



(12) 发明专利

(10) 授权公告号 CN 113170139 B

(45) 授权公告日 2023. 12. 05

(21) 申请号 202080006625.2
 (22) 申请日 2020.01.10
 (65) 同一申请的已公布的文献号
 申请公布号 CN 113170139 A
 (43) 申请公布日 2021.07.23
 (66) 本国优先权数据
 PCT/CN2019/071160 2019.01.10 CN
 PCT/CN2019/071747 2019.01.15 CN
 (85) PCT国际申请进入国家阶段日
 2021.05.31
 (86) PCT国际申请的申请数据
 PCT/CN2020/071334 2020.01.10
 (87) PCT国际申请的公布数据
 W02020/143742 EN 2020.07.16
 (73) 专利权人 北京字节跳动网络技术有限公司
 地址 100041 北京市石景山区实兴大街30
 号院3号楼2层B-0035房间
 专利权人 字节跳动有限公司
 (72) 发明人 张莉 张凯 刘鸿彬 王悦

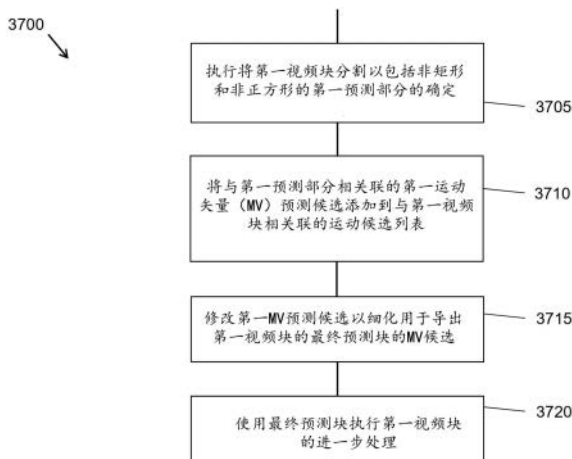
(74) 专利代理机构 北京市柳沈律师事务所
 11105
 专利代理师 张亮
 (51) Int.Cl.
 H04N 19/13 (2006.01)
 H04N 19/167 (2006.01)
 H04N 19/18 (2006.01)
 (56) 对比文件
 CN 108432250 A, 2018.08.21
 US 2018324454 A1, 2018.11.08
 US 2015271515 A1, 2015.09.24
 CN 109076236 A, 2018.12.21
 US 2013188720 A1, 2013.07.25
 KR 20150136017 A, 2015.12.04
 US 2013272423 A1, 2013.10.17
 CN 107736022 A, 2018.02.23
 CN 107750457 A, 2018.03.02
 CN 107113446 A, 2017.08.29
 CN 105704490 A, 2016.06.22
 审查员 黄驰
 权利要求书3页 说明书65页 附图47页

(54) 发明名称

上下文自适应二进制算数编码的简化上下文建模

(57) 摘要

一种用于处理视频的方法,包括:通过处理器执行确定将第一视频块分割以包括非矩形和非正方形的第一预测部分;将与第一预测部分相关联的第一运动矢量(MV)预测候选添加到与第一视频块相关联的运动候选列表中,其中从子块MV预测候选导出第一MV预测候选;以及使用运动候选列表执行第一视频块的进一步处理。



1. 一种视频处理方法,包括:

执行视频的当前视频块和所述视频的比特流之间的转换,其中,所述比特流符合规则,所述规则规定了上下文建模限于来自用于编解码所述当前视频块和满足位置标准的相邻块的信息,

其中,所述规则规定:不允许使用包含所述当前视频块的当前视频单元外部的相邻块,并且

其中所述当前视频单元包括编解码树单元或编解码树单元行。

2. 根据权利要求1所述的方法,其中,所述相邻块是左下块或右上块。

3. 根据权利要求1所述的方法,其中,所述规则规定:将包含所述当前视频块的当前视频单元外部的相邻块用作不可用块。

4. 根据权利要求1所述的方法,其中,所述规则规定:将包含所述当前视频块的当前视频单元外部的相邻块用作具有默认值。

5. 根据权利要求1所述的方法,其中,所述规则规定:将包含所述当前视频块的当前视频单元外部的相邻块用作替换块。

6. 根据权利要求1所述的方法,其中,所述当前视频单元包括视频区域、或视频数据处理单元、或条带、或片、或片组、或在图片边界处。

7. 一种视频处理方法,包括:

执行视频的当前视频块和所述视频的比特流之间的转换,其中,所述比特流符合规则,所述规则规定了将来自相邻块的编解码信息用于表示所述当前视频块的比特流的一个或多个语法元素的上下文建模,

其中,所述规则规定:排除使用所述当前视频块的编解码树单元或编解码树单元行之外的相邻块。

8. 根据权利要求7所述的方法,其中,所述来自相邻块的编解码信息基于在所述比特流中表示所述相邻块的语法元素。

9. 根据权利要求7或8所述的方法,其中,所述相邻块对应于用于从所述比特流重构所述当前视频块所访问的块。

10. 根据权利要求7或8所述的方法,其中,所述相邻块对应于来自用于空域Merge候选构造的五个位置的恰好两个块。

11. 根据权利要求7或8所述的方法,其中,所述相邻块对应于来自用于空域Merge候选构造的五个位置的恰好一个块。

12. 根据权利要求7所述的方法,其中,所述规则规定:对于所述上下文建模,所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素被视为不可用。

13. 根据权利要求7所述的方法,其中,所述规则规定:对于所述上下文建模,所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素被视为具有默认值。

14. 根据权利要求7所述的方法,其中,所述规则规定:对于所述上下文建模,用外部相邻块来替换所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素。

15. 根据权利要求7所述的方法,其中,所述规则规定:对于所述上下文建模,用所述编解码树单元或包括可用编解码信息的编解码树单元中的N个可用相邻块中的一个来替换所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素,其中N是整数。

16. 根据权利要求7所述的方法,其中,所述一个或多个语法元素包括:alf_ctb_flag、qt_split_cu_flag、mtt_split_cu_flag、cu_skip_flag、amvr_mode、inter_affine_flag、merge_triangle_flag and merge_subblock_flag、cu_skip_flag、pred_mode_flag、pcm_flag、intra_luma_ref_idx、intra_luma_mpm_flag、intra_luma_mpm_idx、intra_luma_mpm_remainder、intra_chroma_pred_mode、merge_flag、inter_pred_idc、inter_affine_flag、cu_affine_type_flag、ref_idx_l0、mvp_l0_flag、ref_idx_l1 mvp_l1_flag、amvr_flag、amvr_4pel_flag、gbi_idx、cu_cbf、mmvd_flag、mmvd_merge_flag、或pred_mode_flag。

17. 根据权利要求16所述的方法,其中,语法元素的上下文ctx导出如下:

ctx=A1是帧内?1:0;ctx+=B1是帧内?1:0;

其中,A1是左下相邻块,并且B1是右上相邻块。

18. 根据权利要求16所述的方法,其中pred_mode_flag的上下文ctx导出如下:

ctx=(A1是帧内) || (B1是帧内)?1:0

其中,A1是左下相邻块,并且B1是右上相邻块。

19. 根据权利要求16所述的方法,其中,语法元素的上下文ctx导出如下:

ctx=A1是帧内或帧间帧内?1:0;ctx+=B1是帧内或帧间帧内?1:0;

其中,A1是左下相邻块,并且B1是右上相邻块。

20. 根据权利要求16所述的方法,其中,语法元素的上下文ctx导出如下:

ctx=(A1是帧内或帧间帧内) || (B1是帧内或帧间帧内)?1:0;

其中,A1是左下相邻块,并且B1是右上相邻块。

21. 根据权利要求16所述的方法,其中,通过排除使用用于空域Merge候选构造的五个位置处的相邻块来导出语法元素的上下文ctx。

22. 根据权利要求21所述的方法,其中,所述语法元素对应于pred_mode_flag。

23. 根据权利要求17至22中任一项所述的方法,其中,通过将不可用的相邻块视为帧间编解码块而导出所述上下文。

24. 根据权利要求17至22中任一项所述的方法,其中,通过将不可用的相邻块视为帧内编解码块而导出所述上下文。

25. 根据权利要求17至22中任一项所述的方法,其中,通过将不可用的相邻块视为帧间帧内编解码块而导出所述上下文。

26. 根据权利要求17至22中任一项所述的方法,其中,通过将位于与所述当前视频块不同的编解码树单元行中的相邻块视为帧间编解码块而导出所述上下文。

27. 根据权利要求17至22中任一项所述的方法,其中,通过将位于与所述当前视频块不同的编解码树单元行中的相邻块视为帧内编解码块而导出所述上下文。

28. 根据权利要求17至22中任一项所述的方法,其中,通过将位于与所述当前视频块不同的编解码树单元行中的相邻块视为帧间帧内编解码块而导出所述上下文。

29. 根据权利要求1或7所述的方法,其中,使用几何分割模式执行所述视频的当前视频块和所述视频的比特流之间的转换,所述几何分割模式包括三角分割模式。

30. 根据权利要求29所述的方法,其中,在所述几何分割模式中,所述视频块被分割为多个预测分割,至少一个是非正方形、非矩形分割。

31. 根据权利要求1或7所述的方法,其中,所述转换包括:将所述视频编码到所述比特流中。

32. 根据权利要求1或7所述的方法,其中,所述转换包括:解码所述比特流以生成所述视频的像素值。

33. 根据权利要求32所述的方法,其中,所述解码包括:根据所述规则解析所述比特流的语法元素,以生成解码像素值。

34. 一种视频解码装置,其包括处理器,所述处理器被配置以实现权利要求1至33中任一项所述的方法。

35. 一种视频编码装置,其包括处理器,所述处理器被配置以实现权利要求1至33中任一项所述的方法。

36. 一种其上存储了计算机代码的非瞬时性计算机可读存储介质,当所述代码被处理器执行时,使所述处理器实现权利要求1至33中任一项中所述的方法。

37. 一种用于存储视频的比特流的方法,所述方法包括:

从视频的视频块生成所述视频的所述比特流,其中,所述比特流符合规则,所述规则规定了上下文建模限于来自用于编解码所述视频块和满足位置标准的相邻块的信息;以及

将所述比特流存储在非瞬时性计算机可读存储介质中,

其中,所述规则规定:不允许使用包含所述视频块的当前视频单元外部的相邻块,并且其中所述当前视频单元包括编解码树单元或编解码树单元行。

38. 一种用于存储视频的比特流的方法,所述方法包括:

从视频的视频块生成所述视频的所述比特流,其中,所述比特流符合规则,所述规则规定了将来自相邻块的编解码信息用于表示所述视频块的比特流的一个或多个语法元素的上下文建模;以及

将所述比特流存储在非瞬时性计算机可读存储介质中,

其中,所述规则规定:排除使用所述视频块的编解码树单元或编解码树单元行之外的相邻块。

上下文自适应二进制算数编码的简化上下文建模

[0001] 相关申请的交叉引用

[0002] 根据适用的《专利法》和/或《巴黎公约》的规定,本申请及时要求于2019年1月10日提交的国际专利申请号PCT/CN2019/071160、于2019年1月15日提交的国际专利申请号PCT/CN2019/071747的优先权和利益。根据美国法律,将上述申请的全部公开以引用方式并入本文,作为本申请公开的一部分。

技术领域

[0003] 本文涉及视频和图像编码和解码技术。

背景技术

[0004] 在互联网和其他数字通信网络中,数字视频占用的带宽最大。随着能够接收和显示视频的连接的用户设备数量的增加,预计数字视频使用的带宽需求将继续增长。

发明内容

[0005] 所公开的技术可由视频或图像解码器或编码器实施例使用,其中几何分割可用于视频编码或解码。

[0006] 在一个示例方面,公开了一种视频处理方法。该方法包括:确定视频的视频区域的视频块和视频的编解码表示之间的转换使用几何分割模式(geometry partitioning mode),其中视频块被分割成多个预测分割,多个预测分割至少包括第一预测分割;使用与视频块相关联的一个或多个子块合并(Merge)候选,将N个运动信息候选添加到视频块的运动候选列表,其中N是正整数;从运动候选列表导出多个预测分割的运动候选;以及基于多个预测分割的运动候选执行转换。

[0007] 在另一示例方面,公开了另一种视频处理方法。该方法包括:为视频的视频块和视频的编解码表示之间的转换确定视频块被分割成多个预测分割(prediction partition),所述多个预测分割至少包括第一预测分割;将第一运动信息候选添加到与第一预测分割相关联的第一候选列表、并且将第二运动信息候选添加到与第二预测分割相关联的第二候选列表,其中第一运动信息候选和第二运动信息候选继承自单个运动候选;以及使用第一运动信息候选和/或第二运动信息候选执行转换。

[0008] 在另一示例方面,公开了另一种视频处理方法。该方法包括:确定视频的视频块和视频的编解码表示之间的转换使用启用了子块的几何分割模式(geometric partitioning mode),其中,视频块被分割以包括预测分割,使得预测分割的子块被分配其子块运动信息;以及基于确定执行所述转换。

[0009] 在另一示例方面,公开了另一种视频处理方法。该方法包括:根据继承规则(inheritance rule),为使用其中视频块被分割成多个预测分割的几何分割模式的视频的视频块和视频的编解码表示之间的转换,确定是否在视频块的样点位置处启用从用于预测分割的运动信息推导的运动候选继承广义双向预测(GBi,generalized bi prediction)加

权因子;以及基于所述确定执行转换。GBi也称为具有基于编解码单元加权(BCW)的双向预测。

[0010] 在另一示例方面,公开了另一种视频处理方法。该方法包括:根据规则,为使用其中视频块被分割成多个预测分割的几何分割模式的视频的视频块和视频的编解码表示之间的转换,确定是否至少在视频块的样点位置处启用双向预测的使用;以及基于所述确定执行转换。

[0011] 在另一示例方面,公开了另一种视频处理方法。该方法包括:为使用其中将视频块至少分割成第一预测分割和第二预测分割的几何分割模式的视频的视频块和视频的编解码表示之间的转换,从来自运动信息的第一预测块和第二预测块确定最终预测块,所述运动信息与为第一预测分割的样点位置导出的第一组运动信息、为第二预测分割的样点位置导出的第二组运动信息不同;以及基于最终预测块执行转换;其中视频块使用多假设技术(multi-hypothesis technique)编解码。

[0012] 在另一示例方面,公开了另一种视频处理方法。该方法包括:为视频的视频区域的视频块和视频的比特流表示之间的转换,确定(1)用于将视频块划分成预测分割以使得至少一个预测分割是非矩形和非正方形分割的划分模式与(2)用于转换的分割的Merge候选的索引之间的关系、以及允许在视频区域级更改所述关系的比特流表示的格式;以及基于确定执行转换。

[0013] 在另一示例方面,公开了另一种视频处理方法。该方法包括:根据定义编解码表示的语法元素的语法规则,执行视频的视频块和视频的编解码表示之间的转换;其中,语法规则规定了指示视频块使用几何分割模式的第一标志(flag)、以及指示视频块使用Merge模式的第二标志的信令顺序;其中,几何分割模式包括:将视频块划分成多个预测分割,至少一个预测分割具有非矩形、非正方形形状,并且其中,Merge模式允许从视频块的相邻块(neighboring blocks)继承运动矢量信息。

[0014] 在另一示例方面,公开了另一种视频处理方法。该方法包括:根据定义编解码表示的语法的语法规则,执行视频的视频块和视频的编解码表示之间的转换;其中,语法规则规定了基于是否对视频块的转换使用另一编解码工具,选择性地包括指示视频块使用几何分割模式的标志的信令;其中,几何分割模式包括将视频块划分成多个预测分割,至少一个预测分割具有非矩形、非正方形形状。

[0015] 在另一示例方面,公开了另一种视频处理方法。该方法包括:确定为视频的视频块和视频的编解码表示之间的转换启用几何分割模式,其中,根据基于相邻块的几何分割模式的划分模式(splitting pattern),将视频块划分成多个预测分割;以及基于所述确定执行转换。

[0016] 在另一示例方面,公开了另一种视频处理方法。该方法包括:基于几何分割模式来为视频的视频块与视频的编解码表示之间的转换确定划分模式,其中,根据基于相邻块的划分模式的划分模式,将视频块划分成多个预测分割;以及基于所述确定执行转换。

[0017] 在另一示例方面,公开了另一种视频处理方法。该方法包括:执行视频的视频块和视频的编解码表示之间的转换,其中,根据包括第一预测分割的几何分割模式的划分模式,将视频块分割成多个预测分割;其中,编解码表示包括多个语法元素(syntax element),其中一个语法元素与划分模式相对应,并且其他语法元素与多个预测分割的Merge索引相对

应。

[0018] 在另一示例方面,公开了另一种视频处理方法。该方法包括:确定视频的视频块和视频的编解码表示之间的转换基于几何分割模式,在所述几何分割模式中,根据包括第一预测分割的划分模式将视频块分割成多个预测分割;以及基于所述确定执行转换;其中,使用基于上下文的编解码在编解码表示中信令通知视频块的几何分割模式,其中上下文取决于视频块的编解码条件。

[0019] 在另一示例方面,公开了另一种视频处理方法。该方法包括:执行视频的视频块和视频的编解码表示之间的转换,其中,根据具有第一预测分割的划分模式,使用几何预测模式将视频块分割为多个预测分割;其中,在编解码表示中为候选列表的两个候选索引项分配码字(codeword),并且其中编解码表示的格式允许改变视频片段中码字的分配。

[0020] 在另一示例方面,公开了另一种视频处理方法。该方法包括:为视频的视频块和视频的编解码表示之间的转换,确定是否基于使用视频块的特性的规则启用几何分割模式的使用;以及根据所述确定执行转换,其中,几何分割模式将视频块分割成多个预测分割。

[0021] 在另一示例方面,公开了另一种视频处理方法。该方法包括:根据定义编解码表示的语法元素格式的规则,执行视频的视频块和视频的编解码表示之间的转换;其中,所述规则规定了是否信令通知使用第一编解码模式的指示,或者是否信令通知所述指示基于视频块对第二编解码模式的使用;其中,第二编解码模式是Merge模式;其中,Merge模式使得在没有整个视频块的运动矢量残差的情况下,能够从Merge候选列表中的Merge候选继承运动信息。

[0022] 在另一示例方面,公开了另一种视频处理方法。该方法包括:执行视频的视频块和视频的编解码表示之间的转换;其中,编解码表示的格式符合第一规则和/或第二规则,第一规则规定了是否将信令通知多个编解码模式的语法元素包括在编解码表示中,第二规则规定了将信令通知多个编解码模式的语法元素包括在编解码表示中的顺序;其中,多个编解码模式包括几何分割模式、帧间帧内(MHIntra inter-intra)模式、子块Merge模式或运动矢量残差Merge(MMVD,merge with motion vector differencing)模式中的两个或多个;其中,在帧间帧内(inter-intra)编解码模式中,从帧内预测信号和帧间预测信号导出视频块的预测块;其中,在子块Merge模式中,转换使用为块内的每个子块导出的运动信息;其中,在MMVD模式中,使用Merge和运动矢量残差(MVD)编解码组合的模式;并且其中,Merge模式使得能够在没有整个视频块的MVD的情况下,从Merge候选列表中的Merge候选继承运动信息。

[0023] 在另一示例方面,公开了另一种视频处理方法。该方法包括:为视频的当前视频块和视频的编解码表示之间的转换,检查视频块的相邻块的相邻运动矢量精度信息是否可用于基于相邻块的位置来确定当前块的运动矢量精度;以及基于检查来执行转换。

[0024] 在另一示例方面,公开了另一种视频处理方法。该方法包括:基于相邻块相对于当前视频块的最大编解码单元(LCU,largest coding unit)行的位置,为视频的当前视频块和视频的编解码表示之间的转换确定用于编解码表示的上下文编解码模式;以及基于确定执行转换。

[0025] 在另一示例方面,公开了另一种视频处理方法。该方法包括:执行视频的当前视频块和视频的编解码表示之间的转换,其中,编解码表示符合规则,该规则规定了上下文建模

限于来自用于解码当前视频块和/或满足位置标准的相邻块的信息。

[0026] 在另一示例方面,公开了另一种视频处理方法。该方法包括:执行视频的当前视频块和视频的编解码表示之间的转换,其中,编解码表示符合规则,该规则规定了将来自相邻块的编解码信息用于表示所述当前视频块的编解码表示的一个或多个语法元素的上下文建模。

[0027] 在另一示例方面,公开了一种处理视频的方法。该方法包括:由处理器执行确定将第一视频块分割以包括非矩形和非正方形的第一预测部分;将与第一预测部分相关联的第一运动矢量(MV)预测候选添加到与第一视频块相关联的运动候选列表,其中,从基于子块的运动候选导出第一MV预测候选;以及使用运动候选列表执行第一视频块的进一步处理。

[0028] 在另一示例方面,一种处理视频的方法包括:基于第一划分模式分割第一视频块以包括第一预测部分,第一预测部分与第一Merge索引和第一编解码值相关联;基于第二划分模式分割第二视频块以包括第二预测部分,第二视频块的第二预测部分与第二Merge索引和第二编解码值相关联,其中第一编解码值等于第二编解码值,并且基于第二视频块位于与第一视频块不同的序列、图片(picture)、片(tile)、条带(slice)、编解码树单元(CTU)或块内,第二视频块的第二划分模式、第二Merge索引中的一个或多个不同于第一视频块;以及执行第一视频块和第二视频块的进一步处理。

[0029] 在另一示例方面,一种处理视频的方法包括:由处理器识别第一视频块的特性;基于该特性确定要应用于第一视频块的几何预测模式的状态,几何预测模式的状态是以下中的一个:将几何预测模式应用于第一视频块的启用状态、或者不将几何预测模式应用于第一视频块的禁用状态;以及根据几何预测模式的状态执行第一视频块的进一步处理。

[0030] 在另一示例方面,一种处理视频的方法包括:由处理器执行确定将第一视频块分割以包括第一预测部分和第二预测部分,第一预测部分和第二预测部分是非矩形和非正方形;将与第一预测部分相关联的第一MV预测候选、以及与第二预测部分相关联的第二MV预测候选添加到与第一视频块相关联的运动候选列表,其中从与先前编解码块相关联的单个MV预测候选中导出第一MV预测候选和第二MV预测候选;以及使用运动候选列表执行第一视频块的进一步处理。

[0031] 在另一示例方面,一种处理视频的方法包括:由处理器执行确定将第一视频块分割以包括非矩形和非正方形的第一预测部分;将与第一预测部分相关联的第一运动矢量(MV)预测候选添加到与第一视频块相关联的运动候选列表;修改第一MV预测候选以细化(refine)用于导出第一视频块的最终预测块的MV候选;以及使用最终预测块执行第一视频块的进一步处理。

[0032] 在另一示例方面,一种处理视频的方法包括:由处理器执行确定将第一视频块分割以包括非矩形和非正方形的第一预测部分;将与第一预测部分相关联的第一运动矢量(MV)预测候选添加到与第一视频块相关联的运动候选列表;并且其中,从具有加权因子的双向预测MV预测候选导出第一MV预测候选;以及使用不具有加权因子的运动候选列表执行第一视频块的进一步处理。

[0033] 在另一示例方面,一种处理视频的方法包括:由处理器执行确定将第一视频块分割以包括非矩形和非方形的第一预测部分,并且根据多假设编解码第一视频块。

[0034] 在另一示例方面,一种处理视频的方法包括:由处理器执行确定将第一视频块分

割以包括非矩形和非正方形的第一预测部分,并且其中不应用重叠块运动补偿(OBMC, overlapped block motion compensation)。

[0035] 在另一示例方面,一种处理视频的方法包括:由处理器执行确定将第一视频块分割以包括2个以上预测部分。在一个示例中,第一视频块被分割为每一部分都是非矩形和非正方形的第一预测部分、第二预测部分、第三预测部分和第四预测部分;以及使用第一预测部分、第二预测部分、第三预测部分和第四预测部分执行第一视频块的进一步处理。

[0036] 在另一示例方面,公开了一种视频处理方法。该方法包括:根据规则执行视频块与视频块的比特流表示之间的转换,该规则规定了将编解码的跳过(skip)/Merge模式与第二编解码模式分别对待,其中第二编解码模式是三角分割模式或MHIntra(有时也称为组合帧间帧内模式,或CIIP,combined inter-intra mode)模式或子块Merge列表模式。

[0037] 在另一示例方面,公开了另一种视频处理方法。该方法包括:根据规则执行视频块和当前视频块的比特流表示之间的转换,该规则规定了在转换中使用多个子块编解码工具的指示的特定顺序,其中该特定顺序是以下中的一个:(a)与运动矢量残差的合并(MMVD)、随后是子块Merge列表、随后是三角分割模式TPM、随后是MHIntra,或者(b)MMVD、随后是子块Merge列表、随后是MHIntra、随后是TPM。

[0038] 在又一示例方面,公开了另一种视频处理方法。该方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR,adaptive motion vector resolution)编解码模式中选择性使用相邻块来规定运动矢量预测,其中,所述规则包括:(1)排除位于不同最大编解码单元(LCU)行中的相邻块,或(2)规定固定分辨率值,或(3)在使用仿射模式编解码视频块的情况下,使位于不同LCU行中的仿射编解码(affine-coded)相邻块不可用。

[0039] 在又一示例方面,公开了另一种视频处理方法。该方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻块来规定运动矢量预测,其中,所述规则包括:选择性地使用相邻块,如:(a)在相邻块位于不同最大编解码单元(LCU)行的情况下,禁止将相邻视频块的信息用于视频块的上下文建模(context modeling),或(b)在转换期间仅使用一个上下文来编解码语法元素,或(c)将旁路编解码应用于仅一个语法元素。

[0040] 在又一示例方面,公开了另一种视频处理方法。该方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻视频块来规定运动矢量预测,其中,所述规则包括:基于用于解码当前视频块的相邻视频块,将关于相邻视频块的信息用于上下文建模。

[0041] 在又一示例方面,公开了另一种视频处理方法。该方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻视频块来规定运动矢量预测,其中,所述规则包括:基于用于解码当前视频块的相邻视频块,将关于相邻视频块的信息用于上下文建模,所使用的相邻视频块不包括与当前CTU行不同的CTU行中的、或者与当前CTU不同的CTU中的视频块。

[0042] 在另一示例方面,上述方法可由包括处理器的视频编解码器装置来实现。

[0043] 在又一示例方面,这些方法可以以处理器可执行指令的形式实现并存储在计算机可读程序介质上。

[0044] 将在本文中进一步描述这些方面和其他方面。

附图说明

[0045] 图1示出了Merge候选列表构造的推导处理的示例。

[0046] 图2示出了空域Merge候选的位置的示例。

[0047] 图3示出了考虑空域Merge候选的冗余检查的候选对的示例。

[0048] 图4示出了 $N \times 2N$ 和 $2N \times N$ 分割的第二PU的位置的示例。

[0049] 图5示出了时域Merge候选的运动矢量缩放的示例。

[0050] 图6示出了时域Merge候选的候选位置(C0和C1)的示例。

[0051] 图7示出组合双向预测Merge候选的示例。

[0052] 图8示出了运动矢量预测候选的推导处理的示例。

[0053] 图9示出了空域运动矢量候选的运动矢量缩放的示例。

[0054] 图10示出了CU的ATMVP运动预测的示例。

[0055] 图11示出了具有四个子块(A-D)及其相邻块(a-d)的一个CU示例。

[0056] 图12示出了不同MV精度的编解码流程图的示例。

[0057] 图13示出了将CU划分成两个三角形预测单元(两种分割类型)的示例。

[0058] 图14示出了相邻块的位置的示例。

[0059] 图15示出了应用第一加权因子组的CU的示例。

[0060] 图16示出了运动矢量存储的示例。

[0061] 图17示出了在TPM标志编解码中用于上下文选择的相邻块(A和L)的示例。

[0062] 图18示出了应用OBMC的子块的示例。

[0063] 图19示出了用于导出IC参数的相邻样点的示例。

[0064] 图20示出了简化仿射运动模型的示例。

[0065] 图21示出了每个子块的仿射MVF的示例。

[0066] 图22示出了4参数仿射模型和6参数仿射模型的示例。

[0067] 图23示出了AF_INTER的MVP的示例。

[0068] 图24示出了AF_Merge的候选的示例。

[0069] 图25示出了仿射Merge模式的候选位置的示例。

[0070] 图26示出了光流轨迹的示例。

[0071] 图27示出了具有/不具有块扩展的BIO的示例:a)块外的访问位置;b)使用填充来避免额外的存储器访问和计算。

[0072] 图28示出了基于双边模板匹配的DMVR的示例。

[0073] 图29示出了一个块内的多个几何分割的示例。

[0074] 图30示出了用于TPM标志的上下文选择的相邻块的示例。

[0075] 图31是视频处理装置的示例的框图。

[0076] 图32示出了视频编解码器的示例实现的框图。

[0077] 图33至41示出了视频处理方法的示例的流程图。

- [0078] 图42示出了用于上下文建模的相邻元素的示例。
- [0079] 图43示出了用于导出pred_mode_flag的上下文的相邻块的示例。
- [0080] 图44至63示出了视频处理方法的示例的流程图。
- [0081] 图64是可以实现所公开的技术的示例性视频处理系统的框图。

具体实施方式

[0082] 本文提供了可以由视频比特流的解码器使用以提高解压缩或解码的数字视频的质量的各种技术。此外,视频编解码器也可以在编解码处理期间中实现这些技术,以便重构用于进一步编解码的解码帧。

[0083] 为了便于理解,在本文中使用的章节标题,并且不将实施例和技术限制在相应的章节中。因此,可以将来自一个章节的实施例与来自其他章节的实施例组合。

[0084] 1. 综述

[0085] 本发明涉及视频编解码技术。具体来说,涉及视频编解码中的几何分割下的运动矢量编解码。其可以应用于如高效视频编解码(HEVC)的现有的视频编解码标准,或待定的标准(多功能视频编解码)。它也可适用于未来的视频编解码标准或视频编解码器。

[0086] 2. 讨论

[0087] 视频编解码标准主要通过开发众所周知的ITU-T和ISO/IEC标准而发展起来的。ITU-T制作了H.261和H.263,ISO/IEC制作了MPEG-1和MPEG-4视频,并且这两个组织共同制作了H.262/MPEG-2视频和H.264/MPEG-4高级视频编解码(AVC)和H.265/HEVC标准。自H.262开始,视频编解码标准基于混合视频编解码结构,其中利用了时域预测加变换编解码。为探索HEVC之外的未来视频编解码技术,VCEG和MPEG于2015年共同成立了联合视频探索团队(JVET)。从那时起,JVET采用了许多新的方法,并将其应用到了名为联合探索模型(JEM)的参考软件中。2018年4月,VCEG(Q6/16)和ISO/IEC JTC1 SC29/WG11(MPEG)之间的联合视频专家团队(JVET)成立,以致力于目标是与HEVC相比其降低50%比特率的多功能视频编解码(VVC)标准。

[0088] 图32是视频编解码器的示例实现的框图。图32示出了具有内置的反馈路径的编解码器实现,其中视频编解码器还执行视频解码功能(重构视频数据的压缩表示以用于下一视频数据的编解码)。

[0089] 2.1 HEVC/H.265中的帧间预测

[0090] 每个帧间预测的PU都有一个或两个参考图片列表的运动参数。运动参数包括运动矢量和参考图片索引。两个参考图片列表之一的使用也可以使用inter_pred_idc发信令通知。运动矢量可显式地编解码为相对于预测器的增量。

[0091] 当CU采用跳过(skip)模式编解码时,一个PU与CU相关联,并且没有显著的残差系数,没有编解码运动矢量增量或参考图片索引。规定了一种Merge模式,通过该模式,可以从相邻的PU(包括空域和时域候选)中获取当前PU的运动参数。Merge模式可以应用于任何帧间预测的PU,而不仅仅是跳过模式。Merge模式的另一种选择是运动参数的显式传输,其中运动矢量(更精确地,与运动矢量预测器相比的运动矢量残差)、每个参考图片列表的对应的参考图片索引、以及参考图片列表的使用都会根据每个PU显式地发信令通知。在本公开中,这样的模式被命名为高级运动矢量预测(AMVP)。

[0092] 当信令指示要使用两个参考图片列表中的一个时,从一个样点块中生成PU。这被称为“单向预测”。单向预测对P条带(slice)和B条带都可用。

[0093] 当信令指示要使用两个参考图片列表时,从两个样点块中生成PU。这被称为“双向预测”。双向预测仅对B条带可用。

[0094] 下文提供了在HEVC中规定的关于帧间预测模式的细节。描述将从Merge模式开始。

[0095] 2.1.1参考图片列表

[0096] 在HEVC中,术语帧间预测被用来表示从参考图片而不是当前解码图片的数据元素(例如,样点值或运动矢量)导出的预测。如在H.264/AVC中,可以从多个参考图片预测图片。将用于帧间预测的参考图片组织在一个或多个参考图片列表中。参考索引识别列表中的哪个参考图片应当被用于创建预测信号。

[0097] 将单个参考图片列表(列表0)用于P条带,并且将两个参考图片列表(列表0和列表1)用于B条带。应当注意的是,包括在列表0/1中的参考图片可以按照捕获/显示顺序来自于过去和未来的图片。

[0098] 2.1.2Merge模式

[0099] 2.1.2.1Merge模式的候选的推导

[0100] 当使用Merge模式预测PU时,从比特流中解析指向Merge候选列表中条目的索引,并且使用该索引检索运动信息。在HEVC标准中规定了此列表的构建,并且可按以下步骤顺序进行概括:

[0101] 步骤1:初始候选推导

[0102] 步骤1.1:空域候选推导

[0103] 步骤1.2:空域候选冗余检查

[0104] 步骤1.3:时域候选推导

[0105] 步骤2:附加候选插入

[0106] 步骤2.1:创建双向预测候选

[0107] 步骤2.2:插入零运动候选

[0108] 在图1中也示意性地示出了这些步骤。对于空域Merge候选推导,在位于五个不同位置的候选中最多选择四个Merge候选。对于时域Merge候选推导,在两个候选中最多选择一个Merge候选。由于在解码器处假定每个PU的候选数为常量,因此当从步骤1获得的候选数未达到条带标头中信令通知的最大Merge候选数(maxNumMergeCand)时,生成附加的候选。由于候选数是恒定的,所以最佳Merge候选的索引使用截断一元二值化(TU)进行编解码。如果CU的尺寸等于8,则当前CU的所有PU都共享一个Merge候选列表,这与 $2N \times 2N$ 预测单元的Merge候选列表相同。

[0109] 在下文中,详细描述与前述步骤相关联的操作。

[0110] 图1示出了Merge候选列表构造的推导处理的示例。

[0111] 2.1.2.2空域候选推导

[0112] 在空域Merge候选的推导中,在位于图2所示位置的候选中最多选择四个Merge候选。推导顺序为 A_1 、 B_1 、 B_0 、 A_0 和 B_2 。只有当位置 A_1 、 B_1 、 B_0 、 A_0 的任何PU不可用(例如,因为它属于另一个条带或片)或是帧内编解码时,才考虑位置 B_2 。在增加 A_1 位置的候选后,对剩余候选的增加进行冗余检查,其确保具有相同运动信息的候选被排除在列表之外,从而提高编解

码效率。为了降低计算的复杂度,在所提到的冗余检查中并不考虑所有可能的候选对。相反,只有与图3中的箭头链接的对才会被考虑,并且只有当用于冗余检查的对应候选没有相同的运动信息时,才将候选添加到列表中。复制运动信息的另一个来源是与 $2N \times 2N$ 不同的分割相关的“第二PU”。例如,图4分别描述了 $N \times 2N$ 和 $2N \times N$ 情况下的第二PU。当当前的PU被划分为 $N \times 2N$ 时,对于列表构建不考虑A1位置的候选。实际上,通过添加此候选可能导致两个具有相同运动信息的预测单元,这对于在编解码单元中仅具有一个PU是冗余的。同样地,当当前PU被划分为 $2N \times N$ 时,不考虑位置 B_1 。

[0113] 图2示出了空域Merge候选的位置的示例。

[0114] 图3示出了考虑空域Merge候选的冗余检查的候选对的示例。

[0115] 图4示出了 $N \times 2N$ 和 $2N \times N$ 分割的第二PU的位置的示例。

[0116] 2.1.2.3时域候选推导

[0117] 在此步骤中,只有一个候选添加到列表中。特别地,在这个时域Merge候选的推导中,基于属于与给定参考图片列表中当前图片具有最小POC差的图片的同位PU(co-located PU)导出了缩放运动矢量。用于推导同位PU的参考图片列表在条带标头中显式地发信令通知。获取时域Merge候选的缩放运动矢量(如图5中虚线所示),其使用POC距离 t_b 和 t_d 从同位PU的运动矢量进行缩放,其中 t_b 定义为当前图片的参考图片和当前图片之间的POC差,并且 t_d 定义为同位图片的参考图片与同位图片之间的POC差。时域Merge候选的参考图片索引设置为零。HEVC规范中描述了缩放处理的实际实现。对于B条带,得到两个运动矢量(一个是对于参考图片列表0,另一个是对于参考图片列表1)并将其组合使其成为双向预测Merge候选。

[0118] 图5示出了时域Merge候选的运动矢量缩放的示例。

[0119] 在属于参考帧的同位PU(Y)中,在候选 C_0 和 C_1 之间选择时域候选的位置,如图6所示。如果位置 C_0 处的PU不可用、帧内编解码或在当前CTU行之外,则使用位置 C_1 。否则,位置 C_0 被用于时域Merge候选的推导。

[0120] 图6示出了时域Merge候选的候选位置的示例, C_0 和 C_1 。

[0121] 2.1.2.4附加候选插入

[0122] 除了空域和时域Merge候选,还有两种附加类型的Merge候选:组合双向预测Merge候选和零Merge候选。组合双向预测Merge候选是利用空域和时域Merge候选生成的。组合双向预测Merge候选仅用于B条带。通过将初始候选的第一参考图片列表运动参数与另一候选的第二参考图片列表运动参数相结合,生成组合双向预测候选。如果这两个元组提供不同的运动假设,它们将形成新的双向预测候选。作为示例,图7示出了该情形,其中原始列表(在左侧)中具有MVL0和refIdxL0或MVL1和refIdxL1的两个候选被用于创建添加到最终列表(在右侧)中的组合双向预测Merge候选。存在许多关于被认为生成这些附加Merge候选的组合的规则。

[0123] 图7示出了组合双向Merge候选的示例。

[0124] 插入零运动候选以填充Merge候选列表中的其余条目,从而达到MaxNumMergeCand的容量。这些候选具有零空域位移和从零开始并且每次将新的零运动候选添加到列表中时都会增加的参考图片索引。这些候选使用的参考帧的数目对于单向预测和双向预测分别是1帧和2帧。最后,对这些候选不执行冗余检查。

[0125] 2.1.2.5并行处理的运动估计区域

[0126] 为了加快编解码处理,可以并行执行运动估计,从而同时导出给定区域内所有预测单元的运动矢量。从空域邻域导出Merge候选可能会干扰并行处理,因为一个预测单元在完成相关运动估计之前无法从邻近的PU导出运动参数。为了缓和编解码效率和处理延迟之间的平衡,HEVC使用“log2_parallel_merge_level_minus2”语法元素[1]定义了其尺寸在图片参数集中被信令通知的运动估计区域(MER)。当定义MER时,落入相同区域的Merge候选标志为不可用,并且因此在列表构建中不考虑。

[0127] 2.1.3 AMVP

[0128] AMVP利用运动矢量与相邻的PU的空时相关性,其用于运动参数的显式传输。对于每个参考图片列表,首先通过检查左上方的时域相邻的PU位置的可用性、去掉多余的候选并且加上零矢量以使候选列表长度恒定来构建运动矢量候选列表。然后,编解码器可以从候选列表中选择最佳的预测值,并发送指示所选候选的对应索引。与Merge索引信令类似,最佳运动矢量候选的索引使用截断一元码进行编解码。在这种情况下,要编解码的最大值是2(参见图8)。在以下各章节中,提供了关于运动矢量预测候选的推导处理的细节。

[0129] 2.1.3.1AMVP候选的推导

[0130] 图8概括了运动矢量预测候选的推导处理。

[0131] 在运动矢量预测中,考虑了两种类型的运动矢量候选:空域运动矢量候选和时域运动矢量候选。对于空域运动矢量候选的推导,基于位于图2所示的五个不同位置的每个PU的运动矢量最终推导出两个运动矢量候选。

[0132] 对于时域运动矢量候选的推导,从两个候选中选择一个运动矢量候选,这两个候选是基于两个不同的同位位置推导出的。在做出第一个空时候选列表后,移除列表中重复的运动矢量候选。如果潜在候选的数量大于二,则从列表中移除相关联的参考图片列表中参考图片索引大于1的运动矢量候选。如果空时运动矢量候选数小于二,则会在列表中添加附加的零运动矢量候选。

[0133] 2.1.3.2空域运动矢量候选

[0134] 在推导空域运动矢量候选时,在五个潜在候选中最多考虑两个候选,这五个候选来自图2所示位置上的PU,这些位置与运动Merge的位置相同。当前PU左侧的推导顺序定义为 A_0 、 A_1 、以及缩放的 A_0 、缩放的 A_1 。当前PU上侧的推导顺序定义为 B_0 、 B_1 、 B_2 、缩放的 B_0 、缩放的 B_1 、缩放的 B_2 。因此,每侧有四种情况可以用作运动矢量候选,其中两种情况不需要使用空域缩放,并且两种情况使用空域缩放。四种不同的情况概括如下:

[0135] 无空域缩放

[0136] (1) 相同的参考图片列表,并且相同的参考图片索引(相同的POC)

[0137] (2) 不同的参考图片列表,但是相同的参考图片(相同的POC)

[0138] 空域缩放

[0139] (3) 相同的参考图片列表,但是不同的参考图片(不同的POC)

[0140] (4) 不同的参考图片列表,并且不同的参考图片(不同的POC)

[0141] 首先检查无空域缩放的情况,然后检查允许空域缩放的情况。当POC在相邻PU的参考图片与当前PU的参考图片之间不同时,都会考虑空域缩放,而不管参考图片列表如何。如果左侧候选的所有PU都不可用或是帧内编解码,则允许对上述运动矢量进行缩放,以帮助

左侧和上方MV候选的平行推导。否则,不允许对上述运动矢量进行空域缩放。

[0142] 图9示出了空域运动矢量候选的运动矢量缩放的示例。

[0143] 在空域缩放处理中,以与时域缩放类似的方式缩放相邻PU的运动矢量,如图9所示。主要区别是,给出当前PU的参考图片列表和索引作为输入;实际的缩放处理与时域缩放处理相同。

[0144] 2.1.3.3时域运动矢量候选

[0145] 除了参考图片索引的推导外,时域Merge候选的所有推导处理与空域运动矢量候选的推导处理相同(参见图6)。将参考图片索引信令通知给解码器。

[0146] 2.2JEM中的新的帧间预测方法

[0147] 2.2.1基于子CU的运动矢量预测

[0148] 在具有QTBT的JEM中,每个CU对于每个预测方向最多可以具有一组运动参数。通过将大的CU分割成子CU并推导该大CU的所有子CU的运动信息,编解码器中考虑了两种子CU级的运动矢量预测方法。可选时域运动矢量预测(ATMVP)方法(也称为子块时域运动候选预测(SbTMVP))允许每个CU从同位参考图片中多个小于当前CU的块中提取多组运动信息。在空时运动矢量预测(STMVP)方法中,通过利用时域运动矢量预测值和空域邻接运动矢量递归地推导子CU的运动矢量。

[0149] 为了为子CU运动预测的保持更精确的运动场,当前禁用参考帧的运动压缩。

[0150] 图10示出了CU的ATMVP运动预测的示例。

[0151] 2.2.1.1可选时域运动矢量预测

[0152] 在可选时域运动矢量预测(alternative temporal motion vector prediction, ATMVP)方法中,运动矢量时域运动矢量预测(TMVP)是通过从小于当前CU的块中提取多组运动信息(包括运动矢量和参考索引)来修改的。在一些实施例中,子CU为方形 $N \times N$ 块(默认N设置为4)。

[0153] ATMVP分两步预测CU内的子CU的运动矢量。第一步是用所谓的时域矢量识别参考图片中的对应块。参考图片称为运动源图片。第二步是将当前CU划分成子CU,并从每个子CU对应的块中获取运动矢量以及每个子CU的参考索引。

[0154] 在第一步中,参考图片和对应的块由当前CU的空域相邻块的运动信息确定。为了避免相邻块的重复扫描处理,使用当前CU的Merge候选列表中的第一个Merge候选。第一个可用的运动矢量及其相关联的参考索引被设置为时域矢量和运动源图片的索引。这样,在ATMVP中,与TMVP相比,可以更准确地识别对应的块,其中对应的块(有时称为同位块)始终位于相对于当前CU的右下角或中心位置。

[0155] 在第二步中,通过将时域矢量添加到当前CU的坐标中,通过运动源图片中的时域矢量识别子CU的对应块。对于每个子CU,使用其对应块的运动信息(覆盖中心样点的最小运动网格)来推导子CU的运动信息。在识别出对应 $N \times N$ 块的运动信息后,将其转换为当前子CU的运动矢量和参考索引,与HEVC的TMVP方法相同,其中应用运动缩放和其它处理。例如,解码器检查是否满足低延迟条件(例如,当前图片的所有参考图片的POC都小于当前图片的POC),并可能使用运动矢量 MV_x (与参考图片列表X对应的运动矢量)来为每个子CU预测运动矢量 MV_y (X等于0或1且Y等于1-X)。

[0156] 2.2.1.2空时运动矢量预测(spatial-temporal motion vector prediction,

STMVP)

[0157] 在这种方法中,子CU的运动矢量是按照光栅扫描顺序递归推导的。图11说明了该概念。考虑一个 8×8 的CU,它包含四个 4×4 的子CU A、B、C和D。当前帧中相邻的 4×4 的块标志为a、b、c和d。

[0158] 子CU A的运动推导由识别其两个空域邻居开始。第一个邻居是子CU A上方的 $N \times N$ 块(块c)。如果该块c不可用或是帧内编解码,则检查子CU A上方的其它 $N \times N$ 块(从左到右,从块c处开始)。第二个邻居是子CU A左侧的一个块(块b)。如果块b不可用或是帧内编解码,则检查子CU A左侧的其它块(从上到下,从块b处开始)。每个列表从相邻块获得的运动信息被缩放到给定列表的第一个参考帧。接下来,按照HEVC中规定的与TMVP推导相同的程序,推导子块A的时域运动矢量预测(TMVP)。提取位置D处的同位块的运动信息并进行对应的缩放。最后,在检索和缩放运动信息后,对每个参考列表分别平均所有可用的运动矢量(最多3个)。将平均运动矢量规定为当前子CU的运动矢量。

[0159] 2.2.1.3子CU运动预测模式信令通知

[0160] 子CU模式作为附加的Merge候选启用,并且不需要附加的语法元素来对该模式发信令。将另外两个Merge候选添加到每个CU的Merge候选列表中,以表示ATMVP模式和STMVP模式。如果序列参数集指示启用了ATMVP和STMVP,则最多使用七个Merge候选。附加Merge候选的编解码逻辑与HM中的Merge候选的编解码逻辑相同,这意味着对于P条带或B条带中的每个CU,需要对两个附加Merge候选进行两次更多RD检查。

[0161] 在JEM中,Merge索引的所有bin都由CABAC进行上下文编解码。然而在HEVC中,只有第一个bin是上下文编解码的,并且其余的bin是上下文旁路编解码的。

[0162] 2.2.2自适应运动矢量残差分辨率(adaptive motion vector difference resolution)

[0163] 在HEVC中,当在条带标头中use_integer_mv_flag等于0时,运动矢量残差(MVD)(在PU的运动矢量和预测运动矢量之间)以四分之一亮度样点为单位发信令通知。在JEM中,引入了局部自适应运动矢量分辨率(LAMVR)。在JEM中,MVD可以用四分之一亮度样点、整数亮度样点或四亮度样点的单位进行编解码。MVD分辨率控制在编解码单元(CU)级别,并且MVD分辨率标志有条件地为每个至少有一个非零MVD分量的CU发信令。

[0164] 对于具有至少一个非零MVD分量的CU,发信令通知第一个标志以指示CU中是否使用四分之一亮度样点MV精度。当第一个标志(等于1)指示不使用四分之一亮度样点MV精度时,发信令通知另一个标志以指示是使用整数亮度样点MV精度还是使用四亮度样点MV精度。

[0165] 当CU的第一个MVD分辨率标志为零或没有为CU编解码(意味着CU中的所有MVD都为零)时,CU使用四分之一亮度样点MV分辨率。当一个CU使用整数亮度样点MV精度或四亮度样点MV精度时,该CU的AMVP候选列表中的MVP将取整到对应的精度。

[0166] 在编解码器中,CU级别的RD检查用于确定哪个MVD分辨率将用于CU。也就是说,对每个MVD分辨率执行三次CU级别的RD检查。为了加快编解码器速度,在JEM中应用以下编解码方案。

[0167] 在对具有正常四分之一亮度样点MVD分辨率的CU进行RD检查期间,存储当前CU(整数亮度样点精度)的运动信息。在对具有整数亮度样点和4亮度样点MVD分辨率的相同个CU

进行RD检查时,将存储的运动信息(取整后)用作进一步小范围运动矢量细化的起始点,从而使耗时的运动估计处理不会重复三次。

[0168] 有条件地调用具有4亮度样点MVD分辨率的CU的RD检查。对于CU,当整数亮度样点MVD分辨率的RD检查代价远大于四分之一亮度样点MVD分辨率的RD检查代价时,将跳过对CU的4亮度样点MVD分辨率的RD检查。

[0169] 编解码处理如图12所示。首先,测试1/4像素MV,并且计算RD代价(RD cost)并表示为RDCost0,然后测试整数MV,并且将RD代价表示为RDCost1。如果 $RDCost1 < th * RDCost0$ (其中th为正值),则测试4像素MV;否则,跳过4像素MV。基本上,在检查整数或4像素MV时,1/4像素MV的运动信息和RD代价等都是已知的,其可以被重用来加速整数或4像素MV的编解码处理。

[0170] 2.2.3三角预测模式(triangular prediction mode)

[0171] 三角预测模式(TPM)的概念是引入一种用于运动补偿预测的新的三角分割。如图13所示,在沿对角线方向或相反的对角线方向上将CU划分成两个三角预测单元。CU中的每个三角预测单元是使用从单向预测候选列表导出的其自身的单向预测运动矢量和参考帧索引进行帧间预测的。在对三角预测单元进行预测后,对对角线边缘进行自适应加权处理。然后,将变换和量化处理应用到整个CU中。应当注意的是,此模式仅适用于跳过(skip)和Merge模式。

[0172] 2.2.3.1TPM的单向预测候选列表

[0173] 单向预测候选列表由五个单向预测运动矢量候选组成。它从七个相邻块导出,包括五个空域相邻块(1到5)和两个时域同位块(6到7),如图14所示。图14示出了相邻块的位置的示例。收集7个相邻块的运动矢量,并将其按照单向预测运动矢量、双向预测运动矢量的L0运动矢量、双向预测运动矢量的L1运动矢量、以及双向预测运动矢量的L0和L1运动矢量的平均运动矢量的顺序放入单向预测候选列表中。如果候选数目小于5,则将零运动矢量添加到列表中。

[0174] 更具体地,涉及以下步骤:

[0175] 在不进行任何修剪操作的情况下,从A1、B1、B0、A0、B2、Co1和Co12(对应于图14中的块1至7)中获取运动候选。

[0176] 设置变量numCurrMergeCand=0。

[0177] 对于从A1、B1、B0、A0、B2、Co1和Co12导出的每个运动候选,并且numCurrMergeCand小于5,如果该运动候选是单向预测(来自列表0或列表1),则将其添加到Merge列表中,numCurrMergeCand增加1。这样添加的附加候选被命名为“原始单向预测预测候选”。

[0178] 应用完全修剪(full pruning)。

[0179] 对于从A1、B1、B0、A0、B2、Co1和Co12导出的每个运动候选,并且numCurrMergeCand小于5,如果该运动候选是双向预测,则将来自列表0的运动信息添加到Merge列表(即,修改为来自列表0的单向预测),并且numCurrMergeCand增加1。这样添加的运动候选被命名为“截断的列表0预测候选”。

[0180] 应用完全修剪。

[0181] 对于从A1、B1、B0、A0、B2、Co1和Co12导出的每个运动候选,并且numCurrMergeCand小于5,如果该运动候选是双向预测,则将来自列表1的运动信息添加到Merge列表(即,修改

为来自列表1的单向预测),并且numCurrMergeCand增加1。这样添加的运动候选被命名为“截断的列表1预测候选”。

[0182] 应用完全修剪。

[0183] 对于从A1、B1、B0、A0、B2、Co1和Co12导出的每个运动候选,并且numCurrMergeCand小于5,如果该运动候选是双向预测,

[0184] 如果列表0参考图片的条带QP小于列表1参考图片的条带QP,则首先将列表1的运动信息缩放到列表0参考图片,并且将两个MV(一个来自原始列表0,并且另一个是来自列表1的缩放MV)的平均添加到Merge列表,其是来自列表0运动候选的平均单向预测,并且numCurrMergeCand增加1。

[0185] 否则,首先将列表0的运动信息缩放到列表1参考图片,并且将两个MV(一个来自原始列表1,并且另一个是来自列表0的缩放MV)的平均添加到Merge列表中,其是来自列表1运动候选的平均单向预测,并且numCurrMergeCand增加1。

[0186] 应用完全修剪。

[0187] 如果numCurrMergeCand小于5,则添加零运动矢量候选。

[0188] 2.2.3.2自适应加权处理(adaptive weighting process)

[0189] 在对每个三角预测单元进行预测后,对两个三角预测单元之间的对角线边缘应用自适应加权处理,以导出整个CU的最终预测。两个加权因子组定义如下:

[0190] 第一加权因子组:分别将{7/8、6/8、4/8、2/8、1/8}和{7/8、4/8、1/8}用于亮度和色度样点;

[0191] 第二加权因子组:分别将{7/8、6/8、5/8、4/8、3/8、2/8、1/8}和{6/8、4/8、2/8}用于亮度和色度样点。

[0192] 在比较两个三角预测单元的运动矢量的基础上,选择一个加权因子组。当两个三角预测单元的参考图片彼此不同、或其运动矢量残差大于16像素时,使用第二加权因子组。否则,使用第一加权因子组。示例如图15所示。图15示出了应用第一加权因子组的CU的示例。

[0193] 2.2.3.3运动矢量存储

[0194] 图16示出了运动矢量存储的示例。三角预测单元的运动矢量(图16中的Mv1和Mv2)存储在 4×4 的网格中。对于每个 4×4 的网格,取决于 4×4 网格在CU中的位置来存储单向预测或双向预测运动矢量。如图16所示,为位于非加权区域(即,不位于对角线边缘)的 4×4 网格存储单向预测运动矢量(Mv1或Mv2)。在另一方面,为位于加权区域的 4×4 网格存储双向预测运动矢量。根据以下规则,从Mv1和Mv2导出双向预测运动矢量:

[0195] 在Mv1和Mv2具有来自不同方向(L0或L1)的运动矢量的情况下,将Mv1和Mv2简单地组合以形成双向预测运动矢量。

[0196] 在Mv1和Mv2都来自相同L0(或L1)方向的情况下,如果Mv2的参考图片与L1(或L0)参考图片列表中的图片相同,则将Mv2缩放到该图片。将Mv1和缩放的Mv2组合以形成双向预测运动矢量。

[0197] 如果Mv1的参考图片与L1(或L0)参考图片列表中的图片相同,则将Mv1缩放到该图片。将缩放后的Mv1和Mv2组合以形成双向预测运动矢量。

[0198] 否则,只为加权区域存储Mv1。

[0199] 2.2.3.4三角预测模式(TPM)的信令

[0200] 首先可以以一个比特标志信令通知指示是否使用TPM。然后,进一步信令通知两个划分模式(如图13所示)的指示、以及两个分割的每一个的选定Merge索引。

[0201] 应当注意的是,将TPM视为特殊的Merge模式的子模式。因此,首先信令通知Merge标志。当Merge标志为真时,可进一步信令通知块是否选择TPM。

[0202] 下表(参照VVC编解码器规范的当前版本)示出了修改当前语法的示例(粗体项示出了修改)。

[0203] 7.3.4.6编解码单元语法

	coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
	if(slice_type != I) {	
[0204]	cu_skip_flag [x0][y0]	ae(v)
	if(cu_skip_flag[x0][y0] == 0)	
	pred_mode_flag	ae(v)
	}	

[0205]

if(CuPredMode[x0][y0] == MODE_INTRA) {	
...	
} else { /* MODE_INTER */	
if(cu_skip_flag[x0][y0] == 0) {	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
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)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
.mvp_l1_flag[x0][y0]	ae(v)
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
if(sps_AMVR_enabled_flag && inter_affine_flag == 0 && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0 MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0))	
AMVR_mode[x0][y0]	ae(v)

	if(sps_gbi_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && cbWidth * cbHeight >= 256)	
	gbi_idx [x0][y0]	ae(v)
	}	
	}	
[0206]	if(!pcm_flag[x0][y0]) {	
	if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
	cu_cbf	ae(v)
	if(cu_cbf)	
	transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
	}	
	}	

[0207] 7.3.4.8 Merge数据语法

		描述符
	merge_data(x0, y0, cbWidth, cbHeight) {	
	MMVD_flag [x0][y0]	ae(v)
	if(MMVD_flag[x0][y0] == 1) {	
	MMVD_merge_flag [x0][y0]	ae(v)
	MMVD_distance_idx [x0][y0]	ae(v)
	MMVD_direction_idx [x0][y0]	ae(v)
	} else {	
	if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
	merge_subblock_flag [x0][y0]	ae(v)
	if(merge_subblock_flag[x0][y0] == 1) {	
	if(MaxNumSubblockMergeCand > 1)	
	merge_subblock_idx [x0][y0]	ae(v)
	} else {	
	if(sps_mh_intra_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
[0208]	mh_intra_flag [x0][y0]	ae(v)
	if(mh_intra_flag[x0][y0]) {	
	if(cbWidth <= 2 * cbHeight cbHeight <= 2 * cbWidth)	
	mh_intra_luma_mpm_flag [x0][y0]	ae(v)
	if(mh_intra_luma_mpm_flag[x0][y0])	
	mh_intra_luma_mpm_idx [x0][y0]	ae(v)
	}	
	}	
	} else if(sps_triangle_enabled_flag && slice_type == B && cbWidth * cbHeight >= 16)	
	merge_triangle_flag [x0][y0]	ae(v)
	if(merge_triangle_flag[x0][y0])	
	merge_triangle_idx [x0][y0]	ae(v)
	else if(MaxNumMergeCand > 1)	
	merge_idx [x0][y0]	ae(v)
	}	
	}	

[0209] }

[0210] 2.2.3.4.1 TPM标志的信令

[0211] 让我们分别用W和H来表示一个亮度块的宽度和高度。如果W*H<64,则禁用三角预

测模式。

[0212] 当一个块用仿射模式编解码时,三角预测模式也被禁用。

[0213] 当一个块用Merge模式编解码时,可以信令通知一个比特标志,以指示对该块是启用还是禁用三角预测模式。

[0214] 基于以下公式,该标志由3个上下文编解码:

[0215] $\text{Ctx_index} = ((\text{左块L可用}\&\&\text{L用TPM编解码?})1:0) + ((\text{上块A可用}\&\&\text{A用TPM编解码?})1:0);$

[0216] 图17示出了在TPM标志编解码中用于上下文选择的相邻块(A和L)的示例。

[0217] 2.2.3.4.2两个划分模式(如图13所示)的指示、以及两个分割中的每个的选择Merge索引的信令通知

[0218] 应当注意的是,划分模式、两个分割的Merge索引是联合编解码的。在[5]中,限制了两个分割不能使用相同的参考索引。因此,存在 $2(\text{划分模式}) * N(\text{Merge候选的最大数目}) * (N-1)$ 种可能性,其中N被设置为5。一个指示被编解码,并且从以下定义的阵列导出分割类型、两个Merge索引之间的映射:

[0219] `const uint8_t g_TriangleCombination[TRIANGLE_MAX_NUM_CANDS][3] = {`

[0220] `{0,1,0}, {1,0,1}, {1,0,2}, {0,0,1}, {0,2,0},`

[0221] `{1,0,3}, {1,0,4}, {1,1,0}, {0,3,0}, {0,4,0},`

[0222] `{0,0,2}, {0,1,2}, {1,1,2}, {0,0,4}, {0,0,3},`

[0223] `{0,1,3}, {0,1,4}, {1,1,4}, {1,1,3}, {1,2,1},`

[0224] `{1,2,0}, {0,2,1}, {0,4,3}, {1,3,0}, {1,3,2},`

[0225] `{1,3,4}, {1,4,0}, {1,3,1}, {1,2,3}, {1,4,1},`

[0226] `{0,4,1}, {0,2,3}, {1,4,2}, {0,3,2}, {1,4,3},`

[0227] `{0,3,1}, {0,2,4}, {1,2,4}, {0,4,2}, {0,3,4}};`

[0228] 分割类型(45度或135度) = `g_TriangleCombination[信令通知的指示][0];`

[0229] 候选A的Merge索引 = `g_TriangleCombination[信令通知的指示][1];`

[0230] 候选B的Merge索引 = `g_TriangleCombination[信令通知的指示][2];`

[0231] 一旦导出两个运动候选A和B,就可以从A或B设置两个分割(PU1和PU2)的运动信息。PU1是否使用Merge运动候选A或B的运动信息取决于两个运动候选的预测方向。表1示出了具有两个分割的两个导出的运动候选A和之间的关系。

[0232] 表1:从导出的两个Merge候选(A,B)推导分割运动信息

[0233]

A的预测方向	B的预测方向	PU1的运动信息	PU2的运动信息
L0	L0	A(L0)	B(L0)
L1	L1	B(L1)	A(L1)
L0	L1	A(L0)	B(L1)
L1	L0	B(L0)	A(L1)

[0234] 2.2.3.4.3指示的熵编解码(用merge_triangle_idx表示)

[0235] merge_triangle_idx在[0,39]范围内,包括0和39。K阶指数哥伦布(EG)编解码被用于merge_triangle_idx的二值化,其中K设为1。

[0236] K阶EG

[0237] 为了用较少的比特编解码较大的数字(以使用较多的比特来编解码较小的数字为代价),可以使用非负整数参数 k 来进行归一化。要在 K 阶指数哥伦布编解码中编解码非负整数 x :

[0238] 使用上述 0 阶指数哥伦布编解码对 $\lfloor x/2^k \rfloor$ 进行编解码,然后,

[0239] 在二进制中编解码 $x \bmod 2^k$ 。

[0240] 表2: k 阶指数哥伦布编解码的示例

	x	$k=0$	$k=1$	$k=2$	x	$k=0$	$k=1$	$k=2$
	0	1	10	100	10	0001011	001100	01110
	1	010	11	101	11	0001100	001101	01111
	2	011	0100	110	12	0001101	001110	0010000
[0241]	3	00100	0101	111	13	0001110	001111	0010001
	4	00101	0110	01000	14	0001111	00010000	0010010
	5	00110	0111	01001	15	000010000	00010001	0010011
	6	00111	001000	01010	16	000010001	00010010	0010100
	7	0001000	001001	01011	17	000010010	00010011	0010101
	8	0001001	001010	01100	18	000010011	00010100	0010110
	9	0001010	001011	01101	19	000010100	00010101	0010111

[0242] 2.2.4重叠块运动补偿(overlapped block motion compensation)

[0243] 先前在H.263中已经使用了重叠块运动补偿(OBMC)。在JEM中,不同于在H.263中,可以使用CU级别的语法打开和关闭OBMC。当OBMC用于JEM时,针对所有运动补偿(MC)块边界执行OBMC,但CU的右边界和底边界除外。此外,它还适用于亮度和色度分量(luma and chroma components)。在JEM中,MC块对应于编解码块。当CU用于子CU模式(包括子CU MERGE、仿射和FRUC模式)编解码时,CU的每个子块都是MC块。为了用统一的方式处理CU边界,对所有MC块边界,在子块级别执行OBMC,其中子块尺寸设置为等于 4×4 ,如图18所示。

[0244] 当OBMC应用于当前子块时,除了当前运动矢量外,四个相连的相邻子块的运动矢量(如果可用且与当前运动矢量不同)也可用于推导当前子块的预测块。将这些基于多个运动矢量的多个预测块组合起来,以生成当前子块的最终预测信号。

[0245] 基于相邻子块运动矢量的预测块表示为 P_N ,其中 N 表示相邻的上、下、左、右子块的索引,并且基于当前子块运动矢量的预测块表示为 P_C 。当 P_N 基于包含与当前子块相同的运动信息的相邻子块的运动信息时,OBMC不从 P_N 处执行的。否则,每个 P_N 的样点都添加到 P_C 中的相同样点中,即将 P_N 的四行/列添加到 P_C 。 P_N 使用加权因子 $\{1/4, 1/8, 1/16, 1/32\}$, P_C 使用加权因子 $\{3/4, 7/8, 15/16, 31/32\}$ 。例外情况是小MC块(即编解码块的高度或宽度等于4或CU是子CU模式编解码的),对此在 P_C 中只添加 P_N 的两行/列。在这种情况下, P_N 使用加权因子 $\{1/4, 1/8\}$, P_C 使用加权因子 $\{3/4, 7/8\}$ 。对于基于垂直(水平)相邻子块的运动矢量生成的 P_N ,将 P_N 的相同行(列)中的样点以相同的加权因子添加到 P_C 中。

[0246] 在JEM中,对于尺寸小于或等于256亮度样点的CU,会对CU级别标志发信令,以指示当前CU是否应用OBMC。对于尺寸大于256亮度样点或未使用AMVP模式编解码的CU,默认情况下应用OBMC。在编解码器处,当OBMC应用于CU时,在运动估计阶段会考虑其影响。使用上相邻块和左相邻块的运动信息通过OBMC形成的预测信号被用来补偿当前CU的原始信号的上边界和左边界,并且然后应用正常的运动估计处理。

[0247] 2.2.5局部照明补偿(local illumination compensation)

[0248] 局部照明补偿(LIC)基于用于照明变化的线性模型,使用比例因子a和偏移量b。并且其对每个帧间模式编解码的编解码单元(CU)自适应地启用或禁用。图19示出了用于导出IC参数的相邻块的示例。

[0249] 当LIC应用于CU时,采用最小二乘误差法,通过使用当前CU的相邻样点及其对应的参考样点导出参数a和b。更具体地,如图19所示,使用参考图片中CU的子采样(2:1子采样)相邻样点和对应的样点(由当前CU或子CU的运动信息识别)。推导出IC参数并将其分别应用于每个预测方向。

[0250] 当使用Merge模式对CU进行编解码时,LIC标志以类似于Merge模式下运动信息复制的方式从相邻块复制;否则,将为CU信令通知LIC标志,以指示LIC是否适用。

[0251] 当为图片启用LIC时,需要附加的CU级RD检查来确定是否为CU应用LIC。当对CU启用LIC时,对于整数像素运动搜索和分数像素运动搜索,分别使用绝对差的平均去除和(MR-SAD)和绝对Hadamard变换差的平均去除和(MR-SATD),而不使用SAD和SATD。

[0252] 为了降低编解码复杂度,在JEM中应用以下编解码方案:

[0253] 当当前图片与其参考图片之间没有明显的照度变化时,对整个图片禁用LIC。为了识别这种情况,在编解码器处计算当前图片的直方图和当前图片的每个参考图片的直方图。如果当前图片和当前图片的每个参考图片之间的直方图差小于给定阈值,则对当前图片禁用LIC;否则,对当前图片启用LIC。

[0254] 2.2.6仿射运动补偿预测(affine motion compensation prediction)

[0255] 在HEVC中,运动补偿预测(MCP)仅应用平移运动模型。然而,在真实世界中可能存在多种运动,例如放大/缩小、旋转、透视运动和其他不规则运动。在JEM中应用了简化的仿射变换运动补偿预测。如图20所示,用两个控制点运动矢量描述块的仿射运动场。

[0256] 块的运动矢量场(Motion Vector Field,MVF)由以下等式描述:

$$[0257] \quad \begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases} \quad (1-a)$$

[0258] 对于6参数仿射,

$$[0259] \quad \begin{cases} v_x = \frac{v_{1x} - v_{0x}}{w} x + \frac{v_{2x} - v_{0x}}{h} y + v_{0x} \\ v_y = \frac{v_{1y} - v_{0y}}{w} x + \frac{v_{2y} - v_{0y}}{h} y + v_{0y} \end{cases} \quad (1-b)$$

[0260] 其中, (v_{0x}, v_{0y}) 是左上角控制点的运动矢量,并且 (v_{1x}, v_{1y}) 是右上角控制点的运动矢量。并且 (v_{2x}, v_{2y}) 为左下角控制点的运动矢量, (x, y) 表示代表点相对于当前块内的左上角样点的坐标。在VTM中,代表点被定义为子块的中心位置,例如,当子块的左上角相对于当前块内的左上角样点的坐标为 (x_s, y_s) 时,代表点的坐标被定义为 (x_s+2, y_s+2) 。

[0261] 为了进一步简化运动补偿预测,应用基于子块的仿射变换预测。如等式2推导子块尺寸 $M \times N$, 其中 $MvPre$ 是运动矢量分数精度(例如, JEM中的 $1/16$)。 (v_{2x}, v_{2y}) 是左下控制点的运动矢量,其根据等式1计算。

$$[0262] \quad \begin{cases} M = \text{clip3} \left(4, w, \frac{w \times MvPre}{\max(\text{abs}(v_{1x} - v_{0x}), \text{abs}(v_{1y} - v_{0y}))} \right) \\ N = \text{clip3} \left(4, h, \frac{h \times MvPre}{\max(\text{abs}(v_{2x} - v_{0x}), \text{abs}(v_{2y} - v_{0y}))} \right) \end{cases} \quad (2)$$

[0263] 用等式2推导出之后,如果需要,M和N应当被向下调整使其分别作为w和h的除数(divisor)。

[0264] 为了推导出每个M×N子块的运动矢量,如图21所示,可以根据等式1计算每个子块的中心样点的运动矢量,并且舍入到1/16分数精度。然后应用本文其它地方所述的运动补偿插值滤波器,利用推导出的运动矢量生成各子块的预测。

[0265] 在MCP之后,对每个子块的高精度运动矢量进行舍入,并将其保存为与正常运动矢量相同的精度。

[0266] 2.2.6.1 AF_INTER模式

[0267] 在JEM中,有两个仿射运动模式:AF_INTER模式和AF_Merge模式。对于宽度和高度都大于8的CU,可以应用AF_INTER模式。在比特流中,CU级别的仿射标志被信令通知,以指示是否使用AF_INTER模式。在这种模式中,使用相邻的块构建具有运动矢量对 $\{(v_0, v_1) \mid v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$ 的候选列表。如图23所示,从块A、B或C的运动矢量中选择 v_0 。根据参考列表,以及相邻块参考的POC、当前CU参考的POC和当前CU的POC之间的关系对来自相邻块的运动矢量进行缩放。从相邻块D和E中选择 v_1 的方法类似。当候选列表的数目小于2时,该列表由复制每个AMVP候选组成的运动矢量对来填充。当候选列表大于2时,可以首先根据相邻运动矢量的一致性对候选进行排序(一对候选中两个运动矢量的相似性),并且只保留前两个候选。使用RD代价检查来确定选择哪个运动矢量对候选作为当前CU的控制点运动矢量预测(Control Point Motion Vector Prediction, CPMVP)。并且在比特流中信令通知指示CPMVP在候选列表中的位置的索引。在确定了当前仿射CU的CPMVP后,应用仿射运动估计,并且找到控制点运动矢量(Control Point Motion Vector, CPMV)。然后,在比特流中信令通知CPMV和CPMVP的差。

[0268] 在AF_INTER模式中,当使用4/6参数仿射模式时,需要2/3个控制点,并且因而需要对这些控制点编解码2/3个MVD,如图22所示。在一些实施例中,提出了如下导出MV,即,从mvd0预测mvd1和mvd2。

$$[0269] \quad mv_0 = \overline{mv}_0 + mvd_0$$

$$[0270] \quad mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

$$[0271] \quad mv_2 = \overline{mv}_2 + mvd_2 + mvd_0$$

[0272] 其中, \overline{mv}_i , mvd_i 和 mv_i 分别是左上像素($i=0$)、右上像素($i=1$)或左下像素($i=2$)的预测运动矢量、运动矢量残差和运动矢量,如图22右侧所示。请注意,两个运动矢量的相加(例如, $mv_A(x_A, y_A)$ 和 $mv_B(x_B, y_B)$) 分别等于两个分量的总和,即 $newMV = mv_A + mv_B$, 并且将 $newMV$ 的两个分量分别设为 $(x_A + x_B)$ 和 $(y_A + y_B)$ 。

[0273] 2.2.6.2 AF_MERGE模式

[0274] 当在AF_MERGE模式中应用CU时,它从有效的相邻重构块中获得以仿射模式编解码的第一个块。并且候选块的选择顺序是从左、上、右上、左下到左上,如图24左侧所示。如果

相邻的左下块A以仿射模式被编解码,如图24右侧所示,则导出包含块A的CU的左上角、右上角和左下角的运动矢量 v_2 、 v_3 和 v_4 。并且根据 v_2 、 v_3 和 v_4 计算当前CU左上角的运动矢量 v_0 。其次,计算当前CU右上方的运动矢量 v_1 。

[0275] 在导出当前CU的CPMV v_0 和 v_1 之后,根据简化的仿射运动模型等式1,生成当前CU的MVF。为了识别当前CU是否以AF_MERGE模式编解码,当至少一个相邻块以仿射模式编解码时,在比特流中信令通知仿射标志。

[0276] 在一些实施例中,用以下步骤来构造仿射Merge候选:

[0277] 插入继承仿射候选

[0278] 继承仿射候选是指该候选是从其有效相邻仿射编解码块的仿射运动模型导出的。在共同的基础上,如图25所示,候选位置的扫描顺序为:A1、B1、B0、A0和B2。

[0279] 在导出候选之后,执行完全修剪来检查是否已经将相同的候选插入了列表。如果存在相同的候选,则丢弃导出的候选。

[0280] 插入构造仿射候选

[0281] 如果仿射Merge候选列表中的候选数目小于MaxNumAffineCand(本文中设为5),则将构造仿射候选插入候选列表中。构造仿射候选是指该候选是通过组合每个控制点的相邻运动信息构造的。

[0282] 首先从图25所示的规定的空域邻居和时域邻居导出控制点的运动信息。CPk($k=1,2,3,4$)表示第k个控制点。A0,A1,A2,B0,B1,B2和B3是用于预测CPk($k=1,2,3$)的空域位置;T是用于预测CP4的时域位置。

[0283] CP1、CP2、CP3和CP4的坐标分别为(0,0),(W,0),(H,0)和(W,H),其中W和H是当前块的宽度和高度。

[0284] 按照以下优先顺序获得每个控制点的运动信息:

[0285] 对于CP1,检查优先级为B2- \rightarrow B3- \rightarrow A2。如果B2可用,则使用B₂。否则,如果B2不可用,则使用B3。如果B2和B3都不可用,则使用A2。如果三个候选都不可用,则无法获得CP1的运动信息。

[0286] 对于CP2,检查优先级为B1- \rightarrow B0;

[0287] 对于CP3,检查优先级为A1- \rightarrow A0;

[0288] 对于CP4,使用T。

[0289] 其次,使用控制点的组合来构造仿射Merge候选。

[0290] 需要三个控制点的运动信息来构造6参数仿射候选。可以从以下四个组合中的一个来选择这三个控制点({CP1,CP2,CP4},{CP1,CP2,CP3},{CP2,CP3,CP4},{CP1,CP3,CP4})。组合{CP1,CP2,CP3},{CP2,CP3,CP4},{CP1,CP3,CP4}将被转换到由左上、右上和左下控制点表示的6参数运动模型。

[0291] 需要两个控制点的运动信息来构造4参数仿射候选。可以从以下六个组合中的一个来选择这两个控制点({CP1,CP4},{CP2,CP3},{CP1,CP2},{CP2,CP4},{CP1,CP3},{CP3,CP4})。组合{CP1,CP4},{CP2,CP3},{CP2,CP4},{CP1,CP3},{CP3,CP4}将被转换到由左上和右上控制点表示的4参数运动模型。

[0292] 按以下顺序将构造仿射候选的组合插入到候选列表中:

[0293] {CP1,CP2,CP3},{CP1,CP2,CP4},{CP1,CP3,CP4},{CP2,CP3,CP4},{CP1,CP2},

{CP1,CP3}, {CP2,CP3}, {CP1,CP4}, {CP2,CP4}, {CP3,CP4}。

[0294] 对于组合的参考列表X(X为0或1),选择控制点中使用率最高的参考索引作为列表X的参考索引,并且缩放指向不同参考图片的运动矢量。

[0295] 在导出候选之后,执行完全修剪处理,以检查是否已将相同的候选插入到列表中。如果存在相同的候选,则丢弃导出的候选。

[0296] 填补零运动矢量

[0297] 如果仿射Merge候选列表中的候选数目小于5,则在候选列表中插入具有零参考索引的零运动矢量,直到列表满为止。

[0298] 2.2.7双向光流

[0299] 双向光流(BIO)是在块方式运动补偿之上执行的用于双向预测的样点方式运动细化。样点级的运动细化不使用信令。

[0300] 设 $I^{(k)}$ 为块运动补偿后来自参考k($k=0,1$)的亮度值,并且 $\partial I^{(k)}/\partial x, \partial I^{(k)}/\partial y$ 分别为 $I^{(k)}$ 梯度的水平分量和垂直分量。假设光流是有效的,则运动矢量场(v_x, v_y)由等式给出:

$$[0301] \quad \partial I^{(k)}/\partial t + v_x \partial I^{(k)}/\partial x + v_y \partial I^{(k)}/\partial y = 0. \quad (3)$$

[0302] 将此光流等式与每个样点的运动轨迹的埃尔米特(Hermite)插值相结合,得到唯一的三阶多项式,该多项式在末端同时匹配函数值 $I^{(k)}$ 和其导数 $\partial I^{(k)}/\partial x, \partial I^{(k)}/\partial y$ 。该多项式在 $t=0$ 时的值是BIO预测:

$$[0303] \quad pred_{BIO} = 1/2 \cdot (I^{(0)} + I^{(1)} + v_x/2 \cdot (\tau_1 \partial I^{(1)}/\partial x - \tau_0 \partial I^{(0)}/\partial x) + v_y/2 \cdot (\tau_1 \partial I^{(1)}/\partial y - \tau_0 \partial I^{(0)}/\partial y)) \quad (4)$$

[0304] 这里, τ_0 和 τ_1 表示到参考帧的距离,如图26所示。基于Ref0和Ref1的POC计算距离 τ_0 和 τ_1 : $\tau_0 = POC(\text{current}) - POC(\text{Ref}_0)$, $\tau_1 = POC(\text{Ref}_1) - POC(\text{current})$ 。如果两个预测都来自相同个时域方向(都来自过去或都来自未来),则符号是不同的(例如, $\tau_0 \cdot \tau_1 < 0$)。在这种情况下,只有当预测不是来自相同时刻(即, $\tau_0 \neq \tau_1$)、两个参考区域都具有非零运动($MV_{x_0}, MV_{y_0}, MV_{x_1}, MV_{y_1} \neq 0$)并且块运动矢量与时域距离成比例($MV_{x_0}/MV_{x_1} = MV_{y_0}/MV_{y_1} = -\tau_0/\tau_1$)时,才应用BIO。

[0305] 通过最小化A点和B点(图24中运动轨迹与参考帧平面相交)之间的值的差 Δ 来确定运动矢量场(v_x, v_y)。对 Δ ,模型仅使用局部泰勒展开的第一个线性项:

$$[0306] \quad \Delta = (I^{(0)} - I^{(1)}_0 + v_x (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x) + v_y (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y)) \quad (5)$$

[0307] 等式5中的所有值取决于样点位置(i', j'),目前为止从标志中将其忽略。假设在局部周围区域的运动是一致的,可以在以当前预测点(i, j)为中心的 $(2M+1) \times (2M+1)$ 方形窗口 Ω 内最小化 Δ ,其中M等于2:

$$[0308] \quad (v_x, v_y) = \underset{v_x, v_y}{\operatorname{argmin}} \sum_{[i', j'] \in \Omega} \Delta^2 [i', j'] \quad (6)$$

[0309] 对于这个优化问题,JEM使用简化方法,首先在垂直方向上最小化,然后在水平方向最小化。这导致:

$$[0310] \quad v_x = (s_1 + r) > m? \operatorname{clip3}(-thBIO, thBIO, -\frac{s_3}{(s_1+r)}): 0 \quad (7)$$

$$[0311] \quad v_y = (s_5 + r) > m? \text{clip3} \left(-thBIO, thBIO, -\frac{s_6 - v_x s_2 / 2}{(s_5 + r)} \right) : 0 \quad (8)$$

[0312] 其中,

$$[0313] \quad \begin{aligned} s_1 &= \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_3 = \sum_{[i,j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x); \\ s_2 &= \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y); \\ s_5 &= \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_6 = \sum_{[i,j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad (9)$$

[0314] 为了避免除以零或很小的值,在式7和式8中引入正则化参数r和m。

$$[0315] \quad r = 500 \cdot 4^{d-8} \quad (10)$$

$$[0316] \quad m = 700 \cdot 4^{d-8} \quad (11)$$

[0317] 这里,d是视频样点的位深度。

[0318] 为了使BIO的内存访问与常规双向预测运动补偿相同,仅计算当前块内的位置的所有预测和梯度值 $I^{(k)}$, $\partial I^{(k)} / \partial x$, $\partial I^{(k)} / \partial y$,在式11中,以预测块边界上当前预测点为中心的 $(2M+1) \times (2M+1)$ 的方形窗口 Ω 需要访问块外的位置(如图27左侧所示)。在JEM中,块外的值 $I^{(k)}$, $\partial I^{(k)} / \partial x$, $\partial I^{(k)} / \partial y$ 设置为等于块内最近的可用值。例如,这可以实现为填充,如图27右侧所示。

[0319] 使用BIO,可以对每个样点的运动场进行细化。为了降低计算复杂度,在JEM中采用了基于块设计的BIO。基于 4×4 块计算运动细化。在基于块的BIO中,对 4×4 块中所有样点的等式11中的 s_n 值进行聚合,然后将 s_n 的聚合值用于推导 4×4 块的BIO运动矢量偏移。更具体地说,下面的等式被用于基于块的BIO推导:

$$[0320] \quad \begin{aligned} s_{1,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i,j] \in \Omega(x,y)} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_{3,b_k} = \sum_{(x,y) \in b_k} \sum_{[i,j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x); \\ s_{2,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y); \\ s_{5,b_k} &= \sum_{(x,y) \in b_k} \sum_{[i,j] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_{6,b_k} = \sum_{(x,y) \in b_k} \sum_{[i,j] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad (12)$$

[0321] 其中, b_k 表示属于预测块的第k个 4×4 块的样点组。等式7和8中的 s_n 替换为 $((s_{n,b_k}) > 4)$ 以推导相关联的运动矢量偏移。

[0322] 在某些情况下,由于噪声或不规则运动,BIO的MV团(regiment)可能不可靠。因此,在BIO中,MV团的幅度被限幅到阈值thBIO。该阈值是基于当前图片的参考图片是否全部来自一个方向确定的。如果当前图片的所有参考图片都来自一个方向,则该阈值的值被设置为 $12 \times 2^{14-d}$,否则其被设置为 $12 \times 2^{13-d}$ 。

[0323] 使用与HEVC运动补偿处理一致的操作(2D可分离FIR),与运动补偿插值同时计算BIO的梯度。该2D可分离FIR的输入是与运动补偿处理相同的参考帧样点,以及根据块运动矢量的分数部分的分数位置(fracX, fracY)。如果是水平梯度 $\partial I / \partial x$,首先使用BIOfilterS对信号进行垂直插值,该BIOfilterS对应于具有去缩放位移 $d-8$ 的分数位置fracY。然后在水平方向上应用梯度滤波器BIOfilterG,该BIOfilterG对应于具有去缩放位

移18-d的分数位置fracX。如果是垂直梯度 $\partial I/\partial x$ ，首先使用BIOfilterG垂直地应用梯度滤波器，该BIOfilterG对应于具有去缩放位移d-8的分数位置fracY。然后在水平方向上使用BIOfilterS执行信号移位，该BIOfilterS对应于具有去缩放位移18-d的分数位置fracX。用于梯度计算BIOfilterG和信号移位BIOfilterS的插值滤波器的长度可以更短(6-tap)，以保持合理的复杂度。表3示出了用在BIO中块运动矢量的不同分数位置的梯度计算的示滤波器。表4示出了用在BIO中预测信号生成的插值滤波器。

[0324] 表3 BIO中用于梯度计算的滤波器

分数像素位置	梯度的插值滤波器 (BIOfilterG)
0	{8, -39, -3, 46, -17, 5}
1/16	{8, -32, -13, 50, -18, 5}
1/8	{7, -27, -20, 54, -19, 5}
3/16	{6, -21, -29, 57, -18, 5}
1/4	{4, -17, -36, 60, -15, 4}
5/16	{3, -9, -44, 61, -15, 4}
3/8	{1, -4, -48, 61, -13, 3}
7/16	{0, 1, -54, 60, -9, 2}
1/2	{-1, 4, -57, 57, -4, 1}

[0326] 表4 BIO中用于预测信号生成的插值滤波器

分数像素位置	用于预测信号的插值滤波器 (BIOfilterS)
0	{0, 0, 64, 0, 0, 0}
1/16	{1, -3, 64, 4, -2, 0}
1/8	{1, -6, 62, 9, -3, 1}
3/16	{2, -8, 60, 14, -5, 1}
1/4	{2, -9, 57, 19, -7, 2}
5/16	{3, -10, 53, 24, -8, 2}
3/8	{3, -11, 50, 29, -9, 2}
7/16	{3, -11, 44, 35, -10, 3}
1/2	{3, -10, 35, 44, -11, 3}

[0328] 在JEM中，当两个预测来自不同的参考图片时，BIO应用于所有的双向预测块。当为CU启用LIC时，禁用BIO。

[0329] 在JEM中，在正常MC处理之后将OBMC应用于块。为了降低计算复杂度，在OBMC处理期间不应用BIO。这意味着在OBMC处理期间，当使用自己的MV时，才将BIO应用于块的MC处理，而当使用相邻块的MV时，不将BIO应用于块的MC处理。

[0330] 2.2.8 解码器侧运动矢量细化

[0331] 在双向预测操作中，对于一个块区域的预测，将两个分别由列表0的运动矢量(MV)和列表1的MV形成的预测块组合形成单个预测信号。在解码器侧运动矢量细化(DMVR)方法中，通过双边模板匹配处理进一步细化双向预测的两个运动矢量。解码器中应用的双边模板匹配用于在双边模板和参考图片中的重构样点之间执行基于失真的搜索，以便在不传输

附加运动信息的情况下获得细化的MV。

[0332] 在DMVR中,双边模板被生成为两个预测块的加权组合(即平均),其中两个预测块分别来自初始的列表0的MV0和列表1的MV1,如图28所示。模板匹配操作包括计算生成的模板与参考图片中的样点区域(在初始预测块周围)之间的代价度量。对于两个参考图片中的每一个,产生最小模板代价的MV被视为该列表的更新MV,以替换原始MV。在JEM中,为每个列表搜索九个MV候选。九个MV候选包括原始MV和8个周围MV,这八个周围MV在水平或垂直方向上或两者与原始MV具有一个亮度样点的偏移。最后,使用图28所示的两个新的MV(即MV0'和MV1')生成最终的双向预测结果。绝对差之和(sum of absolute differences,SAD)被用作代价度量。应当注意的是,在计算由一个周围MV生成的预测块的代价时,实际上使用取整的MV(至整数像素)而不是实际的MV来获取预测块。

[0333] 将DMVR应用于双向预测的Merge模式,其中一个MV来自过去的参考图片,并且另一个MV来自未来的参考图片,而无需传输附加语法元素。在JEM中,当为CU启用LIC、仿射运动、FRUC或子CU Merge候选时,不应用DMVR。

[0334] 2.2.9子块Merge候选列表

[0335] 建议将所有与子块相关的运动候选放在除了用于非子块Merge候选的常规Merge列表以外的单独的Merge列表中。

[0336] 将与子块相关的运动候选放在单独的Merge列表中,称为“子块Merge候选列表”。

[0337] 在一个示例中,子块Merge候选列表包括仿射Merge候选、ATMVP候选和/或基于子块的STMVP候选。

[0338] 2.2.9.1子块Merge候选列表的构造

[0339] 在该文献中,将普通Merge列表中的ATMVP Merge候选移动到仿射Merge列表的第一个位置。使得新列表(即基于子块的Merge候选列表)中的所有Merge候选都基于子块编解码工具。

[0340] 2.2.10广义双向预测示例

[0341] 利用广义双向预测(GBi),使用等式(1)生成双向预测 $P_{bi-pred}$ (为编解码的编解码单元信令通知一个加权参数),其中,P0和P1分别是使用列表0和列表1中的参考图片的运动补偿预测。 w 是以1/8精度表示的列表1预测的加权参数。GBi也称为具有基于编解码单元加权(BCW)的双向预测。

[0342]
$$P_{bi-pred} = ((8-w)*P_0 + w*(P_1+4)) \gg 3 \quad (13)$$

[0343] 在目前的GBi设计中,低延迟图片有5个权重{-2/8、3/8、4/8、5/8、10/8}可用,并且非低延迟图片有3个权重{3/8、4/8、5/8}。

[0344] 以下规则适用于确定编解码单元的权重值:

[0345] 对于需要信令通知MVD的每个双向预测PU(即,正常的帧间预测模式),其权值被设置为等于显式地信令通知。

[0346] 对于使用Merge模式编解码的每个双向预测PU,直接从用于相关联的Merge候选的权重值推断其权重值 w 。

[0347] 2.3使用相邻语法元素的上下文建模

[0348] 一种类型的上下文建模涉及在当前语法元素的过去最多具有两个相邻语法元素要编解码的上下文模板,其中,这种邻域类型的特定定义取决于语法元素。通常,特定bin的

这种类型的上下文模型的规范基于当前语法元素的左侧或顶部相邻元素的相关bin值的建模函数,如图25中的A2和B3所示。

[0349] 9.5.4.2.2使用左侧和上方语法元素的ctxInc的推导处理

[0350] 该处理的输入是规定当前亮度块的左上亮度样点相对于当前图片的左上样点的亮度位置(x0,y0)、颜色分量cIdx、当前编解码二叉树深度cqDepth、以及亮度样点中当前编解码块的宽度和高度cbWidth和cbHeight。

[0351] 该处理的输出是ctxInc。

[0352] 位置(xNbL,yNbL)被设置为等于(x0-1,y0),通过调用子条款6.4中规定的z扫描顺序的块的可用性推导处理来导出规定直接位于当前块左侧的块的可用性的变量availableL,将位置(xCurr,yCurr)设置为(x0,y0)、并且将相邻位置(xNbY,yNbY)设置为等于(xNbL,yNbL)作为输入,并将输出分配给availableL。

[0353] 位置(xNbA,yNbA)被设置为等于(x0,y0-1),通过调用子条款6.4中规定的z扫描顺序的块的可用性推导处理来导出规定直接位于当前块上方的块的可用性的变量availableA,将位置(xCurr,yCurr)设置为(x0,y0)、并且将相邻位置(xNbY,yNbY)设置为等于(xNbA,yNbA)作为输入,并将输出分配给availableA。

[0354] 变量sizeC、sizeTh2和sizeTh1导出如下:

[0355] $sizeTh2 = (MaxBtSizeY == 128) ? 1024 : ((MaxBtSizeY == 64) ? 512 : 256)$ (9-19)

[0356] $sizeTh1 = (MaxBtSizeY == 128) ? 128 : 64$ (9-20)

[0357] $sizeC = cbWidth * cbHeight$ (9-21)

[0358] ctxInc的分配规定如下:在表9-11中规定了左侧情况(condL)和上方情况(condA)的语法元素:alf_ctb_flag[x0][y0][cIdx],qt_split_cu_flag[x0][y0],mtt_split_cu_flag[x0][y0],cu_skip_flag[x0][y0],amvr_mode[x0][y0],inter_affine_flag[x0][y0],merge_triangle_flag[x0][y0]和merge_subblock_flag[x0][y0],

[0359] $ctxInc = (condL \& \& availableL) + (condA \& \& availableA) + ctxSetIdx * 3$ (9-22)

[0360] 表9-11使用左侧和上方语法元素的ctxInc的规范

[0361]

语法元素	condL	condA	ctxSetIdx
alf_ctb_flag[x0][y0][cIdx]	alf_ctb_flag[xNbL][yNbL][cIdx]	alf_ctb_flag[xNbA][yNbA][cIdx]	cIdx
qt_split_cu_flag	cqDepth[xNbL][yNbL] > cqDepth	cqDepth[xNbA][yNbA] > cqDepth	(cqDepth < 2) ? 0 : 1
mtt_split_cu_flag treeType == DUAL_TREE_CHROMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	3
mtt_split_cu_flag treeType != DUAL_TREE_CHROMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	(sizeC > sizeTh2) ? 0 : ((sizeC > sizeTh1) ? 1 : 2)
cu_skip_flag[x0][y0]	cu_skip_flag[xNbL][yNbL]	cu_skip_flag[xNbA][yNbA]	0

[0362]	AMVR_mode[x0][y0]	AMVR_mode[xNbL][yNbL]	AMVR_mode[xNbA][yNbA]	0
	merge_subblock_flag[x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0
	merge_triangle_flag[x0][y0]	merge_triangle_flag [xNbL][yNbL]	merge_triangle_flag [xNbA][yNbA]	0
	inter_affine_flag [x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0

[0363] 2.4预测模式标志的改进的上下文

[0364] 在VVC草案3中,使用仅有的一个上下文对预测模式标志(即pred_mode_flag)进行编解码。在一些实施例中,使用相邻块的pred_mode_flag的值导出用于编解码当前块的pred_mode_flag的上下文。如图43所述,如果用帧内预测模式对方(A)或左侧(L)块进行编解码,则上下文值增加1,因此用于编解码red_mode_flag的上下文的数量增加2。

[0365] 在VTM-3.0中,还启用了Merge和帧间预测的组合模式(即帧间帧内模式)。因此,当使用相邻块进行上下文推导时,帧间帧内模式应被视为帧内模式还是帧间模式成为一个问题。在该文献中,测试不同的解决方案以得出结论。

[0366] 在一些实施例中,为了导出用于编解码pred_mode_flag的上下文,测试了四种不同的方法,如下所述。

[0367] 表1:pred_mode_flag的上下文推导的不同方法的描述

上下文 (ctx) 推导的描述	
方法 #1 (3 个上下文)	ctx = A 是帧内 ? 1 : 0 ctx += L 是帧内 ? 1 : 0
[0368] 方法 #2 (2 个上下文)	ctx = (A 是帧内) (L 是帧内) ? 1 : 0
方法 #3 (3 个上下文)	ctx = A 是帧内或帧间帧内 ? 1 : 0 ctx += L 是帧内或帧间帧内 ? 1 : 0
[0369] 方法 #4 (2 个上下文)	ctx = (A 是帧内或帧间帧内) (L 是帧内或帧间帧内) ? 1 : 0

[0370] 3. 实施例解决的问题的示例

[0371] 在三角分割的设计中,一个块可以划分成两个分割。为了节省由于运动补偿而产生的内存带宽,需要对两个分割进行单向预测。在运动补偿处理期间,对角线边缘采用双向预测,并且其余部分均采用单向预测。如果每个分割都允许双向预测,则位于对角线边缘的样点将有四个MV,其中两个MV来自一个分割,并且两个MV来自另一个分割。这种设计存在以下问题:

[0372] 在构造TPM列表时,不使用子块Merge候选来导出TPM候选,而仅使用从空域和时域

块导出的非子块Merge候选。

[0373] 三角分割不允许使用基于子块的方法。

[0374] 仅允许对非对角线边缘进行单向预测的限制可导致编解码效率降低。

[0375] DMVR、BIO与三角分割相互作用。

[0376] 允许两个分割选择不同的TPM候选索引,然而,不能确定两个TPM候选是否来自相同个常规运动候选。例如,可以使用一个双向常规运动候选来导出两个TPM候选,并且每个候选可以被分配一个分割。在这种情况下,两个分割可以使用与非TPM模式相同的双向预测常规运动候选,这可能效率较低。

[0377] 当块宽度*高度小于64时,禁用TPM。然而,对于存储等于4x16或16x4并且4:2:0色度格式(color format)的块,将导致2x8色度块启用TPM,这增加了硬件复杂度。

[0378] VTM3.0中的几个新工具(包括MMVD、子块Merge列表、MHIntra(帧间帧内))也被称为CIIP(组合帧间帧内预测)模式、TPM被视为特殊的Merge模式。因此,存在5个Merge模式。在5个Merge模式的指示之前信令通知Merge标志/跳过标志。因此,为达到原始跳过/Merge模式,将MMVD/子块Merge列表/MHIntra(帧间帧内)/TPM的指示解码为0。

[0379] 在CABAC的上下文建模中,可能需要来自不同CTU行的相邻元素,其需要额外的行缓冲器来存储来自上方CTU行的元素。

[0380] 对于几种语法元素,在上下文建模中使用的相邻块不同于用于重构块的相邻块。

[0381] 4. 实施例的示例

[0382] 所提出的方法可应用于任何非正方形/非矩形分割,例如,几何分割。在下面的描述中,我们使用“三角分割模式”来表示非正方形/非矩形分割模式(TPM)的示例,并且插入到TPM候选列表中的运动矢量预测候选被命名为“TPM候选”。并且与任何先前编解码的块相关联的运动信息被命名为“常规运动候选”。应当注意的是,其他类型的分割也可适用。

[0383] 下面的详细技术应被视为解释一般概念的示例。不应狭隘地解释这些技术。此外,这些技术可以以任何方式组合。

[0384] TPM的扩展

[0385] 1. 提出可将将从子块候选中导出的一个或多个TPM候选添加到TPM列表中。

[0386] a. 在一个示例中,可以从ATMVP候选导出TPM候选。

[0387] b. 在一个示例中,可以从仿射候选导出TPM候选。

[0388] c. 对于子块候选(例如,ATMVP),它可以包括多个运动信息集,并且每个集合对应于一个子块(例如,对于ATMVP是8x8,对于仿射是4x4)。一个代表集可被视为常规运动候选集,并可用于导出TPM候选集。

[0389] i. 可基于与运动信息集相关联的子块位置来定义代表集。例如,可以选择与一个块的左上、右上、左下、右下子块相关联的运动信息。

[0390] ii. 可以选择M个代表集,例如,M被设置为分割的数目。在一个示例中,可以选择与每个分割的中心子块相关联的运动信息。

[0391] 2. 提出可以从一个常规运动候选继承多个分割(例如,三角分割设计中的两个分割)。

[0392] a. 在一个示例中,可以扩展每个分割的划分模式和选定Merge索引的指示范围。假设有M个划分分割、N个允许的Merge候选,则在当前三角设计中,指示的范围被定义为M*N*

N,而不是等于 $M*N*(N-1)$ 。

[0393] b. 在一个示例中,可以首先信令通知一个标志以指示是否从一个常规运动候选预测多个分割。

[0394] c. 在一个示例中,限制了当从一个常规运动候选预测多个分割时,这样的常规运动候选应为双向预测运动候选。

[0395] i. 可选地,此外,两个分割可以分别被分配常规运动候选的列表0运动信息和列表1运动信息。

[0396] ii. 可选地,此外,对于位于加权区域的样点,加权因子组可以不同于用于从不同运动候选预测两个分割的情况的加权因子组。在一个示例中,总是使用平均。

[0397] d. 除了传统的AMVP或Merge或TPM之外,可以将这种方法视为附加的编解码模式。

[0398] 3. 在用于导出一个TPM编解码块的最终预测块之前,可以进一步细化从比特流导出的每个分割的运动矢量。

[0399] a. 在一个示例中,可以将DMVR技术应用于两个分割的运动信息,作为整个块的运动信息。

[0400] b. 在一个示例中,在TPM预测之后应用BIO。

[0401] c. 在一个示例中,如果一个分割是从L X单向预测的,而另一个分割是双向预测或从L (1-X) 预测的,则可以通过使用列表L (1-X) 中的另一个分割的运动信息来应用DMVR或BIO技术。

[0402] d. 在一个示例中,可以存储细化的运动信息。

[0403] e. 在一个示例中,如果分割是从LX单向预测的,则可以存储LX中的细化运动信息。

[0404] a. 提出了可以将TPM中的一个或多个分割与帧内预测相结合。例如,对TPM预测和帧内预测进行加权求和,得到最终的预测结果。

[0405] f. 提出了TPM中使用的一个或多个参考图片可以是当前图片。

[0406] 4. 提出了可对TPM编解码块启用基于子块的技术,其中一个分割内的每个子块可被分配其自身的运动信息。

[0407] a. 在一个示例中,基于子块的Merge候选可以被添加到TPM编解码块的列表中。

[0408] b. 在一个示例中,即使当一个块用TPM编解码时,也可以启用仿射模式。

[0409] c. 可选地,此外,限制对除了位于加权区域的子块之外的所有子块使用单向预测。

[0410] d. 可选地,此外,限制一个分割内不同子块的所有参考图片都相同。

[0411] e. 可以从一个ATMVP候选导出一个或多个TPM候选,例如,一个分割内的每个子块可以继承ATMVP候选的列表0运动信息的运动信息。

[0412] f. 子块TPM编解码块的Merge列表可以检查子块Merge候选列表构造处理中使用的相同空域和/或时域块。

[0413] 5. 当启用广义双向预测时,双向预测常规运动候选进一步与两个加权因子相关联。然而,当使用这种双向预测常规运动候选来导出TPM候选时,提出不继承加权因子。

[0414] a. 在一个示例中,一个TPM候选不与任何加权因子相关联。因此,在推导位于非加权区域的样点的预测值时,直接使用运动补偿结果。对于位于加权区域的样点,使用选定的加权因子组。

[0415] b. 可选地,继承加权因子。可选地,此外,在生成位于加权区域的样点的最终预测

值时,可以将继承的加权因子中的一个应用于从区域的对应TPM候选导出的预测值。

[0416] 6.提出了对不位于一个TPM编解码块的加权区域的样点,可以启用双向预测。

[0417] a.在一个示例中,要求两个参考图片相同。

[0418] b.在一个示例中,对于每个分割,可以分配一个参考图片(例如,在当前设计之后,对于两个分割,可以是相同的或不同的)。同时,可以进一步分配第三参考图片和运动矢量。

[0419] 7.提出可以将多假设应用于TPM编解码块。

[0420] a.在一个示例中,可以首先生成使用当前设计的一个第一最终预测块。基于另一参考图片和运动矢量,还使用单向预测生成第二预测块。并且从第一最终预测块和第二预测块导出一个块的最终预测块。

[0421] b.可选地,可以首先生成使用当前设计的一个第一最终预测块。基于两个附加的参考图片和运动矢量,还使用双向预测生成第二预测块。并且从第一最终预测块和第二预测块导出一个块的最终预测块。

[0422] c.可选地,对于每个分割,首先应用双向预测或多假设预测。然后,对于加权区域,将由双向/多假设预测产生的预测块的加权结果用作最终预测值。

[0423] 8.当为一个块启用TPM时,可以禁用OBMC。

[0424] a.可选地,对于位于加权区域的样点,禁用OBMC,但是对于位于非加权区域的样点,可以启用OBMC。

[0425] 9.一个块可以划分为两个以上的非正方形/非三角分割。

[0426] a.一个块可以划分成四个非正方形/非三角分割。在一个示例中,示例如图29所示。

[0427] b.当有两个以上的非正方形/非三角分割时,限制只有两个分割可以被帧间编解码。

[0428] c.对于位于两个相邻分割的边缘周围的样点,可以应用混合(例如,两个预测块的加权平均)。

[0429] 图29示出了一个块内的多个几何分割的示例。

[0430] TPM所需边信息(side information)的信令

[0431] 10.分割模式、两个Merge索引和编解码值(例如,g_三角形组合)之间的映射可以从序列到序列、从图片到图片、从片到片、从条带到条带、从CTU到CTU、从块到块自适应地改变。

[0432] a.在一个示例中,映射可取决于于构造的Merge候选列表。

[0433] b.在一个示例中,映射可取决于与Merge候选相关联的编解码模式。

[0434] 11.可以在Merge标志之前对TPM使用的指示标志进行编解码。也就是说,也可以为非Merge模式启用TPM。

[0435] a.可选地,此外,如果块是TPM编解码的,则可以不信令通知Merge标志。

[0436] b.可选地,此外,如果块是帧间编解码的(例如,指示它是AMVP或Merge模式),则可以进一步信令通知Merge标志。

[0437] c.TPM只能用于非跳过模式。

[0438] 12.如果一个块使用AMVP模式编解码,则可以对TPM使用的指示标志进行编解码。

[0439] a.可选地,此外,如果一个块是用Merge模式编解码的,则可以不信令通知TPM标

志。

[0440] b. 可选地, 此外, 可以从空域或时域相邻或非邻近块继承TPM使用标志。

[0441] 13. 可以从空域相邻块继承划分模式。

[0442] a. 可选地, 可以从时域块继承划分模式。

[0443] b. 在一个示例中, 划分模式与运动信息一起存储。对于从一个常规运动候选导出的每个TPM候选, 还继承与常规运动候选相关联的划分模式。

[0444] i. 在这种情况下, TPM编解码块可以从相邻TPM编解码块继承运动信息(或编解码模式)和分割模式二者。

[0445] 14. 提出了可以单独信令通知用于几何分割的划分模式和Merge索引。

[0446] a. 在一个示例中, 可以单独编解码一个比特标志, 以指示TPM划分模式。

[0447] i. 可使用相邻块的TPM划分模式导出上下文来编解码该比特。

[0448] b. 在一个示例中, 可以单独编解码每个分割的多个Merge索引。

[0449] i. 可选地, 此外, 还可以利用不同分割的Merge索引之间的预测。

[0450] c. 在一个示例中, Merge索引的编解码可以取决于划分模式的编解码的一比特标志。

[0451] d. 可选地, 可以首先对Merge索引进行编解码(联合或单独地), 并且可以根据Merge索引来对划分模式的一比特标志进行编解码。

[0452] 15. 提出了对TPM使用的指示标志使用旁路编解码。

[0453] a. 可选地, 仍然使用上下文对标志进行编解码, 然而, 上下文取决于块的宽度/高度。

[0454] b. 可选地, 仍然使用上下文对标志进行编解码, 然而, 上下文取决于编解码深度(例如, 四叉树深度、二叉树深度和/或三叉树深度)。

[0455] c. 可选地, 仍然使用上下文对标志进行编解码, 然而, 上下文取决于块的宽度/高度, 诸如块形状(例如, 正方形或非正方形)和/或块宽度和高度之比(例如, 等于1, 等于1/4或4, 等于1/2或2)。

[0456] d. 可选地, 仍然使用上下文对标志进行编解码, 然而, 上下文取决于相邻块的编解码信息, 这些相邻块并不总是在一个块内的左上位置旁边。分别用W和H表示块的宽度和高度, 并且(0,0)是当前块内左上样点的坐标。

[0457] i. 在一个示例中, 上方相邻块可定义为 $(W/2, -1)$, $(W/2-1, -1)$, $(W-1, -1)$ 。

[0458] ii. 在一个示例中, 左侧相邻块可定义为 $(-1, H/2)$ 、 $(-1, H/2-1)$ 、 $(-1, H-1)$ 。

[0459] iii. 图30中描绘了在上下文选择中可以使用的上方和左侧相邻块的一些组合。

[0460] iv. 在一个示例中, 只能选择上方相邻块。可选地, 此外, 当块的宽度和高度之比大于阈值(例如, 1或2)时, 这种情况被启用。

[0461] v. 在一个示例中, 只能选择左侧相邻块。可选地, 此外, 当块的宽度和高度之比大于阈值(例如, 1或2)时, 这种情况被启用。

[0462] 图30示出了用于TPM标志的上下文选择的相邻块的示例。左上角的示例是 $(W/2, -1)$ 和 $(-1, H/2)$ 。右上角的示例是 $(W/2-1, -1)$ 和 $(-1, H/2-1)$ 。左下角的示例是 $(0, -1)$ 和 $(-1, H/2-1)$ 。右下角的示例是 $(W-1, -1)$ 和 $(-1, H/2-1)$ 。

[0463] 16. 当TPM列表中的两个候选索引(例如, a_0 和 a_1 、 b_0 和 b_1)被联合编解码(或与其他

信息进一步连接)时,码字可以从块到块、从CTU到CTU、从片到片、从图片到图片、从序列到序列自适应地改变。

[0464] a. 在一个示例中,可以基于与这两个索引相关联的两个TPM候选是否来自相同常规运动候选来分配码字。

[0465] i. 在一个示例中,如果发生以下情况,码字 $C(a_0, a_1)$ 可以小于 $C(b_0, b_1)$:

[0466] 1. 索引等于 a_0 的TPM候选和索引等于 a_1 的TPM候选是从不同的常规运动候选导出的,并且索引等于 b_0 的TPM候选和索引等于 b_1 的TPM候选是从相同的常规运动候选导出的。

[0467] b. 在一个示例中,可以基于与两个索引相关联的两个TPM候选是否来自相同预测方向(或参考图片列表)来分配码字。

[0468] i. 在一个示例中,如果发生以下情况,码字 $C(a_0, a_1)$ 可以小于 $C(b_0, b_1)$:

[0469] 1. 索引等于 a_0 的TPM候选和索引等于 a_1 的TPM候选来自两个预测方向,并且索引等于 b_0 的TPM候选和索引等于 b_1 的TPM候选来自相同预测方向。

[0470] 17. TPM中的两个分割划分可以用不同的模式进行编解码。

[0471] a. 在一个示例中,一个是帧内编解码的,并且另一个是帧间编解码的。

[0472] b. 在另一个示例中,一个是Merge编解码,另一个是AMVP编解码。

[0473] 启用TPM

[0474] 18. 对于没有信令通知TPM使用标志的颜色分量,可以自动禁用TPM。

[0475] a. 在一个示例中,对于色度块,如果宽度或高度小于或等于阈值(例如2)。

[0476] 19. 对于尺寸等于 $N \times M$ 或 $L \times N$ (例如, N 等于4)的块,可以禁用TPM。

[0477] a. 在一个示例中, M 和 N 被设置为8,16。

[0478] b. 可选地,此外,当色度格式等于4:2:0时,禁用TPM。

[0479] c. 在一个示例中,当块宽度或高度或宽度和高度二者大于(或等于)阈值 L 时,可以禁用TPM。

[0480] vi. 在一个示例中, L 是64。

[0481] d. 在一个示例中,当块的宽度或高度或宽度和高度二者小于(或等于)阈值 L 时,可以禁用上述方法。

[0482] vii. 在一个示例中, L 是4或8。

[0483] e. 在一个示例中,当宽度/高度或高度/宽度大于阈值 L 时,它被禁用,例如, L 是16。

[0484] f. 可以预先定义阈值 L ,或者可以在SPS/PPS/图片/条带/片组/片级中信令通知。

[0485] g. 可选地,阈值可取决于特定编解码信息,例如块尺寸、图片类型、时域层索引等。

[0486] 20. 可以为一个块的不同颜色分量启用和禁用TPM。

[0487] a. 在一个示例中,对于等于 $N \times M$ 或 $L \times N$ (例如, N 等于4)的亮度块尺寸,可以为该亮度块启用TPM,然而,可以为色度块(例如,为4:2:0色度格式)禁用TPM。

[0488] b. 在一个示例中,如果对一个颜色分量禁用TPM,则可以基于一个分割(例如,第一分割)的运动信息或预测模式导出相关联的预测块。

[0489] c. 在一个示例中,如果对一个颜色分量禁用TPM,则可以基于两个分割的运动信息直接导出相关联的预测块。

[0490] viii. 在一个示例中,两个分割的运动信息可以形成为双向预测运动矢量,例如,第一分割的 Mv_1 和第二分割的 Mv_2 具有来自不同方向的运动矢量。

- [0491] ix. 在一个示例中,两个分割的运动信息可以形成为单向预测运动矢量,例如,第一分割的Mv1和第二分割的Mv2具有来自相同方向的运动矢量,分割A的一个运动矢量可以缩放到分割B的另一参考图片,并且可以使用分割B的运动矢量的平均和分割A的缩放MV。
- [0492] x. 在一个示例中,两个分割的运动信息可形成为双向预测运动矢量,例如,第一分割的Mv1和第二分割的Mv2具有来自相同方向的运动矢量,分割A的一个运动矢量可缩放到另一方向的另一参考图片,并且可以使用分割B的运动矢量和分割A的缩放MV。
- [0493] xi. 在以上示例中,可以基于色度格式进一步缩放运动信息。
- [0494] 21. 对于上述方法,可以在SPS/PPS/VPS/图片标头/条带标头/片组标头/CTU行/CTU组/区域中信令通知是否和/或如何启用它们。
- [0495] a. 在一个示例中,当块的宽度或高度、或宽度和高度二者、或宽度*高度大于(或等于)阈值L时,可以禁用上述方法。
- [0496] i. 在一个示例中,L是64、4096。
- [0497] b. 在一个示例中,当块的宽度或高度、或宽度和高度二者、或宽度*高度小于(或等于)阈值L时,可以禁用上述方法。
- [0498] i. 在一个示例中,L是4或8、64
- [0499] c. 可以预先定义阈值L,或者可以在SPS/PPS/图片/条带/片组/片级中信令通知。
- [0500] d. 可选地,阈值可取决于特定编解码信息,例如块尺寸、图片类型、时域层索引等。
- [0501] e. 在一个示例中,是否应用TPM取决于块的宽度/高度,诸如块形状(例如,是正方形或非正方形)和/或块宽度和高度之比(例如,1,1/4或4,1/2或2)。
- [0502] f. 提出将TPM视为与跳过/Merge模式不同的模式。因此,当为一个块启用TPM时,将相关联的Merge标志信令通知为假(false)。
- [0503] 22. 提出将TPM视为与跳过/Merge模式不同的模式。因此,当为一个块启用TPM时,将相关联的Merge标志信令通知为假。
- [0504] 23. 提出将MHIntra视为与跳过/Merge模式不同的模式。因此,当为一个块启用MHIntra时,将相关联的Merge标志信令通知为假。
- [0505] 24. 提出将子块Merge列表视为与跳过/Merge模式不同的模式。因此,当为一个块启用子块Merge列表时,将相关联的Merge标志信令通知为假。
- [0506] 25. 对于22至24项的方法,当一个块被解码为Merge/跳过模式时,跳过TPM/MHIntra/子块Merge列表的指示的信令通知,并将其推断为假。
- [0507] 26. 提出TPM/MHIntra/子块Merge列表/MMVD的使用指示的编解码顺序如下:
- [0508] a. MMVD、子块Merge列表、TPM、MHIntra。
- [0509] b. MMVD、子块Merge列表、MHIntra、TPM。
- [0510] c. 其他种类的顺序。
- [0511] 27. 提出TPM/MHIntra/子块Merge列表/MMVD的使用指示的编解码顺序可以从一个视频单元到另一个视频单元不同,视频单元诸如块/CU/PU/CTU/LCU行/区域/条带/图片/序列。
- [0512] a. 在一个示例中,顺序可取决于低延迟检查标志。
- [0513] b. 在一个示例中,顺序可取决于先前编解码的信息。
- [0514] c. 在一个示例中,顺序可取决于一个块的维度。

[0515] 28. 为了节省行缓冲区尺寸, 当使用相邻块的AMVR信息(即MV精度)来编解码当前块的AMVR信息时, 如果相邻块位于不同的LCU行中, 则将相邻块标记为不可用。

[0516] a. 在一个示例中, 相关联的AMVR信息被设置为默认值, 即: 1/4像素。

[0517] b. 可选地, 相邻块被视为位于不同条带/片中的块。

[0518] c. 在一个示例中, 如果相邻块位于不同区域(例如, 通常被设置为64x64的视频处理数据单元), 则以与上述相同的方式对待该块。

[0519] 29. 为了节省行缓冲区尺寸, 当使用相邻仿射编解码块的AMVR信息(即MV精度)来编解码当前仿射编解码块的AMVR信息时, 如果相邻块位于不同的LCU行中, 则将相邻仿射编解码块标记为不可用。

[0520] a. 在一个示例中, 相关联的AMVR信息被设置为默认值, 即: 1/4像素。

[0521] b. 可选地, 相邻仿射编解码块被视为位于不同条带/片中的那些块。

[0522] c. 在一个示例中, 如果相邻仿射编解码块位于不同区域(例如, 通常被设置为64x64的视频处理数据单元), 则以如上所述的相同方式对待该块。

[0523] 30. 还提出如果相邻块位于不同的LCU行中, 则禁用使用相邻块的信息进行上下文建模。可选地, 只有一个上下文被用于编解码下面提到的语法元素。可选地, 旁路编解码被应用于编解码下面提到的语法元素。

[0524] a. 信息包括alf_ctb_flag。

[0525] b. 信息包括qt_split_cu_flag。

[0526] c. 信息包括mtt_split_cu_flag。

[0527] d. 信息包括mtt_split_cu_flag。

[0528] e. 信息包括cu_skip_flag。

[0529] f. 信息包括amvr_mode。

[0530] g. 信息包括merge_subblock_flag。

[0531] h. 信息包括merge_triangle_flag。

[0532] i. 信息包括inter_affine_flag。

[0533] j. 信息还可包括: cu_skip_flag、pred_mode_flag、pcm_flag、intra_luma_ref_idx、intra_luma_mpm_flag、intra_luma_mpm_idx、intra_luma_mpm_remainder、intra_chroma_pred_mode、merge_flag、inter_pred_idc、inter_affine_flag、cu_affine_type_flag、ref_idx_l0、mvp_l0_flag、ref_idx_l1、mvp_l1_flag、amvr_flag、amvr_4pel_flag、gbi_idx、cu_cbf、mmvd_flag、mmvd_merge_flag等。

[0534] CABAC上下文建模的简化

[0535] 31. 提出上下文建模中使用的邻域信息只能来自用于解码当前块的块。

[0536] a. 在一个示例中, 对于cu_skip_flag, 可以使用图42中的另外两个块(例如, A1和B1), 而不是使用图25中由A3和B3表示的两个相邻块的信息。

[0537] b. 可选地, 此外, 当相邻块在当前CTU/CTU行/区域/视频数据处理单元或其他视频单元(诸如条带/片/片组/图片边界)之外时, 不允许在解析处理中使用来自这些相邻块的相邻信息。

[0538] c. 可选地, 此外, 当相邻块在当前CTU/CTU行/区域/视频数据处理单元或其他视频单元(诸如条带/片/片组/图片边界)之外时, 来自该相邻块的相邻信息可被视为不可用并

在解析处理中使用。

[0539] d. 可选地, 此外, 当相邻块在当前CTU/CTU行/区域/视频数据处理单元或其他视频单元(诸如条带/片/片组/图片边界)之外时, 来自该相邻块的相邻信息可被设置为默认值并在解析处理中使用。

[0540] e. 可选地, 此外, 当相邻块在当前CTU/CTU行/区域/视频数据处理单元或其他视频单元(诸如条带/片/片组/图片边界)之外时, 来自当前CTU/CTU行/区域/视频数据处理单元或其他视频单元(诸如条带/片/片组/图片边界)中的其它一个或多个相邻块的相邻信息可在解析处理中作为替代而使用。

[0541] 32. 提出了对于依赖相邻块的编解码信息的所有语法元素, 上下文建模中使用的相邻信息(诸如相邻块的语法元素(简称相邻元素))可以来自一些预定义的相邻位置。在一个示例中, 在一个块的重构处理期间也可以访问这种相邻位置。

[0542] a. 在一个示例中, 来自图2中A1和B1的相邻元素被用于所有语法元素的上下文建模。

[0543] b. 在一个示例中, 来自图2中A0和B0的相邻元素被用于所有语法元素的上下文建模。

[0544] c. 在一个示例中, 来自图2中A1和B2的相邻元素被用于所有语法元素的上下文建模。

[0545] d. 在一个示例中, 来自图2中A0和B1的相邻元素被用于所有语法元素的上下文建模。

[0546] e. 在一个示例中, 来自图2中A0和B2的相邻元素被用于所有语法元素的上下文建模。

[0547] f. 在一个示例中, 来自图2中A0和A1的相邻元素被用于所有语法元素的上下文建模。

[0548] g. 在一个示例中, 来自图2中A1和B0的相邻元素被用于所有语法元素的上下文建模。

[0549] h. 在一个示例中, 来自图2中B0和B1的相邻元素被用于所有语法元素的上下文建模。

[0550] i. 在一个示例中, 来自图2中B0和B2的相邻元素被用于所有语法元素的上下文建模。

[0551] j. 在一个示例中, 来自图2中B1和B2中的相邻元素被用于所有语法元素的上下文建模。

[0552] k. 在一个示例中, 对于所有语法元素的上下文建模只使用一个相邻元素。

[0553] 33. 提出了在上下文建模中不可使用来自不同于当前CTU行或CTU的CTU行或CTU中的相邻块的相邻元素。

[0554] a. 可选地, 在上下文建模中不可使用来自不同于当前CTU行或CTU的CTU行或CTU中的相邻块的相邻元素。

[0555] b. 在一个示例中, 如果相邻元素来自不同于当前CTU行或CTU的CTU行或CTU的相邻块, 则认为它不可用。

[0556] c. 在一个示例中, 如果相邻元素来自不同于当前CTU行或CTU的CTU行或CTU中的相

邻块,则使用默认值。

[0557] d. 在一个示例中,如果相邻元素来自不同于当前CTU行或CTU的CTU行或CTU中的相邻块,则应将其替换为来自不同于当前CTU行或CTU的CTU行或CTU中的另一个相邻块的另一个相邻元素。

[0558] e. 在一个示例中,如果相邻元素来自不同于当前CTU行或CTU的CTU行或CTU中的相邻块,则可以按顺序进一步检查位于相同CTU行或CTU中的N(N>0)个可选相邻元素,直到找到一个可用的相邻元素、或者N个相邻元素全部都不可用。例如,N可以等于1、2、3或4。

[0559] 34. 提出的方法可以仅应用于一些语法元素,诸如alf_ctb_flag、qt_split_cu_flag、mtt_split_cu_flag、cu_skip_flag、amvr_mode、inter_affine_flag、merge_triangle_flag and merge_subblock_flag、cu_skip_flag、pred_mode_flag、pcm_flag、intra_luma_ref_idx、intra_luma_mpm_flag、intra_luma_mpm_idx、intra_luma_mpm_remainder、intra_chroma_pred_mode、merge_flag、inter_pred_idc、inter_affine_flag、cu_affine_type_flag、ref_idx_l0、mvp_l0_flag、ref_idx_l1 mvp_l1_flag、amvr_flag、amvr_4pel_flag、gbi_idx、cu_cbf、mmvd_flag、mmvd_merge_flag、pred_mode_flag等。

[0560] a. 在一个示例中,从以下方法导出pred_mode_flag的上下文建模:

上下文 (ctx) 推导的描述

方法 #1 ctx = A1 是帧内 ? 1 : 0

(3 个上下文) ctx += B1 是帧内 ? 1 : 0

[0561] 方法 #2 ctx = (A1 是帧内) || (B1 是帧内) ? 1 : 0
(2 个上下文)

方法 #3 ctx = A1 是帧内或帧间帧内 ? 1 : 0

(3 个上下文) ctx += B1 是帧内或帧间帧内 ? 1 : 0

[0562] 方法 #4 ctx = (A1 是帧内或帧间帧内) || (B1 是帧内或
(2 个上下文) 帧间帧内) ? 1 : 0

[0563] b. 可选地,可以从其他相邻块而不是A1和/或B1(例如,图25中的A2/B3或A0/B0)导出pred_mode_flag的上下文建模。

[0564] c. 可以用其他种类的语法元素替代上述语法pred_mode_flag。

[0565] d. 在一个示例中,如果一个相邻块不可用,则在推导上下文时将其视为帧间编解码;

[0566] e. 在一个示例中,如果一个相邻块不可用,则在推导上下文时将其视为帧内编解码;

[0567] f. 在一个示例中,如果一个相邻块不可用,则在导出上下文时将其视为帧间帧内编解码;

[0568] g. 在一个示例中,如果一个相邻块与当前块不在相同个CTU行中,则在推导上下文时将其视为帧间编解码;

[0569] h. 在一个示例中,如果一个相邻块与当前块不在相同个CTU行中,则在导出上下文时将其视为帧内编解码;

[0570] i. 在一个示例中, 如果一个相邻块与当前块不在相同CTU行中, 则在导出上下文时将其视为帧间帧内编解码(也称为帧内帧间组合预测模式)。

[0571] 5. 附加实施例示例

[0572] 5.1 实施例1

[0573] 下面给出将TPM信令通知为不同于Merge模式的模式的示例。新添加的部分用粗体斜体突出显示, 并且删除的部分用大写标记。

[0574] 7.3.4.6 编解码单元语法

	coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
	if(slice_type != I) {	
	<i>cu_skip_flag</i> [x0][y0]	ac(v)
	if(cu_skip_flag[x0][y0] == 0)	
	<i>pred_mode_flag</i>	ac(v)
	}	
[0575]	if(CuPredMode[x0][y0] == MODE_INTRA) {	
	...	
	} else { /* MODE_INTER */	
	if(cu_skip_flag[x0][y0] == 0) {	
	<i>merge_flag</i> [x0][y0]	ac(v)
	if(merge_flag[x0][y0]) {	

[0576]

merge_data(x0, y0, cbWidth, cbHeight)	
} else {	
<i>if(sps_triangle_enabled_flag && slice_type == B && cbWidth * cbHeight >= 16)</i>	
<i>merge_triangle_flag[x0][y0]</i>	<i>ae(v)</i>
<i>if(merge_triangle_flag[x0][y0])</i>	
<i>merge_triangle_idx[x0][y0]</i>	<i>ae(v)</i>
<i>else {</i>	
<i>if(slice_type == B)</i>	
<i>inter_pred_idc[x0][y0]</i>	<i>ae(v)</i>
<i>if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {</i>	
<i>inter_affine_flag[x0][y0]</i>	<i>ae(v)</i>
<i>if(sps_affine_type_flag && inter_affine_flag[x0][y0])</i>	
<i>cu_affine_type_flag[x0][y0]</i>	<i>ae(v)</i>
<i>}</i>	
<i>if(inter_pred_idc[x0][y0] != PRED_L1) {</i>	
<i>if(num_ref_idx_l0_active_minus1 > 0)</i>	
<i>ref_idx_l0[x0][y0]</i>	<i>ae(v)</i>
<i>mvd_coding(x0, y0, 0, 0)</i>	
<i>if(MotionModelIdc[x0][y0] > 0)</i>	
<i>mvd_coding(x0, y0, 0, 1)</i>	
<i>if(MotionModelIdc[x0][y0] > 1)</i>	
<i>mvd_coding(x0, y0, 0, 2)</i>	
<i>mvp_l0_flag[x0][y0]</i>	<i>ae(v)</i>
<i>} else {</i>	
<i>MvdL0[x0][y0][0] = 0</i>	
<i>MvdL0[x0][y0][1] = 0</i>	
<i>}</i>	
<i>if(inter_pred_idc[x0][y0] != PRED_L0) {</i>	
<i>if(num_ref_idx_l1_active_minus1 > 0)</i>	
<i>ref_idx_l1[x0][y0]</i>	<i>ae(v)</i>
<i>if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {</i>	
<i>MvdL1[x0][y0][0] = 0</i>	
<i>MvdL1[x0][y0][1] = 0</i>	
<i>MvdCpL1[x0][y0][0][0] = 0</i>	
<i>MvdCpL1[x0][y0][0][1] = 0</i>	
<i>MvdCpL1[x0][y0][1][0] = 0</i>	
<i>MvdCpL1[x0][y0][1][1] = 0</i>	
<i>MvdCpL1[x0][y0][2][0] = 0</i>	
<i>MvdCpL1[x0][y0][2][1] = 0</i>	
<i>} else {</i>	
<i>mvd_coding(x0, y0, 1, 0)</i>	
<i>if(MotionModelIdc[x0][y0] > 0)</i>	
<i>mvd_coding(x0, y0, 1, 1)</i>	
<i>if(MotionModelIdc[x0][y0] > 1)</i>	

	mvd_coding(x0, y0, 1, 2)	
	mvp_l1_flag [x0][y0]	ae(v)
	} else {	
	MvdL1[x0][y0][0] = 0	
	MvdL1[x0][y0][1] = 0	
	}	
	if(sps_amvr_enabled_flag && inter_affine_flag == 0 && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0 MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0))	
	amvr_mode [x0][y0]	ae(v)
[0577]	if(sps_gbi_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && cbWidth * cbHeight >= 256)	
	gbi_idx [x0][y0]	ae(v)
	}	
	}	
	}	
	if(!pcm_flag[x0][y0]) {	
	if(CuPredMode[x0][y0] != MODE_INTRA && cu_skip_flag[x0][y0] == 0)	
	cu_cbf	ae(v)
	if(cu_cbf)	
	transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
	}	
	}	

[0578] 7.3.4.8 Merge数据语法

	merge_data(x0, y0, cbWidth, cbHeight) {	描述符
	mmvd_flag [x0][y0]	ae(v)
	if(mmvd_flag[x0][y0] == 1) {	
	mmvd_merge_flag [x0][y0]	ae(v)
	mmvd_distance_idx [x0][y0]	ae(v)
	mmvd_direction_idx [x0][y0]	ae(v)
	} else {	
	if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
	merge_subblock_flag [x0][y0]	ae(v)
[0579]	if(merge_subblock_flag[x0][y0] == 1) {	
	if(MaxNumSubblockMergeCand > 1)	
	merge_subblock_idx [x0][y0]	ae(v)
	} else {	
	if(sps_mh_intra_enabled_flag && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	
	mh_intra_flag [x0][y0]	ae(v)
	if(mh_intra_flag[x0][y0]) {	
	if(cbWidth <= 2 * cbHeight cbHeight <= 2 * cbWidth)	
	mh_intra_luma_mpm_flag [x0][y0]	ae(v)
	if(mh_intra_luma_mpm_flag[x0][y0])	
	mh_intra_luma_mpm_idx [x0][y0]	ae(v)

	}	
	}	
	IF(SPS_TRIANGLE_ENABLED_FLAG && SLICE_TYPE == B && CBWIDTH * CBHEIGHT >= 16)	
[0580]	MERGE_TRIANGLE_FLAG[X0][Y0]	AE(V)
	IF(MERGE_TRIANGLE_FLAG[X0][Y0])	
	MERGE_TRIANGLE_IDX[X0][Y0]	AE(V)
	else if(MaxNumMergeCand > 1)	
	<i>merge_idx[x0][y0]</i>	ac(v)
	}	
	}	

[0581] 5.2实施例2

[0582] 在本实施例中,为了上下文建模减小了行缓冲区尺寸。更改以粗体斜体突出显示,如下所示。

[0583] 9.5.4.2.2使用左侧和上方语法元素的ctxInc的推导处理

[0584] 该处理的输入是规定当前亮度块的左上亮度样点相对于当前图片的左上样点的亮度位置(x0,y0)、颜色分量cIdx、当前编解码二叉树深度cqDepth、以及亮度样点中当前编解码块的宽度和高度cbWidth和cbHeight。

[0585] 该处理的输出是ctxInc。

[0586] 位置(xNbL,yNbL)被设置为等于(x0-1,y0),通过调用子条款6.4中规定的z扫描顺序的块的可用性推导处理来导出规定直接位于当前块左侧的块的可用性的变量availableL,将位置(xCurr,yCurr)设置为(x0,y0)、并且将相邻位置(xNbY,yNbY)设置为等于(xNbL,yNbL)作为输入,并将输出分配给availableL。

[0587] 位置(xNbA,yNbA)被设置为等于(x0,y0-1),通过调用子条款6.4中规定的z扫描顺序的块的可用性推导处理来导出规定直接位于当前块上方的块的可用性的变量availableA,将位置(xCurr,yCurr)设置为(x0,y0)、并且将相邻位置(xNbY,yNbY)设置为等于(xNbA,yNbA)作为输入,并将输出分配给availableA。

[0588] 如果(x0,y0-1)对比(x0,y0)在不同的LCU行中,则将availableA重置为假。

[0589] 变量sizeC、sizeTh2和sizeTh1导出如下:

[0590] $sizeTh2 = (MaxBtSizeY == 128) ? 1024 : ((MaxBtSizeY == 64) ? 512 : 256)$ (9-19)

[0591] $sizeTh1 = (MaxBtSizeY == 128) ? 128 : 64$ (9-20)

[0592] $sizeC = cbWidth * cbHeight$ (9-21)

[0593] ctxInc的分配规定如下:在表9-11中规定了左侧情况(condL)和上方情况(condA)的语法元素:alf_ctb_flag[x0][y0][cIdx],qt_split_cu_flag[x0][y0],mtt_split_cu_flag[x0][y0],cu_skip_flag[x0][y0],amvr_mode[x0][y0],inter_affine_flag[x0][y0],merge_triangle_flag[x0][y0]和merge_subblock_flag[x0][y0],

[0594] $ctxInc = (condL \&\& availableL) + (condA \&\& availableA) + ctxSetIdx * 3$ (9-22)

[0595] 表9-11使用左侧和上方语法元素的ctxInc的规范

语法元素	condL	condA	ctxSetIdx
alf_ctb_flag[x0][y0][cIdx]	alf_ctb_flag[xNbL][yNbL][cIdx]	alf_ctb_flag[xNbA][yNbA][cIdx]	cIdx
qt_split_cu_flag	cqtDepth[xNbL][yNbL] > cqtDepth	cqtDepth[xNbA][yNbA] > cqtDepth	(cqtDepth < 2) ? 0 : 1
mtt_split_cu_flag treeType == DUAL_TREE_CHR OMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	3
mtt_split_cu_flag treeType != DUAL_TREE_CHR OMA	cbHeight[xNbL][yNbL] < cbHeight	cbHeight[xNbA][yNbA] < cbWidth	(sizeC > sizeTh2) ? 0 : ((sizeC > sizeTh1) ? 1 : 2)
[0596] cu_skip_flag[x0][y0]	cu_skip_flag[xNbL][yNbL]	cu_skip_flag[xNbA][yNbA]	0
amvr_mode[x0][y0]	amvr_mode[xNbL][yNbL]	amvr_mode[xNbA][yNbA]	0
merge_subblock_flag [x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0
merge_triangle_flag [x0][y0]	merge_triangle_flag [xNbL][yNbL]	merge_triangle_flag [xNbA][yNbA]	0
inter_affine_flag [x0][y0]	merge_subblock_flag[xNbL][yNbL] inter_affine_flag[xNbL][yNbL]	merge_subblock_flag[xNbA][yNbA] inter_affine_flag[xNbA][yNbA]	0

[0597] 图31是视频处理装置3100的框图。装置3100可用于实现本文所述的一个或多个方法。装置3100可以被实现在智能手机、平板电脑、计算机、物联网(IoT)接收器等中。装置3100可以包括一个或多个处理器3102、一个或多个存储器3104和视频处理硬件3106。处理器3102可以被配置为实现本文所述的一个或多个方法。存储器(多个存储器)3104可用于存储用于实现本文所述的方法和技术的代码和数据。视频处理硬件3106可用于在硬件电路中实现本文所述的一些技术。

[0598] 图33是处理视频的方法3300的流程图。方法3300包括：由处理器执行确定(3305)将第一视频块分割以包括非矩形和非正方形的第一预测部分；将与第一预测部分相关联的第一运动矢量(MV)预测候选添加(3310)到与第一视频块相关联的运动候选列表，其中，从基于子块的运动候选导出第一MV预测候选；以及使用运动候选列表执行(3315)第一视频块的进一步处理。

[0599] 图34是处理视频的方法3400的流程图。方法3400包括：基于第一划分模式分割(3405)第一视频块以包括第一预测部分，第一预测部分与第一Merge索引和第一编解码值相关联；基于第二划分模式分割(3410)第二视频块以包括预测部分，第二视频块的预测部分与第二Merge索引和第二编解码值相关联，其中第一编解码值等于第二编解码值，并且基于第二视频块位于与第一视频块不同的序列、图片、片、条带、编解码树单元(CTU)或块内，第二视频块的第二划分模式或第二Merge索引中的一个或多个不同于第一视频块；以及执行(3415)第一视频块和第二视频块的进一步处理。

[0600] 图35是处理视频的方法3500的流程图。方法3500包括：识别(3505)第一视频块的特性；基于该特性确定(3510)要应用于第一视频块的几何预测模式的状态，几何预测模式的状态是以下中的一个：将几何预测模式应用于第一视频块的启用状态、或者不将几何预测模式应用于第一视频块的禁用状态；以及根据几何预测模式的状态执行(3515)第一视频块的进一步处理。

[0601] 图36是处理视频的方法3600的流程图。方法3600包括：执行确定(3605)将第一视频块分割以包括第一预测部分和第二预测部分，第一预测部分和第二预测部分是非矩形和非正方形；将与第一预测部分相关联的第一MV预测候选、以及与第二预测部分相关联的第二MV预测候选添加(3610)到与第一视频块相关联的运动候选列表，其中从与先前编解码块相关联的单个MV预测候选中导出第一MV预测候选和第二MV预测候选；以及使用运动候选列表执行(3615)第一视频块的进一步处理。

[0602] 图37是处理视频的方法3700的流程图。方法3700包括：由处理器执行确定将第一视频块分割以包括非矩形和非正方形的第一预测部分；将与第一预测部分相关联的第一运动矢量(MV)预测候选添加到与第一视频块相关联的运动候选列表；修改第一MV预测候选以细化用于导出第一视频块的最终预测块的MV候选；以及使用最终预测块执行第一视频块的进一步处理。

[0603] 图38是处理视频的方法3800的流程图。方法3800包括：执行确定(3805)将第一视频块分割以包括非矩形和非正方形的第一预测部分；将与第一预测部分相关联的第一运动矢量(MV)预测候选添加(3810)到与第一视频块相关联的运动候选列表；并且其中，从具有加权因子的双向预测MV预测候选导出第一MV预测候选；以及使用不具有加权因子的运动候选列表执行(3815)第一视频块的进一步处理。

[0604] 图39是处理视频的方法3900的流程图。方法3900包括：执行确定(3905)将第一视频块分割以包括非矩形和非方形的第一预测部分，并且根据TPM和多假设对第一视频块进行编解码；将与第一预测部分相关联的第一运动矢量(MV)预测候选添加(3910)到与第一视频块相关联的运动候选列表中；以及使用运动候选列表执行(3915)第一视频块的进一步处理。

[0605] 图40是处理视频的方法4000的流程图。方法4000包括：执行确定(4005)将第一视频块分割以包括非矩形和非方形的第一预测部分，并且其中不应用重叠块运动补偿(OBMC)；以及使用第一预测部分执行(4010)第一视频块的进一步处理。

[0606] 图41是处理视频的方法4100的流程图。方法4100包括：执行确定(4105)将第一视频块分割以包括2个以上预测部分，所述执行包括：确定将第一视频块分割为每一部分都是非矩形和非正方形的第一预测部分、第二预测部分、第三预测部分和第四预测部分；以及使用第一预测部分、第二预测部分、第三预测部分和第四预测部分执行(4110)第一视频块的进一步处理。

[0607] 参考方法3300、3400、3500、3600、3700、3800、3900、4000和4100，在本文的第4节中描述了确定用于编解码的候选及其使用的一些示例。例如，如第4节所述，可以使用非正方形和非矩形的预测部分来处理视频块。

[0608] 参考方法3300、3400、3500、3600、3700、3800、3900、4000和4100，可以将视频块编解码在视频比特流中，其中可以通过使用与运动信息预测相关的比特流生成规则来实现比

特效率。

[0609] 所述方法可以包括:其中,从可选时域运动矢量预测(ATMVP)候选导出第一MV预测候选。

[0610] 所述方法可以包括:其中,从仿射候选导出第一MV候选。

[0611] 所述方法可以包括:其中,添加到运动候选列表的第一MV预测候选包括多个运动信息集,所述多个运动信息集中的一个指示基于先前编解码的视频块的常规运动候选,所述常规运动候选用于导出第一MV预测候选。

[0612] 所述方法可以包括:其中,多个运动信息集基于第一视频块内的第一预测部分的位置。

[0613] 所述方法可以包括:其中,将所述第一视频块分割以包括包含所述第一预测部分的M个预测部分,并且所述多个运动信息集的数目为M。

[0614] 所述方法可以包括:其中,所述第一视频块还被分割以包括非矩形和非方形的第二预测部分,并且其中,使用运动候选列表进一步处理第一视频块,包括使用第一MV预测候选来处理第一预测部分和第二预测部分。

[0615] 所述方法可以包括:其中,第一预测部分和第二预测部分的划分模式的指示范围是 $M \times N \times N$,其中M是第一视频块被分割成的预测部分的数目,并且其中,N是要添加到运动候选列表中的允许的Merge候选的数目。

[0616] 所述方法可以包括:由处理器确定将指示运动候选列表中的单个MV预测候选的标志用于第一预测部分和第二预测部分。

[0617] 所述方法可包括:其中,运动候选列表中的每个MV预测候选都基于第一预测部分的双向预测运动候选,并且第二预测部分基于单独的MV预测候选。

[0618] 所述方法可以包括:其中,第一视频块还被分割以包括非矩形和非正方形的第二预测部分,并且其中,第一预测部分被分配列表0运动信息,并且其中,第二预测部分被分配列表1运动信息。

[0619] 所述方法可以包括:其中,对于位于基于第一预测部分的加权区域的样点,加权因子组不同,并且从不同的MV预测候选预测第二预测部分。

[0620] 所述方法可以包括:其中,还将所述方法应用为除高级运动矢量预测(AMVP)、Merge或三角分割模式(TPM)之外的附加编解码模式。

[0621] 所述方法可以包括:修改运动候选列表以细化用于导出第一视频块的最终预测块的MV候选。

[0622] 所述方法可以包括:其中,利用第一预测分割的运动信息和第一视频块的第二预测分割的运动信息,应用解码器侧运动矢量细化(DMVR),以导出第一视频块的运动信息。

[0623] 所述方法可以包括:其中,在TPM之后应用双向光流(BIO)。

[0624] 所述方法可以包括:其中,存储细化的运动信息。

[0625] 所述方法可以包括:其中,第一预测部分是三角形,并且第一预测部分与帧内预测组合。

[0626] 所述方法可以包括:其中,根据TPM对第一预测部分进行分割,并且TPM中使用的一个或多个参考图片是当前图片。

[0627] 所述方法可以包括:其中,根据TPM对第一视频块进行分割,以包括第一预测部分,

其中为第一预测部分分配不同于第一视频块的第二预测部分的运动信息。

[0628] 所述方法可以包括:其中,运动候选列表包括TPM编解码视频块的Merge候选。

[0629] 所述方法可以包括:其中,基于第一视频块根据TPM编解码而启用仿射模式。

[0630] 所述方法可以包括:其中,将单向预测用于第一预测部分,所述第一预测部分不位于加权区域。

[0631] 所述方法可以包括:其中,第一预测部分内的不同子块的参考图片相同。

[0632] 所述方法可以包括:其中,从ATMVP候选导出运动候选列表中的MV预测候选中的每一个。

[0633] 所述方法可以包括:其中,检查Merge列表中用于Merge候选列表构造处理中的一个或两个空域或时域块。

[0634] 所述方法可以包括:其中,启用双向预测,并且双向预测常规运动候选与加权因子相关联,并且其中,在不使用加权因子的情况下导出TPM候选。

[0635] 所述方法可以包括:其中,启用双向预测,并且双向预测常规运动候选与加权因子相关联,并且其中,使用加权因子导出TPM候选。

[0636] 所述方法可以包括:其中,对不位于第一视频块的加权区域的样点启用双向预测,所述第一视频块根据TPM编解码。

[0637] 所述方法可以包括:其中,两个参考图片相同。

[0638] 所述方法可以包括:其中,为第一视频块的每个分割分配一个参考图片。

[0639] 所述方法可以包括:其中,根据TPM对第一视频块进行编解码,并且对第一视频块应用多假设。

[0640] 所述方法可以包括:其中,使用当前设计生成最终预测块,并且其中,根据单向并且基于不同于最终预测块的另一参考图片和运动矢量生成第二预测块。

[0641] 所述方法可以包括:其中,使用当前设计生成最终预测块,并且其中,根据双向并且基于不同于最终预测块的另一参考图片和运动矢量生成第二预测块。

[0642] 所述方法可以包括:其中,对第一视频块的每个分割应用双向预测或多假设预测,并且其中,由双向预测或多假设导致的加权区域的预测块的加权结果被用作最终预测值。

[0643] 所述方法可以包括:其中,根据TPM对第一视频块进行编解码,并且不应用重叠块运动补偿(OBMC)。

[0644] 所述方法可以包括:其中,不对位于加权区域的样点应用OBMC,并且其中,将OBMC应用于位于非加权区域的样点。

[0645] 所述方法可以包括:其中,第一视频块被分割为还包括非矩形和非正方形的第二预测部分。

[0646] 所述方法可以包括:其中,第一视频块被分割为还包括第三预测部分和第四预测部分,第三预测部分和第四预测部分是非矩形和非正方形的。

[0647] 所述方法可以包括:其中,第一、第二、第三或第四预测部分中的两个被帧间编解码,并且另外两个不被帧间编解码。

[0648] 所述方法可以包括:其中,将混合应用于位于第一预测部分和第二预测部分之间的边缘周围的样点。

[0649] 所述方法可以包括:其中,与第一分割模式、第一Merge索引和第一编解码值相关

联的映射基于构造的Merge候选列表。

[0650] 所述方法可以包括:其中,与第一分割模式、第一Merge索引和第一编解码值相关联的映射基于与Merge候选相关联的编解码模式。

[0651] 所述方法可以包括:其中,基于在Merge标志之前编解码的指示TPM使用的标志为非Merge模式启用TPM。

[0652] 所述方法可以包括:其中,基于编解码的指示TPM使用的标志为非Merge模式启用TPM,并且其中,基于第一视频块未用TPM编解码而不信令通知Merge标志。

[0653] 所述方法可以包括:其中,基于第一视频块被帧间编解码而信令通知Merge标志。

[0654] 所述方法可以包括:其中,从第一视频块的相邻块继承第一分割模式。

[0655] 所述方法可以包括:其中,从第一视频块的时域块继承第一分割模式。

[0656] 所述方法可以包括:其中,分别信令通第一分割模式和第一Merge索引。

[0657] 所述方法可以包括:其中,指示第一划分模式的TPM的使用的标志使用旁路编解码。

[0658] 所述方法可以包括:其中,使用基于第一视频块的宽度或高度的一个或二者的上下文来编解码所述标志。

[0659] 所述方法可以包括:其中,使用基于编解码深度的上下文来编解码所述标志。

[0660] 所述方法可以包括:其中,使用基于第一视频块的块形状或第一视频块的宽高比的上下文来编解码所述标志。

[0661] 所述方法可以包括:其中,使用基于第一视频块的相邻块的编解码信息的上下文来编解码所述标志。

[0662] 所述方法可以包括:其中,第一划分模式与TPM一致,并且其中,第一视频块的分割包括第一区域和第二区域,第一区域和第二区域用不同模式编解码。

[0663] 所述方法可以包括:其中,第一区域或第二区域中的一个是在帧内编解码的,并且另一个是在帧间编解码的。

[0664] 所述方法可以包括:其中,第一区域或第二区域中的一个是在Merge编解码的,并且另一个是在AMVP编解码的。

[0665] 所述方法可以包括:其中,第一视频块的特征包括颜色分量,其中,几何预测模式的状态是禁用状态,并且其中,几何预测模式处于禁用状态,而不信令通知指示几何预测模式处于禁用状态的标志的使用。

[0666] 所述方法可以包括:其中,第一视频块的特征包括:色度块的宽度或高度小于阈值范围。

[0667] 所述方法可以包括:其中,特征包括:第一视频块的尺寸为 $N \times M$ 或 $L \times N$,其中 N 是4。

[0668] 所述方法可以包括:其中,特征包括:色度格式为4:2:0。

[0669] 所述方法可以包括:其中,特征包括:第一视频块的宽度、高度或宽度和高度二者大于或等于阈值 L 。

[0670] 所述方法可以包括:其中,阈值 L 为64。

[0671] 所述方法可以包括:其中,特征包括:第一视频块的宽度、高度或宽度和高度二者小于或等于阈值 L 。

[0672] 所述方法可以包括:其中,阈值 L 为4或8。

[0673] 所述方法可以包括:其中,阈值L基于包括块尺寸、图片类型或时域层索引中的一个或多个的编解码信息。

[0674] 所述方法可以包括:其中,特征是第一视频块的颜色分量。

[0675] 所述方法可以包括:其中,基于在视频参数集(VPS)、序列参数集(SPS)、图片参数集(PPS)、图片标头、片组标头、条带标头、编解码树单元(CTU)行的组、区域或CTU组中信令通知来启用或应用TPM。

[0676] 所述方法可以包括:其中,几何预测模式是TPM。

[0677] 所述方法可以包括:其中,非矩形和非方形预测部分是三角形。

[0678] 在一些实施例中,视频处理方法包括:根据规则执行视频块与视频块的比特流表示之间的转换,该规则规定了将编解码的跳过/Merge模式与第二编解码模式分别对待,其中第二编解码模式是三角分割模式或MHIntra模式或子块Merge列表模式。

[0679] 在一些实施例中,比特流表示被配置如下:(1)在为视频块启用第二编解码模式的情况下,则比特流表示中的Merge标志被信令通知为具有假值,或(2)在当前块使用跳过/Merge模式的情况下,则在比特流表示中显示地信令通知第二编解码模式的Merge列表的对应指示。

[0680] 另一种视频编解码方法包括:根据规则执行视频块和视频块的比特流表示之间的转换,所述规则规定了在转换中使用多个子块编解码工具的指示的特定顺序,其中该特定顺序是以下中的一个:(a)与运动矢量残差的合并(MMVD)、随后是子块Merge列表、随后是三角分割模式TPM、随后是MHIntra,或者(b)MMVD、随后是子块Merge列表、随后是MHIntra、随后是TPM。

[0681] 在一些实施例中,该规则进一步规定对其他视频部分或基于不同的视频条件使用不同的特定顺序。

[0682] 在一些实施例中,其它视频部分包括:其它视频条带、或其它视频编解码单元、或其它视频预测单元、或其它视频编解码单元行、或其它视频图片、或不是视频块的视频序列的其它视频序列。

[0683] 在一些实施例中,不同的视频条件包括:低延迟检查标志的不同值、或先前编解码信息的不同值、或其他视频块的尺寸。

[0684] 另一种视频处理方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻块来规定运动矢量预测,其中,所述规则包括:(1)排除位于不同最大编解码单元(LCU)行中的相邻块,或(2)规定固定分辨率值,或(3)在使用仿射模式编解码视频块的情况下,使位于不同LCU行中的仿射编解码相邻块不可用。

[0685] 另一种视频处理方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻块来规定运动矢量预测,其中,所述规则包括:选择性地使用相邻块,如:(a)在相邻块位于不同最大编解码单元(LCU)行的情况下,禁止将相邻视频块的信息用于视频块的上下文建模,或(b)在转换期间使用仅有的一个上下文来编解码语法元素,或(c)将旁路编解码应用于仅有的一个语法元素。

[0686] 在一些实施例中,仅有的一个语法元素包括:a.alf_ctb_flag、b.qt_split_cu_

flag、c.mtt_split_cu_flag、d.mtt_split_cu_flag、e.cu_skip_flag、f.amvr_mode、g.merge_subblock_flag、h.merge_triangle_flag或i.inter_affine_flag。

[0687] 在一些实施例中,转换包括:从比特流表示生成视频块的像素值、或从视频块的像素值生成比特流表示。

[0688] 另一种视频处理方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻视频块来规定运动矢量预测,其中,所述规则包括:基于用于解码当前视频块的相邻视频块,将关于相邻视频块的信息用于上下文建模。

[0689] 所述方法可以包括:确定相邻视频块在包括当前视频块的部分之外,所述部分是:编解码树单元(CTU)、CTU行、区域、视频数据处理单元、条带、片、片组或图片边界,并且其中,所述规则包括:(1)从解析技术中排除关于相邻视频块的信息,或者(2)在解析技术中使用关于相邻视频块的信息,或者(3)将关于相邻视频块的信息设置为解析技术的默认值,或者(4)在解析技术中将关于相同部分中其它相邻视频块的信息用作替代。

[0690] 所述方法可以包括:其中,关于相邻视频块的信息来自基于相邻视频块的编解码信息的语法元素的预定义相邻位置。

[0691] 另一种视频处理方法包括:根据编解码规则执行视频块和当前视频块的比特流表示之间的转换,其中所述规则规定:在视频块的自适应运动矢量分辨率(AMVR)编解码模式中选择性使用相邻视频块来规定运动矢量预测,其中,所述规则包括:基于用于解码当前视频块的相邻视频块,将关于相邻视频块的信息用于上下文建模,所使用的相邻视频块不包括与当前CTU行不同的CTU行中的、或者与当前CTU不同的CTU中的视频块。

[0692] 所述方法可以包括:其中,来自被排除的相邻视频块的信息不可用于上下文建模。

[0693] 所述方法可以包括:其中,使用默认值来代替被排除的相邻视频块的信息。

[0694] 所述方法可以包括:其中,使用来自当前CTU行或当前CTU内的可选相邻视频块的信息来代替被排除的视频块。

[0695] 所述方法可以包括:其中,语法元素包括以下的一个或多个:alf_ctb_flag、qt_split_cu_flag、mtt_split_cu_flag、cu_skip_flag、amvr_mode、inter_affine_flag、merge_triangle_flag和merge_subblock_flag、cu_skip_flag、pred_mode_flag、pcm_flag、intra_luma_ref_idx、intra_luma_mpm_flag、intra_luma_mpm_idx、intra_luma_mpm_remainder、intra_chroma_pred_mode、merge_flag、inter_pred_idc、inter_affine_flag、cu_affine_type_flag、ref_idx_l0、mvp_l0_flag、ref_idx_l1、mvp_l1_flag、amvr_flag、amvr_4pel_flag、gbi_idx、cu_cbf、mmvd_flag、或mmvd_merge_flag。

[0696] 所述方法可以包括:其中,上下文建模包括使用pred_mode_flag。

[0697] 所述方法可以包括:其中,从以下一个或多个中导出pred_mode_flag:ctx=A1是帧内?1:0并且ctx+=B1是帧内?1:0,ctx=(A1是帧内) || (B1是帧内)?1:0,ctx=A1是帧内或帧间帧内?1:0并且ctx+=B1是帧内或帧间帧内?1:0,或者ctx=(A1是帧内或帧间帧内) | (B1是帧内或帧间帧内)?1:0。

[0698] 所述方法可以包括,其中,从位于与当前视频块的角邻近的相邻块导出pred_mode_flag。

[0699] 所述方法可以包括:其中,用其他类型的语法元素替换pred-mode_flag。

[0700] 所述方法可以包括:其中,当推导上下文时,将不可用的相邻块作为内部编解码处理。

[0701] 所述方法可以包括:其中,当推导上下文时,将不可用的相邻块作为帧内编解码处理。

[0702] 所述方法可以包括:其中,当推导上下文时,将不可用的相邻块作为帧间帧内编解码处理。

[0703] 所述方法可以包括:其中,相邻视频块不在与当前视频块相同的编解码树单元(CTU)行内,并且在推导上下文时将相邻视频块视为帧间编解码。

[0704] 所述方法可以包括:其中,相邻视频块不在与当前视频块相同的CTU行内,并且在推导上下文时将相邻视频块视为帧内编解码。

[0705] 所述方法可以包括:其中,相邻视频块不在与当前视频块相同的CTU行内,并且在推导上下文时将相邻视频块视为帧间帧内编解码。

[0706] 可优选地将以下技术解决方案列表并入视频编解码器或解码器装置的一些实施例中。

[0707] 在先前的章节中提供了以下解决方案的其它特征和变体(例如,项目1至4、17和21)。

[0708] 示例解决方案集A

[0709] 1.一种视频处理方法(例如,图44所示的方法4400),包括:确定(4402)视频的视频区域的视频块和所述视频的编解码表示之间的转换使用几何分割模式,其中所述视频块被分割成多个预测分割,所述多个预测分割至少包括第一预测分割;使用与所述视频块相关联的一个或多个子块Merge候选,将N个运动信息候选添加(4404)到所述视频块的运动候选列表,其中N是正整数;从所述运动候选列表导出(4406)所述多个预测分割的运动候选;以及基于所述多个预测分割的所述运动候选执行(4408)所述转换。

[0710] 2.根据解决方案1所述的方法,其中,将子块时域运动矢量预测候选用于导出所述N个MV候选。

[0711] 3.根据解决方案1所述的方法,其中,将仿射运动候选用于导出所述N个MV候选。

[0712] 4.根据解决方案1所述的方法,其中,将来自一个或多个子块的一个或多个子块运动候选的M个代表性MV候选用于导出M个MV候选,其中M是正整数。

[0713] 5.根据解决方案4所述的方法,其中,基于所述一个或多个子块的位置选择所述M个代表性MV候选。

[0714] 6.根据解决方案4至5中任一项所述的方法,其中,M等于为所述转换将所述视频块进行分割的分割的数目。

[0715] 7.根据解决方案4至5中任一项所述的方法,其中,M=1,并且从与所述第一预测分割的中心子块相关联的运动信息中选择代表性MV候选。

[0716] 8.一种视频处理方法(例如,图45所示的方法4500),包括:为视频的视频块和所述视频的编解码表示之间的转换确定(4502)所述视频块被分割成多个预测分割,所述多个预测分割至少包括第一预测分割;将第一运动信息候选添加(4504)到与所述第一预测分割相关联的第一候选列表,并且将第二运动信息候选添加到与第二预测分割相关联的第二候选列表,其中所述第一运动信息候选和所述第二运动信息候选继承自单个运动候选;以及使

用所述第一运动信息候选和/或所述第二运动信息候选执行(4506)所述转换。

[0717] 9. 根据解决方案8所述的方法,其中,所述第一预测部分和所述第二预测部分的划分模式和MV候选索引的指示范围是 $M \times N \times N$,其中M是所述第一视频块被分割成的预测部分的数目,并且其中,N是要添加到所述运动候选列表中的允许的Merge候选的数目。

[0718] 10. 根据解决方案8或9所述的方法,其中,所述比特流表示中的标志指示将所述运动候选列表中的所述单个MV预测候选用于所述第一预测部分和所述第二预测部分二者。

[0719] 11. 根据解决方案8所述的方法,其中,所述运动候选列表中的所述预测候选是双向预测运动候选。

[0720] 12. 根据解决方案8所述的方法,其中,所述第一MV候选被分配列表0运动信息,并且所述第二MV候选被分配列表1运动信息。

[0721] 13. 根据解决方案8至12中任一项所述的方法,其中,根据所述划分模式,将特定加权因子组用于位于所述视频块的加权区域中的样点,其中,所述特定加权因子组不同于用于另一视频块的另一加权因子组,其中,不同的运动候选被用于确定所述另一视频块的预测部分的候选运动矢量。

[0722] 14. 根据解决方案13所述的方法,其中,所述特定加权因子组对应于位于所述加权区域中的所述样点的平均。

[0723] 15. 根据解决方案8所述的方法,其中,所述转换将所述方法用作除了在转换期间使用的高级运动矢量预测(AMVP)、Merge或三角分割模式(TPM)之外的附加编解码模式。

[0724] 16. 根据解决方案1至15中任一项所述的方法,其中,所述执行所述转换包括:为所述视频块的预测块导出细化的最终运动矢量。

[0725] 17. 根据解决方案16所述的方法,其中,所述导出所述细化的最终运动矢量包括:对所述第一视频块的所述第一预测分割的运动信息和所述第二预测分割的运动信息使用解码器侧运动矢量细化(DMVR)工具,以导出所述视频块的运动信息。

[0726] 18. 根据解决方案16所述的方法,其中,所述执行所述转换包括:将双向光流(BIO)工具应用于三角分割模式预测之后的结果。

[0727] 19. 根据解决方案17或18所述的方法,其中,所述第一预测部分是从第一参考列表单向预测的,并且所述第二预测部分是使用第二参考列表双向预测或单向预测的,然后使用所述第二参考列表中的所述第二预测部分的运动信息应用所述DMVR工具或所述BIO工具。

[0728] 20. 根据解决方案16所述的方法,还包括:由于所述第一预测部分是从参考列表LX单向预测的,在所述转换期间存储所述参考列表LX的运动信息,其中X是0或1。

[0729] 21. 根据解决方案16所述的方法,其中,所述转换使用最终预测,所述最终预测是基于所述几何分割模式的预测结果和基于帧内预测模式的预测结果的加权和。

[0730] 22. 根据解决方案16所述的方法,在几何分割模式中使用的一个或多个参考图片是当前图片。

[0731] 23. 一种视频处理方法(例如,图46所示的方法4600),包括:确定(4602)视频的视频块和所述视频的编解码表示之间的转换使用启用了子块的几何分割模式,其中,所述视频块被分割以包括预测分割,使得所述预测分割的子块被分配其子块运动信息;以及基于所述确定执行(4604)所述转换。

- [0732] 24. 根据解决方案23所述的方法,其中,所述子块运动信息被用作用于所述视频块的所述转换的候选列表中的候选。
- [0733] 25. 根据解决方案23所述的方法,其中,基于所述第一视频块根据所述几何分割模式编解码而启用仿射模式。
- [0734] 26. 根据解决方案23至25中任一项所述的方法,其中,所述转换对不在所述视频块的加权区域中的子块限制为仅使用单向预测。
- [0735] 27. 根据解决方案23所述的方法,其中,所述转换基于规则,所述规则规定了对所述视频块的任何给定预测部分内的所有子块使用相同的参考图片。
- [0736] 28. 根据解决方案23至27中任一项所述的方法,其中,所述转换包括:从与所述子块相关联的子块时域运动矢量预测(SbTMVP)候选中导出运动候选列表中的一个或多个运动矢量预测候选。
- [0737] 29. 根据解决方案28所述的方法,其中,所述转换还使用以下规则:所述预测分割内的每个子块继承所述SbTMVP候选的列表0运动信息的运动信息。
- [0738] 30. 根据解决方案23所述的方法,其中,所述转换还使用以下规则:用与所述转换期间使用的子块Merge候选列表处理的相同的顺序检查所述预测分割的Merge列表。
- [0739] 31. 根据解决方案1至30中任一项所述的方法,其中,所述几何分割模式使用不同的编解码模式来表示所述视频块的所述预测分割。
- [0740] 32. 根据解决方案31所述的方法,其中,所述第一分割模式使用帧内编解码,并且所述第二分割模式使用帧间编解码。
- [0741] 33. 根据解决方案31所述的方法,其中,所述第一预测分割使用Merge编解码,并且所述第二预测分割使用高级运动矢量预测工具。
- [0742] 34. 根据解决方案1至33中任一项所述的方法,其中,所述比特流表示信号中的指示信令通知所述视频块对所述几何分割模式的使用。
- [0743] 35. 根据解决方案34所述的方法,其中,所述指示在序列参数集级、或图片参数集级、或视频参数集级、或图片标头、或条带标头、或片组标头、或编解码树单元级。
- [0744] 36. 根据解决方案1至35中任一项所述的方法,其中,由于所述视频块满足尺寸条件而启用所述几何分割模式。
- [0745] 37. 根据解决方案36所述的方法,其中,所述尺寸条件规定:由于所述视频块具有大于阈值的尺寸而使用所述几何分割模式。
- [0746] 38. 根据解决方案36所述的方法,其中,所述尺寸条件规定:由于所述视频块具有小于阈值的尺寸而使用所述几何分割模式。
- [0747] 39. 根据解决方案37至38中任一项所述的方法,其中,在序列参数集、或图片参数集、或图片、或条带、或片组、或片级的编解码表示中信令通知所述阈值。
- [0748] 40. 根据解决方案37至38中任一项所述的方法,其中,所述尺寸是与所述视频块相关联的图片类型或时域层索引的函数。
- [0749] 41. 根据解决方案1至33中任一项所述的方法,其中,由于所述视频块具有预定义的宽高比而将所述几何分割模式用于所述转换。
- [0750] 在先前的章节中提供了以下解决方案的其它特征和变体(例如,项目5、7、8、9和21)。

[0751] 示例解决方案集B

[0752] 1. 一种视频处理方法(例如,图47A所示的方法4700),包括:根据继承规则,为使用其中视频块被分割成多个预测分割的几何分割模式的视频的视频块和所述视频的编解码表示之间的转换确定(4702)是否根据继承规则,在所述视频块的样点位置处启用从用于所述预测分割的运动信息推导的运动候选继承广义双向预测(GBi)加权因子;以及基于所述确定执行(4704)所述转换。

[0753] 2. 根据解决方案1所述的方法,其中,所述继承规则规定:对所述视频块的所有位置禁用所述继承。

[0754] 3. 根据解决方案2所述的方法,还包括:在不考虑与所述运动候选相关联的所述GBi加权因子的情况下执行所述转换。

[0755] 4. 根据解决方案2所述的方法,其中,将所述视频块的GBi加权因子索引设置为0。

[0756] 5. 根据解决方案1所述的方法,其中,所述继承规则规定:对所述视频块的所有位置启用所述继承。

[0757] 6. 根据解决方案5所述的方法,还包括:对于所述预测分割,将对应的GBi加权因子设置为等于与用于导出所述视频块的运动信息的运动候选相关联的GBi加权因子。

[0758] 7. 根据解决方案1所述的方法,其中,对所述视频块的加权区域之外的样点位置禁用所述继承,并且在所述加权区域中的样点位置处启用所述继承,其中所述加权区域跨越两个预测分割的边界。

[0759] 8. 根据解决方案7所述的方法,还包括:对于不包括所述加权区域的所述预测分割中的样点位置,将所述对应的GBi加权因子设置为等于0。

[0760] 9. 根据解决方案4所述的方法,还包括:对于所述预测分割中的加权区域,将对应的GBi加权因子设置为等于与用于导出所述视频块的运动信息的所述运动候选相关联的GBi加权因子。

[0761] 10. 一种视频处理方法(例如,图47B所示的方法4750),包括:根据规则,为使用其中视频块被分割成多个预测分割的几何分割模式的视频的视频块和所述视频的编解码表示之间的转换,确定(4752)是否至少在所述视频块的样点位置处启用双向预测的使用;以及基于所述确定执行(4754)所述转换。

[0762] 11. 根据解决方案10所述的方法,其中,所述规则规定:对所述视频块的非加权区域中的样点位置启用双向预测。

[0763] 12. 根据解决方案11所述的方法,其中,所述规则还规定:仅通过使用彼此相同的两个参考图片,对所述视频块的所述非加权区域中的样点位置启用双向预测。

[0764] 13. 根据解决方案11至12中任一项所述的方法,其中,所述双向预测将第一参考帧用于第一预测分割,将第二参考帧用于第二预测分割,并且将第三参考帧用于所述第一预测分割和所述第二预测分割二者,其中,所述视频块由所述第一预测分割和所述第二预测分割组成。

[0765] 14. 一种视频处理方法(例如,图48所示的方法4800),包括:为使用其中将视频块至少分割成第一预测分割和第二预测分割的几何分割模式的视频的视频块和所述视频的编解码表示之间的转换,来自运动信息的第一预测块和第二预测块,确定(4802)最终预测块,所述运动信息与为所述第一预测分割的样点位置导出的第一组运动信息、为所

述第二预测分割的样点位置导出的第二组运动信息不同;以及基于所述最终预测块执行(4804)所述转换;其中所述视频块使用多假设技术编解码。

[0766] 15. 根据解决方案14所述的方法,其中,所述第二预测块是使用不同于所述第一组运动信息和所述第二组运动信息的运动矢量或参考图片单向预测地生成的。

[0767] 16. 根据解决方案14所述的方法,其中,所述第二预测块是使用两个附加参考图片双向预测地生成的。

[0768] 17. 根据解决方案14至16中任一项所述的方法,其中,所述第二预测块是使用所述第一预测分割的第三组运动信息和所述第二预测分割的第四组运动信息生成的。

[0769] 18. 根据解决方案14至17中任一项所述的方法,其中,从所述第一预测块或所述第二预测块生成不包括加权区域的所述第一预测分割和所述第二预测分割中的所述最终预测块的样点;并且其中,对所述加权区域中的样点使用来自所述第一预测块和所述第二预测块的加权预测;其中,所述加权区域跨越两个预测分割的所述边界。

[0770] 19. 根据前述解决方案中任一项所述的方法,其中,所述转换还基于重叠块运动补偿(OBMC)规则,所述OBMC规则规定是否在所述视频块的样点位置处启用OBMC编解码的使用。

[0771] 20. 根据解决方案19所述的方法,其中,所述OBMC规则规定:由于所述视频块使用所述几何分割模式而对所述转换禁用所述OBMC编解码。

[0772] 21. 根据解决方案19所述的方法,其中,所述OBMC规则规定:在所述视频块的加权区域的样点位置处禁用OBMC编解码,并在其余样点位置处启用OBMC。

[0773] 22. 根据解决方案1至21中任一项所述的方法,其中,将所述视频块分割成两个或多个分割。

[0774] 23. 根据解决方案22所述的方法,其中,将所述视频块分割成四个三角预测分割。

[0775] 24. 根据解决方案1至23中任一项所述的方法,其中,将所述视频块至少分割成两个非矩形和非方形的分割,使得所述至少两个分割中的至多一个被帧间编解码。

[0776] 25. 根据解决方案22至24中任一项所述的方法,其中,所述转换包括:在所述四个分割的相邻分割的边缘处的样点位置处使用加权预测平均。

[0777] 26. 根据解决方案1至25中任一项所述的方法,其中,所述比特流表示中的指示信令通知所述视频块对所述几何分割模式的使用。

[0778] 27. 根据解决方案26所述的方法,其中,所述指示位于序列参数集级、或图片参数集级、或视频参数集级、或图片标头、或条带标头、或片组标头、或编解码树单元级。

[0779] 28. 根据解决方案1至25中任一项所述的方法,其中,由于所述视频块满足尺寸条件而启用所述几何分割模式。

[0780] 29. 根据解决方案26所述的方法,其中,所述尺寸条件规定:由于所述视频块的尺寸大于阈值而使用所述几何分割模式。

[0781] 30. 根据解决方案26所述的方法,其中,所述尺寸条件规定:由于所述视频块的尺寸小于阈值而使用所述几何分割模式。

[0782] 31. 根据解决方案29至30中任一项所述的方法,其中,在序列参数集、或图片参数集、或图片、或条带、或片组、或片级的编解码表示中信令通知所述阈值。

[0783] 32. 根据解决方案29至30中任一项所述的方法,其中,所述尺寸是与所述视频块相

关联的图片类型或时域层索引的函数。

[0784] 33. 根据解决方案1至32中任一项所述的方法,其中,由于所述视频块具有预定义的宽高比而将所述几何分割模式用于所述转换。

[0785] 在先前的章节中提供了以下解决方案的其它特征和变体(例如,项目10、11、12、13和21)。

[0786] 示例解决方案集C

[0787] 1. 一种视频处理方法(例如,图49所示的方法4900),包括:为视频的视频区域的视频块和所述视频的比特流表示之间的转换,确定(4902) (1) 用于将所述视频块划分成预测分割以使得至少一个预测分割是非矩形和非正方形分割的划分模式与(2) 用于所述转换的所述分割的Merge候选的索引之间的关系、以及允许在所述视频区域级更改所述关系的所述比特流表示的格式;以及基于所述确定执行(4904)所述转换。

[0788] 2. 根据解决方案1所述的方法,其中,所述视频区域对应于视频图片的视频序列、或视频片、或视频条带、或视频编解码树单元、或编解码单元。

[0789] 3. 根据解决方案1至2中任一项所述的方法,其中,所述关系取决于所述视频块的所述转换期间使用的Merge候选列表的特性。

[0790] 4. 根据解决方案1至3中任一项所述的方法,其中,所述关系取决于与所述Merge候选相关联的编解码模式。

[0791] 5. 一种视频处理方法(例如,如图50所示的方法5000),包括:根据定义编解码表示的语法元素的语法规则,执行(5002)视频的视频块和所述视频的编解码表示之间的转换;其中,所述语法规则规定了指示所述视频块使用几何分割模式的第一标志、以及指示所述视频块使用Merge模式的第二标志的信令顺序;其中,所述几何分割模式包括:将所述视频块划分成多个预测分割,至少一个预测分割具有非矩形、非正方形形状,并且其中,所述Merge模式允许从所述视频块的相邻块继承运动矢量信息。

[0792] 6. 根据解决方案5所述的方法,其中,在所述第一标志之后信令通知所述第二标志。

[0793] 7. 根据解决方案5所述的方法,其中,所述语法规则规定:由于信令通知所述第一标志而省略信令通知所述第二标志。

[0794] 8. 根据解决方案5所述的方法,其中,基于所述视频块的编解码模式有条件地信令通知所述第二标志。

[0795] 9. 根据解决方案8所述的方法,其中,由于所述视频块被帧间编解码而信令通知所述第二标志。

[0796] 10. 根据解决方案5所述的方法,其中,所述语法规则规定:由于所述视频块以跳过模式编解码而省略第一标志。

[0797] 11. 一种视频处理方法(例如,图51所示的方法5100),包括:根据定义编解码表示的语法的语法规则,执行(5102)视频的视频块和所述视频的编解码表示之间的转换;其中,所述语法规则规定了基于是否对所述视频块的所述转换使用另一编解码工具,选择性地包括指示所述视频块使用几何分割模式的标志的信令;其中,所述几何分割模式包括:将所述视频块划分成多个预测分割,至少一个预测分割具有非矩形、非正方形形状。

[0798] 12. 根据解决方案11所述的方法,其中,所述语法规则规定:如果将高级运动矢量

预测工具用于所述视频块的所述转换,则包括所述标志。

[0799] 13. 根据解决方案11所述的方法,其中,所述语法规则规定:如果所述视频块使用Merge模式编解码,则禁用所述标志的信令通知。

[0800] 14. 一种视频处理方法(例如,如图52所示的方法5200),包括:确定(5202)为视频的视频块和所述视频的编解码表示之间的转换启用几何分割模式,其中,根据基于相邻块的几何分割模式的划分模式,将所述视频块划分成多个预测分割;以及基于所述确定执行(5204)所述转换。

[0801] 15. 一种视频处理方法(例如,图53所示的方法5300),包括:基于几何分割模式,来为视频的视频块与所述视频的编解码表示之间的转换确定(5302)划分模式,其中,根据基于相邻块的划分模式的所述划分模式,将所述视频块划分成多个预测分割;以及基于所述确定执行(5304)所述转换。

[0802] 16. 根据解决方案14或15所述的方法,其中,所述相邻块是空域相邻块。

[0803] 17. 根据解决方案14或15所述的方法,其中,所述相邻块是时域相邻块。

[0804] 18. 根据解决方案2至4中任一项所述的方法,还包括:将所述划分模式与所述视频块的运动信息一起存储。

[0805] 19. 根据解决方案2至5中任一项所述的方法,其中,所述转换包括:从常规运动候选导出所述多个分割的运动候选,使得与所述常规运动候选相关联的划分模式由所述导出的运动候选继承。

[0806] 20. 根据解决方案14或15所述的方法,其中,所述视频块从所述相邻块继承运动信息和所述划分模式二者。

[0807] 21. 根据解决方案1至20中任一项所述的方法,其中,所述比特流表示信号中的指示信令通知所述视频块对所述几何分割模式的使用。

[0808] 22. 根据解决方案21所述的方法,其中,所述指示位于序列参数集级、或图片参数集级、或视频参数集级、或图片标头、或条带标头、或片组标头、或编解码树单元级。

[0809] 23. 根据解决方案1至20中任一项所述的方法,其中,由于所述视频块满足尺寸条件而启用所述几何分割模式。

[0810] 24. 根据解决方案21所述的方法,其中,所述尺寸条件规定:由于所述视频块的尺寸大于阈值而使用所述几何分割模式。

[0811] 25. 根据解决方案21所述的方法,其中,所述尺寸条件规定:由于所述视频块的尺寸小于阈值而使用所述几何分割模式。

[0812] 26. 根据解决方案24至25中任一项所述的方法,其中,在序列参数集、或图片参数集、或图片、或条带、或片组、或片级的编解码表示中信令通知所述阈值。

[0813] 27. 根据解决方案24至25中任一项所述的方法,其中,所述尺寸是与所述视频块相关联的图片类型或时域层索引的函数。

[0814] 28. 根据解决方案1至20中任一项所述的方法,其中,由于所述视频块具有预定义的宽高比而将所述几何分割模式用于所述转换。

[0815] 在先前的章节中提供了以下解决方案的其它特征和变体(例如,项目14、15、16、21)。

[0816] 示例解决方案集D

[0817] 1. 一种视频处理方法(例如,图54所示的方法5400),包括:执行(5402)视频的视频块和所述视频的编解码表示之间的转换,其中,根据包括第一预测分割的几何分割模式的划分模式,将所述视频块分割成多个预测分割;其中,所述编解码表示包括多个语法元素,其中一个语法元素与所述划分模式相对应,并且其他语法元素与所述多个预测分割的Merge索引相对应。

[0818] 2. 根据解决方案1所述的方法,其中,使用一比特标志来信令通知所述分割模式。

[0819] 3. 根据解决方案2所述的方法,其中,将相邻视频块的相邻划分模式用于所述一比特标志的上下文编解码。

[0820] 4. 根据解决方案1至3中任一项所述的方法,其中,为所述多个预测分割中的每一个预测分割信令通知多个Merge索引。

[0821] 5. 根据解决方案1至3中任一项所述的方法,其中,使用预测编解码来信令通知所述多个预测分割的所述多个Merge索引。

[0822] 6. 根据解决方案1所述的方法,其中,用于信令通知所述Merge索引的编解码方案取决于用于信令通知所述划分模式的编解码方案。

[0823] 7. 根据解决方案1所述的方法,其中,用于信令通知所述划分模式的编解码方案取决于用于信令通知所述Merge索引的编解码方案。

[0824] 8. 根据解决方案1所述的方法,其中,所述转换包括:从所述编解码表示中解析所述多个语法元素。

[0825] 9. 一种视频处理方法(例如,如图55所示的方法5500),包括:确定(5502)视频的视频块和所述视频的编解码表示之间的转换基于几何分割模式,在所述几何分割模式中,根据包括第一预测分割的划分模式将所述视频块分割成多个预测分割;以及基于所述确定执行(5504)所述转换;其中,使用基于上下文的编解码在所述编解码表示中信令通知所述视频块的所述几何分割模式,其中所述上下文取决于所述视频块的编解码条件。

[0826] 10. 根据解决方案9所述的方法,其中,所述编解码条件对应于所述视频块的高度或宽度、或所述高度与所述宽度之比。

[0827] 11. 根据解决方案9所述的方法,其中,所述上下文取决于分割树内的所述视频块的编解码深度。

[0828] 12. 根据解决方案9所述的方法,其中,所述编解码条件对应于相邻块的编解码信息。

[0829] 13. 根据解决方案12所述的方法,其中,所述视频块具有宽度W和高度H,这里,所述视频块的左上样点位置表示为(0,0),并且所述相邻块对应于位置(W/2,-1)、(W/2-1,-1)或(W-1,-1)处的上方相邻块。

[0830] 14. 根据解决方案13所述的方法,其中,由于所述宽度和所述高度之比大于阈值而使所述上下文完全由所述上方相邻块确定。

[0831] 15. 根据解决方案13所述的方法,其中,所述视频块具有宽度W和高度H,并且其中,所述视频块的左上样点位置表示为(0,0),并且所述相邻块对应于位置(-1,H/2)、(-1,H/2-1)、(-1,H-1)处的左侧相邻块。

[0832] 16. 根据解决方案15所述的方法,其中,由于所述高度和所述宽度之比大于阈值而

使所述上下文完全由所述左侧相邻块确定。

[0833] 17. 根据解决方案13和15中任一项所述的方法,其中,所述上下文取决于所述上方相邻块和所述左侧相邻块的组合。

[0834] 18. 一种视频处理方法(例如,图56所示的方法5600),包括:执行(5602)视频的视频块和所述视频的编解码表示之间的转换,其中,根据具有第一预测分割的划分模式,使用几何预测模式将所述视频块分割成多个预测分割;其中,在所述编解码表示中为候选列表的两个候选索引项分配码字,并且其中,所述编解码表示的格式允许改变所述视频的片段中所述码字的所述分配。

[0835] 19. 根据解决方案18所述的方法,其中,所述视频的所述片段对应于所述视频块、或视频编解码树单元、或视频片、或视频图片、或视频序列。

[0836] 20. 根据解决方案18或19中任一项所述的方法,其中,基于所述两个候选索引项是否对应于来自相同常规运动候选的两个候选而改变所述分配。

[0837] 21. 根据解决方案20所述的方法,其中,所述分配使得选择分配给第一对候选索引的第一码字小于分配给第二对候选索引的第二码字,所述第一对候选索引表示从不同的常规运动候选导出的候选,所述第二对候选索引表示从相同的常规运动候选导出的候选。

[0838] 22. 根据解决方案18或19中任一项所述的方法,其中,基于所述两个候选索引项是否对应于来自相同预测方向或相同参考图片列表的两个候选而改变所述分配。

[0839] 23. 根据解决方案20所述的方法,其中,所述分配使得选择分配给第一对候选索引的第一码字小于分配给第二对候选索引的第二码字,所述第一对候选索引具有不同预测方向的候选,所述第二对候选索引表示具有相同预测方向的候选。

[0840] 24. 根据解决方案1至23中任一项所述的方法,其中,所述比特流表示信号中的指示信令通知所述视频块对所述几何分割模式的使用。

[0841] 25. 根据解决方案24所述的方法,其中,所述指示在序列参数集级、或图片参数集级、或视频参数集级、或图片标头、或条带标头、或片组标头、或编解码树单元级。

[0842] 26. 根据解决方案1至25中任一项所述的方法,其中,由于所述视频块满足尺寸条件而启用所述几何分割模式。

[0843] 27. 根据解决方案26所述的方法,其中,所述尺寸条件规定:由于所述视频块的尺寸大于阈值而使用所述几何分割模式。

[0844] 28. 根据解决方案26所述的方法,其中,所述尺寸条件规定:由于所述视频块的尺寸小于阈值而使用所述几何分割模式。

[0845] 29. 根据解决方案27至28中任一项所述的方法,其中,在序列参数集、或图片参数集、或图片、或条带、或片组、或片级的编解码表示中信令通知所述阈值。

[0846] 30. 根据解决方案27至28中任一项所述的方法,其中,所述尺寸是与所述视频块相关联的图片类型或时域层索引的函数。

[0847] 31. 根据解决方案1至30中任一项所述的方法,其中,由于所述视频块具有预定义的宽高比而将所述几何分割模式用于所述转换。

[0848] 在先前的章节中提供了以下解决方案的其它特征和变体(例如,项目18至21)。

[0849] 示例解决方案集E

[0850] 1. 一种视频处理方法(例如,图57所示的方法5700),包括:为视频的视频块和所述

视频的编解码表示之间的转换,确定(5702)是否基于使用所述视频块的特性的规则来启用对几何分割模式的使用;以及根据所述确定执行(5704)所述转换,其中,所述几何分割模式将所述视频块分割成多个预测分割。

[0851] 2. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块表示所述视频的颜色分量而禁用所述几何分割模式。

[0852] 3. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块是具有小于阈值的尺寸或宽度的色度块而禁用所述几何分割模式。

[0853] 4. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块的尺寸为 $N \times M$ 或 $K \times N$ 而禁用所述几何分割模式,其中 K 、 M 、 N 是彼此不同的整数。

[0854] 5. 根据解决方案4所述的方法,其中, N 等于4或128。

[0855] 6. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块的宽度和/或高度大于阈值 L 而禁用所述几何分割模式,其中 L 是整数。

[0856] 7. 根据解决方案6所述的方法,其中, L 等于64。

[0857] 8. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块的宽度和/或高度小于阈值 L 而禁用所述几何分割模式,其中 L 是整数。

[0858] 9. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块的宽度除以高度的比大于阈值 L 而禁用所述几何分割模式,其中 L 是整数。

[0859] 10. 根据解决方案1所述的方法,其中,所述规则规定:由于所述视频块的高度除以宽度的比大于阈值 L 而禁用所述几何分割模式,其中 L 是整数。

[0860] 11. 根据解决方案8至10中任一项所述的方法,其中, L 等于8。

[0861] 12. 根据解决方案5至10中任一项所述的方法,其中,在序列参数集、或图片参数集、或图片标头、或条带标头、或片组标头、或片标头的所述编解码表示中信令通知所述阈值 L 、 M 、 N 和 K 中的至少一个。

[0862] 13. 根据解决方案5至12中任一项所述的方法,其中,所述阈值 L 、 M 、 N 和 K 中的至少一个是与所述视频块相关联的块尺寸、或图片类型、或时域层索引的函数。

[0863] 14. 根据解决方案1至13中任一项所述的方法,其中,当确定对所述视频块禁用所述几何分割模式时,所述编解码表示省略包括所述几何分割模式的使用的指示。

[0864] 15. 根据解决方案1所述的方法,其中,所述视频块的所述特性包括所述视频块所属的所述视频的视频分量。

[0865] 16. 根据解决方案15所述的方法,其中,所述视频块包含具有多个颜色分量的样点,并且所述规则规定:对一个颜色分量禁用所述几何分割模式,而对另一个颜色分量启用所述几何分割模式。

[0866] 17. 根据解决方案1所述的方法,其中,所述规则规定:对尺寸为 $N \times M$ 或 $L \times N$ 的亮度块启用所述几何分割模式,其中 L 、 M 、 N 是正整数,而如果所述视频具有4:2:0格式,则对对应的色度块禁用所述几何分割模式。

[0867] 18. 根据解决方案1所述的方法,其中,如果确定对所述视频块禁用所述几何分割模式,则所述转换包括:使用所述视频块的一个预测分割的运动信息或预测模式导出所述视频块的预测块。

[0868] 19. 根据解决方案1所述的方法,其中,如果确定对所述视频块禁用所述几何分割

模式,则所述转换包括:使用从与所述视频块的两个预测分割相关联的运动信息的导出的运动信息来导出所述视频块的预测块。

[0869] 20. 根据解决方案19所述的方法,其中,使用具有两个不同方向上的运动矢量的所述两个预测分割的所述运动信息来生成所述视频块的双向预测运动矢量。

[0870] 21. 根据解决方案19所述的方法,其中,使用缩放操作,使用具有相同方向上的运动矢量的所述两个预测分割的所述运动信息来生成所述视频块的单向预测运动矢量,在所述缩放操作中,根据所述第二分割的参考图片缩放第一分割的第一运动矢量,并且使用所述缩放运动矢量和所述第二分割的第二运动矢量的平均。

[0871] 22. 根据解决方案16所述的方法,其中,使用缩放操作,使用具有两个方向上的运动矢量的所述两个预测分割的所述运动信息来生成所述视频块的双向预测运动矢量,在所述缩放操作中,根据所述第二分割的参考图片缩放第一分割的第一运动矢量,并且使用所述缩放运动矢量和所述第二分割的第二运动矢量的平均。

[0872] 23. 根据解决方案21和22中任一项所述的方法,其中,所述缩放操作取决于所述视频的色度格式。

[0873] 24. 根据解决方案1至23中任一项所述的方法,其中,所述比特流表示中的指示信令通知所述视频块对所述几何分割模式的使用。

[0874] 25. 根据解决方案24所述的方法,其中,所述指示位于序列参数集级、或图片参数集级、或视频参数集级、或图片标头、或条带标头、或片组标头、或编解码树单元级。

[0875] 26. 根据解决方案24至25中任一项所述的方法,其中,所述尺寸是与所述视频块相关联的图片类型或时域层索引的函数。

[0876] 在先前的章节中提供了以下解决方案的其它特征和变体(如项目22至27)。

[0877] 示例解决方案集F

[0878] 1. 一种视频处理方法(例如,图58所示的方法5800),包括:根据定义编解码表示的语法元素的格式的规则的规则,执行(5802)视频的视频块和所述视频的所述编解码表示之间的转换,其中,所述规则规定:是否已信令通知对第一编解码模式的使用的指示、或者是否要信令通知所述指示基于所述视频块对第二编解码模式的使用;其中,所述第二编解码模式是Merge模式;其中,所述Merge模式使得在没有整个所述视频块的运动矢量残差的情况下,能够从Merge候选列表中的Merge候选继承运动信息。

[0879] 2. 根据解决方案1所述的方法,其中,所述第一编解码模式包括几何分割模式,其中所述视频块被分割成多个预测分割。

[0880] 3. 根据解决方案2所述的方法,其中,所述多个预测分割中的至少一个是非矩形和非正方形预测分割。

[0881] 4. 根据解决方案1所述的方法,其中,所述第一编解码模式包括帧间帧内编解码模式,其中从帧内预测信号和帧间预测信号导出所述视频块的预测块。

[0882] 5. 根据解决方案1所述的方法,其中,所述第一编解码模式包括子块Merge编解码模式,其中,在所述子块Merge模式中,所述转换使用所述块内每个子块的导出的运动信息。

[0883] 6. 根据解决方案1至5中任一项所述的方法,其中,所述规则规定:由于对所述视频块启用所述第二编解码模式而将指示所述第一编解码模式的使用的所述指示的值设置为假。

[0884] 7. 根据解决方案1至5中任一项所述的方法,其中,所述规则规定:由于对所述视频块启用所述第二编解码模式而将所述第一编解码模式的使用的所述指示从所述编解码表示中排除。

[0885] 8. 根据解决方案1至7中任一项所述的方法,其中,所述转换包括:根据所述规则解析所述编解码表示。

[0886] 9. 一种视频处理方法(例如,图59所示的方法5900),包括:执行(5902)视频的视频块和所述视频的编解码表示之间的转换,其中,所述编解码表示的格式符合第一规则和/或第二规则,所述第一规则规定了信令通知多个编解码模式的语法元素是否包括在所述编解码表示中,所述第二规则规定了信令通知所述多个编解码模式的所述语法元素包括在所述编解码表示中的顺序;其中,多个编解码模式包括几何分割模式、帧间帧内(MHIntra)模式、子块Merge模式或运动矢量残差Merge(MMVD)模式中的两个或多个;其中,在所述帧间帧内编解码模式中,从帧内预测信号和帧间预测信号导出所述视频块的预测块;其中,在所述子块Merge模式中,所述转换使用为所述块内的每个子块导出的运动信息;其中,在MMVD模式中,使用Merge和运动矢量残差(MVD)编解码组合的模式;并且其中,Merge模式使得能够在没有整个所述视频块的MVD的情况下,从Merge候选列表中的Merge候选继承运动信息。

[0887] 10. 根据解决方案9所述的方法,其中,所述第二规则规定了语法元素的以下顺序:所述MMVD模式、所述子块Merge模式、所述几何分割模式、以及所述MHIntra模式。

[0888] 11. 根据解决方案9所述的方法,其中,所述第二规则规定了语法元素的以下顺序:所述MMVD模式、所述子块Merge模式、所述MHIntra模式、以及所述几何分割模式。

[0889] 12. 根据解决方案9所述的方法,其中,所述第二规则规定了语法元素的以下顺序:在所述MMVD模式之前信令通知所述子块Merge模式。

[0890] 13. 根据解决方案9所述的方法,其中,所述第二规则规定了语法元素的以下顺序:所述子块Merge模式、所述MMVD模式、所述MHIntra模式、所述几何分割模式。

[0891] 14. 根据解决方案9所述的方法,其中,基于所述视频的编解码条件在视频单元的所述编解码表示中改变所述顺序。

[0892] 15. 根据解决方案12所述的方法,其中,所述视频单元对应于块、编解码单元、预测单元、编解码树单元或最大编解码单元行、视频条带、图片或序列级。

[0893] 16. 根据解决方案12至15中任一项所述的方法,其中,所述编解码条件包括:低延迟检查标志的值、先前编解码的视频块的编解码信息、或所述视频块的维度。

[0894] 17. 一种视频处理方法(例如,图60所示的方法6000),包括:

[0895] 为视频的当前视频块与所述视频的编解码表示之间的转换,检查(6002)所述视频块的相邻块的相邻运动矢量精度信息是否可用于基于所述相邻块的位置来确定所述当前块的运动矢量精度;以及基于所述检查执行(6004)所述转换。

[0896] 18. 根据解决方案17所述的方法,其中,如果所述相邻块和所述当前视频块位于不同的最大编解码单元(LCU)行、或位于所述视频的不同区域,则认为所述相邻运动矢量精度信息不可用。

[0897] 19. 根据解决方案18所述的方法,其中,由于确定所述相邻运动矢量精度信息不可用而将所述当前块的所述运动矢量精度设置为默认值。

[0898] 20. 根据解决方案17所述的方法,其中,如果所述相邻块和所述当前视频块位于不

同的最大编解码单元 (LCU) 行, 则通过假设所述相邻块位于不同条带或不同片来确定所述当前视频块的所述运动矢量精度。

[0899] 21. 根据解决方案17至20中任一项所述的方法, 其中, 所述当前视频块和所述相邻块使用仿射编解码模式编解码。

[0900] 22. 一种视频处理方法 (例如, 图61所示的方法6100), 包括:

[0901] 基于相邻块相对于当前视频块的最大编解码单元 (LCU) 行的位置, 为视频的所述当前视频块和所述视频的编解码表示之间的转换确定 (6102) 用于所述编解码表示的上下文编解码模式; 以及基于所述确定执行 (6104) 所述转换。

[0902] 23. 根据解决方案22所述的方法, 其中, 所述相邻块由于位于不同的LCU行而被认为不可用, 并且其中, 所述上下文编解码模式使用单个上下文来编解码语法元素。

[0903] 24. 根据解决方案22所述的方法, 其中, 所述相邻块由于位于不同的LCU行而被认为不可用, 并且其中, 所述上下文编解码模式使用旁路编解码来编解码语法元素。

[0904] 25. 根据解决方案23至24中任一项所述的方法, 其中, 所述语法元素是以下中的一个: `alf_ctb_flag`、`qt_split_cu_flag`、`mtt_split_cu_flag`、`mtt_split_cu_flag`、`cu_skip_flag`、`amvr_mode`、`merge_subblock_flag`、`merge_triangle_flag`、`inter_affine_flag`、`cu_skip_flag`、`pred_mode_flag`、`pcm_flag`、`intra_luma_ref_idx`、`intra_luma_mpm_flag`、`intra_luma_mpm_idx`、`intra_luma_mpm_remainder`、`intra_chroma_pred_mode`、`merge_flag`、`inter_pred_idc`、`inter_affine_flag`、`cu_affine_type_flag`、`ref_idx_l0`、`mvp_l0_flag`、`ref_idx_l1`、`mvp_l1_flag`、`amvr_flag`、`amvr_4pel_flag`、`gbi_idx`、`cu_cbf`、`mmvd_flag`、或`mmvd_merge_flag`。

[0905] 在先前的章节中提供了以下解决方案的其它特征和变体 (例如, 项目1至4、17和21)。

[0906] 示例解决方案集G

[0907] 1. 一种视频处理方法 (例如, 图62所示的方法6200), 包括: 执行 (6202) 视频的当前视频块和所述视频的编解码表示之间的转换, 其中, 所述编解码表示符合规则, 所述规则规定了上下文建模限于来自用于解码所述当前视频块和/或满足位置标准的相邻块的信息。

[0908] 2. 根据解决方案1所述的方法, 其中, 所述相邻块是左下块或右上块。

[0909] 3. 根据解决方案1所述的方法, 其中, 所述规则规定: 不允许使用包含所述当前视频块的当前视频单元外部的相邻块。

[0910] 4. 根据解决方案1所述的方法, 其中, 所述规则规定: 将包含所述当前视频块的当前视频单元外部的相邻块用作不可用块。

[0911] 5. 根据解决方案1所述的方法, 其中, 所述规则规定: 将包含所述当前视频块的当前视频单元外部的相邻块用作具有默认值。

[0912] 6. 根据解决方案1所述的方法, 其中, 所述规则规定: 将包含所述当前视频块的当前视频单元外部的相邻块用作替换块。

[0913] 7. 根据解决方案3所述的方法, 其中, 所述当前视频单元是编解码树单元、或编解码树单元行、或视频区域、或视频数据处理单元、或条带、或片、或片组、或在图片边界处。

[0914] 8. 一种视频处理方法 (例如, 图63所示的方法6300), 包括: 执行 (6302) 视频的当前视频块和所述视频的编解码表示之间的转换, 其中, 所述编解码表示符合规则, 所述规则规

定了将来自相邻块的编解码信息用于表示所述当前视频块的编解码表示的一个或多个语法元素的上下文建模。

[0915] 9. 根据解决方案8所述的方法,其中,所述来自相邻块的编解码信息基于在所述比特流表示中表示所述相邻块的语法元素。

[0916] 10. 根据解决方案8或9中任一项所述的方法,其中,所述相邻块对应于用于从所述编解码表示重构所述当前视频块所访问的块。

[0917] 11. 根据解决方案8至10中任一项所述的方法,其中,所述相邻块对应于来自用于空域Merge候选构造的五个位置的恰好两个块。

[0918] 12. 根据解决方案8至10中任一项所述的方法,其中,所述相邻块对应于来自用于空域Merge候选构造的五个位置的恰好一个块。

[0919] 13. 根据解决方案8所述的方法,其中,所述规则规定:排除使用所述当前视频块的编解码树单元或编解码树单元行之外的相邻块。

[0920] 14. 根据解决方案8所述的方法,其中,所述规则规定:对于所述上下文建模,所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素被视为不可用。

[0921] 15. 根据解决方案8所述的方法,其中,所述规则规定:对于所述上下文建模,所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素被视为具有默认值。

[0922] 16. 根据解决方案8所述的方法,其中,所述规则规定:对于所述上下文建模,用外部相邻块来替换所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素。

[0923] 17. 根据解决方案8所述的方法,其中,所述规则规定:对于所述上下文建模,用所述编解码树单元或包括可用编解码信息的编解码树单元中的N个可用相邻块中的一个来替换所述当前视频块的编解码树单元或编解码树单元行之外的相邻块的语法元素,其中N是整数。

[0924] 18. 根据解决方案8至17中任一项所述的方法,其中,所述一个或多个语法元素包括:alf_ctb_flag、qt_split_cu_flag、mtt_split_cu_flag、cu_skip_flag、amvr_mode、inter_affine_flag、merge_triangle_flag and merge_subblock_flag、cu_skip_flag、pred_mode_flag、pcm_flag、intra_luma_ref_idx、intra_luma_mpm_flag、intra_luma_mpm_idx、intra_luma_mpm_remainder、intra_chroma_pred_mode、merge_flag、inter_pred_idc、inter_affine_flag、cu_affine_type_flag、ref_idx_l0、mvp_l0_flag、ref_idx_l1、mvp_l1_flag、amvr_flag、amvr_4pel_flag、gbi_idx、cu_cbf、mmvd_flag、mmvd_merge_flag、或pred_mode_flag。

[0925] 19. 根据解决方案18所述的方法,其中,语法元素的上下文ctx导出如下:

[0926] $ctx = A1$ 是帧内? $1:0$; $ctx += B1$ 是帧内? $1:0$;

[0927] 其中,A1是左下相邻块,并且B1是右上相邻块。

[0928] 20. 根据解决方案18所述的方法,其中pred_mode_flag的上下文ctx导出如下:

[0929] $ctx = (A1$ 是帧内) $|| (B1$ 是帧内) $?1:0$

[0930] 其中,A1是左下相邻块,并且B1是右上相邻块。

[0931] 21. 根据解决方案18所述的方法,其中,语法元素的上下文ctx导出如下:

- [0932] $ctx=A1$ 是帧内或帧间帧内? $1:0$; $ctx+=B1$ 是帧内或帧间帧内? $1:0$;
- [0933] 其中, $A1$ 是左下相邻块, 并且 $B1$ 是右上相邻块。
- [0934] 22. 根据解决方案18所述的方法, 其中, 语法元素的上下文 ctx 导出如下:
- [0935] $ctx=(A1\text{是帧内或帧间帧内})\|\|(B1\text{是帧内或帧间帧内})?1:0$;
- [0936] 其中, $A1$ 是左下相邻块, 并且 $B1$ 是右上相邻块。
- [0937] 23. 根据解决方案18所述的方法, 其中, 通过排除使用用于空域Merge候选构造的五个位置处的相邻块来导出语法元素的上下文 ctx 。
- [0938] 24. 根据解决方案23所述的方法, 其中, 所述语法元素对应于 $pred_mode_flag$ 。
- [0939] 25. 根据解决方案18至24中任一项所述的方法, 其中, 通过将不可用的相邻块视为帧间编解码块而导出所述上下文。
- [0940] 26. 根据解决方案18至24中任一项所述的方法, 其中, 通过将不可用的相邻块视为帧内编解码块而导出所述上下文。
- [0941] 27. 根据解决方案18至24中任一项所述的方法, 其中, 通过将不可用的相邻块视为帧间帧内编解码块而导出所述上下文。
- [0942] 28. 根据解决方案18至24中任一项所述的方法, 其中, 通过将位于与所述当前视频块不同的编解码树单元行中的相邻块视为帧间编解码块而导出所述上下文。
- [0943] 29. 根据解决方案18至24中任一项所述的方法, 其中, 通过将位于与所述当前视频块不同的编解码树单元行中的相邻块视为帧内编解码块而导出所述上下文。
- [0944] 30. 根据解决方案18至24中任一项所述的方法, 其中, 通过将位于与所述当前视频块不同的编解码树单元行中的相邻块视为帧间帧内编解码块而导出所述上下文。
- [0945] 参考以上列出的解决方案集, 在一些实施例中, 几何分割模式(在本文中也称为几何体分割模式)可以包括将视频块分割成多个分割, 其中至少一个分割是非正方形和非矩形分割。如本文中进一步公开的, 这些分割称为预测分割可用于形成在视频块的编解码或解码期间使用的预测块。
- [0946] 参考以上列出的解决方案集, 在一些实施例中, 转换可包括编解码视频从视频像素值或其他表示(例如, 转码)生成编解码表示。
- [0947] 参考以上列出的解决方案集, 在一些实施例中, 转换可以包括视频解码操作, 其中, 通过解码逻辑对编解码表示或比特流进行解析以生成或重构视频样点。在这方面, 转换包括解析处理, 其中根据本文中描述的各种规则解析语法元素。
- [0948] 在一些实施例中, 所公开的技术由视频解码器装置实现(例如, 如图31所示)。在一些实施例中, 视频编解码器可以实现本文中描述的各种技术。在一些实施例中, 所公开的解决方案可以被实施为存储用于使用处理器实现以上列出的解决方案的代码的计算机程序产品。
- [0949] 所公开技术的一些实施例包括: 做出启用视频处理工具或模式的决策或确定。在一个示例中, 当视频处理工具或模式被启用时, 编解码器将在视频块的处理中使用或实现该工具或模式, 但不一定基于该工具或模式的使用来修改产生的比特流。也就是说, 当基于决策或确定启用视频处理工具或模式时, 从视频块到视频的比特流表示的转换将使用该视频处理工具或模式。在另一示例中, 当视频处理工具或模式被启用时, 解码器将在知晓已经基于视频处理工具或模式修改了比特流的情况下处理比特流。也就是说, 将使用基于决策

或确定而启用的视频处理工具或模式来执行从视频的比特流表示到视频块的转换。

[0950] 所公开技术的一些实施例包括：做出禁用视频处理工具或模式的决策或确定。在一个示例中，当视频处理工具或模式被禁用时，编解码器将不在将视频块转换为视频的比特流表示的转换中使用该工具或模式。在另一示例中，当视频处理工具或模式被禁用时，解码器将在知晓未使用基于所述决策或确定而启用的视频处理工具或模式修改比特流的情况下来处理比特流。

[0951] 图64是可以实现所公开的技术的示例视频处理系统的框图。

[0952] 图64是示出示例视频处理系统1900的框图，其中可以实施本文公开的各种技术。各种实现可能包括系统1900的部分或全部分量。系统1900可包括用于接收视频内容的输入1902。视频内容可以原始或未压缩格式接收，例如8位或10位多分量像素值，或者可以压缩或编解码格式接收。输入1902可以表示网络接口、外围总线接口或存储接口。网络接口的示例包括诸如以太网、无源光网络(PON)等的有线接口，以及诸如Wi-Fi或蜂窝接口的无线接口。

[0953] 系统1900可包括编解码分量1904，其可实现本文中所描述的各种编解码或编解码方法。编解码分量1904可以降低从输入1902到编解码分量1904的输出的视频的平均比特率，以产生视频的编解码表示。因此，编解码技术有时被称为视频压缩或视频转码技术。编解码分量1904的输出可以被存储，也可以通过连接的通信进行传输，如分量1906所示。输入1902处接收的视频的存储或通信比特流(或编解码)表示可由分量1908用于生成像素值或发送到显示接口1910的可显示视频。从比特流表示生成用户可观看视频的处理有时称为视频解压缩。此外，尽管某些视频处理操作被称为“编解码”操作或工具，但应当理解的是，编解码工具或操作被用于编解码器处，并且逆向编解码结果的对应的解码工具或操作将由解码器执行。

[0954] 外围总线接口或显示接口的示例可以包括通用串行总线(USB)或高清晰度多媒体接口(HDMI)或显示端口等。存储接口的示例包括SATA(串行高级技术附件)、PCI、IDE接口等。本文中所述的技术可实施在各种电子设备中，例如移动电话、笔记本电脑、智能手机或其他能够执行数字数据处理和/或视频显示的设备。

[0955] 应当理解的是，当被压缩的编解码单元具有明显不同于传统的方形块或半方形的矩形块的形状时，可以将所公开的技术实施在视频编解码器或解码器中，以提高压缩效率。例如，使用长或高编解码单元(例如，4x32或32x4尺寸的单元)的新编解码工具可以受益于所公开的技术。

[0956] 还应当理解的是，规定了用于视频处理的各种规则，其中一起使用或排除使用子块编解码模式。

[0957] 本文中描述的所公开的和和其他解决方案、示例、实施例、模块和功能操作可以在数字电子电路或计算机软件、固件或硬件中实现，包括本文中公开的结构及其结构等效物、或者其一个或多个的组合。公开的实施例和其他实施例可以实现为一个或多个计算机程序产品，即一个或多个编解码在计算机可读介质上的计算机程序指令的模块，以供数据处理装置执行或控制数据处理装置的操作。计算机可读介质可以是机器可读存储设备、机器可读存储基板、存储设备、影响机器可读传播信号的物质组成或其中一个或多个的组合。术语“数据处理装置”包括用于处理数据的所有装置、设备和机器，包括例如可编程处理器、计算

机或多处理器或计算机组。除硬件外,该装置还可以包括为计算机程序创建执行环境的代码,例如,构成处理器固件的代码、协议栈、数据库管理系统、操作系统或其中一个或多个的组合。传播信号是人为产生的信号,例如机器产生的电信号、光学信号或电磁信号,生成这些信号以对信息进行编解码,以便传输到适当的接收装置。

[0958] 计算机程序(也称为程序、软件、软件应用、脚本或代码)可以用任何形式的编程语言(包括编译语言或解释语言)编写,并且可以以任何形式部署,包括作为独立程序或作为模块、组件、子程序或其他适合在计算环境中使用的单元。计算机程序不一定与文件系统中的文件对应。程序可以存储在保存其他程序或数据的文件的部分中(例如,存储在标记语言文档中的一个或多个脚本)、专用于该程序的单个文件中、或多个协调文件(例如,存储一个或多个模块、子程序或部分代码的文件)中。计算机程序可以部署在一台或多台计算机上来执行,这些计算机位于一个站点上或分布在多个站点上,并通过通信网络互连。

[0959] 本文中描述的处理和逻辑流可以通过一个或多个可编程处理器执行,该处理器执行一个或多个计算机程序,通过在输入数据上操作并生成输出来执行功能。处理和逻辑流也可以通过特殊用途的逻辑电路来执行,并且装置也可以实现为特殊用途的逻辑电路,例如,FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0960] 例如,适于执行计算机程序的处理器包括通用和专用微处理器,以及任何类型数字计算机的任何一个或多个。通常,处理器将从只读存储器或随机存取存储器或两者接收指令和数据。计算机的基本元件是执行指令的处理器和存储指令和数据的一个或多个存储设备。通常,计算机还将包括一个或多个用于存储数据的大容量存储设备,例如,磁盘、磁光盘或光盘,或通过操作耦合到一个或多个大容量存储设备来从其接收数据或将数据传输到一个或多个大容量存储设备,或两者兼有。然而,计算机不一定具有这样的设备。适用于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储设备,包括例如半导体存储设备,例如EPROM、EEPROM和闪存设备;磁盘,例如内部硬盘或可移动磁盘;磁光磁盘;以及CDROM和DVD-ROM光盘。处理器和存储器可以由专用逻辑电路来补充,或合并到专用逻辑电路中。

[0961] 虽然本专利文件包含许多细节,但不应将其解释为对任何主题或权利要求范围的限制,而应解释为对特定技术的特定实施例的特征的描述。本专利文件在单独实施例的上下文描述的一些特征也可以在单个实施例中组合实施。相反,在单个实施例的上下文中描述的各种功能也可以在多个实施例中单独实施,或在任何合适的子组合中实施。此外,尽管上述特征可以描述为在一些组合中起作用,甚至最初要求是这样,但在一些情况下,可以从组合中移除权利要求组合中的一个或多个特征,并且权利要求的组合可以指向子组合或子组合的变体。

[0962] 同样,尽管图纸中以特定顺序描述了操作,但这不应理解为要获得想要的结果必须按照所示的特定顺序或顺序执行此类操作,或执行所有说明的操作。此外,本专利文件所述实施例中各种系统组件的分离不应理解为在所有实施例中都需要这样的分离。

[0963] 仅描述了一些实现和示例,其他实现、增强和变体可以基于本专利文件中描述和说明的内容做出。

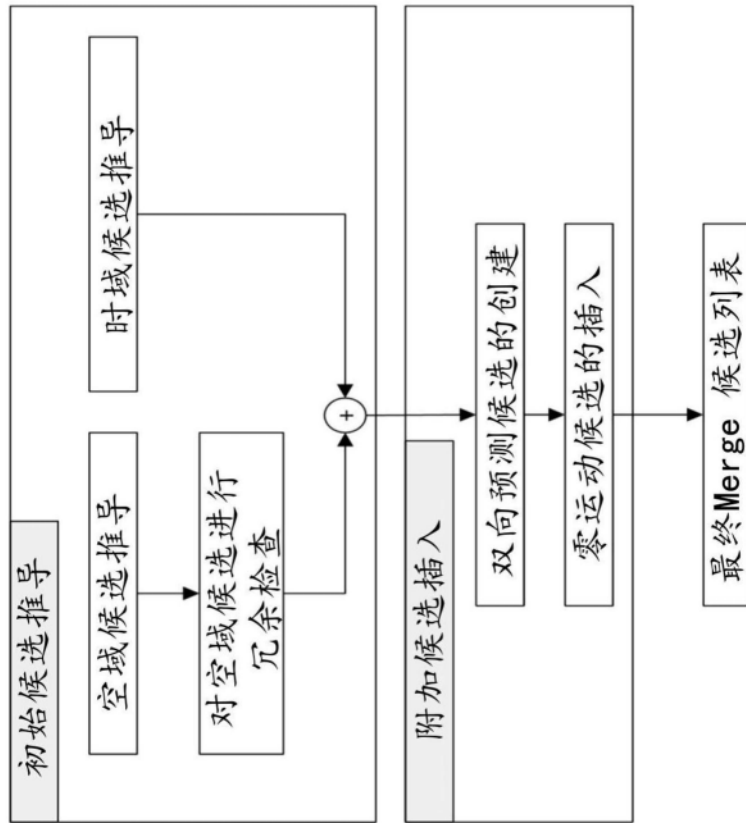


图1

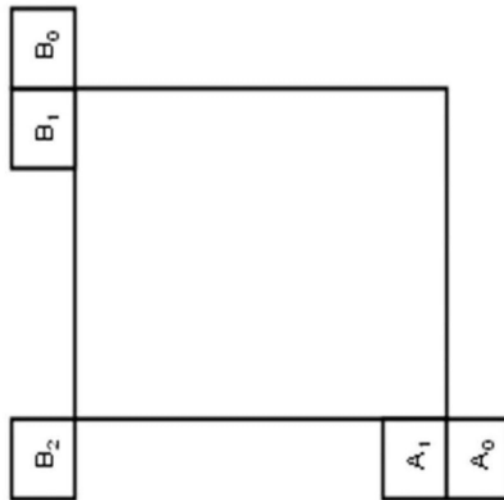


图2

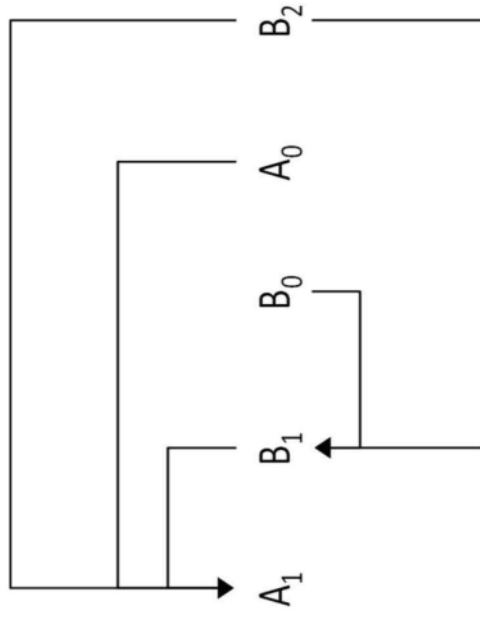


图3

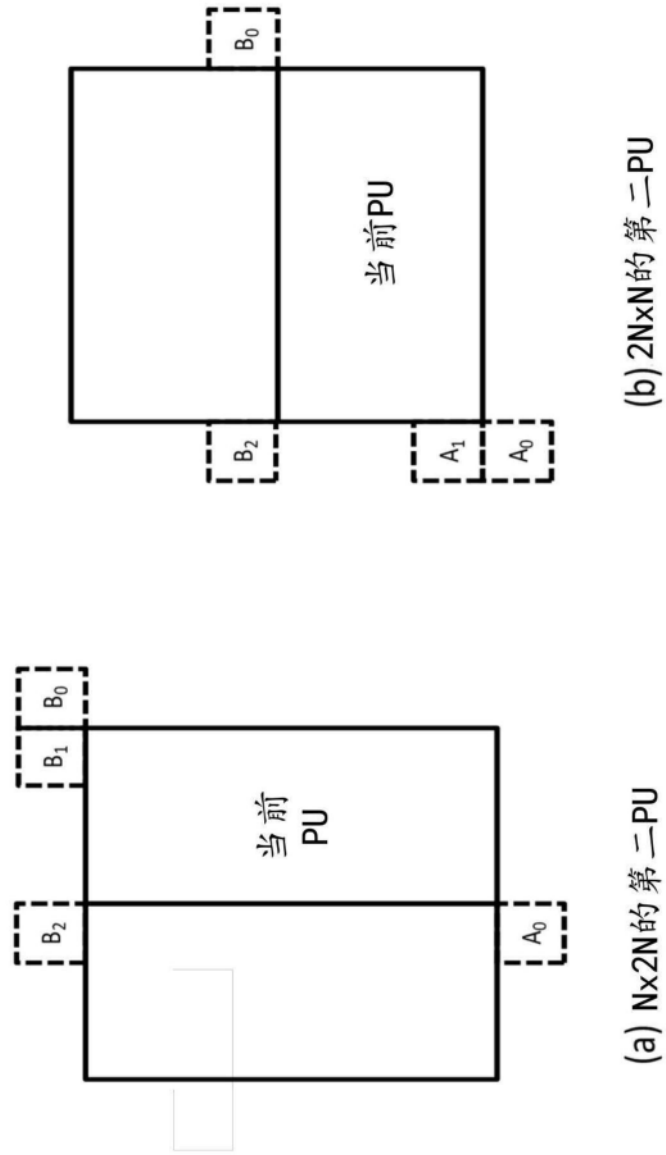


图4

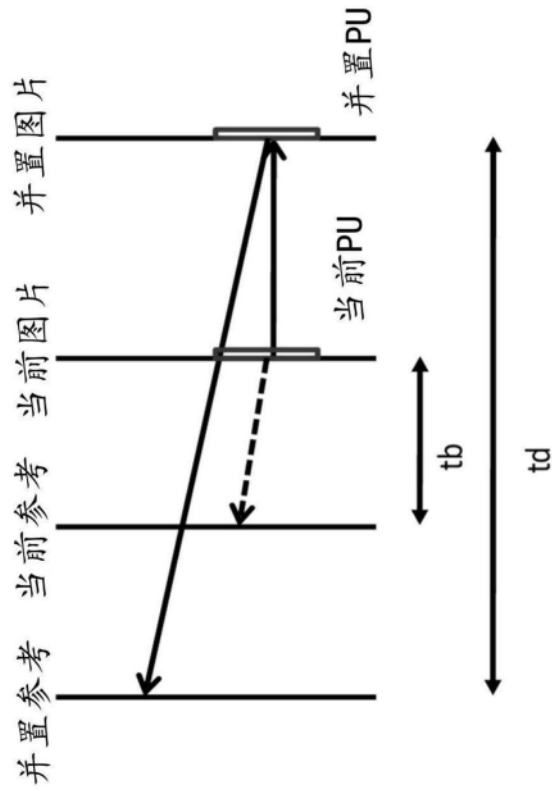


图5

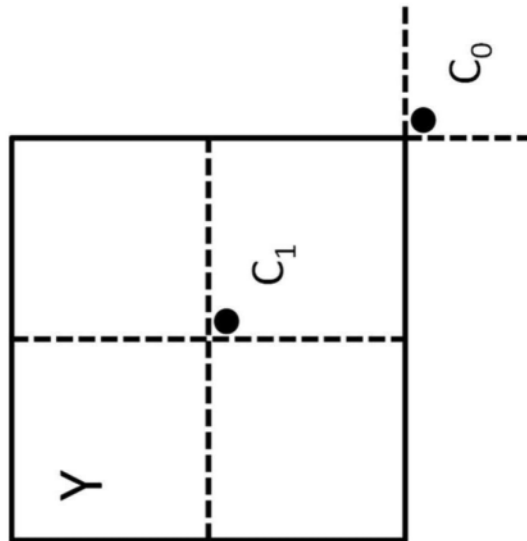


图6

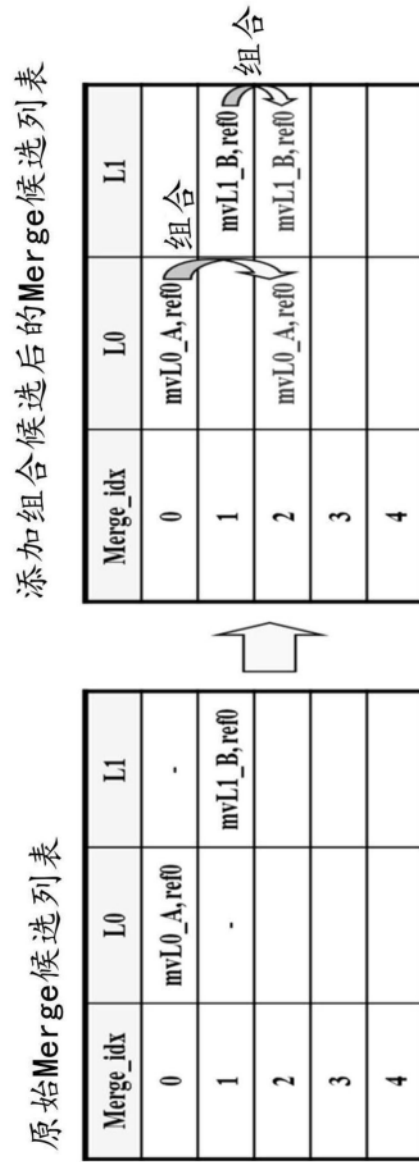


图7

对于参考索引作为输入的每个参考图片列表



图8

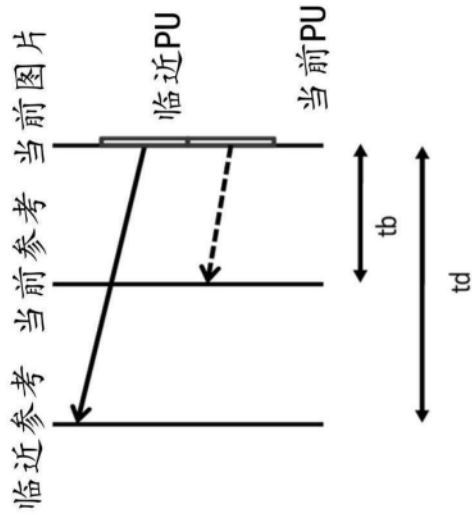


图9

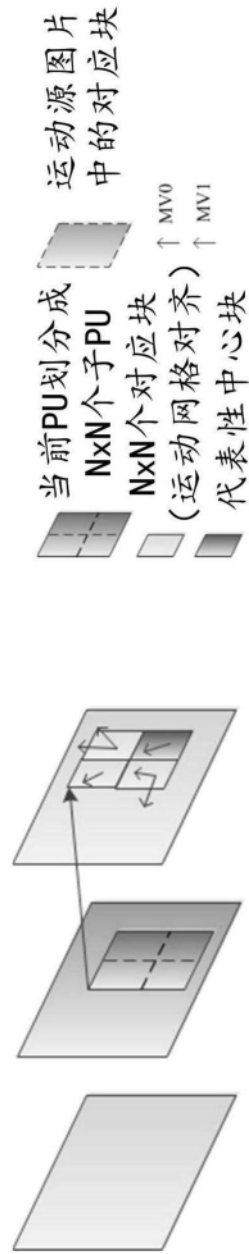


图10

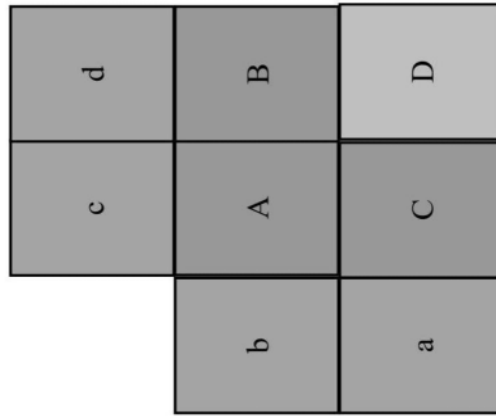


图11

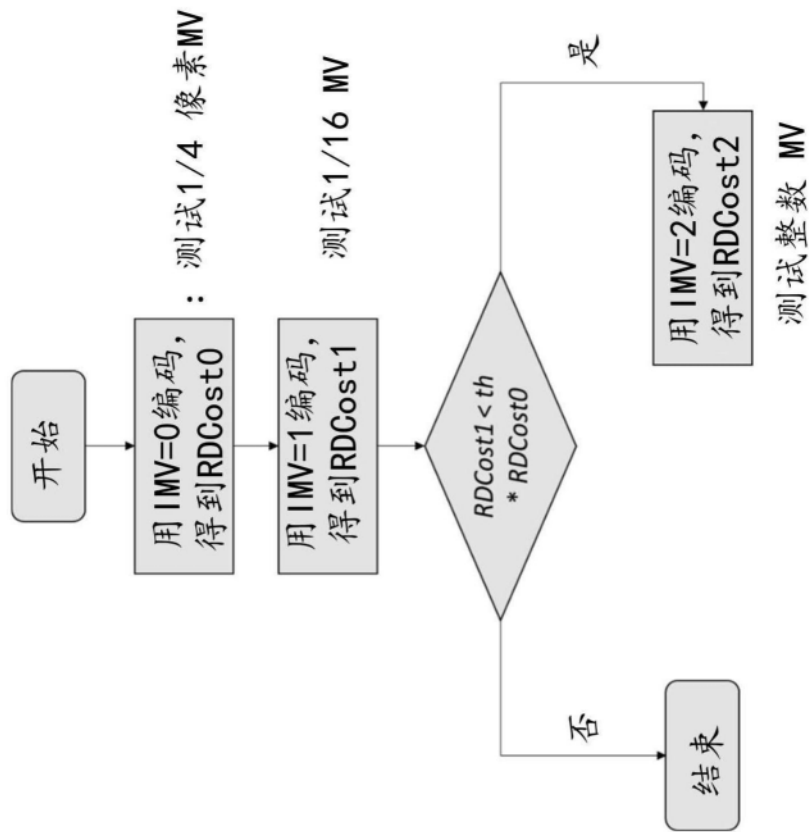


图12

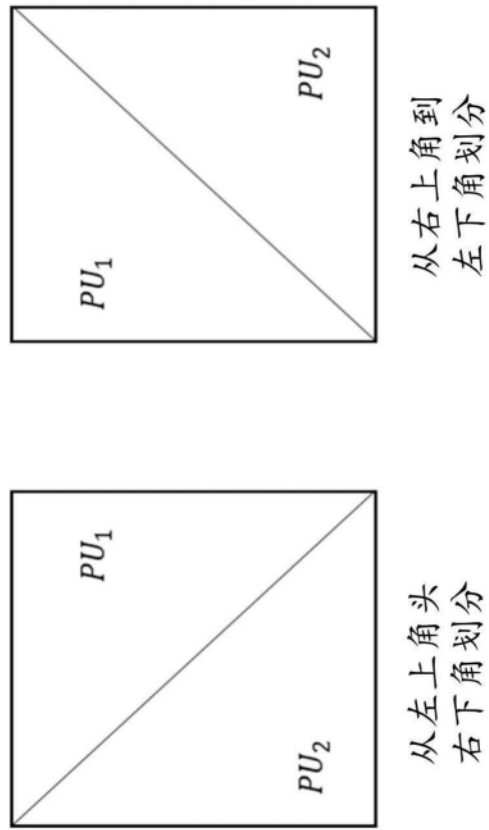


图13

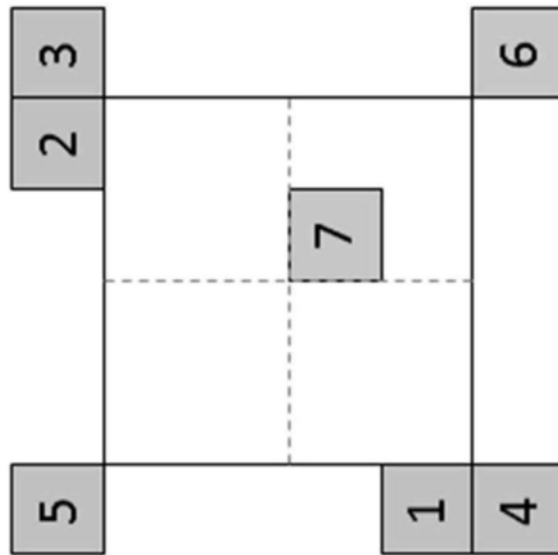


图14

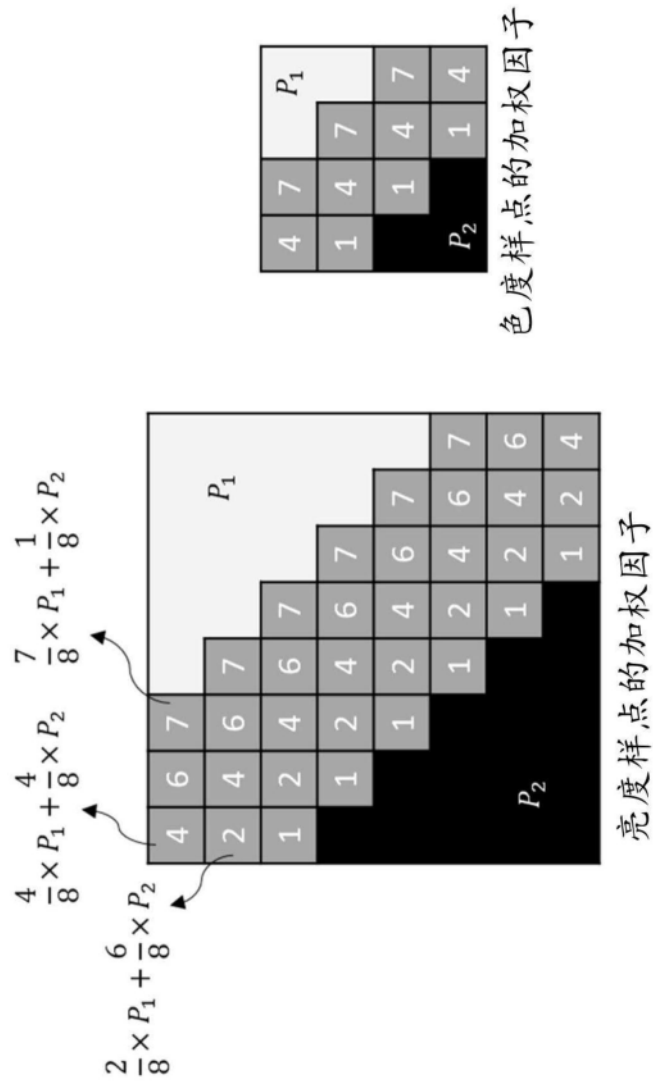


图15

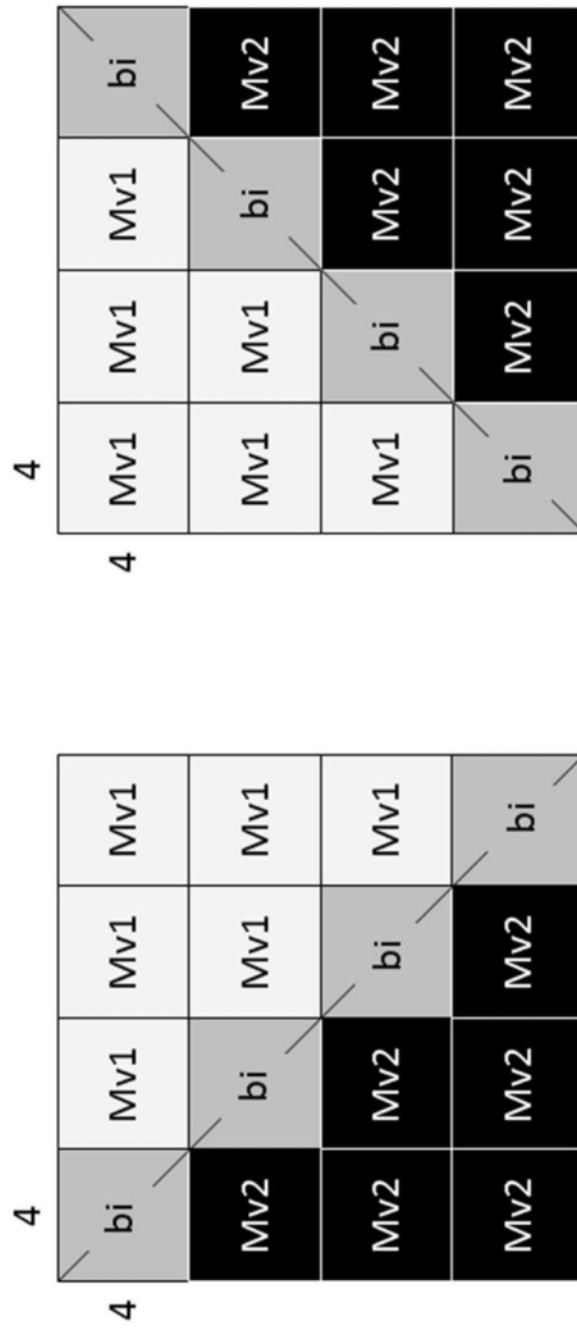


图16

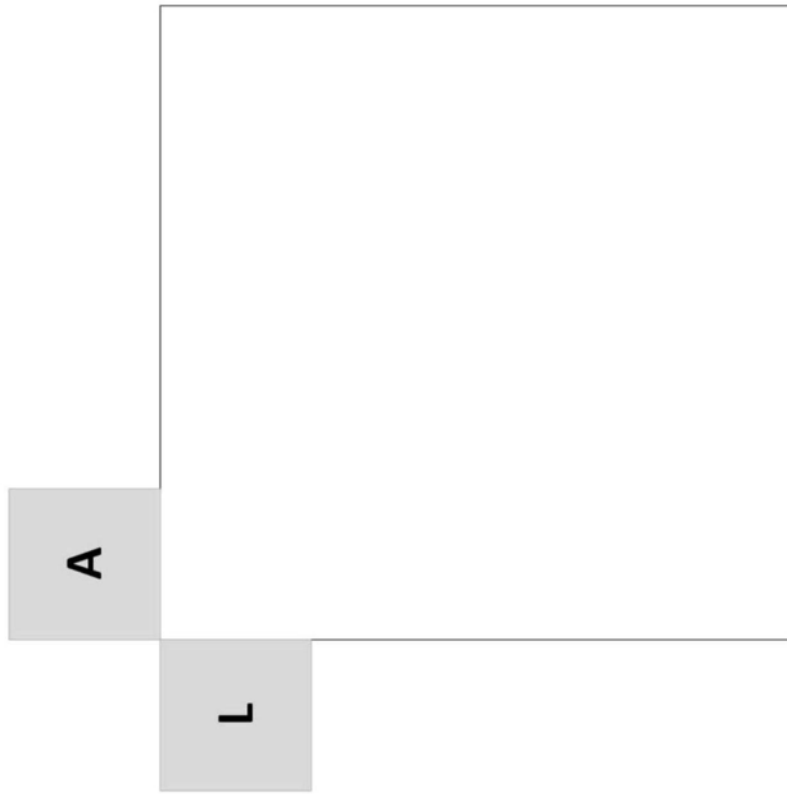


图17

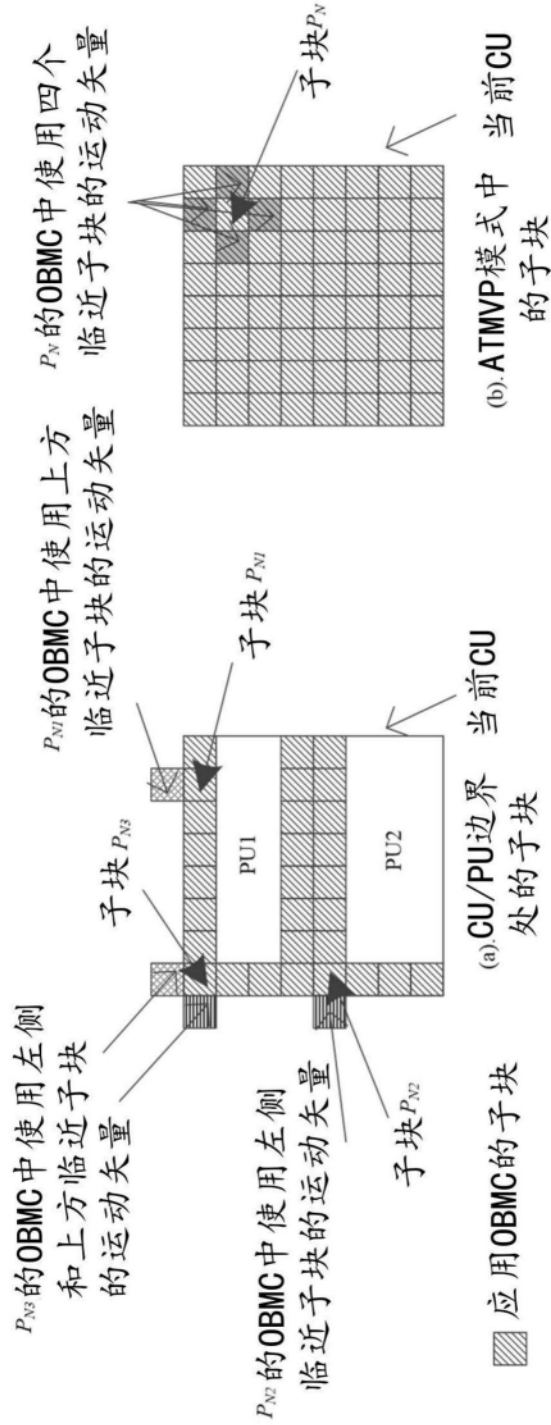


图18

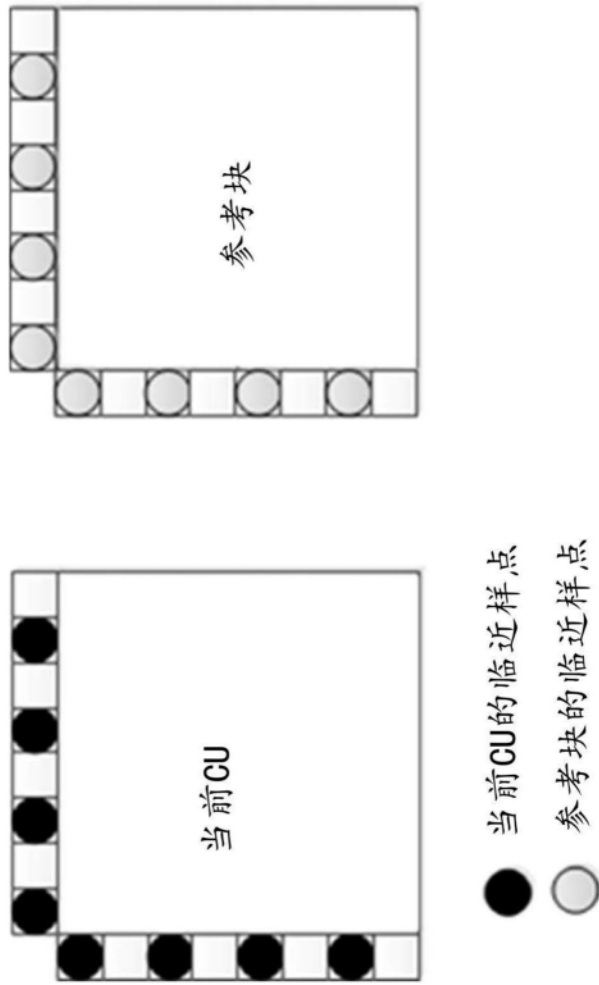


图19

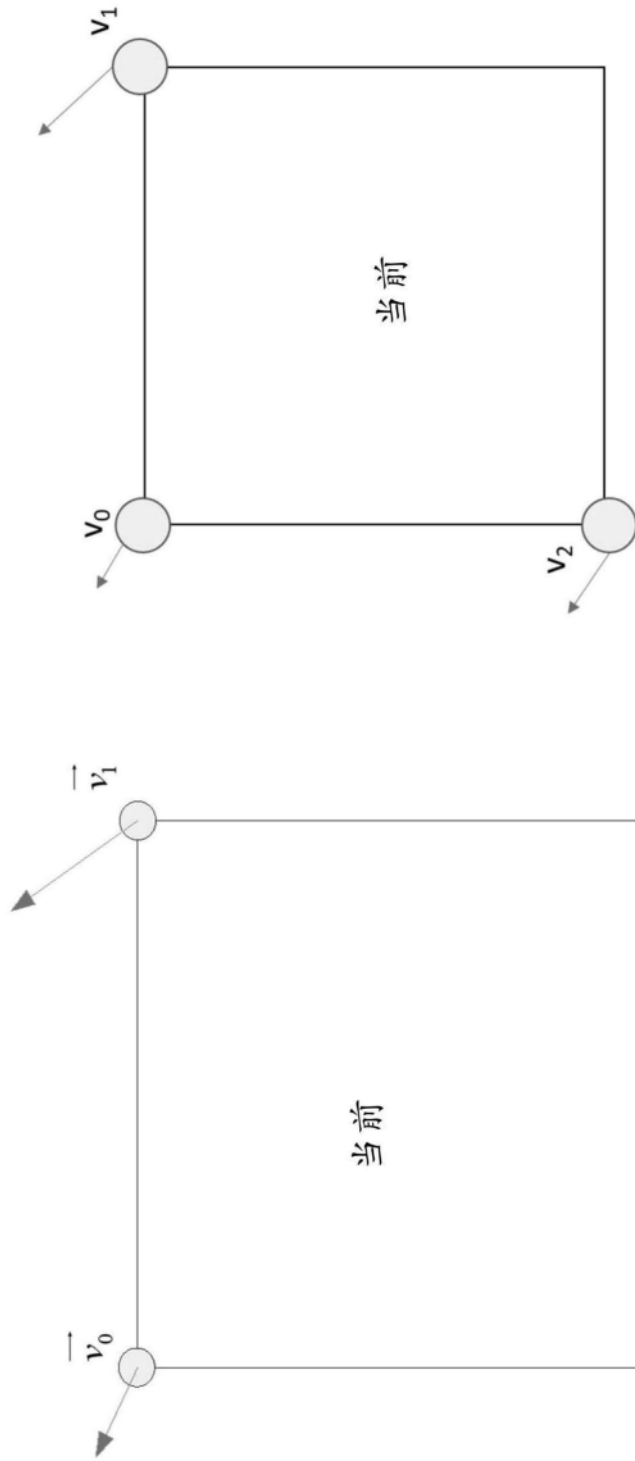


图20

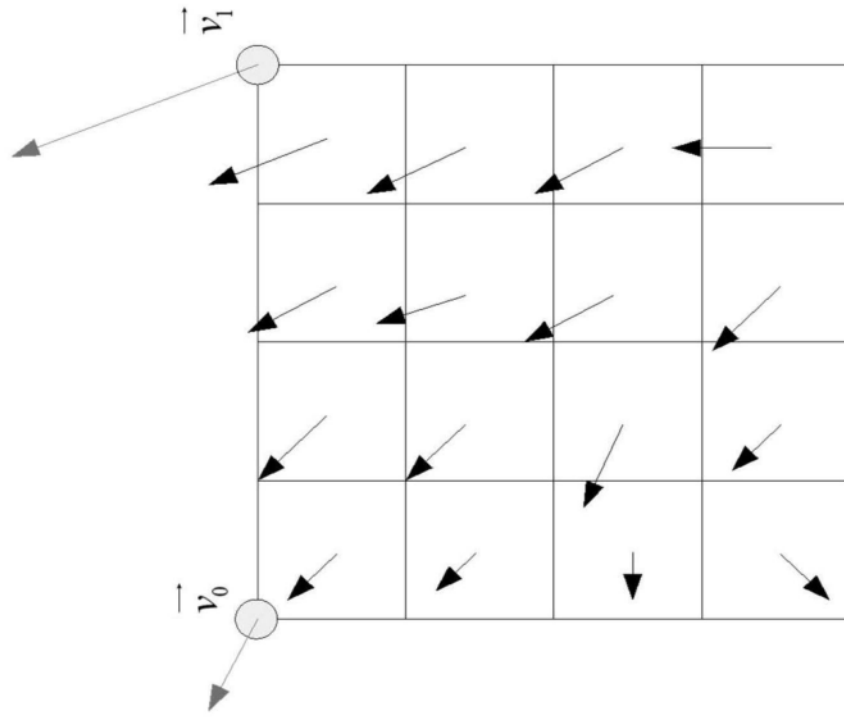


图21

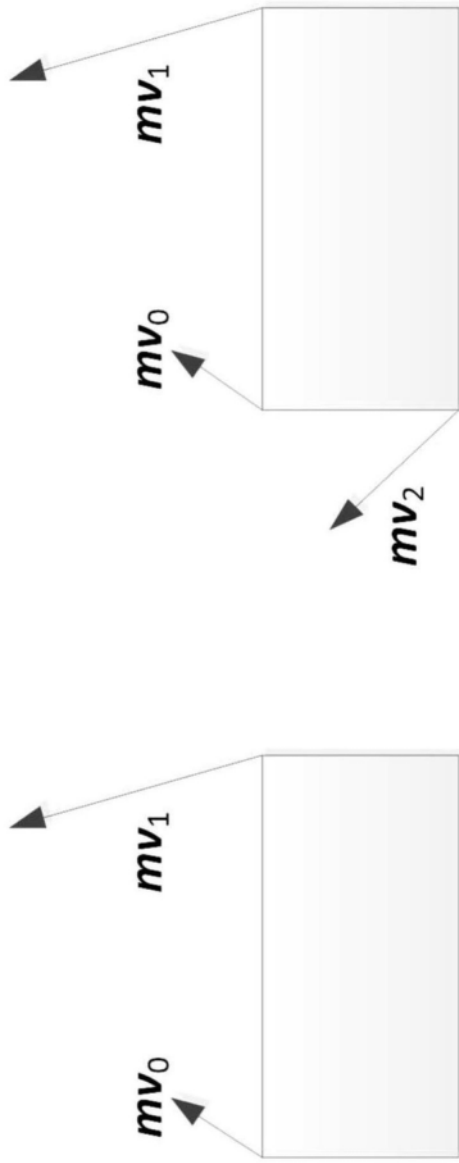


图22



图23

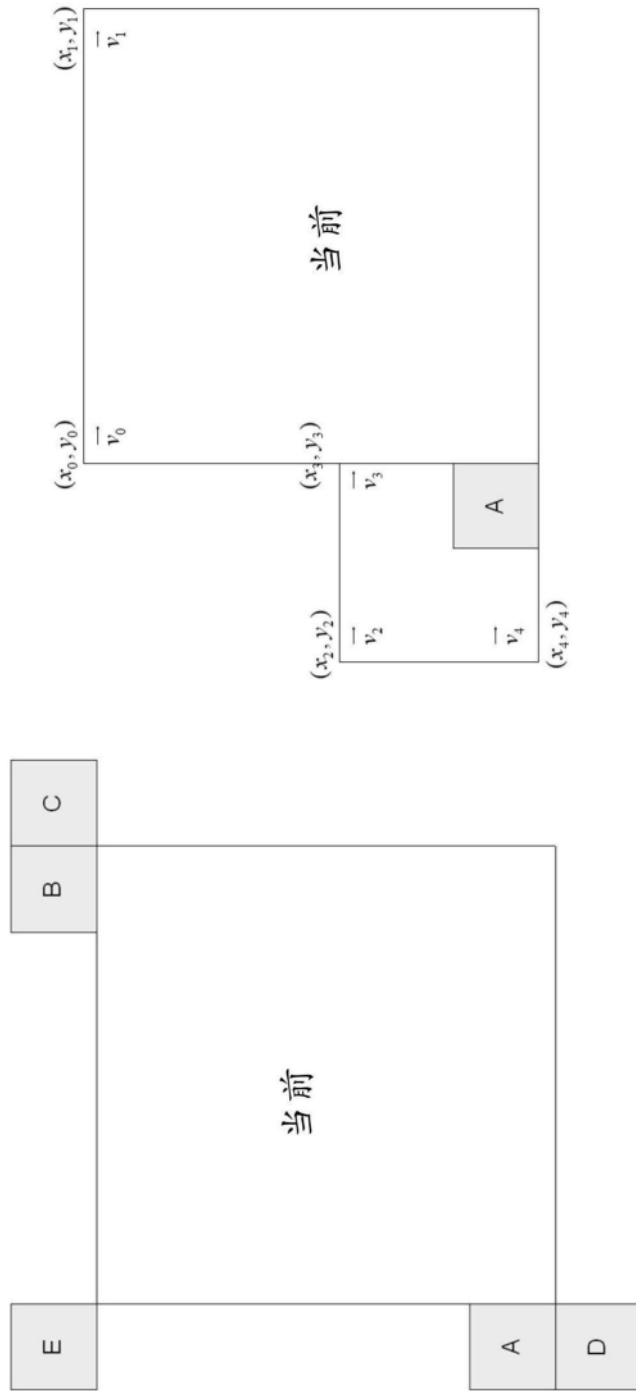


图24

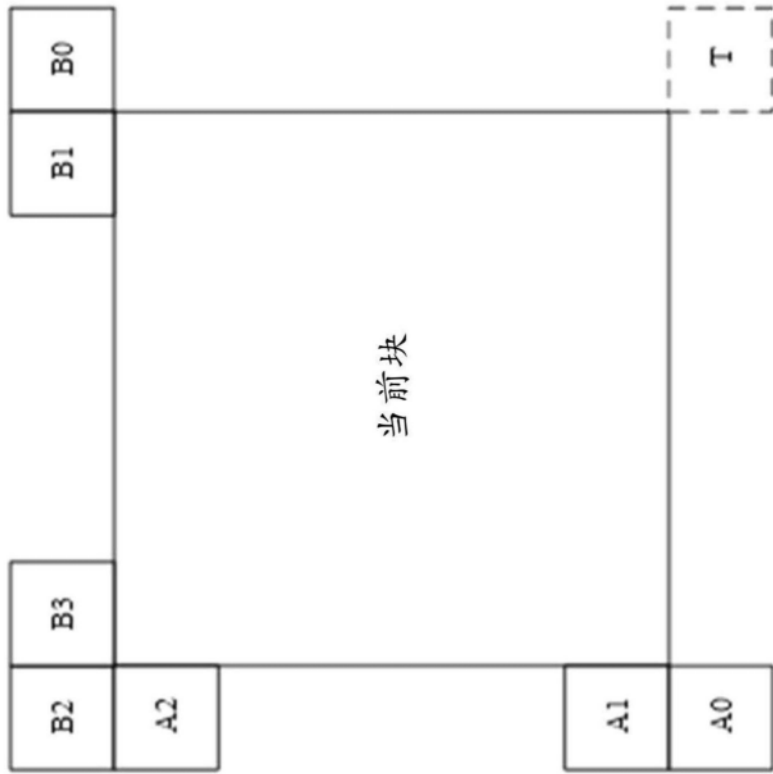


图25

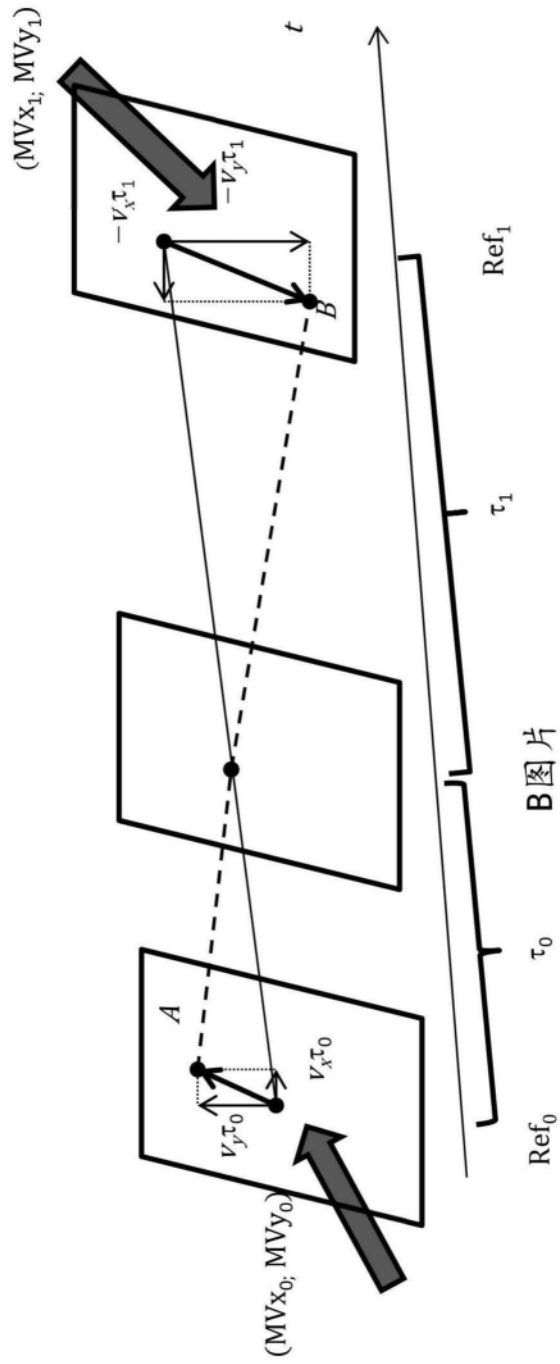


图26

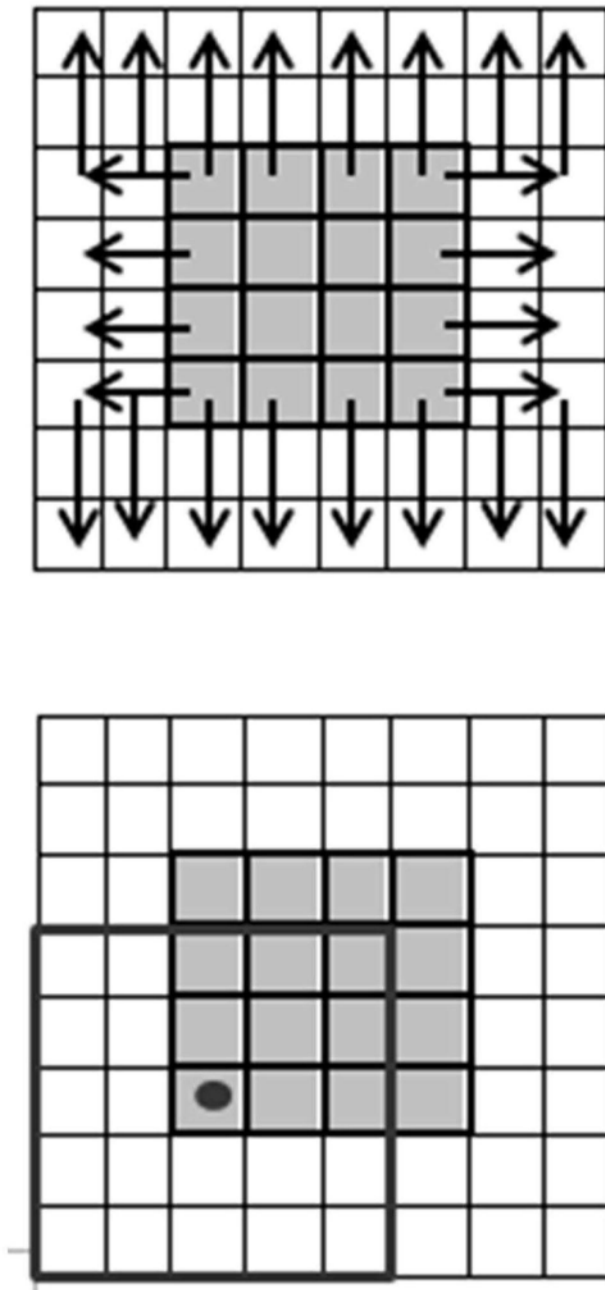


图27

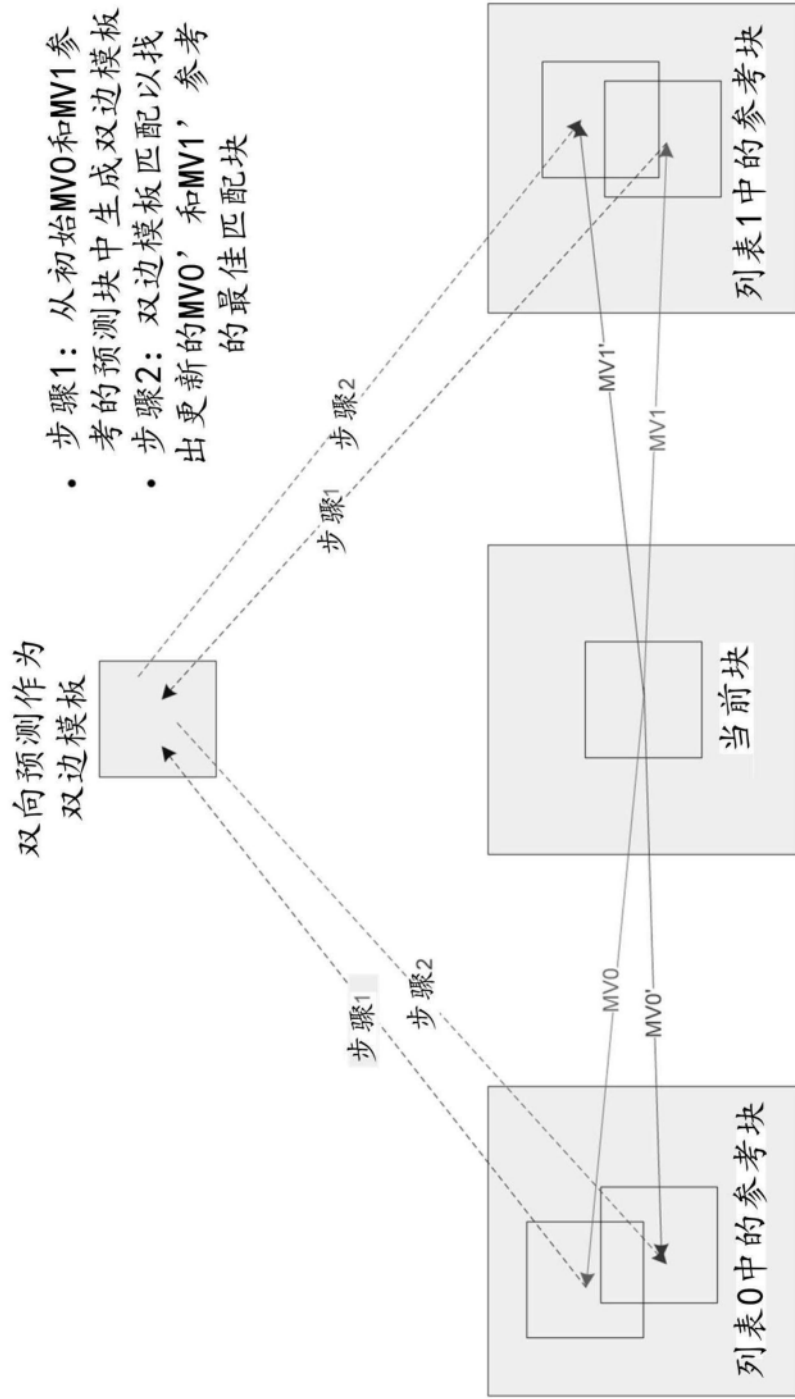


图28

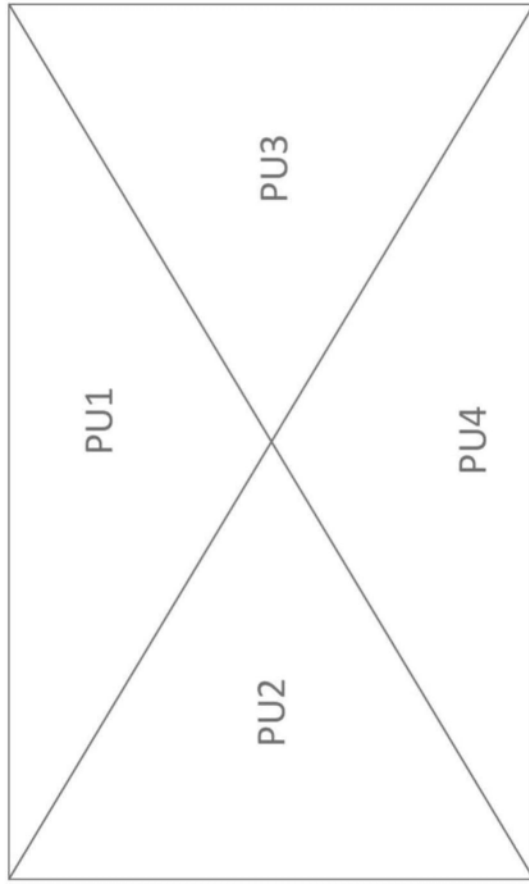


图29

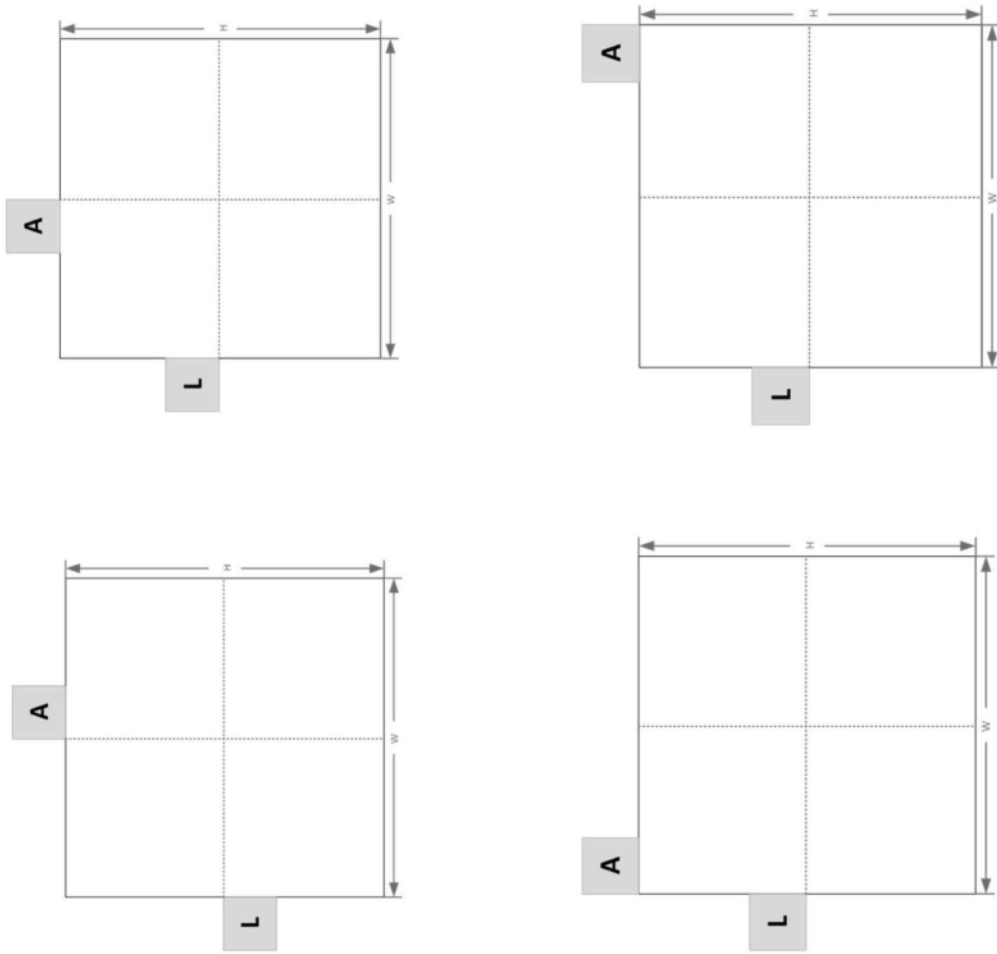


图30

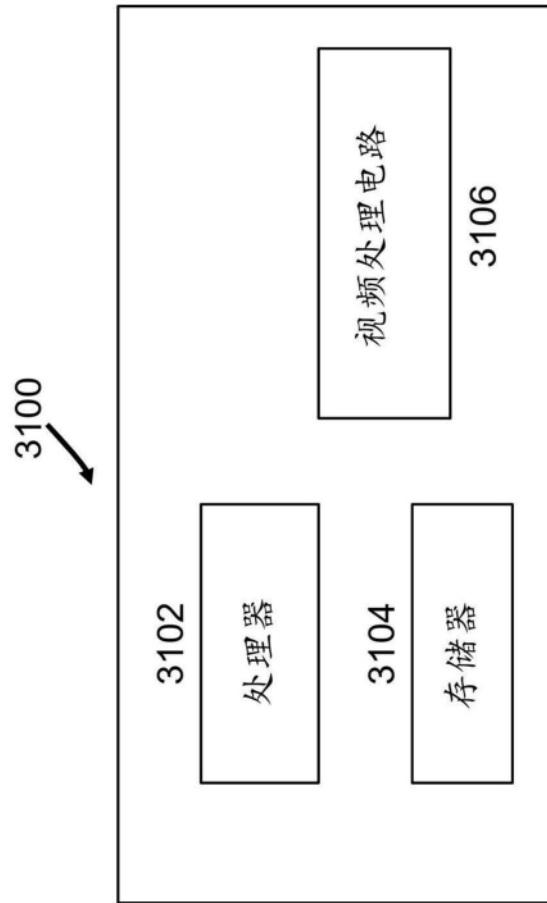


图31

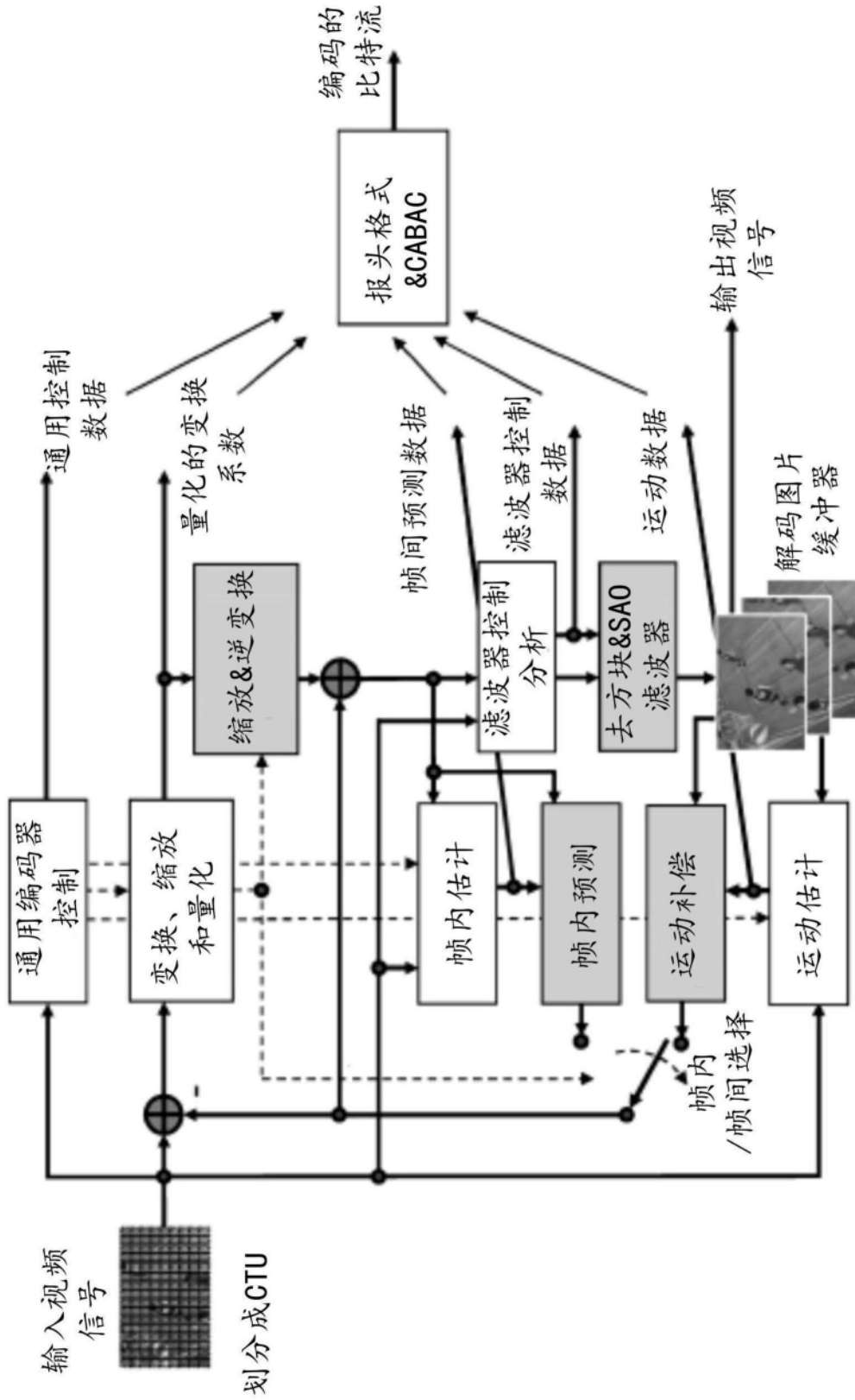


图32

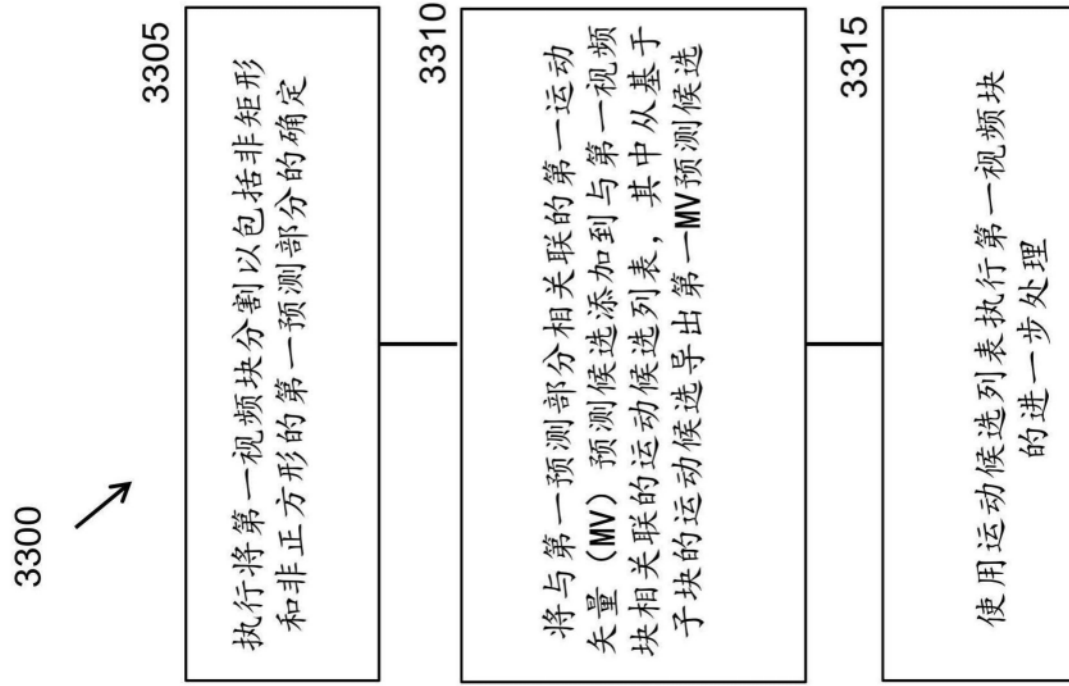


图33

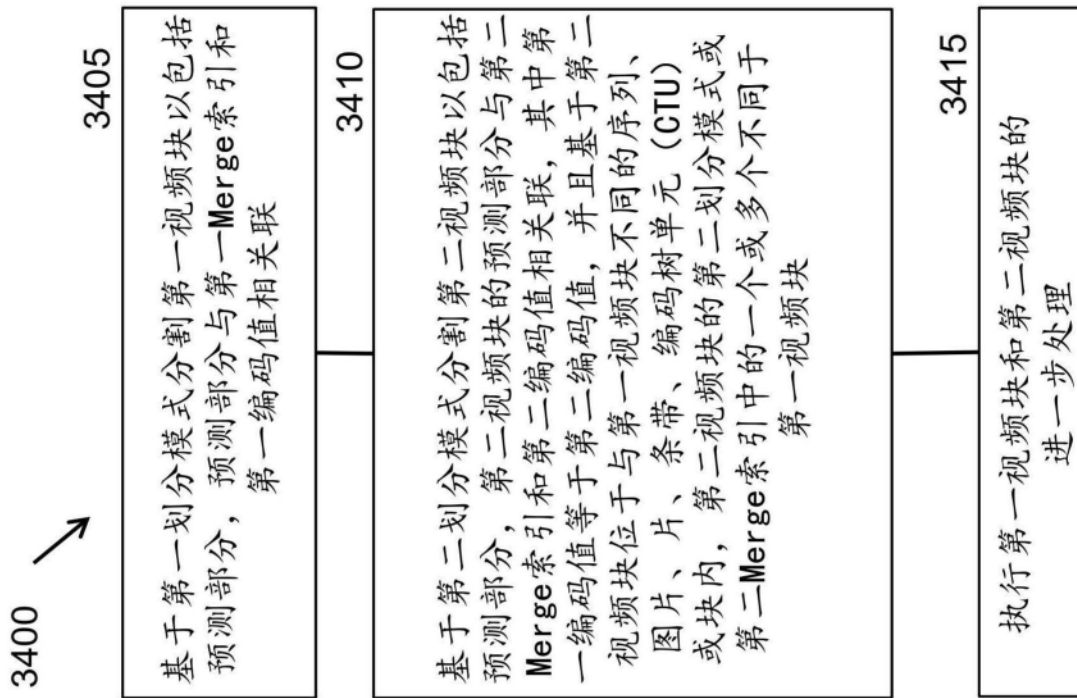


图34

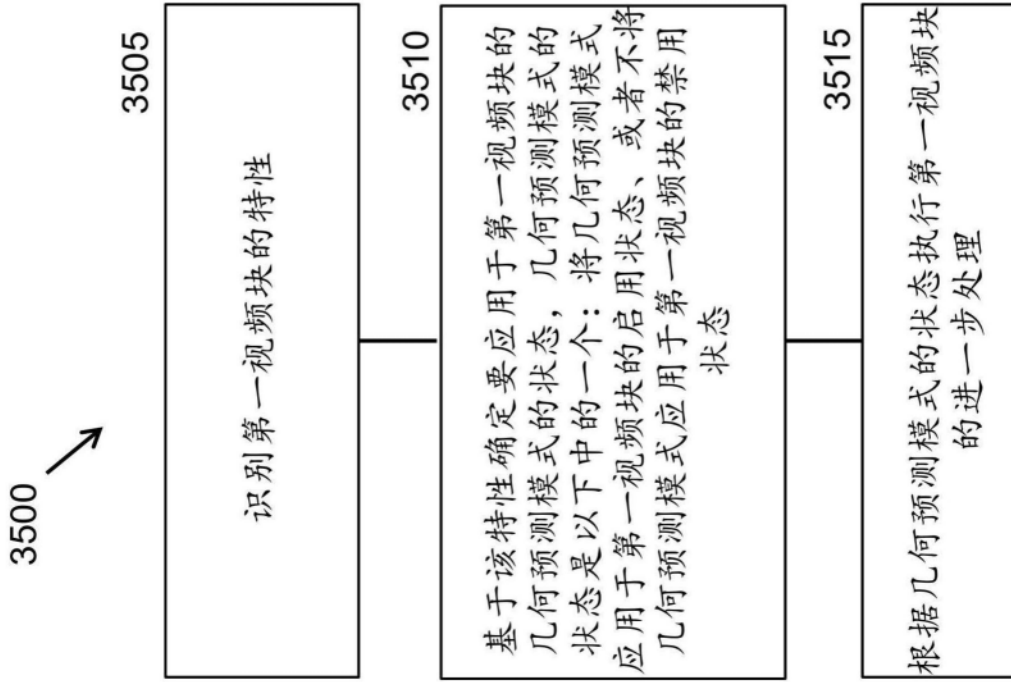


图35

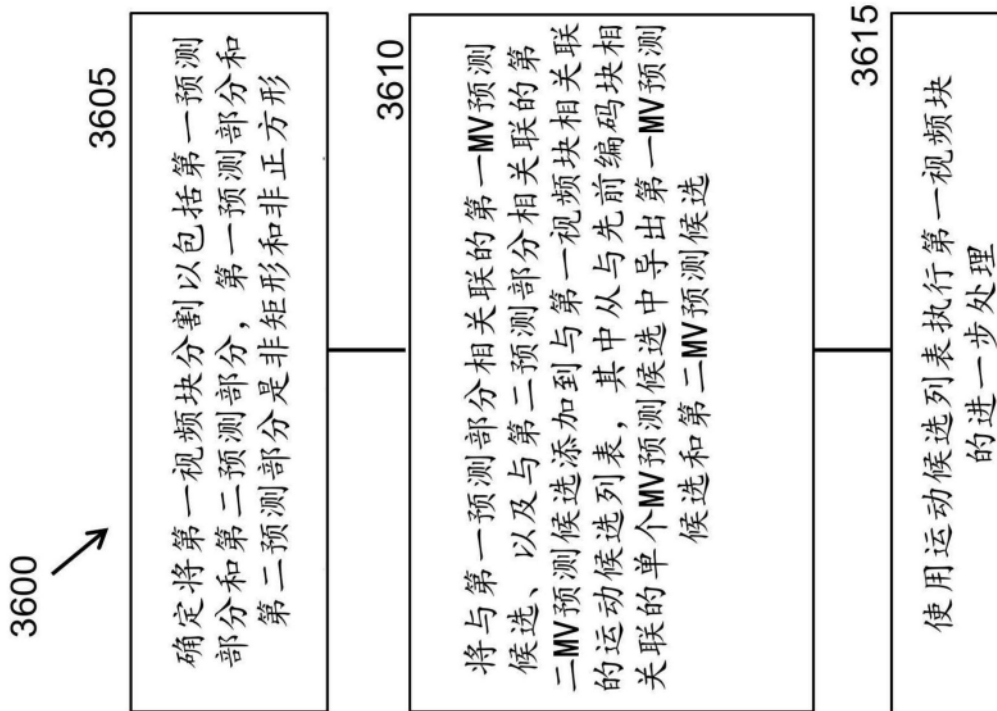


图36

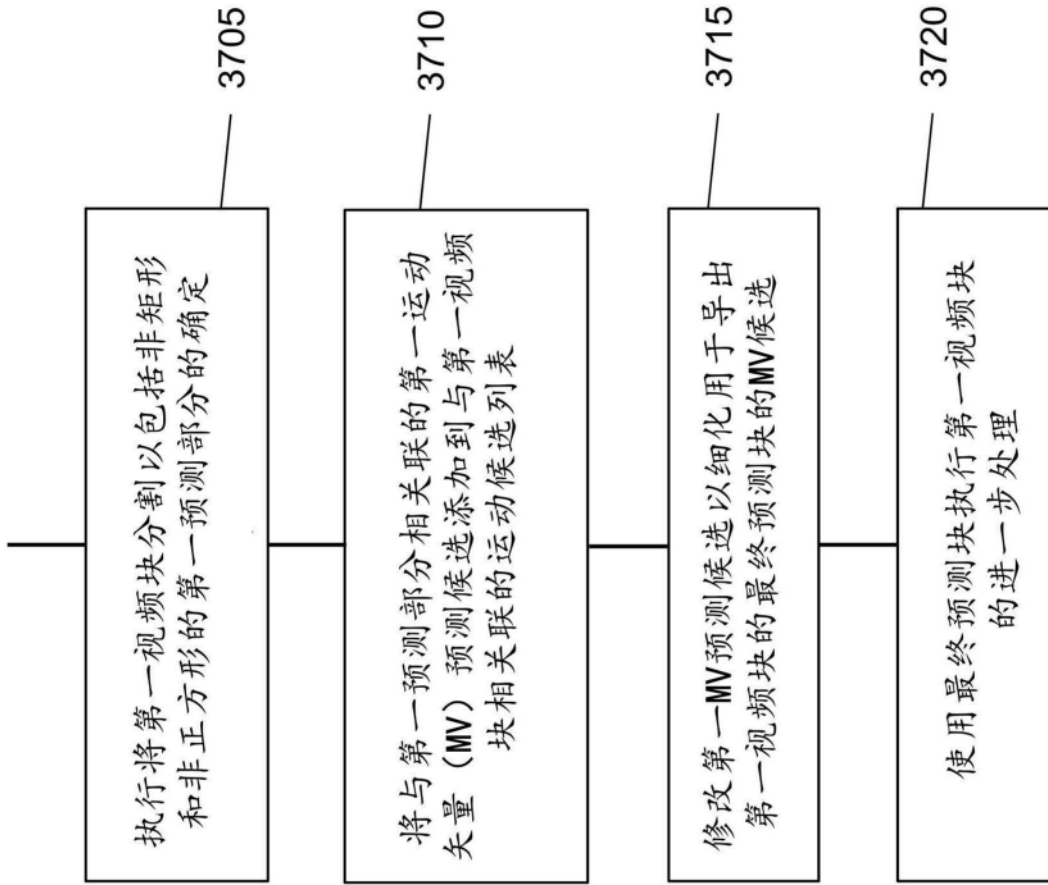


图37

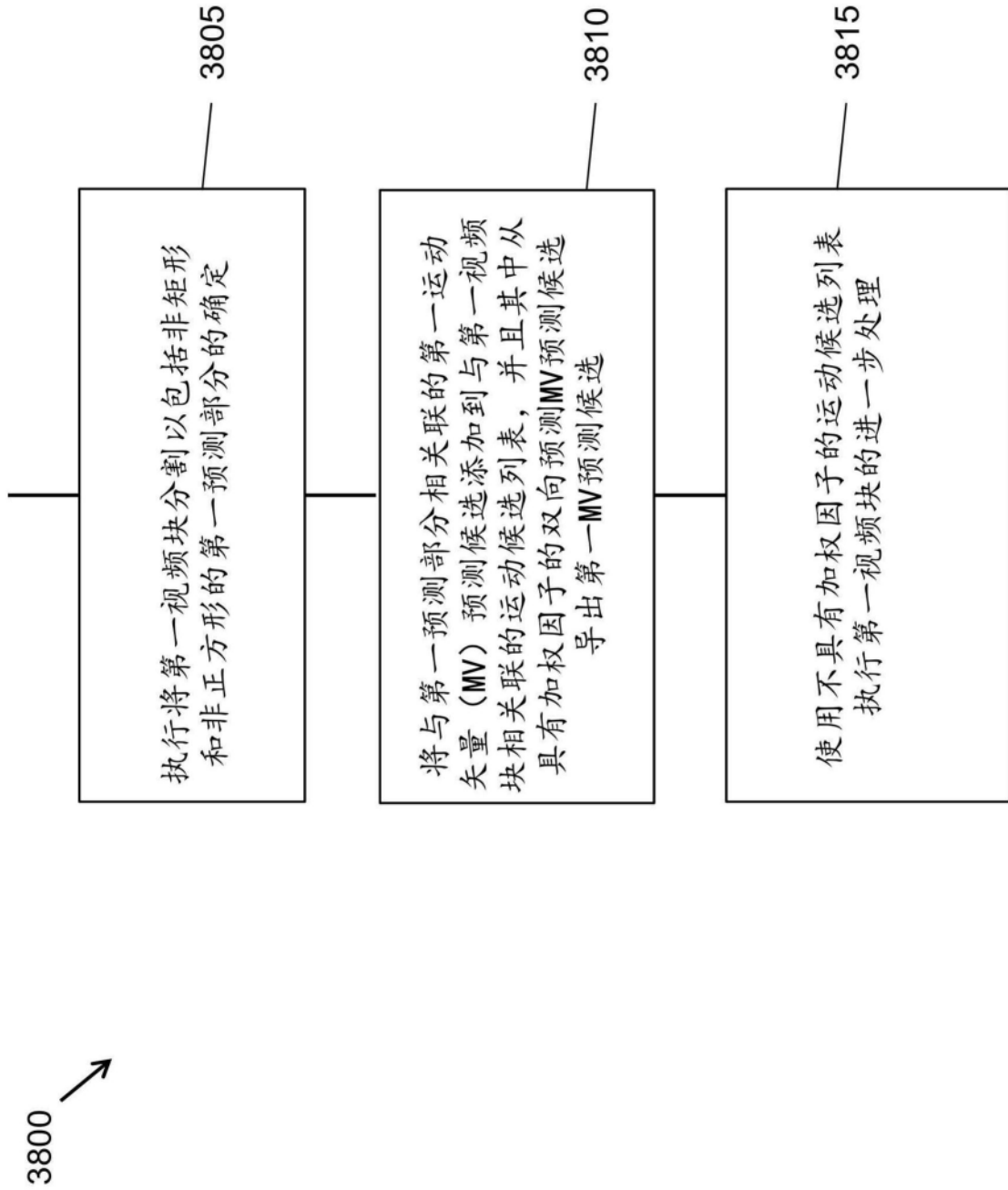


图38

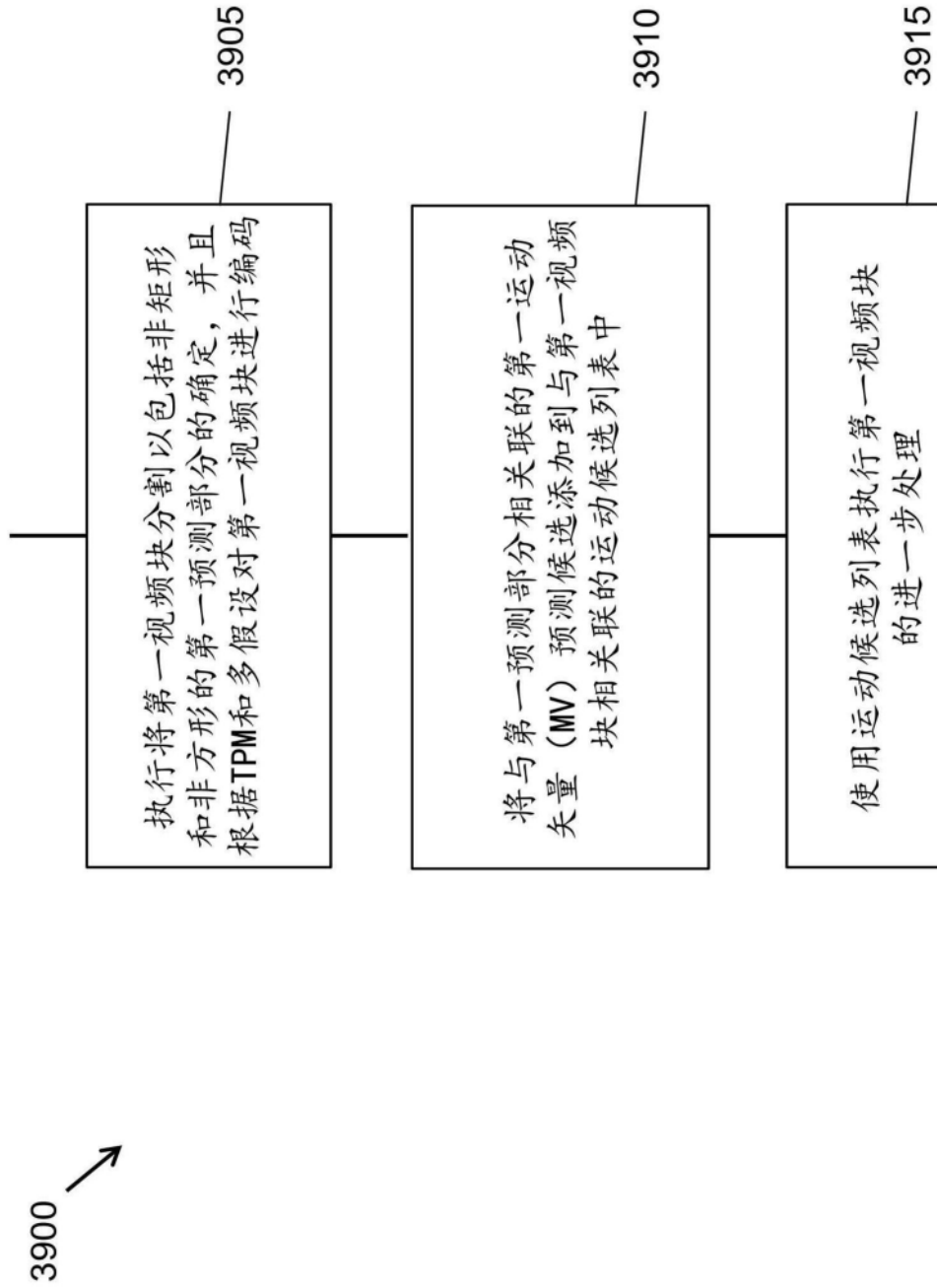


图39

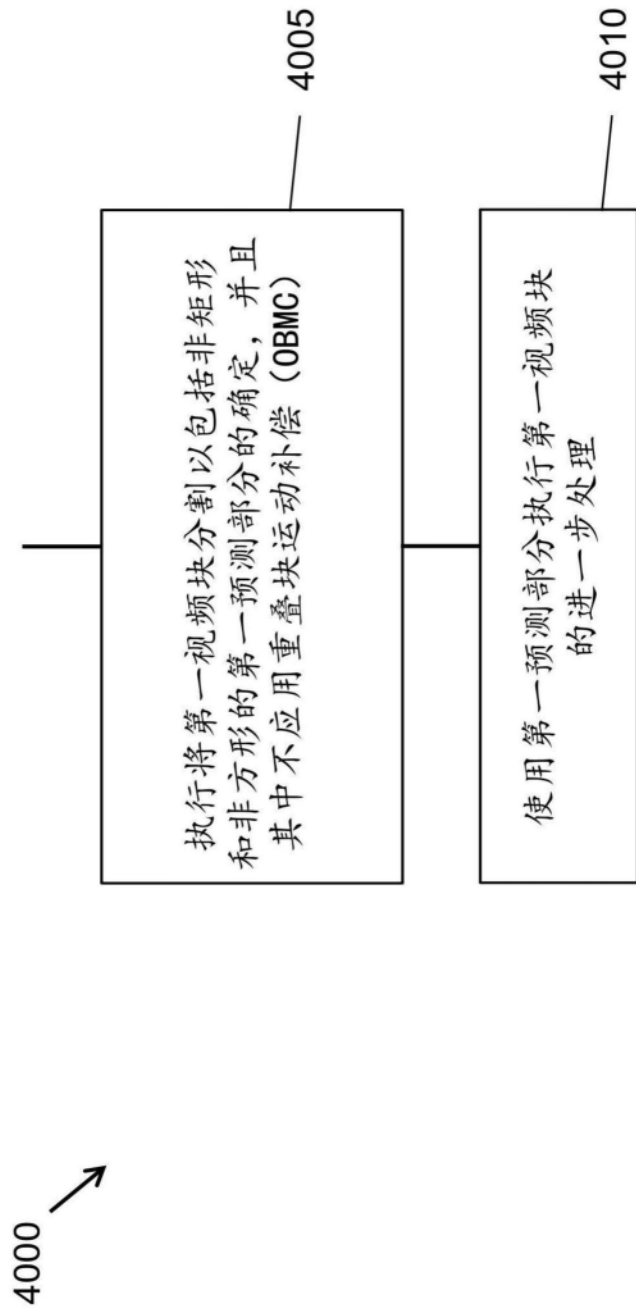


图40

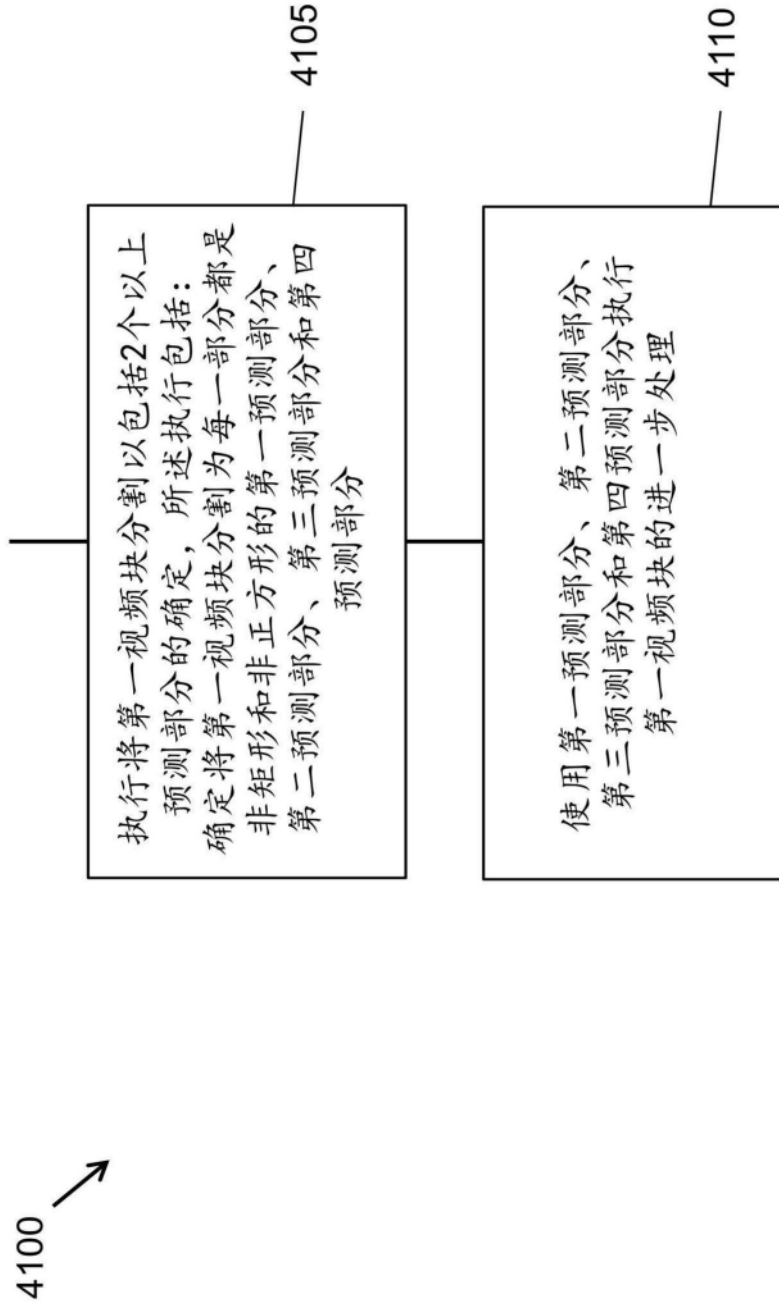


图41

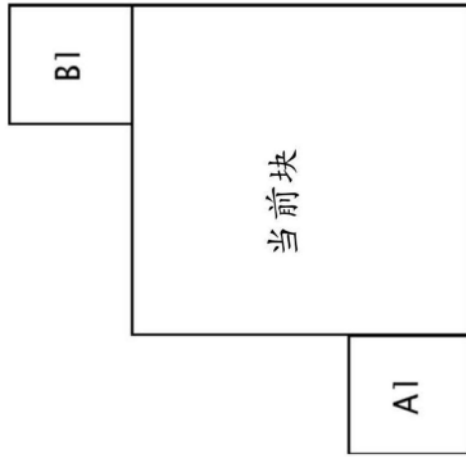


图42

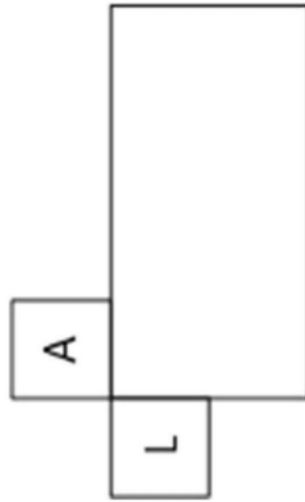


图43

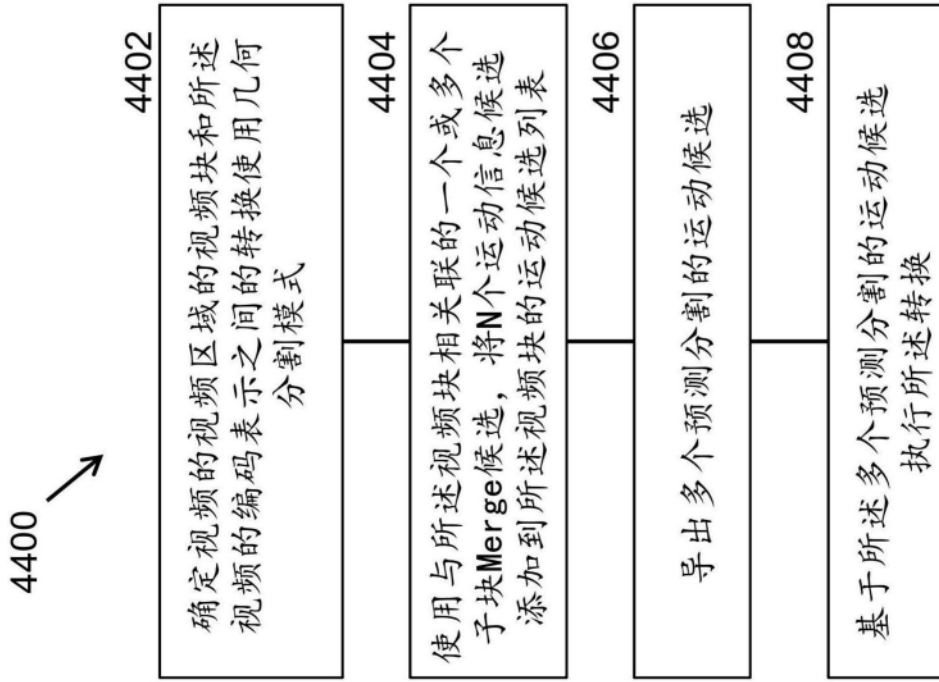


图44

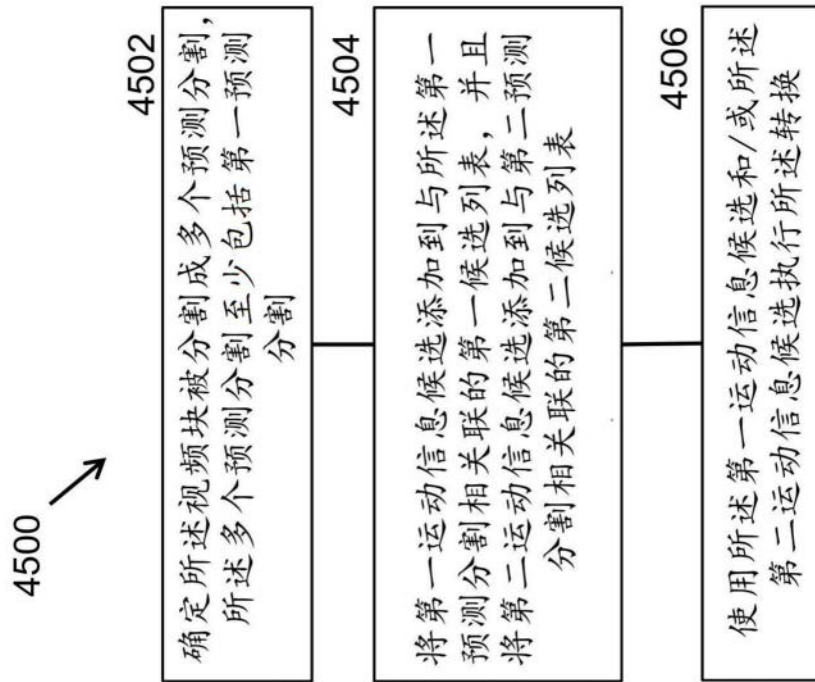


图45

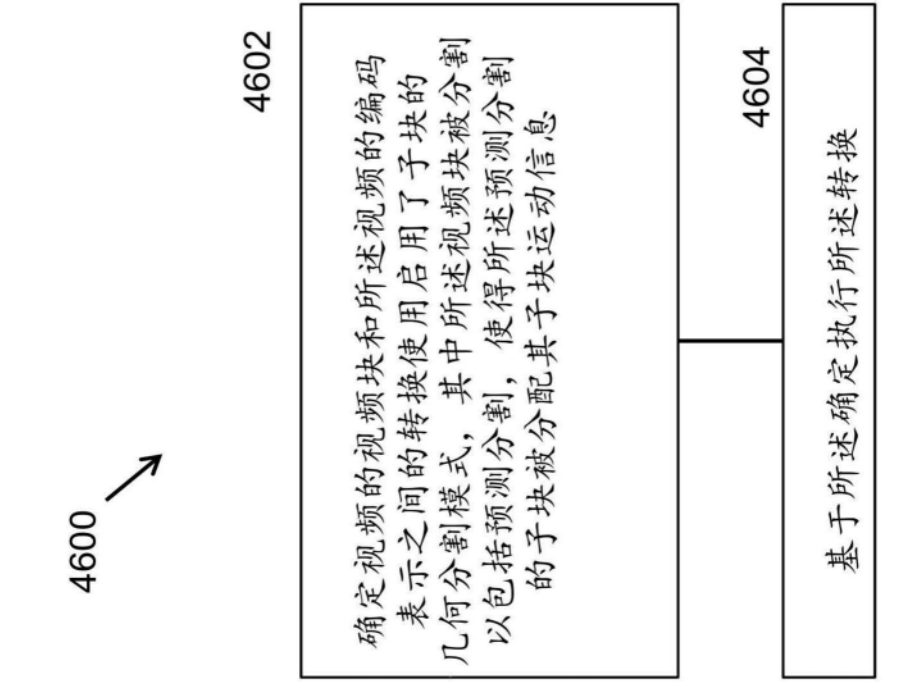


图46

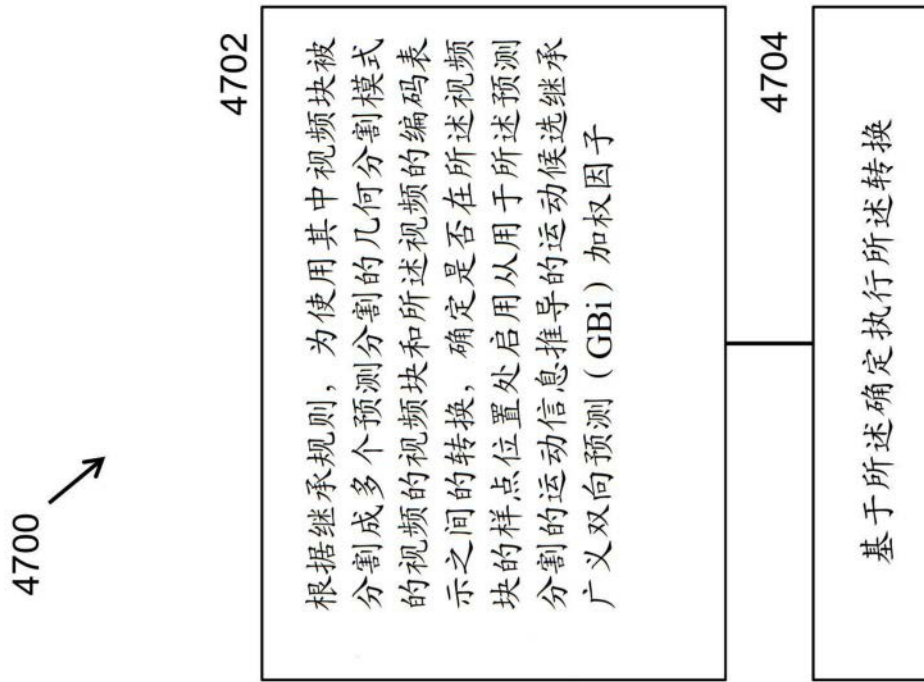


图47A

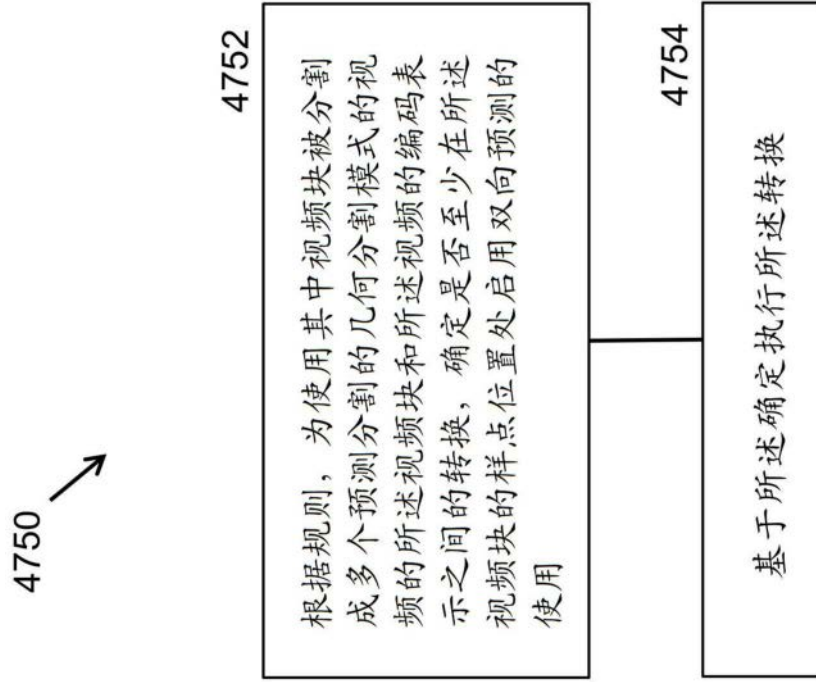


图47B

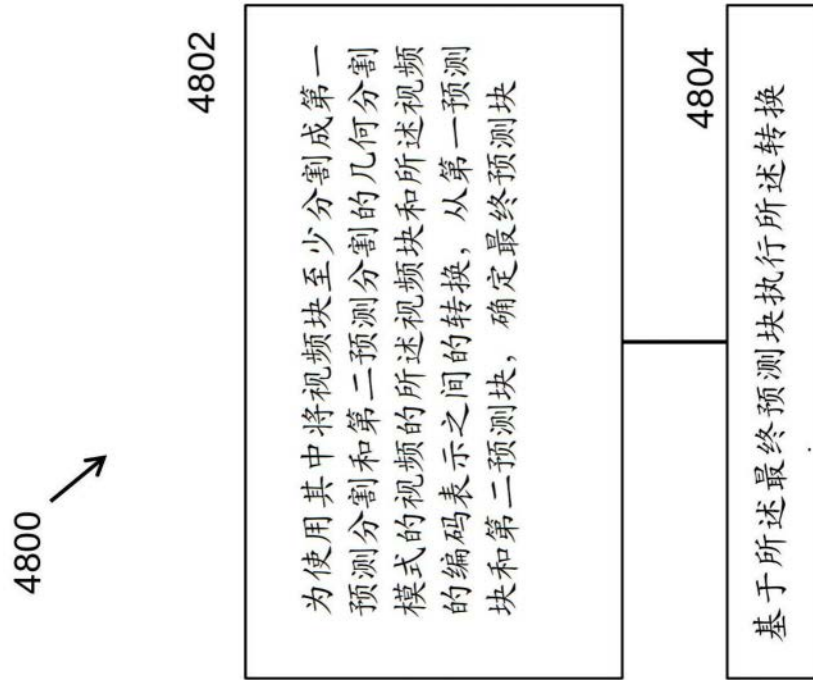


图48

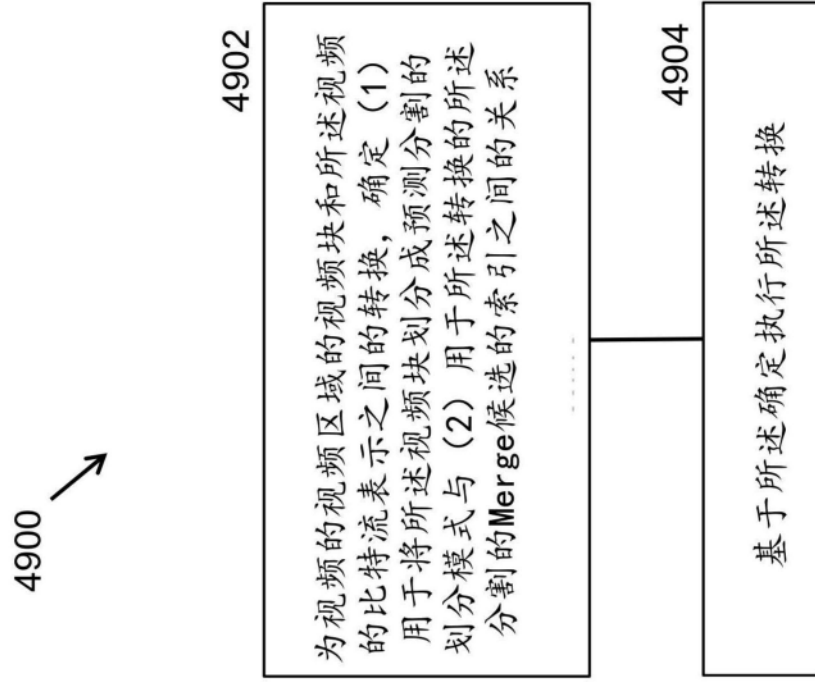


图49

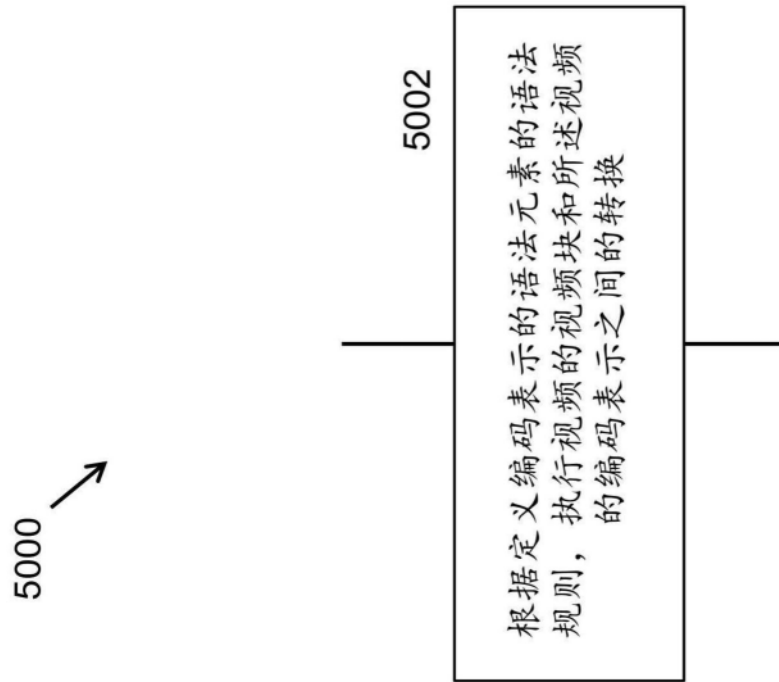


图50

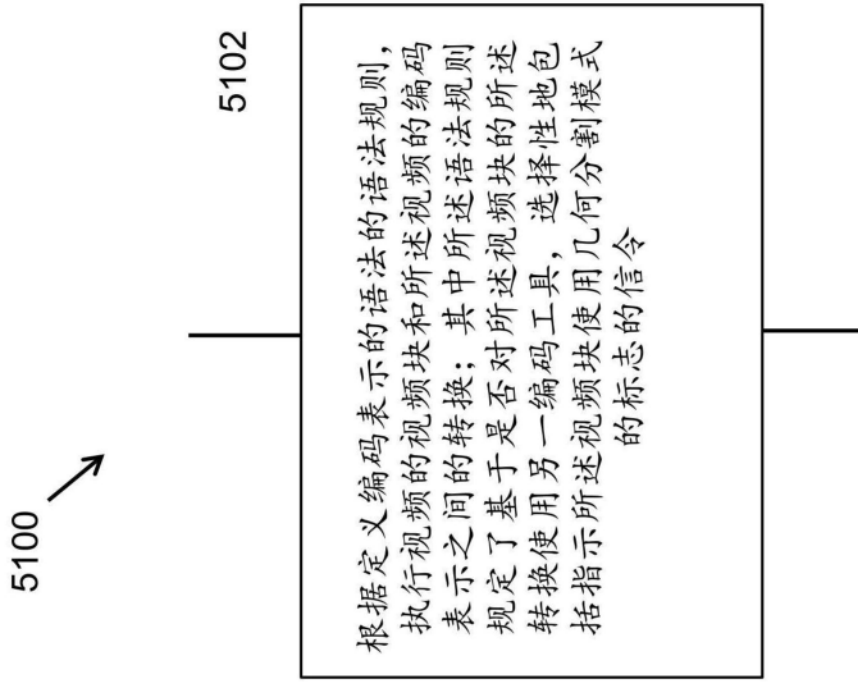


图51

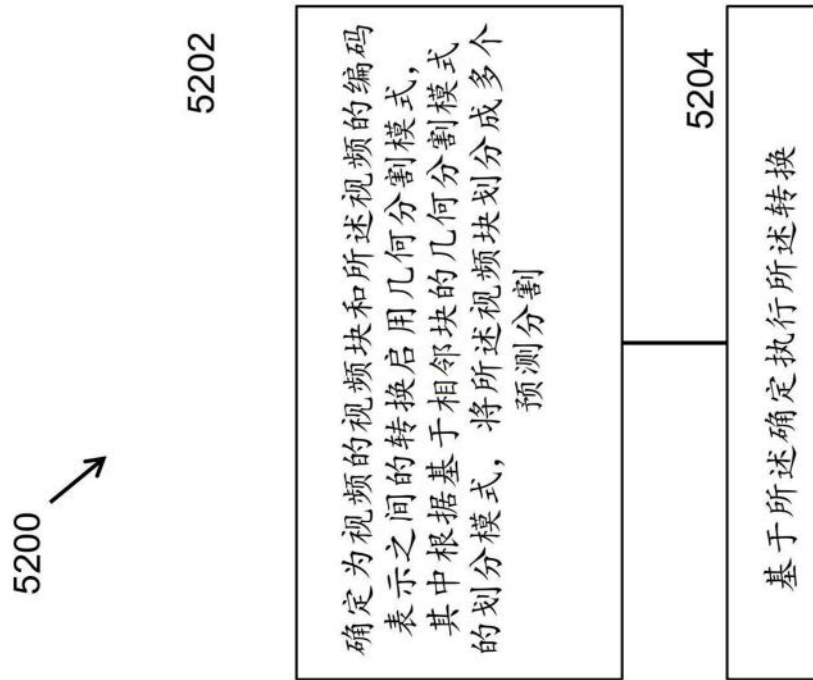


图52

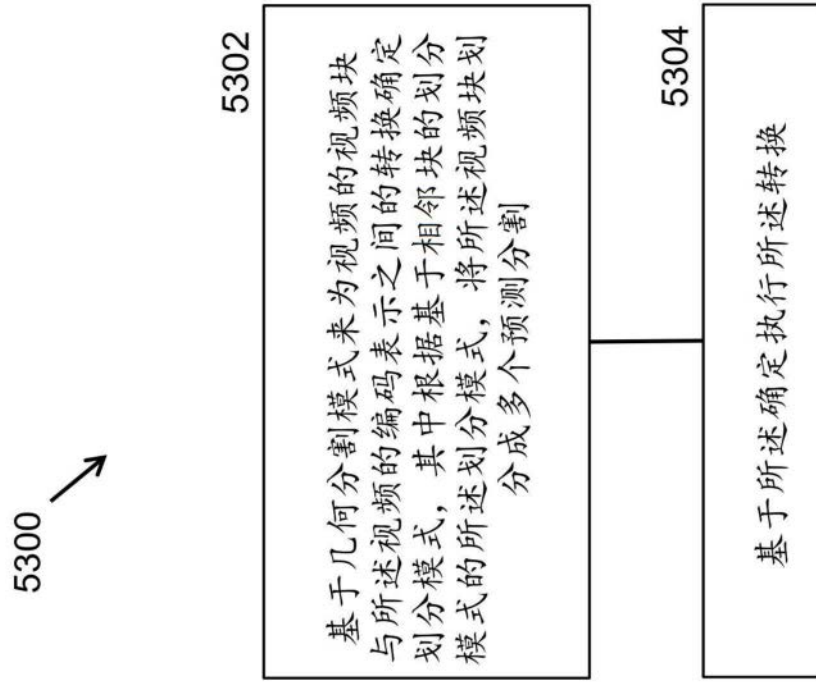


图53

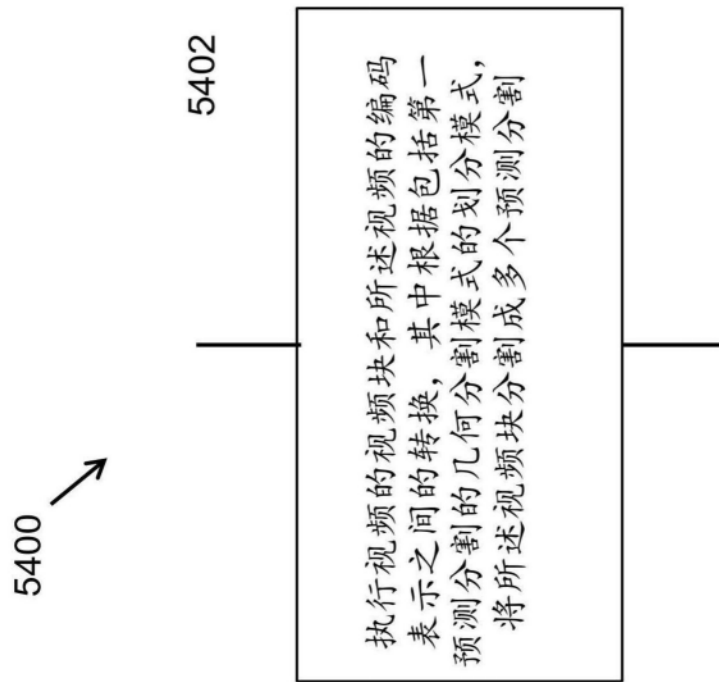


图54

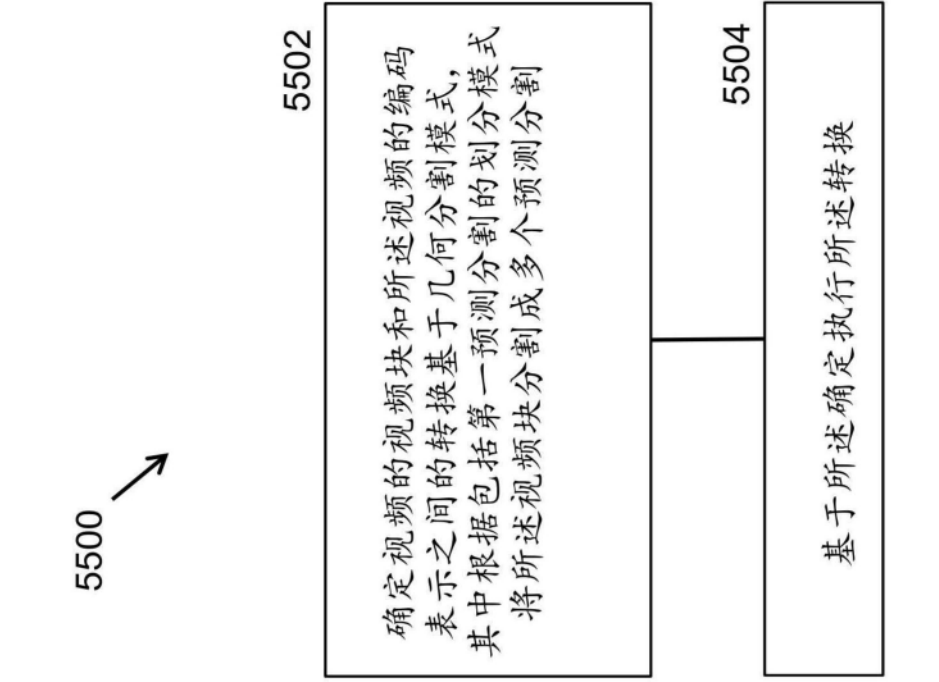


图55

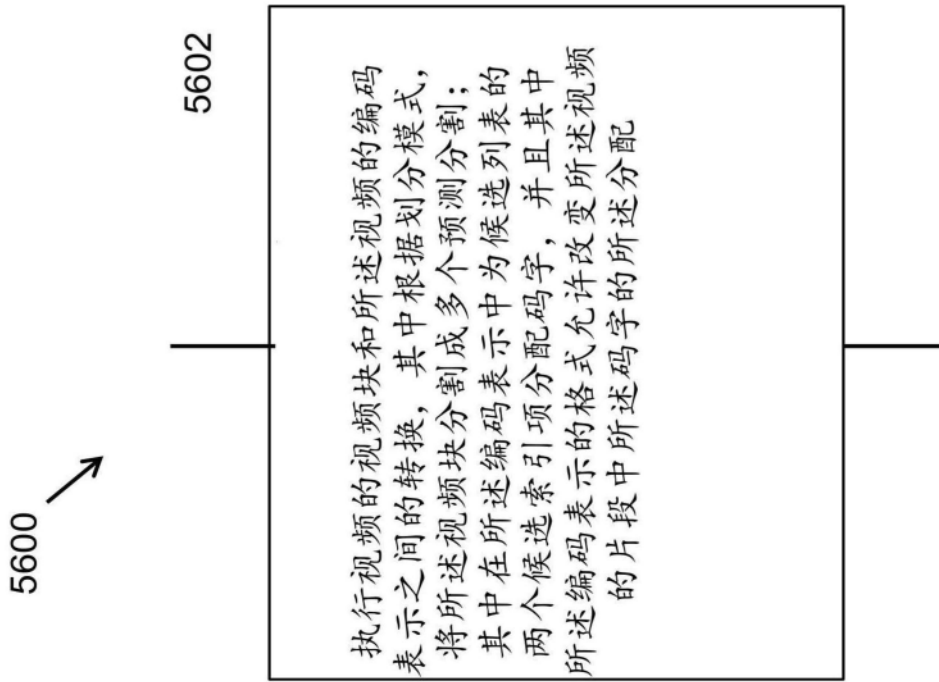


图56

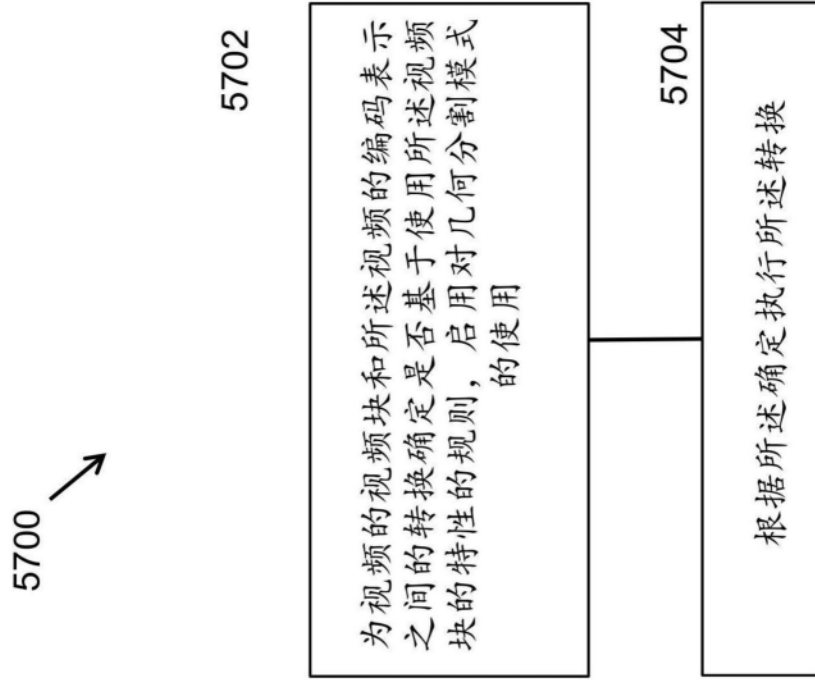


图57

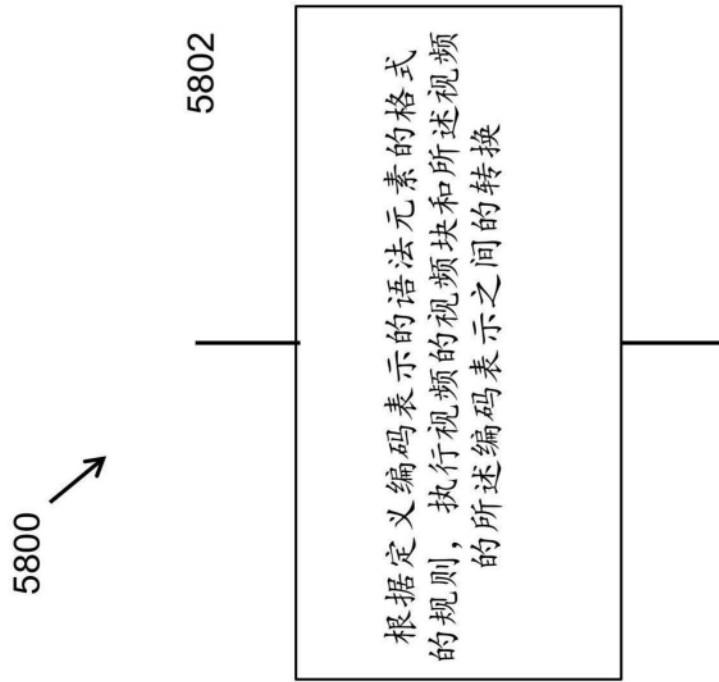


图58

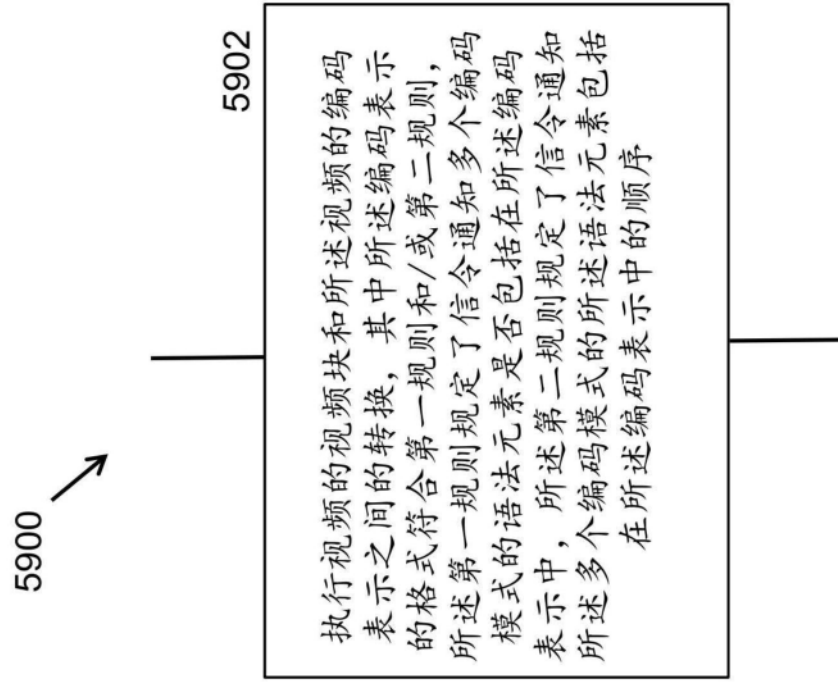


图59

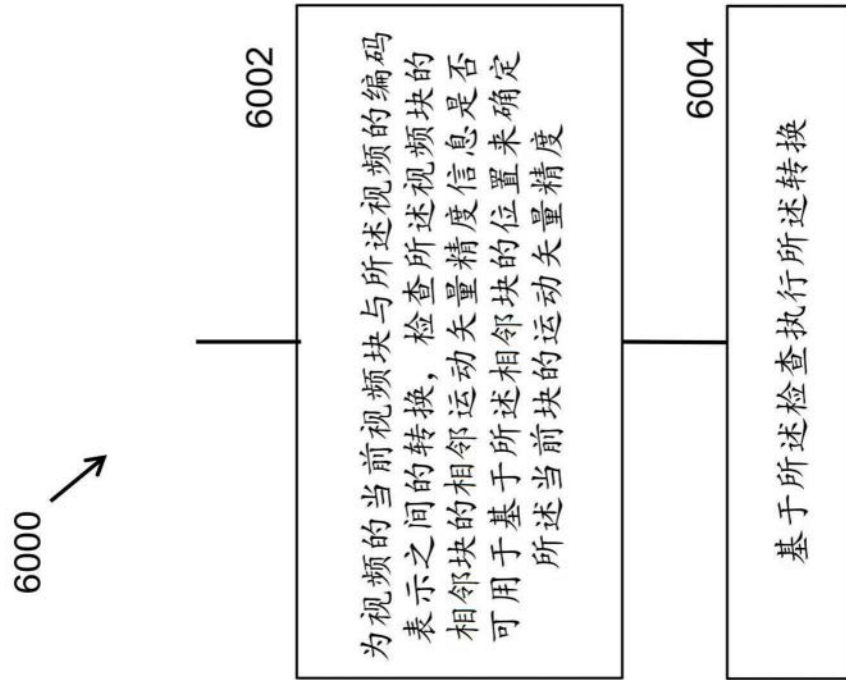


图60

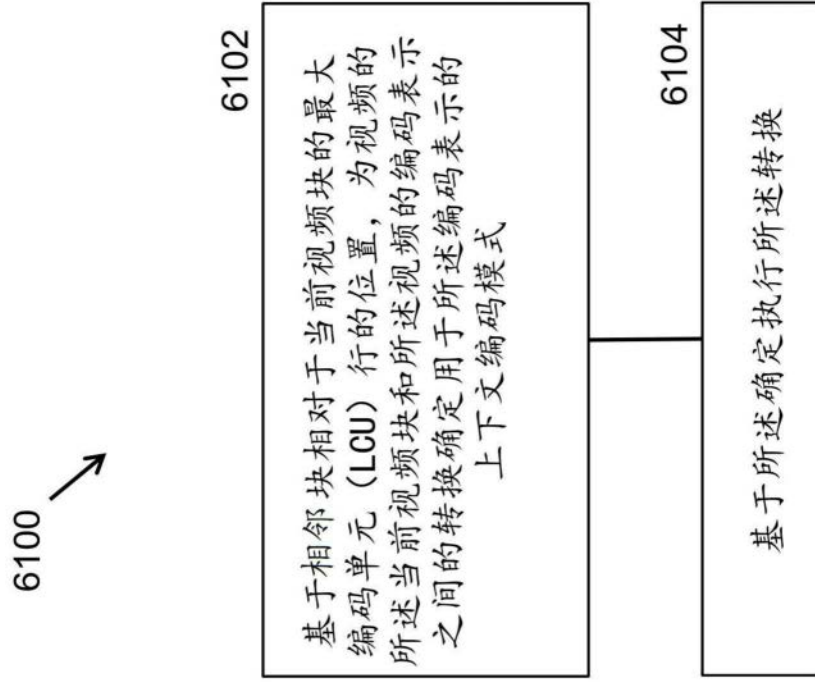


图61

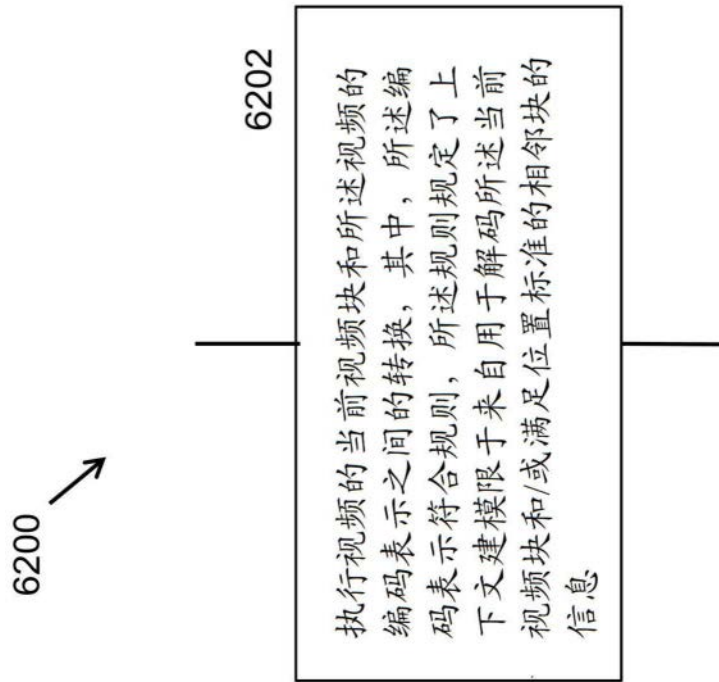


图62

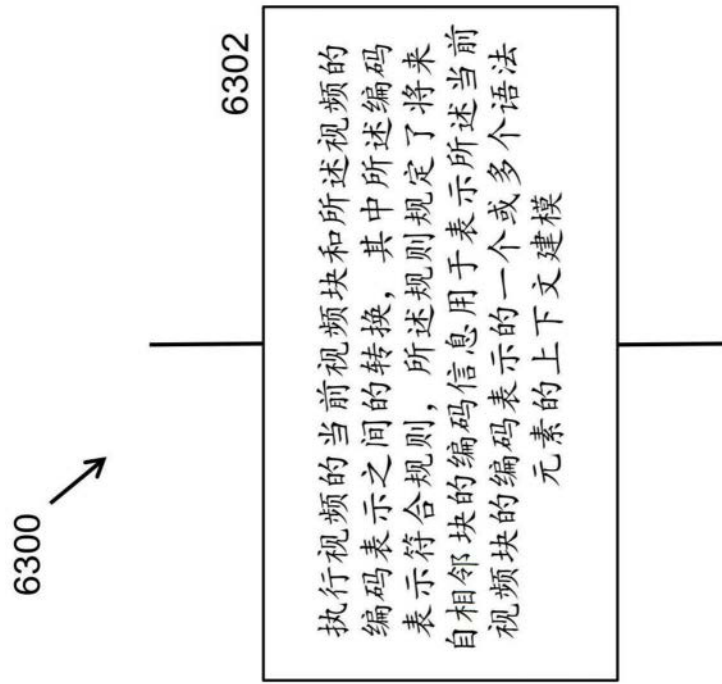


图63

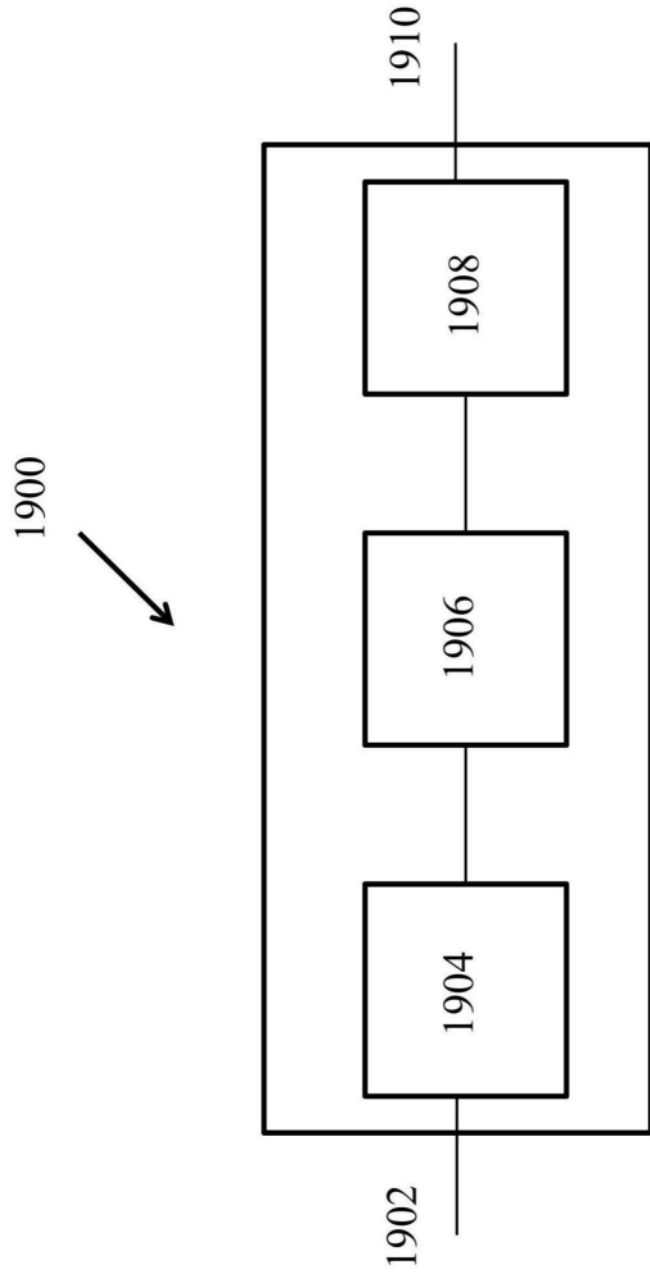


图64