



(12) 发明专利

(10) 授权公告号 CN 115442619 B

(45) 授权公告日 2025. 01. 07

(21) 申请号 202210544150.3

(22) 申请日 2019.06.20

(65) 同一申请的已公布的文献号
申请公布号 CN 115442619 A

(43) 申请公布日 2022.12.06

(30) 优先权数据
201831024666 2018.07.02 IN

(62) 分案原申请数据
201980040320.0 2019.06.20

(73) 专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 斯利拉姆·赛阿瑟拉门
奇甫阿·瑞安·阿
赛格安仁·蔻特查阿

(51) Int.Cl.
H04N 19/517 (2014.01)
H04N 19/523 (2014.01)

(56) 对比文件
CN 101816183 A, 2010.08.25
CN 106878737 A, 2017.06.20

审查员 王瑞

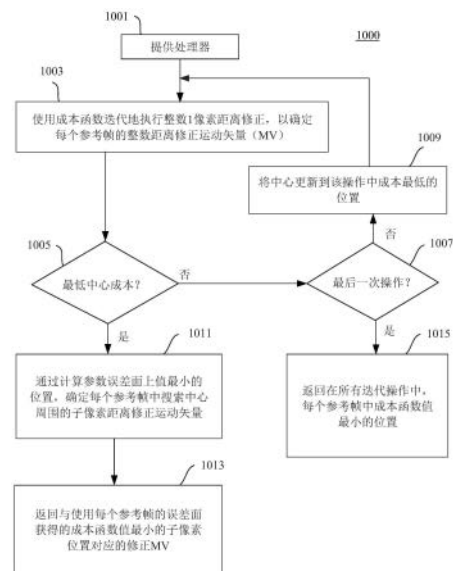
权利要求书3页 说明书23页 附图11页

(54) 发明名称

用于解码端运动矢量修正的基于误差面的子像素精确修正方法

(57) 摘要

鉴于解码端运动矢量修正/推导是译码系统的标准方面,编码器也需要执行相同的误差面技术,以免编码器重建与解码器重建之间有任何漂移。因此,所有实施例的各方面都适用于编码系统和解码系统。在模板匹配中,修正移动仅在参考中发生,从子像素精确中心开始,其中,所述子像素精确中心是基于显式指示的融合索引导出的或通过代价评估隐式推导的。在双边匹配(有或没有平均模板)中,所述修正在参考列表L0和L1中从相应的子像素精确中心开始,其中,所述子像素精确中心是基于所述显式指示的融合索引推导或通过代价评估隐式推导的。



1. 一种解码端运动矢量修正方法,其特征在于,所述方法包括:

通过比较相对于初始运动矢量的候选整数运动矢量位移对应的整数距离代价,确定目标整数运动矢量位移;

通过计算所述整数距离代价,确定子像素运动矢量位移;

根据所述目标整数运动矢量位移、所述子像素运动矢量位移和所述初始运动矢量,确定修正运动矢量;

所述子像素运动矢量位移通过以下等式确定:

$$x0 = (E(-1,0) - E(1,0)) / (2 \times (E(-1,0) + E(1,0) - 2 \times E(0,0))); \text{ 且}$$

$$y0 = (E(0,-1) - E(0,1)) / (2 \times (E(0,-1) + E(0,1) - 2 \times E(0,0)));$$

其中,x0和y0为与子像素运动矢量位移相对于中心(0,0)相关联的坐标;所述中心(0,0)为所述目标整数运动矢量位移所对应的位置,所述位置为与所述位置1像素相邻位置集合相比具有最小代价值的位置;E(-1,0)、E(1,0)、E(0,0)、E(0,-1)和E(0,1)为整数距离代价,所述整数距离代价分别与相对于初始运动矢量的候选整数运动矢量位移(-1,0)、(1,0)、(0,0)、(0,-1)和(0,1)相对应;x0和y0根据增量运动矢量的子像素精度,通过缩放因子N缩放,所述增量运动矢量为所述目标整数运动矢量位移和所述子像素运动矢量位移分别对应的运动矢量的和。

2. 根据权利要求1所述的方法,其特征在于,确定x0的所述等式通过对E(-1,0)、E(1,0)和E(0,0)进行移位、比较和加法操作中的至少一种操作来实现。

3. 根据权利要求1所述的方法,其特征在于,确定y0的所述等式通过对E(0,-1)、E(0,1)和E(0,0)进行移位、比较和加法操作中的至少一种操作来实现。

4. 根据权利要求1所述的方法,其特征在于,所述子像素运动矢量位移限定在-0.5与+0.5个像素之间。

5. 根据权利要求1所述的方法,其特征在于,确定所述目标整数运动矢量位移包括:

计算每个候选整数运动矢量位移的整数距离代价;

选择与最低整数距离代价对应的候选整数运动矢量位移,作为所述目标整数运动矢量位移。

6. 根据权利要求1所述的方法,其特征在于,所述目标整数运动矢量位移包括与参考图像列表L0对应的第一运动矢量位移,和与参考图像列表L1对应的第二运动矢量位移,所述方法还包括:

通过比较整数距离代价,确定所述第一运动矢量位移,其中,所述整数距离代价和所述参考图像列表L0对应的候选整数运动矢量位移相对应;

通过对所述第一运动矢量位移取反,确定所述第二运动矢量位移。

7. 根据权利要求1所述的方法,其特征在于,对于子像素精度1/2、1/4、1/8和1/16,N分别为2、4、8和16。

8. 一种装置,其特征在于,包括:

输入端,用于接收已编码的视频数据;

熵解码单元,用于对所述接收的已编码的视频数据进行解码以获得解码数据;

一个或多个处理器,与所述熵解码单元耦合,用于:

根据所述解码数据,通过比较相对于初始运动矢量的候选整数运动矢量位移对应的整

数距离代价,确定目标整数运动矢量位移;

通过计算所述整数距离代价,确定子像素运动矢量位移;

根据所述目标整数运动矢量位移、所述子像素运动矢量位移和所述初始运动矢量,确定修正运动矢量;

所述子像素运动矢量位移通过以下等式确定:

$$x0 = (E(-1,0) - E(1,0)) / (2 \times (E(-1,0) + E(1,0) - 2 \times E(0,0))); \text{且}$$

$$y0 = (E(0,-1) - E(0,1)) / (2 \times (E(0,-1) + E(0,1) - 2 \times E(0,0)));$$

其中,x0和y0为与所述子像素运动矢量位移相对于中心(0,0)相关联的坐标;所述中心(0,0)为所述目标整数运动矢量位移所对应的位置,所述位置为与所述位置1像素相邻位置集合相比具有最小代价值的位置;E(-1,0)、E(1,0)、E(0,0)、E(0,-1)和E(0,1)为整数距离代价,所述整数距离代价分别与相对于初始运动矢量的候选整数运动矢量位移(-1,0)、(1,0)、(0,0)、(0,-1)和(0,1)相对应;x0和y0根据增量运动矢量的子像素精度,通过缩放因子N缩放,所述增量运动矢量为所述目标整数运动矢量位移和所述子像素运动矢量位移分别对应的运动矢量的和。

9.根据权利要求8所述的装置,其特征在于,确定x0的所述等式通过对E(-1,0)、E(1,0)和E(0,0)进行移位、比较和加法操作中的至少一种操作来实现。

10.根据权利要求8所述的装置,其特征在于,确定y0的所述等式通过对E(0,-1)、E(0,1)和E(0,0)进行移位、比较和加法操作中的至少一种操作来实现。

11.根据权利要求8所述的装置,其特征在于,所述子像素运动矢量位移限定在-0.5与+0.5个像素之间。

12.根据权利要求8所述的装置,其特征在于,所述一个或多个处理器通过以下步骤确定所述目标整数运动矢量位移:

计算每个候选整数运动矢量位移的整数距离代价;

选择与最低整数距离代价对应的候选整数运动矢量位移,作为所述目标整数运动矢量位移。

13.根据权利要求8所述的装置,其特征在于,所述目标整数运动矢量位移包括与参考图像列表L0对应的第一运动矢量位移,和与参考图像列表L1对应的第二运动矢量位移,所述一个或多个处理器用于:

通过比较整数距离代价,确定所述第一运动矢量位移,其中,所述整数距离代价和所述参考图像列表L0对应的候选整数运动矢量位移相对应;

通过对所述第一运动矢量位移取反,确定所述第二运动矢量位移。

14.一种存储指令的计算机可读介质,其特征在于,当所述指令在处理器中执行时,执行根据权利要求1至7中任一项所述的方法的步骤。

15.一种解码端运动矢量修正方法,其特征在于,所述方法包括:

接收已编码的视频码流;

对所述接收的已编码的视频码流进行解码以获得解码数据,所述解码数据用于指示初始运动矢量或者所述解码数据用于解码器推导所述初始运动矢量;

通过比较相对于所述初始运动矢量的候选整数运动矢量位移对应的整数距离代价,确定目标整数运动矢量位移;

通过计算所述整数距离代价,确定子像素运动矢量位移;

根据所述目标整数运动矢量位移、所述子像素运动矢量位移和所述初始运动矢量,确定修正运动矢量;

所述子像素运动矢量位移通过以下等式确定:

$$x_0 = (E(-1,0) - E(1,0)) / (2 \times (E(-1,0) + E(1,0) - 2 \times E(0,0))); \text{ 且}$$

$$y_0 = (E(0,-1) - E(0,1)) / (2 \times (E(0,-1) + E(0,1) - 2 \times E(0,0)));$$

其中, x_0 和 y_0 为与子像素运动矢量位移相对于中心(0,0)相关联的坐标;所述中心(0,0)为所述目标整数运动矢量位移所对应的位置,所述位置为与所述位置1像素相邻位置集合相比具有最小代价值的位置; $E(-1,0)$ 、 $E(1,0)$ 、 $E(0,0)$ 、 $E(0,-1)$ 和 $E(0,1)$ 为整数距离代价,所述整数距离代价分别与相对于初始运动矢量的候选整数运动矢量位移 $(-1,0)$ 、 $(1,0)$ 、 $(0,0)$ 、 $(0,-1)$ 和 $(0,1)$ 相对应; x_0 和 y_0 根据增量运动矢量的子像素精度,通过缩放因子N缩放,所述增量运动矢量为所述目标整数运动矢量位移和所述子像素运动矢量位移分别对应的运动矢量的和。

16.根据权利要求15所述的方法,其特征在于,确定 x_0 的所述等式通过对 $E(-1,0)$ 、 $E(1,0)$ 和 $E(0,0)$ 进行移位、比较和加法操作中的至少一种操作来实现。

17.根据权利要求15所述的方法,其特征在于,确定 y_0 的所述等式通过对 $E(0,-1)$ 、 $E(0,1)$ 和 $E(0,0)$ 进行移位、比较和加法操作中的至少一种操作来实现。

18.根据权利要求15所述的方法,其特征在于,所述子像素运动矢量位移限定在-0.5与+0.5个像素之间。

19.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有程序,当所述程序由一个或多个处理器执行时,以使计算机实现如权利要求15至18中的任一项所述的方法。

用于解码端运动矢量修正的基于误差面的子像素精确修正方法

[0001] 本申请是分案申请,原申请的申请号是201980040320.0,原申请日是2019年06月20日,原申请的全部内容通过引用结合在本申请中。

[0002] 相关申请案交叉申请

[0003] 本申请要求于2018年07月02日提交、申请号为201831024666、发明名称为“用于解码端运动矢量修正的基于误差面的子像素精确修正方法 (AN ERROR SURFACE BASED SUB-PIXEL ACCURATE REFINEMENT METHOD FOR DECODER SIDE MOTION VECTOR REFINEMENT)”的印度临时专利申请的权益,其全部内容通过引用的方式并入本文中。

背景技术

[0004] 目前的混合视频编解码器(如H.264/AVC或H.265/HEVC)采用包括预测译码在内的压缩。将视频序列的图像划分为像素块,然后对这些块进行译码。不是逐个像素对块进行译码,而是使用与整个块空间相邻或时间相邻的已编码像素来预测该块。编码器仅进一步对该块与其预测块之间的差值进行处理。所述进一步处理通常包括将块像素变换为变换域中的系数。然后,可以通过量化进一步压缩系数,并进一步通过熵译码进行压缩以形成码流。所述码流还包括能够对编码视频进行解码的任何信令信息。例如,所述信令信息可包括编码相关的设置,如输入图像的大小、帧率、量化步长指示、应用于图像块的预测等。以编码器和解码器已知的方式在码流内对经译码信令信息和经译码信号进行排序。这使解码器能够对经译码信令信息和经译码信号进行解析。

[0005] 时间预测利用视频图像(也称为帧)之间的时间相关性。时间预测也称为帧间预测,因为它利用(帧间)不同的视频帧之间的相关性进行预测。因此,根据一个或多个先前编码的图像(也称为一个或多个参考图像)对正在编码的块(也称为当前块)进行预测。参考图像不一定按照视频序列的显示顺序,是当前块所在的当前图像的前一张图像。编码器可以按不同于显示顺序的译码顺序对图像进行编码。可以确定参考图像中的共址块作为当前块的预测块。共址块是位于参考图像中与当前图像中的当前块位置相同的块。这种预测对于静止的图像区域(即没有从一个图像移动到另一个图像的图像区域)而言是准确的。

[0006] 为了获得考虑了移动的预测值,即运动补偿预测值,通常在确定当前块的预测时采用运动估计。因此,根据参考图像中的块预测当前块,该块位于与共址块的位置相距运动矢量所给定的距离的位置处。为了使解码器能够确定当前块的相同预测,可以在码流中指示运动矢量。为了进一步减少为每个块指示运动矢量而引起的信令开销,可以估计运动矢量本身。可以根据空域和/或时域中邻块的运动矢量进行运动矢量估计。

[0007] 可以使用一个参考图像或通过从两个或更多个参考图像获得的预测进行加权来计算当前块的预测。参考图像可以是相邻图像,即按显示顺序紧接当前图像之前和/或紧接当前图像之后的图像,因为相邻图像最有可能与当前图像类似。然而,通常,参考图像还可以是码流中按显示顺序在当前图像之前或之后的任何其它图像(解码顺序)。例如,在

视频内容中有遮挡和/或非线性移动的情况下,这是有利的。因此,还可以在码流中指示参考图像标识。

[0008] 帧间预测的一种特殊模式是双向预测,其中使用两个参考图像生成当前块的预测。特别地,将对相应的两个参考图像确定的两个预测组合成当前块的预测信号。双向预测可以产生比单向预测(即仅使用单个参考图像进行的预测)更精确的当前块预测。更精确的预测使得当前块的像素与预测之间的差值(也称为“残差”)较小,从而可以更有效地编码,即压缩成更短的码流。通常,可以使用多于两个参考图像来查找相应的多于两个参考块,以预测当前块,即,可以应用多参考帧间预测。因此,术语“多参考预测”包括双向预测以及使用多于两个参考图像的预测。

[0009] 为了提供更精确的运动估计,可以通过在像素之间插值像素点来提高参考图像的分辨率。可以通过对最近像素进行加权平均来进行分数像素插值。例如,在半像素分辨率的情况下,通常采用双线性插值。其它分数像素计算为按相应的最近像素与所预测像素之间的距离的倒数加权的最近像素的平均值。

[0010] 运动矢量估计是一项计算复杂的任务,计算当前块与参考图像中候选运动矢量所指向的对应预测块之间的相似度。通常,搜索区域包括图像的 $M \times M$ 个像素点,并且测试 $M \times M$ 个候选位置的每个像素点位置。该测试包括计算 $N \times N$ 参考块C与位于搜索区域的测试候选位置的块R之间的相似度度量。绝对差值和(sum of absolute difference, SAD)是为简单起见经常使用的一种度量,通过以下给出:

$$[0011] \quad SAD(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |R_{i,j}(x, y) - C_{i,j}|$$

[0012] 在上述公式中,x和y定义了搜索区域内的候选位置,而索引i和j表示参考块C和候选块R内的像素点。候选位置通常称为块位移或块偏移,反映出块匹配表示为参考块在搜索区域内的移位,并计算参考块C与搜索区域的重叠部分之间的相似度。为了降低复杂度,通常通过将候选运动矢量限制在一定的搜索空间内来减少候选运动矢量的数量。例如,搜索空间可以由参考图像中与当前图像中当前块的位置对应的位置周围的像素的数量和/或位置来定义。在计算所有 $M \times M$ 个候选位置x和y的SAD之后,最佳匹配块R为具有最低SAD的位置上的块,最低SAD对应与参考块C的最大相似度。另一方面,候选运动矢量可以由邻块的运动矢量组成的候选运动矢量列表定义。

[0013] 运动矢量通常至少部分地在编码端确定,并在编码码流内向解码器指示。然而,也可以在解码器侧导出运动矢量。在这种情况下,无法在解码器侧获得当前块,并且该当前块不能用于计算与参考图像中候选运动矢量所指向的块的相似度。因此,使用由已解码块的像素构造的模板代替当前块。例如,可以使用与当前块相邻的已解码像素。这种运动估计的优点在于减少信令:在编码器和解码器侧以相同的方式导出运动矢量,因此不需要信令。另一方面,这种运动估计的精确度可能较低。

[0014] 为了在精确度与信令开销之间实现权衡,运动矢量估计可以分为两个步骤:运动矢量推导和运动矢量修正。例如,运动矢量推导可包括从候选列表中选择运动矢量。这种选择的运动矢量可以通过在搜索空间内搜索等方式进一步修正。搜索空间中的搜索是基于计算每个候选运动矢量的代价函数,即,候选运动矢量所指向的块的每个候选位置的代价函数。

[0015] X.Chen、J.An、J.Zeng发表的文献JVET-D0029:基于双边模板匹配的解码端运动矢量修正(该文献可见于:<http://phenix.it-sudparis.eu/jvet/site>)展示了运动矢量修正,其中,查找 整数像素分辨率的第一运动矢量,并通过在第一运动矢量周围的搜索空间中以半像素分辨率 搜索来进一步修正。采用基于块模板的双向运动矢量搜索。

[0016] 运动矢量估计是现代视频编解码器的关键特性,因为它在质量、速率和复杂度方面的效率都直接影响到视频编解码的效率。

发明内容

[0017] 本发明涉及视频的编码和解码,具体涉及确定运动矢量。

[0018] 在本申请的第一方面中,提供了一种用于在解码端运动矢量修正系统中,获取一个或多个参考帧中子像素精确增量运动矢量的方法。所述一个或多个参考帧在相应的一个或多个初始子像素精确修正中心周围。所述方法可以包括以下步骤:

[0019] 使用代价函数迭代地执行多个整数1像素距离修正操作(迭代),以确定每个参考帧的整数距离修正运动矢量,其中,搜索中心在操作(迭代)之后更新到所述操作(迭代)中代价最低的位置;确定由于给定操作(迭代)中的中心位置代价低于该中心位置周围的1像素相邻位置集合的代价,发生提前退出迭代循环,或确定达到预定次数的操作(迭代)。在发生提前退出的情况下,所述方法可以包括:通过使用最后搜索中心以及最后搜索中心周围的1像素相邻位置集合的代价函数值计算在拟合的参数误差面上值最小的位置,确定每个参考帧中最后搜索中心周围的子像素距离修正运动矢量;并返回总修正运动矢量,作为每个参考帧的所述确定的整数距离修正运动矢量和所述确定的子像素距离修正运动矢量的和。在达到预定次数的操作(迭代)的情况下,所述方法可以包括返回与所有操作(迭代)中代价函数值最小的位置对应的修正运动矢量。

[0020] 在所述第一方面的一种实现方式中,对参考列表L0和L1中的联合修正执行所述代价函数,使得所述参考列表L1中的所述位移与所述参考列表L0中的所述位移相等且在水平方向和垂直方向上相反(代价函数称为SBM_JOINT)。

[0021] 在本申请的第二方面中,提供了一种用于在解码端运动矢量修正系统中,获取一个或多个参考帧中子像素精确增量运动矢量的方法。所述一个或多个参考帧在相应的一个或多个初始子像素精确修正中心周围多个参考帧中获取,所述方法可以包括以下步骤:

[0022] 使用第一代价函数迭代地执行多个整数1像素距离修正操作(迭代),以确定每个参考帧的整数距离修正运动矢量,其中,搜索中心在操作(迭代)之后更新到所述操作(迭代)中代价最低的位置;确定发生提前退出迭代循环,由于给定操作(迭代)中的所述中心位置代价低于该中心位置周围的1像素相邻位置集合的代价,,或确定达到预定次数的操作(迭代)。

[0023] 使用第二代价函数计算最后搜索中心和所述最后搜索中心的1像素相邻位置集合的代价函数值:如果与所述最后搜索中心的所述1像素相邻位置集合的第二代价函数值相比,所述最后搜索中心位置具有最低的第二代价函数值,则通过使用所述第二代价函数值计算在拟合的参数误差面上值最小的位置,确定每个参考帧中最佳整数距离修正位置周围的子像素距离修正运动矢量;并返回总修正运动矢量,作为每个参考帧的所述确定的整数距离修正运动矢量和所述确定的子像素距离修正运动矢量的和。如果与所述最后搜索中

心的所述1像素相邻位置集合的第二代价函数值相比,所述最后搜索中心位置没有最低的第二代价函数值,则所述方法包括:返回与每个参考帧中具有最佳第二代价函数值的位置对应的修正运动矢量。

[0024] 在所述第二方面的一种实现方式中,所述第一代价函数为SBM_JOINT,所述第二代价函数为TBM_INDEPDENDENT,其定义为与在参考列表L0和L1中针对共同的双边平均模板执行独立修正相关的代价函数。

[0025] 所述方法可用于使用模板匹配的情况和使用双边匹配的情况。

[0026] 在本发明的另一方面中,本文公开的方法可以实现为存储在非瞬时性计算机可读介质中的指令,所述指令可由处理器读取和执行以执行所述方法的步骤。

[0027] 在本发明的一些方面中,用于解码端运动矢量修正的方法包括:通过比较相对于初始运动矢量的候选整数运动矢量位移对应的整数距离代价,确定目标整数运动矢量位移;通过计算所述整数距离代价,确定子像素运动矢量位移;根据所述目标整数运动矢量位移、所述子像素运动矢量位移和所述初始运动矢量计算修正运动矢量。

[0028] 相比传统技术,通过本发明获得多种益处。例如,本发明的实施例通过将给定迭代操作中的中心位置代价与中心位置周围的1像素相邻位置集合的代价进行比较,确定提前退出迭代循环。通过提前退出迭代循环,可以减少或消除不必要的计算。

[0029] 此外,可以实现上述方法的装置可以是软件和硬件的组合。例如,编码和/或解码可以由如通用处理器(general purpose processor,CPU)、数字信号处理器(digital signal processor, DSP)、专用集成电路(application specific integrated circuit, ASIC)或现场可编程门阵列(field programmable gate array,FPGA)等集成电路(半导体设备或芯片)执行。然而,本发明的实施例不限于在可编程硬件中实现。本发明的实施例可以在专用集成电路中实现,也可以由CPU、DSP、FPGA和ASIC组件中的一个或多个的组合实现。

附图说明

[0030] 下文结合附图详细描述示例性实施例,其中:

[0031] 图1为本发明一实施例提供的用于对视频信号进行编码的编码器的示例性结构的框图;

[0032] 图2为本发明一实施例提供的用于对视频信号进行解码的解码器的示例性结构的框图;

[0033] 图3为适用于双向预测的示例性模板匹配的示意图;

[0034] 图4A为适用于单向预测和双向预测的示例性模板匹配的示意图;

[0035] 图4B为适用于单向预测和双向预测的示例性双边匹配的示意图;

[0036] 图5为运动矢量搜索的可能实现方式的流程图;

[0037] 图6为应用于视频译码的局部照明补偿的示例的示意图;

[0038] 图7为解码端运动矢量修正的示例的示意图;

[0039] 图8为子像素位置的示例的示意图;

[0040] 图9为本发明一实施例提供的用于获得一个或多个参考帧中的子像素精确增量运动矢量修正的中心像素周围的1像素相邻位置集合的框图;

[0041] 图10为本发明一些实施例提供的用于在解码端运动矢量修正系统中获取一个或多个参考帧中的子像素精确增量运动矢量修正的方法的简化流程图;

[0042] 图11为用于实现本方面实施例的方法的简化流程图;

[0043] 图12为可用于实现本发明各种实施例的装置的框图。

具体实施方式

[0044] 本方面的实施例涉及改进应用于运动矢量修正中的模板匹配。具体地,即使不再(通过局部光照控制)进一步调整所找到的最佳匹配块的均值,仍将模板匹配应用于零均值模板和零均值候选块。

[0045] 使用模板匹配来查找分别指向第一参考图像和第二参考图像的最佳第一运动矢量和最佳第二运动矢量。通过在预定搜索空间中对初始运动矢量给出的位置进行模板匹配,对每个参考图像执行模板匹配,其中,所述初始运动矢量可由解码器推导或向解码器指示。

[0046] 可以根据从初始运动矢量所指向的块导出的块模板执行模板匹配。

[0047] 例如,可以在混合视频编码器和/或解码器中采用这种用于查找最佳匹配块以获得当前块的预测值的模板匹配。例如,可以有利地应用于如HEVC等编码器和/或解码器。特别地, HEVC或新的编解码器/标准的进一步开发可以使用本发明的实施例。

[0048] 图1示出了编码器100,包括用于接收视频流的帧或图像的输入图像像素的输入端102和用于生成编码视频码流的输出端172。如本文所使用的,本发明中的术语“帧”也称为图像。需要说明的是,本发明还适用于视频的交错领域。通常,图像包括 $m \times n$ 个像素,与图像像素点对应,可以包括一个或多个颜色分量。如本文所使用的,以下描述将像素称为亮度像素点。然而,需要说明的是,本发明的运动矢量搜索可以应用于包括色度的任何颜色分量或RGB等颜色空间的分量。此外,可能有利的是,仅对一个分量执行运动矢量估计并将估计的运动矢量应用于多个或所有分量。

[0049] 待译码的输入块的大小不一定相同。一个图像可以包括不同大小的块,且不同图像的块栅格也可以不同。

[0050] 在一个示例性实施例中,编码器100用于对视频流执行预测、变换、量化和熵译码。变换、量化和熵译码分别由变换单元106、量化单元108和熵编码单元170执行,以生成编码视频码流。

[0051] 视频流可以包括多个帧,其中每个帧被划分为一定大小的帧内编码块或帧间编码块。例如,视频流的第一帧的块由帧内预测单元154进行帧内编码。仅使用同一帧内的信息对帧内帧进行编码,使得可以对该帧内帧进行独立解码,并且该帧内帧可在码流中提供用于随机访问的入口点。视频流的其它帧的块可由帧间预测单元144进行帧间编码,即,使用先前编码帧(参考帧)中的信息减少时间冗余,从而根据参考帧中的块预测出帧间编码帧的每个块。模式选择单元160用于在帧内预测单元154处理的帧的块155或帧间预测单元144处理的帧的块145之间进行选择。模式选择单元160还控制帧内预测或帧间预测的参数。为了能够刷新图像信息,可以在帧间编码帧内提供帧内编码块。此外,可以将仅包含帧内编码块的帧内帧规律地插入到视频序列中,以便提供解码的入口点,即,解码器可以在没有先前编码帧中的信息的情况下开始解码的点。

[0052] 帧内估计单元152和帧内预测单元154为用于执行帧内预测的单元。具体地,帧内估计单元152也可以根据原始图像的情况来推导预测模式,而帧内预测单元154为差分编码提供对应的预测值,即,使用所选预测模式预测的像素点。为了执行空间预测或时间预测,编码块还可以通过反量化单元110和逆变换单元112进行处理,以提供逆变换块113。重建单元114将逆变换块113与预测块165组合以提供重建块115,将该重建块115提供给环路滤波单元120以进一步提高解码图像的质量。然后,滤波块形成参考帧,然后存储在解码图像缓冲器130中。反量化单元110、逆变换单元112、重建单元114和环路滤波器120构成解码器(解码环路)的一部分。在编码端的这种解码环路(解码器)的优点在于,产生与在解码端重建的参考图像相同的参考帧。因此,编码器和解码器以对应的方式运行。本文中的术语“重建”是指通过将预测块165添加到逆变换(解码残差)块113来获得重建块115。

[0053] 编码器100还包括帧间估计单元142,所述帧间估计单元142从解码图像缓冲器130接收待帧间编码的当前帧或图像以及一个或多个参考帧的图像块101。运动估计由帧间估计单元142执行,而运动补偿由帧间预测单元144执行。运动估计用于根据某种代价函数(例如也使用待编码的原始图像)来获得运动矢量和参考帧。例如,运动估计(帧间估计)单元142可以提供初始运动矢量估计。然后,可以在码流内直接以运动矢量的形式指示(作为信号传输)初始运动矢量,或作为说明候选列表内的运动矢量候选的索引指示初始运动矢量,其中,在编码器和解码器侧以相同的方式,根据预定规则构造所述候选列表。然后,运动补偿推导当前块的预测值,作为与参考帧中的当前块共址的块到参考帧中的参考块的平移,即通过运动矢量推导当前块的预测值。帧间预测单元144输出当前块的预测块145,其中预测块145最小化代价函数。例如,代价函数可以是待编码的当前块与其预测块之间的差值,即代价函数最小化残差块105。例如,残差块的最小化是基于计算当前块的所有像素(像素点)与候选参考图像中的候选块之间的绝对差值和(sum of absolute difference, SAD)。通常,可以使用任何其它相似度度量,如均方误差(mean square error, MSE)或结构相似度度量(structural similarity metric, SSIM)。

[0054] 代价函数也可以是对此类帧间块进行编码所必需的比特数和/或此类编码导致的失真。因此,率失真优化过程可用于决定运动矢量选择和/或通常用于决定编码参数,如是否对块使用帧间预测或帧内预测,以及使用哪些设置。

[0055] 帧内估计单元152和帧间预测单元154接收待帧内编码的当前帧或图像的图像块101以及来自当前帧的已重建区域的一个或多个参考像素点117作为输入。然后,帧内预测根据当前帧的参考像素点的函数描述当前帧的当前块的像素。帧内预测单元154输出当前块的预测块,其中,所述预测块利于最小化待编码当前块与其预测块之间的差值,即,最小化残差块。例如,可根据率失真优化过程最小化残差块。特别地,获得预测块作为参考像素点的方向插值。所述方向可以通过率失真优化和/或通过计算上述与帧间预测相关的相似度度量来确定。

[0056] 帧间估计单元142接收待帧间编码的当前帧或图像以及两个或更多个已解码图像231的块或更通用形式的图像像素点作为输入。然后,帧间预测根据运动矢量描述当前帧的当前图像像素点,以参考参考图像的图像像素点。帧间预测单元144输出当前图像像素点的一个或多个运动矢量145,其中,运动矢量所指向的参考图像像素点利于最小化待编码的当前图像像素点与其参考图像像素点之间的差值,即,最小化残差图像像素点。然后,

帧间预测单元 144提供当前块的预测值以用于差分编码。

[0057] 然后,变换单元106对当前块与其预测块之间的差值(即,残差块105)进行变换以生成变换系数107。变换系数107由量化单元108进行量化并由熵编码单元170进行熵译码。由此生成的编码图像数据171(即,编码视频码流)包括帧内编码块和帧间编码块以及对应的信令信息(如模式指示、运动矢量的指示和/或帧内预测方向)。变换单元106可以应用线性变换,如离散傅里叶变换(Discrete Fourier Transformation,DFT)、快速傅里叶变换(Fast Fourier Transformation,FFT)或离散余弦变换(Discrete Cosine Transformation,DCT)。所述空间频域变换的优点在于,所得系数107通常在较低频率下具有较高值。因此,在有效系数扫描(如Z形扫描)和量化之后,所得的值序列通常在开头有一些较大的值,并以一串零结束。这使编码效率进一步提高。量化单元108通过降低系数值的分辨率来执行实际的有损压缩。然后,熵编码单元170将二进制码字分配给系数值以产生码流。熵编码单元170也对信令信息(图1中未示出)进行编码。

[0058] 图2示出了视频解码器200。视频解码器200包括解码图像缓冲器230、帧间预测单元244和帧内预测单元254,所述帧间预测单元244和帧内预测单元254为块预测单元。解码图像缓冲器230用于存储从编码视频码流重建的至少一个(对于单向预测而言)或至少两个(对于双向预测而言)参考帧,所述参考帧与编码视频码流的当前帧(当前解码帧)不同。帧内预测单元254用于生成预测块,所述预测块用于估计待解码块。帧内预测单元254用于根据从解码图像缓冲器230获得的参考像素点生成该预测块。

[0059] 解码器200用于对视频编码器100所生成的编码视频码流进行解码,优选地,解码器200和编码器100为待编码/解码的相应块生成相同的预测块。解码图像缓冲器230和帧内预测单元254的特征与图1的解码图像缓冲器130和帧内预测单元154的特征类似。

[0060] 视频解码器200还包括在视频编码器100中也存在的单元,例如,反量化单元210、逆变换单元212和环路滤波单元220,分别对应视频编码器100的反量化单元110、逆变换单元112和环路滤波单元120。

[0061] 熵解码单元204用于对接收到的编码视频码流进行解码,并因此获得经量化的残差变换系数209和信令信息。将经量化的残差变换系数209提供给反量化单元210和逆变换单元212,以生成残差(逆变换)块。在重建单元214中将残差块添加到预测块265中,并将该块添加发送给环路滤波单元220以获得解码视频。解码视频的帧可存储在解码图像缓冲器230中并用作解码图像231进行帧间预测。

[0062] 通常,图1和图2的帧内预测单元154和254可以使用来自已编码区域的参考像素点来为需要编码或需要解码的块生成预测信号。

[0063] 熵解码单元204接收编码码流171作为其输入。通常,首先解析所述码流,即从码流中提取信令参数和残差。通常,所述码流的语法和语义由标准定义,使得编码器和解码器可互通。如上文背景部分所述,编码码流不仅包括预测残差。在进行运动补偿预测的情况下,还将运动矢量指示编码到码流中并在解码器侧从码流中解析。运动矢量指示可以通过提供运动矢量的参考图像和运动矢量坐标来给出。到目前为止,已经考虑了对完整的运动矢量进行译码。然而,也可以仅对码流中当前运动矢量与前一运动矢量之间的差值进行编码。这种方法可以利用邻块的各运动矢量之间的冗余。

[0064] 为了对参考图像进行有效译码,H.265编解码器(ITU-T,H265,H系列:视听和多媒

体系统:高效视频译码)提供参考图像列表,该参考图像列表为相应的参考帧分配列表索引。然后,通过在码流中包括对应的分配列表索引来指示参考帧。此类列表可以在标准中定义,或在视频的开头或许多帧的集合开始处指示。应注意,在H.265中定义了两个参考图像列表,称为参考列表L0和L1。然后,通过指示参考列表(L0或L1)并指示该列表中与所需参考图像相关联的索引,在码流中指示参考图像。提供两个或多个列表可能具有更好的压缩效果。例如,参考列表L0可以用于单向帧间预测条带和双向帧间预测条带,而参考列表L1可以仅用于双向帧间预测条带。然而,通常,本发明不限于列表L0和L1的任何内容。

[0065] 参考列表L0和L1可以在标准中定义并且是固定的。然而,可以通过在视频序列的开始处指示这些参考列表来实现编码/解码的更高灵活性。因此,编码器可以用根据索引排序的特定参考图像配置列表L0和L1。参考列表L0和L1可以具有相同的固定大小。通常,可以有多个列表。可以直接通过参考图像中的坐标来指示运动矢量。或者,H.265中还规定,可以构造候选运动矢量的列表,并且可以传输列表中特定运动矢量相关联的索引。

[0066] 当前块的运动矢量通常与当前图像或先前译码图像中邻块的运动矢量相关联。这是因为各邻块可能与具有相似运动的同一运动对象相对应,并且该对象的运动不太可能随时间而突然改变。因此,使用邻块中的运动矢量作为预测值减小了所指示的运动矢量差值的大小。运动矢量预测值(Motion Vector Predictor,MVP)通常从共址图像中空域相邻块或时域相邻块中已经编码/解码的运动矢量导出。在H.264/AVC中,通过获得三个空域相邻运动矢量的逐分量中值来导出运动矢量预测值。使用这种方法,不需要预测值的信令。共址图像中的时间MVP仅在H.264/AVC的所谓的时间直接模式中考虑。H.264/AVC直接模式也用于推导除运动矢量之外的其它运动数据。因此,这些直接模式与HEVC中的块融合概念更相关。在HEVC中,隐式推导MVP的方法被称为运动矢量竞争的技术所取代,该技术显式指示MVP列表中用于运动矢量推导的MVP。HEVC中的可变译码四叉树块结构可以导致一个块具有多个邻块,这些邻块具有作为潜在MVP候选的运动矢量。以左邻块为例,在最坏的情况下,当不进一步划分 64×64 亮度译码树块并且将左邻块划分到最大深度时, 64×64 亮度预测块在左侧可以具有16个 4×4 亮度预测块。

[0067] 引入先进的运动矢量预测(Advanced Motion Vector Prediction,AMVP)来修改运动矢量竞争以解释这种柔性块结构。在HEVC的开发过程中,大大简化了最初的AMVP设计,从而在译码效率与有利于实现的设计之间达到了良好的权衡。AMVP的初始设计包括来自三种不同预测值的五个MVP:来自空域相邻块的三个运动矢量、三个空域预测值的中值和来自共址的时域相邻块的缩放运动矢量。此外,通过重新排序,将最可能的运动预测值置于第一位置,并删除冗余候选项,对预测值列表进行了修改,以确保信号开销最小。AMVP候选列表构造的最终设计包括以下两个MVP候选项:(a)最多两个空间候选MVP,从五个空域相邻块中导出;(b)当两个空间候选MVP不可用或两者相同时,一个时间候选MVP,从两个时间共址块中导出;和(c)当空间候选项、时间候选项或两者都不可用时,零运动矢量。关于运动矢量确定的细节可以见于V.Sze等人(编)于2014年发表、施普林格(Springer)出版的书:《高效视频译码(HEVC):算法和架构(High Efficiency Video Coding(HEVC): Algorithms and Architectures)》,特别是第5章,其以引用的方式并入本文。

[0068] 为了进一步改进运动矢量估计而不进一步增加信令开销,进一步修正在编码端导

出并在 码流中提供的运动矢量可以有益于改进。可以在解码器侧执行运动矢量修正,而无需编码器的协助。解码环路中的编码器可以使用相同的修正以获得对应的运动矢量。在包括参考图像的整数像素位置和分数像素位置的搜索空间中执行运动矢量修正。例如,分数像素位置可以是半像素位置、四分之一像素位置或其它分数位置。分数像素位置可以通过如双线性插值等 插值从整数(全像素)位置获得。

[0069] 在当前块的双向预测中,将分别使用参考列表L0的第一运动矢量和参考列表L1的第二 运动矢量获得的两个预测块组合为单个预测信号,双向预测对原始信号的适应性可以比单向 预测更好,使残差信息更少,压缩效率可能更高。

[0070] 在解码器侧,当前块因为正在解码而不可用,为了进行运动矢量修正,使用模板,该模板为当前块的估计值,并根据已处理的(即,在编码端编码并在解码端解码)图像部分构建。

[0071] 首先,在解码器200侧接收第一运动矢量MV0的估计值和第二运动矢量MV1的估计值 作为输入。在编码器100侧,运动矢量估计值MV0和MV1可以通过块匹配和/或通过候选列表(如融合列表)中搜索获得,其中,所述候选列表由(在同一图像中或相邻图像中)与 当前块相邻的块的运动矢量形成。然后,将MV0和MV1在码流内向解码端指示。然而,需要说明的是,通常,编码器侧的第一确定阶段也可以通过模板匹配来执行,这将有利于减少 信令开销。

[0072] 在解码器200侧,有利地根据码流中的信息获得运动矢量MV0和MV1。对运动矢量MV0 和MV1进行直接指示,或进行差分指示,和/或指示运动矢量列表(融合列表)中的索引。然而,本发明不限于在码流中指示运动矢量。而是,根据本发明,运动矢量可以通过在运动矢量估计的第一阶段中已有的模板匹配来确定,对应于编码器的操作。可以根据与第二运动 矢量修正阶段的搜索空间不同的搜索空间进行第一阶段(运动矢量推导)的模板匹配。具体地,可以对具有较高分辨率(即,搜索位置之间的距离较短)的搜索空间进行运动矢量修正。

[0073] 还向解码器提供相应MV0和MV1所指向的两个参考图像RefPic0和RefPic1的指示。参考图像作为先前处理(即,相应的编码和解码)的结果存储在编码器和解码器侧的解码图像缓冲器中。通过搜索选择这些参考图像之一进行运动矢量修正。装置用于确定运动矢量的 参考图像选择单元用于选择MV0所指向的第一参考图像和MV1所指向的第二参考图像。在 选择之后,参考图像选择单元确定是第一参考图像还是第二参考图像用于运动矢量修正。为 了执行运动矢量修正,在运动矢量MV0所指向的候选位置周围定义第一参考图像中的搜索区 域。通过在搜索空间内执行模板匹配并确定相似度度量(如绝对差值和(sum of absolute difference, SAD)),分析搜索区域内的候选搜索空间位置,以查找与模板块最相似的块。搜 索空间的位置表示模板左上顶点所匹配的位置。如上所述,左上顶点仅仅是约定,搜索空间 的任何点(如中心点)通常可以用来表示匹配位置。

[0074] 图4A示出了也适用于单向预测的替代模板匹配。详细信息见文献JVET-A1001,特别是 第2.4.6节:Jianle Chen等人所著的题为“联合探索试验模型1的算法描述(Algorithm Description of Joint Exploration Test Model 1)”的文献JVET-A1001的“模式匹配的运动矢量 推导”,网址:<http://phenix.it-sudparis.eu/jvet/>。这种模板匹配方法中的模板确定为当前帧中 与当前块相邻的像素点。如文献JVET-A1001的图1所示,

可取与当前块的上边界和左边界 相邻的已重建像素点,称为“L形模板”。

[0075] 根据文献JVET-D0029 (该文献以引用的方式并入),解码端运动矢量修正(decoder-side motion vector refinement,DMVR)具有指向两个相应的参考图像RefPict0和RefPict1的初始 运动矢量MV0和MV1作为输入。这些初始运动矢量用于确定RefPict0和RefPict1中各自的 搜索空间。此外,使用运动矢量MV0和MV1,如下根据MV0和MV1所指向的(像素点) 各自的块A和B构造模板:

[0076] 模板=函数(块A,块B)。

[0077] 所述函数可以是结合逐像素点加权求和的像素点裁剪操作。然后,使用模板在根据相应 的参考图像Refpic 0和Refpic 1中的MV0和MV1确定的搜索空间中进行模板匹配。用于确定相应搜索空间中的最佳模板匹配的代价函数为SAD(模板,块candA'),其中,块candA' 为在跨越MV0给定的位置的搜索空间中候选MV所指向的候选译码块。图3示出了确定最佳匹配块A' 和所得修正运动矢量MV0'。因此,如图3所示,使用相同的模板来查找最佳匹配块B' 和指向块B' 的对应运动矢量MV1'。换句话说,在根据初始运动矢量MV0和MV1指向的块A和B构造模板之后,利用模板对RefPic0和RefPic1进行搜索来查找修正运动矢量MV0' 和MV1'。

[0078] 运动矢量推导技术有时也被称为帧率上转换(frame rate up-conversion,FRUC)。通常可以在码流中指示初始运动矢量MV0和MV1,以确保编码器和解码器可以使用相同的初始点进行运动矢量修正。或者,可以通过提供包括一个或多个初始候选项的初始候选列表来获得 初始运动矢量。对于每个初始运动矢量,确定修正运动矢量,并在最后选择代价函数最低的 修正运动矢量。

[0079] 如上所述,模板匹配的运动矢量推导模式是一种基于帧率上转换(Frame-Rate Up Conversion,FRUC)技术的特殊融合模式。在该模式下,在解码端导出块的运动信息。根据文献JVET-A1001 (“联合探索试验模型1的算法描述(Algorithm Description of Joint Exploration Test Model 1)”,网址:<http://phenix.it-sudparis.eu/jvet/>)中描述的具体实现方式,当融合标志为真时,为CU或PU指示FRUC标志。当FRUC标志为假时,指示融合索引并 使用常规融合模式。当FRUC标志为真时,指示附加的FRUC模式标志以指示推导块的运动信息所使用的方法(双边匹配或模板匹配)。

[0080] 综上所述,在运动矢量推导过程中,首先根据双边匹配或模板匹配为整个预测单元(Prediction Unit,PU)推导初始运动矢量。首先,生成MV候选列表,例如,该列表可以是PU的融合列表。检查列表,选择匹配代价最小的候选项作为起始点(初始运动矢量)。然后,根据双边匹配或模板匹配在起点周围进行局部搜索,并将匹配代价最小的一个或多个运动矢量(Motion Vector,MV)作为PU的MV。然后以导出的PU运动矢量为起点,进一步修正运动信息。术语“预测单元(prediction unit,PU)”和“译码单元(coding unit,CU)”在本文中可互换使用,以描述图像(帧)内的像素点块。

[0081] 如图4B所示,使用双边匹配(在文献JVET-A1001中有所描述),通过在两个不同参考图像中查找沿当前CU的运动轨迹的两个块之间的最接近匹配来推导当前CU的运动信息。在假设运动轨迹连续的情况下,在当前图像与两个参考图像之间,指向两个参考块的运动矢量MV0和MV1应与时间距离(即TD0和TD1)成正比。因此,在本方面的一个实施例中,在每个测试的候选矢量对中,两个相应的矢量在图像平面中的直线上。作为一种特殊情况,

当当前图像在时间上位于两个参考图像之间,且当前图像到两个参考图像的时间距离相同时,双边匹配成为基于镜像的双向MV。

[0082] 如图4A所示,使用模板匹配(在文献JVET-A1001中有所描述),通过查找当前图像中的模板(当前CU的顶部块和/或左侧邻块)与参考图像中的块(与模板大小相同)之间的最接近匹配来推导当前CU的运动信息。文献JVET-A1001的“模式匹配的运动矢量推导”部分描述了模板匹配和双边匹配方法的具体实现方式。一个示例公开了只有在“融合标志”为真时,才应用双边匹配操作,从而指示选择了“块融合”操作模式。这里,文献JVET-A1001的作者参考了H.265标准的“融合模式”。需要说明的是,JVET-A1001中描述的模板匹配和双边匹配方法也可以应用于其它视频译码标准,导致具体实现方式发生变化。

[0083] 图5为解码端运动矢量修正(decoder-side motion vector refinement,DMVR)操作的流程图。根据文献JVET-D0029,在以下两种情况下应用DMVR:(1)预测类型设置为跳过模式或融合模式;(2)预测模式为双向预测。导出(参考列表L0的)第一初始运动矢量MV0和(参考列表L1的)MV1。根据相应的跳过和融合操作执行推导过程。这里,文献JVET-D0029的作者参考了H.265标准的跳过模式和融合模式。这些模式的描述可以见于:v.Sze、M. Budagavi和G.J.Sullivan(编)2014年发表的书《高效视频译码(HEVC),算法与体系结构(High Efficiency Video Coding(HEVC),Algorithms and Architectures)》的第5.2.2.3节“融合运动数据指示和跳过模式”。在H.265中,如果使用跳过模式对块指示推断而不是显式指示运动数据,并且预测残差为零,即不传输变换系数。如果选择融合模式,则也推断运动数据,但预测残差不为零,即,显示指示变换系数。

[0084] 从输入视频流中解析(510)出解析索引。解析的索引指向构造(520)的MV候选列表的最佳运动矢量候选项。然后,选择(530)最佳运动矢量候选项,并通过加权平均(540)获得模板。如下应用DMVR(550)。通过将MV0和MV1引用的块相加来计算块模板,如上文结合图3所述。之后进行裁剪。使用所述模板查找初始运动矢量MV0周围的修正运动矢量MV0'。搜索区域为整数像素分辨率(搜索空间的各点之间以整数像素点距离间隔开)。使用绝对差值和(Sum of Absolute Difference,SAD)代价度量来比较模板块和MV0'所指向的新块。使用该模板查找MV0'周围的修正MV0"。搜索区域为半像素分辨率(搜索空间的各点之间以半个像素点距离间隔开)。术语“像素(pixel)”和“像素(pixel)”在本文中可互换使用。使用相同的代价度量。重复后两个步骤以查找MV1"。通过将MV0"和MV1"所指向的块相加来形成新的双向预测块。然后,对此类修正运动矢量MV0"和MV1"所指向的块block_A'和block_B'取平均值,例如进行加权平均(560),以获得最终预测。

[0085] 图6为本发明一实施例提供的可用于视频编码和解码的局部照明补偿(local illumination compensation,LIC)的示例的示意图。局部照明补偿(Local Illumination Compensation,LIC)基于光照变化的线性模型,使用缩放因子“a”和偏移量“b”。对于每个帧间模式译码的译码单元(coding unit,CU),可以自适应地启用或禁用LIC。当对CU应用LIC时,可以使用最小二乘误差法,利用当前CU的相邻像素点及其对应的参考像素点推导参数a和b。更具体地,如图6所示,使用CU的子采样(2:1子采样)相邻像素点和参考图像中的对应像素点(通过当前CU或子CU的运动信息标识)。分别对每个预测方向推导和应用LIC参数。这里,子采样2:1意味着在当前CU边界和参考块上每两个像素子采样一次。有关使用缩放因子/乘法加权因子和偏移量的更多细节可见于文献JVET-A1001的“第2.4.4节局部

照明补偿”。

[0086] 图7为对参考图像RefPic0进行的解码端运动矢量修正(decoder-side motion vector refinement, DMVR)迭代的示意图。当前图像包括当前块710,针对该当前块710,应根据 RefPic0中的运动矢量MV0找到运动矢量MV0'。确定包括5个整数位置的搜索空间;候选位置所指向的块称为Ax。输出为运动矢量MV0'所指向的块Ax的最佳匹配。

[0087] 每当指示显式融合模式索引时,解码端运动矢量修正从一个或多个运动矢量和从指示索引中规范推导的参考索引开始。当没有指示显式融合模式索引时,在解码器侧使用代价函数评估初始运动矢量候选项,并且选择代价最低的候选项作为修正的起点。因此,无论解码端运动矢量推导方法是基于预测/重建的邻块边界像素点(通常称为模板匹配(Template Matching, TM),参考图4A)(还是)基于通过参考列表L0和参考列表L1中对应连通区域之间的差值(通常称为双边匹配(bilateral matching, BM)代价)最小化进行的双边匹配(还是)基于参考列表L0和参考列表L1中对应连通区域的平均版本与L0/L1的移位之间的差值(称为DMVR代价),都需要围绕可以是子像素精确运动矢量的起点进行修正搜索。

[0088] 为了评估代价函数,需要执行插值,以根据在整数网格位置处参考帧的值导出子像素精确中心的值。插值滤波器可以如双线性插值滤波器一样简单,也可以是如基于二维DCT的可分离插值滤波器等更长的滤波器。为了降低在修正期间考虑的每个位置处反复导出块的插值像素点的复杂度,另一发明中已提出L0/或L1中以一个或多个子像素精确位置为中心的修正点的整数像素距离网格。利用这一点,当考虑接近当前最佳代价位置的新位置,只需要执行增量插值。在对整数像素距离网格进行修正之后,获得相对融合MV最佳的整数增量MV。

[0089] 为了进一步提高压缩增益,可以进行子像素距离修正。半像素距离修正要求插值像素点在与一个或多个参考帧中的一个或多个最佳整数距离MV位置相距半像素的位置处。可以在L0与L1之间联合执行子像素精确修正,其中,对L0中相对于L0中的最佳整数距离MV位置的位移取反,以获得L1中相对于L1中的最佳整数距离MV位置的对应位移。还可以在L0和L1中独立地执行子像素精确修正。

[0090] 图8为本发明一实施例提供的整数像素点位置和分数像素点位置的示例的概念图。参见图8,表示为“A”的像素位置为整数像素;半像素位置表示为b、h和j;所有四分之一像素位置表示为a、c、d、e、f、g、I、k、n、p、q和r。为了使用8点正方形图案进行半像素修正,需要3个平面,类似地,根据HEVC标准,四分之一像素修正需要8个平面。新标准正在考虑1/16像素精确插值。

[0091] 当“A”为整数像素时,像素位置a、b和c只需要水平插值,像素位置d、h和n只需要垂直插值,其它所有点都需要垂直插值和水平插值。

[0092] 在整数像素距离修正和子像素距离修正过程中,可以使用与用于最终运动补偿预测过程的插值不同的插值过程。例如,双线性插值是可用于修正的更简单插值,而最终运动补偿预测(motion compensated prediction, MCP)可能需要基于二维DCT的插值滤波器。从参考帧缓冲器获取的整数像素网格数据需要保持在内部存储器(例如,SRAM)中的第一缓冲区中,直到可以完成所有插值,以避免多次从外部存储器(例如DDR)重新获取该数据。需要将通过整数像素网格像素点的插值导出的、从子像素精确修正中心开始进行整数

像素距离修正所需的整数像素距离网格保持在内部存储器中的第二缓冲区中,直到所有整数像素距离修正完成。对于最佳整数像素距离位置(其可以是相对于整数像素网格的子像素位置)周围的半像素距离修正,鉴于中心任一侧的对称半像素修正点之间的距离相隔1个整数像素,2个水平半像素距离位置共享插值平面,2个垂直半像素距离位置共享插值平面,4个对角半像素距离位置共享插值平面。

[0093] 当用于修正的插值与最终MCP插值相同时,可以优选保留存储在第二缓冲区中的整数像素距离修正数据,因为当最佳整数像素距离位置在经过子像素修正后成为最佳位置时,可以避免最终MCP插值计算。当修正与最终MCP之间的插值不同时,可以用半像素距离修正所需的三个平面之一的插值覆盖第二缓冲区。

[0094] 为了减少计算负载和内部存储需求,一些现有技术仅评估4个对角线半像素距离位置,而另一些现有技术仅评估水平和垂直的半像素距离位置。因此,所需的内部存储器的缓冲区组在2个(当仅评估对角半像素距离位置且插值不同时)到5个(当对所有三个半像素距离平面进行评估且插值相同时)之间。虽然一些现有技术执行额外的四分之一像素距离和八分之一像素距离修正,但这些修正中的每个位置都需要单独的插值,这在计算上不可行,并且与附加的压缩增益不相称。当插值采用较多的抽头滤波器时,首先进行水平滤波,将滤波结果存储在临时缓冲区中,再进行垂直滤波。

[0095] 本发明的实施例提供了如下方法和装置:用于在解码端运动矢量修正/推导过程中,通过利用在整数像素距离网格位置处评估的代价的误差面技术,在不进行显式修正的情况下,推导最佳整数距离运动矢量周围的子像素距离增量运动矢量。利用这些代价拟合参数误差平面,通过求解一组联立方程获得参数误差面的最小位置。本发明的各个方面利用整数距离像素位置(其自身可以是相对于整数像素网格的子像素位置)周围进行的解码端子像素距离修正,并处理解码端运动矢量修正的不同变型,如模板匹配、双边匹配和基于模板的双边匹配。

[0096] 鉴于解码端运动矢量修正/导出是译码系统的标准方面,编码器也需要执行相同的误差面技术,以免编码器重建与解码器重建之间有任何漂移。因此,本发明的所有实施例的各方面都适用于编码系统和解码系统。

[0097] 在模板匹配中,修正移动仅在参考中发生,从子像素精确中心开始,其中,所述子像素精确中心是基于显式指示的融合索引导出的或通过代价评估隐式导出的。

[0098] 在双边匹配(有或没有平均模板)中,所述修正从参考列表L0和L1中相应的子像素精确中心开始,其中,所述子像素精确中心是基于显式指示的融合索引导出或通过代价评估隐式导出的。

[0099] 应说明的是,当根据参考列表L1中相等且相反的水平 and 垂直位移,评估参考列表L0中给定的水平和垂直位移的双边匹配代价时,假设图中所示的位置对应参考列表L0,并通过对参考列表L1中相对于当前迭代中心在水平和垂直方向上的位移取反来获得参考列表L1中的位置。

[0100] 因此,需要在不增加存储器大小和计算复杂度的情况下,实现显式子像素距离修正的大部分译码增益。

[0101] 实施例1

[0102] 设N为规范上允许的整数1像素距离修正迭代的最大次数。第一整数距离修正迭代

从上述修正的子像素精确中心开始,继续进行直到中心位置的估计代价函数值低于1像素相邻位置的预定集合的估计代价函数值,或者直到已执行N次迭代。当达到N次迭代,且中心位置不是与其1像素相邻位置集合相比代价函数值最小(最低)的位置时,不执行基于误差面的子像素修正,并且声明所有迭代中代价函数值最小(最低)的位置为解码端运动矢量修正过程的最终增量运动矢量。当修正退出时,中心位置在1像素相邻位置的现有迭代集合中具有最小(最低)代价函数值,应用以下基于误差面的子像素修正过程。

[0103] 在下文中,设中心C周围的1像素相邻位置的集合在其左侧(L)、上方(T)、右侧(R)和下方(B)的整数1像素距离处。图9为本发明一实施例提供的,用于获得一个或多个参考帧中的子像素精确增量运动矢量修正,的中心像素周围的1像素相邻位置集合的框图。参考图9,二维坐标系的原点在C处,L、T、R、B处的坐标对应(-1,0)、(0,1)、(1,0)、(0,-1)。这5个位置的评估代价函数值为E(0,0)、E(-1,0)、E(0,1)、E(1,0)和E(0,-1)。使用二维抛物线函数的5参数误差面定义如下:

$$[0104] \quad E(x,y) = A*(x-x_0)^2 + B*(y-y_0)^2 + C \quad (1)$$

[0105] 其中,E(x,y)为笛卡尔坐标(x,y)的估计代价函数;(x₀,y₀)为与子像素位移相对于中心(0,0)相关联(对应)的具有最小(最低)误差的笛卡尔坐标;C为与该位置处的误差对应的参数;A和B为误差面模型参数。A、B、C为计算E(x,y)的常数值。

[0106] 另一方面,考虑到A、B、C、x₀和y₀为未知参数,利用5个位置(0,0)、(-1,0)、(0,-1)、(1,0)和(0,1)处的可用估计代价函数值求解5个方程,(x₀,y₀)可以计算(确定)如下:

$$[0107] \quad x_0 = (E(-1,0) - E(1,0)) / (2*(E(-1,0) + E(1,0) - 2*E(0,0))) \quad (2)$$

$$[0108] \quad y_0 = (E(0,-1) - E(0,1)) / (2*(E(0,-1) + E(0,1) - 2*E(0,0))) \quad (3)$$

[0109] x₀和y₀为与子像素运动矢量位移相关联的坐标。E(-1,0)、E(1,0)、E(0,0)、E(0,-1)和E(0,1)为整数距离代价,所述整数距离代价分别与相对于初始运动矢量的候选整数运动矢量位移(-1,0)、(1,0)、(0,0)、(0,-1)和(0,1)相对应。坐标x₀和y₀可以根据增量运动矢量的子像素精度,通过缩放因子N缩放,其中,例如,对于子像素精度1/2、1/4、1/8和1/16,N分别为2、4、8和16。

[0110] 在另一实施例中,等式(2)和(3)可以表示为:

$$[0111] \quad x_0 = (E(-1,0) - E(1,0)) / (2*N*(E(-1,0) + E(1,0) - 2*E(0,0))) \quad (2')$$

$$[0112] \quad y_0 = (E(0,-1) - E(0,1)) / (2*N*(E(0,-1) + E(0,1) - 2*E(0,0))) \quad (3')$$

[0113] 其中,例如,对于像素精度1/2、1/4、1/8和1/16,N分别为1、2、4和8。

[0114] 根据等式(2)和(3),需要说明的是,x₀可以只根据位置(-1,0)、(1,0)和(0,0)计算(确定)。类似地,y₀可以只根据位置(0,-1)、(0,1)和(0,0)计算(确定)。

[0115] 还需说明的是,当所有4个位置(-1,0)、(0,-1)、(1,0)和(0,1)相对于中心(0,0)的评估代价(与相对于初始运动矢量的最佳整数距离位移对应)不可用时,无法拟合参数误差面。在这类情况下,不将子像素精确的增量位移加到最佳整数距离位移。

[0116] 根据最终运动补偿支持的子像素精度,可以适当地选择缩放因子N。

[0117] 根据等式(1)、(2)和(3),子像素运动矢量位移在x和y方向上受-0.5和+0.5限定,并且由于通常关注高达1/16像素的精度,只需移位、比较和增量操作,就可以有效地进行2次划分。

[0118] 在一些实施例中,用于解码端运动矢量修正的方法包括:通过比较相对于初始运

动矢量 的候选整数运动矢量位移对应的整数距离代价,确定目标整数运动矢量位移;通过计算所述 整数距离代价,确定子像素运动矢量位移;根据所述目标整数运动矢量位移、所述子像素运动矢量位移和所述初始运动矢量计算修正运动矢量。

[0119] 在一个实施例中,所述方法还可以包括:在确定子像素运动矢量位移之前,确定预定运动矢量位移是否包括目标整数运动矢量位移;在所述预定运动矢量位移包括所述目标整数运动矢量位移的情况下:通过计算所述整数距离代价确定所述子像素运动矢量位移。在另一实施例中,所述方法还可以包括:在确定子像素运动矢量位移之前,确定相对于目标整数运动 矢量位移在位置 $(-1, 0)$ 、 $(0, -1)$ 、 $(1, 0)$ 和 $(0, 1)$ 处的评估代价是否可用;在确定目标整数运动矢量位移在位置 $(-1, 0)$ 、 $(0, -1)$ 、 $(1, 0)$ 和 $(0, 1)$ 的评估代价可用的情况下,通过计算整数距离代价,确定子像素运动矢量位移。

[0120] 在一个实施例中,所述方法还可以包括:在预定运动矢量位移不包括目标整数运动矢量 位移的情况下:根据目标整数运动矢量位移和初始运动矢量计算修正运动矢量。在另一实施例中,所述方法还可以包括:在确定相对于目标整数运动矢量位移在位置 $(-1, 0)$ 、 $(0, -1)$ 、 $(0, 1)$ 和 $(1, 0)$ 处至少一个或多个评估代价不可用的情况下,根据目标整数运动矢量位移和初始运动矢量计算修正运动矢量。

[0121] 在一个实施例中,确定目标整数运动矢量位移可以包括:计算每个候选整数运动矢量位移的整数距离代价;选择与最小整数距离代价对应的候选整数运动矢量位移,作为目标整数 运动矢量位移。

[0122] 在一个实施例中,目标整数运动矢量位移包括与参考图像列表L0对应的第一运动矢量位移,和与参考图像列表L1对应的第二运动矢量位移。所述方法还可以包括:通过比较对应与 参考图像列表L0对应的候选整数运动矢量位移的整数代价确定第一运动矢量位移;通过对第一运动矢量位移取反确定第二运动矢量位移。

[0123] 应当理解,所描述的实施例为示例性的,不具有限制性,如果整数距离修正过程在1像素相邻集合内产生多个这些评估代价函数值,则可以使用参考文献中其它基于5点、6点和9 点的误差面方法。

[0124] 根据本实施例的方法只需要2个内部存储器缓冲区。第一缓冲区用于存储来自重建图像 缓冲器的整数像素网格;第二缓冲区用于存储整数距离修正(用于获得参数误差面)期间的 整数距离网格,并且可用于用最终子像素精确运动补偿预测像素点覆盖。

[0125] 根据本发明的实施例,在任一子像素精度级别上不进行显式修正的情况下获得高精度的 子像素精确增量运动矢量。

[0126] 当在所有迭代中使用相同的代价函数时,除了整数距离修正迭代所需的计算外,不需要 额外的插值或代价函数求值计算。

[0127] 图10为本发明一个实施例提供的用于在解码端运动矢量修正系统中获取一个或多个参 考帧中的子像素精确增量运动矢量的方法1000的简化流程图。方法1000可以包括以下步骤:

[0128] 步骤1001:提供处理器。处理器可以是集成在视频编码器和/或解码器中的一个或多个处 理单元(CPU、DSP),或者集成在视频压缩软件中用于执行本文所描述的方法的程序代码。

[0129] 步骤1003:所述处理器使用代价函数迭代地执行整数1像素距离修正操作(迭代),

以 确定所述一个或多个参考帧中每个参考帧的整数距离修正运动矢量,从而启动循环。

[0130] 步骤1005:所述处理器针对1像素相邻位置集合的代价函数值,确定当前操作的搜索中心位置的代价函数值。在确定当前迭代的搜索中心的代价为最低,即提前退出迭代循环(1005, 是)的情况下:

[0131] 步骤1011:通过使用最后搜索中心和该最后搜索中心周围的1像素相邻位置的代价函数 值计算在拟合的参数误差面上值最小的位置,确定每个参考中最后搜索中心周围的子像素距 离修正运动矢量;

[0132] 步骤1013:返回总修正运动矢量,作为每个参考帧的所述确定的整数距离修正运动矢量和所述确定的子像素距离修正运动矢量的和。

[0133] 所述方法还包括:在所述搜索中心位置的代价函数值不是最低代价函数值(1005, 否)的情况下,执行步骤(1007)。

[0134] 步骤(1007):确定当前操作为最后一次操作(迭代)。在当前操作为最后一次操作(1007, 是)的情况下,返回与所有操作中代价值最低的位置对应的修正运动矢量(1015)。在当前操作不是最后一次操作(1007, 否)的情况下:将中心更新到该最后一次操作中代价最低的位置,并返回循环并重复步骤(1003)。

[0135] 实施例2

[0136] 当用于推导出参数误差面的整数距离位置处的代价函数求值不同于在整数距离修正的早期迭代期间执行的代价函数求值时,存在实施例1的不同变型。

[0137] 例如,当使用双边匹配时,可以有三种修正。第一种修正类型是在参考列表L0和L1两者中执行联合修正,使得参考列表L1中的位移等于参考列表L0中的位移并在水平和垂直方向上与其相反(第一种修正类型称为SBM_JOINT)。第二种修正类型是在参考列表L0和L1两者中针对共同的双边平均模板执行独立修正(第二种修正类型称为TBM_INDEPENDENT)。第三种修正类型是在参考列表L0或L1中针对双边平均模板执行修正,并在获得另一个参考列表中的位移时对估计的水平和垂直位移取反(如果L0用于修正,则该另一个参考列表为 L1;如果L1用于修正,则该另一个参考列表为L0。第三种修正类型称为TBM_IN_1REF_NEG_IN_OTHER)。

[0138] 由于难以预测迭代从迭代循环提前退出(由于中心位置的代价低于周围1像素相邻位置的代价),当用于推导参数误差面的整数距离位置处的代价函数求值不同于在整数距离修正的 先前迭代期间执行的代价函数求值时,利用所需代价函数执行以提前退出中心为中心的额外 一组代价函数求值。

[0139] 下表1提供了在双边匹配过程中的一些情况,其中在整数距离修正过程中,在提前退出 之后需要额外求值。

[0140] 表1

[0141]	用于整数距离修正的修正类型	用于整数距离修正的最后一次迭代（或）提前退出，以推导参数误差面的修正类型	提前退出之后是否需要额外的代价函数求值？
	SBM_JOINT	SBM_JOINT	否
	SBM_JOINT	TBM_INDEPENDENT	是
	SBM_JOINT	TBM_IN_1REF_NEG_IN_OTHER	是
	TBM_INDEPENDENT	TBM_INDEPENDENT	否
	TBM_IN_1REF_NEG_N_OTHER	TBM_INDEPENDENT	是

[0142] 据观察,修正类型SBM_JOINT在整数距离修正迭代中效果最好。当使用修正类型TBM_INDEPENDENT获得L0和L1中的独立子像素增量运动矢量时,与L0和L1中相等且相反的位移相比,可获得较小的额外译码增益。然而,L0和L1中的独立修正要求在L0和L1位置独立进行代价评估,因此计算比联合的相等和相反的位移估计选择(SBM_JOINT修正)更为复杂。除了SBM_JOINT修正之外,提前退出不需要额外的代价函数求值。

[0143] 图11为本发明一个实施例提供的用于在解码端运动矢量修正系统中,获取一个或多个参考帧中子像素精确增量运动矢量的方法1100的简化流程图。所述一个或多个参考帧在相应的一个或多个初始子像素精确修正中心周围。方法1100可以包括以下步骤:

[0144] 步骤1101:提供处理器。处理器可以是集成在视频编码器和/或解码器中的一个或多个处理单元(CPU、DSP),或者集成在视频压缩软件中用于执行本文所描述的方法的程序代码。

[0145] 步骤1103:所述处理器使用代价函数迭代地执行整数1像素距离修正操作(迭代),以确定所述一个或多个参考帧中每个参考帧的整数距离修正运动矢量。

[0146] 步骤1105:确定当前操作是否为最后一次操作。在当前操作是最后一次操作(1105,是)的情况下:

[0147] 步骤1111:使用第二代价函数进行整数1像素距离修正,以获得一个或多个参考帧中每个参考帧的独立的子像素增量运动矢量。

[0148] 在当前操作不是最后一次操作(1105,否)的情况下,执行步骤1107。

[0149] 步骤1107:确定当前操作的搜索中心位置的代价函数值是否为1像素相邻位置集合中的最低代价函数值。在搜索中心位置的代价函数值为最低代价函数值(1107,是)的情况下:执行步骤1111(即,提前退出迭代循环)。在搜索中心位置的代价函数值不是最低代价函数值(1107,否)的情况下:执行步骤1109。

[0150] 步骤1109:将中心更新到该操作中代价值最低的位置,并返回循环以执行下一次整数1像素距离修正操作。

[0151] 步骤1111:使用第二代价函数进行整数1像素距离修正,以获得一个或多个参考帧中每个参考帧的独立的子像素增量运动矢量。

[0152] 步骤1113:针对最后搜索中心的1像素相邻位置集合处的第二代价函数值,确定最后搜索中心位置是否具有最低的第二代价函数值。在最后搜索中心位置具有最低的第二代价函数值(1113,是)的情况下:

[0153] 步骤1115:通过使用第二代价函数值计算在拟合的参数误差面上具有最小值的位置,确定每个参考帧中最佳整数距离修正位置周围的子像素距离修正运动矢量。

[0154] 步骤1117:返回总修正MV,作为每个参考帧的所述确定的整数距离修正MV和所述确定的子像素距离修正MV的和。

[0155] 本发明的实施例还提供了用于执行上述方法的装置。所述装置可以是软件和硬件的组合。例如,可由如通用处理器(general purpose processor,CPU)、或数字信号处理器(digital signal processor,DSP)、或现场可编程门阵列(field programmable gate array,FPGA)等芯片执行编码和/或解码。然而,本方面的实施例并不限于可编程硬件实现。本发明的一些实施例可以使用专用集成电路(application-specific integrated circuit,ASIC),或上述硬件组件的组合来实现。

[0156] 编码和/或解码也可以通过存储在计算机可读介质中的程序指令或程序代码来实现。处理器或计算机执行所述程序指令时,所述处理器或计算机执行上述方法的步骤。所述计算机可读介质可以是存储程序代码的任何介质,如DVD、CD、USB(闪存)驱动器、硬盘、可通过网络访问的服务器存储器等。

[0157] 图12为可用于实现本发明各种实施例的装置1200的框图。装置1200可以是图1所示的编码装置100和图2所示的解码装置200。此外,装置1200可以容纳一个或多个所描述的元件。在一些实施例中,装置1200配备有一个或多个输入/输出设备,如扬声器、麦克风、鼠标、触摸屏、小键盘、键盘、打印机、显示器等。装置1200可包括连接到总线的一个或多个中央处理单元(central processing unit,CPU)1210、存储器1220、大容量存储器1230、视频适配器1240和I/O接口1260。所述总线是任何类型的若干总线架构中的一种或多种,包括内存总线或内存控制器、外围总线、视频总线等。

[0158] CPU 1210可具有任何类型的电子数据处理器。存储器1220可具有或者可以是任何类型的系统存储器,如静态随机存取存储器(static random access memory,SRAM)、动态随机存取存储器(dynamic random access memory,DRAM)、同步DRAM(synchronous DRAM,SDRAM)、只读存储器(read-only memory,ROM)、其组合等。在实施例中,存储器1220可包括在开机时使用的ROM以及在执行程序时使用的存储程序 and 数据的DRAM。在一些实施例中,存储器1220可以是非瞬时性的。大容量存储器1230可以包括存储数据、程序和其它信息并使数据、程序和其它信息可通过总线访问的任何类型的存储设备。例如,大容量存储器1230包括用于存储程序代码的固态硬盘、硬盘驱动器、磁盘驱动器、光盘驱动器等中的一种或多种,其中,CPU 1210执行所述程序代码时,所述CPU执行本文所述的方法。CPU 1210可以用于迭代地执行结合图10和11所述的步骤的多个操作。CPU 1210可以包括参考选择单元,用于选择MV0所指向的第一参考图像和MV1所指向的第二参考图像。在选择参考图像之后,参考图像选择单元可以确定是第一参考图像还是第二参考图像用于执行运动矢量修正。

[0159] 视频适配器1240和I/O接口1260提供接口以使外部输入和输出设备与装置1200耦合。例如,装置1200可向客户端提供SQL命令接口。如图所示,输入和输出设备的示例包括与视频适配器1240耦合的显示器1290和与I/O接口1260耦合的鼠标/键盘/打印机1270的任何组合。其它设备可与装置1200耦合,并且可使用额外的或更少的接口卡。例如,串行接口卡(未图示)可以用于为打印机提供串行接口。

[0160] 装置1200还可以包括一个或多个网络接口1250,或一个或多个网络1280,其中,网络接口1250包括如以太网线等有线链路,和/或用于接入节点的无线链路。网络接口1250允许装置1200通过网络1280与远程单元通信。例如,网络接口1250可提供与数据库的通信。在一个实施例中,装置1200与局域网或广域网耦合,以进行数据处理并与如其它处理单元、互联网、远程存储设施等远程设备进行通信。装置1200可用于对从输入端(例如,网络接口)接收的一个或多个图像块进行编码和/或对来自码流的视频图像进行解码。装置1200可包括:码流解析器,用于从码流中提取压缩图像块;运动矢量修正单元,用于获得子像素精确增量运动矢量修正;重建单元,用于根据获取的运动矢量进行块重建。

[0161] 尽管本发明的特定特征或方面可能已经仅结合几种实现方式或实施例中的一种进行公开,但此类特征或方面可以和其它实现方式或实施例中的一个或多个其它特征或方面相结合,只要对于任何给定或特定应用是有需要或有利的。此外,在一定程度上,术语“包括”、“有”、“具有”或这些词的其它变形在详细说明或权利要求书中使用,这类术语和术语“包括”是类似的,都是表示包括的含义。此外,术语“示例性”、“例如”仅表示为示例,而不是最好或最佳的。可以使用术语“耦合”和“连接”及其派生词。应理解,这些术语可以用于指示两个元件彼此协作或交互,而不管它们是直接物理接触还是电接触,或者它们彼此不直接接触。

[0162] 尽管本文中已说明和描述特定方面,但本领域的技术人员应了解,多种替代和/或等效实现方式可在不脱离本发明范围的情况下替代所示和描述的特定方面。本申请旨在覆盖本文论述的特定方面的任何修改或变更。

[0163] 尽管以上权利要求书中的元件是利用对应的标签按照特定顺序列举的,除非对权利要求的阐述另有暗示用于实现部分或所有这些元件的特定顺序,否则这些元件不必限于以所述特定顺序来实现。

[0164] 通过以上教导,对于本领域技术人员来说,许多替代、修改和变化是显而易见的。当然,本领域技术人员容易认识到,除本文所述的应用之外,还存在本发明的众多其它应用。虽然已结合一个或多个特定实施例描述了本发明,但本领域技术人员认识到,在不脱离本发明范围的情况下,仍可对本发明作出许多改变。因此,应理解,只要是在所附权利要求书及其等效物的范围内,可以用不同于本文具体描述的方式来实践本发明。

[0165] 上述电路也可以是单个集成芯片。然而,本发明并不限于此,电路可以包括不同的部件或硬件,或硬件和软件的组合,如用对应代码编程的通用处理器或DSP。

[0166] 上述流程图旨在说明解码端运动矢量修正技术的示例。本领域技术人员可以在不脱离本发明的精神和范围的情况下修改或组合步骤来实现本发明。

[0167] 在本发明的一个方面中,提供了一种用于在解码端运动矢量修正系统中获得一个或多个参考帧中的子像素精确增量运动矢量修正的方法。所述方法可以包括:

[0168] 提供处理器(1001);

[0169] 所述处理器使用代价函数迭代地执行(1003)整数1像素距离修正操作,以确定所述一个或多个参考帧中每个参考帧的整数距离修正运动矢量,从而启动循环操作;

[0170] 所述处理器确定搜索中心位置的代价函数值是否为1像素相邻位置集合中的最低函数值(1005);

[0171] 在搜索中心位置的代价函数值是最低代价函数值(1005,是)的情况下:退出所述

循环 操作。

[0172] 通过使用搜索中心和1像素相邻位置集合的代价函数值计算在拟合的参数误差面上值最小的位置,确定每个参考帧中搜索中心周围的子像素距离修正运动矢量(1011);

[0173] 返回使用每个参考帧的参数误差面获得的子像素位置(1013)。

[0174] 在一个实施例中,所述方法还可以包括:在搜索中心位置的代价函数值不是最低代价函数值(1005,否)的情况下:确定当前操作是否为最后一次操作(1007);

[0175] 在当前操作是最后一次操作(1007,是)的情况下:

[0176] 返回每个参考框架中具有最低代价函数的位置(1015);

[0177] 在当前操作不是最后一次操作(1007,否)的情况下:

[0178] 所述处理器将当前整数1像素距离修正操作的搜索中心的位置更新(1009)为当前操作中代价最低的位置;

[0179] 重复所述循环操作(1003、1005、1007、1009)。

[0180] 在一个实施例中,参数误差面包括5个像素,布置成中心像素由与中心像素以等距离间隔开的四个周边像素包围。

[0181] 在一个实施例中,通过以下表达式计算代价函数:

[0182] $E(x, y) = A * (x - x_0)^2 + B * (y - y_0)^2 + C$;

[0183] 其中, $E(x, y)$ 是坐标 (x, y) 的估计代价函数值; x_0 和 y_0 是与相对于中心 $(0, 0)$ 的子像素位移相关联的误差最小的坐标; C 为与坐标 (x_0, y_0) 处的误差关联的参数; A 和 B 为常数值; x 和 y 为与相邻位置相关联的坐标; x 和 y 各自为包括-1、0和1的整数。

[0184] 在一个实施例中,通过以下表达式获得子像素位置:

[0185] $x_0 = (E(-1, 0) - E(1, 0)) / (2 * N * (E(-1, 0) + E(1, 0) - 2 * E(0, 0)))$;

[0186] $y_0 = (E(0, -1) - E(0, 1)) / (2 * N * (E(0, -1) + E(0, 1) - 2 * E(0, 0)))$;

[0187] 其中,

[0188] $E(x, y) = A * (x - x_0)^2 + B * (y - y_0)^2 + C$;

[0189] $E(x, y)$ 是估计代价函数值; x_0 和 y_0 是与相对于中心 $(0, 0)$ 的子像素位移相关联的误差最小的坐标; C 为与坐标 (x_0, y_0) 处的误差关联的参数; A 和 B 为常数值; x 和 y 为与相邻位置相关联的坐标; x 和 y 各自为包括-1、0和1的整数;对于1/2、1/4、1/8或1/16的子像素精度, N 为整数1、2、4或8。

[0190] 在一个实施例中,通过模板匹配获得子像素精确增量运动矢量修正。

[0191] 在本发明的另一方面中,提供了一种用于在解码端运动矢量修正系统中获得一个或多个参考帧中的子像素精确增量运动矢量的方法。所述装置包括处理单元和其中具有计算机可读指令的非瞬时性计算机可读介质,所述非瞬时性计算机可读介质用于使所述处理单元执行以下步骤:

[0192] 使用代价函数迭代地执行 N 次整数1像素距离修正操作,以确定每个参考帧的整数距离修正运动矢量,从而启动循环操作,其中,用上一次操作中代价最低的位置更新当前迭代的搜索中心的位置;

[0193] 确定搜索中心位置的代价是否低于1像素相邻位置集合的任一代价;

[0194] 在搜索中心位置的代价为最低代价的情况下:

[0195] 退出循环操作;

- [0196] 通过使用搜索中心和1像素相邻位置集合的代价函数值计算在拟合的参数误差面上值最小的位置,确定每个参考帧中搜索中心周围的子像素距离修正运动矢量;
- [0197] 返回使用每个参考帧的参数误差面获得的子像素位置。
- [0198] 在一个实施例中,其中具有计算机可读指令的非瞬时性计算机可读介质进一步使处理单元执行以下步骤:
- [0199] 在搜索中心位置的代价函数值不是最低代价函数值(1005,否)的情况下:确定当前操作是否为最后一次操作;
- [0200] 在当前操作为最后一次操作的情况下:返回每个参考帧中代价函数最低的位置;
- [0201] 在当前操作不是最后一次操作的情况下:所述处理器将当前整数1像素距离修正操作的搜索中心的位置更新(1009)为当前修正操作中代价最低的位置;重复循环操作。
- [0202] 在一个实施例中,通过以下表达式计算代价函数:
- [0203] $E(x,y) = A*(x-x_0)^2 + B*(y-y_0)^2 + C$;
- [0204] 其中, $E(x,y)$ 是估计代价函数值; x_0 和 y_0 是与相对于中心(0,0)的子像素位移相关联的误差最小的坐标; C 为与坐标(x_0,y_0)处的误差关联的参数; A 和 B 为常数值; x 和 y 为与相邻位置相关联的坐标; x 和 y 各自为包括-1、0和1的整数。
- [0205] 在一个实施例中,通过以下表达式获得子像素位置:
- [0206] $x_0 = (E(-1,0) - E(1,0)) / (2*N*(E(-1,0) + E(1,0) - 2*E(0,0)))$;
- [0207] $y_0 = (E(0,-1) - E(0,1)) / (2*N*(E(0,-1) + E(0,1) - 2*E(0,0)))$;
- [0208] 其中,
- [0209] $E(x,y) = A*(x-x_0)^2 + B*(y-y_0)^2 + C$;
- [0210] $E(x,y)$ 是估计代价函数值; x_0 和 y_0 是与相对于中心(0,0)的子像素位移相关联的误差最小的坐标; C 为与坐标(x_0,y_0)处的误差关联的参数; A 和 B 为常数值; x 和 y 为与相邻位置相关联的坐标; x 和 y 各自为包括-1、0和1的整数;对于1/2、1/4、1/8或1/16的子像素精度, N 为整数1、2、4或8。
- [0211] 本发明的另一方面提供了一种用于在解码端运动矢量修正系统中获取参考列表L0和参考列表1中的一个或多个参考帧中的子像素精确增量运动矢量修正的方法。所述方法可以包括:
- [0212] 提供处理器(1101);
- [0213] 所述处理器使用第一代价函数迭代地执行(1103)整数1像素距离修正操作,以确定所述一个或多个参考帧中每个参考帧的整数距离修正运动矢量,从而启动循环操作;
- [0214] 确定当前操作是否为最后一次操作(1105);
- [0215] 在当前操作是最后一次操作(1105,是)的情况下:
- [0216] 使用第二代价函数进行整数1像素距离修正,以获得每个参考帧的独立的子像素增量运动矢量(1111);
- [0217] 在当前操作不是最后一次操作(1105,否)的情况下:
- [0218] 所述处理器确定当前操作的搜索中心位置的代价函数值是否为1像素相邻位置集合中的第一最低函数值(1107);
- [0219] 在搜索中心位置的代价函数值是第一最低函数值(1107,是)的情况下:
- [0220] 退出循环操作;

- [0221] 使用第二代价函数进行整数1像素距离修正,以获得每个参考帧的独立的子像素增量运动矢量(1111);
- [0222] 在当前操作的搜索中心位置的代价函数值不是第一最低函数值(1107,否)的情况下:
- [0223] 将中心更新到当前操作具有第一最低代价的位置;
- [0224] 重复所述循环操作(1103、1105、1107、1109)。
- [0225] 在一个实施例中,所述方法还可以包括:
- [0226] 所述处理器确定获取的独立子像素增量运动矢量的搜索中心位置的第二代价函数值是否 为第二最低函数值(1113);
- [0227] 在所述第二代价函数值为最低代价函数值(1113,是)的情况下:
- [0228] 通过使用搜索中心和1像素相邻位置集合的代价函数值计算在拟合的参数误差面上值最小的位置,确定每个参考帧中搜索中心周围的子像素距离修正运动矢量(1115);
- [0229] 返回总修正运动矢量,作为使用每个参考帧的参数误差面获得的每个参考帧的确定的整数距离修正运动矢量和确定的子像素距离修正运动矢量的和(1117);
- [0230] 在第二代价函数值不是第二最低代价函数值(1113,否)的情况下:
- [0231] 返回与每个参考帧中具有第二最低代价函数值的位置对应的修正运动矢量(1119)。
- [0232] 在一个实施例中,通过以下获得子像素位置:
- [0233] $x_0 = (E(-1,0) - E(1,0)) / (2 * N * (E(-1,0) + E(1,0) - 2 * E(0,0)))$
- [0234] $y_0 = (E(0,-1) - E(0,1)) / (2 * N * (E(0,-1) + E(0,1) - 2 * E(0,0)))$
- [0235] 其中,
- [0236] $E(x,y) = A * (x - x_0)^2 + B * (y - y_0)^2 + C$;
- [0237] (x_0, y_0) 对应相对于 $(0,0)$ 的误差最小的子像素位移; C 对应 (x_0, y_0) 处的误差; A 和 B 为常数; (x, y) 对应相邻位置; x 包括 $-1, 0, 1$; y 包括 $-1, 0, 1$ 。
- [0238] 在一个实施例中,第一代价函数和第二代价函数都通过以下表达式计算:
- [0239] $E(x,y) = A * (x - x_0)^2 + B * (y - y_0)^2 + C$;
- [0240] 其中, $E(x,y)$ 是估计代价函数值; x_0 和 y_0 是与相对于中心 $(0,0)$ 的子像素位移相关联的误差最小的笛卡尔坐标; C 为与笛卡尔坐标 (x_0, y_0) 处的误差关联的参数; A 和 B 为常数值; x 和 y 为与相邻位置相关联的笛卡尔坐标; x 和 y 各自为包括 $-1, 0$ 和 1 的整数。
- [0241] 在一个实施例中,通过双边匹配获得子像素精确增量运动矢量修正。
- [0242] 在一个实施例中,双边匹配包括在参考列表 L_0 和参考列表 L_1 两者中执行联合修正。
- [0243] 在一个实施例中,双边匹配包括在参考列表 L_0 和参考列表 L_1 中针对共同的双边平均模板执行独立的联合修正。
- [0244] 在一个实施例中,双边匹配包括:在参考列表 L_0 或参考列表 L_1 中针对共同的双边平均模板进行修正;在获得其它参考列表中的位移时,对估计的水平 and 垂直位移取反。
- [0245] 总之,本发明的实施例涉及根据用于双向运动矢量估计的模板匹配确定运动矢量。特别地,将块模板构造为待修正的初始运动矢量所指向的块的平均值。然后,在两个不同的参考图像中通过模板匹配进行运动矢量修正。通过在两个参考图像中的每个参考图

像中查找与最佳匹配块对应的匹配函数的优值(最小或最大,取决于函数)来执行匹配。搜索(在搜索空间的运动矢量候选项所指向的块位置之中)零均值模板和零均值候选块的优值。换句话说,在进行函数优化之前,模板减去模板的均值,候选块减去候选块的均值。然后,将当前块的预测值计算为相应参考图像中最佳匹配块的加权平均值。

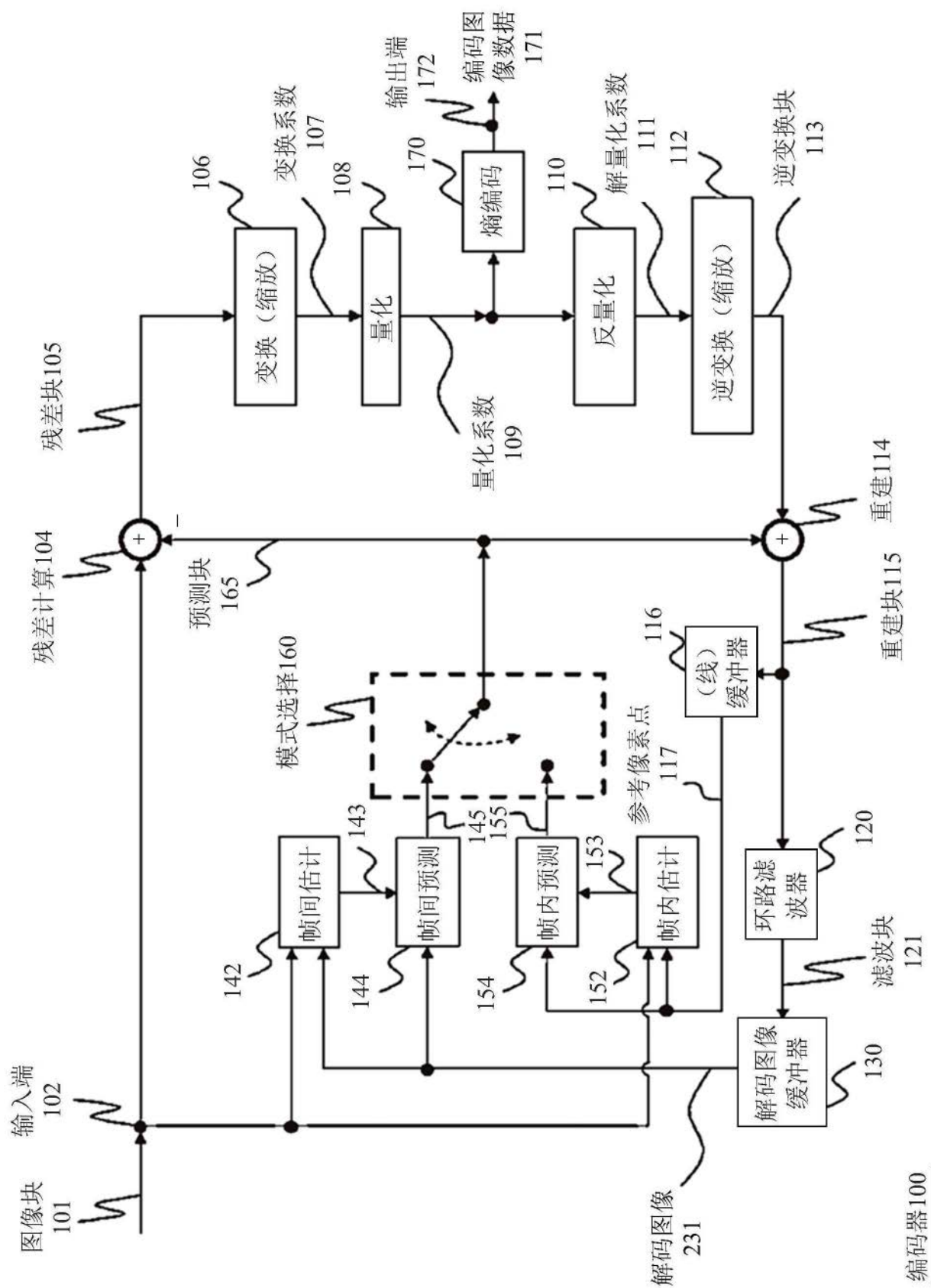


图1

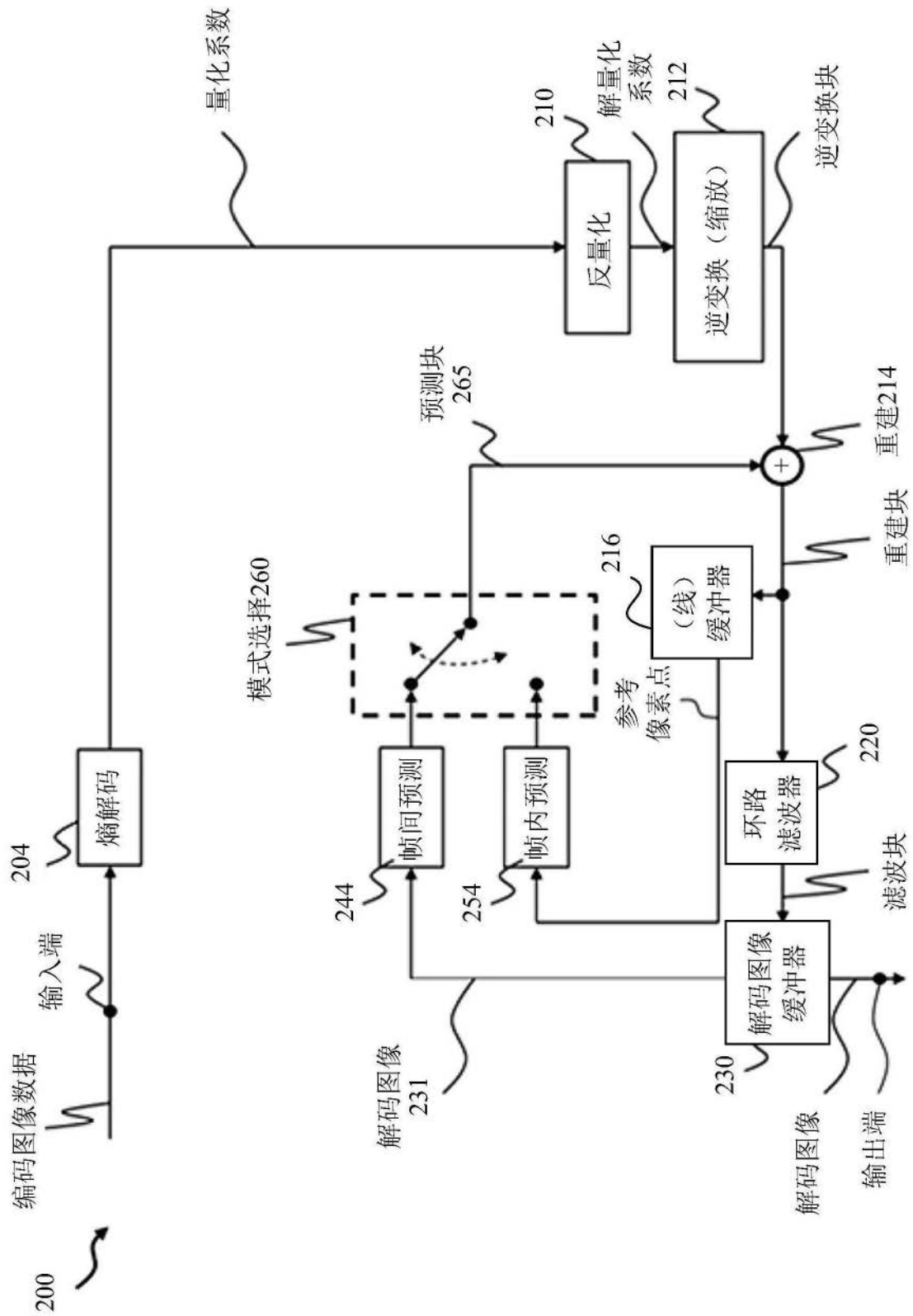


图2

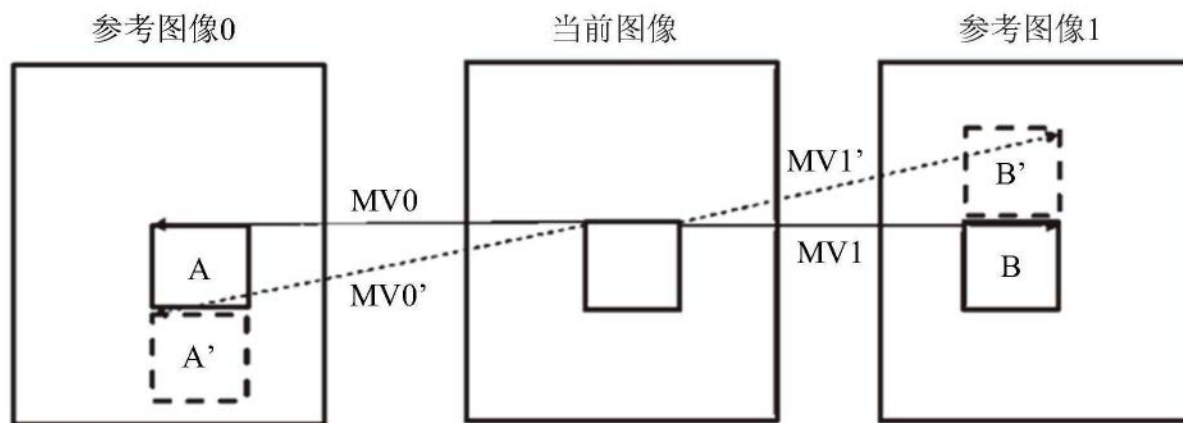


图3

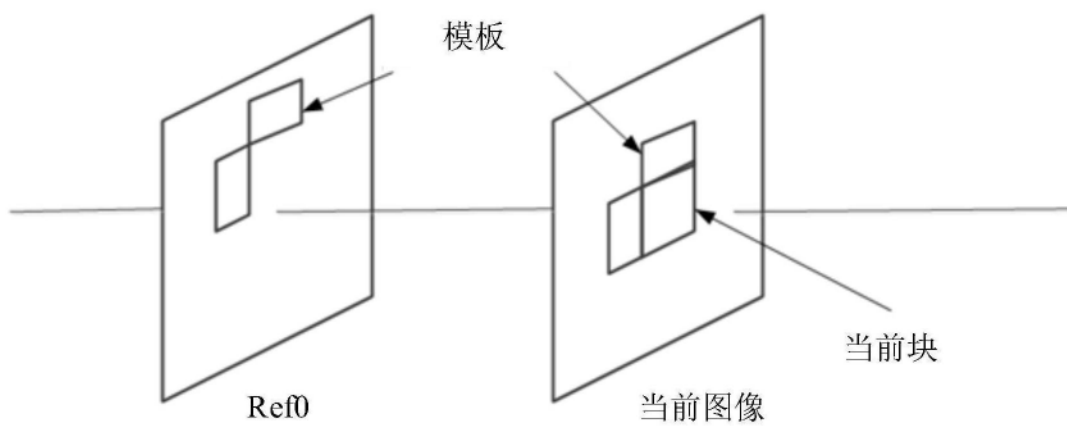


图4A

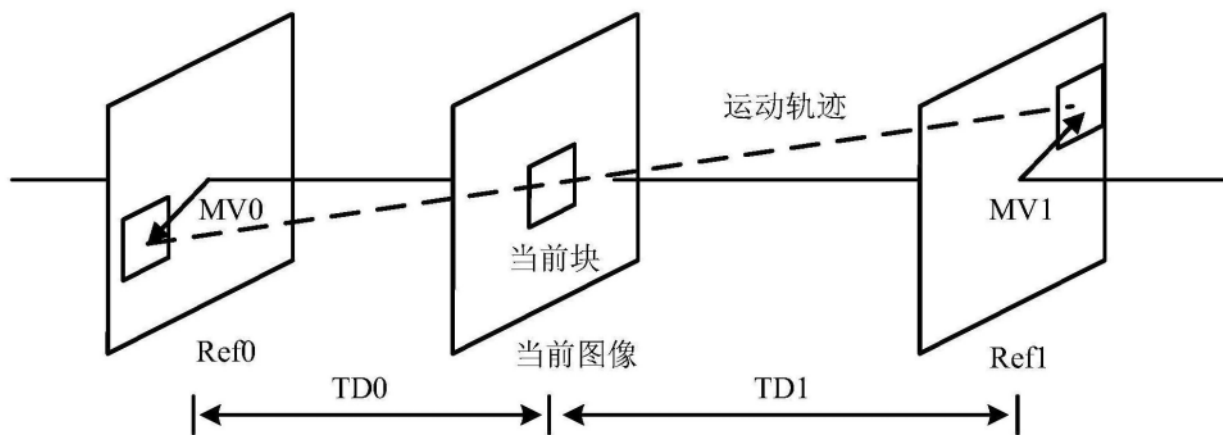


图4B

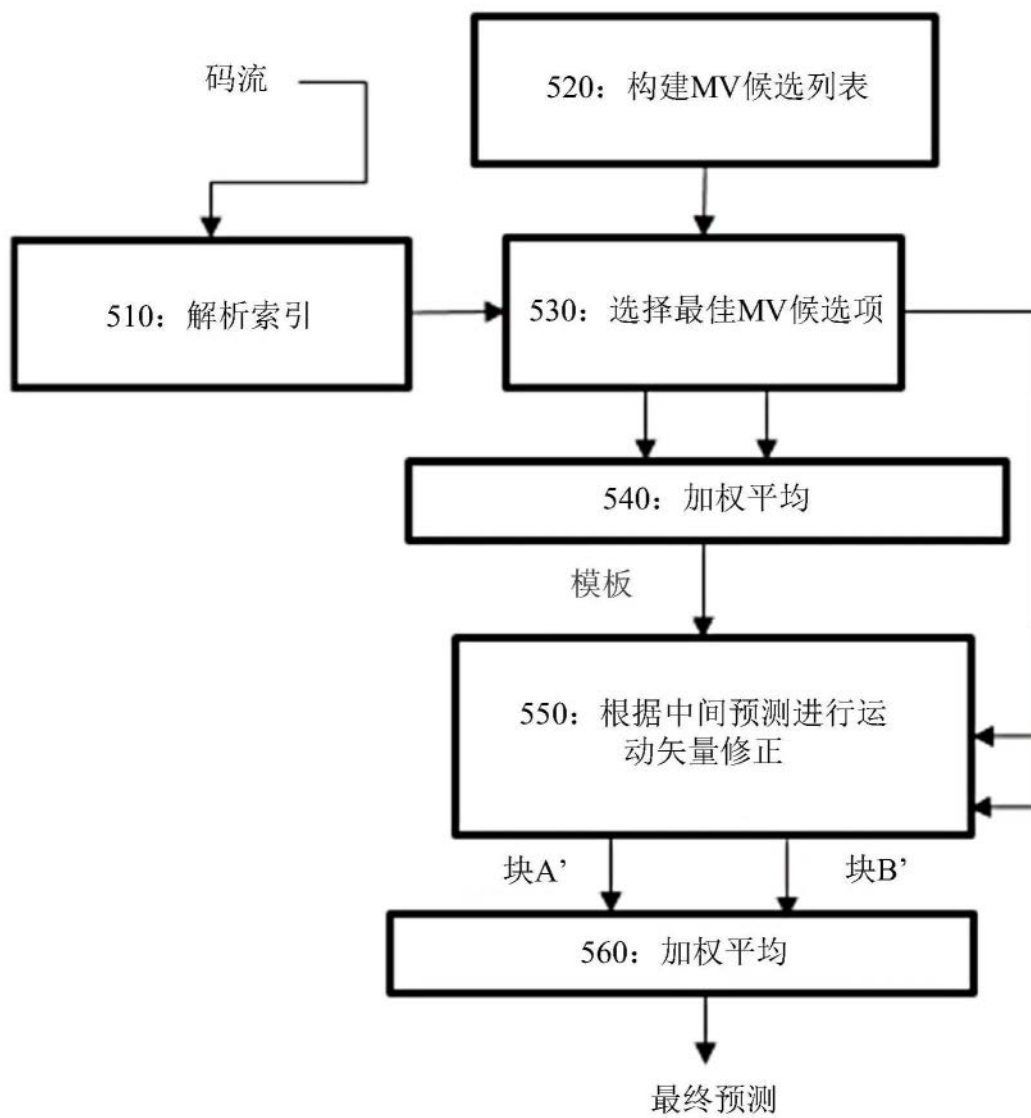


图5

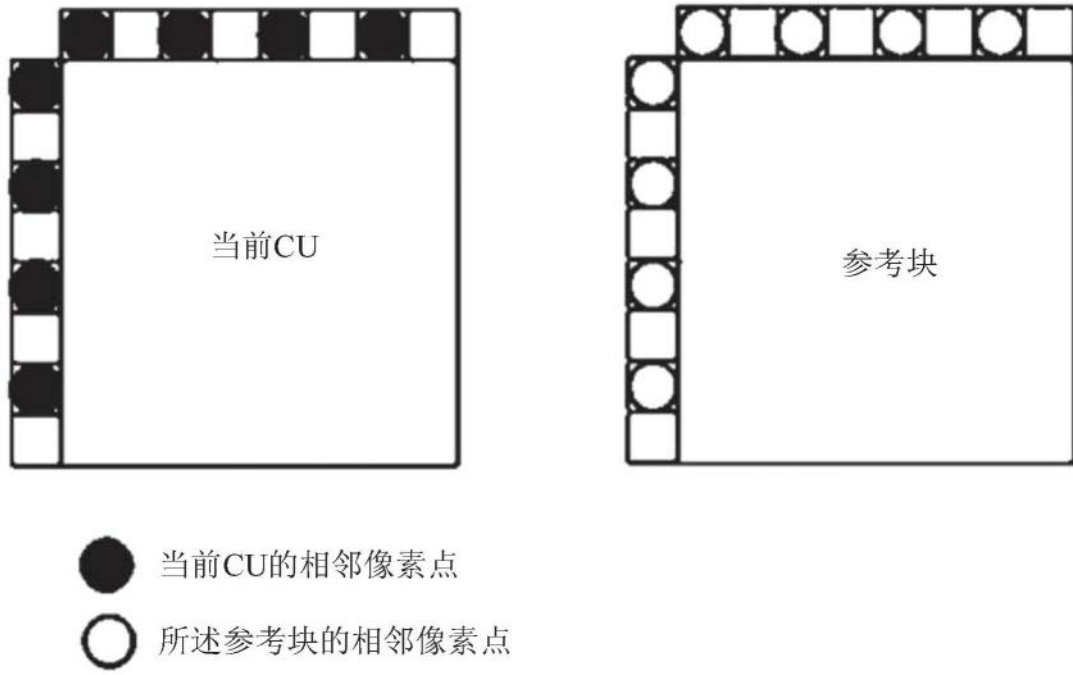


图6

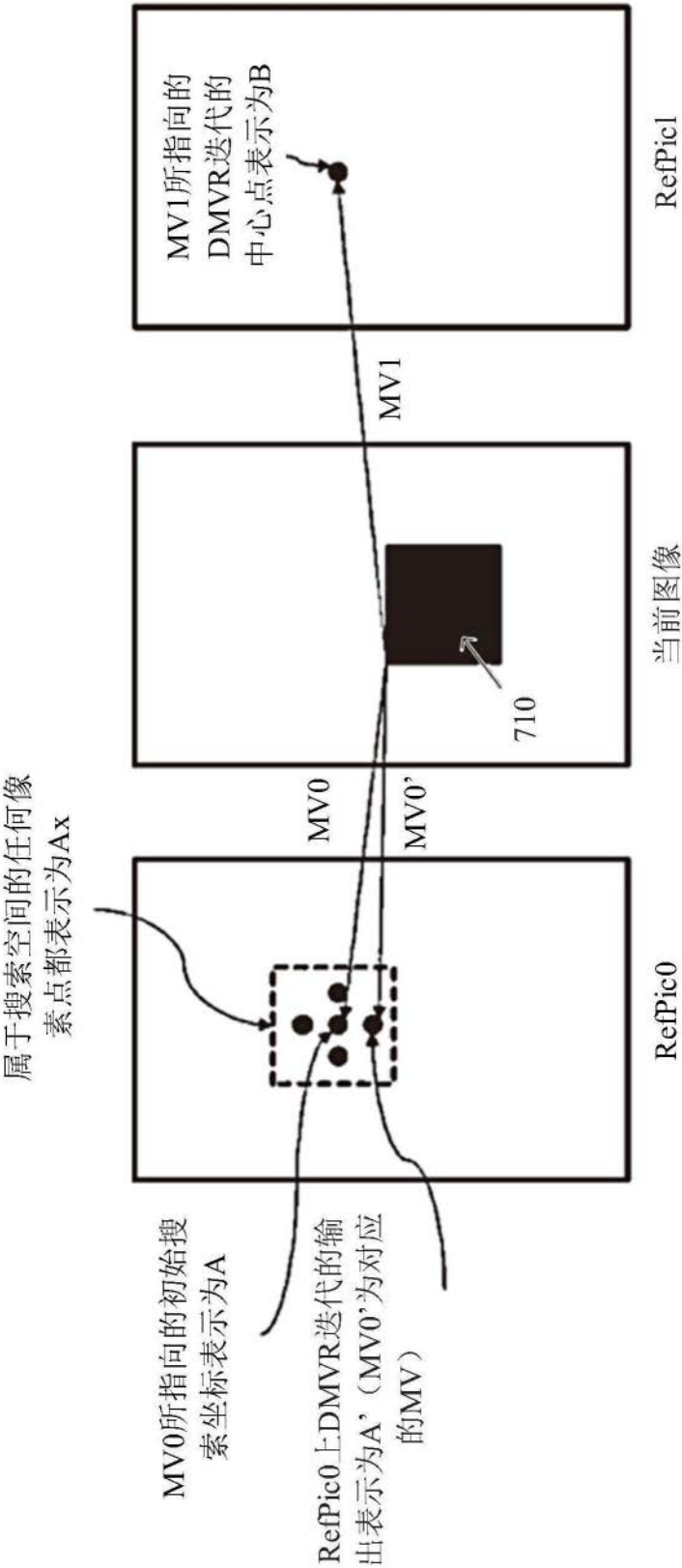


图7

$A_{-1,-1}$				$A_{0,-1}$	$a_{0,-1}$	$b_{0,-1}$	$c_{0,-1}$	$A_{1,-1}$				$A_{2,-1}$
$A_{-1,0}$				$A_{0,0}$	$a_{0,0}$	$b_{0,0}$	$c_{0,0}$	$A_{1,0}$				$A_{2,0}$
$d_{-1,0}$				$d_{0,0}$	$e_{0,0}$	$f_{0,0}$	$g_{0,0}$	$d_{1,0}$				$d_{2,0}$
$h_{-1,0}$				$h_{0,0}$	$i_{0,0}$	$j_{0,0}$	$k_{0,0}$	$h_{1,0}$				$h_{2,0}$
$n_{-1,0}$				$n_{0,0}$	$p_{0,0}$	$q_{0,0}$	$r_{0,0}$	$n_{1,0}$				$n_{2,0}$
$A_{-1,1}$				$A_{0,1}$	$a_{0,1}$	$b_{0,1}$	$c_{0,1}$	$A_{1,1}$				$A_{2,1}$
$A_{-1,2}$				$A_{0,2}$	$a_{0,2}$	$b_{0,2}$	$c_{0,2}$	$A_{1,2}$				$A_{2,2}$

图8

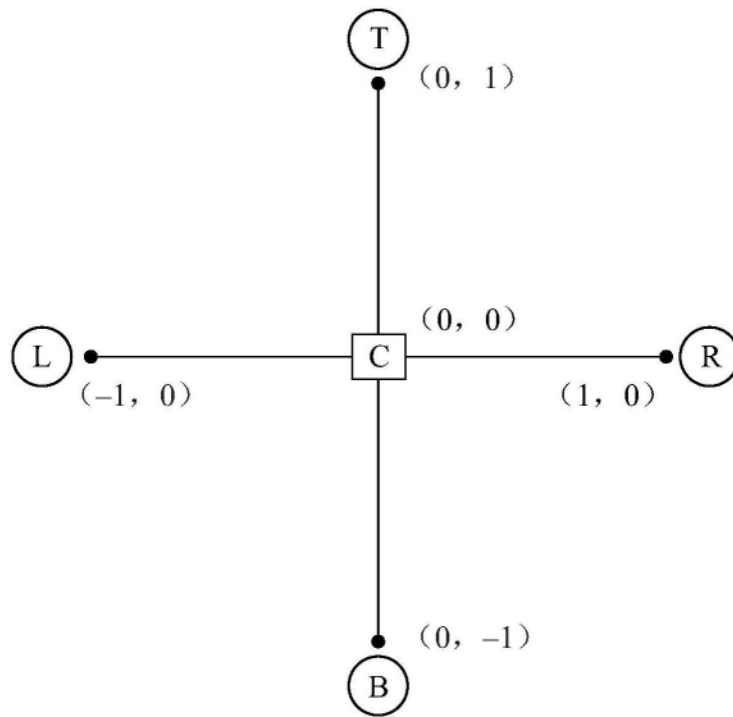


图9

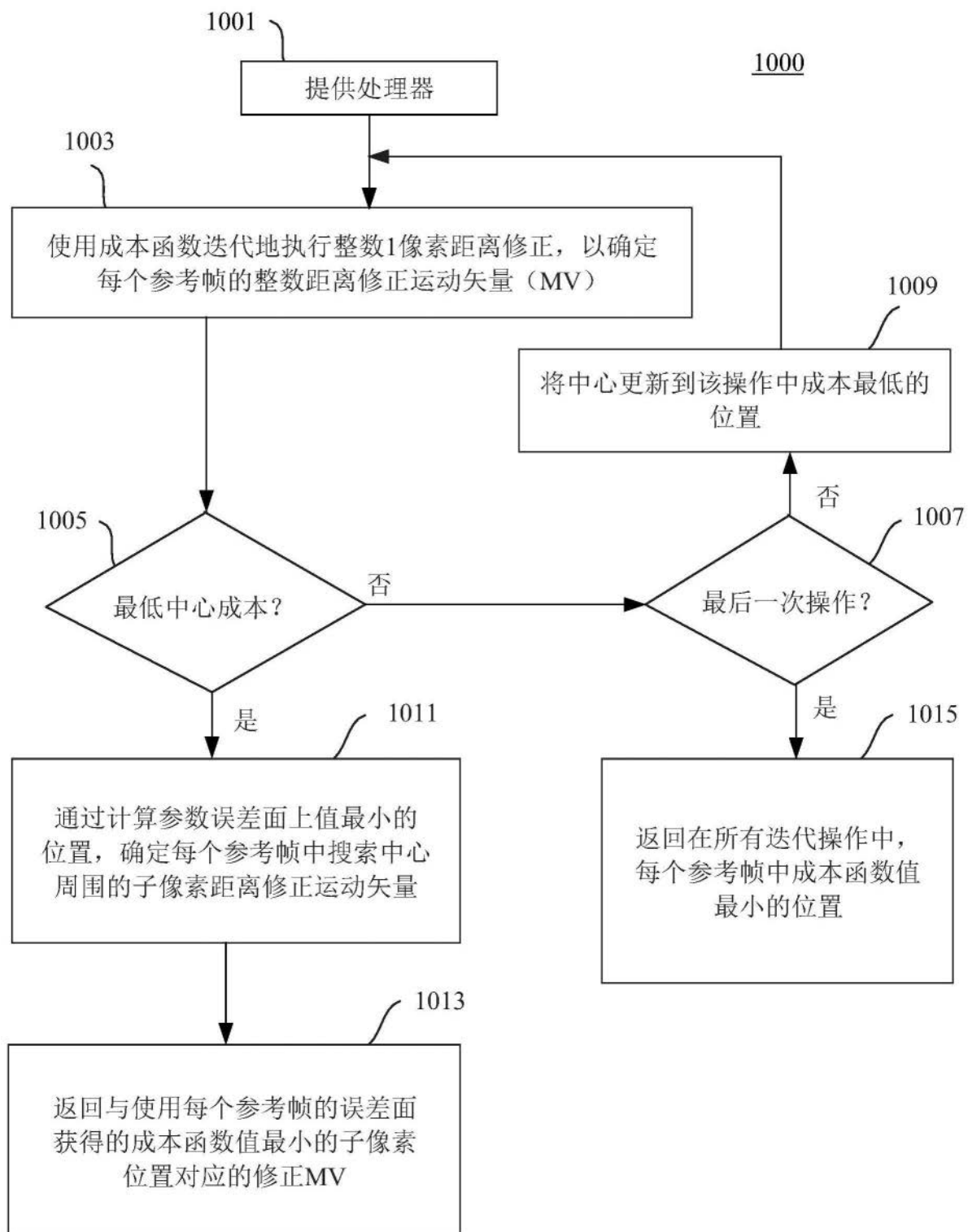


图10

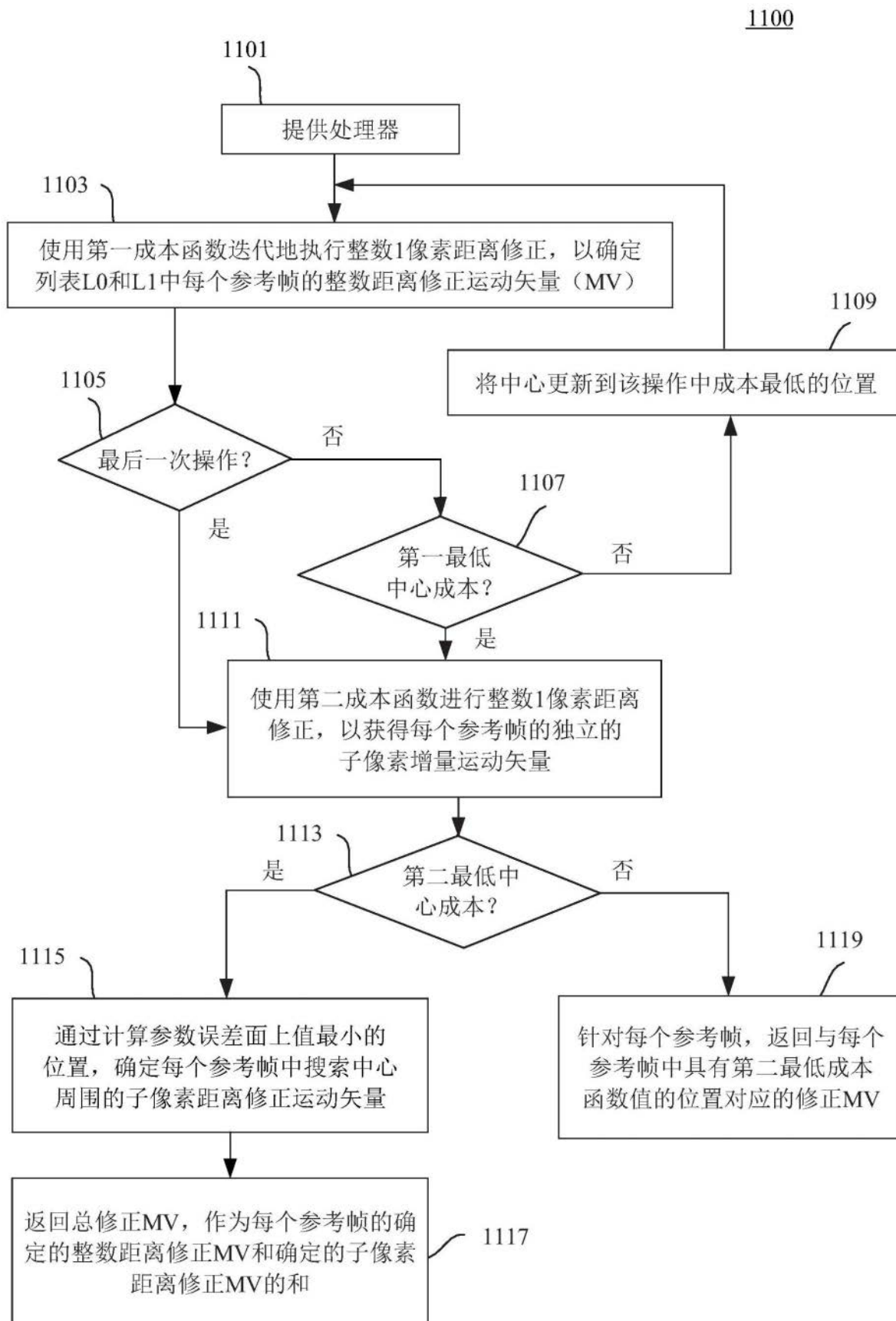


图11

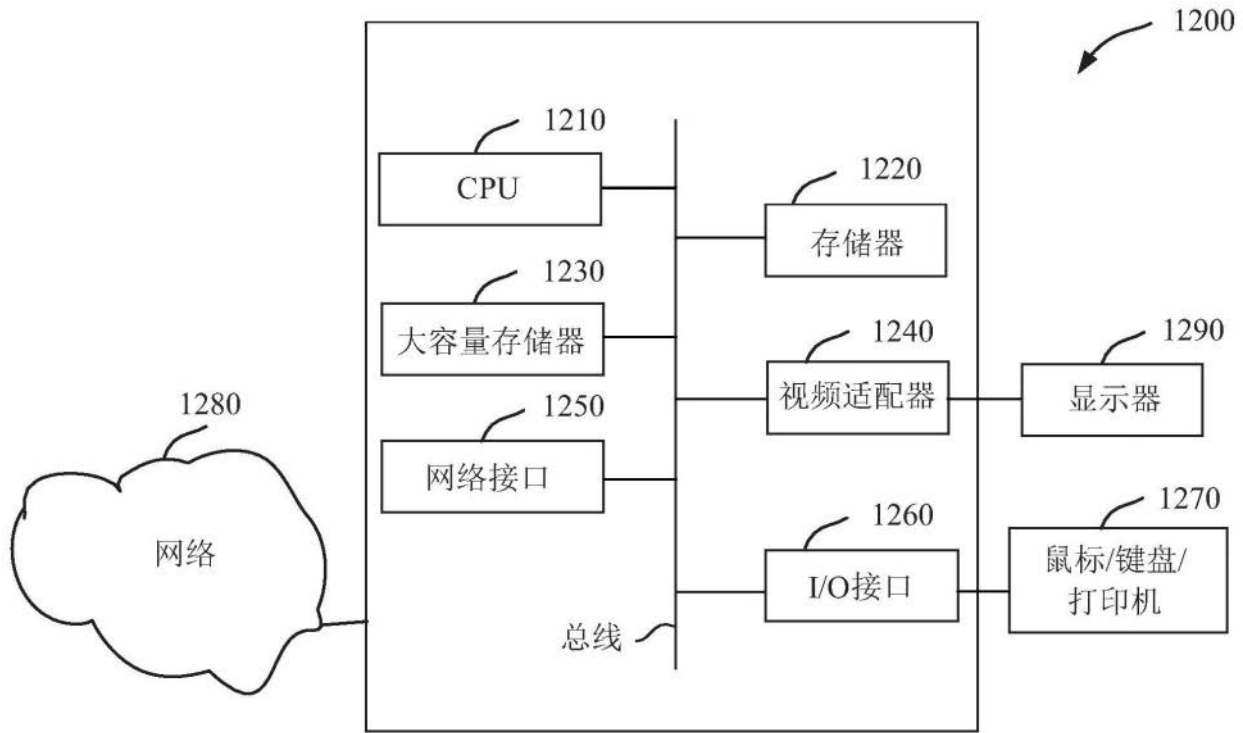


图12