

视频解码方法及视频解码器

本申请要求于 2018 年 9 月 10 日提交中国国家知识产权局、申请号为 201811053068.0、发明名称为“视频解码方法及视频解码器”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5 技术领域

本申请实施例大体上涉及视频编码领域，更确切地说，涉及视频解码方法及视频解码器。

背景技术

视频编码（视频编码和解码）广泛用于数字视频应用，例如广播数字电视、互联网和
10 移动网络上的视频传播、视频聊天和视频会议等实时会话应用、DVD 和蓝光光盘、视频内容采集和编辑系统以及便携式摄像机的安全应用。

随着 1990 年 H.261 标准中基于块的混合型视频编码方式的发展，新的视频编码技术和工具得到发展并为新的视频编码标准形成基础。其它视频编码标准包括 MPEG-1 视频、MPEG-2 视频、ITU-T H.262/MPEG-2、ITU-T H.263、ITU-T H.264/MPEG-4 第 10 部分高级
15 视频编码(Advanced Video Coding, AVC)、ITU-T H.265/高效视频编码(High Efficiency Video Coding, HEVC) ... 以及此类标准的扩展，例如可扩展性和/或 3D (three-dimensional) 扩展。随着视频创建和使用变得越来越广泛，视频流量成为通信网络和数据存储的最大负担。因此大多数视频编码标准的目标之一是相较之前的标准，在不牺牲图片质量的前提下减少比特率。即使最新的高效视频编码 (High Efficiency video coding, HEVC) 可以在不牺牲图片
20 质量的前提下比 AVC 大约多压缩视频一倍，仍然亟需新技术相对 HEVC 进一步压缩视频。

发明内容

本申请实施例提供视频解码方法及视频解码器，可以减少编码器或解码器存储上下文所需的空

前述和其它目标通过独立权利要求的主题实现。其它实现方式通过从属权利要求、说
25 明书以及附图是显而易见的。

第一方面，提供了一种视频解码方法，包括：解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；对所述当前块的待熵解码的语法元素进行熵解码；其中，使用预设的上下文模型完成所述当前块的语法元素 1 的熵解码；或使用所述上下文模型完成所述当前
30 块的语法元素 2 的熵解码；基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；基于所述当前块的预测块获得所述当前块的重构图像。

其中，由于当前块的语法元素 1 和语法元素 2 共用了一个上下文模型，因此解码器在进行熵解码时不需要进行上下文模型的确认，从而提高了解码器进行视频解码的解码效率；
35 同时，由于视频解码器针对语法元素 1 和语法元素 2 只需要存储一个上下文模型，能够减

少视频解码器的存储空间占用。

第二方面，提供了一种视频解码方法，包括：解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素 1 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 2 的上下文模型；基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码；基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；基于所述当前块的预测块获得所述当前块的重构图像。

- 10 其中，由于当前块的语法元素 1 和语法元素 2 共用了一个上下文模型集合，因此视频解码器针对语法元素 1 和语法元素 2 也只需要存储一个上下文模型，减少了视频解码器的存储空间占用。

结合第二方面，在可能的实施方式中，所述预设的上下文模型集合中的上下文模型的数量为 2 个或 3 个。

- 15 结合第二方面，在可能的实施方式中，所述从预设的上下文模型集合中确定所述当前块的语法元素 1 的上下文模型包括：根据所述当前块的左方相邻块的语法元素 1 和语法元素 2，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 1 的上下文索引，所述当前块的语法元素 1 的上下文索引用于指示当前块的语法元素 1 的上下文模型；

- 20 或

所述从预设的上下文模型集合中确定所述当前块的语法元素 2 的上下文模型包括：根据所述当前块的左方相邻块的语法元素 1 和语法元素 2，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 2 的上下文索引，所述当前块的语法元素 2 的上下文索引用于指示当前块的语法元素 2 的上下文模型。

- 25 结合第二方面，在可能的实施方式中，若所述预设的上下文模型集合中的上下文模型的数量为 3 个，所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和；

或

- 30 所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和。

结合第二方面，在可能的实施方式中，若所述预设的上下文模型集合中的上下文模型的数量为 2 个，所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作

35 的值再进行或操作的结果；

或

所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果。

结合第一方面或第二方面，在可能的实施方式中，所述当前块的语法元素 1 为
5 affine_merge_flag，用于指示所述当前块是否采用基于仿射运动模型的 merge 模式；或所述当前块的语法元素 2 为 affine_inter_flag，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式；

或

所述当前块的语法元素 1 为 subblock_merge_flag，用于指示当前块是否采用基于子块
10 的 merge 模式；或所述当前块的语法元素 2 为 affine_inter_flag，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式。

第三方面，提供了一种视频解码方法，包括：解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 3 或所述当前块的语法元素 4；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上
15 下文模型集合中确定所述当前块的语法元素 3 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 4 的上下文模型；基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码；基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；基于所述当前块的预测块获得所述当前块的重构图像。

20 其中，由于当前块的语法元素 3 和语法元素 4 共用了一个上下文模型集合，因此视频解码器针对语法元素 3 和语法元素 4 也只需要存储一个上下文模型，减少了视频解码器的存储空间占用。

结合第三方面，在可能的实施方式中，所述预设的上下文模型集合包括 5 个上下文模型。

25 结合第三方面，在可能的实施方式中，所述当前块的语法元素 3 为 merge_idx，用于指示所述当前块的 merge 候选列表的索引值；或所述当前块的语法元素 4 为 affine_merge_idx，用于指示所述当前块的仿射 merge 候选列表的索引值；

或

所述当前块的语法元素 3 为 merge_idx，用于指示所述当前块的 merge 候选列表的索引
30 值；或所述当前块的语法元素 4 为 subblock_merge_idx，用于指示针对子块 merge 候选列表的索引值。

第四方面，提供了一种视频解码方法，包括：解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；根据所述当前块的左方相邻块的语法元素 1 和语法元素 2 的取值，以及
35 及所述当前块的上方相邻块的语法元素 1 和语法元素 2 的取值，确定所述当前块的待熵解码的语法元素的上下文索引的值；基于所述当前块的待熵解码的语法元素的上下文索引的

值对所述待熵解码的语法元素进行熵解码；基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；基于所述当前块的预测块获得所述当前块的重构图像。

- 结合第四方面，在可能的实施方式中，所述当前块的语法元素 1 为 `affine_merge_flag`，
5 用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式；或所述当前块的语法元素 2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式；

或

- 所述当前块的语法元素 1 为 `subblock_merge_flag`，用于指示当前块是否采用基于子块
10 的 `merge` 模式；或所述当前块的语法元素 2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式。

结合第四方面，在可能的实施方式中，所述根据所述当前块的左方相邻块的语法元素 1 和语法元素 2 的取值，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 的取值，确定所述当前块的待熵解码的语法元素的上下文索引的值包括：

- 15 根据如下逻辑表达式确定所述当前块的待熵解码的语法元素的上下文索引的值：

$$\text{Context index} = (\text{condL} \ \&\& \ \text{availableL}) + (\text{condA} \ \&\& \ \text{availableA});$$

其中， $\text{condL} = \text{语法元素 } 1[x0-1][y0] \mid \text{语法元素 } 2[x0-1][y0]$ ，语法元素 $1[x0-1][y0]$ 表示所述左方相邻块的语法元素 1 的取值，语法元素 $2[x0-1][y0]$ 表示所述左方相邻块的语法元素 2 的取值；

- 20 $\text{condA} = \text{语法元素 } 1[x0][y0-1] \mid \text{语法元素 } 2[x0][y0-1]$ ，语法元素 $1[x0][y0-1]$ 表示所述上方相邻块的语法元素 1 的取值，语法元素 $2[x0][y0-1]$ 表示所述上方相邻块的语法元素 2 的取值；

availableL 表示所述左方相邻块是否可得， availableA 表示所述上方相邻块是否可得。

- 第五方面，提供了一种视频解码器，包括：熵解码单元，用于解析接收的码流，以获
25 得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；根据所述当前块的左方相邻块的语法元素 1 和语法元素 2 的取值，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 的取值，确定所述当前块的待熵解码的语法元素的上下文索引的值；基于所述当前块的待熵解码的语法元素的上下文索引的值对所述待熵解码的语法元素进行熵解码；预测处理单元，用于基于熵解
30 码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；重构单元，用于基于所述当前块的预测块获得所述当前块的重构图像。

- 结合第五方面，在可能的实施方式中，所述当前块的语法元素 1 为 `affine_merge_flag`，
用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式；或所述当前块的语法元素
2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述
35 当前块是否采用基于仿射运动模型的 AMVP 模式；

或

所述当前块的语法元素 1 为 subblock_merge_flag, 用于指示当前块是否采用基于子块的 merge 模式; 或所述当前块的语法元素 2 为 affine_inter_flag, 用于在所述当前块所在条带为 P 型条带或者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 AMVP 模式。

5 结合第五方面, 在可能的实施方式中, 所述熵解码单元具体用于:

根据如下逻辑表达式确定所述当前块的待熵解码的语法元素的上下文索引的值:

Context index=(condL && availableL)+(condA && availableA);

其中, condL= 语法元素 1[x0-1][y0] | 语法元素 2[x0-1][y0], 语法元素 1[x0-1][y0] 表示所述左方相邻块的语法元素 1 的取值, 语法元素 2[x0-1][y0] 表示所述左方相邻块的语法元素 2 的取值;

condA=语法元素 1[x0][y0-1] | 语法元素 2[x0][y0-1], 语法元素 1[x0][y0-1] 表示所述上方相邻块的语法元素 1 的取值, 语法元素 2[x0][y0-1] 表示所述上方相邻块的语法元素 2 的取值;

availableL 表示所述左方相邻块是否可得, availableA 表示所述上方相邻块是否可得。

15 第六方面, 提供了一种视频解码器, 包括: 熵解码单元, 用于解析接收的码流, 以获得当前块的待熵解码的语法元素, 所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2; 对所述当前块的待熵解码的语法元素进行熵解码; 其中, 使用预设的上下文模型完成所述当前块的语法元素 1 的熵解码; 或使用所述上下文模型完成所述当前块的语法元素 2 的熵解码; 预测处理单元, 用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理, 以获得所述当前块的预测块; 重构单元, 用于基于所述当前块的预测块获得所述当前块的重构图像。

20 第七方面, 提供了一种视频解码器, 包括: 熵解码单元, 用于解析接收的码流, 以获得当前块的待熵解码的语法元素, 所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2; 获取所述待熵解码的语法元素对应的上下文模型; 其中, 从预设的上下文模型集合中确定所述当前块的语法元素 1 的上下文模型; 或从预设的所述上下文模型集合中确定所述当前块的语法元素 2 的上下文模型; 基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码; 预测处理单元, 用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理, 以获得所述当前块的预测块; 重构单元, 用于基于所述当前块的预测块获得所述当前块的重构图像。

30 结合第七方面, 在可能的实施方式中, 所述预设的上下文模型集合中的上下文模型的数量为 2 个或 3 个。

结合第七方面, 在可能的实施方式中, 所述熵解码单元具体用于: 根据所述当前块的左方相邻块的语法元素 1 和语法元素 2, 以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 1 的上下文索引, 所述当前块的语法元素 1 的上下文索引用于指示当前块的语法元素 1 的上下文模型;

或

根据所述当前块的左方相邻块的语法元素 1 和语法元素 2, 以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 2 的上下文索引, 所述当前块的语法元素 2 的上下文索引用于指示当前块的语法元素 2 的上下文模型。

- 5 结合第七方面, 在可能的实施方式中, 若所述预设的上下文模型集合中的上下文模型的数量为 3 个, 所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和;

或

- 10 所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和。

结合第七方面, 在可能的实施方式中, 若所述预设的上下文模型集合中的上下文模型的数量为 2 个, 所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果;

或

所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果。

- 20 结合第六方面或第七方面, 在可能的实施方式中, 所述当前块的语法元素 1 为 `affine_merge_flag`, 用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式; 或所述当前块的语法元素 2 为 `affine_inter_flag`, 用于在所述当前块所在条带为 P 型条带或者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 `AMVP` 模式;

或

- 25 所述当前块的语法元素 1 为 `subblock_merge_flag`, 用于指示当前块是否采用基于子块的 `merge` 模式; 或所述当前块的语法元素 2 为 `affine_inter_flag`, 用于在所述当前块所在条带为 P 型条带或者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 `AMVP` 模式。

第八方面, 提供了一种视频解码器, 包括: 熵解码单元, 用于解析接收的码流, 以获得当前块的待熵解码的语法元素, 所述当前块待熵解码的语法元素包括所述当前块的语法元素 3 或所述当前块的语法元素 4; 获取所述待熵解码的语法元素对应的上下文模型; 其中, 从预设的上下文模型集合中确定所述当前块的语法元素 3 的上下文模型; 或从预设的所述上下文模型集合中确定所述当前块的语法元素 4 的上下文模型; 基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码; 预测处理单元, 用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理, 以获得所

- 35 述当前块的预测块; 重构单元, 用于基于所述当前块的预测块获得所述当前块的重构图像。

结合第八方面，在可能的实施方式中，所述预设的上下文模型集合包括 5 个上下文模型。

结合第八方面，在可能的实施方式中，所述当前块的语法元素 3 为 merge_idx，用于指示所述当前块的 merge 候选列表的索引值；或所述当前块的语法元素 4 为 affine_merge_idx，
5 用于指示所述当前块的仿射 merge 候选列表的索引值；

或

所述当前块的语法元素 3 为 merge_idx，用于指示所述当前块的 merge 候选列表的索引值；或所述当前块的语法元素 4 为 subblock_merge_idx，用于指示针对子块 merge 候选列表的索引值。

10 第九方面，提供了一种编码方法，包括：获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；对所述当前块的待熵编码的语法元素进行熵编码；在对所述当前块待熵编码的语法元素进行熵编码时，使用预设的上下文模型完成所述当前块的语法元素 1 的熵编码；或使用所述上下文模型完成所述当前块的语法元素 2 的熵编码；输出包括所述熵编码后的所述当前块的语法
15 元素的码流。

其中，具体的语法元素和上下文模型可以参考第一方面。

第十方面，提供了一种编码方法，包括：获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前
20 前块的语法元素 1 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 2 的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出包括所述熵编码后的所述当前块的语法元素的码流。

其中，具体的语法元素和上下文模型可以参考第二方面。

第十一方面，提供了一种编码方法，包括：获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素 3 或所述当前块的语法元素 4；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素 3 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 4 的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出包括所述熵编码后的所述当前块的语法元素的码
30 流。

其中，具体的语法元素和上下文模型可以参考第三方面。

第十二方面，提供了一种视频编码器，包括：熵编码单元，用于获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；对所述当前块的待熵编码的语法元素进行熵编码；在对所述当前块待熵编
35 码的语法元素进行熵编码时，使用预设的上下文模型完成所述当前块的语法元素 1 的熵编码；或使用所述上下文模型完成所述当前块的语法元素 2 的熵编码；输出，用于输出包括

所述熵编码后的所述当前块的语法元素的码流。

其中，具体的语法元素和上下文模型可以参考第四方面。

第十三方面，提供了一种视频编码器，包括：熵编码单元，用于获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素 1 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 2 的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出，用于输出包括所述熵编码后的所述当前块的语法元素的码流。

10 其中，具体的语法元素和上下文模型可以参考第五方面。

第十四方面，提供了一种视频编码器，包括：熵编码单元，用于获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素 3 或所述当前块的语法元素 4；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素 3 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 4 的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出，用于输出包括所述熵编码后的所述当前块的语法元素的码流。

其中，具体的语法元素和上下文模型可以参考第六方面。

第十五方面，本发明涉及解码视频流的装置，包含处理器和存储器。所述存储器存储指令，所述指令使得所述处理器执行根据第一或第二或第三方面或第四方面或第一或第二或第三或第四方面任何可能实施例的方法。

第十六方面，本发明涉及解码视频流的装置，包含处理器和存储器。所述存储器存储指令，所述指令使得所述处理器执行根据第七或第八或第九方面或第七或第八或第九方面任何可能实施例的方法。

25 第十七方面，提出计算机可读存储介质，其上储存有指令，所述指令执行时，使得一个或多个处理器编码视频数据。所述指令使得所述一个或多个处理器执行根据第一或第二或第三或第四或第七或第八或第九方面或第一或第二或第三或第四或第七或第八或第九方面任何可能实施例的方法。

第十八方面，本发明涉及包括程序代码的计算机程序，所述程序代码在计算机上运行时执行根据第一或第二或第三或第四或第七或第八或第九方面或第一或第二或第三或第四或第七或第八或第九方面任何可能实施例的方法。

在附图及以下说明中阐述一个或多个实施例的细节。其它特征、目的和优点通过说明书、附图以及权利要求是显而易见的。

附图说明

35 为了更清楚地说明本申请实施例或背景技术中的技术方案，下面将对本申请实施例或

背景技术中所需要使用的附图进行说明。

图 1 是用于实现本发明实施例的视频编码系统实例的框图；

图 2 是示出用于实现本发明实施例的视频编码器实例结构的框图；

图 3 是示出用于实现本发明实施例的视频解码器实例结构的框图；

5 图 4 是绘示包含图 2 的编码器 20 和图 3 的解码器 30

图 5 是绘示另一种编码装置或解码装置实例的框图；

图 6 是绘示根据一实施例的当前块空域和时域候选运动信息的位置示意图；

图 7 是绘示根据一实施例的当前块和多个相邻位置块的位置示意图；

图 8A 是绘示根据一实施例的对构造的控制点运动矢量预测方法流程图；

10 图 8B 是绘示根据一实施例的构造的控制点运动矢量预测方法流程图；

图 9A 是绘示根据一实施例的帧间预测方法流程图；

图 9B 是绘示根据一实施例的利用构造的控制点运动矢量预测方法流程图；

图 9C 是绘示根据一实施例的运动补偿单元中心点的运动补偿单元位置示意图；

图 10 是绘示根据一实施例的视频解码方法的流程图；

15 图 11 是绘示根据一实施例的视频解码方法的流程图；

图 12 是绘示根据一实施例的视频解码方法的流程图。

以下如果没有关于相同参考符号的具体注释，相同的参考符号是指相同或至少功能上等效的特征。

具体实施方式

20 以下描述中，参考形成本公开一部分并以说明之方式示出本发明实施例的具体方面或可使用本发明实施例的具体方面的附图。应理解，本发明实施例可在其它方面中使用，并可包括附图中未描绘的结构或逻辑变化。因此，以下详细描述不应以限制性的意义来理解，且本发明的范围由所附权利要求书界定。

例如，应理解，结合所描述方法的揭示内容可以同样适用于用于执行所述方法的对应
25 设备或系统，且反之亦然。例如，如果描述一个或多个具体方法步骤，则对应的设备可以包含如功能单元等一个或多个单元，来执行所描述的一个或多个方法步骤（例如，一个单元执行一个或多个步骤，或多个单元，其中每个都执行多个步骤中的一个或多个），即使附图中未明确描述或说明这种一个或多个单元。另一方面，例如，如果基于如功能单元等一个或多个单元描述具体装置，则对应的方法可以包含一个步骤来执行一个或多个单元的功能性（例如，一个步骤执行一个或多个单元的功能性，或多个步骤，其中每个执行多个单元中一个或多个单元的功能性），即使附图中未明确描述或说明这种一个或多个步骤。进一步，应理解的是，除非另外明确提出，本文中所描述的各示例性实施例和/或方面的特征可以相互组合。

视频编码通常是指处理形成视频或视频序列的图片序列。在视频编码领域，术语“图片

(picture)”、“帧 (frame)”或“图像 (image)”可以用作同义词。本申请 (或本公开) 中使用的视频编码表示视频编码或视频解码。视频编码在源侧执行, 通常包括处理 (例如, 通过压缩) 原始视频图片以减少表示该视频图片所需的数据量 (从而更高效地存储和/或传输)。视频解码在目的地侧执行, 通常包括相对于编码器作逆处理, 以重构视频图片。实施
5 例涉及的视频图片 (或总称为图片, 下文将进行解释) “编码”应理解为涉及视频序列的“编码”或“解码”。编码部分和解码部分的组合也称为编解码 (编码和解码)。

无损视频编码情况下, 可以重构原始视频图片, 即经重构视频图片具有与原始视频图片相同的质量 (假设存储或传输期间没有传输损耗或其它数据丢失)。在有损视频编码情况下, 通过例如量化执行进一步压缩, 来减少表示视频图片所需的数据量, 而解码器侧无法
10 完全重构视频图片, 即经重构视频图片的质量相比原始视频图片的质量较低或较差。

H.261 的几个视频编码标准属于“有损混合型视频编解码” (即, 将样本域中的空间和时间预测与变换域中用于应用量化的 2D 变换编码结合)。视频序列的每个图片通常分割成不重叠的块集合, 通常在块层级上进行编码。换句话说, 编码器侧通常在块 (视频块) 层级处理亦即编码视频, 例如, 通过空间 (图片内) 预测和时间 (图片间) 预测来产生预测块,
15 从当前块 (当前处理或待处理的块) 减去预测块以获得残差块, 在变换域变换残差块并量化残差块, 以减少待传输 (压缩) 的数据量, 而解码器侧将相对于编码器的逆处理部分应用于经编码或经压缩块, 以重构用于表示的当前块。另外, 编码器复制解码器处理循环, 使得编码器和解码器生成相同的预测 (例如帧内预测和帧间预测) 和/或重构, 用于处理亦即编码后续块。

20 如本文中所用, 术语“块”可以为图片或帧的一部分。为便于描述, 参考多用途视频编码 (VVC: Versatile Video Coding) 或由 ITU-T 视频编码专家组 (Video Coding Experts Group, VCEG) 和 ISO/IEC 运动图像专家组 (Motion Picture Experts Group, MPEG) 的视频编码联合工作组 (Joint Collaboration Team on Video Coding, JCT-VC) 开发的高效视频编码 (High-Efficiency Video Coding, HEVC) 描述本发明实施例。本领域普通技术人员理解本发
25 明实施例不限于 HEVC 或 VVC。可以指 CU、PU 和 TU。在 HEVC 中, 通过使用表示为编码树的四叉树结构将 CTU 拆分为多个 CU。在 CU 层级处作出是否使用图片间 (时间) 或图片内 (空间) 预测对图片区域进行编码的决策。每个 CU 可以根据 PU 拆分类型进一步拆分为一个、两个或四个 PU。一个 PU 内应用相同的预测过程, 并在 PU 基础上将相关信息传输到解码器。在通过基于 PU 拆分类型应用预测过程获取残差块之后, 可以根据类似于
30 用于 CU 的编码树的其它四叉树结构将 CU 分割成变换单元 (transform unit, TU)。在视频压缩技术最新的发展中, 使用四叉树和二叉树 (Quad-tree and binary tree, QTBT) 分割帧来分割编码块。在 QTBT 块结构中, CU 可以为正方形或矩形形状。在 VVC 中, 编码树单元 (coding tree unit, CTU) 首先由四叉树结构分割。四叉树叶节点进一步由二进制树结构分割。二进制树叶节点称为编码单元 (coding unit, CU), 所述分段用于预测和变换处理,
35 无需其它任何分割。这表示 CU、PU 和 TU 在 QTBT 编码块结构中的块大小相同。同时, 还提出与 QTBT 块结构一起使用多重分割, 例如三叉树分割。

以下基于图 1 至 4 描述编码器 20、解码器 30 和编解码系统 10、40 的实施例 (在基

于图 10 更详细描述本发明实施例之前)。

图 1 为绘示示例性编码系统 10 的概念性或示意性框图,例如,可以利用本申请(本公开)技术的视频编码系统 10。视频编码系统 10 的编码器 20(例如,视频编码器 20)和解码器 30(例如,视频解码器 30)表示可用于根据本申请中描述的各种实例执行用于.....(分割/帧内预测/.....)的技术的设备实例。如图 1 中所示,编码系统 10 包括源设备 12,用于向例如解码经编码数据 13 的目的地设备 14 提供经编码数据 13,例如,经编码图片 13。

源设备 12 包括编码器 20,另外亦即可选地,可以包括图片源 16,例如图片预处理单元 18 的预处理单元 18,以及通信接口或通信单元 22。

图片源 16 可以包括或可以为任何类别的图片捕获设备,用于例如捕获现实世界图片,和/或任何类别的图片或评论(对于屏幕内容编码,屏幕上的一些文字也认为是待编码的图片或图像的一部分)生成设备,例如,用于生成计算机动画图片的计算机图形处理器,或用于获取和/或提供现实世界图片、计算机动画图片(例如,屏幕内容、虚拟现实(virtual reality, VR)图片)的任何类别设备,和/或其任何组合(例如,实景(augmented reality, AR)图片)。

15 (数字)图片为或者可以视为具有亮度值的采样点的二维阵列或矩阵。阵列中的采样点也可以称为像素(pixel)(像素(pixel element)的简称)或像素(pel)。阵列或图片在水平和垂直方向(或轴线)上的采样点数目定义图片的尺寸和/或分辨率。为了表示颜色,通常采用三个颜色分量,即图片可以表示为或包含三个采样阵列。RGB 格式或颜色空间中,图片包括对应的红色、绿色及蓝色采样阵列。但是,在视频编码中,每个像素通常以亮度/色度格式或颜色空间表示,例如, YCbCr, 包括 Y 指示的亮度分量(有时也可以用 L 指示)以及 Cb 和 Cr 指示的两个色度分量。亮度(简称为 luma)分量 Y 表示亮度或灰度水平强度(例如,在灰度等级图片中两者相同),而两个色度(简称为 chroma)分量 Cb 和 Cr 表示色度或颜色信息分量。相应地, YCbCr 格式的图片包括亮度采样值(Y)的亮度采样阵列,和色度值(Cb 和 Cr)的两个色度采样阵列。RGB 格式的图片可以转换或变换为 YCbCr 格式,反之亦然,该过程也称为色彩变换或转换。如果图片是黑白的,该图片可以只包括亮度采样阵列。

图片源 16(例如,视频源 16)可以为,例如用于捕获图片的相机,例如图片存储器的存储器,包括或存储先前捕获或产生的图片,和/或获取或接收图片的任何类别的(内部或外部)接口。相机可以为,例如,本地的或集成在源设备中的集成相机,存储器可为本地的或例如集成在源设备中的集成存储器。接口可以为,例如,从外部视频源接收图片的外部接口,外部视频源例如为外部图片捕获设备,比如相机、外部存储器或外部图片生成设备,外部图片生成设备例如为外部计算机图形处理器、计算机或服务器。接口可以为根据任何专有或标准化接口协议的任何类别的接口,例如有线或无线接口、光接口。获取图片数据 17 的接口可以是与通信接口 22 相同的接口或是通信接口 22 的一部分。

35 区别于预处理单元 18 和预处理单元 18 执行的处理,图片或图片数据 17(例如,视频数据 16)也可以称为原始图片或原始图片数据 17。

预处理单元 18 用于接收（原始）图片数据 17 并对图片数据 17 执行预处理，以获得经预处理的图片 19 或经预处理的图片数据 19。例如，预处理单元 18 执行的预处理可以包括整修、色彩格式转换（例如，从 RGB 转换为 YCbCr）、调色或去噪。可以理解，预处理单元 18 可以是可选组件。

- 5 编码器 20（例如，视频编码器 20）用于接收经预处理的图片数据 19 并提供经编码图片数据 21（下文将进一步描述细节，例如，基于图 2 或图 4）。在一个实例中，编码器 20 可以用于对图片进行编码。

源设备 12 的通信接口 22 可以用于接收经编码图片数据 21 并传输至其它设备，例如，目的地设备 14 或任何其它设备，以用于存储或直接重构，或用于在对应地存储经编码数据 10 13 和/或传输经编码数据 13 至其它设备之前处理经编码图片数据 21，其它设备例如为目的设备 14 或任何其它用于解码或存储的设备。

目的地设备 14 包括解码器 30（例如，视频解码器 30），另外亦即可选地，可以包括通信接口或通信单元 28、后处理单元 32 和显示设备 34。

目的地设备 14 的通信接口 28 用于例如，直接从源设备 12 或任何其它源接收经编码图片数据 15 21 或经编码数据 13，任何其它源例如为存储设备，存储设备例如为经编码图片数据存储设备。

通信接口 22 和通信接口 28 可以用于藉由源设备 12 和目的地设备 14 之间的直接通信链路或藉由任何类别的网络传输或接收经编码图片数据 21 或经编码数据 13，直接通信链路例如为直接有线或无线连接，任何类别的网络例如为有线或无线网络或其任何组合，或 20 任何类别的私网和公网，或其任何组合。

通信接口 22 可以例如用于将经编码图片数据 21 封装成合适的格式，例如包，以在通信链路或通信网络上传输。

形成通信接口 22 的对应部分的通信接口 28 可以例如用于解封装经编码数据 13，以获得经编码图片数据 21。

- 25 通信接口 22 和通信接口 28 都可以配置为单向通信接口，如图 1 中用于经编码图片数据 13 的从源设备 12 指向目的地设备 14 的箭头所指示，或配置为双向通信接口，以及可以用于例如发送和接收消息来建立连接、确认和交换任何其它与通信链路和/或例如经编码图片数据传输的数据传输有关的信息。

30 解码器 30 用于接收经编码图片数据 21 并提供经解码图片数据 31 或经解码图片 31（下文将进一步描述细节，例如，基于图 3 或图 5）。

目的地设备 14 的后处理器 32 用于后处理经解码图片数据 31（也称为经重构图片数据），例如，经解码图片 131，以获得经后处理图片数据 33，例如，经后处理图片 33。后处理单元 32 执行的后处理可以包括，例如，色彩格式转换（例如，从 YCbCr 转换为 RGB）、调色、整修或重采样，或任何其它处理，用于例如准备经解码图片数据 31 以由显示设备 34 显示。

- 35 目的地设备 14 的显示设备 34 用于接收经后处理图片数据 33 以向例如用户或观看者显示图片。显示设备 34 可以为或可以包括任何类别的用于呈现经重构图片的显示器，例如，

集成的或外部的显示器或监视器。例如，显示器可以包括液晶显示器（liquid crystal display, LCD）、有机发光二极管（organic light emitting diode, OLED）显示器、等离子显示器、投影仪、微LED显示器、硅基液晶（liquid crystal on silicon, LCoS）、数字光处理器（digital light processor, DLP）或任何类别的其它显示器。

5 虽然图1将源设备12和目的地设备14绘示为单独的设备，但设备实施例也可以同时包括源设备12和目的地设备14或同时包括两者的功能性，即源设备12或对应的功能性以及目的地设备14或对应的功能性。在此类实施例中，可以使用相同硬件和/或软件，或使用单独的硬件和/或软件，或其任何组合来实施源设备12或对应的功能性以及目的地设备14或对应的功能性。

10 本领域技术人员基于描述明显可知，不同单元的功能性或图1所示的源设备12和/或目的地设备14的功能性的存在和（准确）划分可能根据实际设备和应用有所不同。

编码器20（例如，视频编码器20）和解码器30（例如，视频解码器30）都可以实施为各种合适电路中的任一个，例如，一个或多个微处理器、数字信号处理器（digital signal processor, DSP）、专用集成电路（application-specific integrated circuit, ASIC）、现场可编程门阵列（field-programmable gate array, FPGA）、离散逻辑、硬件或其任何组合。如果部分地以软件实施所述技术，则设备可将软件的指令存储于合适的非暂时性计算机可读存储介质中，且可使用一或多个处理器以硬件执行指令从而执行本公开的技术。前述内容（包含硬件、软件、硬件与软件的组合等）中的任一者可视为一或多个处理器。视频编码器20和视频解码器30中的每一个可以包含在一或多个编码器或解码器中，所述编码器或解码器20中的任一个可以集成为对应设备中的组合编码器/解码器（编解码器）的一部分。

源设备12可称为视频编码设备或视频编码装置。目的地设备14可称为视频解码设备或视频解码装置。源设备12以及目的地设备14可以是视频编码设备或视频编码装置的实例。

源设备12和目的地设备14可以包括各种设备中的任一个，包含任何类别的手持或静止设备，例如，笔记本或膝上型计算机、移动电话、智能电话、平板或平板计算机、摄像机、台式计算机、机顶盒、电视、显示设备、数字媒体播放器、视频游戏控制台、视频流式传输设备（例如内容服务服务器或内容分发服务器）、广播接收器设备、广播发射器等，并可以不使用或使用任何类别的操作系统。

在一些情况下，源设备12和目的地设备14可以经装备以用于无线通信。因此，源设备12和目的地设备14可以为无线通信设备。

在一些情况下，图1中所示视频编码系统10仅为示例，本申请的技术可以适用于不必包含编码和解码设备之间的任何数据通信的视频编码设置（例如，视频编码或视频解码）。在其它实例中，数据可从本地存储器检索、在网络上流式传输等。视频编码设备可以对数据进行编码并且将数据存储到存储器，和/或视频解码设备可以从存储器检索数据并且对数据35 据进行解码。在一些实例中，由并不彼此通信而是仅编码数据到存储器和/或从存储器检索数据且解码数据的设备执行编码和解码。

应理解，对于以上参考视频编码器 20 所描述的实例中的每一个，视频解码器 30 可以用于执行相反过程。关于信令语法元素，视频解码器 30 可以用于接收并解析这种语法元素，相应地解码相关视频数据。在一些例子中，视频编码器 20 可以将一个或多个定义.....的语法元素熵编码成经编码视频比特流。在此类实例中，视频解码器 30 可以解析这种语法元素，并相应地解码相关视频数据。

编码器&编码方法

图 2 示出用于实现本申请（公开）技术的视频编码器 20 的实例的示意性/概念性框图。在图 2 的实例中，视频编码器 20 包括残差计算单元 204、变换处理单元 206、量化单元 208、逆量化单元 210、逆变换处理单元 212、重构单元 214、缓冲器 216、环路滤波器单元 220、经解码图片缓冲器（decoded picture buffer, DPB）230、预测处理单元 260 和熵编码单元 270。预测处理单元 260 可以包含帧间预测单元 244、帧内预测单元 254 和模式选择单元 262。帧间预测单元 244 可以包含运动估计单元和运动补偿单元（未图示）。图 2 所示的视频编码器 20 也可以称为混合型视频编码器或根据混合型视频编解码器的视频编码器。

例如，残差计算单元 204、变换处理单元 206、量化单元 208、预测处理单元 260 和熵编码单元 270 形成编码器 20 的前向信号路径，而例如逆量化单元 210、逆变换处理单元 212、重构单元 214、缓冲器 216、环路滤波器 220、经解码图片缓冲器（decoded picture buffer, DPB）230、预测处理单元 260 形成编码器的后向信号路径，其中编码器的后向信号路径对应于解码器的信号路径（参见图 3 中的解码器 30）。

编码器 20 通过例如输入 202，接收图片 201 或图片 201 的块 203，例如，形成视频或视频序列的图片序列中的图片。图片块 203 也可以称为当前图片块或待编码图片块，图片 201 可以称为当前图片或待编码图片（尤其是在视频编码中将当前图片与其它图片区分开时，其它图片例如同一视频序列亦即也包括当前图片的视频序列中的先前经编码和/或经解码图片）。

分割

编码器 20 的实施例可以包括分割单元（图 2 中未绘示），用于将图片 201 分割成多个例如块 203 的块，通常分割成多个不重叠的块。分割单元可以用于对视频序列中所有图片使用相同的块大小以及定义块大小的对应栅格，或用于在图片或子集或图片群组之间更改块大小，并将每个图片分割成对应的块。

在一个实例中，视频编码器 20 的预测处理单元 260 可以用于执行上述分割技术的任何组合。

如图片 201，块 203 也是或可以视为具有亮度值（采样值）的采样点的二维阵列或矩阵，虽然其尺寸比图片 201 小。换句话说，块 203 可以包括，例如，一个采样阵列（例如黑白图片 201 情况下的亮度阵列）或三个采样阵列（例如，彩色图片情况下的一个亮度阵列和两个色度阵列）或依据所应用的色彩格式的任何其它数目和/或类别的阵列。块 203 的水平和垂直方向（或轴线）上采样点的数目定义块 203 的尺寸。

如图 2 所示的编码器 20 用于逐块编码图片 201，例如，对每个块 203 执行编码和预测。

残差计算

残差计算单元 204 用于基于图片块 203 和预测块 265（下文提供预测块 265 的其它细节）计算残差块 205，例如，通过逐样本（逐像素）将图片块 203 的样本值减去预测块 265 的样本值，以在样本域中获取残差块 205。

5 变换

变换处理单元 206 用于在残差块 205 的样本值上应用例如离散余弦变换（discrete cosine transform, DCT）或离散正弦变换（discrete sine transform, DST）的变换，以在变换域中获取变换系数 207。变换系数 207 也可以称为变换残差系数，并在变换域中表示残差块 205。

- 变换处理单元 206 可以用于应用 DCT/DST 的整数近似值，例如为 HEVC/H.265 指定的
10 变换。与正交 DCT 变换相比，这种整数近似值通常由某一因子按比例缩放。为了维持经正变换和逆变换处理的残差块的范数，应用额外比例缩放因子作为变换过程的一部分。比例缩放因子通常是基于某些约束条件选择的，例如，比例缩放因子是用于移位运算的 2 的幂、变换系数的位深度、准确性和实施成本之间的权衡等。例如，在解码器 30 侧通过例如逆变换处理单元 212 为逆变换（以及在编码器 20 侧通过例如逆变换处理单元 212 为对应逆变换）
15 指定具体比例缩放因子，以及相应地，可以在编码器 20 侧通过变换处理单元 206 为正变换指定对应比例缩放因子。

量化

- 量化单元 208 用于例如通过应用标量量化或向量量化来量化变换系数 207，以获得经量化变换系数 209。经量化变换系数 209 也可以称为经量化残差系数 209。量化过程可以减
20 少与部分或全部变换系数 207 有关的位深度。例如，可在量化期间将 n 位变换系数向下舍入到 m 位变换系数，其中 n 大于 m。可通过调整量化参数（quantization parameter, QP）修改量化程度。例如，对于标量量化，可以应用不同的标度来实现较细或较粗的量化。较小量化步长对应较细量化，而较大量化步长对应较粗量化。可以通过量化参数（quantization parameter, QP）指示合适的量化步长。例如，量化参数可以为合适的量化步长的预定义集
25 合的索引。例如，较小的量化参数可以对应精细量化（较小量化步长），较大量化参数可以对应粗糙量化（较大量化步长），反之亦然。量化可以包含除以量化步长以及例如通过逆量化 210 执行的对应的量化或逆量化，或者可以包含乘以量化步长。根据例如 HEVC 的一些标准的实施例可以使用量化参数来确定量化步长。一般而言，可以基于量化参数使用包含除法的等式的定点近似来计算量化步长。可以引入额外比例缩放因子来进行量化和反量化，
30 以恢复可能由于在用于量化步长和量化参数的等式的定点近似中使用的标度而修改的残差块的范数。在一个实例实施方式中，可以合并逆变换和反量化的标度。或者，可以使用自定义量化表并在例如比特流中将其从编码器通过信号发送到解码器。量化是有损操作，其中量化步长越大，损耗越大。

- 逆量化单元 210 用于在经量化系数上应用量化单元 208 的逆量化，以获得经反量化系
35 数 211，例如，基于或使用与量化单元 208 相同的量化步长，应用量化单元 208 应用的量化方案的逆量化方案。经反量化系数 211 也可以称为经反量化残差系数 211，对应于变换系数

207, 虽然由于量化造成的损耗通常与变换系数不相同。

逆变换处理单元 212 用于应用变换处理单元 206 应用的变换的逆变换, 例如, 逆离散余弦变换 (discrete cosine transform, DCT) 或逆离散正弦变换 (discrete sine transform, DST), 以在样本域中获取逆变换块 213。逆变换块 213 也可以称为逆变换经反量化块 213 或逆变换残差块 213。

重构单元 214 (例如, 求和器 214) 用于将逆变换块 213 (即经重构残差块 213) 添加至预测块 265, 以在样本域中获取经重构块 215, 例如, 将经重构残差块 213 的样本值与预测块 265 的样本值相加。

可选地, 例如线缓冲器 216 的缓冲器单元 216 (或简称“缓冲器”216) 用于缓冲或存储经重构块 215 和对应的样本值, 用于例如帧内预测。在其它的实施例中, 编码器可以用于使用存储在缓冲器单元 216 中的未经滤波的经重构块和/或对应的样本值来进行任何类别的估计和/或预测, 例如帧内预测。

例如, 编码器 20 的实施例可以经配置以使得缓冲器单元 216 不只用于存储用于帧内预测 254 的经重构块 215, 也用于环路滤波器单元 220 (在图 2 中未示出), 和/或, 例如使得缓冲器单元 216 和经解码图片缓冲器单元 230 形成一个缓冲器。其它实施例可以用于将经滤波块 221 和/或来自经解码图片缓冲器 230 的块或样本 (图 2 中均未示出) 用作帧内预测 254 的输入或基础。

环路滤波器单元 220 (或简称“环路滤波器”220) 用于对经重构块 215 进行滤波以获得经滤波块 221, 从而顺利进行像素转变或提高视频质量。环路滤波器单元 220 旨在表示一个或多个环路滤波器, 例如去块滤波器、样本自适应偏移 (sample-adaptive offset, SAO) 滤波器或其它滤波器, 例如双边滤波器、自适应环路滤波器 (adaptive loop filter, ALF), 或锐化或平滑滤波器, 或协同滤波器。尽管环路滤波器单元 220 在图 2 中示出为环内滤波器, 但在其它配置中, 环路滤波器单元 220 可实施为环后滤波器。经滤波块 221 也可以称为经滤波的经重构块 221。经解码图片缓冲器 230 可以在环路滤波器单元 220 对经重构编码块执行滤波操作之后存储经重构编码块。

编码器 20 (对应地, 环路滤波器单元 220) 的实施例可以用于输出环路滤波器参数 (例如, 样本自适应偏移信息), 例如, 直接输出或由熵编码单元 270 或任何其它熵编码单元熵编码后输出, 例如使得解码器 30 可以接收并应用相同的环路滤波器参数用于解码。

经解码图片缓冲器 (decoded picture buffer, DPB) 230 可以为存储参考图片数据供视频编码器 20 编码视频数据之用的参考图片存储器。DPB 230 可由多种存储器设备中的任何一个形成, 例如动态随机存储器 (dynamic random access memory, DRAM) (包含同步 DRAM (synchronous DRAM, SDRAM)、磁阻式 RAM (magnetoresistive RAM, MRAM)、电阻式 RAM (resistive RAM, RRAM)) 或其它类型的存储器设备。可以由同一存储器设备或单独的存储器设备提供 DPB 230 和缓冲器 216。在某一实例中, 经解码图片缓冲器 (decoded picture buffer, DPB) 230 用于存储经滤波块 221。经解码图片缓冲器 230 可以进一步用于存储同一当前图片或例如先前经重构图片的不同图片的其它先前的经滤波块, 例如先前经

重构和经滤波块 221, 以及可以提供完整的先前经重构亦即经解码图片 (和对应参考块和样本) 和/或部分经重构当前图片 (和对应参考块和样本), 例如用于帧间预测。在某一实例中, 如果经重构块 215 无需环内滤波而得以重构, 则经解码图片缓冲器 (decoded picture buffer, DPB) 230 用于存储经重构块 215。

5 预测处理单元 260, 也称为块预测处理单元 260, 用于接收或获取块 203 (当前图片 201 的当前块 203) 和经重构图片数据, 例如来自缓冲器 216 的同一 (当前) 图片的参考样本和/或来自经解码图片缓冲器 230 的一个或多个先前经解码图片的参考图片数据 231, 以及用于处理这类数据进行预测, 即提供可以为经帧间预测块 245 或经帧内预测块 255 的预测块 265。

10 模式选择单元 262 可以用于选择预测模式 (例如帧内或帧间预测模式) 和/或对应的用作预测块 265 的预测块 245 或 255, 以计算残差块 205 和重构经重构块 215。

模式选择单元 262 的实施例可以用于选择预测模式 (例如, 从预测处理单元 260 所支持的那些预测模式中选择), 所述预测模式提供最佳匹配或者说最小残差 (最小残差意味着传输或存储中更好的压缩), 或提供最小信令开销 (最小信令开销意味着传输或存储中更好
15 的压缩), 或同时考虑或平衡以上两者。模式选择单元 262 可以用于基于码率失真优化 (rate distortion optimization, RDO) 确定预测模式, 即选择提供最小码率失真优化的预测模式, 或选择相关码率失真至少满足预测模式选择标准的预测模式。

下文将详细解释编码器 20 的实例 (例如, 通过预测处理单元 260) 执行的预测处理和 (例如, 通过模式选择单元 262) 执行的模式选择。

20 如上文所述, 编码器 20 用于从 (预先确定的) 预测模式集合中确定或选择最好或最优的预测模式。预测模式集合可以包括例如帧内预测模式和/或帧间预测模式。

帧内预测模式集合可以包括 35 种不同的帧内预测模式, 例如, 如 DC (或均值) 模式和平面模式的非方向性模式, 或如 H.265 中定义的方向性模式, 或者可以包括 67 种不同的帧内预测模式, 例如, 如 DC (或均值) 模式和平面模式的非方向性模式, 或如正在发展中
25 的 H.266 中定义的方向性模式。

(可能的) 帧间预测模式集合取决于可用参考图片 (即, 例如前述存储在 DPB 230 中的至少部分经解码图片) 和其它帧间预测参数, 例如取决于是否使用整个参考图片或只使用参考图片的一部分, 例如围绕当前块的区域的搜索窗区域, 来搜索最佳匹配参考块, 和/或例如取决于是否应用如半像素和/或四分之一像素内插的像素内插。

30 除了以上预测模式, 也可以应用跳过模式和/或直接模式。

预测处理单元 260 可以进一步用于将块 203 分割成较小的块分区或子块, 例如, 通过迭代使用四叉树 (quad-tree, QT) 分割、二叉树 (binary-tree, BT) 分割或三叉树 (triple-tree or ternary-tree, TT) 分割, 或其任何组合, 以及用于例如为块分区或子块中的每一个执行
35 预测, 其中模式选择包括选择分割的块 203 的树结构和选择应用于块分区或子块中的每一个的预测模式。

帧间预测单元 244 可以包含运动估计 (motion estimation, ME) 单元 (图 2 中未示出)

和运动补偿 (motion compensation, MC) 单元 (图 2 中未示出)。运动估计单元用于接收或获取图片块 203 (当前图片 201 的当前图片块 203) 和经解码图片 231, 或至少一个或多个先前经重构块, 例如, 一个或多个其它/不同先前经解码图片 231 的经重构块, 来进行运动估计。例如, 视频序列可以包括当前图片和先前经解码图片 31, 或换句话说, 当前图片和先前经解码图片 31 可以是形成视频序列的图片序列的一部分, 或者形成该图片序列。

例如, 编码器 20 可以用于从多个其它图片中的同一或不同图片的多个参考块中选择参考块, 并向运动估计单元 (图 2 中未示出) 提供参考图片 (或参考图片索引) 和/或提供参考块的位置 (X、Y 坐标) 与当前块的位置之间的偏移 (空间偏移) 作为帧间预测参数。该偏移也称为运动向量 (motion vector, MV)。

运动补偿单元用于获取, 例如接收帧间预测参数, 并基于或使用帧间预测参数执行帧间预测来获取帧间预测块 245。由运动补偿单元 (图 2 中未示出) 执行的运动补偿可以包含基于通过运动估计 (可能执行对子像素精确度的内插) 确定的运动/块向量取出或生成预测块。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编码图片块的候选预测块的数目。一旦接收到用于当前图片块的 PU 的运动向量, 运动补偿单元 246 可以在一个参考图片列表中定位运动向量指向的预测块。运动补偿单元 246 还可以生成与块和视频条带相关联的语法元素, 以供视频解码器 30 在解码视频条带的图片块时使用。

帧内预测单元 254 用于获取, 例如接收同一图片的图片块 203 (当前图片块) 和一个或多个先前经重构块, 例如经重构相邻块, 以进行帧内估计。例如, 编码器 20 可以用于从多个 (预定) 帧内预测模式中选择帧内预测模式。

编码器 20 的实施例可以用于基于优化标准选择帧内预测模式, 例如基于最小残差 (例如, 提供最类似于当前图片块 203 的预测块 255 的帧内预测模式) 或最小码率失真。

帧内预测单元 254 进一步用于基于如所选择的帧内预测模式的帧内预测参数确定帧内预测块 255。在任何情况下, 在选择用于块的帧内预测模式之后, 帧内预测单元 254 还用于向熵编码单元 270 提供帧内预测参数, 即提供指示所选择的用于块的帧内预测模式的信息。在一个实例中, 帧内预测单元 254 可以用于执行下文描述的帧内预测技术的任意组合。

熵编码单元 270 用于将熵编码算法或方案 (例如, 可变长度编码 (variable length coding, VLC) 方案、上下文自适应 VLC (context adaptive VLC, CAVLC) 方案、算术编码方案、上下文自适应二进制算术编码 (context adaptive binary arithmetic coding, CABAC)、基于语法的上下文自适应二进制算术编码 (syntax-based context-adaptive binary arithmetic coding, SBAC)、概率区间分割熵 (probability interval partitioning entropy, PIPE) 编码或其它熵编码方法或技术) 应用于经量化残差系数 209、帧间预测参数、帧内预测参数和/或环路滤波器参数中的单个或所有上 (或不应用), 以获得可以通过输出 272 以例如经编码比特流 21 的形式输出的经编码图片数据 21。可以将经编码比特流传输到视频解码器 30, 或将其存档稍后由视频解码器 30 传输或检索。熵编码单元 270 还可用于熵编码正被编码的当前视频条带的其它语法元素。

视频编码器 20 的其它结构变型可用于编码视频流。例如, 基于非变换的编码器 20 可

以在没有针对某些块或帧的变换处理单元 206 的情况下直接量化残差信号。在另一实施方式中，编码器 20 可具有组合成单个单元的量化单元 208 和逆量化单元 210。

图 3 示出示例性视频解码器 30，用于实现本申请的技术。视频解码器 30 用于接收例如由编码器 20 编码的经编码图片数据（例如，经编码比特流）21，以获得经解码图片 231。

5 在解码过程期间，视频解码器 30 从视频编码器 20 接收视频数据，例如表示经编码视频条带的图片块的经编码视频比特流及相关联的语法元素。

在图 3 的实例中，解码器 30 包括熵解码单元 304、逆量化单元 310、逆变换处理单元 312、重构单元 314（例如求和器 314）、缓冲器 316、环路滤波器 320、经解码图片缓冲器 330 以及预测处理单元 360。预测处理单元 360 可以包含帧间预测单元 344、帧内预测单元 354 和模式选择单元 362。在一些实例中，视频解码器 30 可执行大体上与参照图 2 的视频编码器 20 描述的编码遍次互逆的解码遍次。

熵解码单元 304 用于对经编码图片数据 21 执行熵解码，以获得例如经量化系数 309 和/或经解码的编码参数（图 3 中未示出），例如，帧间预测、帧内预测参数、环路滤波器参数和/或其它语法元素中（经解码）的任意一个或全部。熵解码单元 304 进一步用于将帧间 15 预测参数、帧内预测参数和/或其它语法元素转发至预测处理单元 360。视频解码器 30 可接收视频条带层级和/或视频块层级的语法元素。

逆量化单元 310 功能上可与逆量化单元 110 相同，逆变换处理单元 312 功能上可与逆变换处理单元 212 相同，重构单元 314 功能上可与重构单元 214 相同，缓冲器 316 功能上可与缓冲器 216 相同，环路滤波器 320 功能上可与环路滤波器 220 相同，经解码图片缓冲器 20 器 330 功能上可与经解码图片缓冲器 230 相同。

预测处理单元 360 可以包括帧间预测单元 344 和帧内预测单元 354，其中帧间预测单元 344 功能上可以类似于帧间预测单元 244，帧内预测单元 354 功能上可以类似于帧内预测单元 254。预测处理单元 360 通常用于执行块预测和/或从经编码数据 21 获取预测块 365，以及从例如熵解码单元 304（显式地或隐式地）接收或获取预测相关参数和/或关于所选择 25 的预测模式的信息。

当视频条带经编码为经帧内编码（I）条带时，预测处理单元 360 的帧内预测单元 354 用于基于信号表示的帧内预测模式及来自当前帧或图片的先前经解码块的数据来产生用于当前视频条带的图片块的预测块 365。当视频帧经编码为经帧间编码（即 B 或 P）条带时，预测处理单元 360 的帧间预测单元 344（例如，运动补偿单元）用于基于运动向量及从熵 30 解码单元 304 接收的其它语法元素生成用于当前视频条带的视频块的预测块 365。对于帧间预测，可从一个参考图片列表内的一个参考图片中产生预测块。视频解码器 30 可基于存储于 DPB 330 中的参考图片，使用默认建构技术来建构参考帧列表：列表 0 和列表 1。

预测处理单元 360 用于通过解析运动向量和其它语法元素，确定用于当前视频条带的视频块的预测信息，并使用预测信息产生用于正经解码的当前视频块的预测块。例如，预 35 测处理单元 360 使用接收到的一些语法元素确定用于编码视频条带的视频块的预测模式（例如，帧内或帧间预测）、帧间预测条带类型（例如，B 条带、P 条带或 GPB 条带）、用于

条带的参考图片列表中的一个或多个的建构信息、用于条带的每个经帧间编码视频块的运动向量、条带的每个经帧间编码视频块的帧间预测状态以及其它信息，以解码当前视频条带的视频块。

逆量化单元 310 可用于逆量化（即，反量化）在比特流中提供且由熵解码单元 304 解
5 码的经量化变换系数。逆量化过程可包含使用由视频编码器 20 针对视频条带中的每一视频块所计算的量化参数来确定应该应用的量化程度并同样确定应该应用的逆量化程度。

逆变换处理单元 312 用于将逆变换（例如，逆 DCT、逆整数变换或概念上类似的逆变换过程）应用于变换系数，以便在像素域中产生残差块。

重构单元 314（例如，求和器 314）用于将逆变换块 313（即经重构残差块 313）添加
10 到预测块 365，以在样本域中获取经重构块 315，例如通过将经重构残差块 313 的样本值与预测块 365 的样本值相加。

环路滤波器单元 320（在编码循环期间或在编码循环之后）用于对经重构块 315 进行滤波以获得经滤波块 321，从而顺利进行像素转变或提高视频质量。在一个实例中，环路滤波器单元 320 可以用于执行下文描述的滤波技术的任意组合。环路滤波器单元 320 旨在
15 表示一个或多个环路滤波器，例如去块滤波器、样本自适应偏移（sample-adaptive offset, SAO）滤波器或其它滤波器，例如双边滤波器、自适应环路滤波器（adaptive loop filter, ALF），或锐化或平滑滤波器，或协同滤波器。尽管环路滤波器单元 320 在图 3 中示出为环内滤波器，但在其它配置中，环路滤波器单元 320 可实施为环后滤波器。

随后将给定帧或图片中的经解码视频块 321 存储在存储用于后续运动补偿的参考图片
20 的经解码图片缓冲器 330 中。

解码器 30 用于例如，藉由输出 332 输出经解码图片 31，以向用户呈现或供用户查看。

视频解码器 30 的其它变型可用于对压缩的比特流进行解码。例如，解码器 30 可以在没有环路滤波器单元 320 的情况下生成输出视频流。例如，基于非变换的解码器 30 可以在没有针对某些块或帧的逆变换处理单元 312 的情况下直接逆量化残差信号。在另一实施方式
25 中，视频解码器 30 可以具有组合成单个单元的逆量化单元 310 和逆变换处理单元 312。

图 4 是根据一示例性实施例的包含图 2 的编码器 20 和/或图 3 的解码器 30 的视频编码系统 40 的实例的说明图。系统 40 可以实现本申请的各种技术的组合。在所说明的实施方式中，视频编码系统 40 可以包含成像设备 41、视频编码器 20、视频解码器 30（和/或藉由处理单元 46 的逻辑电路 47 实施的视频编码器）、天线 42、一个或多个处理器 43、一个或
30 多个存储器 44 和/或显示设备 45。

如图所示，成像设备 41、天线 42、处理单元 46、逻辑电路 47、视频编码器 20、视频解码器 30、处理器 43、存储器 44 和/或显示设备 45 能够互相通信。如所论述，虽然用视频编码器 20 和视频解码器 30 绘示视频编码系统 40，但在不同实例中，视频编码系统 40 可以只包含视频编码器 20 或只包含视频解码器 30。

35 在一些实例中，如图所示，视频编码系统 40 可以包含天线 42。例如，天线 42 可以用于传输或接收视频数据的经编码比特流。另外，在一些实例中，视频编码系统 40 可以包含

显示设备 45。显示设备 45 可以用于呈现视频数据。在一些实例中，如图所示，逻辑电路 47 可以通过处理单元 46 实施。处理单元 46 可以包含专用集成电路（application-specific integrated circuit, ASIC）逻辑、图形处理器、通用处理器等。视频编码系统 40 也可以包含可选处理器 43，该可选处理器 43 类似地可以包含专用集成电路（application-specific integrated circuit, ASIC）逻辑、图形处理器、通用处理器等。在一些实例中，逻辑电路 47 可以通过硬件实施，如视频编码专用硬件等，处理器 43 可以通过通用软件、操作系统等实施。另外，存储器 44 可以是任何类型的存储器，例如易失性存储器（例如，静态随机存取存储器（Static Random Access Memory, SRAM）、动态随机存储器（Dynamic Random Access Memory, DRAM）等）或非易失性存储器（例如，闪存等）等。在非限制性实例中，存储器 44 可以由超速缓存内存实施。在一些实例中，逻辑电路 47 可以访问存储器 44（例如用于实施图像缓冲器）。在其它实例中，逻辑电路 47 和/或处理单元 46 可以包含存储器（例如，缓存等）用于实施图像缓冲器等。

在一些实例中，通过逻辑电路实施的视频编码器 20 可以包含（例如，通过处理单元 46 或存储器 44 实施的）图像缓冲器和（例如，通过处理单元 46 实施的）图形处理单元。图形处理单元可以通信耦合至图像缓冲器。图形处理单元可以包含通过逻辑电路 47 实施的视频编码器 20，以实施参照图 2 和/或本文中所描述的任何其它编码器系统或子系统所论述的各种模块。逻辑电路可以用于执行本文所论述的各种操作。

视频解码器 30 可以以类似方式通过逻辑电路 47 实施，以实施参照图 3 的解码器 30 和/或本文中所描述的任何其它解码器系统或子系统所论述的各种模块。在一些实例中，逻辑电路 47 实施的视频解码器 30 可以包含（通过处理单元 2820 或存储器 44 实施的）图像缓冲器和（例如，通过处理单元 46 实施的）图形处理单元。图形处理单元可以通信耦合至图像缓冲器。图形处理单元可以包含通过逻辑电路 47 实施的视频解码器 30，以实施参照图 3 和/或本文中所描述的任何其它解码器系统或子系统所论述的各种模块。

在一些实例中，视频编码系统 40 的天线 42 可以用于接收视频数据的经编码比特流。如所论述，经编码比特流可以包含本文所论述的与编码视频帧相关的数据、指示符、索引值、模式选择数据等，例如与编码分割相关的数据（例如，变换系数或经量化变换系数，（如所论述的）可选指示符，和/或定义编码分割的数据）。视频编码系统 40 还可包含耦合至天线 42 并用于解码经编码比特流的视频解码器 30。显示设备 45 用于呈现视频帧。

图 5 是根据一示例性实施例的可用作图 1 中的源设备 12 和目的地设备 14 中的任一个或两个的装置 500 的简化框图。装置 500 可以实现本申请的技术，装置 500 可以采用包含多个计算设备的计算系统的形式，或采用例如移动电话、平板计算机、膝上型计算机、笔记本电脑、台式计算机等单个计算设备的形式。

装置 500 中的处理器 502 可以为中央处理器。或者，处理器 502 可以为现有的或今后将研发出的能够操控或处理信息的任何其它类型的设备或多个设备。如图所示，虽然可以使用例如处理器 502 的单个处理器实践所揭示的实施方式，但是使用一个以上处理器可以实现速度和效率方面的优势。

在一实施方式中，装置 500 中的存储器 504 可以为只读存储器（Read Only Memory，ROM）设备或随机存取存储器（random access memory，RAM）设备。任何其他合适类型的存储设备都可以用作存储器 504。存储器 504 可以包括代码和由处理器 502 使用总线 512 访问的数据 506。存储器 504 可进一步包括操作系统 508 和应用程序 510，应用程序 510 包含至少一个准许处理器 502 执行本文所描述的方法的程序。例如，应用程序 510 可以包括应用 1 到 N，应用 1 到 N 进一步包括执行本文所描述的方法的视频编码应用。装置 500 还可包含采用从存储器 514 形式的附加存储器，该从存储器 514 例如可以为与移动计算设备一起使用的存储卡。因为视频通信会话可能含有大量信息，这些信息可以整体或部分存储在从存储器 514 中，并按需要加载到存储器 504 用于处理。

10 装置 500 还可包含一或多个输出设备，例如显示器 518。在一个实例中，显示器 518 可以为将显示器和可操作以感测触摸输入的触敏元件组合的触敏显示器。显示器 518 可以通过总线 512 耦合于处理器 502。除了显示器 518 还可以提供其它准许用户对装置 500 编程或以其它方式使用装置 500 的输出设备，或提供其它输出设备作为显示器 518 的替代方案。当输出设备是显示器或包含显示器时，显示器可以以不同方式实现，包含通过液晶显示器（liquid crystal display，LCD）、阴极射线管（cathode-ray tube，CRT）显示器、等离子显示器或发光二极管（light emitting diode，LED）显示器，如有机 LED（organic LED，OLED）显示器。

装置 500 还可包含图像感测设备 520 或与其连通，图像感测设备 520 例如为相机或为现有的或今后将研发出的可以感测图像的任何其它图像感测设备 520，所述图像例如为运行装置 500 的用户的图像。图像感测设备 520 可以放置为直接面向运行装置 500 的用户。在一实例中，可以配置图像感测设备 520 的位置和光轴以使其视野包含紧邻显示器 518 的区域且从该区域可见显示器 518。

装置 500 还可包含声音感测设备 522 或与其连通，声音感测设备 522 例如为麦克风或为现有的或今后将研发出的可以感测装置 500 附近的声音的任何其它声音感测设备。声音感测设备 522 可以放置为直接面向运行装置 500 的用户，并可以用于接收用户在运行装置 500 时发出的声音，例如语音或其它发声。

虽然图 5 中将装置 500 的处理器 502 和存储器 504 绘示为集成在单个单元中，但是还可以使用其它配置。处理器 502 的运行可以分布在多个可直接耦合的机器中（每个机器具有一个或多个处理器），或分布在本地区域或其它网络中。存储器 504 可以分布在多个机器中，例如基于网络的存储器或多个运行装置 500 的机器中的存储器。虽然此处只绘示单个总线，但装置 500 的总线 512 可以由多个总线形成。进一步地，从存储器 514 可以直接耦合至装置 500 的其它组件或可以通过网络访问，并且可包括单个集成单元，例如一个存储卡，或多个单元，例如多个存储卡。因此，可以以多种配置实施装置 500。

下面对本申请中涉及到的概念进行描述。

35 1) 帧间预测模式

在HEVC中，使用两种帧间预测模式，分别为先进的运动矢量预测（advanced motion

vector prediction, AMVP) 模式和融合 (merge) 模式。

对于AMVP模式, 先遍历当前块空域或者时域相邻的已编码块 (记为邻块), 根据各个邻块的运动信息构建候选运动矢量列表 (也可以称为运动信息候选列表), 然后通过率失真代价从候选运动矢量列表中确定最优的运动矢量, 将率失真代价最小的候选运动信息作为当前块的运动矢量预测值 (motion vector predictor, MVP)。其中, 邻块的位置及其遍历顺序都是预先定义好的。率失真代价由公式(1)计算获得, 其中, J 表示率失真代价RD Cost, SAD为使用候选运动矢量预测值进行运动估计后得到的预测像素值与原始像素值之间的绝对误差和 (sum of absolute differences, SAD), R 表示码率, λ 表示拉格朗日乘子。编码端将选择的运动矢量预测值在候选运动矢量列表中的索引值和参考帧索引值传递到解码端。进一步地, 在MVP为中心的邻域内进行运动搜索获得当前块实际的运动矢量, 编码端将MVP与实际运动矢量之间的差值 (motion vector difference) 传递到解码端。

$$J = SAD + \lambda R \quad (1)$$

对于Merge模式, 先通过当前块空域或者时域相邻的已编码块的运动信息, 构建候选运动矢量列表, 然后通过计算率失真代价从候选运动矢量列表中确定最优的运动信息作为当前块的运动信息, 再将最优的运动信息在候选运动矢量列表中位置的索引值 (记为merge index, 下同) 传递到解码端。当前块空域和时域候选运动信息如图6所示, 空域候选运动信息来自于空间相邻的5个块 (A0, A1, B0, B1和B2), 若相邻块不可得 (相邻块不存在或者相邻块未编码或者相邻块采用的预测模式不为帧间预测模式), 则该相邻块的运动信息不加入候选运动矢量列表。当前块的时域候选运动信息根据参考帧和当前帧的图序计数 (picture order count, POC) 对参考帧中对应位置块的MV进行缩放后获得。首先判断参考帧中T位置的块是否可得, 若不可得则选择C位置的块。

与 AMVP 模式类似, Merge 模式的邻块的位置及其遍历顺序也是预先定义好的, 且邻块的位置及其遍历顺序在不同模式下可能不同。

可以看到, 在 AMVP 模式和 Merge 模式中, 都需要维护一个候选运动矢量列表。每次向候选列表中加入新的运动信息之前都会先检查列表中是否已经存在相同的运动信息, 如果存在则不会将该运动信息加入列表中。我们将这个检查过程称为候选运动矢量列表的修剪。列表修剪是为了防止列表中出现相同的运动信息, 避免冗余的率失真代价计算。

在 HEVC 的帧间预测中, 编码块内的所有像素都采用了相同的运动信息, 然后根据运动信息进行运动补偿, 得到编码块的像素的预测值。然而在编码块内, 并不是所有的像素都有相同的运动特性, 采用相同的运动信息可能会导致运动补偿预测的不准确, 进而增加了残差信息。

现有的视频编码标准使用基于平动运动模型的块匹配运动估计, 并且假设块中所有像素点的运动一致。但是由于在现实世界中, 运动多种多样, 存在很多非平动运动的物体, 如旋转的物体, 在不同方向旋转的过山车, 投放的烟花和电影中的一些特技动作, 特别是在 UGC 场景中的运动物体, 对它们的编码, 如果采用当前编码标准中的基于平动运动模型的块运动补偿技术, 编码效率会受到很大的影响, 因此, 产生了非平动运动模型, 比如

仿射运动模型，以便进一步提高编码效率。

基于此，根据运动模型的不同，AMVP 模式可以分为基于平动模型的 AMVP 模式以及基于非平动模型的 AMVP 模式；Merge 模式可以分为基于平动模型的 Merge 模式和基于非平动运动模型的 Merge 模式。

5 2) 非平动运动模型

非平动运动模型预测指在编解码端使用相同的运动模型推导出当前块内每一个子运动补偿单元的运动信息，根据子运动补偿单元的运动信息进行运动补偿，得到预测块，从而提高预测效率。常用的非平动运动模型有 4 参数仿射运动模型或者 6 参数仿射运动模型。

其中，本申请实施例中涉及到的子运动补偿单元可以是一个像素点或按照特定方法划分的大小为 $N_1 \times N_2$ 的像素块，其中， N_1 和 N_2 均为正整数， N_1 可以等于 N_2 ，也可以不等于 N_2 。

4 参数仿射运动模型如公式 (2) 所示：

$$\begin{cases} vx = a_1 + a_3x + a_4y \\ vy = a_2 - a_4x + a_3y \end{cases} \quad (2)$$

4 参数仿射运动模型可以通过两个像素点的运动矢量及其相对于当前块左上顶点像素的坐标来表示，将用于表示运动模型参数的像素点称为控制点。若采用左上顶点(0,0)和右上顶点(W,0)像素点作为控制点，则先确定当前块左上顶点和右上顶点控制点的运动矢量 (vx_0, vy_0) 和 (vx_1, vy_1) ，然后根据公式 (3) 得到当前块中每一个子运动补偿单元的运动信息，其中 (x, y) 为子运动补偿单元相对于当前块左上顶点像素的坐标，W 为当前块的宽。

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W}x - \frac{vy_1 - vy_0}{W}y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W}x + \frac{vx_1 - vx_0}{W}y + vy_0 \end{cases} \quad (3)$$

6 参数仿射运动模型如公式 (4) 所示：

$$\begin{cases} vx = a_1 + a_3x + a_4y \\ vy = a_2 + a_5x + a_6y \end{cases} \quad (4)$$

6 参数仿射运动模型可以通过三个像素点的运动矢量及其相对于当前块左上顶点像素的坐标来表示。若采用左上顶点(0, 0)、右上顶点(W, 0)和左下顶点(0, H)像素点作为控制点，则先确定当前块左上顶点、右上顶点和左下顶点控制点的运动矢量分别为 (vx_0, vy_0) 和 (vx_1, vy_1) 和 (vx_2, vy_2) ，然后根据公式 (5) 得到当前块中每一个子运动补偿单元的运动信息，其中 (x, y) 为子运动补偿单元相对于当前块的左上顶点像素的坐标，W 和 H 分别为当前块的宽和高。

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W}x + \frac{vx_2 - vx_0}{H}y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W}x + \frac{vy_2 - vy_0}{H}y + vy_0 \end{cases} \quad (5)$$

采用仿射运动模型进行预测的编码块称为仿射编码块。

通常的，可以使用基于仿射运动模型的先进运动矢量预测 (Advanced Motion Vector Prediction, AMVP) 模式或者基于仿射运动模型的融合 (Merge) 模式，获得仿射编码块的控制点的运动信息。

当前编码块的控制点的运动信息可以通过继承的控制点运动矢量预测方法或者构造的控制点运动矢量预测方法得到。

3) 继承的控制点运动矢量预测方法

继承的控制点运动矢量预测方法, 是指利用相邻已编码的仿射编码块的运动模型, 确定当前块的候选的控制点运动矢量。

以图 7 所示的当前块为例, 按照设定的顺序, 比如 A1->B1->B0->A0->B2 的顺序遍历当前块周围的相邻位置块, 找到该当前块的相邻位置块所在的仿射编码块, 获得该仿射编码块的控制点运动信息, 进而通过仿射编码块的控制点运动信息构造的运动模型, 推导出当前块的控制点运动矢量(用于 Merge 模式)或者控制点的运动矢量预测值(用于 AMVP 模式)。A1->B1->B0->A0->B2 仅作为一种示例, 其它组合的顺序也适用于本申请。另外, 相邻位置块不仅限于 A1、B1、B0、A0、B2。

10 相邻位置块可以作为一个像素点, 按照特定方法划分的预设大小的像素块, 比如可以作为一个 4x4 的像素块, 也可以为一个 4x2 的像素块, 也可以为其他大小的像素块, 不作限定。

下面以 A1 为例描述确定过程, 其他情况以此类推:

如图 7 所示, 若 A1 所在的编码块为 4 参数仿射编码块, 则获得该仿射编码块左上顶点(x4, y4)的运动矢量(vx4, vy4)、右上顶点(x5, y5)的运动矢量(vx5, vy5); 利用公式(6)计算
15 获得当前仿射编码块左上顶点(x0, y0)的运动矢量(vx0, vy0), 利用公式(7)计算获得当前仿射编码块右上顶点(x1, y1)的运动矢量(vx1, vy1)。

$$\begin{cases} vx_0 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_0 - x_4) - \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (y_0 - y_4) \\ vy_0 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (y_0 - y_4) \end{cases} \quad (6)$$

$$\begin{cases} vx_1 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_1 - x_4) - \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (y_1 - y_4) \\ vy_1 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (y_1 - y_4) \end{cases} \quad (7)$$

通过如上基于 A1 所在的仿射编码块获得的当前块的左上顶点(x0, y0)的运动矢量(vx0, vy0)、右上顶点(x1, y1)的运动矢量(vx1, vy1)的组合为当前块的候选的控制点运动矢量。
20

若 A1 所在的编码块为 6 参数仿射编码块, 则获得该仿射编码块左上顶点(x4, y4)的运动矢量(vx4, vy4)、右上顶点(x5, y5)的运动矢量(vx5, vy5)、左下顶点(x6, y6)的运动矢量(vx6, vy6); 利用公式(8)计算获得当前块左上顶点(x0, y0)的运动矢量(vx0, vy0), 利用公式(9)计算获得当前块右上顶点(x1, y1)的运动矢量(vx1, vy1)、利用公式(10)计算获得当前块左下顶点
25 (x2, y2)的运动矢量(vx2, vy2)。

$$\begin{cases} vx_0 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vx_6 - vx_4)}{y_6 - y_4} \times (y_0 - y_4) \\ vy_0 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vy_6 - vy_4)}{y_6 - y_4} \times (y_0 - y_4) \end{cases} \quad (8)$$

$$\begin{cases} vx_1 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vx_6 - vx_4)}{y_6 - y_4} \times (y_1 - y_4) \\ vy_1 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vy_6 - vy_4)}{y_6 - y_4} \times (y_1 - y_4) \end{cases} \quad (9)$$

$$\begin{cases} vx_2 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_2 - x_4) + \frac{(vx_6 - vx_4)}{y_6 - y_4} \times (y_2 - y_4) \\ vy_2 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_2 - x_4) + \frac{(vy_6 - vy_4)}{y_6 - y_4} \times (y_2 - y_4) \end{cases} \quad (10)$$

通过如上基于 A1 所在的仿射编码块获得的当前块的左上顶点(x0, y0)的运动矢量(vx0, vy0)、右上顶点(x1, y1)的运动矢量(vx1, vy1)、当前块左下顶点(x2, y2)的运动矢量(vx2, vy2)的组合为当前块的候选的控制点运动矢量。

5 需要说明的是, 其他运动模型、候选位置、查找遍历顺序也可以适用于本申请, 本申请实施例对此不做赘述。

需要说明的是, 采用其他控制点来表示相邻和当前编码块的运动模型的方法也可以适用于本申请, 此处不做赘述。

4) 构造的控制点运动矢量 (constructed control point motion vectors) 预测方法 1:

10 构造的控制点运动矢量预测方法, 是指将当前块的控制点周边邻近的已编码块的运动矢量进行组合, 作为当前仿射编码块的控制点的运动矢量, 而不需要考虑周边邻近的已编码块是否为仿射编码块。

利用当前编码块周边邻近的已编码块的运动信息确定当前块左上顶点和右上顶点的运动矢量。以图 8A 所示为例对构造的控制点运动矢量预测方法进行描述。需要说明的是,

15 图 8A 仅作为一种示例。

如图 8A 所示, 利用左上顶点相邻已编码块 A2, B2 和 B3 块的运动矢量, 作为当前块左上顶点的运动矢量的候选运动矢量; 利用右上顶点相邻已编码块 B1 和 B0 块的运动矢量, 作为当前块右上顶点的运动矢量的候选运动矢量。将上述左上顶点和右上顶点的候选运动矢量进行组合, 构成多个二元组, 二元组包括的两个已编码块的运动矢量可以作为当前块

20 的候选的控制点运动矢量, 参见如下公式 (11A) 所示:

$$\{v_{A2}, v_{B1}\}, \{v_{A2}, v_{B0}\}, \{v_{B2}, v_{B1}\}, \{v_{B2}, v_{B0}\}, \{v_{B3}, v_{B1}\}, \{v_{B3}, v_{B0}\} \quad (11A);$$

其中, v_{A2} 表示 A2 的运动矢量, v_{B1} 表示 B1 的运动矢量, v_{B0} 表示 B0 的运动矢量, v_{B2} 表示 B2 的运动矢量, v_{B3} 表示 B3 的运动矢量。

如图 8A 所示, 利用左上顶点相邻已编码块 A2, B2 和 B3 块的运动矢量, 作为当前块左上顶点的运动矢量的候选运动矢量; 利用右上顶点相邻已编码块 B1 和 B0 块的运动矢量, 作为当前块右上顶点的运动矢量的候选运动矢量, 利用坐下顶点相邻已编码块 A0、A1 的运动矢量作为当前块左下顶点的运动矢量的候选运动矢量。将上述左上顶点、右上顶点以及左下顶点的候选运动矢量进行组合, 构成三元组, 三元组包括的三个已编码块的运动矢量可以作为当前块的候选的控制点运动矢量, 参见如下公式 (11B)、(11C) 所示:

$$\begin{aligned} & \{v_{A2}, v_{B1}, v_{A0}\}, \{v_{A2}, v_{B0}, v_{A0}\}, \{v_{B2}, v_{B1}, v_{A0}\}, \{v_{B2}, v_{B0}, v_{A0}\}, \{v_{B3}, v_{B1}, v_{A0}\}, \{v_{B3}, v_{B0}, v_{A0}\} \quad (11B) \\ & \{v_{A2}, v_{B1}, v_{A1}\}, \{v_{A2}, v_{B0}, v_{A1}\}, \{v_{B2}, v_{B1}, v_{A1}\}, \{v_{B2}, v_{B0}, v_{A1}\}, \{v_{B3}, v_{B1}, v_{A1}\}, \{v_{B3}, v_{B0}, v_{A1}\} \quad (11C); \end{aligned}$$

其中, v_{A2} 表示 A2 的运动矢量, v_{B1} 表示 B1 的运动矢量, v_{B0} 表示 B0 的运动矢量, v_{B2} 表示 B2 的运动矢量, v_{B3} 表示 B3 的运动矢量, v_{A0} 表示 A0 的运动矢量, v_{A1} 表示 A1 的运动矢量。

需要说明的是，其他控制点运动矢量的组合的方法也可适用于本申请，此处不做赘述。

需要说明的是，采用其他控制点来表示相邻和当前编码块的运动模型的方法也可以适用于本申请，此处不做赘述。

- 5 5) 构造的控制点运动矢量 (constructed control point motion vectors) 预测方法2, 参见图8B所示。

步骤801, 获取当前块的各个控制点的运动信息。

以图8A所示为例, CP_k ($k=1, 2, 3, 4$)表示第 k 个控制点。 $A_0, A_1, A_2, B_0, B_1, B_2$ 和 B_3 为当前块的空域相邻位置, 用于预测 CP_1 、 CP_2 或 CP_3 ; T 为当前块的时域相邻位置, 用于预
10 测 CP_4 。

设, CP_1, CP_2, CP_3 和 CP_4 的坐标分别为 $(0, 0), (W, 0), (H, 0)$ 和 (W, H) , 其中 W 和 H 为当前块的宽度和高度。

对于每个控制点, 其运动信息按照以下顺序获得:

- (1) 对于 CP_1 , 检查顺序为 $B_2 \rightarrow A_2 \rightarrow B_3$, 如果 B_2 可得, 则采用 B_2 的运动信息。否则,
15 检测 A_2, B_3 。若三个位置的运动信息均不可得, 则无法获得 CP_1 的运动信息。

(2) 对于 CP_2 , 检查顺序为 $B_0 \rightarrow B_1$; 如果 B_0 可得, 则 CP_2 采用 B_0 的运动信息。否则, 检测 B_1 。若两个位置的运动信息均不可得, 则无法获得 CP_2 的运动信息。

(3) 对于 CP_3 , 检测顺序为 $A_0 \rightarrow A_1$;

(4) 对于 CP_4 , 采用 T 的运动信息。

- 20 此处 X 可得表示包括 X (X 为 $A_0, A_1, A_2, B_0, B_1, B_2, B_3$ 或 T) 位置的块已经编码并且采用帧间预测模式; 否则, X 位置不可得。

需要说明的是, 其他获得控制点的运动信息的方法也可适用于本申请, 此处不做赘述。

步骤802, 将控制点的运动信息进行组合, 得到构造的控制点运动信息。

- 25 将两个控制点的运动信息进行组合构成二元组, 用来构建4参数仿射运动模型。两个控制点的组合方式可以为 $\{CP_1, CP_4\}, \{CP_2, CP_3\}, \{CP_1, CP_2\}, \{CP_2, CP_4\}, \{CP_1, CP_3\}, \{CP_3, CP_4\}$ 。例如, 采用 CP_1 和 CP_2 控制点组成的二元组构建的4参数仿射运动模型, 可以记作Affine (CP_1, CP_2)。

- 30 将三个控制点的运动信息进行组合构成三元组, 用来构建6参数仿射运动模型。三个控制点的组合方式可以为 $\{CP_1, CP_2, CP_4\}, \{CP_1, CP_2, CP_3\}, \{CP_2, CP_3, CP_4\}, \{CP_1, CP_3, CP_4\}$ 。例如, 采用 CP_1, CP_2 和 CP_3 控制点构成的三元组构建的6参数仿射运动模型, 可以记作Affine (CP_1, CP_2, CP_3)。

- 35 将四个控制点的运动信息进行组合构成的四元组, 用来构建8参数双线性模型。采用 CP_1, CP_2, CP_3 和 CP_4 控制点构成的四元组构建的8参数双线性模型, 记做Bilinear (CP_1, CP_2, CP_3, CP_4)。

本申请实施例中，为了描述方便，将由两个控制点（或者两个已编码块）的运动信息组合简称为二元组，将三个控制点（或者两个已编码块）的运动信息组合简称为三元组，将四个控制点（或者四个已编码块）的运动信息组合简称为四元组。

- 按照预置的顺序遍历这些模型，若组合模型对应的某个控制点的运动信息不可得，则认为该模型不可得；否则，确定该模型的参考帧索引，并将控制点的运动矢量进行缩放，若缩放后的所有控制点的运动信息一致，则该模型不合法。若确定控制该模型的控制点的运动信息均可得，并且模型合法，则将该构建该模型的控制点的运动信息加入运动信息候选列表中。

控制点的运动矢量缩放的方法如公式(12)所示：

$$10 \quad \mathbf{MV}_s = \frac{\mathbf{CurPoc} - \mathbf{DesPoc}}{\mathbf{CurPoc} - \mathbf{SrcPoc}} \times \mathbf{MV} \quad (12)$$

其中， $\mathbf{CurPoc}$ 表示当前帧的POC号， $\mathbf{DesPoc}$ 表示当前块的参考帧的POC号， $\mathbf{SrcPoc}$ 表示控制点的参考帧的POC号， $\mathbf{MV}_s$ 表示缩放得到的运动矢量， $\mathbf{MV}$ 表示控制点的运动矢量。

需要说明的是，亦可将不同控制点的组合转换为同一位置的控制点。

- 例如将{CP1, CP4}, {CP2, CP3}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4}组合得到的4参数仿射运动模型转换为通过{CP1, CP2}或{CP1, CP2, CP3}来表示。转换方法为将控制点的运动矢量及其坐标信息，代入公式(2)，得到模型参数，再将{CP1, CP2}的坐标信息代入公式(3)，得到其运动矢量。

- 更直接地，可以按照以下公式(13)-(21)来进行转换，其中， W 表示当前块的宽度， H 表示当前块的高度，公式(13)-(21)中， (vx_0, vy_0) 表示CP1的运动矢量， (vx_1, vy_1) 表示CP2的运动矢量， (vx_2, vy_2) 表示CP3的运动矢量， (vx_3, vy_3) 表示CP4的运动矢量。

{CP1, CP2}转换为{CP1, CP2, CP3}可以通过如下公式(13)实现，即{CP1, CP2, CP3}中CP3的运动矢量可以通过公式(13)来确定：

$$\begin{cases} vx_2 = -\frac{vy_1 - vy_0}{W} H + vx_0 \\ vy_2 = +\frac{vx_1 - vx_0}{W} H + vy_0 \end{cases} \quad (13)$$

{CP1, CP3}转换为{CP1, CP2}或{CP1, CP2, CP3}可以通过如下公式(14)实现：

$$25 \quad \begin{cases} vx_1 = +\frac{vy_2 - vy_0}{H} W + vx_0 \\ vy_1 = -\frac{vx_2 - vx_0}{H} W + vy_0 \end{cases} \quad (14)$$

{CP2, CP3}转换为{CP1, CP2}或{CP1, CP2, CP3}可以通过如下公式(15)实现：

$$\begin{cases} vx_0 = \frac{vx_2 - vx_1}{W * W + H * H} W * W - \frac{vy_2 - vy_1}{W * W + H * H} H * W + vx_1 \\ vy_0 = \frac{vy_2 - vy_1}{W * W + H * H} W * W + \frac{vx_2 - vx_1}{W * W + H * H} H * W + vy_1 \end{cases} \quad (15)$$

{CP1, CP4}转换为{CP1, CP2}或{CP1, CP2, CP3}可以通过如下公式(16)或者(17)实现：

$$30 \quad \begin{cases} vx_1 = \frac{vx_3 - vx_0}{W * W + H * H} W * W + \frac{vy_3 - vy_0}{W * W + H * H} H * W + vx_0 \\ vy_1 = \frac{vy_3 - vy_0}{W * W + H * H} W * W - \frac{vx_3 - vx_0}{W * W + H * H} H * W + vy_0 \end{cases} \quad (16)$$

$$\begin{cases} vx_2 = \frac{vx_3 - vx_0}{W*W + H*H} H * H - \frac{vy_3 - vy_0}{W*W + H*H} H * W + vx_0 \\ vy_2 = \frac{vy_3 - vy_0}{W*W + H*H} W * H + \frac{vx_3 - vx_0}{W*W + H*H} H * H + vy_0 \end{cases} \quad (17)$$

{CP2, CP4}转换为{CP1, CP2}可以通过如下公式 (18) 实现, {CP2, CP4}转换为{CP1, CP2, CP3}可以通过公式 (18) 和 (19) 实现:

$$\begin{cases} vx_0 = -\frac{vy_3 - vy_1}{H} W + vx_1 \\ vy_0 = +\frac{vx_3 - vx_1}{H} W + vy_1 \end{cases} \quad (18)$$

$$\begin{cases} vx_2 = -\frac{vy_3 - vy_1}{H} W + vx_3 \\ vy_2 = +\frac{vx_3 - vx_1}{H} W + vy_3 \end{cases} \quad (19)$$

{CP3, CP4}转换为{CP1, CP2}可以通过如下公式 (20) 实现, {CP3, CP4}转换为{CP1, CP2, CP3}可以通过如下公式 (20) 和 (21) 实现:

$$\begin{cases} vx_0 = +\frac{vy_3 - vy_2}{W} H + vx_2 \\ vy_0 = -\frac{vx_3 - vx_2}{W} H + vy_2 \end{cases} \quad (20)$$

$$\begin{cases} vx_1 = +\frac{vy_3 - vy_2}{W} H + vx_3 \\ vy_1 = -\frac{vx_3 - vx_2}{W} H + vy_3 \end{cases} \quad (21)$$

例如将{CP1, CP2, CP4}, {CP2, CP3, CP4}, {CP1, CP3, CP4}组合的6参数仿射运动模型转换为控制点{CP1, CP2, CP3}来表示。转换方法为将控制点的运动矢量及其坐标信息, 代入公式(4), 得到模型参数, 再将{CP1, CP2, CP3}的坐标信息代入公式 (5), 得到其运动矢量。

更直接地, 可以按照以下公式 (22) - (24) 进行转换, 其中, W表示当前块的宽度, H表示当前块的高度, 公式(13)-(21)中, (vx_0, vy_0) 表示CP1的运动矢量, (vx_1, vy_1) 表示CP2的运动矢量, (vx_2, vy_2) 表示CP3的运动矢量, (vx_3, vy_3) 表示CP4的运动矢量。

{CP1, CP2, CP4}转换为{CP1, CP2, CP3}可以通过公式 (22) 实现:

$$\begin{cases} vx_2 = vx_3 + vx_0 - vx_1 \\ vy_2 = vy_3 + vy_0 - vy_1 \end{cases} \quad (22)$$

{CP2, CP3, CP4}转换为{CP1, CP2, CP3}可以通过公式 (23) 实现:

$$\begin{cases} vx_0 = vx_1 + vx_2 - vx_3 \\ vy_0 = vy_1 + vy_2 - vy_3 \end{cases} \quad (23)$$

{CP1, CP3, CP4}转换为{CP1, CP2, CP3}可以通过公式 (24) 实现:

$$\begin{cases} vx_1 = vx_3 + vx_0 - vx_2 \\ vy_1 = vy_3 + vy_0 - vy_2 \end{cases} \quad (24)$$

6) 基于仿射运动模型的先进运动矢量预测模式 (Affine AMVP mode):

(1) 构建候选运动矢量列表

利用继承的控制点运动矢量预测方法和/或构造的控制点运动矢量预测方法, 构建基于

仿射运动模型的AMVP模式的候选运动矢量列表。在本申请实施例中可以将基于仿射运动模型的AMVP模式的候选运动矢量列表称为控制点运动矢量预测值候选列表（control point motion vectors predictor candidate list），每个控制点的运动矢量预测值包括2个（4参数仿射运动模型）控制点的运动矢量或者包括3个（6参数仿射运动模型）控制点的运动矢量。

- 5 可选的，将控制点运动矢量预测值候选列表根据特定的规则进行剪枝和排序，并可将其截断或填充至特定的个数。

（2）确定最优的控制点运动矢量预测值

在编码端，利用控制点运动矢量预测值候选列表中的每个控制点运动矢量预测值，通过公式(3)/(5)获得当前编码块中每个子运动补偿单元的运动矢量，进而得到每个子运动补偿单元的运动矢量所指向的参考帧中对应位置的像素值，作为其预测值，进行采用仿射运动模型的运动补偿。计算当前编码块中每个像素点的原始值和预测值之间差值的平均值，选择最小平均值对应的控制点运动矢量预测值为最优的控制点运动矢量预测值，并作为当前编码块2个/3个控制点的运动矢量预测值。将表示该控制点运动矢量预测值在控制点运动矢量预测值候选列表中位置的索引号编码入码流发送给解码器。

- 15 在解码端，解析索引号，根据索引号从控制点运动矢量预测值候选列表中确定控制点运动矢量预测值（control point motion vectors predictor，CPMVP）。

（3）确定控制点的运动矢量

在编码端，以控制点运动矢量预测值作为搜索起始点在一定搜索范围内进行运动搜索获得控制点运动矢量（control point motion vectors，CPMV）。并将控制点运动矢量与控制点运动矢量预测值之间的差值（control point motion vectors differences，CPMVD）传递到解码端。

在解码端，解析控制点运动矢量差值，与控制点运动矢量预测值相加，得到控制点运动矢量。

7)仿射融合模式（Affine Merge mode）：

- 25 利用继承的控制点运动矢量预测方法和/或构造的控制点运动矢量预测方法，构建控制点运动矢量融合候选列表（control point motion vectors merge candidate list）。

可选的，将控制点运动矢量融合候选列表根据特定的规则进行剪枝和排序，并可将其截断或填充至特定的个数。

在编码端，利用融合候选列表中的每个控制点运动矢量，通过公式(3)/(5)获得当前编码块中每个子运动补偿单元（像素点或特定方法划分的大小为 $N_1 \times N_2$ 的像素块）的运动矢量，进而得到每个子运动补偿单元的运动矢量所指向的参考帧中位置的像素值，作为其预测值，进行仿射运动补偿。计算当前编码块中每个像素点的原始值和预测值之间差值的平均值，选择差值的平均值最小对应的控制点运动矢量作为当前编码块2个/3个控制点的运动矢量。将表示该控制点运动矢量在候选列表中位置的索引号编码入码流发送给解码器。

- 35 在解码端，解析索引号，根据索引号从控制点运动矢量融合候选列表中确定控制点运

动矢量（control point motion vectors, CPMV）。

另外，需要说明的是，本申请中，“至少一个”是指一个或者多个，“多个”是指两个或两个以上。“和/或”，描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B的情况，其中A,B可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项(个)”或其类似表达，是指的这些项中的任意组合，包括单项（个）或复数项（个）的任意组合。例如，a,b,或c中的至少一项（个），可以表示：a, b, c, a-b, a-c, b-c, 或a-b-c，其中a,b,c可以是单个，也可以是多个。

在本申请中，当使用帧间预测模式来解码当前块时，可使用语法元素来用信号发送帧间预测模式。

目前所采用的解析当前块采用的帧间预测模式的部分语法结构，可以参见表1所示。需要说明的是，语法结构中的语法元素还可以通过其他标识来表示，本申请对此不作具体限定。

表1

coding_unit(x0,y0,cbWidth,cbHeight) {	Descriptor or (描述子)
...	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if (allowAffineMerge)	
affine_merge_flag[x0][y0]	ae(v)
if(MaxNumMergeCand > 1 && affine_merge_flag[x0][y0] == 0)	
merge_idx[x0][y0]	ae(v)
if(MaxNumAffineMergeCand > 1 && affine_merge_flag[x0][y0])	
affine_merge_idx[x0][y0]	ae(v)
} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(allowAffineInter) {	
affine_inter_flag[x0][y0]	ae(v)
if(affine_inter_flag[x0][y0])	
affine_type_flag[x0][y0]	ae(v)
}	

MotionModelIdc[x0][y0] = affine_inter_flag[x0][y0] + affine_type_flag[x0][y0]	
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0) {	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
}	
mvp_l0_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0) {	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
}	
mvp_l1_flag[x0][y0]	ae(v)
}	
}	
...	
}	

语法元素merge_flag[x0][y0]可用于指示针对当前块是否采用融合模式。比如，当merge_flag[x0][y0]=1时，指示针对当前块采用融合模式，当merge_flag[x0][y0]=0时，指示针对当前块不采用融合模式。x0, y0表示当前块在视频图像的坐标。

变量allowAffineMerge可用于指示当前块是否满足采用基于仿射运动模型的merge模式的条件。比如allowAffineMerge=0，指示不满足采用基于仿射运动模型的merge模式的条件，allowAffineMerge=1，指示满足采用基于仿射运动模型的merge模式的条件。采用基于仿射运动模型的merge模式的条件可以是：当前块的宽和高中均大于或者等于8。cbWidth表示当前块的宽，cbHeight表示当前块的高，即，当cbWidth<8或cbHeight<8时，allowAffineMerge=0，当cbWidth>=8且cbHeight>=8时，allowAffineMerge=1。

变量allowAffineInter可用于指示当前块是否满足采用基于仿射运动模型的AMVP模式的条件。比如allowAffineInter=0，指示不满足采用基于仿射运动模型的AMVP模式的条件，allowAffineInter=1，指示满足采用基于仿射运动模型的AMVP模式的条件。采用基于仿射运动模型的AMVP模式的条件可以是：当前块的宽和高中均大于或者等于16。即，当
 5 cbWidth<16或cbHeight<16时，allowAffineInter=0，当cbWidth>=16且cbHeight>=16时，allowAffineInter=1。

语法元素affine_merge_flag[x0][y0]可用于指示针对当前块是否采用基于仿射运动模型的merge模式。当前块所在条带的类型（slice_type）为P型或者B型。比如，
 affine_merge_flag[x0][y0]=1，指示针对当前块采用基于仿射运动模型的merge模式，
 10 affine_merge_flag[x0][y0]=0，指示针对当前块不采用基于仿射运动模型的merge模式，可以采用平运运动模型的merge模式。

语法元素merge_idx[x0][y0]可用于指示针对merge候选列表的索引值。

语法元素affine_merge_idx[x0][y0]可用于指示针对仿射merge候选列表的索引值。

语法元素affine_inter_flag[x0][y0]可用于指示在当前块所在条带为P型条带或者B型条带
 15 时，针对当前块是否采用基于仿射运动模型的AMVP模式。比如，allowAffineInter=0，指示针对当前块采用基于仿射运动模型的AMVP模式，allowAffineInter=1，指示针对当前块不采用基于仿射运动模型的AMVP模式，可以采用平动运动模型的AMVP模式。

语法元素affine_type_flag[x0][y0]可以用于指示：在当前块所在条带为P型条带或者B型条带时，针对当前块是否采用6参数仿射运动模型进行运动补偿。
 20 affine_type_flag[x0][y0]=0，指示针对当前块不采用6参数仿射运动模型进行运动补偿，可以仅采用4参数仿射运动模型进行运动补偿；affine_type_flag[x0][y0]=1，指示针对当前块采用6参数仿射运动模型进行运动补偿。

参见表2所示，MotionModelIdc[x0][y0]=1，指示采用4参数仿射运动模型，
 MotionModelIdc[x0][y0]=2，指示采用6参数仿射运动模型，MotionModelIdc[x0][y0]=0指示
 25 采用平动运动模型。

表 2

MotionModelIdc[x0][y0]	motion model for motion compensation（运动补偿采用的运动模型）
0	translational motion（平动运动）
1	4-parameter affine motion（4 参数仿射运动）
2	6-parameter affine motion（6 参数仿射运动）

变量MaxNumMergeCand、MaxAffineNumMrgCand用于表示最大列表长度，指示构造的候选运动矢量列表的最大长度。inter_pred_idc[x0][y0]用于指示预测方向。PRED_L1用于指示后向预测。num_ref_idx_l0_active_minus1指示前向参考帧列表的参考帧个数，
 30 ref_idx_l0[x0][y0]指示当前块的前向参考帧索引值。mvd_coding(x0, y0, 0, 0)指示第一个运动矢量差。mvp_l0_flag[x0][y0]指示前向MVP候选列表索引值。PRED_L0指示前向预测。

num_ref_idx_l1_active_minus1指示后向参考帧列表的参考帧个数。ref_idx_l1[x0][y0]指示当前块的后向参考帧索引值，mvp_l1_flag[x0][y0]表示后向MVP候选列表索引值。

表1中，ae(v)表示采用基于自适应二元算术编码(context-based adaptive binary arithmetic coding, cabac)编码的语法元素。

5 下面针对帧间预测流程进行详细描述。参见图9A所示。

步骤601: 按照表1所示的语法结构，解析码流，确定当前块的帧间预测模式。

若确定当前块的帧间预测模式为基于仿射运动模型的AMVP模式，执行步骤602a。

即，语法元素merge_flag=0且affine_inter_flag=1，指示当前块的帧间预测模式为基于仿射运动模型的AMVP模式。

10 若确定当前块的帧间预测模式为基于仿射运动模型的融合(merge)模式，执行步骤502b。

即，语法元素merge_flag=1且affine_merge_flag=1，指示当前块的帧间预测模式为基于仿射运动模型的AMVP模式。

15 步骤602a: 构造基于仿射运动模型的AMVP模式对应的候选运动矢量列表，执行步骤503a。

利用继承的控制点运动矢量预测方法和/或构造的控制点运动矢量预测方法，推导得到当前块的候选的控制点运动矢量，来加入候选运动矢量列表。

20 候选运动矢量列表可以包括二元组列表(当前编码块为4参数仿射运动模型)或三元组列表。二元组列表中包括一个或者多个用于构造4参数仿射运动模型的二元组。三元组列表中包括一个或者多个用于构造6参数仿射运动模型的三元组。

可选的，将候选运动矢量二元组/三元组列表根据特定的规则进行剪枝和排序，并可将其截断或填充至特定的个数。

A1: 针对利用继承的控制运动矢量预测方式，构造候选运动矢量列表的流程进行说明。

25 以图7所示为例，比如，按照图7中A1->B1->B0->A0->B2的顺序遍历当前块周围的相邻位置块，找到相邻位置块所在的仿射编码块，获得该仿射编码块的控制点运动信息，进而通过仿射编码块的控制点运动信息构造运动模型，推导出当前块的候选的控制点运动信息。具体的，可以参见前面3)继承的控制点运动矢量预测方法中的相关描述，此处不再赘述。

30 示例性地，在当前块采用的仿射运动模型为4参数仿射运动模型(即，MotionModelIdc=1)，若相邻仿射解码块为4参数仿射运动模型，则获取该仿射解码块2个控制点的运动矢量：左上控制点(x4, y4)的运动矢量值(vx4, vy4)和右上控制点(x5, y5)的运动矢量值(vx5, vy5)。仿射解码块为在编码阶段采用仿射运动模型进行预测的仿射编码块。

35 利用相邻仿射解码块的2个控制点组成的4参数仿射运动模型，按照4仿射运动模型公式(6)、(7)分别推导得到当前块左上、右上2个控制点的运动矢量。

若相邻仿射解码块为6参数仿射运动模型，则获取相邻仿射解码块3个控制点的运动矢量，比如图7中，左上控制点(x4, y4)的运动矢量值(vx4, vy4)和右上控制点(x5, y5)的运动矢量值(vx5, vy5)和左下顶点(x6, y6)的运动矢量(vx6, vy6)。

利用相邻仿射解码块3个控制点组成的6参数仿射运动模型，按照6参数运动模型公式5 (8)、(9)分别推导得到当前块左上、右上2个控制点的运动矢量。

示例性地，当前解码块的仿射运动模型是6参数仿射运动模型(即MotionModelIdc=2)，

若相邻仿射解码块采用的仿射运动模型为6参数仿射运动模型，则获取相邻仿射解码块的3个控制点的运动矢量，比如图7中，左上控制点(x4, y4)的运动矢量值(vx4, vy4)和右上控制点(x5, y5)的运动矢量值(vx5, vy5)和左下顶点(x6, y6)的运动矢量(vx6, vy6)。

10 利用相邻仿射解码块3个控制点组成的6参数仿射运动模型，按照6参数仿射运动模型对应的公式(8)、(9)、(10)分别推导得到当前块左上、右上、左下3个控制点的运动矢量。

若相邻仿射解码块采用的仿射运动模型为4参数仿射运动模型，则获取该仿射解码块2个控制点的运动矢量：左上控制点(x4, y4)的运动矢量值(vx4, vy4)和右上控制点(x5, y5)的运动矢量值(vx5, vy5)。

15 利用相邻仿射解码块2个控制点组成的4参数仿射运动模型，按照4参数仿射运动模型公式(6)、(7)分别推导得到当前块左上、右上、左下3个控制点的运动矢量。

需要说明的是，其他运动模型、候选位置、查找顺序也可以适用于本申请，在此不做赘述。需要说明的是，采用其他控制点来表示相邻和当前编码块的运动模型的方法也可以适用于本申请，在此不做赘述。

20 A2：针对利用构造的控制运动矢量预测方式，构造候选运动矢量列表的流程进行说明。

示例性的，当前解码块采用的仿射运动模型是4参数仿射运动模型（即，

MotionModelIdc为1），利用当前编码块周边邻近的已编码块的运动信息确定当前编码块左上顶点和右上顶点的运动矢量。具体可以采用构造的控制点运动矢量预测方式1，或者采用构造的控制点运动矢量预测方法2，来构造候选运动矢量列表，具体方式参见上述4)和5)中的描述，此处不再赘述。

示例性的，当前解码块仿射运动模型是6参数仿射运动模型（即，MotionModelIdc为2），利用当前编码块周边邻近的已编码块的运动信息确定当前编码块左上顶点和右上顶点以及左下顶点的运动矢量。具体可以采用构造的控制点运动矢量预测方式1，或者采用构造的控制点运动矢量预测方法2，来构造候选运动矢量列表，具体方式参见上述4)和5)中的描述，此处不再赘述。

需要说明的是，其他控制点运动信息组合方式也可以适用于本申请，在此不做赘述。

步骤603a：解析码流，确定最优的控制点运动矢量预测值，执行步骤604a。

B1，若当前解码块采用的仿射运动模型是4参数仿射运动模型（MotionModelIdc为1），
35 则解析索引号，根据索引号从候选运动矢量列表中确定2个控制点的最优运动矢量预测

值。

示例性的，索引号为mvp_l0_flag或mvp_l1_flag。

B2，若当前解码块采用的仿射运动模型是6参数仿射运动模型（MotionModelIdc为2），则解析索引号，根据索引号从候选运动矢量列表中确定3个控制点的最优运动矢量预测值。

步骤604a：解析码流，确定控制点的运动矢量。

C1，当前解码块采用的仿射运动模型是4参数仿射运动模型（MotionModelIdc为1），从码流中解码得到当前块的2个控制点的运动矢量差值，分别根据各控制点的运动矢量差值和运动矢量预测值获得控制点的运动矢量值。以前向预测为例，2个控制点的运动矢量差分别为mvd_coding(x0, y0, 0, 0)和mvd_coding(x0, y0, 0, 1)。

示例性的，从码流中解码得到左上位置控制点和右上位置控制点的运动矢量差值，并分别与运动矢量预测值相加，得到当前块左上位置控制点和右上位置控制点的运动矢量值。

C2，当前解码块仿射运动模型是6参数仿射运动模型（MotionModelIdc为2）

15 从码流中解码得到当前块的3个控制点的运动矢量差，分别根据各控制点的运动矢量差值和运动矢量预测值获得控制点的运动矢量值。以前向预测为例，3个控制点的运动矢量差分别为mvd_coding(x0, y0, 0, 0)和mvd_coding(x0, y0, 0, 1)、mvd_coding(x0, y0, 0, 2)。

示例性的，从码流中解码得到左上控制点、右上控制点和左下控制点的运动矢量差值，20 并分别与运动矢量预测值相加，得到当前块左上控制点、右上控制点和左下控制点的运动矢量值。

步骤605a：根据控制点的运动信息以及当前解码块采用的仿射运动模型获得当前块中每个子块的运动矢量值。

对于当前仿射解码块的每一个子块（一个子块也可以等效为一个运动补偿单元，子块25 的宽和高小于当前块的宽和高），可采用运动补偿单元中预设位置像素点的运动信息来表示该运动补偿单元内所有像素点的运动信息。假设运动补偿单元的尺寸为MxN，则预设位置像素点可以为运动补偿单元中心点(M/2, N/2)、左上像素点(0, 0)，右上像素点(M-1, 0)，或其他位置的像素点。以下以运动补偿单元中心点为例说明，参见图9C所示。图9C中V0表示左上控制点的运动矢量，V1表示右上控制点的运动矢量。每个小方框表示一个运动30 补偿单元。

运动补偿单元中心点相对于当前仿射解码块左上顶点像素的坐标使用公式(25)计算得到，其中i为水平方向第i个运动补偿单元（从左到右），j为竖直方向第j个运动补偿单元（从上到下），(x_(i,j), y_(i,j))表示第(i, j)个运动补偿单元中心点相对于当前仿射解码块左上控制点像素的坐标。

35 若当前仿射解码块采用的仿射运动模型为6参数仿射运动模型，将(x_(i,j), y_(i,j))代入6

参数仿射运动模型公式(26), 获得每个运动补偿单元中心点的运动矢量, 作为该运动补偿单元内所有像素点的运动矢量($vx_{(i,j)}$, $vy_{(i,j)}$)。

若当前仿射解码块采用的仿射运动模型为 4 仿射运动模型, 将($x_{(i,j)}$, $y_{(i,j)}$)代入 4 参数仿射运动模型公式(27), 获得每个运动补偿单元中心点的运动矢量, 作为该运动补偿单元内所有像素点的运动矢量($vx_{(i,j)}$, $vy_{(i,j)}$)。

$$\begin{cases} x_{(i,j)} = M \times i + \frac{M}{2}, i = 0, 1, \dots \\ y_{(i,j)} = N \times j + \frac{N}{2}, j = 0, 1, \dots \end{cases} \quad (25)$$

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x + \frac{vx_2 - vx_0}{H} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vy_2 - vy_0}{H} y + vy_0 \end{cases} \quad (26)$$

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x - \frac{vy_1 - vy_0}{W} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vx_1 - vx_0}{W} y + vy_0 \end{cases} \quad (27)$$

步骤606a: 针对每个子块根据确定的子块的运动矢量值进行运动补偿得到该子块的像素预测值。

步骤602b: 构造基于仿射运动模型的merge模式的运动信息候选列表。

具体的, 可以利用继承的控制点运动矢量预测方法和/或构造的控制点运动矢量预测方法, 构造基于仿射运动模型的融合模式的运动信息候选列表。

可选的, 将运动信息候选列表根据特定的规则进行剪枝和排序, 并可将其截断或填充至特定的个数。

D1: 针对利用继承的控制运动矢量预测方式, 构造候选运动矢量列表的流程进行说明。

利用继承的控制点运动矢量预测方法, 推导得到当前块的候选的控制点运动信息, 加入运动信息候选列表。

20 按照图8A中A1, B1, B0, A0, B2的顺序遍历当前块的周边相邻位置块, 找到该位置所在的仿射编码块, 获得该仿射编码块的控制点运动信息, 进而通过其运动模型, 推导出当前块的候选的控制点运动信息。

如果此时候选运动矢量列表为空, 则将该候选的控制点运动信息加入候选列表; 否则依次遍历候选运动矢量列表中的运动信息, 检查候选运动矢量列表中是否存在与该候选的控制点运动信息相同的运动信息。如果候选运动矢量列表中不存在与该候选的控制点运动信息相同的运动信息, 则将该候选的控制点运动信息加入候选运动矢量列表。

其中, 判断两个候选运动信息是否相同需要依次判断它们的前后向参考帧、以及各个前后向运动矢量的水平和垂直分量是否相同。只有当以上所有元素都不相同时才认为这两个运动信息是不同的。

30 如果候选运动矢量列表中的运动信息个数达到最大列表长度MaxAffineNumMrgCand (MaxAffineNumMrgCand为正整数, 如1, 2, 3, 4, 5等, 以下以5为例进行描述, 不再

赘述)，则候选列表构建完毕，否则遍历下一个相邻位置块。

D2：利用构造的控制点运动矢量预测方法，推导得到当前块的候选的控制点运动信息，加入运动信息候选列表，参见图9B所示。

步骤601c，获取当前块的各个控制点的运动信息。可以参见5)中构造的控制点运动矢量预测方式2中，步骤801，此处不再赘述。

步骤602c，将控制点的运动信息进行组合，得到构造的控制点运动信息，可以参见图8B中步骤801，此处不再赘述。

步骤603c，将构造的控制点运动信息加入候选运动矢量列表。

若此时候选列表的长度小于最大列表长度MaxAffineNumMrgCand，则按照预置的顺序遍历这些组合，得到合法的组合作为候选的控制点运动信息，如果此时候选运动矢量列表为空，则将该候选的控制点运动信息加入候选运动矢量列表；否则依次遍历候选运动矢量列表中的运动信息，检查候选运动矢量列表中是否存在与该候选的控制点运动信息相同的运动信息。如果候选运动矢量列表中不存在与该候选的控制点运动信息相同的运动信息，则将该候选的控制点运动信息加入候选运动矢量列表。

15 示例性的，一种预置的顺序如下：Affine (CP1, CP2, CP3) -> Affine (CP1, CP2, CP4) -> Affine (CP1, CP3, CP4) -> Affine (CP2, CP3, CP4) -> Affine (CP1, CP2) -> Affine (CP1, CP3) -> Affine (CP2, CP3) -> Affine (CP1, CP4) -> Affine (CP2, CP4) -> Affine (CP3, CP4)，总共10种组合。

20 若组合对应的控制点运动信息不可得，则认为该组合不可得。若组合可得，确定该组合的参考帧索引（两个控制点时，选择参考帧索引最小的作为该组合的参考帧索引；大于两个控制点时，先选择出现次数最多的参考帧索引，若有多个参考帧索引的出现次数一样多，则选择参考帧索引最小的作为该组合的参考帧索引），并将控制点的运动矢量进行缩放。若缩放后的所有控制点的运动信息一致，则该组合不合法。

25 可选地，本申请实施例还可以针对候选运动矢量列表进行填充，比如，经过上述遍历过程后，此时候选运动矢量列表的长度小于最大列表长度MaxAffineNumMrgCand，则可以对候选运动矢量列表进行填充，直到列表的长度等于MaxAffineNumMrgCand。

可以通过补充零运动矢量的方法进行填充，或者通过将现有列表中已存在的候选的运动信息进行组合、加权平均的方法进行填充。需要说明的是，其他获得候选运动矢量列表填充的方法也可适用于本申请，在此不做赘述。

30 步骤603b：解析码流，确定最优的控制点运动信息。

解析索引号，根据索引号从候选运动矢量列表中确定最优的控制点运动信息。

步骤604b：根据最优的控制点运动信息以及当前解码块采用的仿射运动模型获得当前块中每个子块的运动矢量值。

同步步骤605a。

35 步骤605b：针对每个子块根据确定的子块的运动矢量值进行运动补偿得到该子块的像

素预测值。

本发明使用与上下文自适应二进制算术解码（CABAC）熵解码器或例如概率区间分割熵解码（PIPE）或相关解码器等其它熵解码器的相关技术。算术解码是在具有高解码效率的许多压缩算法中使用的一种形式的熵解码，因为其能够将符号映射到非整数长度码子。

5 大体上，使用CABAC对数据符号进行解码涉及以下步骤中的一个或者多个：

（1）二进制化：如果待解码符号是非二进制的，那么将其映射到“二进位”的序列，每个二进位可具有“0”或“1”的值；

（2）上下文指派：（在常规模式下）将每个二进位指派于一个上下文。上下文模型确定如何基于可用于给定二进位的信息来计算用于所述二进位的上下文，所述信息例如先前
10 经解码符号或二进位数的值。

（3）二进位编码：以算术编码器对二进位进行编码。为了对二进位进行编码，算术编码器需要二进位的值得概率作为输入，所述概率即二进位的值等于“0”的概率以及二进位的值等于“1”的概率。每个上下文的（估计）概率由称为“上下文状态”的整数值表示。每个上下文具有一个状态，且因此所述状态（即，估计概率）对于指派于一个上下文的二进位
15 是相同的，且在上下文之间不同。

（4）状态更新：用于选定上下文的概率（状态）是基于二进位的实际经解码值（例如，如果二进位的值为“1”，那么“1”的概率增加）来更新。

现有技术中通过CABAC解析仿射运动模型的参数信息，比如表1中affine_merge_flag、affine_merge_idx、affine_inter_flag、affine_type_flag时，在进行CABAC解析时，不同的语
20 法元素，需要采用不同的上下文。在本发明中，提出减少CABAC中使用的上下文的数目。由此，在不影响编码效率的情况下，减少了编码器和解码器存储上下文所需的空间。

对于affine_merge_flag和affine_inter_flag，现有技术在CABAC时使用两个不同的上下文集合（每个上下文集合有3个上下文）。在每个集合中使用的实际上下文索引等于与当前解码块左方相邻的同一语法元素的值、上方相邻的同一语法元素的值相加的结果，如表3
25 所示。其中，availableL表示当前解码块左方位置相邻块的可得性（是否存在且已解码）、availableA表示当前解码块上方位置相邻块的可得性（是否存在且已解码）。现有技术中，affine_merge_flag和affine_inter_flag的上下文个数为6个。

表 3 上下文索引

语法元素	condL	condA	Context index
affine_merge_flag[x0][y0]	affine_merge_flag[x0 - 1][y0]	affine_merge_flag[x0][y0 - 1]	(condL && availableL) + (condA && availableA)
affine_inter_flag[x0][y0]	affine_inter_flag[x0 - 1][y0]	affine_inter_flag[x0][y0 - 1]	(condL && availableL) + (condA && availableA)

			eA)
--	--	--	------

图10描述了本发明一个实施例提供的一种视频解码方法的流程，该实施例可以由如图3所示的视频解码器执行，如图10所示，该方法包括：

1001、解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2。

5 在一种实施方式中，所述当前块的语法元素1为affine_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

在一种实施方式中，所述当前块的语法元素1为subblock_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

该步骤具体可以由图3中的熵解码单元304执行。

10 其中，本发明实施例中的当前块可以为CU。

1002、对所述当前块的待熵解码的语法元素进行熵解码；其中，使用预设的上下文模型完成所述当前块的语法元素1的熵解码；或使用所述上下文模型完成所述当前块的语法元素2的熵解码。

该步骤具体可以由图3中的熵解码单元304执行。

15 1003、基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

该步骤具体可以由图3中的预测处理单元360执行。

1004、基于所述当前块的预测块获得所述当前块的重构图像。

该步骤具体可以由图3中的重构单元314执行。

20 本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型，因此解码器在进行熵解码时不需要进行上下文模型的确认，从而提高了解码器进行视频解码的解码效率；同时，视频解码器针对语法元素1和语法元素2也只需要存储一个上下文模型，减少了视频解码器的存储空间占用。

与图10描述的视频解码方法对应，本发明实施例还提供了一种编码方法，包括：

25 获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2；对所述当前块的待熵编码的语法元素进行熵编码；在对所述当前块待熵编码的语法元素进行熵编码时，使用预设的上下文模型完成所述当前块的语法元素1的熵编码；或使用所述上下文模型完成所述当前块的语法元素2的熵编码；输出包括所述熵编码后的所述当前块的语法元素的码流。其中，对所述当前块进行熵编码

30 时使用的上下文模型与图10描述的视频解码方法中的上下文模型相同。本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型，因此编码器在进行熵解码时不需要进行上下文模型的确认，从而提高了编码器进行视频编码的解码效率；同时，视频编码器针对语法元素1和语法元素2也只需要存储一个上下文模型，减少了视频编码器的存储空间占用。

图11描述了本发明另一个实施例提供的一种视频解码方法的流程，该实施例可以由如图3所示的视频解码器执行，如图11所示，该方法包括：

1101、解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2。

5 在一种实施方式中，所述当前块的语法元素1为affine_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

在一种实施方式中，所述当前块的语法元素1为subblock_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

该步骤具体可以由图3中的熵解码单元304执行。

10 1102、获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素1的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素2的上下文模型。

视频解码器针对所述语法元素1和语法元素2只需要存储一个上下文模型集合。

15 在一些实施方式中，所述预设的上下文模型集合只包括2个上下文模型；在另一些实施方式中，所述预设的上下文模型集合只包括3个上下文模型。可以理解的是，所述预设的上下文模型集合还可以包括4、5或6个上下文模型。本发明实施例对所述预设的上下文模型集合包括的上下文模型的数量不做限制。

在一种实施方式中，所述从预设的上下文模型集合中确定所述当前块的语法元素1的上下文模型包括：根据所述当前块的左方相邻块的语法元素1和语法元素2，以及所述当前块20 的上方相邻块的语法元素1和语法元素2确定所述当前块的语法元素1的上下文索引，所述当前块的语法元素1的上下文索引用于指示当前块的语法元素1的上下文模型。

在另一种实施方式中，所述从预设的上下文模型集合中确定所述当前块的语法元素2的上下文模型包括：根据所述当前块的左方相邻块的语法元素1和语法元素2，以及所述当前块的上方相邻块的语法元素1和语法元素2确定所述当前块的语法元素2的上下文索引，25 所述当前块的语法元素2的上下文索引用于指示当前块的语法元素2的上下文模型。

例如，在所述预设的上下文模型集合中的上下文模型的数量为3个时，所述当前块的语法元素1的上下文索引的值可以为所述上方相邻块的语法元素1和语法元素2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值之和。或者，所述当前块的语法元素2的上下文索引的值为所述上方相邻块的语法元素1和语法元素2进行或操作的值与30 所述左方相邻块的语法元素1和语法元素2进行或操作的值之和。

具体地，语法元素1 affine_merge_flag和语法元素2 affine_inter_flag可以共享一个上下文模型集合（集合有3个上下文模型），每个集合中使用的实际上下文索引等于当前解码块左方相邻的两个语法元素进行或操作的值、上方相邻的两个语法元素进行或操作的值相加的结果，如表4所示。其中“|”表示或操作。

35 表 4 本发明的上下文索引

语法元素	condL	condA	Context index
affine_merge_flag[x0][y0]	affine_merge_flag[x0 - 1][y0]	affine_merge_flag[x0][y0 - 1]	(condL && availableL) +
affine_inter_flag[x0][y0]	affine_inter_flag[x0 - 1][y0]	affine_inter_flag[x0][y0 - 1]	(condA && availableA)

例如，在所述预设的上下文模型集合中的上下文模型的数量为2个时，所述当前块的语法元素1的上下文索引的值可以为所述上方相邻块的语法元素1和语法元素2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值再进行或操作的结果。或者，所述当前块的语法元素2的上下文索引的值为所述上方相邻块的语法元素1和语法元素2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值再进行或操作的结果。

具体地，语法元素1 affine_merge_flag和语法元素2 affine_inter_flag共享一个上下文模型集合（集合有2个上下文模型），每个集合中使用的实际上下文索引等于当前解码块左方相邻的两个语法元素进行或操作的值、上方相邻的两个语法元素进行或操作的值再进行或操作的结果，如表5所示。其中“|”表示或操作。本发明实施例二将affine_merge_flag和affine_inter_flag上下文的个数减少到2个。

表 5 本发明的上下文索引

语法元素	condL	condA	Context index
affine_merge_flag[x0][y0]	affine_merge_flag[x0 - 1][y0]	affine_merge_flag[x0][y0 - 1]	(condL && availableL)
affine_inter_flag[x0][y0]	affine_inter_flag[x0 - 1][y0]	affine_inter_flag[x0][y0 - 1]	(condA && availableA)

1103、基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码。

15 该步骤具体可以由图3中的熵解码单元304执行。

1104、基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块。

该步骤具体可以由图3中的预测处理单元360执行。

1105、基于所述当前块的预测块获得所述当前块的重构图像。

20 该步骤具体可以由图3中的重构单元314执行。

本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型集合，因此视频解码器针对语法元素1和语法元素2也只需要存储一个上下文模型，减少了视频解码器的存储空间占用。

与图11描述的视频解码方法对应，本发明实施例还提供了一种编码方法，包括：

25 获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2；获取所述待熵解码的语法元素对应的上下文模型；

其中，从预设的上下文模型集合中确定所述当前块的语法元素1的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素2的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出包括所述熵编码后的所述当前块的语法元素的码流。其中，对所述当前块进行熵编码时使用的

5 上下文模型集合与图11描述的视频解码方法中的上下文模型集合相同。本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型集合，因此视频编码器针对语法元素1和语法元素2也只需要存储一个上下文模型集合，减少了视频编码器的存储空间占用。

图12描述了本发明一个实施例提供的一种视频解码方法的流程，该实施例可以由如图3所示的视频解码器执行，如图12所示，该方法包括：

10 1201、解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素3或所述当前块的语法元素4。

在一种实施方式中，所述当前块的语法元素3为merge_idx，或所述当前块的语法元素4为affine_merge_idx。

在一种实施方式中，所述当前块的语法元素3为merge_idx，或所述当前块的语法元素4

15 为subblock_merge_idx。

该步骤具体可以由图3中的熵解码单元304执行。

1202、获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素3的上下文模型；或从预设的所述上下文模型中确定所述当前块的语法元素4的上下文模型。

20 在一种实施方式中，所述预设的上下文模型集合包括的上下文模型的数量为5个。可以理解的是，所述预设的上下文模型集合包括的上下文模型的数量也可以为其他数值，例如可以为1、2、3、或4个。其中，在所述预设的上下文模型集合包括的上下文模型的数量为1个，所述预设的上下文模型集合即为所述1个上下文模型。本发明实施例对所述预设的上下文模型集合包括的上下文模型的数量不做限制。

25 该步骤具体可以由图3中的熵解码单元304执行。

1203、基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码；

该步骤具体可以由图3中的熵解码单元304执行。

1204、基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得

30 所述当前块的预测块。

该步骤具体可以由图3中的预测处理单元360执行。

1205、基于所述当前块的预测块获得所述当前块的重构图像。

该步骤具体可以由图3中的重构单元314执行。

本实施例中，当前块的语法元素3和语法元素4共用了一个上下文模型集合，因此视频

35 解码器针对语法元素3和语法元素4也只需要存储一个上下文模型，减少了视频解码器的存

存储空间占用。

与图12描述的视频解码方法对应，本发明实施例还提供了一种编码方法，包括：获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素3或所述当前块的语法元素4；获取所述待熵解码的语法元素对应的上下文模型；其中，从
5 预设的上下文模型集合中确定所述当前块的语法元素3的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素4的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出包括所述熵编码后的所述当前块的语法元素的码流。其中，对所述当前块进行熵编码时使用的上下文模型集合与图12描述的视频解码方法中的上下文模型集合相同。本实施例中，当前块的语法
10 元素3和语法元素4共用了一个上下文模型集合，因此视频编码器针对语法元素3和语法元素4也只需要存储一个上下文模型，减少了视频编码器的存储空间的占用。

本发明一个实施例提供了一种视频解码器30，包括：

熵解码单元304，用于解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2；对
15 所述当前块的待熵解码的语法元素进行熵解码；其中，使用预设的上下文模型完成所述当前块的语法元素1的熵解码；或使用所述上下文模型完成所述当前块的语法元素2的熵解码。

在一种实施方式中，所述当前块的语法元素1为affine_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

在一种实施方式中，所述当前块的语法元素1为subblock_merge_flag，或所述当前块的
20 语法元素2为affine_inter_flag。

预测处理单元360，用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块。

重构单元314，用于基于所述当前块的预测块获得所述当前块的重构图像。

本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型，因此解码器在
25 进行熵解码时不需要进行上下文模型的确认，从而提高了解码器进行视频解码的解码效率；同时，视频解码器针对语法元素1和语法元素2也只需要存储一个上下文模型，减少了视频解码器的存储空间的占用。

相应地，本发明一个实施例提供了一种视频编码器20，包括：熵编码单元270，用于获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法
30 元素1或所述当前块的语法元素2；对所述当前块的待熵编码的语法元素进行熵编码；在对所述当前块待熵编码的语法元素进行熵编码时，使用预设的上下文模型完成所述当前块的语法元素1的熵编码；或使用所述上下文模型完成所述当前块的语法元素2的熵编码；输出
272，用于输出包括所述熵编码后的所述当前块的语法元素的码流。其中，对所述当前块进行熵编码时使用的上下文模型与图10描述的方法中的上下文模型相同。本实施例中，当
35 前块的语法元素1和语法元素2共用了一个上下文模型，因此编码器在进行熵解码时不需要进行上下文模型的确认，从而提高了编码器进行视频编码的解码效率；同时，视频编码器

针对语法元素1和语法元素2也只需要存储一个上下文模型，减少了视频编码器的存储空间的占用。

本发明另一个实施例提供了一种视频解码器30，包括：

熵解码单元304，用于解析接收的码流，以获得当前块的待熵解码的语法元素，所述
5 当前块待熵解码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素1的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素2的上下文模型；基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码。

10 在一种实施方式中，所述当前块的语法元素1为affine_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

在一种实施方式中，所述当前块的语法元素1为subblock_merge_flag，或所述当前块的语法元素2为affine_inter_flag。

在一种实施方式中，所述熵解码单元304可以具体用于：根据所述当前块的左方相邻
15 块的语法元素1和语法元素2，以及所述当前块的上方相邻块的语法元素1和语法元素2确定所述当前块的语法元素1的上下文索引，所述当前块的语法元素1的上下文索引用于指示当前块的语法元素1的上下文模型；或

根据所述当前块的左方相邻块的语法元素1和语法元素2，以及所述当前块的上方相邻
20 块的语法元素1和语法元素2确定所述当前块的语法元素2的上下文索引，所述当前块的语法元素2的上下文索引用于指示当前块的语法元素2的上下文模型。

例如，所述当前块的语法元素1的上下文索引的值为所述上方相邻块的语法元素1和语法元素2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值之和；
或

所述当前块的语法元素2的上下文索引的值为所述上方相邻块的语法元素1和语法元素
25 2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值之和。

例如，所述当前块的语法元素1的上下文索引的值为所述上方相邻块的语法元素1和语法元素2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值再进行或操作的结果；或

所述当前块的语法元素2的上下文索引的值为所述上方相邻块的语法元素1和语法元素
30 2进行或操作的值与所述左方相邻块的语法元素1和语法元素2进行或操作的值再进行或操作的结果。

预测处理单元360，用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块。

重构单元314，用于基于所述当前块的预测块获得所述当前块的重构图像。

35 本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型集合，因此视频

解码器针对语法元素1和语法元素2也只需要存储一个上下文模型，减少了视频解码器的存储空间占用。

相应地，本发明实施例还提供了一种视频编码器，包括：熵编码单元270，用于获得当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素1或所述当前块的语法元素2；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素1的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素2的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出272，用于输出包括所述熵编码后的所述当前块的语法元素的码流。其中，对所述当前块进行熵编码时使用的上下文模型集合与图11描述的视频解码方法中的上下文模型集合相同。本实施例中，当前块的语法元素1和语法元素2共用了一个上下文模型集合，因此视频编码器针对语法元素1和语法元素2也只需要存储一个上下文模型集合，减少了视频编码器的存储空间的占用。

本发明另一个实施例提供了一种视频解码器30，包括：

熵解码单元304，用于解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素3或所述当前块的语法元素4；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素3的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素4的上下文模型；基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码。

其中，所述预设的上下文模型集合包括1、2、3、4或5个上下文模型。可以理解的是，在所述预设的上下文模型集合只包括1个上下文模型时，所述预设的上下文模型集合即为所述1个上下文模型。

在一种实施方式中，所述当前块的语法元素3为merge_idx，用于指示所述当前块的merge候选列表的索引值；或所述当前块的语法元素4为affine_merge_idx，用于指示所述当前块的仿射merge候选列表的索引值；或

所述当前块的语法元素3为merge_idx，用于指示所述当前块的merge候选列表的索引值；或所述当前块的语法元素4为subblock_merge_idx，用于指示针对子块merge候选列表的索引值。

预测处理单元360，用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块。

重构单元314，用于基于所述当前块的预测块获得所述当前块的重构图像。

本实施例中，当前块的语法元素3和语法元素4共用了一个上下文模型集合，因此视频解码器针对语法元素3和语法元素4也只需要存储一个上下文模型，减少了视频解码器的存储空间的占用。

相应地，本发明实施例还提供了一种视频编码器，包括：熵编码单元270，用于获得

当前块待熵编码的语法元素，所述当前块待熵编码的语法元素包括所述当前块的语法元素3或所述当前块的语法元素4；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素3的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素4的上下文模型；基于所述当前块的待熵编码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵编码；输出272，用于输出包括所述熵编码后的所述当前块的语法元素的码流。其中，对所述当前块进行熵编码时使用的上下文模型集合与图12描述的视频解码方法中的上下文模型集合相同。本实施例中，当前块的语法元素3和语法元素4共用了一个上下文模型集合，因此视频编码器针对语法元素3和语法元素4也只需要存储一个上下文模型，减少了视频编码器的存储空间的占用。

10 本发明实施例一提出affine_merge_flag和affine_inter_flag共享一个上下文集合（集合有3个上下文），每个集合中使用的实际上下文索引等于当前解码块左方相邻的两个语法元素进行或操作的值、上方相邻的两个语法元素进行或操作的值相加的结果，如表4所示。其中“|”表示或操作。本发明实施例一将affine_merge_flag和affine_inter_flag上下文的个数减少到3个。

15 本发明实施例二提出affine_merge_flag和affine_inter_flag共享一个上下文集合（集合有2个上下文），每个集合中使用的实际上下文索引等于当前解码块左方相邻的两个语法元素进行或操作的值、上方相邻的两个语法元素进行或操作的值再进行或操作的结果，如表5所示。其中“|”表示或操作。本发明实施例二将affine_merge_flag和affine_inter_flag上下文的个数减少到2个。

20 本发明实施例三提出affine_merge_flag和affine_inter_flag共享一个上下文。本发明实施例三将affine_merge_flag和affine_inter_flag上下文的个数减少到1个。

对于merge_idx和affine_merge_idx，现有技术利用截断一元码对其进行二值化，并在CABAC时使用两个不同的上下文集合（每个上下文集合有5个上下文），对其二值化后的每个二进位使用不同的上下文。现有技术中，merge_idx和affine_merge_idx的上下文个数为10个。

25 本发明实施例四提出merge_idx和affine_merge_idx共享一个上下文集合（每个上下文集合有5个上下文）。本发明实施例四将merge_idx和affine_merge_idx上下文的个数减少到5个。

在另外一些技术中，表一中的语法元素affine_merge_flag[x0][y0]可以替换为subblock_merge_flag[x0][y0]，用于指示当前块是否采用基于子块的merge模式；表一中的语法元素affine_merge_idx[x0][y0]可以替换为subblock_merge_idx[x0][y0]，用于指示针对子块merge候选列表的索引值。

此时，本发明实施例一至实施例四仍然适用，即subblock_merge_flag和affine_inter_flag共享一个上下文集合（或上下文）及获得索引方法，merge_idx和subblock_merge_idx共享一个上下文集合（或上下文）。

35 本发明实施例还提供了一种视频解码器，包括用于执行上述任一所述的方法的执行电

路。

本发明实施例还提供了一种视频解码器，包括：至少一个处理器；和与所述至少一个处理器耦合的非易失性计算机可读存储介质，所述非易失性计算机可读存储介质存储有可被所述至少一个处理器执行的计算机程序，当所述计算机程序被所述至少一个处理器执行
5 时，使得所述视频解码器用于执行上一任一所述的方法。

本发明实施例还提供了一种计算机可读存储介质，用于存储可被处理器执行的计算机程序，当所述计算机程序被所述至少一个处理器执行时，执行上述任一所述的方法。

本发明实施例还提供了一种计算机程序，当所述计算机程序被执行时，执行上述任一所述的方法。

10 在一个或一个以上实例中，所描述功能可以硬件、软件、固件或其任何组合来实施。如果在软件中实施，那么所述功能可作为一或多个指令或代码在计算机可读介质上存储或传输，并且由基于硬件的处理单元执行。计算机可读介质可以包含计算机可读存储介质，其对应于例如数据存储介质或通信介质的有形介质，通信介质例如根据通信协议包含有助于将计算机程序从一处传送到另一处的任何介质。以此方式，计算机可读介质通常可对应
15 于（1）非暂时性的有形计算机可读存储介质，或（2）通信介质，例如，信号或载波。数据存储介质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介质。计算机程序产品可包含计算机可读介质。

借助于实例而非限制，此类计算机可读存储介质可包括RAM、ROM、EEPROM、
20 CD-ROM或其它光盘存储器、磁盘存储器或其它磁性存储设备、闪存，或可用以存储呈指令或数据结构形式的所需程序代码且可由计算机存取的任何其它介质。并且，任何连接可适当地称为计算机可读介质。举例来说，如果使用同轴电缆、光纤缆线、双绞线、数字订户线（digital subscriber line, 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，则同轴电缆、光纤缆线、双绞线、DSL或例如红外线、无线电
25 及微波等无线技术包含在介质的定义中。但是，应理解，所述计算机可读存储介质及数据存储介质并不包括连接、载波、信号或其它暂时性介质，而是实际上针对于非暂时性有形存储介质。如本文中所使用，磁盘和光盘包含压缩光盘（compact disc, CD）、激光光盘、光学光盘、数字多功能光盘（digital versatile disc, DVD）、软性磁盘及蓝光光盘，其中磁盘通常以磁性方式再现数据，而光盘用激光以光学方式再现数据。以上各项的组合也应包
30 含于计算机可读介质的范围内。

指令可以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（digital signal processor, DSP）、通用微处理器、专用集成电路（application specific integrated circuit, ASIC）、现场可编程逻辑阵列（field programmable logic arrays, 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
35 构或适用于实施本文中所描述的技术的任何其它结构中的任一者。另外，在一些方面中，本文中所描述的功能性可在用于编码和解码的专用硬件和/或软件模块内提供，或并入在

合成编解码器中。并且，所述技术可完全实施于一或多个电路或逻辑元件中。

本公开的技术可以在包含无线手持机、集成电路 (integrated circuit, IC) 或 IC 集合 (例如, 芯片组) 的多种设备或装置中实施。本公开描述各种组件、模块或单元是为了强调用于执行所揭示的技术的设备的功能方面, 但未必需要通过不同硬件单元实现。确切地, 如
5 上文所描述, 各种单元可结合合适的软件和/或固件组合在编解码器硬件单元中, 或由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。

权 利 要 求

1、一种视频解码方法，其特征在于，包括：

解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；

5 对所述当前块的待熵解码的语法元素进行熵解码；其中，使用预设的上下文模型完成所述当前块的语法元素 1 的熵解码；或使用所述上下文模型完成所述当前块的语法元素 2 的熵解码；

基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

10 基于所述当前块的预测块获得所述当前块的重构图像。

2、一种视频解码方法，其特征在于，包括：

解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；

获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中

15 确定所述当前块的语法元素 1 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 2 的上下文模型；

基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码；

20 基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

基于所述当前块的预测块获得所述当前块的重构图像。

3、根据权利要求 2 所述的方法，其特征在于，所述预设的上下文模型集合中的上下文模型的数量为 2 个或 3 个。

25 4、根据权利要求 2 或 3 所述的方法，其特征在于，所述从预设的上下文模型集合中确定所述当前块的语法元素 1 的上下文模型包括：

根据所述当前块的左方相邻块的语法元素 1 和语法元素 2，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 1 的上下文索引，所述当前块的语法元素 1 的上下文索引用于指示当前块的语法元素 1 的上下文模型；

或

30 所述从预设的上下文模型集合中确定所述当前块的语法元素 2 的上下文模型包括：

根据所述当前块的左方相邻块的语法元素 1 和语法元素 2，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 2 的上下文索引，所述当前块的语法元素 2 的上下文索引用于指示当前块的语法元素 2 的上下文模型。

5、根据权利要求 4 所述的方法，若所述预设的上下文模型集合中的上下文模型的数量

为 3 个, 所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和; 或

所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和。

6、根据权利要求 4 所述的方法, 若所述预设的上下文模型集合中的上下文模型的数量为 2 个, 所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果; 或

10 所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果。

7、根据权利要求 1 至 6 任一所述的方法, 其特征在于, 所述当前块的语法元素 1 为 `affine_merge_flag`, 用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式; 或所述当前块的语法元素 2 为 `affine_inter_flag`, 用于在所述当前块所在条带为 P 型条带或者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 `AMVP` 模式;

或

所述当前块的语法元素 1 为 `subblock_merge_flag`, 用于指示当前块是否采用基于子块的 `merge` 模式; 或所述当前块的语法元素 2 为 `affine_inter_flag`, 用于在所述当前块所在条带为 P 型条带或者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 `AMVP` 模式。

8、一种视频解码方法, 其特征在于, 包括:

解析接收的码流, 以获得当前块的待熵解码的语法元素, 所述当前块待熵解码的语法元素包括所述当前块的语法元素 3 或所述当前块的语法元素 4;

获取所述待熵解码的语法元素对应的上下文模型; 其中, 从预设的上下文模型集合中确定所述当前块的语法元素 3 的上下文模型; 或从预设的所述上下文模型集合中确定所述当前块的语法元素 4 的上下文模型;

基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码;

基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理, 以获得所述当前块的预测块;

基于所述当前块的预测块获得所述当前块的重构图像。

9、根据权利要求 8 所述的方法, 其特征在于, 所述预设的上下文模型集合包括 5 个上下文模型。

10、根据权利要求 8 或 9 所述的方法, 其特征在于, 所述当前块的语法元素 3 为 `merge_idx`, 用于指示所述当前块的 `merge` 候选列表的索引值; 或所述当前块的语法元素 4

为 `affine_merge_idx`，用于指示所述当前块的仿射 `merge` 候选列表的索引值；或

所述当前块的语法元素 3 为 `merge_idx`，用于指示所述当前块的 `merge` 候选列表的索引值；或所述当前块的语法元素 4 为 `subblock_merge_idx`，用于指示针对子块 `merge` 候选列表的索引值。

- 5 11、一种视频解码方法，其特征在于，包括：

解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；

- 10 根据所述当前块的左方相邻块的语法元素 1 和语法元素 2 的取值，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 的取值，确定所述当前块的待熵解码的语法元素的上下文索引的值；

基于所述当前块的待熵解码的语法元素的上下文索引的值对所述待熵解码的语法元素进行熵解码；

基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

- 15 基于所述当前块的预测块获得所述当前块的重构图像。

12、根据权利要求 11 所述的方法，其特征在于，所述当前块的语法元素 1 为 `affine_merge_flag`，用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式；或所述当前块的语法元素 2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式；

- 20 或

所述当前块的语法元素 1 为 `subblock_merge_flag`，用于指示当前块是否采用基于子块的 `merge` 模式；或所述当前块的语法元素 2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式。

- 25 13、根据权利要求 11 或 12 所述的方法，其特征在于，所述根据所述当前块的左方相邻块的语法元素 1 和语法元素 2 的取值，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 的取值，确定所述当前块的待熵解码的语法元素的上下文索引的值包括：

根据如下逻辑表达式确定所述当前块的待熵解码的语法元素的上下文索引的值：

$\text{Context index} = (\text{condL} \ \&\& \ \text{availableL}) + (\text{condA} \ \&\& \ \text{availableA})$;

- 30 其中，`condL` = 语法元素 1[`x0 - 1`][`y0`] | 语法元素 2[`x0 - 1`][`y0`]，语法元素 1[`x0 - 1`][`y0`] 表示所述左方相邻块的语法元素 1 的取值，语法元素 2[`x0 - 1`][`y0`] 表示所述左方相邻块的语法元素 2 的取值；

`condA` = 语法元素 1[`x0`][`y0 - 1`] | 语法元素 2[`x0`][`y0 - 1`]，语法元素 1[`x0`][`y0 - 1`] 表示所述上方相邻块的语法元素 1 的取值，语法元素 2[`x0`][`y0 - 1`] 表示所述上方相邻块的语法元素 2 的取值；

- 35 `availableL` 表示所述左方相邻块是否可得，`availableA` 表示所述上方相邻块是否可得。

14、一种视频解码器，其特征在于，包括：

熵解码单元，用于解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；对所述当前块的待熵解码的语法元素进行熵解码；其中，使用预设的上下文模型完成所述当前块 5 的语法元素 1 的熵解码；或使用所述上下文模型完成所述当前块的语法元素 2 的熵解码；

预测处理单元，用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

重构单元，用于基于所述当前块的预测块获得所述当前块的重构图像。

15、一种视频解码器，其特征在于，包括：

10 熵解码单元，用于解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；获取所述待熵解码的语法元素对应的上下文模型；其中，从预设的上下文模型集合中确定所述当前块的语法元素 1 的上下文模型；或从预设的所述上下文模型集合中确定所述当前块的语法元素 2 的上下文模型；基于所述当前块的待熵解码的语法元素对应的上下文模型对所述 15 待熵解码的语法元素进行熵解码；

预测处理单元，用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

重构单元，用于基于所述当前块的预测块获得所述当前块的重构图像。

16、根据权利要求 15 所述的视频解码器，其特征在于，所述预设的上下文模型集合中 20 的上下文模型的数量为 2 个或 3 个。

17、根据权利要求 15 或 16 所述的视频解码器，其特征在于，所述熵解码单元具体用于：

根据所述当前块的左方相邻块的语法元素 1 和语法元素 2，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 1 的上下文索引，所述当前块 25 的语法元素 1 的上下文索引用于指示当前块的语法元素 1 的上下文模型；或

根据所述当前块的左方相邻块的语法元素 1 和语法元素 2，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 确定所述当前块的语法元素 2 的上下文索引，所述当前块的语法元素 2 的上下文索引用于指示当前块的语法元素 2 的上下文模型。

18、根据权利要求 17 所述的视频解码器，若所述预设的上下文模型集合中的上下文模 30 型的数量为 3 个，所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和；或

所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值之和。

35 19、根据权利要求 17 所述的视频解码器，若所述预设的上下文模型集合中的上下文模

型的数量为 2 个, 所述当前块的语法元素 1 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果; 或

所述当前块的语法元素 2 的上下文索引的值为所述上方相邻块的语法元素 1 和语法元素 2 进行或操作的值与所述左方相邻块的语法元素 1 和语法元素 2 进行或操作的值再进行或操作的结果。

20、根据权利要求 14 至 19 任一所述的视频解码器, 其特征在于, 所述当前块的语法元素 1 为 `affine_merge_flag`, 用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式; 或所述当前块的语法元素 2 为 `affine_inter_flag`, 用于在所述当前块所在条带为 P 型条带或 10 者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 AMVP 模式;

或

所述当前块的语法元素 1 为 `subblock_merge_flag`, 用于指示当前块是否采用基于子块的 `merge` 模式; 或所述当前块的语法元素 2 为 `affine_inter_flag`, 用于在所述当前块所在条带为 P 型条带或者 B 型条带时, 指示所述当前块是否采用基于仿射运动模型的 AMVP 模式。

15 21、一种视频解码器, 其特征在于, 包括:

熵解码单元, 用于解析接收的码流, 以获得当前块的待熵解码的语法元素, 所述当前块待熵解码的语法元素包括所述当前块的语法元素 3 或所述当前块的语法元素 4; 获取所述待熵解码的语法元素对应的上下文模型; 其中, 从预设的上下文模型集合中确定所述当前块的语法元素 3 的上下文模型; 或从预设的所述上下文模型集合中确定所述当前块的语法元素 4 的上下文模型; 基于所述当前块的待熵解码的语法元素对应的上下文模型对所述待熵解码的语法元素进行熵解码;

预测处理单元, 用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理, 以获得所述当前块的预测块;

重构单元, 用于基于所述当前块的预测块获得所述当前块的重构图像。

25 22、根据权利要求 21 所述的视频解码器, 其特征在于, 所述预设的上下文模型集合包括 5 个上下文模型。

23、根据权利要求 21 或 22 所述的视频解码器, 其特征在于,

所述当前块的语法元素 3 为 `merge_idx`, 用于指示所述当前块的 `merge` 候选列表的索引值; 或所述当前块的语法元素 4 为 `affine_merge_idx`, 用于指示所述当前块的仿射 `merge` 候选列表的索引值;

或

所述当前块的语法元素 3 为 `merge_idx`, 用于指示所述当前块的 `merge` 候选列表的索引值; 或所述当前块的语法元素 4 为 `subblock_merge_idx`, 用于指示针对子块 `merge` 候选列表的索引值。

35 24、一种视频解码器, 其特征在于, 包括:

熵解码单元，用于解析接收的码流，以获得当前块的待熵解码的语法元素，所述当前块待熵解码的语法元素包括所述当前块的语法元素 1 或所述当前块的语法元素 2；根据所述当前块的左方相邻块的语法元素 1 和语法元素 2 的取值，以及所述当前块的上方相邻块的语法元素 1 和语法元素 2 的取值，确定所述当前块的待熵解码的语法元素的上下文索引 5 的值；基于所述当前块的待熵解码的语法元素的上下文索引的值对所述待熵解码的语法元素进行熵解码；

预测处理单元，用于基于熵解码得到的所述当前块的语法元素对所述当前块进行预测处理，以获得所述当前块的预测块；

重构单元，用于基于所述当前块的预测块获得所述当前块的重构图像。

10 25、根据权利要求 24 所述的视频解码器，其特征在于，所述当前块的语法元素 1 为 `affine_merge_flag`，用于指示所述当前块是否采用基于仿射运动模型的 `merge` 模式；或所述当前块的语法元素 2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式；

或

15 所述当前块的语法元素 1 为 `subblock_merge_flag`，用于指示当前块是否采用基于子块的 `merge` 模式；或所述当前块的语法元素 2 为 `affine_inter_flag`，用于在所述当前块所在条带为 P 型条带或者 B 型条带时，指示所述当前块是否采用基于仿射运动模型的 AMVP 模式。

26、根据权利要求 24 或 25 所述的视频解码器，其特征在于，所述熵解码单元具体用于：

20 根据如下逻辑表达式确定所述当前块的待熵解码的语法元素的上下文索引的值：

$\text{Context index} = (\text{condL} \ \&\& \ \text{availableL}) + (\text{condA} \ \&\& \ \text{availableA})$;

其中，`condL` = 语法元素 1[`x0 - 1`][`y0`] | 语法元素 2[`x0 - 1`][`y0`]，语法元素 1[`x0 - 1`][`y0`] 表示所述左方相邻块的语法元素 1 的取值，语法元素 2[`x0 - 1`][`y0`] 表示所述左方相邻块的语法元素 2 的取值；

25 `condA` = 语法元素 1[`x0`][`y0 - 1`] | 语法元素 2[`x0`][`y0 - 1`]，语法元素 1[`x0`][`y0 - 1`] 表示所述上方相邻块的语法元素 1 的取值，语法元素 2[`x0`][`y0 - 1`] 表示所述上方相邻块的语法元素 2 的取值；

`availableL` 表示所述左方相邻块是否可得，`availableA` 表示所述上方相邻块是否可得。

27、一种视频解码器，其特征在于，包括用于执行如权利要求 1 至 13 任一所述的方法 30 的执行电路。

28、一种视频解码器，其特征在于，包括：

至少一个处理器；和

与所述至少一个处理器耦合的非易失性计算机可读存储介质，所述非易失性计算机可读存储介质存储有可被所述至少一个处理器执行的计算机程序，当所述计算机程序被所述 35 至少一个处理器执行时，使得所述视频解码器用于执行如权利要求 1 至 13 任一所述的方法。

29、一种计算机可读存储介质，用于存储可被处理器执行的计算机程序，当所述计算机程序被所述至少一个处理器执行时，执行如权利要求 1 至 13 任一所述的方法。