



(12)发明专利申请

(10)申请公布号 CN 109417630 A

(43)申请公布日 2019.03.01

(21)申请号 201780041091.5

(74)专利代理机构 北京市柳沈律师事务所

(22)申请日 2017.06.27

11105

代理人 刘茵

(30)优先权数据

16305827.4 2016.06.30 EP

(51)Int.Cl.

(85)PCT国际申请进入国家阶段日

2018.12.29

H04N 19/52(2006.01)

(86)PCT国际申请的申请数据

PCT/EP2017/065809 2017.06.27

H04N 19/139(2006.01)

H04N 19/176(2006.01)

(87)PCT国际申请的公布数据

W02018/002024 EN 2018.01.04

H04N 19/46(2006.01)

H04N 19/14(2006.01)

H04N 19/513(2006.01)

(71)申请人 交互数字VC控股公司

地址 美国特拉华州

(72)发明人 F.加尔平 F.拉卡普 T.波里尔

P.博尔德斯

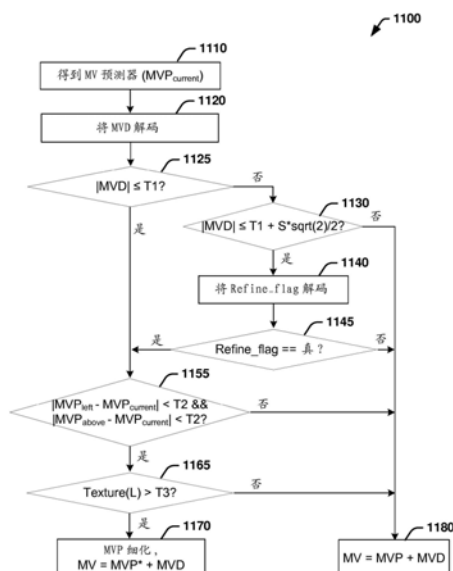
权利要求书2页 说明书19页 附图12页

(54)发明名称

利用自动运动信息细化的视频译码的方法和装置

(57)摘要

在特定实现方式中,视频解码器可以细化对于当前块从比特流解码的初始运动矢量预测器。为了减少信令开销,解码器可以基于已经可用的信息,例如,基于运动场是否在当前块周围均匀,以及在相邻块中是否存在某个级别的纹理性,来确定是否使用运动细化。从比特流解码的运动矢量差可以用于自动激活或停用运动细化而不在比特流中接收显式信令。例如,在运动矢量差小于阈值时,自动激活运动细化,并且在运动矢量差大于另一阈值时,自动停用运动细化。对应的视频编码器可以基于编码器判定来选择是否使用和以信号通知运动细化。



1. 一种视频解码的方法 (1100), 包括:

访问 (1110) 视频的当前块的运动矢量预测器和运动矢量差, 所述运动矢量预测器与第一运动分辨率相关联; 以及

基于所述运动矢量差来确定 (1125, 1130) 是否细化所述运动矢量预测器, 其中细化 (1170) 所述运动矢量预测器包括:

基于运动搜索来形成细化的运动矢量预测器, 所述细化的运动矢量预测器与第二运动分辨率相关联, 并且所述第二运动分辨率高于所述第一运动分辨率, 以及

基于所述细化的运动矢量预测器和所述运动矢量差来形成 (1170) 所述当前块的运动矢量, 其中基于所形成的运动矢量对所述当前块进行解码。

2. 一种装置 (1400), 包括至少一个存储器和一个或多个处理器, 所述一个或多个处理器被配置为:

访问视频的当前块的运动矢量预测器和运动矢量差, 所述运动矢量预测器与第一运动分辨率相关联; 以及

基于所述运动矢量差来确定是否细化所述运动矢量预测器, 其中所述一个或多个处理器被配置为通过执行以下来细化所述运动矢量预测器:

基于运动搜索来形成细化的运动矢量预测器, 所述细化的运动矢量预测器与第二运动分辨率相关联, 并且所述第二运动分辨率高于所述第一运动分辨率, 以及

基于所述细化的运动矢量预测器和所述运动矢量差来形成所述当前块的运动矢量, 其中基于所形成的运动矢量对所述当前块进行解码。

3. 如权利要求1所述的方法, 或者如权利要求2所述的装置, 其中在所述运动矢量差的幅度小于第一阈值时, 确定所述运动矢量预测器要被细化。

4. 如权利要求1或3所述的方法, 或者如权利要求2或3所述的装置, 其中在所述运动矢量差的幅度超过第二阈值时, 确定所述运动矢量预测器不要被细化, 还包括基于所述运动矢量预测器和所述运动矢量差对所述当前块进行解码。

5. 如权利要求1和3-4中任一项所述的方法, 或者如权利要求2-4中任一项所述的装置, 在所述运动矢量差的幅度超过第一阈值时, 还包括从比特流对标志进行解码, 其中是否要细化所述运动矢量预测器基于所解码的标志。

6. 如权利要求1和3-5中任一项所述的方法, 还包括, 或者如权利要求2-5中任一项所述的装置, 所述一个或多个处理器还被配置为执行:

访问相邻解码块的至少一个运动矢量, 其中如果所述当前块的所述运动矢量预测器与所述至少一个运动矢量之间的差小于第三阈值, 则确定所述运动矢量预测器要被细化。

7. 如权利要求1和3-6中任一项所述的方法, 还包括, 或者如权利要求2-6中任一项所述的装置, 所述一个或多个处理器还被配置为执行:

访问相邻解码块的多个像素, 其中如果所述多个像素的纹理级别超过第四阈值, 则确定所述运动矢量预测器要被细化。

8. 一种视频编码的方法 (1200), 包括:

访问 (1220) 运动矢量预测器, 所述运动矢量预测器与第一运动分辨率相关联;

确定 (1225) 与所述运动矢量预测器相对应的运动矢量差;

确定 (1225) 是否细化所述运动矢量预测器, 所述细化与第二运动分辨率相关联, 并且

所述第二运动分辨率高于所述第一运动分辨率；

基于所确定的运动矢量差来确定 (1280,1260,1265) 是否显式地以信号通知所述运动矢量预测器的所述细化；以及

对所述运动矢量差进行编码 (1290)。

9. 一种装置 (1400)，包括至少一个存储器和一个或多个处理器，所述一个或多个处理器被配置为：

访问运动矢量预测器，所述运动矢量预测器与第一运动分辨率相关联；

确定与所述运动矢量预测器相对应的运动矢量差；

确定是否细化所述运动矢量预测器，所述细化与第二运动分辨率相关联，并且所述第二运动分辨率高于所述第一运动分辨率；

基于所确定的运动矢量差来确定是否显式地以信号通知所述运动矢量预测器的所述细化；以及

对所述运动矢量差进行编码。

10. 如权利要求8所述的方法，或者如权利要求9所述的装置，其中在所述运动矢量差的幅度小于第一阈值时，不显式地以信号通知所述运动矢量预测器的所述细化。

11. 如权利要求8或10所述的方法，或者如权利要求9或10所述的装置，其中在所述运动矢量差的幅度超过第二阈值时，在没有显式信令的情况下停用所述运动矢量预测器的所述细化。

12. 如权利要求8和10-11中任一项所述的方法，或者如权利要求9-11中任一项所述的装置，在所述运动矢量差的幅度超过第一阈值时，还包括将标志编码到比特流中以显式地以信号通知是否要细化所述运动矢量预测器。

13. 如权利要求8和10-12中任一项所述的方法，还包括，或者如权利要求9-12中任一项所述的装置，所述一个或多个处理器还被配置为执行：

访问相邻重建块的至少一个运动矢量，其中如果所述当前块的所述运动矢量预测器与所述至少一个运动矢量之间的差小于第三阈值，则确定所述运动矢量预测器要被细化。

14. 如权利要求8和10-13中任一项所述的方法，还包括，或者如权利要求9-13中任一项所述的装置，所述一个或多个处理器还被配置为执行：

访问相邻重建块的多个像素，其中如果所述多个像素的纹理级别超过第四阈值，则确定所述运动矢量预测器要被细化。

15. 一种非临时性计算机可读储存介质，其上存储有用于实现根据权利要求1、3-8和10-14中任一项的方法的指令。

16. 一种根据权利要求8和10-14中任一项的方法生成的比特流。

利用自动运动信息细化的视频译码的方法和装置

技术领域

[0001] 本原理通常涉及视频编码和解码的方法和装置,并且更具体地涉及利用运动信息细化(refinement)的视频编码和解码的方法和装置。

背景技术

[0002] 为了实现高压缩效率,图像和视频译码方案通常采用预测和变换来影响(leverage)视频内容中的空间和时间冗余。通常,使用帧内或帧间预测来开发帧内或帧间相关,然后对原始图像与预测图像之间的差异(通常表示为预测误差或预测残差)进行变换、量化和熵译码。为了重建视频,通过与预测、变换、量化和熵译码相对应的逆处理来对压缩数据进行解码。

发明内容

[0003] 根据总体方面,呈现了一种视频解码的方法,包括:访问视频的当前块的初始运动矢量预测器,初始运动矢量预测器与第一运动分辨率相关联;基于运动矢量差来确定是否细化初始运动矢量预测器;以及如果确定初始运动矢量预测器要被细化,则基于运动搜索来细化初始运动矢量预测器以形成细化的运动矢量预测器,并且基于细化的运动矢量预测器对当前块进行解码,细化的运动矢量预测器与第二运动分辨率相关联,并且第二运动分辨率高于第一运动分辨率。

[0004] 在运动矢量差的幅度小于第一阈值时,可以确定初始运动矢量预测器要被细化。在运动矢量差的幅度超过第二阈值时,可以确定初始运动矢量预测器不要被细化,并且基于初始运动矢量预测器对当前块进行解码。这里,自动执行运动细化的激活或停用,而无需显式信令。

[0005] 在运动矢量差的幅度超过第一阈值而不是第二阈值时,可以从比特流对标志进行解码,并且是否要细化初始运动矢量预测器基于所解码的标志。

[0006] 相邻解码块的运动矢量也可以用于确定是否执行运动细化。例如,如果当前块的初始运动矢量预测器与相邻解码块的至少一个运动矢量之间的差小于第三阈值,则确定初始运动矢量预测器要被细化。此外,也可以使用纹理级别。例如,如果相邻解码块的多个像素的纹理级别超过第四阈值,则确定初始运动矢量预测器要被细化。

[0007] 根据另一总体方面,呈现了一种视频编码的方法,包括:访问初始运动矢量预测器,初始运动矢量预测器与第一运动分辨率相关联;基于与初始运动矢量预测器相对应的运动矢量差来确定是否显式地以信号通知初始运动矢量预测器的细化,该细化与第二运动分辨率相关联,并且第二运动分辨率高于第一运动分辨率;以及对运动矢量差进行编码。

[0008] 在运动矢量差的幅度小于第一阈值而不是第二阈值时,不显式地以信号通知初始运动矢量预测器的细化。在运动矢量差的幅度超过第二阈值时,在没有显式信令的情况下停用初始运动矢量预测器的细化。

[0009] 在运动矢量差的幅度超过第一阈值时,将标志编码到比特流中以显式地以信号通知是否要细化初始运动矢量预测器。

[0010] 相邻重建块的运动矢量也可以用于确定是否执行运动细化。例如,如果当前块的初始运动矢量预测器与相邻重建块的至少一个运动矢量之间的差小于第三阈值,则确定初始运动矢量预测器要被细化。此外,也可以使用纹理级别。例如,如果多个像素的纹理级别超过第四阈值,则确定初始运动矢量预测器要被细化。

[0011] 本实施例还提供了用于执行这些方法的装置。

[0012] 本实施例还提供了非临时性计算机可读储存介质,其上存储有用于执行上述方法中的任何一个的指令。

[0013] 本实施例还提供了根据上述方法生成的比特流。

附图说明

[0014] 图1图示了示例性HEVC(高效视频译码)编码器。

[0015] 图2A是描绘当前块的五个空间候选 $\{a_1, b_1, b_0, a_0, b_2\}$ 的位置的图示示例,图2B是描绘使用AMVP(高级运动矢量预测)的示例性运动矢量表示的图示示例,以及图2C是描绘运动补偿预测的图示示例。

[0016] 图3图示了示例性HEVC视频解码器的框图。

[0017] 图4图示了根据本原理的实施例的在解码器处执行运动矢量细化的示例性方法。

[0018] 图5是描绘整数像素、半像素、四分之一像素和八分之一像素的像素位置的图示示例。

[0019] 图6A图示了要解码的示例性PU,以及图6B是图示出用于度量不连续性的相邻重建样本的L形集合和预测样本的L形集合的图示示例。

[0020] 图7A是图示出局部梯度的图示示例,以及图7B是图示出梯度的二阶矩的图示示例。

[0021] 图8图示了根据本原理的实施例的在编码器处执行运动矢量细化的示例性方法。

[0022] 图9图示了根据本原理的实施例的在解码器处执行运动矢量预测器细化的示例性方法。

[0023] 图10图示了根据本原理的实施例的在编码器处执行运动矢量预测器细化的示例性方法。

[0024] 图11图示了根据本原理的实施例的在解码器处利用自动运动细化激活来执行运动矢量预测器细化的示例性方法。

[0025] 图12图示了根据本原理的实施例的在编码器处利用自动运动细化激活来执行运动矢量预测器细化的示例性方法。

[0026] 图13图示了根据本原理的实施例的确定是否对特定MVP使用自适应运动细化的示例性方法。

[0027] 图14图示了示例性系统的框图,其中可以实现本原理的示例性实施例的各个方面。

具体实施方式

[0028] 图1图示了示例性HEVC (高效视频译码) 编码器100。为了对具有一个或多个图片的视频序列进行编码,将图片划分成一个或多个切片,其中每个切片可以包括一个或多个切片片段。将切片片段组织成译码单元、预测单元和变换单元。

[0029] HEVC规范在“块”与“单元”之间进行区分,其中“块”寻址样本阵列(例如,亮度,Y)中的特定区域,而“单元”包括所有编码色彩分量(Y,Cb,Cr或单色)的同位(collocated)块、与块相关联的语法元素和预测数据(例如,运动矢量)。

[0030] 对于译码,将图片划分成具有可配置大小的方形的译码树块(CTB),并且将连续的译码树块集合分组成切片。译码树单元(CTU)包含编码色彩分量的CTB。CTB是划分成译码块(CB)的四叉树的根,并且将译码块划分成一个或多个预测块(PB)并形成划分成变换块(TB)的四叉树的根。对应于译码块、预测块和变换块,译码单元(CU)包括预测单元(PU)和树形结构的变换单元(TU)集合,PU包括所有色彩分量的预测信息,并且TU包括每个色彩分量的残差译码语法结构。亮度分量的CB、PB和TB的大小适用于对应的CU、PU和TU。在本申请中,术语“块”可以用于指代CTU、CU、PU、TU、CB、PB和TB中的任何一个。另外,“块”也可以用于指代如H.264/AVC或其他视频译码标准中指定的宏块、分区和子块,并且更一般地指代各种大小的数据阵列。

[0031] 在示例性编码器100中,如下所述,由编码器元件对图片进行编码。要编码的图片以CU为单位进行处理。使用帧内或帧间模式对每个CU进行编码。在以帧内模式对CU进行编码时,其执行帧内预测(160)。在帧间模式中,执行运动估计(175)和补偿(170)。编码器判定(105)使用帧内模式或帧间模式中的哪一个对CU进行编码,并且通过预测模式标志来指示帧内/帧间判定。通过从原始图像块中减去(110)预测块来计算预测残差。

[0032] 根据相同切片内的重建的相邻样本来预测帧内模式中的CU。在HEVC中,35个帧内预测模式的集合是可用的,其包括DC、平面和33个角度预测模式。根据与当前块相邻的行和列来重建帧内预测参考。使用来自先前重建块的可用样本,该参考在水平和垂直方向上延伸超过块大小的两倍。在角度预测模式用于帧内预测时,可以沿着由角度预测模式指示的方向来复制参考像素。

[0033] 可以使用两个不同选项对当前块的适用亮度帧内预测模式进行译码。如果适用模式被包括在三个最可能模式(MPM)的构建列表中,则通过MPM列表中的索引来以信号通知模式。否则,通过模式索引的固定长度二进制化来以信号通知模式。从顶部和左侧相邻块的帧内预测模式导出三个最可能模式。

[0034] 对于帧间CU,将对应的译码块进一步划分成一个或多个预测块。在PB级上执行帧间预测,并且对应的PU包含关于如何执行帧间预测的信息。

[0035] 可以以两种方法、即“合并模式”和“高级运动矢量预测(AMVP)”来以信号通知运动信息(即,运动矢量和参考索引)。

[0036] 在合并模式中,视频编码器或解码器基于已译码块来组装候选列表,并且视频编码器以信号通知候选列表中的候选之一的索引。在解码器侧,基于以信号通知的候选来重建运动矢量(MV)和参考图片索引。

[0037] 合并模式中的可能候选的集合包括空间相邻者候选、时间候选和所生成候选。图2A示出了当前块210的五个空间候选 $\{a_1, b_1, b_0, a_0, b_2\}$ 的位置。对于每个候选位置,根据 a_1 、

b_1 、 b_0 、 a_0 、 b_2 的顺序来检查可用性,并且然后移除候选中的冗余。

[0038] 在切片首标中指定合并候选的最大数量 N 。如果合并候选的数量大于 N ,则仅使用前 $N-1$ 个空间候选和时间候选。否则,如果合并候选的数量小于 N ,则将候选集合填充直到最大数量 N 。

[0039] 在AMVP中,视频编码器或解码器基于根据已译码块确定的运动矢量来组装候选列表。然后,视频编码器以信号通知候选列表中的索引以识别运动矢量预测器(MVP)并以信号通知运动矢量差(MVD)。在解码器侧,将运动矢量(MV)重建为 $MVP+MVD$ 。

[0040] 在AMVP中,仅选择两个空间运动候选。第一空间运动候选选自左侧位置 $\{a_0, a_1\}$,而第二空间运动候选选自上方位置 $\{b_0, b_1, b_2\}$,同时保持搜索顺序,如两个集合中指示的那样。如果运动矢量预测器的数量不等于二,则可以包括时间MV预测候选。如果时间候选不可用,则使用零运动矢量来填充候选集合。

[0041] 图2B图示了使用AMVP的示例性运动矢量表示。对于要编码的当前块(240),可以通过运动估计来获得运动矢量($MV_{current}$)。使用来自左侧块(230)的运动矢量(MV_{left})和来自上方块(220)的运动矢量(MV_{above}),可以将运动矢量预测器选择为 $MVP_{current}$ 。然后,可以将运动矢量差计算为 $MVD_{current} = MV_{current} - MVP_{current}$ 。

[0042] 在HEVC中,用于运动补偿的运动信息的精度是亮度分量的四分之一样本(也称为四分之一像素或1/4像素)和色度分量的八分之一样本(也称为1/8像素)。7抽头或8抽头内插滤波器用于分数样本(fractional-sample)像素位置的内插,即,对于亮度可以寻址水平和垂直方向二者上的全像素定位的1/4、1/2和3/4。

[0043] 然后,对预测残差进行变换(125)和量化(130)。对量化的变换系数以及运动矢量和其他语法元素进行熵译码(145)以输出比特流。编码器也可以跳过变换并且在 4×4 TU的基础上将量化直接应用于未变换的残差信号。编码器也可以绕过变换和量化二者,即,在不应用变换或量化处理的情况下直接对残差进行译码。在直接PCM译码中,不应用预测,并且将译码单元样本直接译码到比特流中。

[0044] 编码器对编码块进行解码以提供用于进一步预测的参考。对量化的变换系数进行解量化(140)和逆变换(150)以对预测残差进行解码。将解码预测残差和预测块组合(155),重建图像块。例如,将滤波器(165)应用于重建图片,以执行去块/SAO(样本自适应偏移)滤波以减少块效应伪像。将滤波图像存储在参考图片缓冲器(180)处。

[0045] 图3图示了示例性HEVC视频解码器300的框图。在示例性解码器300中,如下所述,由解码器元件对比特流进行解码。视频解码器300通常执行与如图1中描述的编码通道对等的解码通道,其执行视频解码,作为编码视频数据的部分。

[0046] 具体地,解码器的输入包括视频比特流,其可以由视频编码器100生成。首先对比特流进行熵解码(330)以获得变换系数、运动矢量和其他译码信息。对变换系数进行解量化(340)和逆变换(350)以对预测残差进行解码。将解码预测残差和预测块组合(355),重建图像块。预测块可以从帧内预测(360)或运动补偿预测(即,帧间预测)(375)获得(370)。如上所述,AMVP和合并模式技术可以用于导出用于运动补偿的运动矢量,运动补偿可以使用内插滤波器来计算参考块的子整数像素的内插值。将滤波器(365)应用于重建图像。将滤波图像存储在参考图片缓冲器(380)处。

[0047] 在视频压缩中,帧间预测模式允许使用来自先前重建/解码的图片的至少一个运

动补偿块来预测一个块(例如,预测单元)。例如,如图2C中所示,使用帧间预测模式对当前块(255)进行编码,块225和245是参考图片中的共同定位块,并且块215和235是用于运动补偿预测的块。对识别解码图片缓冲器(DPB)中的重建图片和一个运动信息(运动矢量差,MVD)的至少一个参考索引(例如,refIdx0)进行编码,以在解码器处实现至少一个运动矢量(MV,例如,mv0)的重建。在使用双向预测时,对附加参考索引(例如,refIdx1)和运动信息进行编码,以便重建运动矢量(例如,mv0和mv1)。在一些情况下,可以根据先前解码的数据来推断运动矢量(例如,HEVC中的合并模式)。

[0048] 可以如下执行MV值的重建:

[0049] $MV_X = MVP_X + MVD_X, MV_Y = MVP_Y + MVD_Y$, (1)

[0050] 其中 MV_X 和 MV_Y 分别是水平和垂直运动矢量分量,MVP(MVP_X, MVP_Y)是根据先前重建的数据构建的运动矢量预测器,以及MVD(MVD_X, MVD_Y)是在比特流中编码和传送的运动矢量差。

[0051] 通常以与经解码MV值相对应的精度对MVD值进行编码。例如,HEVC使用四分之一像素(即,1/4像素)作为运动矢量分辨率。

[0052] 例如,将MV分辨率从1/4像素增加到1/8像素一般可以改善预测。然而,对于低比特率,相对于每块编码的其他数据,MVD数据的译码可能具有相对高的比特率成本。因此,整体压缩效率可能不一定随MV分辨率而提高。

[0053] 为了提高压缩效率,存在一些关于使用自适应运动分辨率的现有工作,例如,以选择整数像素或1/4像素之间的运动分辨率,或者在1/4像素与1/6像素之间选择。然而,指示使用哪个运动矢量分辨率可能导致压缩效率的降低,因为需要在比特流中发送额外的边信息。

[0054] 为了减少开销,Lakshman、Haricharan等人的题为“Conditional motion vector refinement for improved prediction”,图片编码会议(PCS),2012.IEEE的文章(在下文中为“Lakshman”)定义了编码器和解码器都知道的一组规则,以在四分之一样本MV分辨率与1/6样本MV分辨率之间推断MV分辨率,而无需任何显式的逐块前向信令。高分辨率MV作为以细化信息增强的常规四分之一样本MV被传送,该细化信息将运动矢量的分辨率从四分之一样本分辨率增加到样本的六分之一。指向整数或半样本位置的重建MV分量保持不变。对于指向四分之一或四分之三位置的MV分量,解码器使用以下条件来推断细化的存在:

[0055] -在P切片的情况下,始终发送MV细化信息。

[0056] -在Bi预测的情况下,针对访问来自包含高纹理的参考图片的样本的预测发送MV细化。

[0057] -在Bi预测不访问高纹理参考图片的情况下,仅针对来自预定义参考图片列表的图片发送MV细化。

[0058] -对于B切片中的单假设预测,不使用MV细化。

[0059] Chen、Jianle等人的题为“Further improvements to HMKTA-1.0”,VCEG-AZ07,ITU-T/SG16视频编码专家组(VCEG)第52次会议:2015年6月19日至26日,华沙,波兰的文章(在下文中为“Chen”)描述了高级运动矢量分辨率(AMVR)模式。具体地,可以利用四分之一像素分辨率或整数像素分辨率对PU的运动矢量与预测运动矢量之间的MVD进行译码。在译码单元(CU)级控制MVD分辨率,并且对于具有至少一个非零MVD分量的每个CU有条件地以信

号通知整数MVD分辨率标志。在整数MVD分辨率标志为假或者对于CU不被译码时,默认的四分之一像素MV分辨率用于属于CU的所有PU。否则,属于CU的以AMVP模式译码的所有PU使用整数MV分辨率,而以合并模式译码的PU仍然使用四分之一像素MV分辨率。在PU使用整数MV分辨率时,通过将四分之一像素MV舍入(round)到整数像素MV,利用整数MV来填充AMVP候选列表。

[0060] 本原理涉及视频编码和解码中的运动矢量细化。在一个实施例中,在比特流中接收第一分辨率MVD(例如,1/4像素),并且解码器基于MVD获得初始运动矢量,并且使用已经解码的相邻样本,细化初始运动矢量以获得高于第一分辨率的第二运动分辨率(例如,1/8像素)。也在编码器处执行运动信息的细化以避免不匹配。因为在编码器和解码器二者处执行运动细化,所以在比特流中不传送附加信息以指示运动细化,例如,没有使用显式信令来指示从1/4像素到1/8像素的运动矢量的细化。在变型中,合并模式用于当前块(即,在比特流中没有接收到MVD),并且获得初始运动矢量作为候选列表中的运动矢量预测器(可能舍入到第一分辨率),其被细化以获得第二运动分辨率。

[0061] 可以在编码器或解码器处使用运动估计来执行运动信息细化。与使用第一运动分辨率(例如,1/4像素)而没有运动分辨率细化的方法相比,所提出的实施例可以在没有传送细化运动信息的开销的情况下提高运动精度。与在解码器处使用第二运动分辨率(例如,1/8像素)而没有运动分辨率细化的方法相比,所提出的实施例可以减少传送运动细化信息的开销。应注意,根据本原理的解码器也可以执行运动搜索,并且因此可能比不需要运动搜索的解码器更复杂。

[0062] 图4图示了根据本原理的实施例的在解码器处执行运动矢量细化的示例性方法400。在该示例中,我们假设解码器可以将运动矢量分辨率从1/4像素细化到1/8像素。应注意,本原理可以应用于其他运动分辨率之间的细化,例如但不限于从整数像素到1/4像素、从1/4像素到1/6像素。

[0063] 对于要解码的块,解码器例如使用比特流中的标志,或者使用如上所述的现有方法,检查自适应运动矢量分辨率是否被启用(410)。如果自适应运动矢量分辨率被启用,则解码器例如使用HEVC的AMVP模式,以1/4像素分辨率生成MV预测器(MVP)(420)。如果预测器为1/8像素,则将其舍入为1/4像素。然后,解码器对1/4像素分辨率的MVD进行解码(430)。可以获得初始运动矢量(440),作为 $MV_0 = MVP + MVD$ 。然后,初始运动矢量(MV_0)可以被细化为1/8像素分辨率(450)并用于运动补偿预测。

[0064] 在420处,也可以以1/8像素生成MV预测器。MV预测器是以1/4像素还是1/8像素生成,应与编码器使用的一致。

[0065] 从由参考图片索引标识的参考图片中的运动矢量所移位的位置处的对应块的样本获得运动补偿预测块的样本。在运动矢量不是整数时,使用分数样本内插来生成预测样本。

[0066] HEVC支持1/4像素的运动矢量。让我们将运动矢量表示为 $MV = iMV + sMV$,其中 iMV 是 MV 的整数部分,并且 sMV 是1/4像素部分(第一运动分辨率), $sMV = p/4$, $p = 0, 1, 2, 3$,于是内插样本值可以计算为:

$$[0067] \quad lval[x] = \sum_{i=0}^{N-1} c[p][i] \times s[x + iMV - N/2 + i] \quad (2)$$

[0068] 其中 x 是像素定位, $c[p][i]$, $i=0, \dots, N-1$ 是对应于 $1/4$ 像素位置的滤波器系数, 并且 N 是滤波器抽头的数量。HEVC中用于 $1/4$ 像素内插的滤波器系数如下: $c[p=0][] = \{0, 0, 0, 64, 0, 0, 0, 0\}$, 对应于整数位置, 并且只有缩放, $c[p=1][] = \{-1, 4, -10, 58, 17, -5, 1, 0\}$, 对应于 $1/4$ 像素位置和7抽头滤波器, $c[p=2][] = \{-1, 4, -11, 40, 40, -11, 4, -1\}$, 对应于 $1/2$ 像素位置和8抽头滤波器, $c[p=3][] = \{0, 1, -5, 17, 58, -10, 4, -1\}$, 对应于 $1/4$ 像素位置和7抽头滤波器。

[0069] 在使用更高分辨率 $1/8$ 像素时, 使用 $1/8$ 像素内插滤波器来对 $1/8$ 像素部分 (包括 $sMV=p/8$, $p=0, 1, 2, 3, 4, 5, 6, 7, 8$) 进行内插。例如, 可以使用以下 $1/8$ 像素内插滤波器 ($N=4$):

[0070] $c[p=0][] = \{0, 64, 0, 0\}$,

[0071] $c[p=1][] = \{-2, 58, 10, -2\}$,

[0072] $c[p=2][] = \{-4, 54, 16, -2\}$,

[0073] $c[p=3][] = \{-6, 46, 28, -4\}$,

[0074] $c[p=4][] = \{-4, 36, 36, -4\}$,

[0075] $c[p=5][] = \{-4, 28, 46, -6\}$,

[0076] $c[p=6][] = \{-2, 16, 54, -4\}$,

[0077] $c[p=7][] = \{-2, 10, 58, -2\}$ 。

[0078] 运动细化处理可以被视为运动估计, 其具有与细化之前的运动分辨率相关的范围。例如, 在将运动矢量从 $1/4$ 像素的第一分辨率细化到 $1/8$ 像素的第二分辨率时, 运动搜索范围包括初始运动矢量周围的候选。更一般地, 用于运动细化的搜索候选可以在水平方向和垂直方向上分别为 $MVX_0 - \Delta MVX \sim MVX_0 + \Delta MVX$, $MVY_0 - \Delta MVY \sim MVY_0 + \Delta MVY$, 其中 MVX_0 和 MVY_0 分别为初始运动矢量 MV_0 的水平 and 垂直分量, 并且 ΔMVX 和 ΔMVY 分别定义水平和垂直方向上的搜索范围。

[0079] 例如, 水平方向上的细化候选可以是 $(MVX_0 - k \times \text{step2}, MVX_0 + k \times \text{step2})$, 其中 k 是整数并且定义为

[0080] $-\text{step1} < (k \times \text{step2}) < \text{step1}$, (3)

[0081] 其中 step1 是第一分辨率, 并且 step2 是第二分辨率。当 $\text{step1}=1/4$ 且 $\text{step2}=1/8$ 时, $-2 < k < 2$ 。图5图示了整数像素、半像素、四分之一像素和八分之一像素的像素位置。为了便于表示, 我们也可以将这些候选称为运动细化候选, 并且我们将通过利用运动细化候选的运动补偿构建的预测块表示为运动细化预测块, 并且将运动细化预测块内的样本表示为运动细化预测样本。

[0082] 在 (MVX_0, MVY_0) 对应于位置510时, 示例性搜索候选集合包括框520内的位置。搜索范围以及更一般的搜索候选集合可以与图5中所示的不同。应在编码器和解码器处使用相同的搜索候选集合。

[0083] 如果自适应运动矢量分辨率未被启用, 则解码器以用于经解码运动矢量的运动分辨率获得MVP和MVD, 并且在解码器处不执行运动细化。如图4中所示, 解码器可以以 $1/8$ 像素分辨率得到(460, 470) MVP和MVD二者, 并且以 $1/8$ 像素分辨率将运动矢量解码(480)为 $MV = MVP + MVD$ 。在变型中, 解码器可以以 $1/4$ 像素分辨率得到(460, 470) MVP和MVD, 并且以 $1/4$ 像素分辨率将运动矢量解码(480)为 $MV = MVP + MVD$ 。

[0084] 在下文中,我们进一步详细描述可以用于执行运动矢量细化(450)的若干个实施例。

[0085] 为了在解码器处细化运动矢量,我们使用图片信号通常平滑且连续的特性。因此,在解码器侧,如果运动矢量准确,则典型地,解码块应相对于相邻块连续。在一个实施例中,我们使用重建的相邻样本来细化MV。具体地,通过度量通过利用运动细化候选的运动补偿构建的预测块与先前重建的样本(例如,如图6A中所示的相邻上部和左侧块的样本)之间的不连续性,来评估如上所述的搜索候选集合。

[0086] 图6A图示了要解码的示例性PU(650)。当前PU上方或左侧的其他PU(610,620,630,640)已经被解码,并且可用于解码当前PU。具体地,相邻块(610,620,640)中的L形解码样本集合(670)可以用于细化当前PU(650)的运动矢量。应注意,PU可以具有与图6A中所示的不同的尺寸或形状,并且更大或更小的相邻重建样本集合可以用于细化当前PU的运动矢量。在本申请中,我们使用L形样本集合以用于运动细化。更一般地,不同的样本集合可以用于运动细化,并且细化可以应用于块。

[0087] 在一个实施例中,我们使用基于相邻块中的L形解码样本集合(670)(称为“相邻L形”)和当前预测块的L形样本集合(680)的不连续性。可以将不连续性度量为重建样本(n)与最接近的运动细化预测样本(p)之间的绝对差之和(SAD),如图6B中所示。数学上,细化的运动矢量差可以计算为:

[0088]

$$MVD_{\text{refine}}^* = \underset{MVD_{\text{refine}}}{\operatorname{argmin}} \sum_p |I_{\text{ref}}(p + MVP + MVD + MVD_{\text{refine}}) - I_{\text{rec}}(n)| \quad (4)$$

[0089] 其中p是PU的L形中的运动细化预测样本,其定位在 (x_p, y_p) , $\sum_p | \cdot |$ 是PU的L形集合上的总和, $MV_0 = MVP + MVD$ 是要细化的运动矢量,n是p旁边的L形中的重建样本,其定位在 (x_n, y_n) (例如,在p属于PU的左侧边界的情况下, $x_n = x_p - 1, y_n = y_p$,并且在p属于PU的上部边界的情况下, $x_n = x_p, y_n = y_p - 1$), $I_{\text{rec}}(n)$ 是当前图片的重建(或解码)样本值, $I_{\text{ref}}(p + MVP + MVD + MVD_{\text{refine}})$ 是在选择 MVD_{refined} 时的运动补偿预测值。

[0090] 在变型中,我们考虑残差和运动补偿预测块之和,并且在选择 MVD_{refine} 的情况下, $I_{\text{ref}}(p + MV_0 + MVD_{\text{refine}}) + \text{Res}(p)$ 将是样本p的重建值。然后,细化的运动矢量差可以计算为:

[0091]

$$MVD_{\text{refine}}^* = \underset{MVD_{\text{refine}}}{\operatorname{argmin}} \sum_p |I_{\text{ref}}(p + MVP + MVD + MVD_{\text{refine}}) + \text{Res}(p) - I_{\text{rec}}(n)| \quad (5)$$

[0092] 其中 $\text{Res}(p)$ 是样本p处的残差。

[0093] 在根据本原理将HEVC解码器修改为包括运动矢量细化时,可以在AMVP中将 MV_0 设置为 $MVD + MVP$,或者在合并模式中将 MV_0 设置为在候选列表中以信号通知的 MV_0 ,其中获得MVD和MVP,如HEVC指定的那样。通常, MV_0 是解码器获得的运动矢量,而不在解码器处执行运动搜索。

[0094] 运动细化可以被视为解码器处的运动搜索,具有从初始运动矢量周围选择的、较高运动分辨率的候选运动矢量集合。最佳运动矢量的选择可以是使某个准则最小化的最佳运动矢量,例如,如等式(4)或(5)中描述的不连续性度量。也就是说,在获得初始运动矢量 MV_0 之后,还执行运动搜索以细化初始运动矢量。解码器中的额外复杂性通常是小的,因为只需要检查初始运动矢量周围的小的搜索候选集合。

[0095] 数学上,导出的细化MV可以表达为:

$$[0096] \quad MV = MV_0 + MVD_{\text{refine}}^* \quad (6)$$

[0097] 然后,例如,使用如上所述的内插滤波器,将与参考图片中的细化MV相对应的块用作预测块以用于对PU进行解码。典型地,运动细化使得编码器能够对低分辨率的MVD进行编码,并且因此与对全分辨率的MVD进行编码相比,降低了编码运动信息的成本,同时解码器仍然可以恢复全分辨率的MV。

[0098] 在本申请中,我们使用SAD作为各种实施例中的差异度量。应注意,可以使用其他差异度量来代替SAD,例如但不限于平方误差之和 (SSE)。

[0099] 在另一实施例中,我们使用相邻像素处的梯度通常相似的属性,并且我们计算定位 (n) 和 (p) 处的局部梯度以度量不连续性。假设信号 (即,图片样本) 在空间上静止,可以将信号局部模型化为泰勒级数,舍项为线性项:

$$[0100] \quad I(x+d) = I(x) + g(x) \cdot d \quad (7)$$

[0101] 其中 $I(x)$ 是在定位 x 处的图片样本值, g 是利用局部梯度估计的 2×2 矩阵。

[0102] 使用等式 (7), 相邻重建样本 $R(x)$ 与运动细化预测块 $P(x)$ 之间的关系变为:

$$[0103] \quad R(n) = P(p) + g(p) \cdot (n-p) \quad (8)$$

[0104] 其中 $P(p) = I_{\text{ref}}(p + MV_0 + MVD_{\text{refine}})$, 并且 $R(n) = I_{\text{rec}}(n)$ 。在变型中,在计算 $P(p)$ 时可以包括残差。位置 $p = (x_p, y_p)$ 处的局部梯度可以表达为如图7A中所示:

$$[0105] \quad G11(p) = R(n) - P(p), \text{ 其中 } n = (x_p - 1, y_p - 1)$$

$$[0106] \quad G12(p) = R(n) - P(p), \text{ 其中 } n = (x_p - 1, y_p)$$

$$[0107] \quad G13(p) = R(n) - P(p), \text{ 其中 } n = (x_p - 1, y_p + 1)$$

[0108] 然后,局部梯度的绝对值之和可以用于计算不连续性,并且选择使不连续性最小化的运动矢量细化:

$$[0109] \quad MVD_{\text{refine}}^* = \underset{MVD_{\text{refine}}}{\operatorname{argmin}} \sum_p \{|G11(p)| + |G12(p)| + |G13(p)|\} \quad (9)$$

[0110] 位置 $p = (x_p, y_p)$ 处的二阶矩可以表达为如图7B中所示 (在 n 和 p 位于L形的垂直左侧边界处时):

$$[0111] \quad G21(p) = R(n2) - R(n) - G11, \text{ 其中 } n = (x_p - 1, y_p - 1) \text{ 且 } n2 = (x_p - 2, y_p - 2)$$

$$[0112] \quad G22(p) = R(n2) - R(n) - G12, \text{ 其中 } n = (x_p - 1, y_p) \text{ 且 } n2 = (x_p - 2, y_p)$$

$$[0113] \quad G23(p) = R(n2) - R(n) - G13, \text{ 其中 } n = (x_p - 1, y_p + 1) \text{ 且 } n2 = (x_p - 2, y_p + 2)$$

$$[0114] \quad G24(p) = R(n2) - R(n) - P(p) + P(p2), \text{ 其中 } n = (x_p - 1, y_p) \text{ 且 } n2 = (x_p - 1, y_p + 1) \text{ 以及 } p2 = (x_p, y_p + 1)。$$

[0115] 在样本 p 在PU的上部边界处时,可以类似地导出梯度。对于左上角,它可以如它在左侧边界或上部边界处那样被处理,或者可以被处理两次 (一次作为边界,一次作为上部)。

[0116] 然后,梯度的二阶矩的绝对值之和可以用于计算不连续性,并且选择使不连续性最小化的运动矢量细化:

$$[0117] \quad MVD_{\text{refine}}^* = \underset{MVD_{\text{refine}}}{\operatorname{argmin}} \sum_p \{|G21(p)| + |G22(p)| + |G23(p)| + |G24(p)|\} \quad (10)$$

[0118] 不连续性也可以计算为局部梯度和二阶矩的绝对值之和:

[0119]

$$\text{MVD}_{\text{refine}}^* = \underset{\text{MVD}_{\text{refine}}}{\operatorname{argmin}} \sum_p \{ |G11(p)| + |G12(p)| + |G13(p)| + |G21(p)| + |G22(p)| + |G23(p)| + |G24(p)| \} \quad (11)$$

[0120] 应注意,本原理也可以应用于其他形式的梯度计算。

[0121] 图8图示了根据本原理的实施例的在编码器处执行运动矢量细化的示例性方法800。方法800的输出可以用作方法400的输入比特流。在初始化步骤(805),编码器可以访问要编码的视频序列作为输入。另外,编码器可以将参数设置为初始值,例如,设置Best_flag=1,并且将Best_RDcost设置为大值。

[0122] 以1/8分辨率执行运动估计(810)以获得运动矢量(MV)。然后,编码器检查使用自适应运动细化对运动矢量进行编码是否更有效。根据步骤820-840,编码器检查具有运动细化的MV编码成本,例如,使用RD(速率失真)成本。根据步骤845-875,编码器检查没有运动细化的MV编码成本。然后,基于最终MV(Best_MV)执行运动补偿,并且可以计算残差(885)。将残差、最终MVD(Best_MVD)和自适应运动细化标志(Best_flag)编码(890)到比特流中。

[0123] 更具体地,在步骤815,例如,与对应解码器如何构建运动矢量预测器列表一致地使用来自AMVP的运动矢量预测器候选列表、1/8像素初始运动矢量预测器的舍入版本或相邻运动矢量的平均值,以1/4像素分辨率构建运动矢量预测器列表。降低分辨率可以使运动矢量预测器成为更“相关”的预测器(即,可以准确地预测被预测的运动矢量,使得可以正好传送索引以指示运动矢量预测器),并且使运动细化获得高分辨率部分(即,不太“相关”部分)。在一些情况下,MVP列表可以只包含一个MVP。对于MVP列表中的每个MVP,将运动矢量差(MVD)计算(820)为 $\text{MVD} = \text{MV} - \text{MVP}$,并且可以将初始运动矢量计算为 $\text{MV}_0 = \text{MVP} + \text{MVD}$ 。

[0124] 在变型中,可以在步骤815以1/8像素分辨率构建预测运动矢量候选列表,并且在步骤820将运动矢量差MVD舍入为1/4像素。注意, MV_0 可能因为应用于MVP或MVD的舍入而与MV不同。

[0125] 执行(825)细化以获得 $\text{MVD}_{\text{refine}}$,例如,如等式(4)或(5)中所述,并且细化的运动矢量为 $\text{MV}^* = \text{MV}_0 + \text{MVD}_{\text{refine}}$ 。然后,可以基于 MV^* 来计算(827)残差。在步骤830,可以使用RD(速率失真)成本来估计调整的运动矢量(MV^*)的编码成本。在步骤835,编码器检查当前调整的运动矢量是否具有比当前Best_RDcost小的RD成本。如果是,则将参数Best_RDcost、Best_MV、Best_MVD设置为当前RD成本、当前调整的运动矢量和当前MVD,并且也可以存储一些其他相关编码信息。

[0126] 在步骤845,例如使用来自AMVP的运动矢量预测器候选列表,以1/8像素分辨率构建运动矢量预测器列表。步骤845类似于步骤815,除了编码器不将运动分辨率调整为1/4像素。对于MVP列表中的每个MVP,基于1/8像素分辨率的MVP来计算(850)MVD。可以针对运动矢量 $\text{MV}_1 = \text{MVP} + \text{MVD}$ 来计算(852)残差,并且可以在步骤855使用RD成本来估计运动矢量 MV_1 的编码成本。在步骤865,编码器检查当前运动矢量是否具有比当前Best_RDcost小的RD成本。如果是,则将参数Best_RDcost、Best_MV和Best_MVD设置为当前RD成本、当前运动矢量和当前MVD,并且将自适应运动细化标志设置为假(0)。也可以存储其他相关编码信息。

[0127] 在MVD为0时,例如,在使用合并模式时,不需要步骤820和850,并且在步骤840和875,不需要计算Best_MVD,并且在步骤890,不需要编码Best_MVD。

[0128] 运动细化可以应用于与运动矢量预测器相对应的所有运动矢量,或者可以应用于子集。例如,运动细化仅用于AMVP运动矢量预测器,或者仅用于合并模式。

[0129] 在方法800中,编码器基于编码成本来判定是否使用自适应运动细化,并且通过比特流中的标志以信号通知选择。在其他实施例中,编码器可以在不检查编码成本的情况下基于视频特性来判定是否使用自适应运动细化,例如,如下面进一步描述的或者在Lakshman和Chen中描述的。因此,编码器可能仅需要执行方法800的部分(例如,具有运动细化的部分,或者没有运动细化的部分)。

[0130] 注意,根据方法800,可能需要发送附加语法元素Best_flag。然而,发送Best_flag可以仍然比发送1/8像素分辨率的MVD更有效,因为运动矢量的高分辨率部分通常是随机的且编码昂贵。

[0131] 通过细化运动矢量,编码器可以达到1/8像素运动分辨率,而以1/4像素运动分辨率在比特流中传送运动矢量差MVD。因此,MVD可以需要较少的比特来编码。例如,如表1中所示,来自运动估计(810)的当前块的运动矢量(MV)为3.625,运动矢量预测器列表(815)包括{3.0,...}。对于运动矢量预测器 $MVP_0=3.0$,在未启用运动细化时,MVD(850)是 $MV-MVP=0.625$ (以1/8像素编码)。另一方面,在启用运动细化时,将MVD舍入到0.5(1/4像素),并且初始运动矢量(820) $MV_0=MVP+MVD=3.5$ 。然后,将运动矢量 MV_0 细化(825)为 $MV^*=3.625$ 。在该示例中,细化运动与从运动估计获得的MV相同,并且以1/4像素传送MVD,因此可以比不使用运动细化(即,以1/8像素传送MVD)时需要更少的比特。因此,编码器可能选择启用运动细化。

[0132] 表1

[0133]

	分辨率	示例
MVP列表(左侧、上部)	1/4	{3.0...}
MVP	1/4	3.0
MVD	1/4	0.5
MV^*	1/8	3.625
来自ME的MV	1/8	3.625
MC的MV	1/8	3.625

[0134] 在以上示例中,细化的运动矢量与来自运动估计的运动矢量相同。应注意,细化的运动矢量可以与从运动估计获得的运动矢量不同。

[0135] MVP细化

[0136] 在上文中,我们讨论了对基于运动矢量预测器MVP和运动矢量差MVD生成的初始运动矢量的细化。在另一实施例中,可以关于运动矢量预测器(不包括MVD)执行细化。返回参考图6,在获得初始运动矢量预测器(MVP_0)之后,例如,使用如HEVC指定的AMVP,可以使用重建的L形(670)来细化运动矢量预测器。

[0137] 具体地,解码器可以形成与初始运动矢量预测器周围的运动矢量预测器相对应的MC L形(680),例如, $MVP_0 + \Delta MV$, $\Delta MV = \{(-\frac{1}{8}, 0), \dots, (0, \frac{1}{8})\}$ 。然后,解码器可以比较重建的L形(670)和不同的MC L形(680)之间的差异,例如,如上所述的不连续性或梯度,并且选择产生最小差异的运动细化(ΔMV^*)作为对初始运动矢量预测器的细化,即,可以将细化的

运动矢量预测器计算为 $MVP^* = MVP_0 + \Delta MV^*$ 。注意,重建的L形(670)和MCL形(680)都基于解码样本,使得可以在编码器侧和解码器侧二者处执行相同的处理。

[0138] 通常,运动矢量预测器的运动细化可以被视为具有搜索范围的运动搜索,该搜索范围包括初始运动矢量预测器周围的全运动分辨率的几个子样本位移。例如,如图5中所示,可以使用520内的搜索候选来细化510处的运动矢量预测器 MVP_0 。类似于细化运动矢量(具有MVD),可以使用与图5中所示的不同的搜索候选集合。

[0139] 在变型中,可以在初始运动矢量预测器的舍入版本($\text{round}(MVP_0)$)上执行运动细化,并且运动细化的搜索范围包括舍入运动矢量预测器周围的全运动分辨率的子样本位移。例如,在细化的运动分辨率为1/8像素时, MVP_0 的舍入版本可以是整数像素、半像素或四分之一像素分辨率。这是因为实际运动矢量更可能集中在较低分辨率的运动矢量周围,并且初始运动矢量预测器的舍入版本可以为搜索提供更好的起点。

[0140] 在另一变型中,可以关于在MVP候选列表中使用的相邻运动矢量的平均值来执行运动细化。例如,运动细化可以使用左侧和上方运动矢量的平均值 $(MV_{\text{left}} + MV_{\text{above}})/2$ 作为初始运动矢量预测器,并且然后执行初始运动矢量预测器周围的运动搜索。我们观察到,通常在几乎或相对均匀的运动区域中选择运动细化。通过对与当前块相邻的运动矢量求平均,我们可以为搜索提供更好的起点。在选择运动细化时,即,在运动场是均匀的时,可以将预测器候选列表缩减为仅使用列表中的一个(左侧或上方)候选,并且因此减少可能的候选的数量。这可以提高压缩效率,因为不需要在比特流中编码或传送AMVP列表的索引。

[0141] 图9图示了根据本原理的实施例的在解码器处执行运动矢量预测器细化的示例性方法900。在该示例中,我们假设解码器可以将运动矢量预测器分辨率从1/4像素细化到1/8像素。

[0142] 对于块,解码器例如使用比特流中的标志,或者使用如上所述的现有方法,检查(910)自适应运动矢量分辨率是否被启用。如果自适应运动矢量分辨率被启用,则解码器以1/4像素分辨率生成MV预测器(MVP)(920)。可以例如但不限于使用HEVC的AMVP模式、使用相邻运动矢量的平均值或者使用运动矢量预测器的舍入版本,来生成初始运动矢量预测器。然后,例如,使用上述实施例,可以将运动矢量预测器细化为1/8像素分辨率(930)。解码器以1/8像素分辨率对MVD进行解码(940)。然后,运动矢量可以获得(950)为 $MV = MVP^* + MVD$ 并用于运动补偿预测。

[0143] 如果自适应运动矢量分辨率未被启用,则解码器以用于经解码运动矢量的运动分辨率获得MVP和MVD,并且在解码器处不执行运动细化。如图9中所示,解码器可以以1/8像素分辨率得到(960,970) MVP和MVD二者,并且以1/8像素分辨率将运动矢量解码(980)为 $MV = MVP + MVD$ 。在变型中,解码器可以以1/4像素分辨率得到(960,970) MVP和MVD二者,并且以1/4像素分辨率将运动矢量解码(980)为 $MV = MVP + MVD$ 。

[0144] 图10图示了根据本原理的实施例的在编码器处执行运动矢量预测器细化的示例性方法1000。方法1000的输出可以用作方法900的输入比特流。在初始化步骤(1005),编码器可以访问要编码视频序列作为输入。另外,编码器可以将参数设置为初始值,例如,设置 $\text{Best_flag} = 1$,并且将 Best_RDCost 设置为大值。

[0145] 以1/8分辨率执行运动估计(1010)以获得运动矢量MV。然后,编码器检查使用自适应运动细化对运动矢量进行编码是否更有效。根据步骤1020-1040,编码器检查具有运动细

化的MV编码成本,例如,使用RD(速率失真)成本。根据步骤1045-1075,编码器检查没有运动细化的MV编码成本。然后,基于最终MV (Best_MV) 执行运动补偿,并且可以计算残差(1085)。将残差、最终MVD (Best_MVD) 和自适应运动细化标志 (Best_flag) 编码(1090)到比特流中。

[0146] 更具体地,在步骤1015,例如,与对应解码器如何构建运动矢量预测器列表一致地使用来自AMVP的运动矢量预测器候选列表、初始运动矢量预测器的舍入版本(例如,舍入到所需分辨率的最接近整数)或相邻运动矢量的平均值,以1/4像素分辨率构建运动矢量预测器列表。在一些情况下,MVP列表可以只包含一个MVP。在步骤1015,也可以以1/8像素分辨率构建运动矢量预测器列表。然而,使用较低分辨率(1/4像素)的一个优点在于,它可以避免平坦区域中的漂移,在平坦区域中RD优化不能在1/8像素或1/4像素分辨率的MV之间进行区分。使用降低的分辨率允许MV预测器的一种平滑,其可以减少MV噪声。此外,如果用于细化的搜索窗口的大小相对小(针对1/8细化为 $\pm 1/4$),则搜索窗口的起始/中心MV的精度可能改变结果。

[0147] 对于MVP列表中的每个MVP,执行MVP细化(1020),例如,如以上针对解码器所描述的那样。基于细化的MVP (MVP*) 来计算MVD(1025),并且也可以计算可以用于运动补偿的调整的运动矢量(MV*) ($MV^* = MVD + MVP^*$)。在步骤1030,可以使用RD成本来估计调整的运动矢量(MV*)的编码成本。在步骤1035,编码器检查当前调整的运动矢量是否具有比当前Best_RDcost小的RD成本。如果是,则将参数Best_RDcost、Best_MV和Best_MVD设置(1040)为当前RD成本、当前调整的运动矢量和当前MVD,并且Best_flag保持为真(1)。

[0148] 在步骤1045,例如,使用来自AMVP的运动矢量预测器候选列表,以1/8像素分辨率构建运动矢量预测器列表。对于MVP列表中的每个MVP,以1/8像素分辨率基于MVP来计算(1050)MVD。在步骤1055,可以使用RD成本来估计运动矢量(MV)的编码成本。在步骤1065,编码器检查当前运动矢量是否具有比当前Best_RDcost小的RD成本。如果是,则将参数Best_RDcost、Best_MV和Best_MVD设置(1075)为当前RD成本、当前运动矢量和当前MVD,并且将自适应运动细化标志 (Best_flag) 设置为假(0)。

[0149] 在方法1000中,编码器基于编码成本来判定是否使用自适应运动细化,并且通过比特流中的标志以信号通知选择。在其他实施例中,编码器可以在不检查编码成本的情况下基于视频特性来判定是否使用自适应运动细化,例如,如下面进一步描述的或者在Lakshman和Chen中描述的。因此,编码器可能仅需要执行方法1000的部分(例如,具有运动细化的部分,或者没有运动细化的部分)。

[0150] 通过细化运动矢量预测器,MV预测器可以更准确。因此,MVD可以在统计上具有较低的值,并且然后编码成本可被降低。例如,如表2中所示,来自运动估计(1010)的当前块的运动矢量(MV)为3.625,运动矢量预测器列表(1015)包括{3.0,...}。对于运动矢量预测器 $MVP_0 = 3.0$,通过运动细化将其细化(1020)为 $MVP^* = 3.5$ 。随后,运动矢量差为 $MVD = MV - MVP^* = 0.125$ 。与不使用运动细化的运动矢量差 $MV - MVP_0 = 0.625$ 相比,要编码的MVD更小并且可以需要更少的比特来编码。在该示例中,使用运动细化可以提高压缩效率,并且因此,编码器可能选择启用自适应运动细化。用于运动补偿的运动矢量为 $MVP^* + MVD = 3.625$ 。

[0151] 表2

[0152]

	分辨率	示例
--	-----	----

来自ME的MV	1/8	3.625
MVP列表(左侧、上部)	1/4	{3.0...}
MVP	1/4	3.0
MVP-细化的(MVP*)	1/8	3.5
MVD	1/8	0.125
MC的MV	1/8	3.625

[0153] 在MVD为0时,例如,在使用合并模式时,不需要步骤1025和1050,并且在步骤1040和1075,不需要计算Best_MVD,并且在步骤1090,不需要编码Best_MVD。注意,在MVD为0时,方法1000可以变得与方法800相同。运动细化可以应用于与运动矢量预测器相对应的所有运动矢量,或者可以应用于子集。例如,运动细化仅用于AMVP运动矢量预测器,或者仅用于合并模式。

[0154] 自动运动细化激活

[0155] 在以上实施例中,我们描述了标志(例如,如图8或图10中所示的Best_flag)可以用于指示是否启用自适应运动矢量分辨率。在下文中,我们描述自动激活或停用运动矢量分辨率细化的各种实施例。因此,编码器可以在没有显式标志的情况下指示是否启用自适应运动矢量分辨率,并且解码器也可以在不参考显式标志的情况下判定是否自适应运动矢量分辨率。

[0156] 在一个实施例中,我们提出基于以下准则中的一个或多个来自动激活自适应运动细化:

[0157] -运动相似性:当前块的运动类似于周围块的运动,使得编码器或解码器可以使用周围块的运动矢量以用于运动细化。

[0158] -纹理性(或纹理级别):当前块和周围块包含可以允许稳健的子像素运动细化的某个纹理。

[0159] -运动矢量差(MVD):MVD在编码器和解码器二者处可用,并且可以用于确定是使用自动自适应运动细化还是显式地以信号通知自适应运动细化。

[0160] 可以使用以下条件中的一个或多个来度量运动相似性准则:

[0161] 1) 存在 MVP_{left} 和 MVP_{above} (可选地, $MVP_{above-left}$),即,当前块周围存在运动场。更一般地,存在具有运动矢量的一个或多个相邻块。

[0162] 2) $|MVP_{left} - MVP_{current}| < T2$, $|MVP_{above} - MVP_{current}| < T2$, 以及 $|MVP_{leftabove} - MVP_{current}| < T2$ 。该条件也可以是这三个子条件的不同逻辑组合: $|MVP_{left} - MVP_{current}| < T2$, $|MVP_{above} - MVP_{current}| < T2$, 以及 $|MVP_{leftabove} - MVP_{current}| < T2$ 。也就是说,周围块的运动矢量是相似的,并且当前块周围的运动场有些均匀。这里我们使用相同的T2来检查 MVP_{left} 与 $MVP_{current}$ 之间、 MVP_{above} 与 $MVP_{current}$ 之间以及 $MVP_{leftabove}$ 与 $MVP_{current}$ 之间的差异。应注意,不同的阈值可以用于这些差异。

[0163] 可以使用以下条件中的一个或多个来度量运动纹理性准则:

[0164] 1) $Texture(L) > T3$, 其中L是用于执行运动细化的相邻区域,例如,如图6A中所示的670,并且 $texture(X)$ 是纹理化的度量,例如,X的亮度方差。也就是说,相邻区域具有某个纹理。

[0165] 2) $Err(mc(L, MVP, Iref), L) < Err(mc(L, MVP + MVr, Iref), L) + T4$, 其中MVP是与在比

特流中以信号通知的MVD相对应的所选运动矢量预测器,使得编码器和解码器二者可以使用相同的MVP来检查该条件,Err (X,Y) 是一组像素X和一组像素Y之间的误差的度量,例如,SAD、HAD或SSE,mc (X,v,I) 是使用参考图像I中的运动矢量v的像素组X的运动补偿,Iref是与运动矢量预测器MVP相关联的参考图像,T4是关于误差相似性的阈值,MVP+Mvr是细化之后的运动矢量预测器。如前所述,MVP可以处于较低的运动分辨率,例如,是整数舍入的。

[0166] 自动运动细化激活可以基于运动相似性或纹理性准则二者下的所有条件,或者也可以使用条件的子集。是否以信号通知运动细化激活或停用可以基于运动矢量差(MVD)。在 $|MVD| \leq T1$ 时,其中MVD是比特流中指示的运动矢量差并且T1是运动阈值,在典型示例中, $T1 = \sqrt{2}/2$,我们可以在没有显式信令的情况下选择激活运动矢量细化。也就是说,在运动矢量差小并且当前运动矢量接近当前运动矢量预测器时,编码器和解码器可以自动激活运动细化。另外,编码器和解码器二者可以访问MVD,并且可以以相同的方式使用MVD来确定运动细化激活,以便使编码器和解码器同步。另一方面,在 $|MVD|$ 变大时,编码器和解码器可以在没有显式信令的情况下自动停用运动细化。

[0167] 在另一示例中,在使用时间候选来构建运动矢量预测器列表时,也就是说,在MVP_{left}和MVP_{above}不共存或者相同时,不使用运动相似性准则中的条件1)和2)。编码器可以选择是否使用时间候选以用于自动细化。

[0168] 图11图示了根据本原理的实施例的在解码器处利用自动运动细化激活或停用来执行运动矢量预测器细化的示例性方法1100。

[0169] 对于要解码的块,解码器生成(1110) MV预测器(MVP_{current}),并且对MV差(MVD)进行解码(1120)。在 $|MVD| \leq T1$ (1125) 为真时,解码器检查其他条件以查看是否要激活运动细化。否则,如果 $|MVD| > T1$,则解码器检查是否 $|MVD| \leq T1 + S * \sqrt{2}/2$ 。在一个示例中, $S = 1/2$,指示用于细化的运动搜索在 $[-1/2, 1/2]$ 中。如果 $|MVD| \leq T1 + S * \sqrt{2}/2$ (1130),则解码器基于从比特流解码(1140)的标志来判定是否使用运动细化。否则,如果 $|MVD| > T1 + S * \sqrt{2}/2$,则解码器不执行运动细化(即,自动停用运动细化),并且基于MVP和MVD对运动矢量进行解码(1180),即, $MV = MVP + MVD$ 。

[0170] 更具体地,解码器检查(1155)是否 $|MVP_{left} - MVP_{current}| < T2$ 且 $|MVP_{above} - MVP_{current}| < T2$ 。如果是,则解码器检查(1165)是否Texture (L) > T3。如果满足1155和1165处的两个条件,则解码器例如使用上述各种实施例来执行运动矢量预测器细化,并且基于细化的运动矢量预测器(MVP*)和MVD对运动矢量进行解码(1170),即, $MV = MVP* + MVD$ 。这里,在没有标志的情况下自动激活运动细化。

[0171] 在 $T1 < |MVD| \leq T1 + S * \sqrt{2}/2$ 时,解码器对指示是否可以使用运动细化的标志(例如,Refine_flag)进行解码(1140)。如果标志为真(1145),则解码器继续到步骤1155。否则,如果标志为假(1145),则基于MVP和MVD获得(1180)运动矢量。

[0172] 在图11中,解码器在一些条件下自动停用或激活运动细化,并且在其他条件下依赖于标志来停用或激活运动细化。对应的编码器使用相同的条件来确保编码器与解码器之间的同步。

[0173] 在MVD为0时,例如,在使用合并模式时,步骤1125处的条件总是为真并且可以被移

除,并且不需要步骤1120、1130、1140、1145和1180。替代地,在合并模式中,可以禁用运动细化以便避免在合并推导的运动矢量预测器上添加噪声。

[0174] 与图11中所示的不同,方法1100也可以跳过步骤1130-1145。也就是说,在 $|MVD| > T1$ 时,解码器执行步骤1180而没有运动细化。也可以实现其他变型,例如但不限于跳过步骤1155和/或步骤1165,在步骤1155检查两个条件之一。

[0175] 图12图示了根据本原理的实施例的在编码器处执行自动运动细化的示例性方法1200。方法1200的输出可以用作方法1100的输入比特流。在初始化步骤(1205),编码器可以访问要编码的视频序列作为输入。另外,编码器可以将参数设置为初始值,例如,将Best_RDcost设置为大值。

[0176] 以1/8像素分辨率执行运动估计(1210)以获得运动矢量MV。在步骤1220,例如,使用来自AMVP的运动矢量预测器候选列表来构建运动矢量预测器列表。可以与对应解码器如何构建运动矢量预测器列表一致地使用初始运动矢量预测器的舍入版本(例如,舍入到所需分辨率的最接近整数)或相邻运动矢量的平均值。在一些情况下,MVP列表可以只包含一个MVP。在步骤1220,也可以以其他分辨率来构建运动矢量预测器列表。

[0177] 在步骤1225,对于特定MVP,编码器然后检查使用自适应运动细化对运动矢量进行编码是否更有效,并且将Refine_flag设置为0或1。在运动细化更有效时,将Refine_flag设置为1,否则为0。在步骤1225也计算对应的MVD。

[0178] 图13图示了根据本原理的实施例的确定是否对特定MVP使用自适应运动细化的示例性方法1300。方法1300可以用于实现方法1200的步骤1225。

[0179] 具体地,编码器检查(1305)是否 $|MVP_{left} - MVP_{current}| < T2$ 、 $|MVP_{above} - MVP_{current}| < T2$ 且Texture(L) $> T3$ 。如果满足1305处的条件,则编码器例如使用上述各种实施例来执行运动矢量预测器细化(1330),并且基于细化的运动矢量预测器(MVP*)来计算(1340)运动矢量差,即, $MVD = MV - MVP^*$ 。在使用运动细化时,估计(1350)RD成本(RDCost1)。编码器也在没有细化的运动矢量预测器的情况下计算(1360)运动矢量差,即, $MVD = MV - MVP$ 。在不使用运动细化时,估计(1370)RD成本(RDCost2)。对具有和没有运动细化的RD成本进行比较(1375)。如果RDCost1较小,则将Refine_flag设置(1380)为1,并且将MVD设置为 $MV - MVP^*$ 。否则,将Refine_flag设置(1390)为0,并且将MVD设置为 $MV - MVP$ 。

[0180] 如果不满足1305处的条件,则基于运动矢量预测器(MVP)来计算(1310)运动矢量差,即, $MVD = MV - MVP$,并且将Refine_flag设置为(1320)为0。

[0181] 在步骤1225确定MVD和Refine_flag之后,在步骤1230,编码器检查是否 $|MVD| \leq T1$ 。如果是,则编码器进一步检查(1235)Refine_flag是否等于1。如果Refine_flag被设置为1,则编码器估计(1270)RD成本。如果RD成本小于当前Best_RDcost(1275),则将参数Best_RDcost和Best_MVD设置(1280)为当前RD成本和当前MVD,并且将Write_flag设置为no,其指示不使用显式信令来指示自适应运动细化。也就是说,如果选择当前MVP以用于编码,则将自动激活运动细化而无需发送标志。

[0182] 在 $|MVD| > T1$ (1230)时,编码器估计(1240)RD成本。如果RD成本小于Best_RDcost(1245),则将参数Best_RDcost和Best_MVD设置(1250)为当前RD成本和当前MVD。然后,编码器检查(1255)是否 $|MVD| \leq T1 + S * \sqrt{2}/2$ 。如果是,则将Write_flag设置为yes。这里,如果选择与条件分支1260相对应的运动矢量预测器,则将基于Refine_flag显式地以信

号通知是否激活运动细化。

[0183] 否则,如果不满足1255处的条件,则将Write_flag设置(1265)为no。也就是说,如果选择与条件分支1265相对应的运动矢量预测器,则将自动停用运动细化而无需发送标志。

[0184] 编码器检查(1285)是否到达MVP列表的末尾。如果是,则编码器对Best_MVD、所选MVP的索引以及对应的残差(如果有的话)进行编码。如果Write_flag被设置为yes,则也对Refine_flag进行编码。否则,如果不满足1285处的条件,则控制返回到步骤1225。如果选择合并模式,则在步骤1290不需要对Best_MVD进行编码。对应于方法1100,方法1200也可以与图12中所示的不同,例如,可以跳过步骤1240-1260。

[0185] 在上文中,关于运动矢量预测器讨论了运动细化的自动激活或停用。应注意,自动激活或停用可以应用于其他类型的运动信息,例如但不限于已经包括运动矢量差的运动矢量。

[0186] 在本申请中使用各种数值,例如,基于阈值确定运动相似性或纹理性。应注意,特定值用于示例性目的,并且本原理不限于这些特定值。

[0187] 在上文中,关于HEVC标准描述了各种实施例。例如,如上所述的各种运动细化或自动激活方法可以用于修改如图1中所示的HEVC编码器的运动估计模块(175),或者如图3中所示的HEVC解码器的运动补偿模块(375)。然而,本原理不限于HEVC,并且可以应用于其他标准、建议书及其扩展。

[0188] 在上文中,我们讨论了从1/4像素到1/8像素的运动细化。应注意,本原理可以应用于其他运动分辨率之间的细化,例如但不限于从整数像素到1/4像素,从1/4像素到1/6像素。

[0189] 图14图示了示例性系统的框图,其中可以实现本原理的示例性实施例的各个方面。系统1400可以实施为包括下面描述的各种组件并且被配置为执行上述处理的设备。这样的设备的示例包括但不限于个人计算机、膝上型计算机、智能电话、平板计算机、数字多媒体机顶盒、数字电视接收器、个人视频记录系统、连接的家用电器和服务。系统1400可以通信地耦接到其他类似系统,以及经由如图14中所示并且如本领域技术人员已知的通信信道耦接到显示器,以实现上述示例性视频系统。

[0190] 系统1400可以包括至少一个处理器1410,其被配置为执行加载在其中的指令,以用于实现如上所述的各种处理。处理器1410可以包括嵌入式存储器、输入输出接口和本领域已知的各种其他电路。系统1400也可以包括至少一个存储器1420(例如,易失性存储器设备,非易失性存储器设备)。系统1400可以另外包括储存设备1440,其可以包括非易失性存储器,包括但不限于EEPROM、ROM、PROM、RAM、DRAM、SRAM、闪存存储器、磁盘驱动器和/或光盘驱动器。作为非限制性示例,储存设备1440可以包括内部储存设备、附接储存设备和/或网络可访问储存设备。系统1400也可以包括编码器/解码器模块1430,其被配置为处理数据以提供经编码视频或经解码视频。

[0191] 编码器/解码器模块1430表示可以包括在设备中以执行编码和/或解码功能的模块。如已知的那样,设备可以包括编码和解码模块中的一个或两个。另外,编码器/解码器模块1430可以实现为系统1400的单独元件,或者可以作为硬件和软件的组合并入处理器1410内,如本领域技术人员已知的那样。

[0192] 要加载到处理器1410上以执行上文描述的各种处理的程序代码可以存储在储存设备1340中,并且随后加载到存储器1420上以供处理器1410执行。根据本原理的示例性实施例,处理器1410、存储器1420、储存设备1440和编码器/解码器模块1430中的一个或多个可以在执行上文讨论的处理期间存储各种项中的一个或多个,包括但不限于输入视频、比特流、等式、公式、矩阵、变量、运算和运算逻辑。

[0193] 系统1400也可以包括通信接口1450,其使得能够经由通信信道1460与其他设备进行通信。通信接口1450可以包括但不限于被配置为从通信信道1460传送和接收数据的收发器。通信接口可以包括但不限于调制解调器或网卡,并且通信信道可以实现在有线和/或无线介质中。系统1400的各种组件可以使用各种合适的连接(包括但不限于内部总线、布线和印刷电路板)而连接或通信地耦接在一起。

[0194] 可以通过由处理器1410实现的计算机软件、通过硬件或者通过硬件和软件的组合来执行根据本原理的示例性实施例。作为非限制性示例,可以通过一个或多个集成电路来实现根据本原理的示例性实施例。存储器1420可以是适合于技术环境的任何类型,并且可以使用任何适当的数据储存技术来实现,诸如作为非限制性示例的光学存储器设备、磁存储器设备、基于半导体的存储器设备、固定存储器和可移动存储器。处理器1410可以是适合于技术环境的任何类型,并且可以涵盖作为非限制性示例的微处理器、通用计算机、专用计算机和基于多核架构的处理器中的一个或多个。

[0195] 在此描述的实现方式可以例如以方法或处理、装置、软件程序、数据流或信号来实现。即使仅在单一形式的实现方式的背景下进行了讨论(例如,仅作为方法进行了讨论),但是所讨论的特征的实现方式也可以以其他形式(例如,装置或程序)来实现。装置例如可以实现在适当的硬件、软件和固件中。方法例如可以实现在诸如例如处理器(其通常指代处理设备,包括例如计算机、微处理器、集成电路或可编程逻辑器件)之类的装置中。处理器还包括通信设备,诸如例如计算机、蜂窝电话、便携式/个人数字助理(“PDA”),以及便于最终用户之间的信息通信的其他设备。

[0196] 对本原理的“一个实施例”或“实施例”或“一个实现方式”或“实现方式”的引用以及其他变型意指结合实施例描述的特定特征、结构、特性等包括在本原理的至少一个实施例中。因此,在整个说明书中出现在各处的短语“在一个实施例中”或“在实施例中”或“在一个实现方式中”或“在实现方式中”以及任何其他变型的出现不一定都指代相同的实施例。

[0197] 另外,本申请或其权利要求可以涉及“确定”各种信息。确定信息可以包括例如估计信息、计算信息、预测信息或从存储器取回信息中的一个或多个。

[0198] 此外,本申请或其权利要求可以涉及“访问”各种信息。访问信息可以包括例如接收信息,(例如,从存储器)取回信息、存储信息、处理信息、传送信息、移动信息、复制信息、擦除信息、计算信息、确定信息、预测信息或估计信息中的一个或多个。

[0199] 另外,本申请或其权利要求可以涉及“接收”各种信息。与“访问”一样,接收旨在为广义术语。接收信息可以包括例如访问信息或(例如,从存储器)取回信息中的一个或多个。此外,在诸如例如存储信息、处理信息、传送信息、移动信息、复制信息、擦除信息、计算信息、确定信息、预测信息或估计信息的操作期间,典型地以某种方式涉及“接收”。

[0200] 如对于本领域技术人员将明显的那样,实现方式可以产生被格式化为携带例如可

以被存储或传送的信息的各种信号。信息可以包括例如用于执行方法的指令或者由所描述的实现方式之一产生的数据。例如,可以将信号格式化为携带所描述的实施例的比特流。可以将这样的信号格式化为例如电磁波(例如,使用频谱的射频部分)或者基带信号。格式化可以包括例如对数据流进行编码以及以编码的数据流来调制载波。信号携带的信息例如可以是模拟或数字信息。如已知的那样,可以通过各种不同的有线或无线链路来传送信号。信号可以存储在处理器可读介质上。

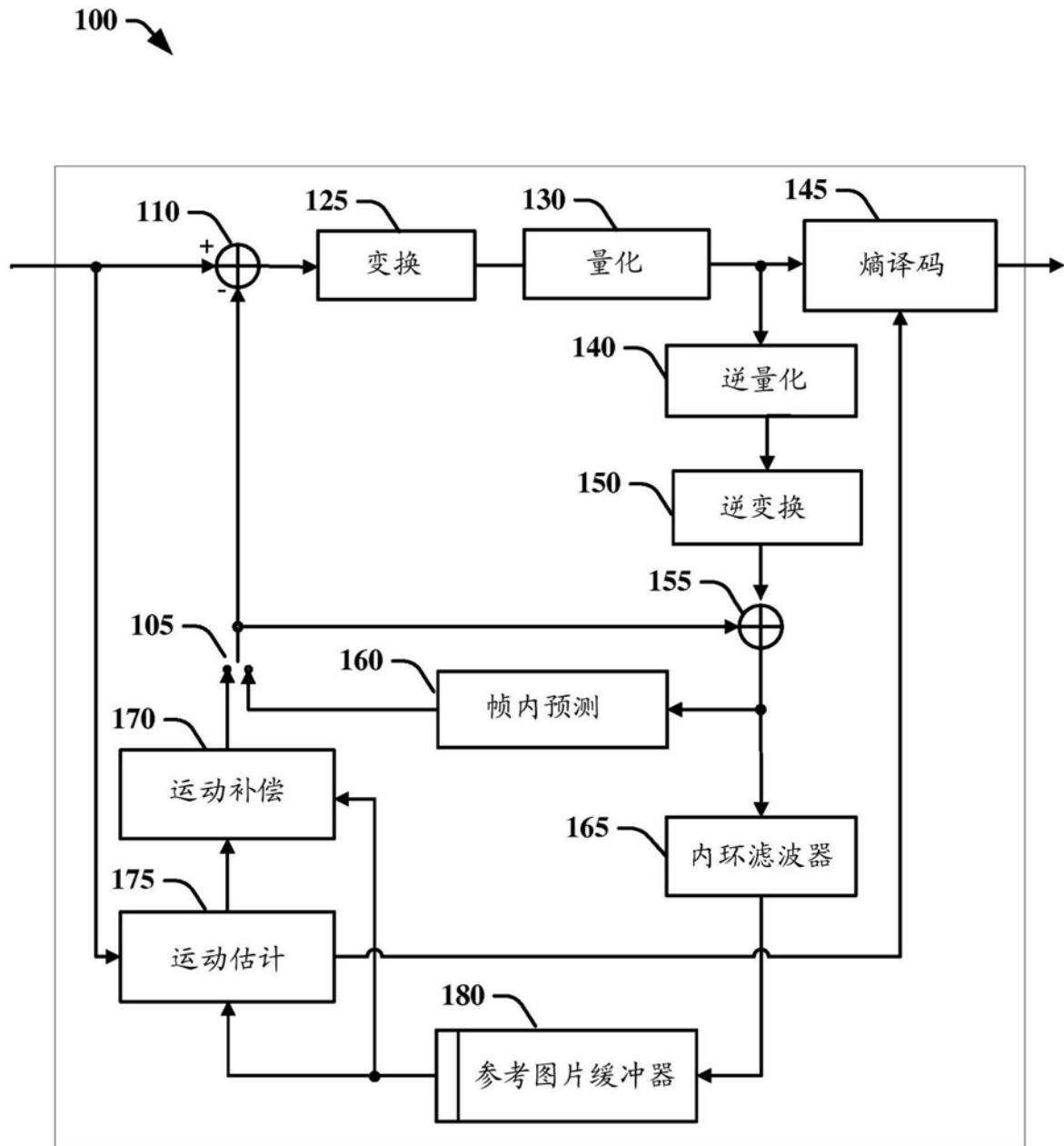


图1

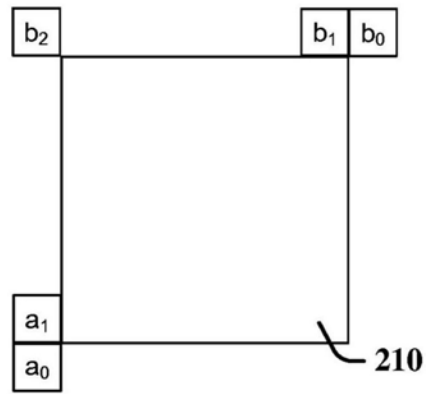


图2A

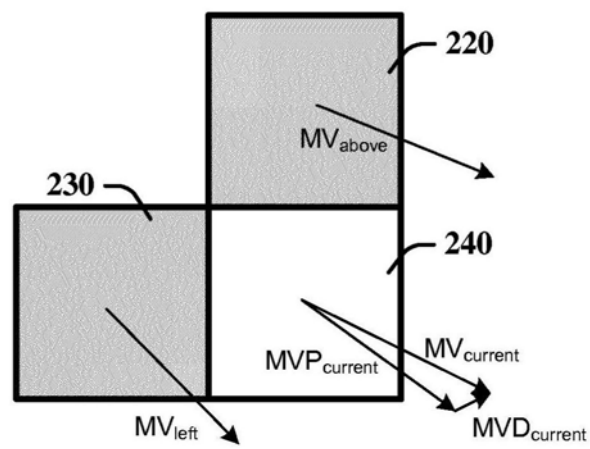


图2B

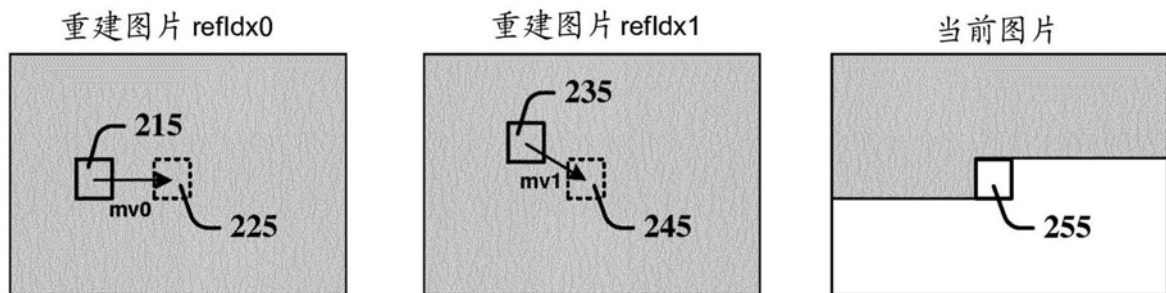


图2C

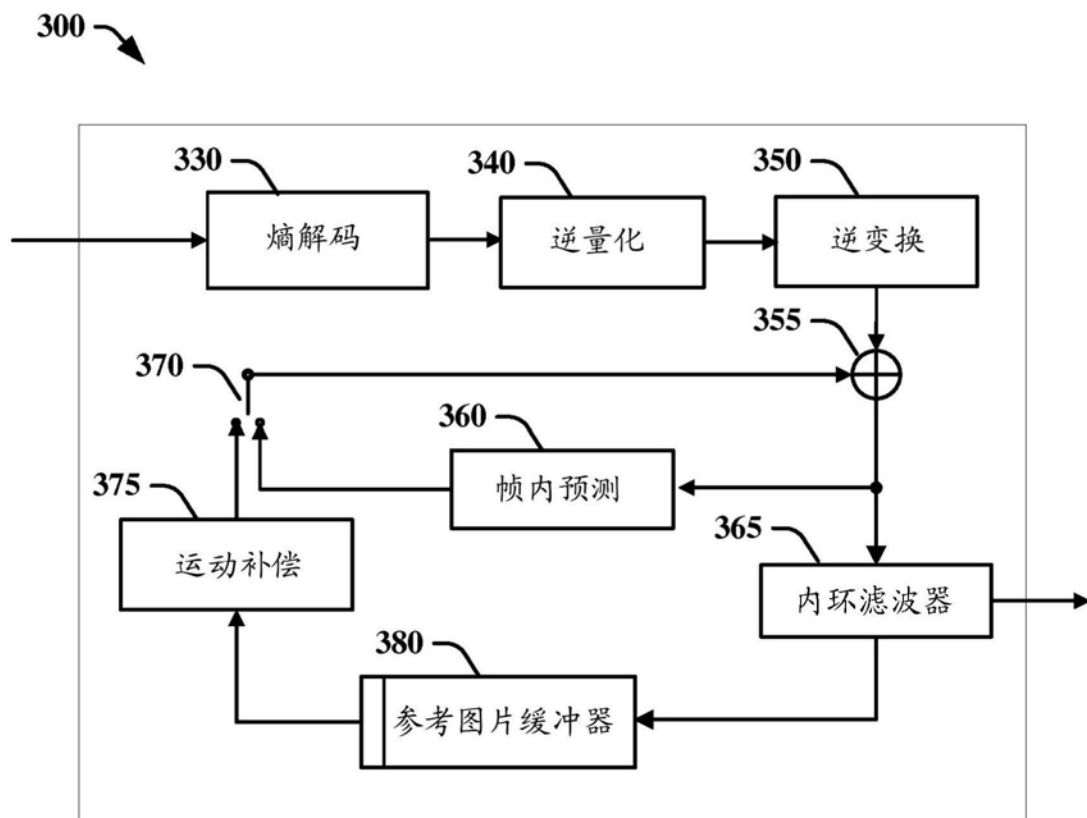


图3

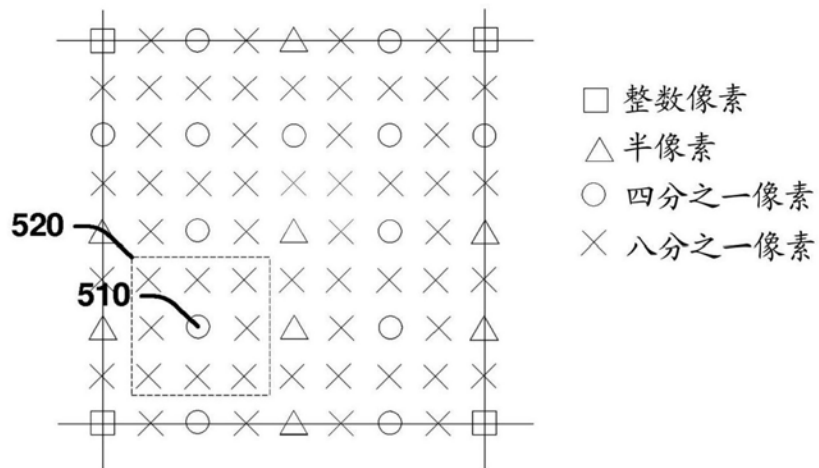


图5

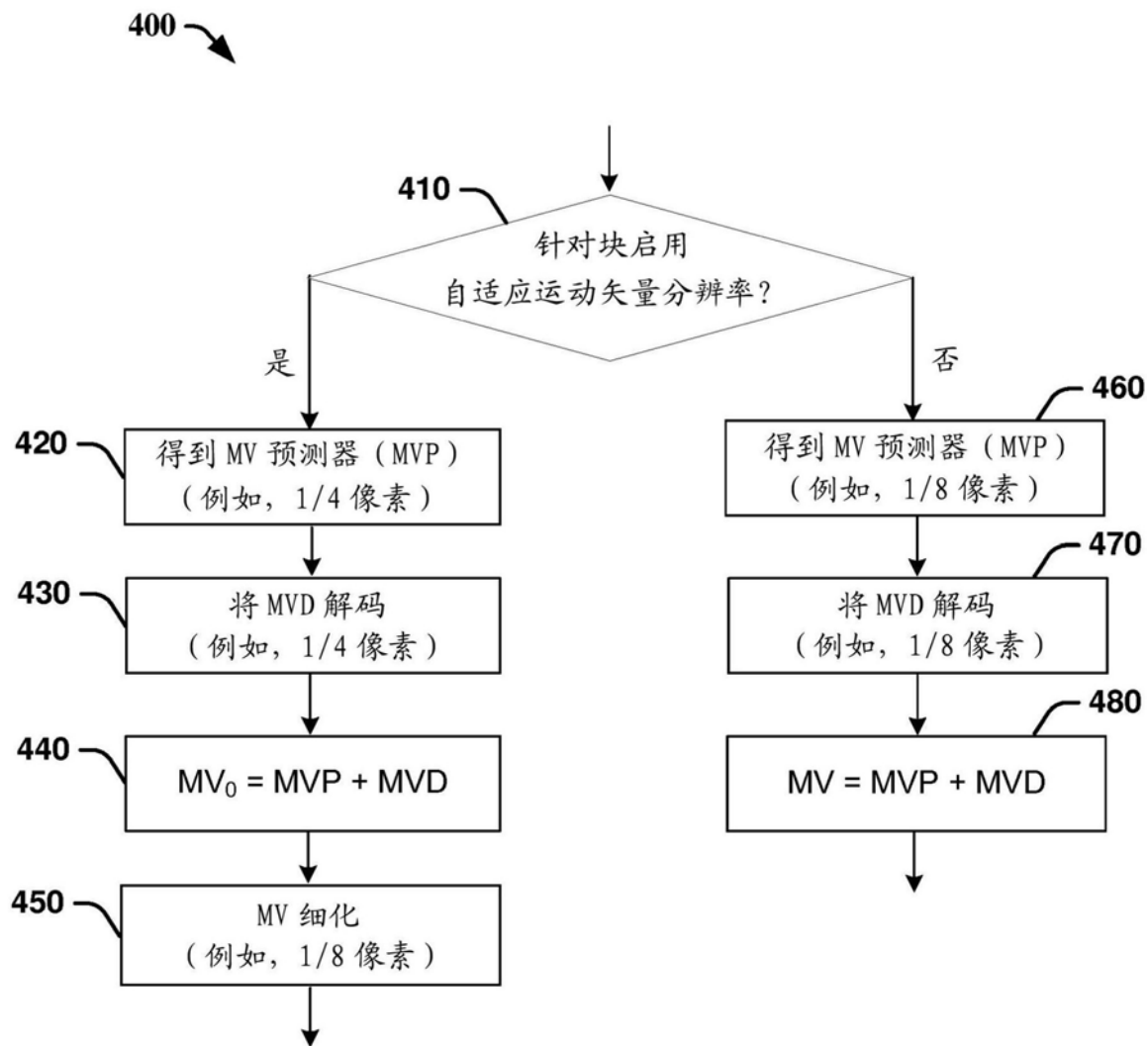


图4

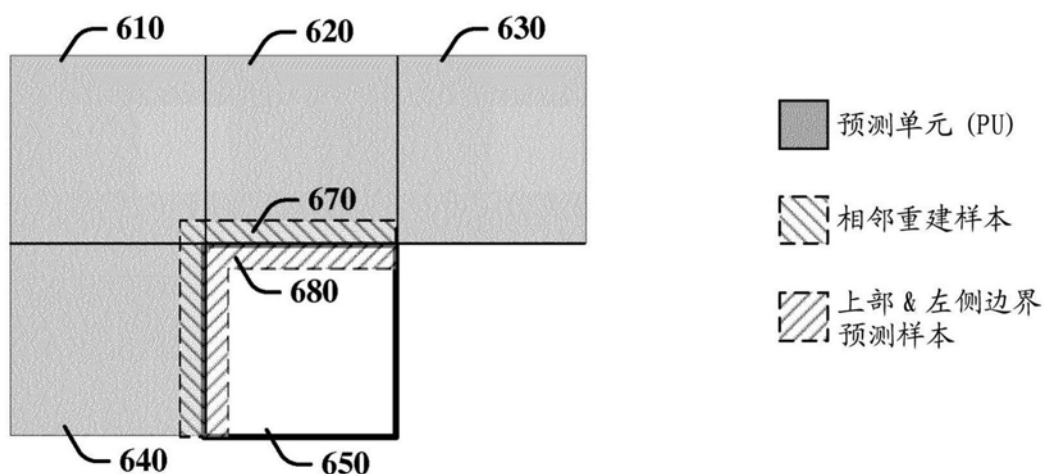


图6A

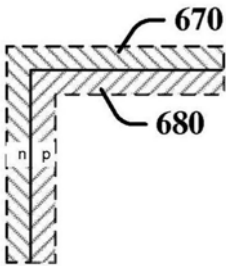


图6B

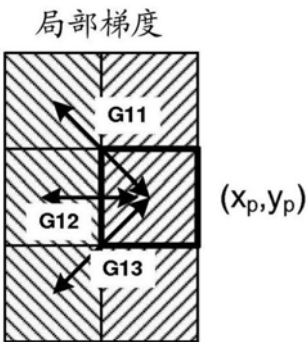


图7A

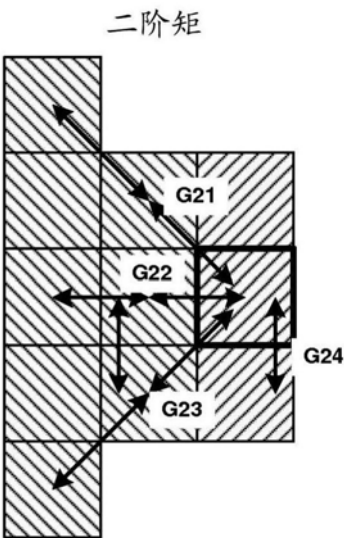
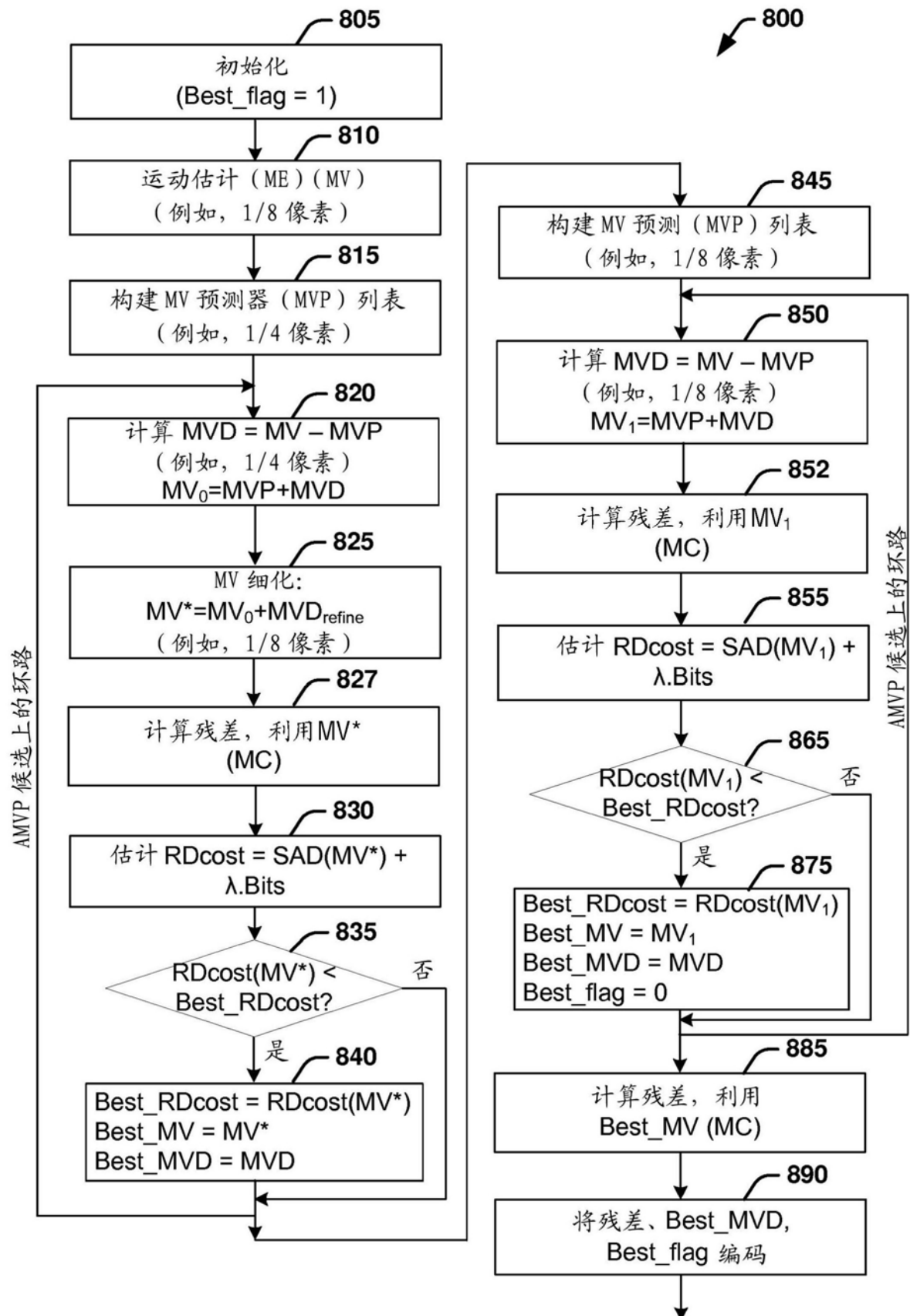


图7B



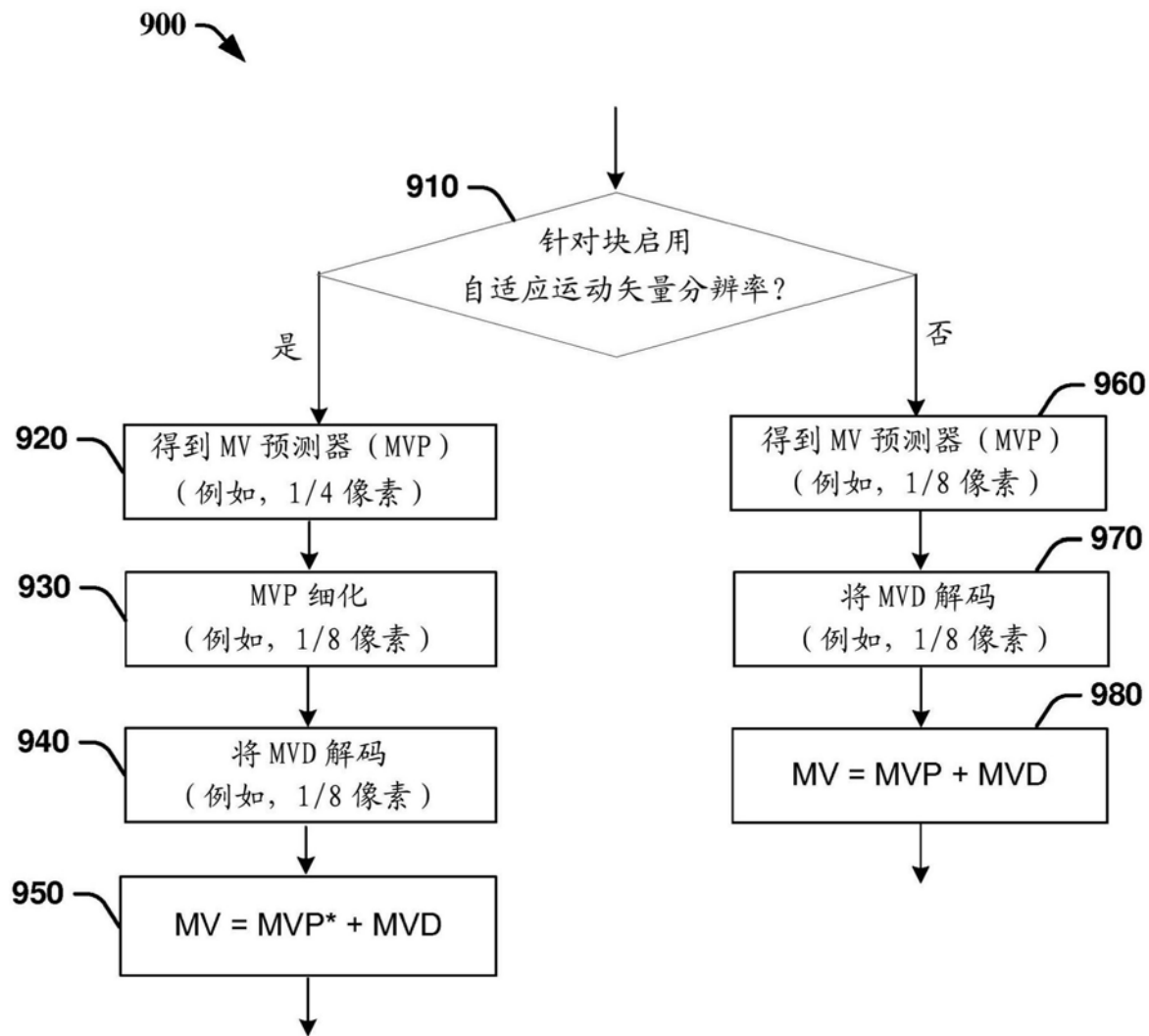


图9

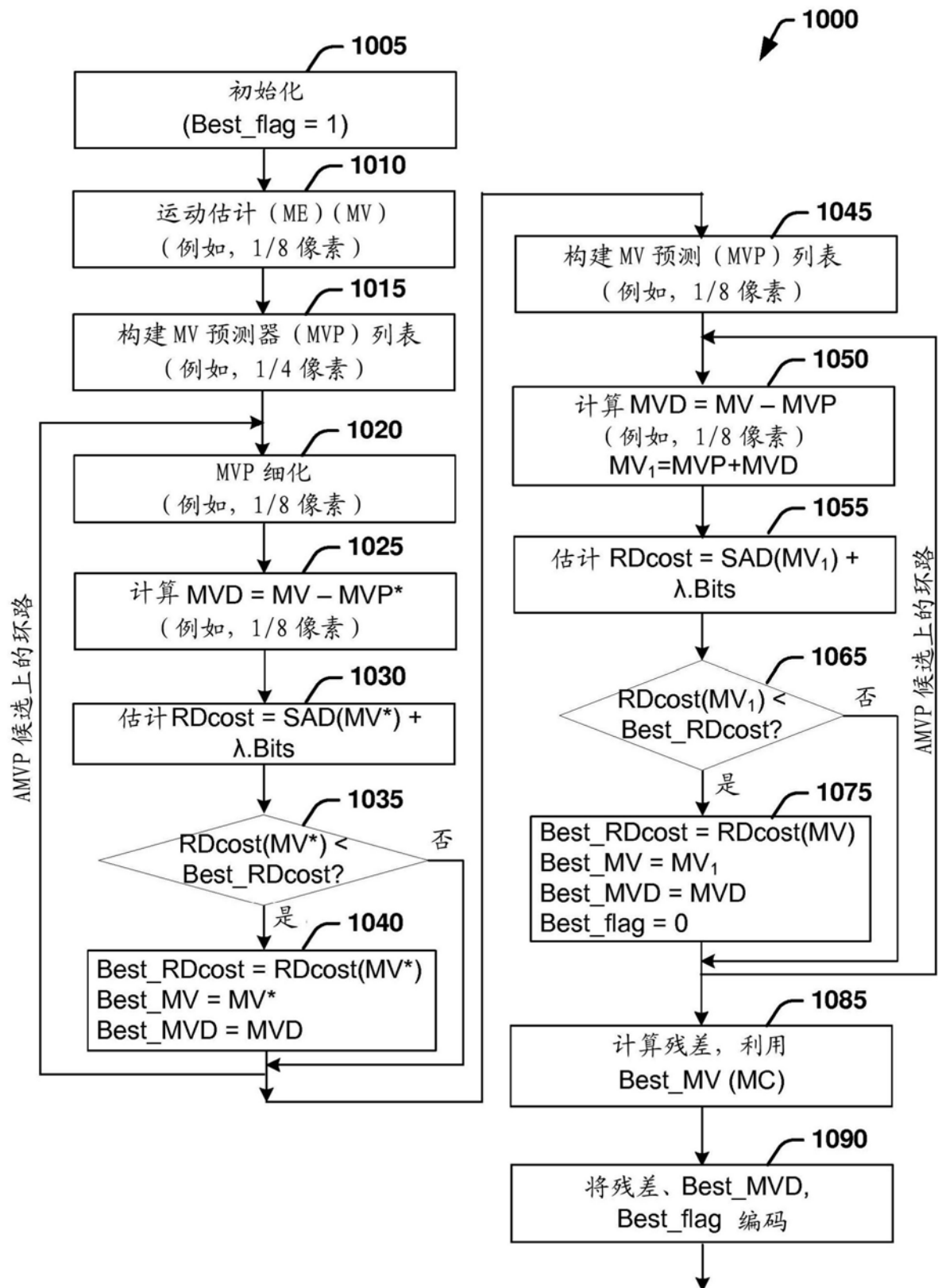


图10

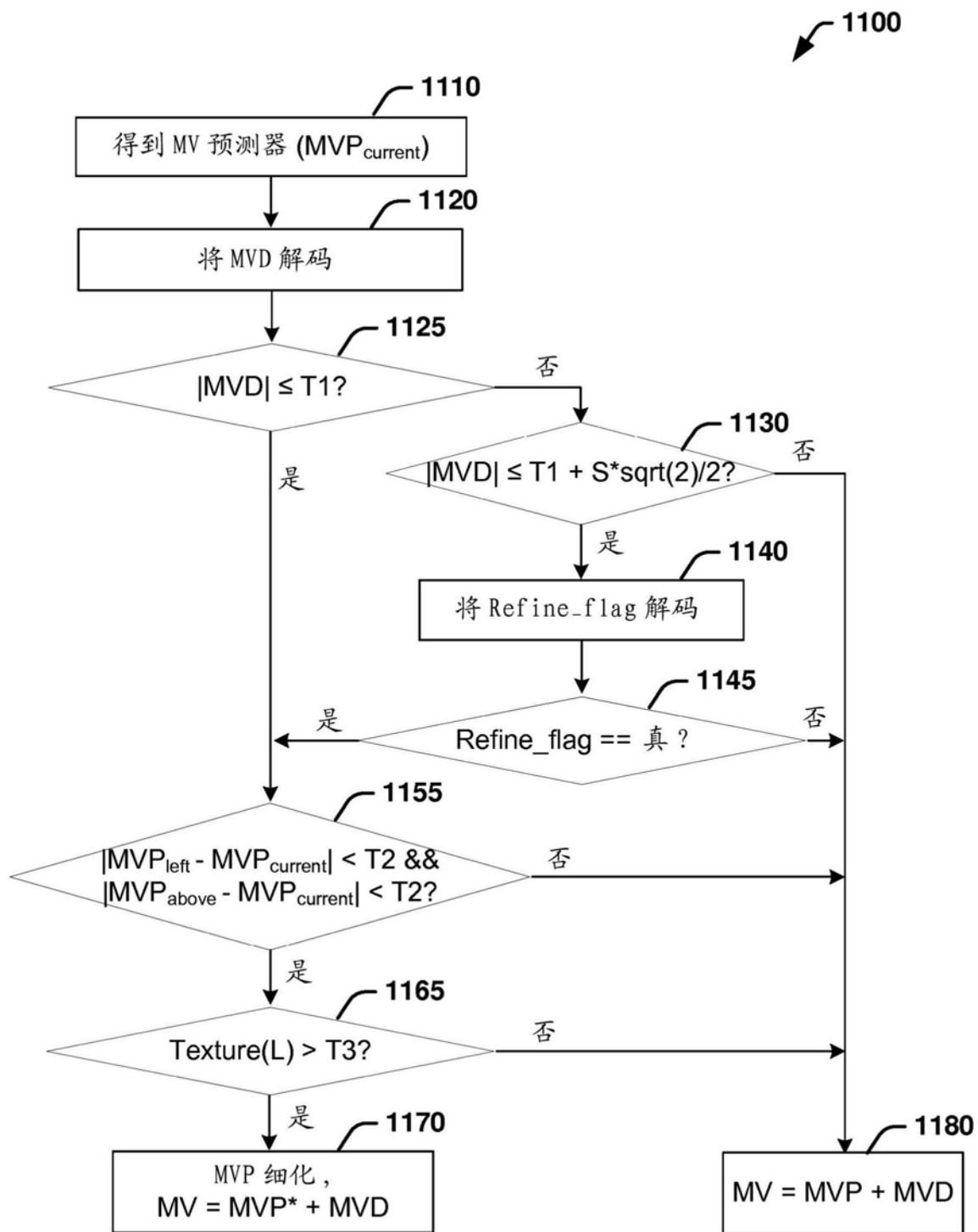


图11

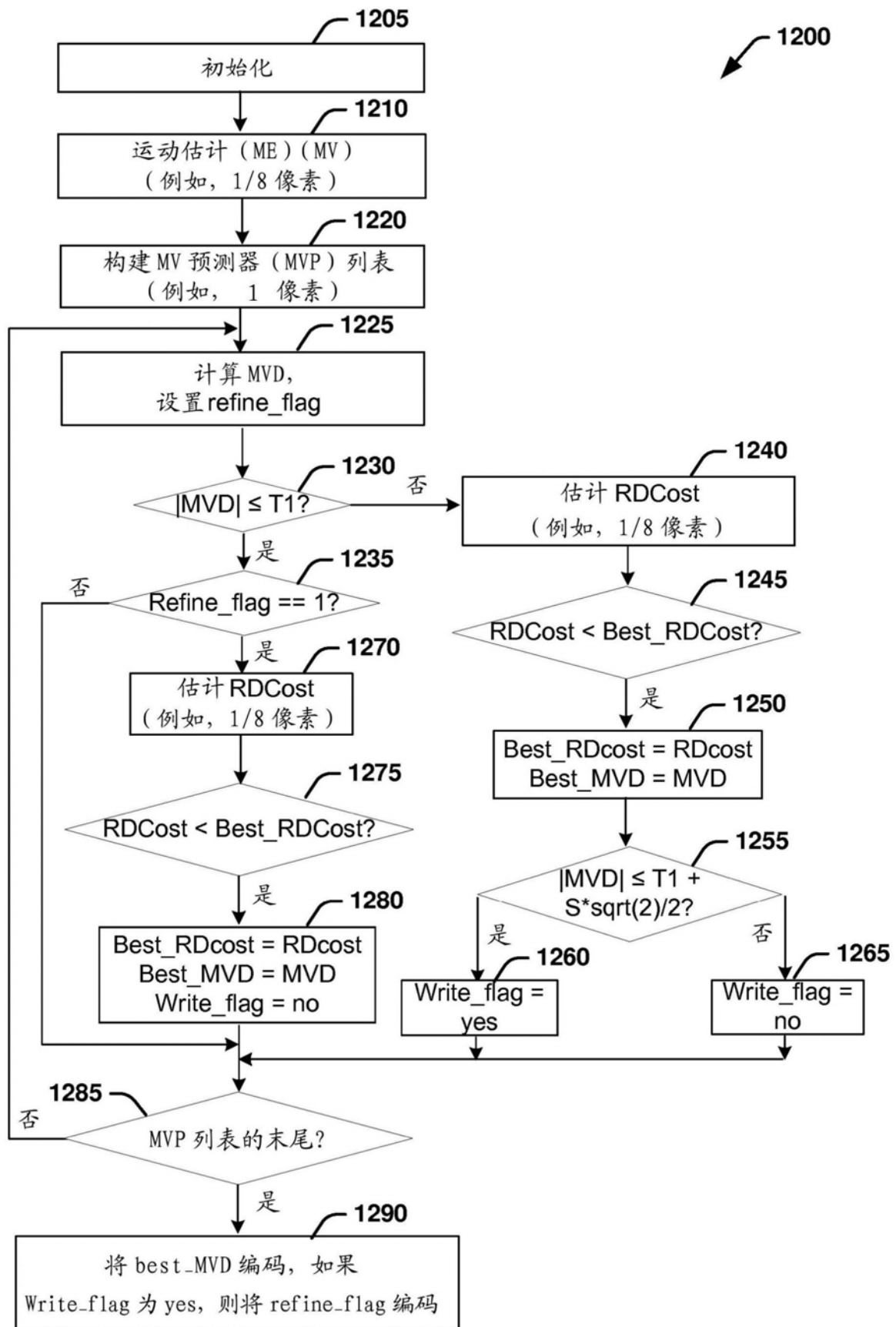


图12

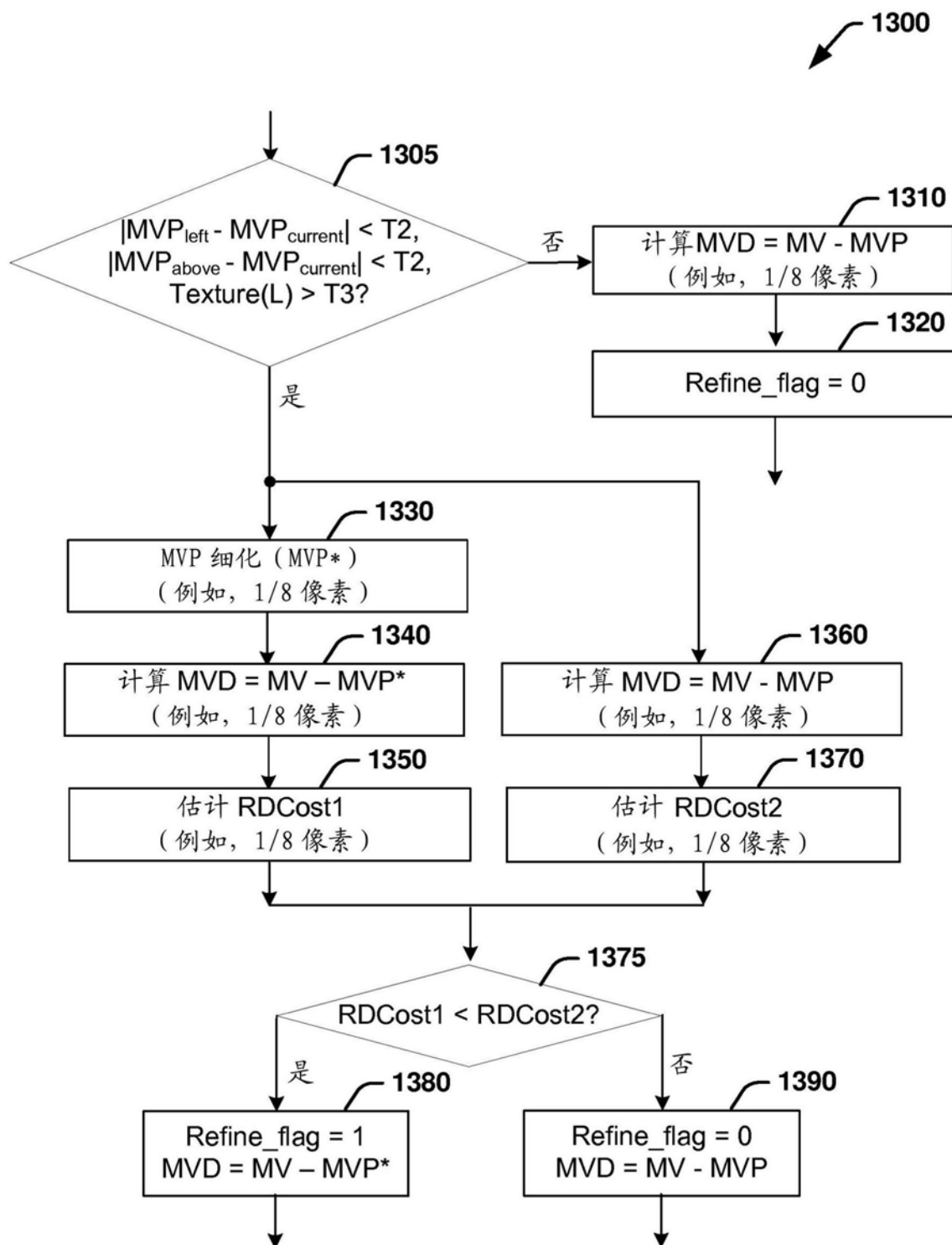


图13

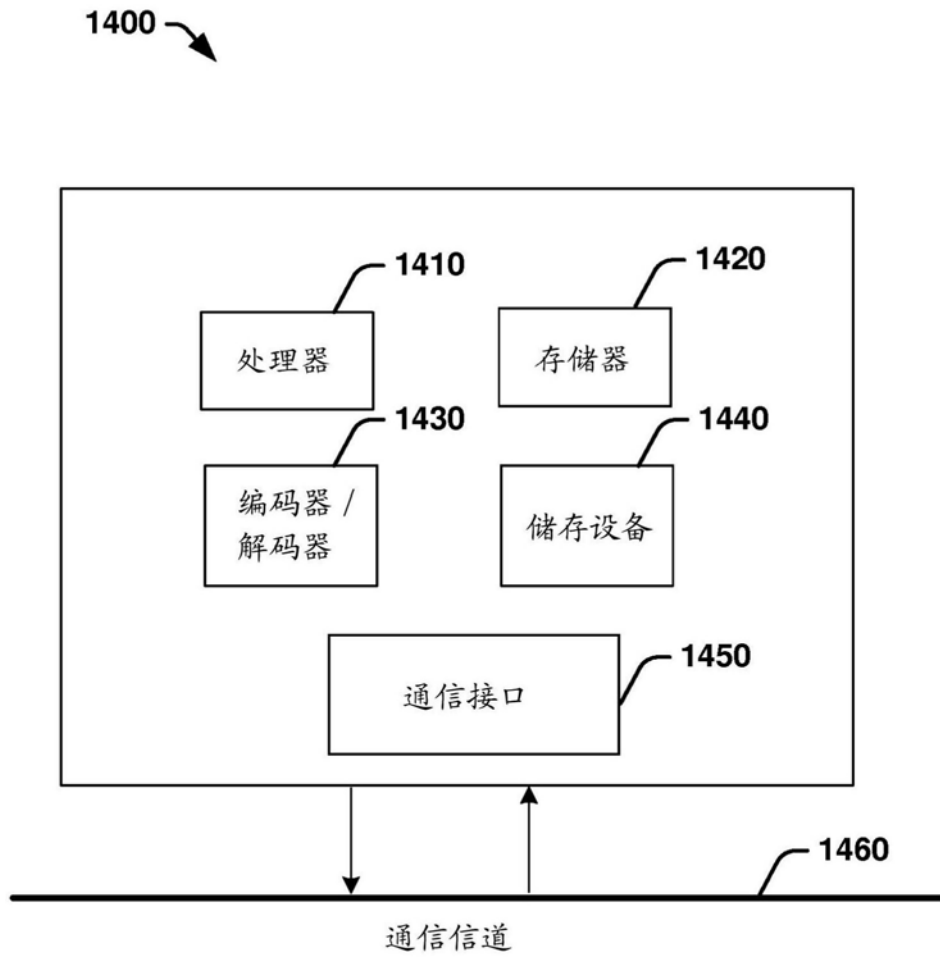


图14