝光光盘,其中磁盘通常以磁性方式复制数据,而光盘用激光以光学方式复制数据。上述各者的组合也应包含在计算机可读媒体的范围内。

[0378] 指令可由一或多个处理器执行,所述一或多个处理器例如是一或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程逻辑阵列(FPGA)、或其它等效的集成或离散逻辑电路。因此,如本文中所使用的术语“处理器”可指代上述结构或适合于实施本文中所描述的技术的任何其它结构中的任一者。此外,在一些方面中,本文所述的功能性可在经配置用于编码及解码的专用硬件及/或软件模块内提供,或并入在组合编解码器中。并且,所述技术可完全在一或多个电路或逻辑元件中实现。

[0379] 本发明的所述技术可在包含无线手持机、集成电路(IC)或一组IC(例如,芯片组)的广泛多种装置或设备中实施。本发明中描述各种组件、模块或单元来强调经配置以执行所揭示的技术的装置的功能方面,但其未必需要由不同硬件单元实现。实际上,如上文所描述,各种单元可以结合合适的软件及/或固件组合在编解码器硬件单元中,或者通过互操作硬件单元的集合来提供,所述硬件单元包含如上文所描述的一或多个处理器。

[0380] 已描述本发明的各种实例。预期所描述的系统、操作或功能的任何组合。这些及其他实例在以下权利要求书的范围内。

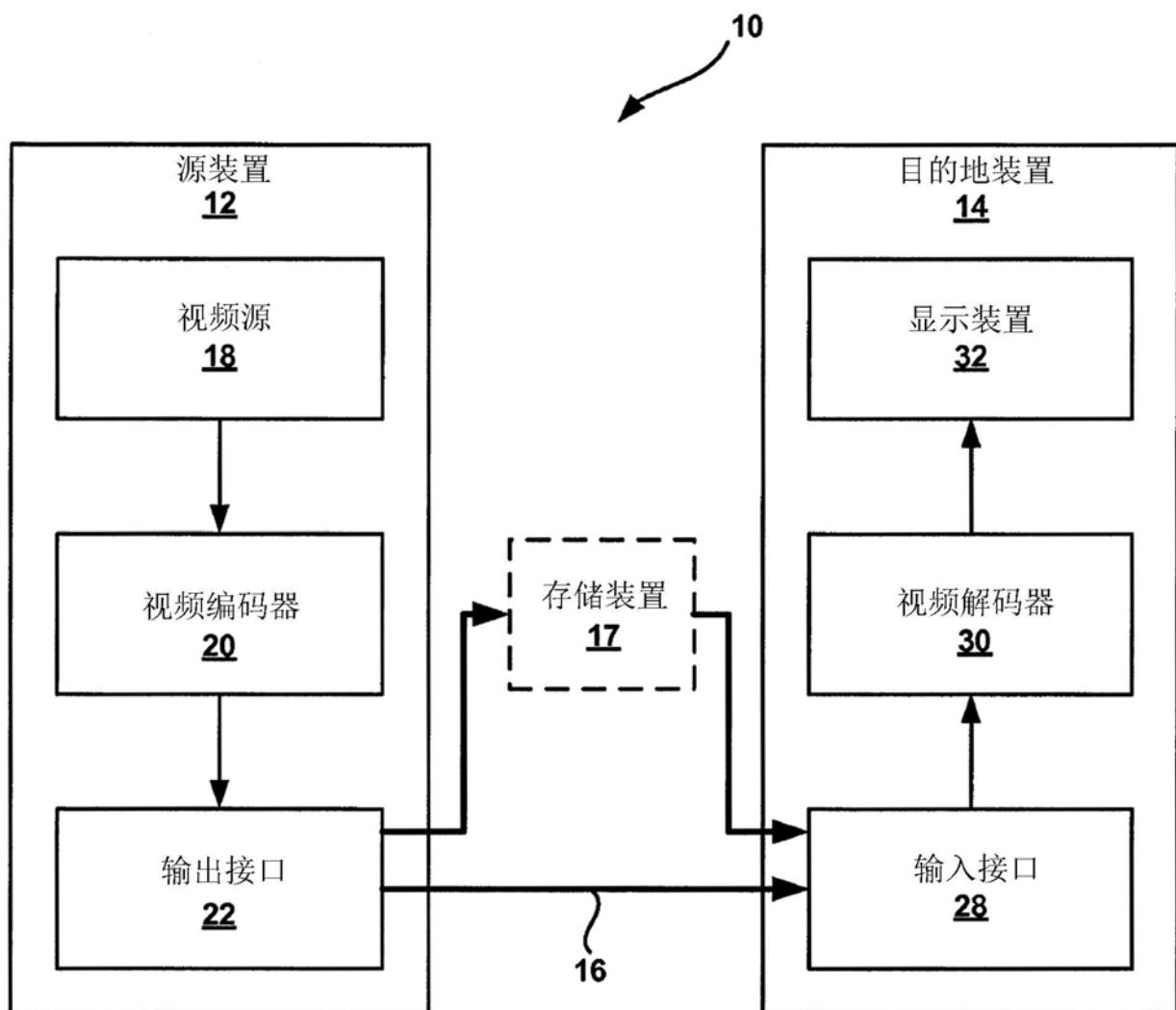
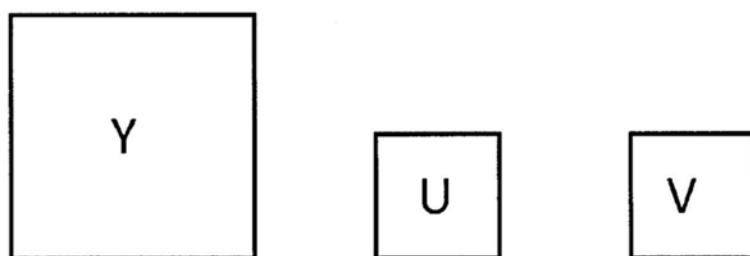
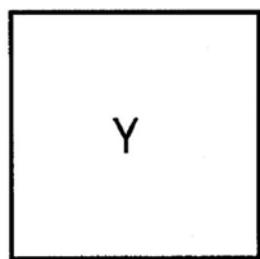


图1



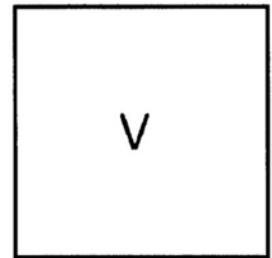
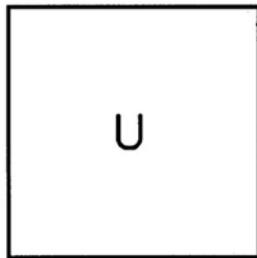
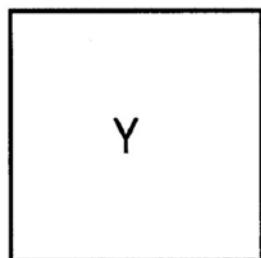
4:2:0样本格式

图2A



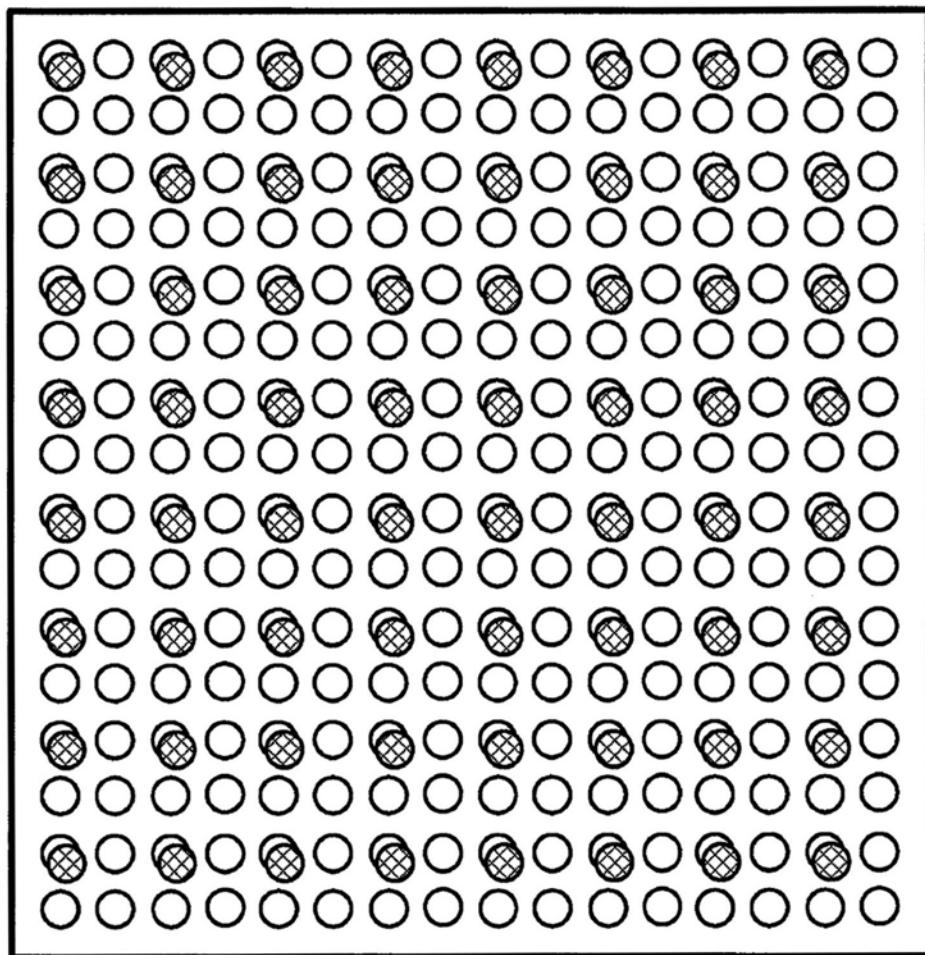
4:2:2样本格式

图2B



4:4:4样本格式

图2C

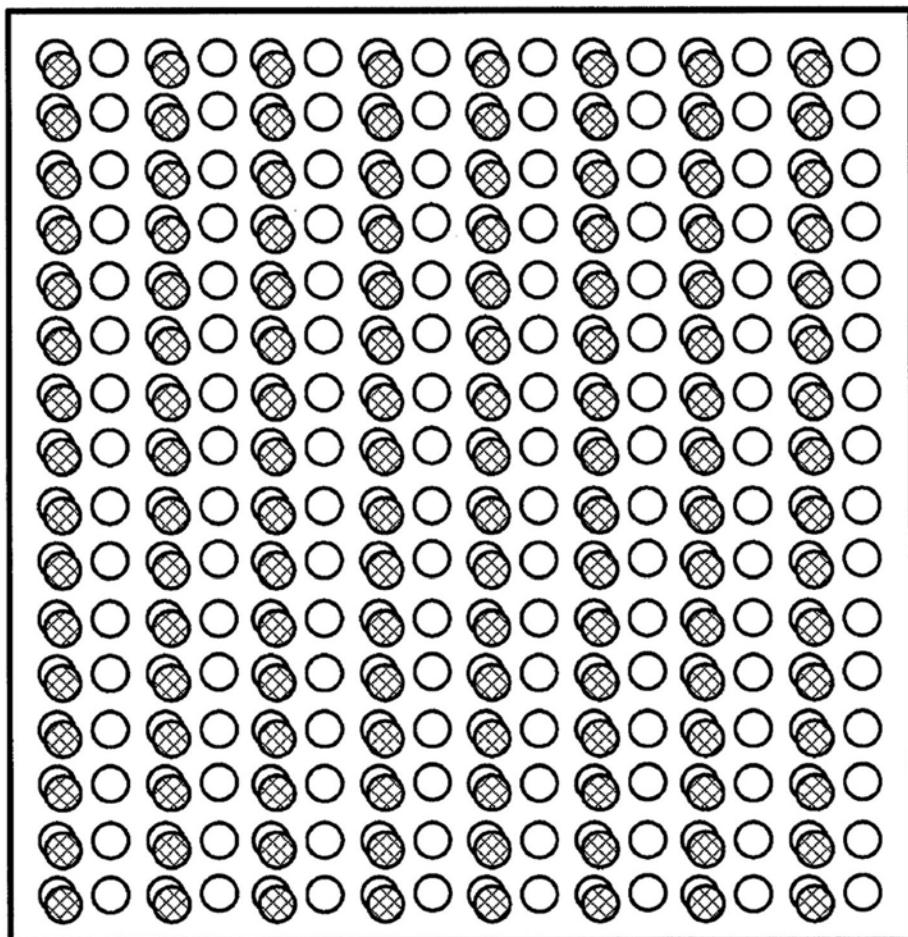


以4:2:0样本格式化的 16×16 译码单元

○ 明度样本

◎ 色度样本

图3



以4:2:2样本格式化的16×16译码单元

- 明度样本
- ◎ 色度样本

图4

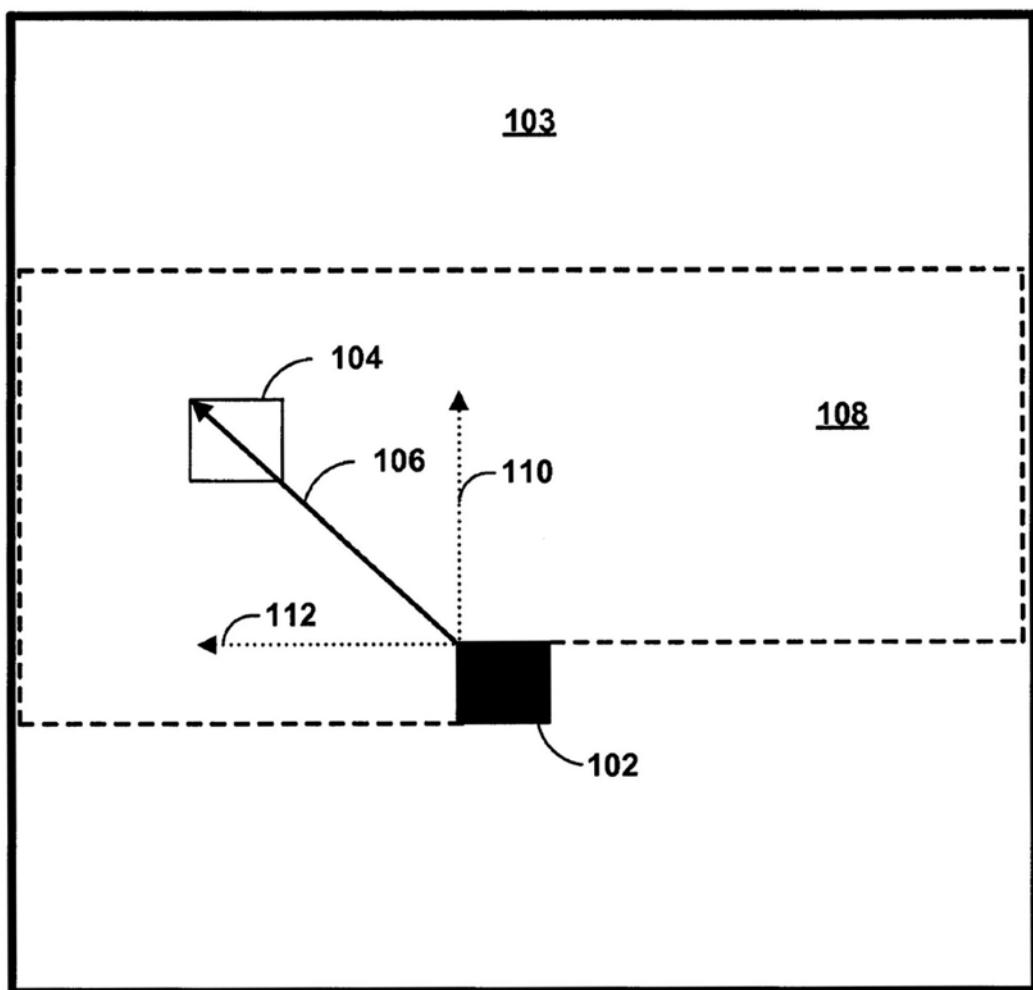


图5

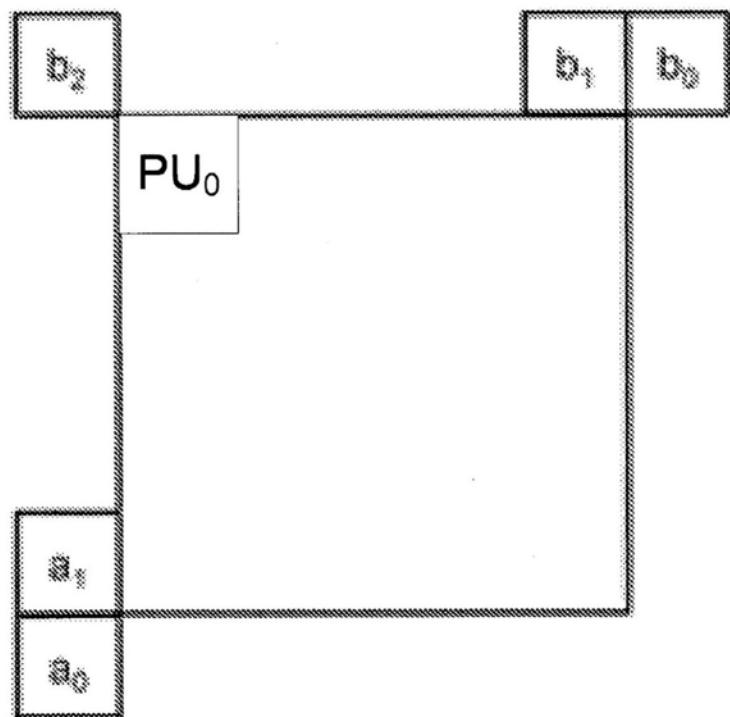


图6

P				Q			
$p_{3,0}$	$p_{2,0}$	$p_{1,0}$	$p_{0,0}$	$q_{3,0}$	$q_{2,0}$	$q_{1,0}$	$q_{0,0}$
$p_{3,1}$	$p_{2,1}$	$p_{1,1}$	$p_{0,1}$	$q_{3,1}$	$q_{2,1}$	$q_{1,1}$	$q_{0,1}$
$p_{3,2}$	$p_{2,2}$	$p_{1,2}$	$p_{0,2}$	$q_{3,2}$	$q_{2,2}$	$q_{1,2}$	$q_{0,2}$
$p_{3,3}$	$p_{2,3}$	$p_{1,3}$	$p_{0,3}$	$q_{3,3}$	$q_{2,3}$	$q_{1,3}$	$q_{0,3}$

由相邻块P及Q形成的四个像素长的垂直块边界

图7

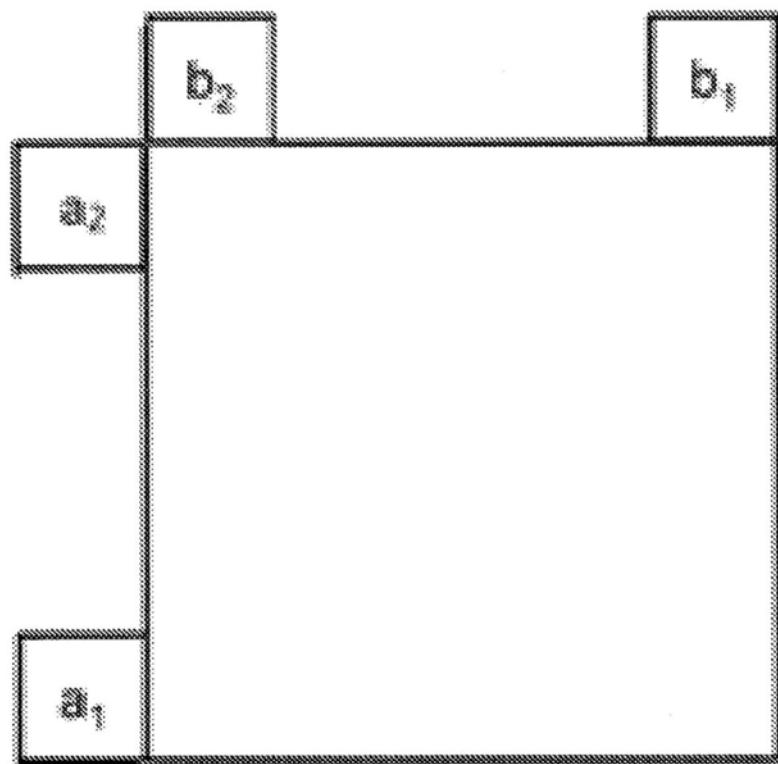


图8

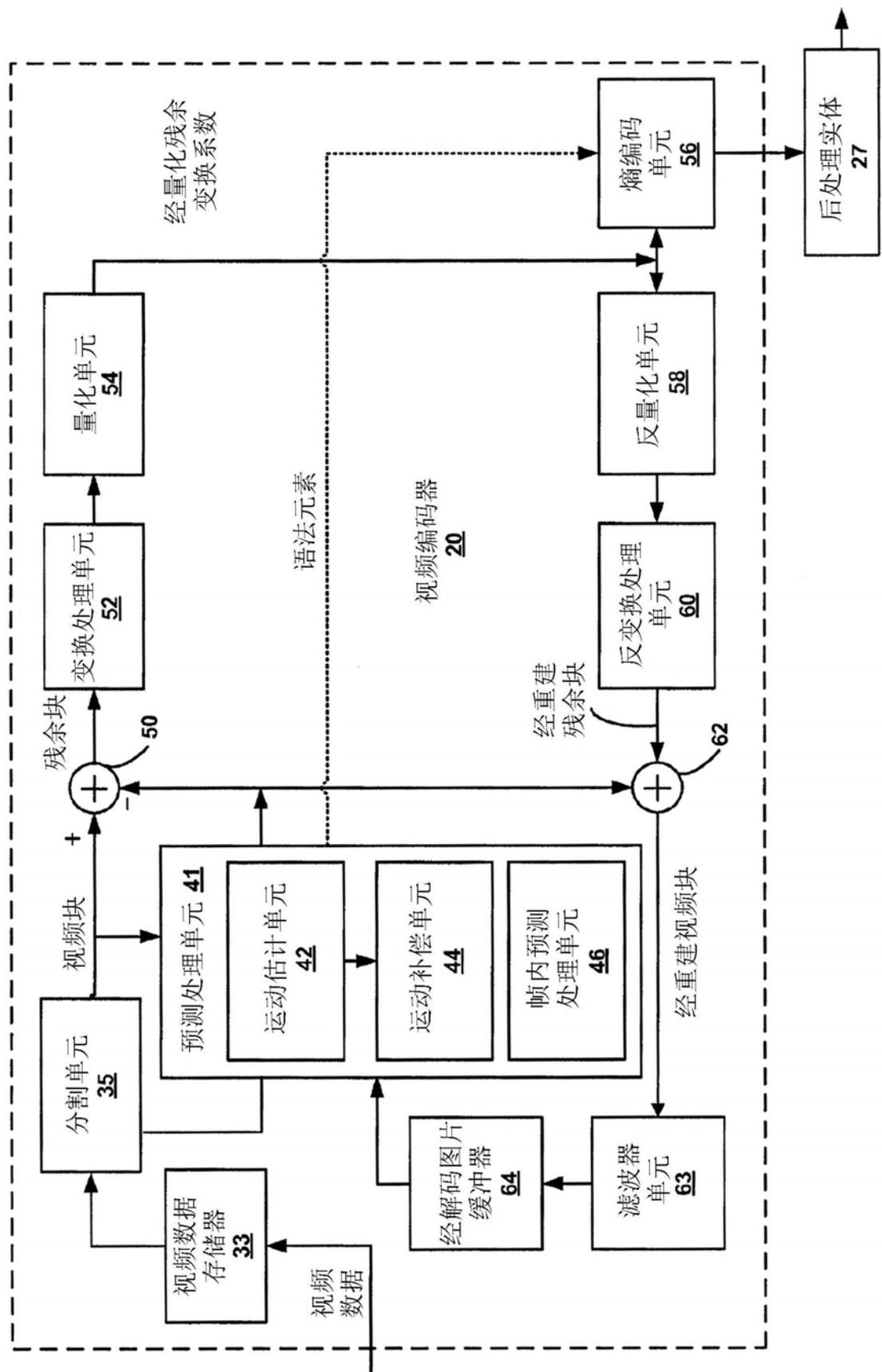


图9

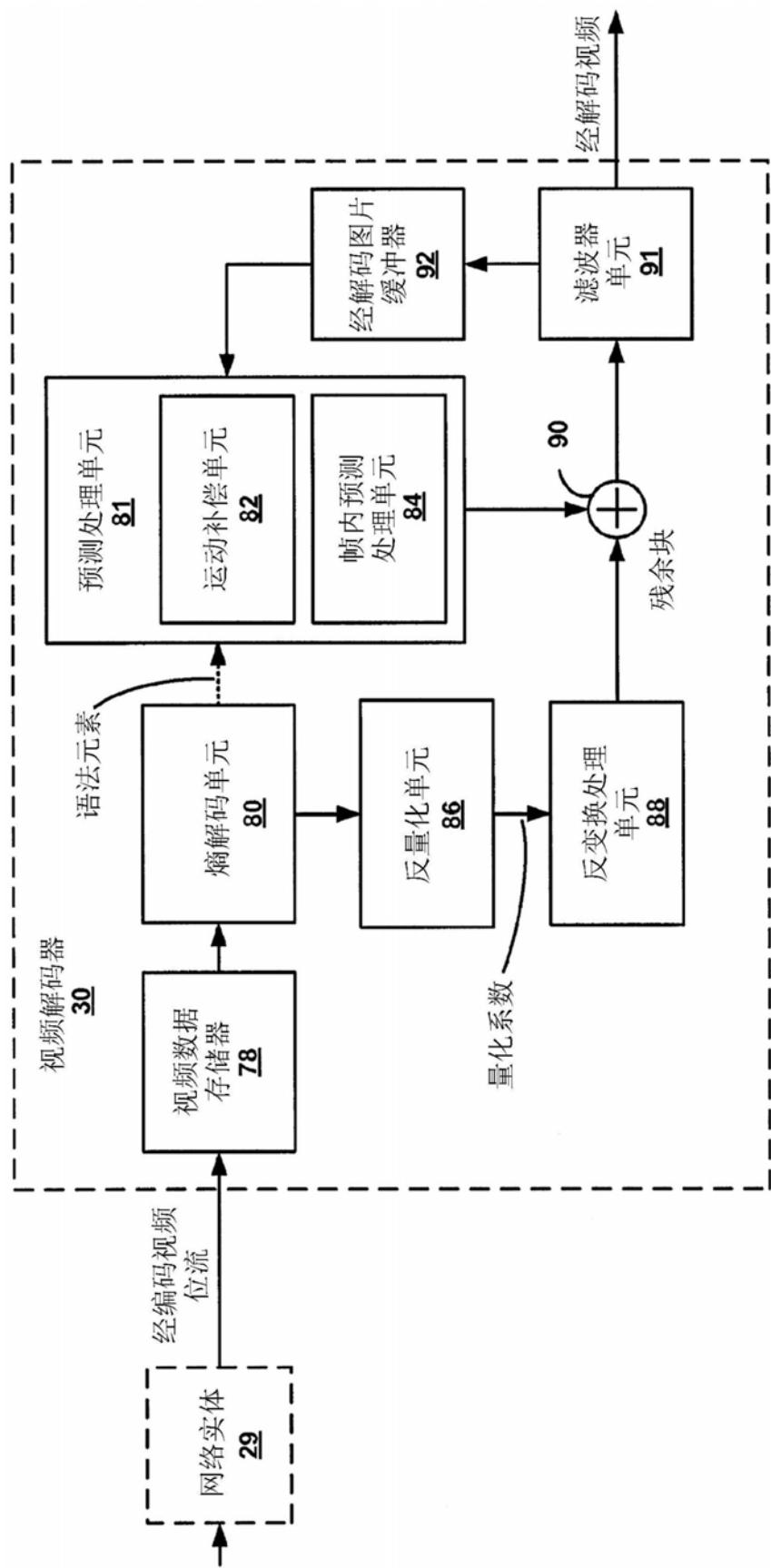


图 10

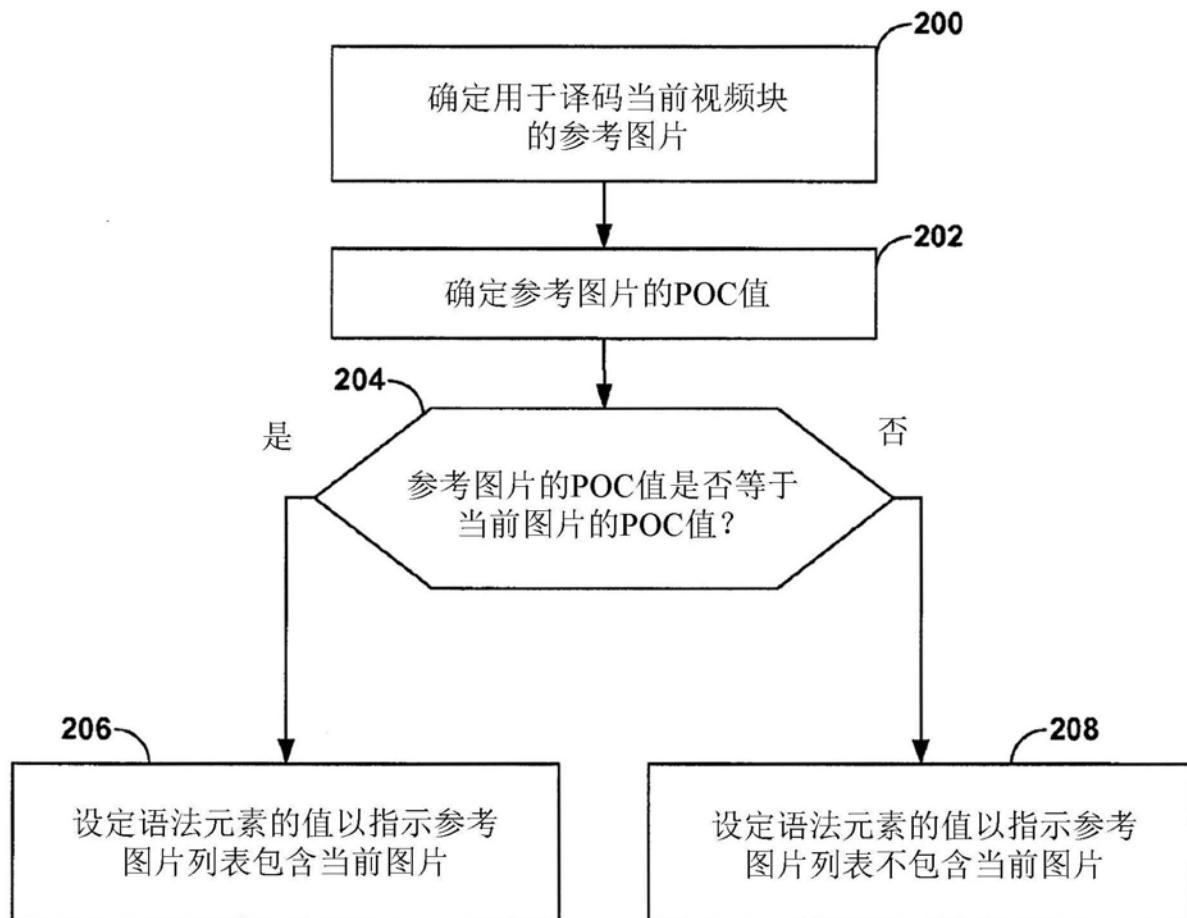


图11

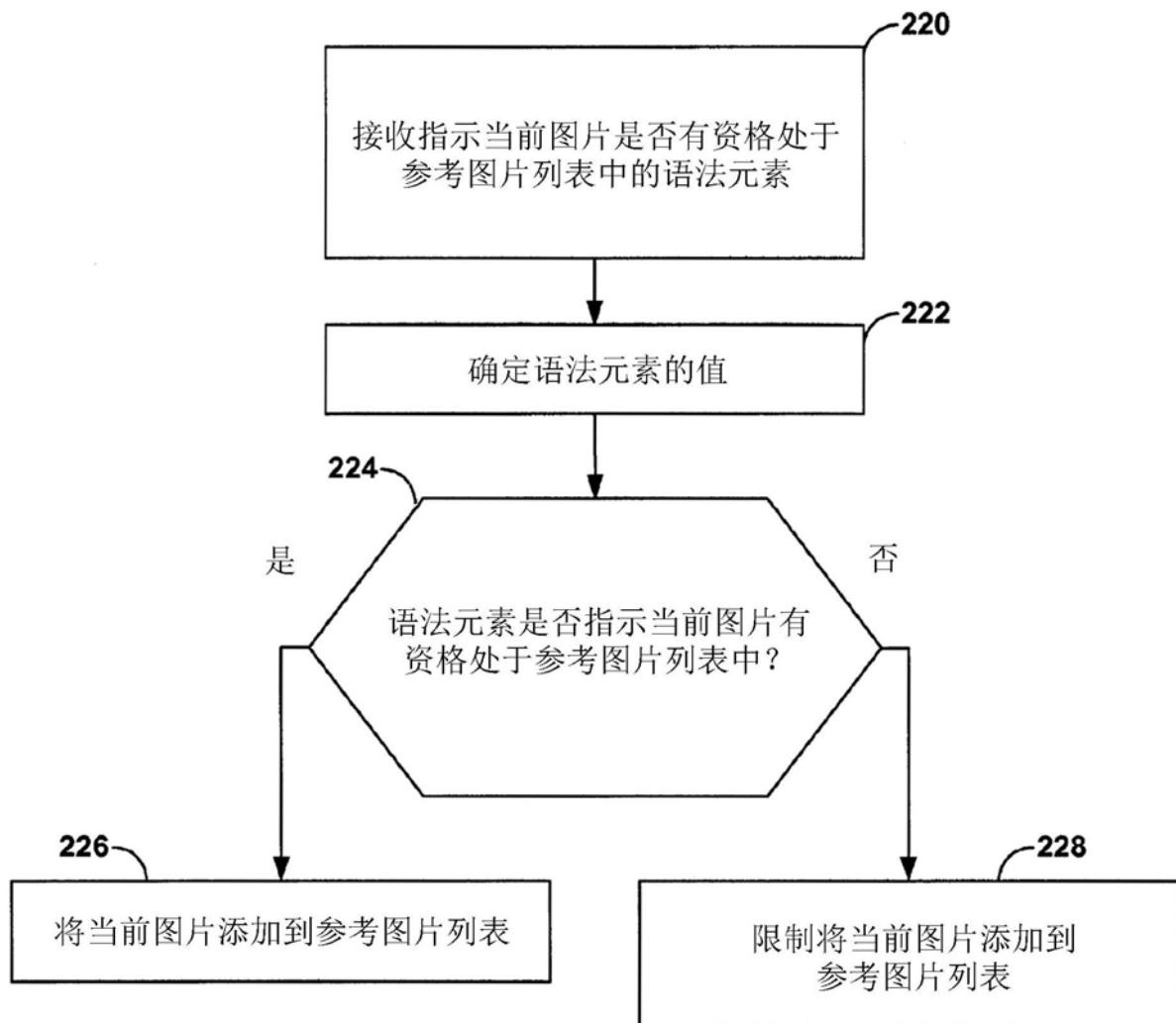


图12