(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119316608 A
(43)申请公布日 2025.01.14

(21)申请号 202411592190.0

(22)申请日 2019.02.21

(30)优先权数据
1802972.8 2018.02.23 GB
1820023.8 2018.12.07 GB

(62)分案原申请数据
201980014855.0 2019.02.21

(71)申请人 佳能株式会社
地址 日本

(72)发明人 G·拉罗彻 乔纳森·泰奎特
P·乌诺 C·吉斯科特

(74)专利代理机构 北京魏启学律师事务所 11398
专利代理师 陈涛

(51)Int.Cl.
H04N 19/50(2014.01)
H04N 19/186(2014.01)
H04N 19/156(2014.01)
H04N 19/42(2014.01)

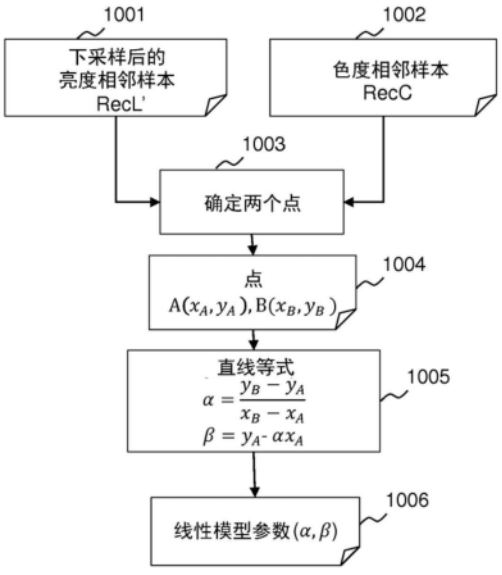
权利要求书2页 说明书36页 附图11页

(54)发明名称

推导线性模型的方法和装置以及解码一个或多个图像的装置

(57)摘要

本发明涉及推导线性模型的方法和装置以及解码一个或多个图像的装置。本发明还涉及跨分量预测、以及用于推导线性模型的方法，该线性模型从同一帧中的第二分量块的第二分量样本获得第一分量块的关联的重建第一分量样本，该方法包括：基于第一分量和第二分量这两者的重建样本来确定表示通过两个点的直线的线性方程的参数，各点由第一变量和第二变量这两个变量定义，所述第一变量对应于第二分量样本值，所述第二变量对应于第一分量样本值；以及推导由直线参数定义的线性模型。



CN 119316608 A

1.一种用于推导线性模型以从同一帧中的亮度块的关联亮度样本值获得目标区域的色度块的色度样本值的方法,所述方法包括:

第一确定步骤,用于基于所述目标区域的邻域中的预定位置处的样本值来确定用于推导所述线性模型的两对值;以及

第二确定步骤,用于使用用于推导所述线性模型的两对值来确定所述线性模型的参数,

其中,所述两对值中的每对值由两个变量来定义,

所述两个变量中的第一变量对应于亮度样本值,

所述两个变量中的第二变量对应于色度样本值,以及

所述两对值中的每个值是通过对基于所述邻域中的样本值的两个值求平均来确定的。

2.根据权利要求1所述的方法,其中,所述线性模型的参数包括与所述线性模型的斜率相对应的参数。

3.根据权利要求1所述的方法,其中,所述线性模型的参数包括与所述线性模型的截距相对应的参数。

4.一种用于通过推导线性模型以从同一帧中的亮度块的关联亮度样本值获得目标区域的色度块的色度样本值来从位流解码一个或多个图像的方法,所述方法包括:

第一确定步骤,用于基于所述目标区域的邻域中的预定位置处的样本值来确定用于推导所述线性模型的两对值;以及

第二确定步骤,用于使用用于推导所述线性模型的两对值来确定所述线性模型的参数,

其中,所述两对值中的每对值由两个变量来定义,

所述两个变量中的第一变量对应于亮度样本值,

所述两个变量中的第二变量对应于色度样本值,以及

所述两对值中的每个值是通过对基于所述邻域中的样本值的两个值求平均来确定的。

5.根据权利要求4所述的方法,其中,所述线性模型的参数包括与所述线性模型的斜率相对应的参数。

6.根据权利要求4所述的方法,其中,所述线性模型的参数包括与所述线性模型的截距相对应的参数。

7.一种用于推导线性模型以从同一帧中的亮度块的关联亮度样本值获得目标区域的色度块的色度样本值的装置,所述装置包括:

第一确定单元,其被配置为基于所述目标区域的邻域中的预定位置处的样本值来确定用于推导所述线性模型的两对值;以及

第二确定单元,其被配置为使用用于推导所述线性模型的两对值来确定所述线性模型的参数,

其中,所述两对值中的每对值由两个变量来定义,

所述两个变量中的第一变量对应于亮度样本值,

所述两个变量中的第二变量对应于色度样本值,以及

所述第一确定单元通过对基于所述邻域中的样本值的两个值求平均来确定所述两对值中的每个值。

8.根据权利要求7所述的装置,其中,所述线性模型的参数包括与所述线性模型的斜率相对应的参数。

9.根据权利要求7所述的装置,其中,所述线性模型的参数包括与所述线性模型的截距相对应的参数。

10.一种用于通过推导线性模型以从同一帧中的亮度块的关联亮度样本值获得目标区域的色度块的色度样本值来从位流解码一个或多个图像的装置,所述装置包括:

第一确定单元,其被配置为基于所述目标区域的邻域中的预定位置处的样本值来确定用于推导所述线性模型的两对值;以及

第二确定单元,其被配置为使用用于推导所述线性模型的两对值来确定所述线性模型的参数,

其中,所述两对值中的每对值由两个变量来定义,

所述两个变量中的第一变量对应于亮度样本值,

所述两个变量中的第二变量对应于色度样本值,以及

所述第一确定单元通过对基于所述邻域中的样本值的两个值求平均来确定所述两对值中的每个值。

11.根据权利要求10所述的装置,其中,所述线性模型的参数包括与所述线性模型的斜率相对应的参数。

12.根据权利要求10所述的装置,其中,所述线性模型的参数包括与所述线性模型的截距相对应的参数。

推导线性模型的方法和装置以及解码一个或多个图像的装置

[0001] (本申请是申请日为2019年2月21日、申请号为2019800148550、发明名称为“用于线性分量样本预测的新样本集和新下采样方法和装置”的申请的分案申请。)

技术领域

[0002] 本发明涉及对给定视频分量的块进行编码或解码,特别是对这样的分量块进行帧内预测或者获得这样的块的样本。本发明可应用于从另一组件的样本(通常是亮度样本)获得视频数据的分量的块(通常是色度分量的块)。

背景技术

[0003] 视频数据的预测编码是基于将帧分割成像素块。对于各像素块,在可用数据中搜索预测结果块。预测结果块可以是与INTER(帧间)编码模式中的当前帧不同的参考帧中的块,或者是根据INTRA(帧内)编码模式中的当前帧中的相邻像素生成的。根据确定预测结果块的不同方式来定义不同的编码模式。编码的结果是预测结果块以及涉及要编码的块和预测结果块之间的差的残差块的信令(signaling)。

[0004] 对于INTRA编码模式,通常提出了诸如直流(DC)模式、平面模式和角模式等的各种模式。这些模式中的各模式均试图使用来自空间相邻块的先前解码的边界样本来预测块的样本。

[0005] 可以对形成视频数据的像素的各分量进行编码。尽管RGB(红色-绿色-蓝色的缩写)表示是众所周知的,但优选将YUV表示用于编码以减少通道间冗余。根据这些编码模式,像素块可被视为包括是多个(通常为三个)分量块。RGB像素块包括:R分量块,其包含该块的像素的R分量的值;G分量块,其包含这些像素的G分量的值;B分量块,其包含这些像素的B分量的值。同样,YUV像素块包括Y分量块(亮度)、U分量块(色度)和V分量块(也是色度)。另一示例是YCbCr,其中Cb和Cr也称为色度分量。然而,仍能在局部观察到分量间(也称为跨分量)相关性。

[0006] 为了提高压缩效率,在现有技术中研究了跨分量预测(CCP)的使用。CCP的主要应用是亮度到色度预测。这意味着已根据编码数据对亮度样本进行了编码和重建(如解码器一样)、并且根据亮度来预测色度。然而,变形例将CCP用于色度到色度预测,或者更一般地将CCP用于第一分量到第二分量预测(包括RGB)。

[0007] 跨分量预测可以直接应用于色度像素块,或者可以应用于残差色度块(其意味着色度块和色度块预测结果之间的差)。

[0008] 线性模型(LM)模式使用用以根据亮度预测色度的线性模型作为色度帧内预测模式,这依赖于要确定的一个或两个参数(斜率(α)和偏移(β))。由此可以使用具有这些参数的线性模型从当前亮度块的重建亮度样本来推导色度帧内预测结果。

[0009] 线性度(即,参数α和β)是从重建因果样本推导出的,特别是从包括与要预测的当前色度块相邻的重建色度样本的相邻色度样本集、以及从包括与当前亮度块相邻的亮度样本的相邻亮度样本集推导出的。

[0010] 特别地,对于N×N色度块,使用上方行的N个近邻和左方列的N个近邻来形成相邻色度样本集,以进行推导。

[0011] 相邻亮度样本集也由相应亮度块的正上方的N个相邻样本和该亮度块的左侧的N个相邻样本组成。

[0012] 已知通过对色度分量进行子采样来在不会使视觉渲染显著下降的情况下减小要编码的视频数据的大小。已知的子采样模式被标记为4:1:1、4:2:2、4:2:0。

[0013] 在对视频色度数据进行子采样的情形中,与N×N色度块相对应的亮度块大于N×N。在这种情况下,对相邻亮度样本集进行下采样以匹配色度分辨率。必须使用具有所推导出的一个或多个参数α和β的线性模型、以及先前进行了下采样以匹配色度分辨率的当前亮度块的重建亮度样本来生成用以预测当前N×N色度块中的色度样本的色度帧内预测结果。将重建亮度样本下采样到色度分辨率,这使得可以检索到与色度样本相同数量的样本,以形成亮度样本集和色度帧内预测结果这两者。

[0014] 因而,从当前色度块中减去色度帧内预测结果以获得在编码器处编码的残差色度块。相反,在解码器处,将色度帧内预测结果与所接收到的残差色度块相加以检索到色度块,这也称为解码块的重建。这还可以涉及针对超出样本范围的相加结果的限幅(clipping)。

[0015] 有时,残差色度块是可忽略的,因而在编码期间不考虑。在这种情况下,使用上述色度帧内预测结果作为色度块本身。结果,上述LM模式使得可以使用具有一个或多个参数的线性模型来从同一帧中的另一分量的块的关联(即,同位置或对应)重建样本获得给定分量的当前块的样本。该样本是使用具有所推导出的一个或多个参数的线性模型以及另一分量的块中的关联的重建样本所获得的。如果需要,该另一分量的块由下采样以匹配当前分量的块分辨率的样本所组成。尽管当前分量的块通常是色度块、并且另一分量的块是亮度块,但这可能并非如此。为了清楚和简单起见,这里给出的示例集中于从亮度块预测色度块,应该很清楚,所述的机制可以应用于从另一分量的任何分量预测。

[0016] 联合视频探索组(JVET)的联合探索模型(JEM)将六个跨分量(亮度到色度)线性模型模式添加到已知的传统帧内预测模式。所有这些模式彼此竞争,以预测或生成色度块,该选择通常是基于编码器端处的率失真标准进行的。

[0017] 六个跨分量(亮度到色度)线性模型模式在以下方面彼此不同:用于对重建亮度样本进行下采样的不同下采样方案、以及/或者推导参数α和β所根据的样本的不同样本集。

[0018] 例如,样本集可以由与当前亮度或色度块相邻的两个线(即,行和列)的样本所组成,这些线是平行的且以色度分辨率紧邻当前亮度或色度块的上方和/或左方边界中的每一个边界。在公开文本US 9,736,487中描述了这样的示例性样本集。

[0019] 在公开文本US 9,288,500和US 9,462,273中也公开了其它的示例性样本集。

[0020] JEM中所使用的下采样方案包括:6抽头滤波器,用于从六个重建亮度样本确定下采样后的重建亮度样本;三个2抽头滤波器,用于选择六个重建亮度样本中的右上样本和右下样本、或者下方样本和右下样本、或者上方样本和右上样本;以及4抽头滤波器,用于选择六个重建亮度样本的上方样本、右上样本、下方样本和右下样本。

发明内容

[0021] JEM在处理方面是复杂的。例如,JEM需要线性模型参数的复杂推导以计算色度预测结果块样本。

[0022] 设计了本发明以解决一个或多个前述问题。本发明涉及用于可能地通过色度帧内预测来获得当前色度块的色度样本的得以改进的方法。

[0023] 根据本发明的第一方面,提供一种根据第一方面所述的方法。

[0024] 根据本发明的另一方面,提供一种根据第二方面所述的用于推导线性模型的装置。

[0025] 根据本发明的另一方面,提供一种根据第三方面所述的用于从同一帧中的第二分量块的第二分量样本获得第一分量块的关联的重建第一分量样本的装置。

[0026] 根据本发明的另一方面,提供一种根据第四方面所述的计算机程序产品、计算机可读介质或计算机程序。

[0027] 在其它方面中提供了本发明的更多方面。

[0028] 根据又一方面,提供一种用于推导线性模型的方法,所述线性模型用于从第二分量样本值中推导关联的重建第一分量样本值,所述方法包括:从两个或更多个集合中取两个集合,各集合包括来自第一分量和第二分量的重建样本值的第一分量样本值和第二分量样本值;以及基于所述两个集合之间的所述第一分量样本值和所述第二分量样本值的变化率来推导所述线性模型,使得所述两个集合的第一分量样本值是使用所推导出的线性模型能够从相应集合的第二分量样本值获得的。

[0029] 应理解,第一分量样本值和关联的重建第二分量样本值通过预设的关系彼此关联。

[0030] 合适地,预设关系是这两者彼此同位置或相对应。可以针对各样本值单独地、或者在第一分量样本值的块/组与第二分量样本值的块/组之间定义该同位置关系或对应关系。

[0031] 合适地,预设关系是这两者与要处理的当前像素块中的至少一个像素相关联,例如,这两者是要处理的至少一个像素的同位置的或对应的样本值。可以针对各样本值单独地、或者在样本值的块/组与像素的块/组之间定义该同位置关系或对应关系。

[0032] 还应理解,可以将下采样或上采样处理应用于第一分量样本值或第二分量样本值的块,使得可以在下采样/上采样之后建立块之间或与当前像素块的至少一个像素的预设关系。

[0033] 合适地,第一分量样本值和关联的第二分量样本值与要处理的相同图像或帧的像素块相关联。应理解,这里,包括第一分量样本值和第二分量样本值的集合是第一分量样本值和第二分量样本值的分量样本值集合。因此,该集合是具有第一分量样本值和第二分量样本值作为其元素的n元组。合适地,该集合是2元组。可替代地,该集合是具有多于两个的元素(n个元素)的n元组。

[0034] 合适地,第一分量和第二分量的重建样本值与同要处理的当前块相邻的一个或多个块相关联。合适地,与当前块相邻的一个或多个块在当前块的上方或左方。

[0035] 合适地,所取的两个集合是包括两个或更多个集合内的第二分量样本值中的最小第二分量样本值和最大第二分量样本值的集合。合适地,所取的两个集合是包括两个或更多个集合内的第一分量样本值中的最小第一分量样本值和最大第一分量样本值的集合。

[0036] 合适地,取两个集合包括:确定第一组集合,该第一组集合包括两个或更多个集合内的第二分量样本值中的最小第二分量样本值和最大第二分量样本值;确定第二组集合,该第二组集合包括两个或更多个集合内的第一分量样本值中的最小第一分量样本值和最大第一分量样本值;以及从第一组和第二组的集合中选择两个集合。

[0037] 合适地,从第一组和第二组的集合中选择两个集合包括:在最小第二分量样本值和最大第二分量样本值之间的差大于最小第一分量样本值和最大第一分量样本值之间的差的情况下,选择第一组;否则,选择第二组。

[0038] 合适地,从第一组和第二组的集合中选择两个集合包括:确定第一组和第二组的集合的样本值的位置;以及基于所确定的样本值的位置来选择两个集合。合适地,关于与要处理的第一分量样本值的块相关联的重建第二分量样本值的块,针对重建第二分量样本值来确定样本值的位置。合适地,关于要处理的重建第一分量样本值的块,针对重建第一分量样本值来确定样本值的位置。合适地,基于与要处理的像素块相关地定义的关联/同位置/对应位置来确定样本值的位置。

[0039] 合适地,基于所确定的样本值的位置来选择两个集合包括:选择包括与要处理的块相邻的预定位置处的样本值的集合。合适地,基于所确定的样本值的位置来选择两个集合包括:确定第一组和第二组的集合中的任何集合是否包括预定位置处的样本值;以及将包括该预定位置处的样本值的集合选择为这两个集合其中之一。合适地,基于所确定的样本值的位置来选择两个集合包括:在包括预定位置处的样本值的集合不可用的情况下,确定第一组和第二组的集合中的任何集合是否包括另一预定位置处的样本值;以及将包括该另一预定位置处的样本值的集合选择为这两个集合其中之一。合适地,该预定位置或该另一预定位置是与要处理的当前块相邻的位置中的左下或右上位置。

[0040] 合适地,从第一组和第二组的集合中选择两个集合包括:比较来自第一组和第二组的集合的两个集合之间的距离,其中这些距离是在第一分量样本值和第二分量样本值的空间中定义的,该空间由集合的元素来定义,使得两个或更多个集合中的各集合对应于所述空间中的位置。

[0041] 合适地,选择两个集合包括:确定第一组的集合之间的距离是否大于第二组的集合之间的距离;以及在第一组的集合之间的距离大于第二组的集合之间的距离的情况下,选择第一组,否则选择第二组。合适地,选择两个集合包括:从第一组和第二组的集合中选择二者之间的距离最大的两个集合。

[0042] 合适地,选择两个集合包括:确定第一组的集合的相应元素是否具有相同或不同的值;以及在相应元素不具有相同值或具有不同值的情况下,选择第一组,并且在相应元素具有相同值或不具有不同值的情况下,选择第二组。合适地,集合的相应元素是第一分量样本值和第二分量样本值中的任一个或这两个。

[0043] 合适地,选择两个集合包括:确定第二组的集合的相应元素是否具有相同值或不同值;以及在相应元素不具有相同值或具有不同值的情况下,选择第二组,并且在相应元素具有相同值或不具有不同值的情况下,选择第一组。

[0044] 合适地,选择两个集合包括:获得第一组的集合之间的第一分量样本值和第二分量样本值的变化率;确定所获得的比率是否大于、等于或小于预设值;以及在所获得的比率大于、等于或小于预设值的情况下,选择第一组,否则选择第二组。合适地,选择两个集合包

括:获得第二组的集合之间的第一分量样本值和第二分量样本值的变化率;确定所获得的比率是否大于、等于或小于预设值;以及在所获得的比率大于、等于或小于预设值的情况下,选择第二组,否则选择第一组。

[0045] 合适地,所取的两个集合是包括来自与同要处理的当前块相关联的第二分量样本值的块相邻的一个或多个块的第二分量样本值的集合,以及取两个集合包括:基于两个集合的第二分量样本值来选择这两个集合。合适地,所取的两个集合是包括相应第二分量样本值的块的重建样本值中的两个最频繁发生的第二分量样本值的集合。

[0046] 合适地,将第二分量的重建样本值分割成至少两个组,并且针对各组,取两个集合且基于所取的两个集合来推导线性模型。合适地,如果针对组所取的两个集合所具有的这些集合之间的第一分量样本值和第二分量样本值的变化率小于或等于预设值,则基于针对另一组所取的两个集合来推导该组的线性模型。合适地,如果针对组所取的两个集合所具有的这些集合之间的第一分量样本值和第二分量样本值的变化率小于或等于预设值,则基于在第二分量的所有重建样本值都在单个组中的情况下已取的两个集合来推导针对该组的线性模型。

[0047] 根据又一方面,提供一种用于将一个或多个图像编码到位流中或者从位流解码一个或多个图像的方法,所述方法包括:根据本发明第一方面的方法来推导线性模型,所述线性模型用于从第二分量样本值获得关联的重建第一分量样本值。

[0048] 合适地,所述方法还包括:选择多个线性模型推导模式其中之一以获得要处理的图像的当前块的第一分量样本值,其中所述多个线性模型推导模式包括使用单个线性模型的第一模式和使用多于一个的线性模型的第二模式,并且所推导出的线性模型可用在所选择的线性模型推导模式中。合适地,仅第一模式使用所推导出的线性模型。可替代地,仅第二模式使用所推导出的线性模型。

[0049] 根据又一方面,提供一种用于推导线性模型的装置,所述线性模型用于从第二分量样本值获得关联的重建第一分量样本值,所述装置被配置为进行本发明第一方面的方法。

[0050] 根据又一方面,提供一种用于将一个或多个图像编码到位流中或者从位流解码一个或多个图像的装置,所述装置被配置为进行本发明第二方面的方法。

[0051] 根据又一方面,提供一种用于从第二分量样本值获得关联的重建第一分量样本值的方法,所述方法包括:从多个线性模型模式中选择一个线性模型模式以获得第一分量样本值;以及使用所选择的线性模型模式来获得所述第一分量样本值,其中所述多个线性模型模式至少之一使用利用根据本发明第一方面的推导方法所推导出的线性模型。合适地,所述多个线性模型模式包括使用单个线性模型的第一模式和使用多于一个的线性模型的第二模式。合适地,仅第一模式使用根据本发明第一方面的推导方法。可替代地,仅第二模式使用根据本发明第一方面的推导方法。

[0052] 根据又一方面,提供一种用于从第二分量样本值获得关联的重建第一分量样本值的装置,所述装置被配置为进行本发明第五方面的方法。

[0053] 根据又一方面,提供一种用于将一个或多个图像编码到位流中的方法,其中所述方法包括:根据本发明第五方面从第二分量样本值获得关联的重建第一分量样本值。合适地,所述方法还包括:在所述位流中提供表示可用于获得所述第一分量样本的线性模型模

式的选择的信息。

[0054] 根据又一方面,提供一种用于从位流解码一个或多个图像的方法,其中所述方法包括:根据本发明第五方面从第二分量样本值获得关联的重建第一分量样本值。合适地,所述方法还包括:从所述位流获得表示可用于获得所述第一分量样本的线性模型模式的选择的信息,并且从多个线性模型模式选择一个线性模型模式是基于所获得的信息来进行的。

[0055] 根据又一方面,提供一种用于将一个或多个图像编码到位流中的装置,所述装置被配置为进行本文所述的方法。

[0056] 根据又一方面,提供一种在执行时使得进行本文所述的方法的计算机程序、以及存储有用于实现本文所述的方法的指令的(非暂时性)计算机可读介质。

[0057] 根据本发明,提供了如所附权利要求书所述的装置、方法、计算机程序(产品)和计算机可读存储介质。本发明实施例的其它特征是在所附权利要求书以及以下的说明中定义的。这些特征中的一些特征在下文参考方法来进行说明,而这些特征可被转置成专用于装置的系统特征。

[0058] 根据本发明的方法的至少一部分可以通过计算机来实现。因此,本发明可以采用以下形式:完全硬件的实施例;完全软件的实施例(包括固件、常驻软件、微代码等);或者组合这里通常可全部称为“处理器和存储器”、“电路”、“模块”或“系统”的软件和硬件方面的实施例。此外,本发明可以采用以介质中嵌入有计算机可用程序代码的表现的任何有形介质中体现的计算机程序产品的形式。

[0059] 由于本发明可以以软件来实现,因此本发明可以体现为计算机可读代码以提供至可编程设备的任何适当载体介质上。有形的载体介质可以包括诸如硬盘驱动器、磁带装置或固态存储器装置等的存储介质。暂时性载体介质可以包括诸如电气信号、电子信号、光学信号、声学信号、磁信号或者例如微波或RF信号的电磁信号等的信号。

附图说明

[0060] 现在将仅通过示例的方式并且参考以下附图来说明本发明的实施例,其中:

[0061] 图1示出视频编码器逻辑架构;

[0062] 图2示出与图1所示的视频编码器逻辑架构相对应的视频解码器逻辑架构;

[0063] 图3示意性示出用于4:2:0采样的YUV采样方案的示例;

[0064] 图4使用流程图示出由编码器或解码器进行的、用于使用LM模式生成块预测结果的一般步骤;

[0065] 图5示意性示出如在现有技术中已知的、色度块和关联或同位置亮度块,其中对亮度样本以及相邻的色度样本和亮度样本进行下采样;

[0066] 图6示意性示出如在现有技术中已知的用于LM参数推导的示例性样本集;

[0067] 图7示出现有技术中已知的一些下采样滤波器;

[0068] 图8示出用以表示LM模式的信令标志的示例性编码;

[0069] 图9示出亮度和色度相邻样本的点、以及表示在本发明的一个实施例中获得的线性模型参数的直线;

[0070] 图10示出本发明的一个实施例中的简化LM推导的处理的主要步骤;

[0071] 图11示出本发明的一些实施例中的、亮度和色度相邻样本的数个点以及用于确定

最佳两个点的线段;

[0072] 图12示出本发明的一个实施例中的MMLM推导的处理的主要步骤;以及

[0073] 图13是用于实现本发明的一个或多个实施例的计算装置的示意框图。

具体实施方式

[0074] 图1示出视频编码器架构。在视频编码器中,原始序列101被分割成被称为HEVC的编码块或编码单元的像素块102。然后,将编码模式影响到各块。视频编码通常使用两族编码模式:基于空间预测的编码模式或者“INTRA模式”103、以及基于时间预测的编码模式或者基于运动估计104和运动补偿105的“INTER模式”。

[0075] INTRA编码块通常是通过被称为INTRA预测的处理根据其因果边界处的编码像素来预测的。针对INTRA编码块的各像素的预测结果由此形成预测结果块。根据使用哪些像素来预测INTRA编码块,提出了各种INTRA模式:例如,DC模式、平面模式和角模式。

[0076] 尽管图1涉及视频编码器架构的一般描述,但应当注意,像素在这里对应于图像的元素,该元素通常包括多个分量(例如,红色分量、绿色分量和蓝色分量)。图像样本是图像的包含仅一个分量的元素。

[0077] 时间预测首先涉及在运动估计步骤104中在被称为参考帧116的先前帧或将来帧中寻找离编码块最近的参考区域。该参考区域构成预测结果块。接着,在运动补偿步骤105中,使用该预测结果块来预测该编码块以计算残余(residue)或残差块。

[0078] 在空间和时间预测这两种情况下,通过从编码块中减去所获得的预测结果块来计算残余或残差块。

[0079] 在INTRA预测中,对预测模式进行编码。

[0080] 在时间预测中,对表示所使用的参考帧的索引和表示参考帧中的参考区域的运动矢量进行编码。然而,为了进一步降低与运动矢量编码有关的位率成本,不直接对运动矢量进行编码。实际上,假定运动是均匀的,则将运动矢量编码为该运动矢量与其周围的运动矢量(或运动矢量预测结果)之间的差是特别有利的。例如,在H.264/AVC编码标准中,针对根据与位于当前块的上方和左方的三个块相关联的运动矢量所计算出的中值矢量来对运动矢量进行编码。仅在中值矢量和当前块运动矢量之间计算出的差(还被称为残差运动矢量)被编码在位流中。这在模块“Mv预测和编码”117中进行处理。将各编码矢量的值存储在运动矢量字段118中。用于预测的相邻运动矢量是从运动矢量字段118中提取的。

[0081] HEVC标准使用三个不同的INTER模式:帧间模式、合并模式和合并跳过模式,这三个模式的主要不同之处在于运动信息在位流110(即,运动矢量和经由所谓的参考帧索引的关联参考帧)中的信令。为了简单起见,以下将运动矢量和运动信息混合。关于运动矢量预测,HEVC提供了在率失真竞争期间评价的运动矢量预测结果的多个候选,以分别针对帧间模式和合并模式找到最佳运动矢量预测结果或最佳运动信息。将与最佳预测结果或运动信息的最佳候选相对应的索引插入在位流110中。由于该信令,解码器可以推导出预测结果或候选的相同集合,并且根据解码后的索引来使用最佳集合。

[0082] 运动矢量预测结果和候选的推导的设计有助于在不会大幅影响复杂度的情况下实现最佳编码效率。在HEVC中提出两个运动矢量推导:一个运动矢量推导用于帧间模式(已知为高级运动矢量预测(AMVP))并且另一运动矢量推导用于合并模式(已知为合并推导处

理)。

[0083] 接着,在模块106中选择用于优化针对当前考虑的编码块的率失真标准的编码模式。为了进一步减少所获得的残差数据内的冗余,在模块107中对残差块应用通常为DCT的变换,并且在模块108中对所获得的系数应用量化。然后,在模块109中对系数的量化块进行熵编码,并且将结果插入位流110中。

[0084] 然后,在模块111～116中,编码器对帧的各个编码块进行解码,以进行将来运动估计。这些步骤使得编码器和解码器能够具有相同的参考帧116。为了重建编码帧,在模块111中对量化后和变换后的各个残差块进行逆量化,并且在模块112中对其进行逆变换,以在像素域中提供相应的“重建”残差块。由于量化的损失,该“重建”残差块不同于在步骤106获得的原始残差块。

[0085] 接着,根据在106处选择的编码模式(INTER或INTRA),将该“重建”残差块添加到INTER预测结果块114或INTRA预测结果块113,以获得“预重建”块(编码块)。

[0086] 接着,在模块115中,通过一种或多种后滤波对“预重建”块进行滤波以获得“重建”块(编码块)。相同的后滤波器集成在编码器(在解码环路中)处和解码器处以采用相同的方式使用,从而在编码器端和解码器端获得完全相同的参考帧。该后滤波的目的是去除压缩伪影。

[0087] 图2示出与图1所示的视频编码器架构相对应的视频解码器架构。

[0088] 首先,在模块202中对视频流201进行熵解码。然后,在模块203中对所获得的各残差块(编码块)进行逆量化,并在模块204中对其进行逆变换,以获得“重建”残差块。这类似于编码器端处的解码循环的开始。

[0089] 接着,根据在位流201中表示的解码模式(INTRA型解码或INTER型解码),来构建预测结果块。

[0090] 在INTRA模式的情况下,基于在位流201中指定的INTRA预测模式来确定(205)INTRA预测结果块。

[0091] 在INTER模式的情况下,在熵解码202期间从位流提取运动信息。该运动信息包括例如采用HEVC和JVET形式的参考帧索引和运动矢量残差。

[0092] 运动矢量预测结果是以与由编码器(从相邻块)使用运动矢量字段数据211中所存储的已计算出的运动矢量所进行的方式相同的方式获得的。因而,将该运动矢量预测结果添加(210)到所提取的运动矢量残差块以获得运动矢量。将该运动矢量添加到运动矢量字段数据211,以用于预测下一解码运动矢量。

[0093] 运动矢量也用于定位作为INTER预测结果块的参考帧206中的参考区域。

[0094] 接着,将在204处获得的“重建”残差块添加到INTER预测结果块206或INTRA预测结果块205,从而以与编码器的解码环路相同的方式获得“预重建”块(编码块)。

[0095] 接着,如在编码器端处进行的操作那样,在模块207中对该“预重建”块进行后滤波(可以从位流201中检索到要使用的后滤波的信令)。

[0096] 如此获得了“重建”块(编码块),其形成作为解码器的输出的解压缩视频209。

[0097] 上述编码/解码处理可以应用于单色帧。然而,最常见的帧是通常由三个阵列的颜色样本组成的彩色帧,各阵列对应于“颜色分量”(例如,R(红色)、G(绿色)和B(蓝色))。图像的像素包括三个同位置/相应样本,其中各个同位置/相应样本针对各分量。

[0098] 在R、G、B分量之间通常具有高相关性。因而,在图像和视频压缩中,在处理帧之前通过在另一颜色空间中对帧进行转换来使颜色分量去相关,这很常见。最常见的格式是YUV(YCbCr),其中Y是亮度(或亮度)分量,并且U(Cb)和V(Cr)是色度(或色度)分量。

[0099] 为了减少要处理的数据量,可以对彩色帧的一些颜色分量进行子采样,这样使得针对三个颜色分量具有不同的采样比。子采样方案通常被表示为描述概念上的2像素高区域中的亮度样本和色度样本的数量的三部分比J:a:b。“J”定义概念区域的水平采样参考(即,以像素为单位的宽度),其通常为4。“a”定义J个像素的第一行中的色度样本(Cr,Cb)的数量,而“b”定义J个像素的第二行中的(附加)色度样本(Cr,Cb)的数量。

[0100] 利用这些子采样方案,色度样本的数量与亮度样本的数量相比有所减少。

[0101] 4:4:4YUV或RGB格式不提供子采样,并且对应于亮度帧和色度帧具有相同大小W×H的非子采样帧。

[0102] 4:0:0YUV或RGB格式具有仅一个颜色分量,因此对应于单色帧。

[0103] 示例性采样格式如下。

[0104] 4:2:0YUV格式在第一行中具有的色度样本与亮度样本的一半一样多,并且在第二行中不具有色度样本。因而,这两个色度帧是W/2像素宽和H/2像素高,其中亮度帧是W×H。

[0105] 4:2:2YUV格式在第一行中具有的色度样本与亮度样本的一半一样多,并且在第二行中具有的色度样本与亮度样本的一半一样多。因而,这两个色度帧是W/2像素宽和H像素高,其中亮度帧是W×H。

[0106] 4:1:1YUV格式在第一行中具有的色度样本与亮度样本相比少75%,并且在第二行中具有的色度样本与亮度样本相比少75%。因而,这两个色度帧是W/4像素宽和H像素高,其中亮度帧是W×H。

[0107] 在被子采样时,帧中的色度样本的位置与亮度样本位置相比发生偏移。

[0108] 图3示出4:2:0YUV帧的色度样本(三角形)相对于亮度样本(圆形)的示例性定位。

[0109] 图1的编码处理可以应用于输入帧的各颜色分量帧。

[0110] 由于颜色分量之间的相关性(RGB之间的相关性或YUV之间的剩余相关性,尽管进行了RGB到YUV转换),开发了跨分量预测(CCP)方法以利用这些(剩余)相关性,从而提高编码效率。

[0111] CCP方法可应用在编码或解码处理的不同阶段,特别是在第一预测阶段(以预测当前颜色分量)或在第二预测阶段(以预测分量的当前残差块)。

[0112] 一个已知的CCP方法是LM模式,其还称为CCLM(跨分量线性模型预测)。LM模式用于根据亮度Y(更具体地,根据(编码器端处或解码器端处的)重建亮度)来预测色度分量Cb和Cr(或者U和V)这两者。针对各分量生成一个预测结果。该方法在(色度和亮度)块级别(例如,在CTU(编码树单位)、CU(编码单位)级别、PU(预测单位)级别、子PU或TU(变换单位)级别)工作。

[0113] 图4使用流程图示出由编码器(以下用作参考)或解码器进行的、用于使用LM模式生成块预测结果的一般步骤,作为示例。

[0114] 在以下的说明中,示例性第一分量是色度,而示例性第二分量是亮度。

[0115] 考虑同一帧中的要编码或解码的当前色度块502(图5A)及其关联或相应(即,“同位置”)亮度块505(即,例如同一CU),编码器(或解码器)在步骤401中接收包括与当前亮度

块相邻的亮度样本503的相邻亮度样本集RecL,并且接收包括与当前色度块相邻的色度样本501的相邻色度样本集RecC(表示为402)。应当注意,对于一些色度采样格式和色度相位,如图5A所示,亮度样本504和503不直接邻接亮度块505。例如,在图5A中,为了获得左方行RecL'(503),仅需要第二个左方行,而不需要直接左方行。同样,对于上方线504,如图5A所示,也考虑将第二个上方线用于亮度样本的下采样。

[0116] 当使用色度采样格式(例如,4:2:0、4:2:2等)时,在步骤403中将相邻亮度样本集下采样到RecL'404以匹配色度分辨率(即,相应色度帧/块的样本分辨率)。因而,RecL'包括下采样后的与当前亮度块相邻的重建亮度样本504。由于下采样,RecL'和RecC包括相同数量的2N个样本(色度块502为N×N)。然而,在需要较少样本来获得RecL'的现有技术中,存在亮度边界的特定下采样。另外,即使RecL和RecC具有相同的分辨率,通过使用低通卷积滤波器,RecL'也可被视为RecL的去噪版本。

[0117] 在图5A的示例中,相邻亮度和色度样本集分别由下采样后的上方和左方相邻亮度样本以及上方和左方相邻色度样本组成。更准确地,这两个样本集各自由相应亮度或色度块的紧邻左方边界的第一线和紧邻上方边界的第一线组成。由于下采样(图5A中的4:2:0),从非下采样的重建亮度样本RecL(左方或上方)的两个线获得相邻亮度样本RecL'的单个线。

[0118] US 9,565,428建议使用子采样,该子采样选择仅用于上方线(即,邻接亮度块的上方边界)而不用于亮度块本身的单个样本(如以下参考步骤408所述)。在图6A中示出所提议的子采样。这种方法的动机是减少上方线的线缓冲。

[0119] 由一个或两个参数(斜率α和偏移β)定义的线性模型是从RecL'(在存在的情况下,否则为RecL)和RecC推导出的。这是用以获得参数406的步骤405。

[0120] LM参数α和β是使用以下等式,通过基于最小均方的方法来获得的:

[0121]
$$\alpha = \frac{M.\sum_{i=1}^{M} RecC_i.RecL'_i - \sum_{i=1}^{M} RecC_i . \sum_{i=1}^{M} RecL'_i}{M.\sum_{i=1}^{M} RecL'^2_i - (\sum_{i=1}^{M} RecL'_i)^2} = \frac{A_1}{A_2}$$

$$\beta = \frac{\sum_{i=1}^{M} RecC_i - \alpha.\sum_{i=1}^{M} RecL'_i}{M}$$

[0122] 其中:M是取决于所考虑的块的大小的值。在如图5A和5B所示的正方形块的一般情况下,M=2N。然而,基于LM的CCP可以应用于任何块形状,其中M例如是块高度H+块宽度W的总和(对于矩形块形状而言)。

[0123] 应当注意,可以调整该等式中的用作权重的M的值,以避免编码器和解码器处的计算溢出。确切地,当使用具有32位或64位带符号体系结构的算法时,这些计算中的一些计算有时可能溢出,由此导致未指定的行为(其在任何跨平台标准中都是严格禁止的)。为了应对该情形,可以评价可能最大幅值给定输入RecL'和RecC值,并且可以相应地缩放M(并且反过来可以缩放以上总和),以确保不会发生溢出。

[0124] 参数的推导通常由图5A所示的样本集RecL'和RecC组成。

[0125] 提出了样本集的变形。

[0126] 例如,US 9,288,500提出了三个竞争的样本集,这些样本集包括由邻接上方边界的外线和邻接左方边界的外线组成的第一样本集、仅由邻接上方边界的外线组成的第二样

本集、以及仅由邻接左方边界的外线组成的第三样本集。这三个样本集在图6B中仅针对色度块示出(因此可被转置到亮度块)。

[0127] US 9,462,273将第二样本集和第三样本集扩展到扩展外线(通常使外线的长度加倍)的附加样本。扩展后的样本集在图6C中仅针对色度块示出。该文献还提供了减少可用的LM模式的数量,以降低用于表示位流中所使用的LM模式的信令成本。该减少可以是与上下文有关的,例如基于针对关联亮度块所选择的帧内模式。

[0128] US 9,736,487提出了与US 9,288,500的样本集相似的三个竞争样本集,但每次都由平行且紧邻所考虑的边界的两个线的外部相邻样本组成。这些样本集在图6D中仅针对色度块示出。

[0129] 此外,US 9,153,040和同一专利族的文献提出了由各边界的单个线组成的附加样本集,其中各线的样本小于先前集合。

[0130] 返回到图4的处理,使用具有所推导出的一个或多个参数406的线性模型,因而可以从在505中表示的当前亮度块的重建亮度样本407获得色度块502的色度帧内预测结果413。再次地,如果使用色度采样格式(例如,4:2:0、4:2:2等),则在步骤408中将重建亮度样本下采样到L'409以匹配色度分辨率(即,相应色度帧/块的样本分辨率)。

[0131] 可以使用与步骤403相同的下采样,或者由于线缓冲原因而使用另一下采样。例如,可以使用6抽头滤波器来提供下采样值,作为下采样位置周围的左上样本、上方样本、右上样本、左下样本、下方样本和右下样本的加权和。当一些周围样本缺失时,代替6抽头滤波器,仅使用2抽头滤波器。

[0132] 通过应用到重建亮度样本L,按照如下获得示例性6抽头滤波器的输出L':

[0133] $L'[i,j]=(2\times L[2i,2j]+2\times L[2i,2j+1]+L[2i-1,2j]+L[2i+1,2j]$

[0134] $+L[2i-1,2j+1]+L[2i+1,2j+1]+4)>>3$

[0135] 其中:(i,j)是下采样块内的样本的坐标,并且>>是位右移运算。

[0136] 此外,如在US2017/0244975中所述,可以使用自适应亮度下采样。仅使用亮度块的内容来确定对亮度块的各重建亮度样本使用哪个下采样滤波器。可利用1抽头滤波器。该方法的动机是避免下采样后的亮度块中的边缘的传播。

[0137] 由于下采样步骤408,L'和C块(色度块502中的色度样本的集合)包括相同数量的N^2个样本(色度块502是N×N)。

[0138] 接着,按照如下的公式,使用循环410-411-412来计算色度帧内预测结果PredC 413的各样本:

[0139] $PredC[i,j]=\alpha.L'[i,j]+\beta$

[0140] 其中:(i,j)是色度和亮度块内的所有样本的坐标。

[0141] 为了避免相除和相乘,可以基于查找表和移位运算使用不太复杂的方法来实现计算。例如,可以按照如下进行实际色度帧内预测结果推导411:

[0142] $PredC[i,j]=(A.L'[i,j])>>S+\beta$

[0143] 其中:S是整数,并且A是使用前面所述的查找表从(在计算α和β时以上引入的)A1和A2推导出的。它实际上对应于α的重新缩放后的值。运算(x>>S)对应于位右移运算,其等同于将x(带截断)整数除以2^S。

[0144] 当解析下采样后的亮度块的所有样本时(412),色度帧内预测结果413可用于在编

码器端处从色度块502中减去(以获得色度残差块),或者用于在解码器端处相加到色度残差块(以获得重建色度块)。

[0145] 注意,色度残差块可能不重要的因而被丢弃,其中在这种情况下,所获得的色度帧内预测结果413直接对应于预测色度样本(形成色度块502)。

[0146] 定义了HEVC标准的两个标准化组ITU-T VCEG(Q6/16)和ISO/IEC MPEG(JTC 1/SC 29/WG 11)正在共同协作努力研究HEVC的继任者的将来视频编码技术(被称为联合视频探索组(JVET))。联合探索模型(JEM)包含HEVC工具和该JVET组所选择的新添加工具。特别地,如在文献JVET-G1001中所述,本参考软件包含一些CCP工具。

[0147] 在JEM中,对于色度编码,总共允许11个帧内模式。这些模式包括五个传统的帧内模式和(在位流101、201中表示的)用以根据Y来预测Cb的六个跨分量LM模式、以及用以根据Cb来预测Cr的一个跨分量LM模式。

[0148] 六个Y到Cb CC LM模式其中之一是上述的CCLM,其中如图5A所示,相邻亮度和色度样本集RecL'和RecC各自均由各自的亮度或色度块的紧邻左方边界的第一线和紧邻上方边界的第一线组成。

[0149] 五个其它的Y到Cb CC LM模式是基于被称为多模型(MM)的特殊推导。这些模式被标记为MMLM。

[0150] 与CCLM相比,MMLM模式使用两个线性模型。将来自RecL'集的相邻重建亮度样本和来自RecC集的相邻色度样本分类为两组,各组用于推导一个线性模型的参数α和β,从而得到两组线性模型参数(α_1,β_1)和(α_2,β_2)。

[0151] 例如,阈值可被计算为形成RecL'的相邻重建亮度样本的平均值。接着,将具有RecL'[i,j]≤阈值的相邻亮度样本分类为组1;而将具有RecL'[i,j]>阈值的相邻亮度样本分类为组2。

[0152] 接着,根据以下公式来获得色度帧内预测结果(或当前色度块602的预测色度样本):

[0153] $PredC[i,j]=\alpha_1.L'[i,j]+\beta_1$,$L'[i,j]\leq$threshold的情况

[0154] $PredC[i,j]=\alpha_2.L'[i,j]+\beta_2$,$L'[i,j]>$threshold的情况。

[0155] 另外,与CCLM相比,MMLM模式使用相邻亮度和色度样本集RecL'和RecC,其各自由所考虑的块的平行且紧邻左方边界和上方边界的两个线的外部相邻样本组成。在图5B中示出例示4:2:0采样格式的示例,对于该4:2:0采样格式,从四个线的非下采样的重建亮度样本(使用下采样)获得两个线的相邻亮度样本。

[0156] 五个MMLM模式在用以对重建亮度样本进行下采样以匹配色度分辨率(以获得RecL'和/或L')的五个不同的下采样滤波器方面彼此不同。

[0157] 第一MMLM模式依赖于与在CCLM中使用的相同的6抽头滤波器(参见图7的参考标记701中的6个黑点)。第二MMLM模式至第四MMLM模式依赖于2抽头滤波器,这些2抽头滤波器分别提供下采样值作为以下的加权和:

[0158] -下采样位置周围的(6抽头滤波器所使用的)六个样本中的右上和右下样本(参见滤波器1,图7的702):

[0159] $L'[i,j]=(L[2i+1,2j]+L[2i+1,2j+1]+1)>>1$(这同样适用于RecL),

[0160] -下采样位置周围的(6抽头滤波器所使用的)六个样本中的下方和右下样本(参见

滤波器2,图7的703):

[0161] L′[i,j]=(L[2i,2j+1]+L[2i+1,2j+1]+1)>>1,以及

[0162] -下采样位置周围的(6抽头滤波器所使用的)六个样本中的上方和右上样本(参见滤波器4,图7的705):

[0163] L′[i,j]=(L[2i,2j]+L[2i+1,2j]+1)>>1。

[0164] 第五MMLM模式依赖于4抽头滤波器,该4抽头滤波器提供下采样值作为下采样位置周围的(6抽头滤波器所使用的)六个样本中的上方、右上、下方和右下样本的加权和(参见滤波器3,图7的704):

[0165] L′[i,j]=(L[2i,2j]+L[2i,2j+1]+L[2i+1,2j]+L[2i+1,2j+1]+2)>>2。

[0166] 如上所述,CCLM或MMLM模式必须在位流110或201中表示。图8示出JEM的示例性LM模式信令。第一二进制标志表示当前块是使用LM模式还是其它帧内模式(包括所谓的DM模式)来预测的。在LM模式的情况下,需要表示六个可能的LM模式。第一MMLM模式(使用6抽头滤波器)是通过将一个第二二进制标志设置为1来表示的。该第二二进制标志针对其余模式设置为0,其中在这种情况下,第三二进制标志被设置为1以表示CCLM模式,并且针对其余MMLM模式被设置为0。然后,使用两个附加的二进制标志来表示其余四个MMLM模式其中之一。

[0167] 针对各色度分量表示一个模式。

[0168] 以上引入的Cb到Cr CCLM模式用在DM模式中,并且应用于残差级别。实际上,DM模式将亮度在预定位置中所使用的帧内模式用于色度。传统上,像HEVC这样的编码模式使用与CU的左上角同位置的一个单一DM模式。在无需涉及过多细节的情况下并且为了清楚起见,JVET提供多个这样的位置。然后,使用该模式来确定预测方法,因此针对色度分量创建通常的帧内预测,其中该色度分量当从参考/原始数据中减去时,产生上述残差数据。通过以下公式来从Cb残差(以下称为ResidualCb)获得Cr残差的预测:

[0169] PredCr[i,j]=α.ResidualCb[i,j]

[0170] 其中:α是以与CCLM亮度到色度预测相同的方式推导出的。唯一的不同之处是在误差函数中添加了相对于默认α值的回归成本,使得所推导出的缩放因子按照如下偏向默认值-0.5:

[0171] $$\alpha = \frac{M.\sum_{i=1}^{M} RecCb_i.RecCr_i - \sum_{i=1}^{M} RecCb_i . \sum_{i=1}^{M} RecL'_i + \lambda\,(-0.5)}{M.\sum_{i=1}^{M} RecCb_i^2 - (\sum_{i=1}^{M} RecCb_i)^2 + \lambda}$$

[0172] 其中:$RecCb_i$表示相邻重建Cb样本的值,$RecCr_i$表示相邻重建Cr样本,并且

[0173] $\lambda = \sum_{i=1}^{M} RecCb_i^2 \gg 9$。

[0174] 已知的LM模式特别是当使用最小二乘法推导线性模型参数时,表现出大的计算复杂度。

[0175] 本发明试图改善编码效率和/或计算复杂度方面的情形。

[0176] 本发明是基于用基于直线的等式对线性模型的参数的确定来代替用于从亮度块样本计算色度预测结果块样本的线性模型的推导。该直线由基于块的邻域中的重建样本对所定义的两个样本对来定义。首先,确定要使用的两个样本对。然后,从这两个样本对来确定线性模型的参数。通过将线性模型的确定中所使用的样本对的数量限制到两个,可以避

免使用最小均方法。因此,所提出的方法与使用最小均方法的已知方法相比计算密集度更小。

[0177] 图9通过在这里考虑当前块的邻域内的样本对的集合中的亮度样本值的最小值和最大值来示出该方法的原理。所有的样本对都是根据其色度值和亮度值而绘制在该图上的。在该图上标识出两个不同的点(即,点A和点B),各点对应于样本对。点A对应于具有来自RecL'的最低亮度值x_A及其来自RecC的同位置色度值y_A。点B对应于具有最高亮度值x_B及其同位置色度值y_B的样本对。

[0178] 图10给出所提出的用以推导线性模型参数的方法的流程图。该流程图是图4的简化版。该方法是基于在步骤1001中获得的相邻亮度样本RecL'和在步骤1002中获得的色度样本RecC。

[0179] 在步骤1003中,确定与两个样本对相对应的两个点A和B(1004)。在第一实施例中,这两个点A和B对应于分别具有最低和最高亮度样本值x_A和x_B及其相应色度样本值y_A和y_B的样本对。

[0180] 然后,在步骤1005中,根据以下等式来计算穿过点A和B的直线等式:

[0181] $$\alpha = \frac{y_B - y_A}{x_B - x_A}$$

[0182] $\beta = y_A - \alpha x_A$。

[0183] 所获得的α、β是用于生成色度预测结果的线性模型参数1006。

[0184] 基于现有技术中所使用的LMS算法的线性模型推导具有一定复杂度。在该已知方法中,通过以下等式来获得模型的α参数的计算:

[0185] $$\alpha = \frac{M\sum_{i=1}^{M} \mathrm{Re}cC_i \,\mathrm{Re}cL'_i - \sum_{i=1}^{M} \mathrm{Re}cC_i \sum_{i=1}^{M} \mathrm{Re}cL'_i}{M\sum_{i=1}^{M} \mathrm{Re}cL'^{\,2}_i - \left(\sum_{i=1}^{M} \mathrm{Re}cL'_i\right)^2} = \frac{B_1 - B_2}{B_3 - B_4} = \frac{A_1}{A_2}$$

。

[0186] 对与计算复杂度有关的该等式的分析给出以下结果。计算B_1需要M+1次相乘和M次求和,其中M是样本对的数量。计算B_2需要1次相乘和2M次求和。计算B_3需要M+1次相乘和M次求和,并且计算B_4需要1次相乘和2M次求和。计算与$\frac{B_1 - B_2}{B_3 - B_4}$相对应的α需要两次附加求和和一次相除。

[0187] 为了计算β,需要一次相乘和2M+1次求和以及一次相除。如前面所述,M是样本对$RecC_i$和$RecL'_i$的数量。

[0188] 因此,α和β的LMS推导的复杂度是(2M+2+2)次相乘、(7M+3)次相加和两次相除。

[0189] 相比之下,对基于仅使用两个点的直线的等式的计算的所提出的方法的分析给出了以下结果。如所报道的,推导步骤1005仅需一次相乘、三次求和和一次相除。在生成线性模型参数方面的该大的复杂度降低是所提出的发明的主要优点。

[0190] 应当注意,搜索最小值和最大值本身具有通常与排序算法有关的复杂度。该操作不是完全连续的:可以将N个点与N个其它点进行比较,从而生成N个最小值/最大值。然后,可以将N/2个最小值点和N/2个最大值点与N/2个其它点进行比较,然后再与N/4个其它点进

行比较,等等,直到仅保留期望数量的最小值点和最大值点为止。通常,搜索最小值和最大值如此得到约2*N-2次比较(各自为N-1)。

[0191] 如已经说明的,可以利用整数相乘和移位而非浮点相乘、以及计算斜率时的相除来计算色度预测结果。该简化涉及用

[0192] $$pred_C(i,j) = \left(L \cdot rec_L'^{(i,j)}\right) \gg S + \beta$$

[0193] 来替换

[0194] $$pred_C(i,j) = \alpha \cdot rec_L'^{(i,j)} + \beta。$$

[0195] 为了仅使用整数相乘和移位,在一个实施例中,按照如下获得直线等式:

[0196] S=10

[0197] $$L = \frac{(y_B - y_A) \ll S}{x_B - x_A}$$

[0198] $\beta = y_A - L(x_A >> S)$。

[0199] 请注意,如果用L和S代替α,则在下文中β参考该等式,否则β参考传统等式$\beta = y_A - \alpha x_A$。

[0200] 这种推导的另一优点是偏移值S始终具有相同的值。这对于在利用该性质时可以变得更简单的硬件实现而言特别令人关注。

[0201] 在又一实施例中,由于L可能大、并且需要更大的乘数运算,因此S的值被迫低。实际上,将8位值乘以8位值与例如8*16乘法器相比更容易实现。L的典型实际值通常等同于小于8位的乘数。

[0202] 然而,优选实施例是被称为固定点的实现:对于有可能被量化(例如,将2D+0和2D+1的结果存储为单个值)的$D = (x_B - x_A)$的所有值,将(1<<S)/D的值存储在表中。优选地,这些值仅用于正值,因为可以容易地检索到符号。使用数组TAB,L的计算变为:

[0203] $$L = \begin{cases} (y_B - y_A) * TAB[abs(x_B - x_A)/Q], & x_B - x_A \geq 0的情况 \\ -1 * (y_B - y_A) * TAB[abs(x_B - x_A)/Q], & 其它情况 \end{cases}$$

[0204] Q控制量化,并由此控制表中的元素的数量。因而,使用Q=1意味着无量化。此外,注意,作为代替,查询索引可以为$(abs(x_B - x_A) + R)/Q$,通常具有R=Q/2、或其相除舍入的变体。结果,Q理想地是2的幂,使得除以$Q = 2^P$等同于右移了P。

[0205] 最后,该表中的一些值可以不等于0:$abs(x_B - x_A)$或$abs(y_B - y_A)$的低值通常得到L的非常差的估计。然后可以使用预先确定的或明确的(诸如在条带头部或者诸如PPS或SPS等的参数集中的)值。例如,对于D低于4的所有值,数组TAB可以包含默认值,例如-(1<<S)/8。

[0206] 对于10位内容和Q=1,需要数组中的多达2048个条目。通过利用如上所示的带符号的对称,这可以减少到1024。进一步增加Q将同样地减小TAB的大小。

[0207] 如果这些样本中的一些样本(RecL或RecC或这两者)是残差样本(即,由于可能被量化的两个块之间的差而得到的残差样本本身),如在具有Cb到Cr预测的JVET的情况那样,则可以相应地改变表的大小(和内容)。

[0208] 在另一优选实施例中,计算以下公式:$pred_C(i,j) = \alpha \cdot rec_L'^{(i,j)} + \beta$中的直线

的两个参数α和β的确定,使得整数运算可用于容易地在硬件中实现。

[0209] 更准确地,参数α和β的确定可以仅通过对整数的整数相乘和位移位运算来完成。这样的计算与诸如浮点运算等的其它类型的计算相比利用更少的计算资源(例如,内存和时间)。

[0210] 为了进行该整数运算,进行以下步骤。

[0211] 第一中间值“pshift”是通过考虑到要处理的亮度和色度样本的位深度来确定的。该位移位值确保分母(称为“diff”)α的指定最大值α。在本实施例中,“diff”的最大值为512,并且如此可以在具有512个条目的表中表示。通过强制“diff”的特定最大值,可以将共同的表(或表的集合)用于各种不同的位深度,从而减少总内存需求。

[0212] 因而,pshift的值取决于样本的位深度,这是因为,例如,如果使用10位来对样本进行编码,则最大值与最小值之间的最大差为1024。为了在512个条目的表中表示这一点,必须将该差除以2或使该差位移了1位,因此pshift=1。例如,pshift和bitdepth之间的关系可以是针对以下的表1a所提取的、或者是通过以下的表达式给出的:

[0213] pShift=(BitDepth>9)?BitDepth-9:0

[0214] 这可以通过以下的表达式来可替代地表示:

[0215] $$pShift = \begin{cases} BitDepth - 9, & BitDepth > 9\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0216] 还可以计算可选的舍入值“add”,以使“diff”在位移位之后成为整数。根据以下的表达式,“add”值与pshift有关:

[0217] add=pshift?1<<(pshift-1):0

[0218] 这可以通过以下表达式来可替代地表示:

[0219] $$add = \begin{cases} 2^{(pshift-1)}, & pshift > 0\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0220] 可替代地,可以通过以下的表达式提供“BitDepth”和“add”之间的直接关系:

[0221] add=(BitDepth>9)?1<<(BitDepth-10):0

[0222] 或者

[0223] $$add = \begin{cases} 2^{(Bitdepth-10)}, & Bitdepth > 9\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0224] 以下的表1给出与要处理的亮度和色度样本的位深度(其范围为从8位到16位)相对应的“pshift”和“add”的示例值。

[0225] 表1:“pshift”和“add”值的示例

[0226]

位深度	“pShift”的值	“add”的值
8或9位	0	0
10位	1	1
11位	2	2
12位	3	4
13位	4	8

14位	5	16
15位	6	32
16位	7	64

[0227] 该表可以存储在内存中,以避免需要重新计算“pshift”和“add”,这可以减少处理运算的数量。然而,某些实现可能使减少内存使用优先于减少处理运算的数量,并且如此可能每次均计算“pshift”和“add”。

[0228] 然后,“diff”值以适合于使用整数运算进行处理的方式来表示在亮度样本的最小值和最大值之间的范围值。“diff”的值是通过使用“pshift”而约束在某个范围内的整数。根据以下公式来计算该值“diff”:

[0229] $diff=(x_B-x_A+add)>>pshift$

[0230] 然后,计算α和β参数,应想到,α和β定义截取点A和B的线性模型的斜率和截距:

[0231] $$\alpha=\frac{y_B-y_A}{x_B-x_A}$$

[0232] $\beta=y_A-\alpha x_A$。

[0233] 如果表示α的“diff”的值等于零,则按照如下指派α和β参数:

[0234] $\alpha=0$

[0235] $\beta=y_A$(或$\beta=y_B$)。

[0236] 选择使用点A或B可以由哪个点当前存储在内存中来确定,以减少处理运算的数量。

[0237] 否则,如果值“diff”严格为正,则通过以下公式来确定α的值:

[0238] $\alpha = (((y_B - y_A) * Floor(2^k / diff) + div + add) >> pshift)$ (1)

[0239] 其中:函数Floor(x)提供小于或等于x的最大整数值,以及

[0240] 其中:按照如下计算中间参数“div”:

[0241] $div = ((y_B - y_A) * (Floor((2^k * 2^k) / diff) - Floor(2^k / diff) * 2^k) + 2^{(k-1)}) >> k$ (2)

[0242] 该相除的精度由变量“k”来表示。k=16的值已被发现提供最佳编码效率,并且允许使用整数运算来准确地表示α和β。当使用相应的亮度样本时,这将使得能够精确预测色度样本。

[0243] 如以下将更详细地说明,k的值还定义各条目需要多少内存。k=16的值允许在寻址表中的各个条目时使用可以以2字节表示的16位内存寄存器。

[0244] 参数β是通过将直线等式应用于直线的单个点(其可以是如下的点A或点B)来确定:

[0245] $\beta=y_A-((\alpha*x_A)>>k)$,或者

[0246] $\beta=y_B-((\alpha*x_B)>>k)$。

[0247] 选择使用点A或B可以通过哪个点当前存储在内存中来确定,从而减少处理运算的数量。可选地,点A或B可以是固定选择(例如,在标准中定义的选择)。

[0248] 从硬件的实现角度来看,可以用存储了预先计算值的表来替换公式(1)和(2)中的一些项。这样的表的主要优点是避免在每次进行α和β参数的推导时计算中间“Floor”函数。以这种方式,可以通过查找运算替换多个处理运算。

[0249] 例如,可以按照如下简化等式(1),以通过使用表TAB1[diff]来提供等式(3):

[0250] $\alpha = (((yB - yA) * TAB1[diff] + div + add) >> pshift)$ (3)

[0251] 其中:$TAB1[diff]=Floor(2^k/diff)$

[0252] 同样,可以通过使用预定的表TAB1[diff]和TAB2[diff]来简化公式(2),以避免迭代相同的运算。

[0253] $div = ((y_B - y_A) * (TAB2[diff] - TAB1[diff] * 2^k) + 2^{(k-1)}) >> k$ (4)

[0254] 其中:$TAB2[diff]=Floor((2^k*2^k)/diff)$

[0255] 可以通过以下等式来进一步简化该等式(4):

[0256] $div = ((y_B - y_A) * (TAB3[diff]) + 2^{(k-1)}) >> k$ (5)

[0257] 其中:

[0258] $TAB3[diff]=TAB2[diff]-TAB1[diff]*2^k=Floor((2^k*2^k)/diff)-2^k$

[0259] $*Floor(2^k/diff)$

[0260] TAB1和TAB3(以及TAB2)是各自具有N个条目$N=2^{(BitDetpth-pshift)}$的表,并且各条目由k位表示。

[0261] 根据上述整数变量“diff”的定义,并且例如取以10位表示的亮度或色度样本值,最大“diff”值为512(使用以上的表1a)。这意味着表TAB1和TAB3(以及TAB2)针对各自可以由具有512个条目的数组表示,并且各条目被编码在“k=16”位上。在表1中给出的“pshift”变量使得能够根据要处理的样本的位深度来获得相同数量的条目(这里为512)。

[0262] 如前面所述,这些数组(TAB1~TAB3)可以存储在内存中,以减少在等式(1)和(2)中为了推导参数α和β所要执行的运算的数量。

[0263] 在VVC标准化工作的范围内,这样的方法可以用于实现相除以检索线性模型的α和β参数,从而根据亮度样本来预测色度样本。然而,令人惊讶地发现,可以在不会不利地影响编码效率的情况下减小表的大小和各条目的表示。

[0264] 如上所述,存储各表所需的总内存取决于“pshift”值的值,并且可以通过使用“k”值来对用以表示各条目的位数进行编码。在上述实施例中,使用两个表(TAB1和TAB3),并且各表具有512个条目且k=16。表示这两个表TAB1和TAB3所需的内存是2*512*16=16384位,其可以存储在2048个字节上。

[0265] 现在将论述修改(独立地和共同地)确定内存需求的参数。

[0266] 各表(数组)中的条目的数量

[0267] 尽管通过使用大小为512个条目的表达到了良好的压缩,但这些表可被视为相当大,并且期望减小这些表的大小。

[0268] 表3和表4通过分别将条目数量从512个条目减少到256个条目和128个条目来示出根据Bjøntegard度量的编码效率影响(例如,参见用于说明如何确定这些度量的Bjontegard.Calculation of average PSNR differences between rd-curves.Doc.VCEG-M33 ITU-T Q6/16,April 2001.79Z.Xiong,A.Liveris,and S.Cheng)。通过使“pshift”的值增加1($N=2^{(BitDetpth-pshift)}$)来进行表的大小的减小。可以相应地调整“add”的值。在以下的表2中示出将N改变为256或128。

[0269] 表2:256或128个条目的表的“pshift”和“add”值的示例

[0270]

	256 个条目		128 个条目	
位深度	“pShift”的值	“add”的值	“pShift”的值	“add”的值
7 或以下	0	0	0	0
8 位	0	0	1	1
9 位	1	1	2	2
10 位	2	2	3	4
11 位	3	4	4	8
12 位	4	8	5	16
13 位	5	16	6	32
14 位	6	32	7	64
15 位	7	64	8	128
16 位	8	128	9	256

[0271] 该表可以通过以下的表达式来表示：

[0272] 256个条目

[0273] pShift＝(BitDepth>8)?BitDepth-8:0

[0274] 这可以可替代地通过以下表达式来表示：

[0275] $$pShift = \begin{cases} BitDepth - 8, & BitDepth > 8\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0276] 还可以计算可选的舍入值“add”,以使“diff”在位移位之后成为整数。根据以下表达式,“add”值与pshift有关：

[0277] add＝pshift?1<<(pshift-1):0

[0278] 这可以可替代地通过以下表达式来表示：

[0279] $$add = \begin{cases} 2^{(pshift-1)}, & pshift > 0\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0280] 可替代地,可以通过以下表达式来提供“BitDepth”和“add”之间的直接关系：

[0281] add＝(BitDepth>8)?1<<(BitDepth-9):0

[0282] 或者：

[0283] $$add = \begin{cases} 2^{(Bitdepth-9)} & Bitdepth > 8\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0284] 128个条目

[0285] pShift＝(BitDepth>7)?BitDepth-7:0

[0286] 这可以可替代地通过以下表达式来表示：

[0287] $$pShift = \begin{cases} BitDepth - 7, & BitDepth > 7\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0288] 还可以计算可选的舍入值“add”,以使“diff”在位移位之后成为整数。根据以下表达式,“add”值与pshift有关：

[0289] add＝pshift?1<<(pshift-1):0

[0290] 这可以可替代地通过以下表达式来表示：

[0291] $$add = \begin{cases} 2^{(pshift-1)}, & pshift > 0\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0292] 可替代地,可以通过以下表达式来提供“BitDepth”和“add”之间的直接关系：

[0293] add＝(BitDepth>7)?1<<(BitDepth-8):0

[0294] 或者：

[0295] $$add = \begin{cases} 2^{(Bitdepth-8)}, & Bitdepth > 7\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$

[0296] 减小表的大小得到样本值的最大值和最小值之间的差的更粗略表示。

[0297] 对在文献JVET-L1010中定义的JVET标准化协会所使用的一组视频序列进行编码效率评价测试。在以下的表中,负值示出更好的编码效率,而正值对应于编码效率的下降。

[0298] 表3:使用大小为256的表时的压缩性能

[0299]

	所有帧内主10		
	Y	U	V
类A1	-0.03%	0.11%	0.10%
类A2	0.00%	-0.02%	-0.02%
类B	0.02%	0.01%	0.01%
类C	0.03%	0.00%	0.00%
类D	0.08%	-0.46%	-0.46%
类E	0.01%	0.16%	0.16%
总体	0.01%	0.04%	0.13%

[0300] 如表3所示,尽管表TAB1和TAB3中的条目的数量减少了2倍,但编码效率令人惊讶地基本上没有受到影响。可以观察到,影响CCLM模式的修改所引入的损耗非常有限,并且这些损耗在色度通道(U)和(V)中小于0.2%(其基本上可忽略(并且更有可能表示噪声))。

[0301] 进行通过使pshift的值增加附加的1(如以上的表2所示)所生成的表大小128的类似实验。

[0302] 表4:使用大小为128的表时的压缩性能

[0303]

	所有帧内主10		
	K16S128		
	Y	U	V
类A1	0.03%	0.15%	0.08%
类A2	0.02%	0.04%	0.04%
类B	0.02%	-0.13%	-0.13%
类C	0.03%	0.06%	0.06%
类D	-0.01%	-0.19%	-0.19%
类E	-0.01%	-0.06%	-0.06%
总体	0.02%	0.00%	0.03%

[0304] 如表4所示,更令人惊讶的是,尽管表TAB1和TAB3中的条目的数量减少了4倍,但编码效率仍然基本上没有受到影响。可以观察到,影响CCLM模式的修改所引入的损耗非常有限,并且这些损耗在色度通道(U)和(V)中小于0.05%(其基本上可忽略(并且更有可能表示噪声))。

[0305] 然而,(通过使pshift增加附加的1)将表的大小进一步减少到64个条目,这样得到如以下的表5所示的压缩性能的更大损耗:

[0306]

	所有帧内主 10		
	K16S64		
	Y	U	V
类 A1	0.14%	0.39%	0.36%
类 B	0.00%	0.10%	0.10%
类 C	-0.02%	0.07%	0.07%
类 D	0.04%	-0.16%	-0.16%
类 E	0.04%	0.07%	0.07%

[0307] 这些结果是局部的,因为这些结果不包括类A2,因此没有“总体”数字。

[0308] 以上示出的结果表明,在不会不利地影响使用两个点A和B推导出参数α和β的CCLM模式的效率的情况下,可以将表的大小减少2倍甚至4倍。

[0309] 在另一实施例中,表中的条目的数量(即,pshift的值)可以根据位深度而变化(例如,对于多达10位为128(或256),并且对于超过10位为256(或512))。这可能是由于将需要更强大的编码器来对由(例如)16位所表示的样本进行编码这一事实,并且因此使用更大的表进行计算的复杂度将不会产生如此大的问题。在这种情况下,可以优先通过使用更大数量(例如,512或更多)的条目来略微提高编码性能。

[0310] 表示表(数组)中的各条目的位数

[0311] 为了进一步减小表的大小,表中的各条目也可以在小于针对各条目使用2个字节的初始k=16位上表示。减小k的值表示降低相除的精度,因为其基本上对应于减小用整数表示α的幅值。以下的表6示出(与k=16相比)在减少位数以表示各条目时的编码效率的影响。

[0312] 表6:减小k值时的编码性能

[0313]

	所有帧内主 10					
	k=14			k=12		
	Y	U	V	Y	U	V
类 A1	-0.02%	0.15%	0.08%	-0.02%	0.15%	0.08%
类 A2	0.00%	0.02%	0.02%	0.00%	0.02%	0.02%
类 B	0.01%	-0.18%	-0.18%	0.01%	-0.18%	-0.18%
类 C	-0.02%	0.01%	0.01%	-0.02%	0.01%	0.01%
类 D	0.02%	-0.24%	-0.24%	0.02%	-0.24%	-0.24%
类 E	0.04%	-0.14%	-0.14%	0.04%	-0.14%	-0.14%
总体	**0.00%**	**-0.04%**	**0.05%**	**0.00%**	**-0.04%**	**0.05%**
	k=10			**k=8**		
	Y	U	V	Y	U	V
类 A1	-0.02%	0.11%	0.08%	0.02%	0.25%	0.05%
类 A2	0.01%	-0.10%	-0.10%	0.02%	-0.02%	-0.02%
类 B	0.00%	-0.03%	-0.03%	0.03%	0.04%	0.04%
类 C	-0.03%	0.15%	0.15%	0.00%	-0.16%	-0.16%
类 D	0.04%	-0.54%	-0.54%	0.07%	0.08%	0.08%
类 E	0.02%	0.06%	0.06%	0.01%	-0.20%	-0.20%
总体	**0.00%**	**0.04%**	**0.16%**	**0.01%**	**-0.02%**	**0.07%**
	k=7			**k=6**		
	Y	U	V	Y	U	V
类 A1	0.02%	0.33%	0.05%	0.05%	0.80%	0.31%
类 A2	0.02%	0.07%	0.07%	0.03%	0.21%	0.21%
类 B	0.03%	-0.09%	-0.09%	0.05%	0.03%	0.03%
类 C	-0.03%	0.09%	0.09%	0.00%	0.22%	0.22%
类 D	0.01%	0.05%	0.05%	-0.01%	-0.04%	-0.04%
类 E	0.01%	0.09%	0.09%	0.02%	0.31%	0.31%
总体	**0.01%**	**0.08%**	**0.07%**	**0.03%**	**0.28%**	**0.28%**
	k=5			**k=4**		

	Y	U	V	Y	U	V
类 A1	0.30%	2.13%	1.04%	0.89%	6.18%	3.31%
类 A2	0.09%	0.83%	0.83%	0.30%	2.73%	2.73%
类 B	0.11%	0.31%	0.31%	0.33%	1.60%	1.60%
类 C	0.08%	0.75%	0.75%	0.40%	2.78%	2.78%
类 D	0.07%	-0.10%	-0.10%	0.26%	1.49%	1.49%
类 E	0.07%	0.50%	0.50%	0.15%	1.83%	1.83%
总体	**0.13%**	**0.83%**	**0.70%**	**0.40%**	**2.85%**	**2.47%**

[0314]

	k=3		
	Y	U	V
类 A1	2.46%	17.76%	10.99%
类 A2	1.06%	8.33%	8.33%
类 B	0.78%	6.86%	6.86%
类 C	1.27%	9.95%	9.95%
类 D	0.72%	5.90%	5.90%
类 E	0.33%	3.66%	3.66%
总体	**1.14%**	**9.07%**	**8.35%**

[0315] 以上的这些表6示出如下的令人惊讶的结果:当这些条目在8位或更少位上表示时,编码效率与k=16位相比基本上相同。

[0316] 上述表示出针对包括在6~8的间隔中的k获得了可接受的编码效率结果。针对在间隔9~15中的k实现了类似的结果,但这种表示将仍然需要2个字节,因此在减少存储该表所需的内存方面不会提供太大的优势。

[0317] 表6示出:针对等于5位或更少的k,由于计算α时的相除变得不准确,因此观察到更大的劣化。

[0318] 因而,令人惊讶地发现,在性能和存储之间的最佳折衷是k=8位。

[0319] 与各条目在2字节上表示的k=16位相比,本实施例可以仅使用单个字节来表示表TAB1或TAB3的条目。这降低了涉及k的所有计算的复杂度,因此减少了处理需求。

[0320] k=6、7或8的值令人惊讶地提供了与k=16相似的编码性能,其中k的更高值提供了略微更好的编码性能。

[0321] 特别令人惊讶的是,可以在没有看到编码性能的任何明显下降的情况下,使k减小完整字节(8位)。

[0322] 此外,令人惊讶的发现,k可以减小到低至k=6,并且仅在超出该范围时才检测到编码性能的明显下降。

[0323] 通过使k从16减小到8,各表所使用的总内存减少了2倍。

[0324] 这令人惊讶的,因为通常涉及位移位的运算的准确度受到对位移位参数的值的小变化强烈影响,并且当k的值即使减小了小的量时,也将预期编码性能的大幅度下降。然而,上述结果反直觉地表明,k值的大幅变化(例如,从16下降6)仅使编码性能下降了可忽略的量(<0.14%)。

[0325] k的选择可以根据位深度而变化(例如,对于多达10位为8,并且对于超过10位为16)。这可能是由于将需要更强大的编码器来对由(例如)16位表示的样本进行编码这一事

实,并且因此使用更大的表进行计算的复杂度将不会产生如此大的问题。在这种情况下,可以优先通过使用更大数量(例如,多于8)的位来略微提高编码性能。

[0326] 条目数量和表示各条目的位数的组合

[0327] 以下的表7示出当(与各表中的k＝16的基线和512个条目相比)在各表中存在256个条目(即,在以上的表3和表6中示出的两组结果的组合)时用于使k从8改变为5的编码性能的结果。

[0328] 表7:将k值减小和条目数量从512减小到256时的编码性能

[0329]

	所有帧内主 10					
	k=16			k=14		
	Y	U	V	Y	U	V
类 A1	-0.03%	0.11%	0.10%	0.01%	0.23%	0.16%
类 A2	0.00%	-0.02%	-0.02%	-0.02%	0.08%	0.08%
类 B	0.02%	0.01%	0.01%	0.01%	0.04%	0.04%
类 C	0.03%	0.00%	0.00%	-0.02%	0.18%	0.18%
类 D	0.08%	-0.46%	-0.46%	0.00%	-0.23%	-0.23%
类 E	0.01%	0.16%	0.16%	0.00%	0.04%	0.04%
总体	**0.01%**	**0.04%**	**0.13%**	**0.00%**	**0.11%**	**0.11%**
	k=12			k=10		
	Y	U	V	Y	U	V
类 A1	-0.01%	0.13%	-0.02%	-0.04%	-0.02%	0.08%

[0330]

类 A2	0.01%	-0.13%	-0.13%	0.03%	-0.12%	-0.12%
类 B	-0.01%	0.08%	0.08%	-0.01%	0.01%	0.01%
类 C	-0.03%	0.15%	0.15%	-0.02%	-0.01%	-0.01%
类 D	0.04%	-0.13%	-0.13%	0.03%	0.01%	0.01%
类 E	-0.04%	0.11%	0.11%	0.00%	-0.24%	-0.24%
总体	**-0.01%**	**0.07%**	**0.13%**	**-0.01%**	**-0.06%**	**0.20%**
	k=8			**k=7**		
	Y	U	V	Y	U	V
类 A1	0.00%	0.02%	-0.06%	0.01%	0.11%	0.01%
类 A2	0.02%	-0.05%	-0.05%	0.02%	-0.01%	-0.01%
类 B	0.03%	-0.04%	-0.04%	0.01%	-0.03%	-0.03%
类 C	0.00%	0.13%	0.13%	0.01%	-0.12%	-0.12%
类 D	0.01%	-0.12%	-0.12%	0.02%	0.04%	0.04%
类 E	0.00%	0.06%	0.06%	-0.03%	0.37%	0.37%
总体	**0.01%**	**0.02%**	**0.10%**	**0.01%**	**0.04%**	**0.03%**
	k=6			**k=5**		
	Y	U	V	Y	U	V
类 A1	0.03%	0.23%	0.08%	0.12%	0.80%	0.43%
类 A2	0.02%	0.02%	0.02%	0.06%	0.24%	0.24%
类 B	0.03%	-0.06%	-0.06%	0.06%	0.00%	0.00%
类 C	0.01%	0.17%	0.17%	0.02%	0.10%	0.10%
类 D	0.06%	-0.23%	-0.23%	0.06%	0.06%	0.06%
类 E	-0.01%	0.17%	0.17%	0.01%	0.20%	0.20%
总体	**0.02%**	**0.09%**	**0.14%**	**0.05%**	**0.23%**	**0.26%**
	k=4			**k=3**		
	Y	U	V	Y	U	V
类 A1	0.46%	2.62%	1.67%	1.36%	8.47%	5.60%
类 A2	0.16%	1.33%	1.33%	0.52%	4.45%	4.45%
类 B	0.17%	0.74%	0.74%	0.54%	3.53%	3.53%
类 C	0.18%	1.05%	1.05%	0.76%	5.21%	5.21%
类 D	0.16%	0.96%	0.96%	0.49%	3.29%	3.29%
类 E	0.13%	1.13%	1.13%	0.24%	2.85%	2.85%
总体	**0.21%**	**1.29%**	**1.21%**	**0.67%**	**4.77%**	**4.60%**

[0331] 以上编码效率结果示出如下的令人惊讶的结果:通过使用具有各自使用单个字节编码的256个条目的两个表(TAB1和TAB3)(即,k≤8),与具有在2个字节上表示的512个条目的两个表的情况相比,可以获得类似的结果。本实施例的特别优势是:在不会影响编码效率结果的情况下,使存储这些表TAB1和TAB3所需的内存减少了4倍。在该特定实施例中,表示两个表(TAB1和TAB3)所需的内存是:2*256*8=4092位,其可以存储在512个字节上。

[0332] 以下的表8示出当(与各表中的k=16的基线和512个条目相比)在各表中存在128个条目(即,在以上的表4和表6中示出的两组结果的组合)时用于使k从8改变为5的编码性能的结果。

[0333] 表8:将k值减小和条目数量从512减少至128时的编码性能

[0334]

	所有帧内主 10					
	k=16			k=14		
	Y	U	V	Y	U	V
类 A1	0.03%	0.15%	0.08%	0.01%	0.25%	0.15%
类 A2	0.02%	0.04%	0.04%	0.02%	-0.04%	-0.04%
类 B	0.02%	-0.13%	-0.13%	0.01%	-0.12%	-0.12%
类 C	0.03%	0.06%	0.06%	-0.03%	0.22%	0.22%
类 D	-0.01%	-0.19%	-0.19%	0.05%	0.02%	0.02%
类 E	-0.01%	-0.06%	-0.06%	-0.01%	0.04%	0.04%
总体	**0.02%**	**0.00%**	**0.03%**	**0.00%**	**0.06%**	**0.16%**
	k=12			**k=10**		
	Y	U	V	Y	U	V
类 A1	0.05%	-0.04%	-0.06%	0.00%	0.05%	0.20%
类 A2	0.01%	0.01%	0.01%	0.02%	-0.17%	-0.17%
类 B	0.00%	0.15%	0.15%	0.02%	-0.06%	-0.06%
类 C	-0.02%	0.11%	0.11%	0.00%	0.02%	0.02%
类 D	0.10%	-0.58%	-0.58%	0.05%	-0.29%	-0.29%
类 E	-0.04%	0.07%	0.07%	0.01%	-0.05%	-0.05%
总体	**0.00%**	**0.07%**	**0.09%**	**0.01%**	**-0.04%**	**0.13%**
	k=8			**k=7**		
	Y	U	V	Y	U	V
类 A1	0.03%	0.02%	0.04%	0.02%	0.07%	0.01%
类 A2	0.01%	-0.13%	-0.13%	0.01%	-0.01%	-0.01%
类 B	0.01%	-0.09%	-0.09%	0.01%	0.02%	0.02%
类 C	0.01%	-0.17%	-0.17%	-0.01%	0.11%	0.11%
类 D	0.01%	-0.25%	-0.25%	0.01%	0.08%	0.08%
类 E	0.00%	-0.20%	-0.20%	0.00%	-0.09%	-0.09%
总体	**0.01%**	**-0.11%**	**0.09%**	**0.01%**	**0.03%**	**0.06%**
	k=6			**k=5**		
	Y	U	V	Y	U	V
类 A1	0.04%	0.27%	0.17%	0.10%	0.35%	0.34%
类 A2	0.04%	-0.07%	-0.07%	0.03%	0.12%	0.12%
类 B	0.02%	-0.02%	-0.02%	0.04%	-0.03%	-0.03%
类 C	0.01%	0.19%	0.19%	0.03%	0.20%	0.20%

[0335]

类 D	0.07%	0.22%	0.22%	0.03%	-0.04%	-0.04%
类 E	0.03%	-0.04%	-0.04%	-0.01%	0.14%	0.14%
总体	**0.03%**	**0.06%**	**0.17%**	**0.04%**	**0.14%**	**0.21%**
	k=4			**k=3**		
	Y	U	V	Y	U	V
类 A1	0.27%	1.41%	0.98%	0.82%	4.99%	3.28%
类 A2	0.09%	0.84%	0.84%	0.30%	2.84%	2.84%
类 B	0.11%	0.18%	0.18%	0.31%	1.89%	1.89%
类 C	0.08%	0.60%	0.60%	0.38%	2.42%	2.42%
类 D	0.13%	0.33%	0.33%	0.26%	1.51%	1.51%
类 E	0.12%	0.47%	0.47%	0.20%	2.07%	2.07%
总体	**0.13%**	**0.64%**	**0.59%**	**0.39%**	**2.71%**	**2.78%**

[0336] 以上编码效率结果示出如下的令人惊讶的结果:通过使用具有各自使用单个字节编码的128个条目的两个表(TAB1和TAB3)(即,k≤8),与具有在2个字节上表示的512个条目的两个表的情况相比,可以获得类似的结果。特别令人惊讶的是,在多个示例中,使用128个条目与使用256个条目相比实际上提高了编码性能。例如,对于k=8(针对各条目一个字节),结果表明,表大小为128与256个条目的表相比使得提高编码性能。

[0337] 本实施例的特别优势是:在不会影响编码效率结果的情况下,使存储这些表TAB1和TAB3所需的内存减少了8倍。在该特定实施例中,表示两个表(TAB1和TAB3)所需的内存是:2*128*8=2046位,其可以存储在256个字节上。

[0338] 因此,CCLM模式可以使用该相除来检索通过整数运算可以实现的α和β参数,以进行高效硬件实现。

[0339] 特别是,已经表明,将减少表中的条目数量和减少各条目的大小进行组合,这不会(如预期那样)导致性能下降的恶化,相反,在将减少表中的条目数量和减少各条目的大小进行组合时,与独立地进行各操作相比,实现了基本上相同的性能。

[0340] 为了完整性,表9示出当与k=16和N=512的基线相比、使用N=64个条目的表时的部分结果。应当注意,在U和V分量中,性能损失对于(作为VVC的主要目标的)类A1而言极大。

[0341] 表9:将k值减小和条目数量从512减少至64时的部分编码性能结果

[0342]

	所有帧内主 10					
	k=16			k=14		
	Y	U	V	Y	U	V
类 A1	0.14%	0.39%	0.36%	0.11%	0.52%	0.49%
类 B	0.00%	0.10%	0.10%	0.02%	0.11%	0.11%
类 C	-0.02%	0.07%	0.07%	0.03%	0.08%	0.08%
类 D	0.04%	-0.16%	-0.16%	0.05%	-0.57%	-0.57%
类 E	0.04%	0.07%	0.07%	0.04%	-0.19%	-0.19%
	k=12			k=10		
	Y	U	V	Y	U	V
类 A1	0.13%	0.49%	0.56%	0.11%	0.39%	0.33%
类 B	0.04%	0.09%	0.09%	0.02%	-0.06%	-0.06%
类 C	0.00%	-0.04%	-0.04%	-0.04%	0.19%	0.19%
类 D	-0.01%	0.07%	0.07%	0.03%	-0.32%	-0.32%
类 E	0.01%	-0.14%	-0.14%	0.04%	-0.08%	-0.08%
	k=8			k=7		
	Y	U	V	Y	U	V
类 A1	0.10%	0.43%	0.34%	0.14%	0.57%	0.42%
类 B	0.03%	0.04%	0.04%	0.00%	-0.02%	-0.02%
类 C	-0.05%	0.27%	0.27%	-0.04%	0.19%	0.19%
类 D	0.07%	-0.64%	-0.64%	-0.04%	0.10%	0.10%
类 E	0.06%	-0.37%	-0.37%	0.03%	0.22%	0.22%
	k=6			k=5		
	Y	U	V	Y	U	V
类 A1	0.13%	0.39%	0.56%	0.14%	0.74%	0.35%
类 B	0.04%	0.10%	0.10%	0.03%	-0.04%	-0.04%
类 C	0.00%	0.05%	0.05%	0.00%	0.08%	0.08%
类 D	0.01%	-0.14%	-0.14%	0.03%	-0.01%	-0.01%
类 E	0.05%	-0.05%	-0.05%	0.07%	0.15%	0.15%
	k=4			k=3		
	Y	U	V	Y	U	V
类 A1	0.29%	1.20%	0.87%	0.63%	4.01%	2.39%
类 B	0.03%	0.22%	0.22%	0.23%	1.43%	1.43%
类 C	0.04%	0.19%	0.19%	0.25%	1.39%	1.39%
类 D	0.18%	-0.40%	-0.40%	0.18%	1.50%	1.50%
类 E	0.06%	0.70%	0.70%	0.16%	1.66%	1.66%

[0343] α的表示

[0344] 在另一实施例中,修改α参数的值,使得该值可以在“L”位上表示。通过上述的α和β的推导处理,当k等于8时,α的值可以多达17位(并且在k=16的情况下,多达25位)。修改α的值的主要原因是限制以下的预测公式中的相乘的位宽:

[0345] $C_C = ((\alpha * L_c) >> k) + \beta$

[0346] 其中:C_C是与亮度Lc值相对应的色度预测值,并且α和β是斜率的参数(其可以如上所述推导出)。

[0347] 如果亮度样本编码在10位上,则这意味着需要预测的核心环路以能够处理多达10位与17位的相乘,这在计算上是复杂的(并且可以利用使用大量内存)。在本实施例中,修改α的值,使得相乘不超过16位。这样的计算非常适合硬件实现(例如,16位处理器可以使用单个内存寄存器来进行计算)。

[0348] 为了实现此,α需要由6位或更少位来表示。为了达到该6位表示,可以"限幅"α的范围,以将更大的值强制到指定范围中。

[0349] 在另一实施例中,确定α的幅值,使得可以通过将该幅值除以适当量并相应地调整偏移参数k来减小该幅值。确定向偏移值(k)的调整量包括:求出"最高有效位"位置P(通过对例如前导0进行计数或者采用以2为底的对数所进行的经典运算)。在这种情况下,如果超出限制L(在优选实施例中为5或6),则进行以下运算:

[0350] $\alpha = \alpha >> (P-L)$

[0351] $k = k+L-P$

[0352] 即,将α除以因子$2^{(P-L)}$,并且k的值由相反量(L-P)来补偿。

[0353] L的值可以取决于位深度,但对于多个位深度可以是相同的,以简化实现。另外,L可以考虑符号位,即L=5。

[0354] 还观察到,在大多数实现情况中,在CCLM模式中,α的浮点值在范围[-2.0;2.0]内。例如,使用仅6位可以表示在间隔[-2.0;2.0]中的64个值,其中落在该范围外的任何计算值均被该范围的端点替换。

[0355] 在任何情况下,将α限幅到任何范围中可以在减小其幅值之前进行。这样确保了在进行幅值减小处理之前去除无关值。

[0356] 图11示出在本发明的实施例中选择两个点(A和B)的不同方式。

[0357] 所提出的推导的简化对编码效率有影响。为了降低该编码效率损失,仔细选择这两个点是非常重要的步骤。

[0358] 在如前面所述的第一实施例中,与图11的点A和B相对应地选择相邻亮度样本值的最小值和最大值。

[0359] 在替代实施例中,所选择的两个点是图11的点C和D,其对应于与相邻色度样本值的最小值和最大值相对应的一对亮度和色度样本。该替代实施例有时在编码效率方面是令人关注的。

[0360] 在替代实施例中,确定线段[AB]和线段[CD]之间的最长线段,并且如果[AB]线段长于[CD]线段,则选择点A和B,否则选择点C和D。可以利用欧几里德距离来计算各线段的长度。然而,还可以使用另一距离测量值。与前两个实施例相比,本实施例提高了编码效率。实际上,当所选择的两个点远时,通常所生成的线性模型是相关的。结果,所生成的色度块预测结果与当前块预测有关。

[0361] 在另一实施例中,在A、B、C、D之间可以生成的所有可能线段中的最长线段给出了所选择的两个点。这对应于如图11所示的线段[AB]、[CD]、[AC]、[AD]、[CB]和[DB]。本实施例与先前实施例相比以更高复杂度为代价来提高编码效率。

[0362] 在优选实施例中,设置表示RecL'亮度样本值的最小值和最大值的点以创建A和B

点,并且如果点A的一个分量等于来自B的相应分量($x_B=x_A$或$y_B=y_A$),则选择表示色度样本值C和D的最小值和最大值的点。该实施例获得了最佳编码效率,这是因为,如果$x_B=x_A$或$y_B=y_A$、则α(或L)分别是无穷大或等于0并且结果色度预测结果块不可用或等同于DC预测。只要表示α(或L)的分数的分子或分母过低,就是这种情况(例如,可以验证以下条件:|α|<0.1):即使小量的误差(诸如由于量化引起的误差等)也得到α(或L)的非常不同的值。在本文的其余部分中,基本上几乎为水平或垂直斜率的这些情况得到所谓的异常斜率(不论该斜率是α还是L)。

[0363] 在附加实施例中,测试如图11所示的多对点,直到α不是"异常"为止。本实施例提高了前一实施例的编码效率,但增加了计算复杂度。

[0364] 在一个替代实施例中,计算两个分量(色度和亮度)的最大值和最小值之间的差。另外,选择具有最大差的分量以确定定义模型参数的计算所用的线的两个点。当两个分量是两个色度分量或两个RGB分量时,本实施例是高效的。

[0365] 可以对当前块的样本值进行两个点A和B的选择。在一个实施例中,基于当前下采样后的亮度块(图5中的505)的样本值来设置简化的线性模型推导的两个点。将块的邻域中的样本对的亮度样本值与亮度块的亮度样本值进行比较。选择具有最大出现率的值以创建x_A,并且选择具有最大出现率的第二值以创建x_B。相应的色度值y_A和y_B是块的邻域内的样本对中的同位置色度样本的平均值。当α(或L)为"异常"(等于0或接近0(|α|<0.1))时,代替第二最多选择值,x_B是选择较少的亮度值其中之一。同样,y_B是同位置色度样本的平均值。本实施例与先前实施例相比以最高复杂度为代价来提高编码效率。

[0366] 两个点A和B的选择可以在样本对的空间位置上进行。

[0367] 在先前实施例中,需要确定M对亮度和色度相邻样本中的亮度(A,B)或/和色度(C,D)的最小值和最大值。这可被视为额外复杂度。因此,对于一些实现,优选以最小复杂度获得这两个点。

[0368] 在一个实施例中,利用色度样本RecC(图5中的501)以及利用边界RecL'的下采样亮度样本(503)来生成一个线性模型。所选择的第一点是亮度的附图标记为5004的左方行的下方样本以及同位置的色度样本5001。所选择的第二点是右上亮度样本5003和同位置的色度样本5002。这两个点的选择非常简单,但该选择与基于值的先前实施例相比效率也不高。

[0369] 另外,如果例如针对图像或条带的边界上的块不存在上方边缘或左方边其中之一、或者例如由于复杂度或错误恢复原因而不可用,则选择这两个样本(例如,可用边缘上的亮度为504或5003的样本)而不是缺失样本。因而,可以看出,存在多个条件以选择样本。

[0370] 描述了进一步的实施例,其中,如果可以选择不足的点来计算斜率、或者这些点得到"异常"α(或L),则作为代替,可以选择默认点。本实施例也可应用于具有自适应的MMLM模式。为了创建第一组的线性模型参数,第一点是亮度的第二左方行的下方样本(5009)和同位置色度样本(5005)。此外,第二点是第一左方行的上方亮度样本(5010)和同位置色度样本(5006)。

[0371] 为了创建第二组的线性模型参数,第一点是亮度的第一上方线的左方样本(5011)和同位置色度样本(5007)。并且,第二点是第二上方线的右方亮度样本(5012)和同位置色度样本(5008)。

[0372] 本实施例简化了对第一组和第二组的四个点的选择。

[0373] 在又一实施例中,MMLM模式的阈值是点5010或5011的亮度值(即,在该示例中为左方邻域的右上点和上方邻域的左下点)、或者这些点之间的平均值。该附加实施例简化了该阈值的计算。

[0374] 在针对与两个点的选择有关的所有这些实施例的又一实施例中,禁用亮度的下采样处理,并且这被抽取取代,即将两个亮度样本其中之一用于RecL'样本。在这种情况下,将跳过图10中的步骤1001和以下详细说明的图12中的步骤1201。本实施例在对编码效率的影响很小的情况下降低了复杂度。

[0375] 点A、B、C和D是基于样本的解码版本确定的,因此可能不匹配原始样本值。如在定义所谓的"异常"斜率时已经说明的,这可能导致异常短的线段或者仅仅是噪声估计。A和C是两个最低点,B和D是两个最高点,代替使用这四个点中的任两个点,以一些简单的补充运算为代价,可以使用被定义为A和C之间的平均值的点E、以及被定义为B和D之间的平均值的点F:

[0376] $x_E=(x_A+x_C+1)>>1$且$y_E=(y_A+y_C+1)>>1$

[0377] $x_F=(x_B+x_D+1)>>1$且$y_F=(y_B+y_D+1)>>1$

[0378] $$A=\frac{(y_E-y_F)\ll S}{x_E-x_F}$$

[0379] $\beta=y_E-A.(x_E>>S)$。

[0380] 显然,如果y_E-y_F或x_E-x_F等于0或过低(即,推导斜率为"异常"),则照常考虑点A、B、C和D,以获得更好的参数。

[0381] 由此应理解,模型中的斜率的计算中所用的两个点可以不是由RecL'或RecC的样本值组成的两个实际点。这解释了在步骤1003中使用"确定"措词而不是"选择"。

[0382] 在又一实施例中,对于MMLM模式,如果定义一个组的斜率的α(或L)参数是"异常",则相应的LM参数被设置成等于在多于两个组的LM参数的情况下的其它组或另一组的LM参数。图12示出本实施例。在步骤1203中确定定义了两个组的两个模型的(α1,β1)和(α2,β2)之后,在步骤1204和1205中测试α1和α2以检查这两者是否等于0。如果是这种情况,则"异常"斜率参数α(或L)被设置成等于另一斜率参数α,同样,在步骤1206和1207中也使用另一组的相应β参数值。因此,在这种情况下,不论当前块的下采样亮度样本值如何,都使用仅一组参数,不存在与阈值的比较,获得了与CCLM模式相同的复杂度。本实施例的优点是:由于无需推导附加的线性模型参数,因此可以在复杂度小的情况下提高编码效率。

[0383] 在替代实施例中,当一个斜率参数α(或L)为"异常"时,通过考虑MMLM的所有初始样本来重新推导一组线性参数(其对应于利用两个上方线和两个相邻行而不是一个上方线和一个相邻行的CCLM推导)。本实施例比先前实施例相比给出了更好的编码效率,但由于需要重新推导一组线性模型参数,因此更加复杂。

[0384] 如本文所述的利用两个点的简化LM推导除了当多个LM模式正竞争时不能替换所有的LMS推导以外,与经典的LMS推导相比通常效率不高。

[0385] 在一个实施例中,利用两个点的LM推导仅用于CCLM模式以推导色度块预测结果。本实施例给出了编码效率提高。

[0386] 在一个实施例中,利用两个点的推导仅用于MMLM模式,因为MMLM模式是最复杂的

预测方法。

[0387] 在一个实施例中,将利用两个点的LM推导用于CCLM模式和MMLM模式以推导色度块预测结果。本实施例具有与JEM相似的编码效率,但通过将该简化的LM推导用于色度块预测结果的生成降低了最坏情况的复杂度。实际上,基于亮度的色度预测是预测线性模型模式中的表现出最坏情况复杂度的模式。该色度预测与残差色度预测相比更为复杂。

[0388] 在一个实施例中,利用两个点的LM推导取代所有LMS推导(色度块预测结果生成和残差预测)。本实施例与JEM相比降低了编码效率,但显著降低了复杂度。请注意,不论在步骤1203中用于参数的推导方法如何,这两个实施例都给出了编码效率提高。

[0389] 在又一实施例中,如果斜率参数α(或L)中的一个或这两个是"异常",则在步骤1206和/或1207中作为代替使用默认值(诸如-(1<<S)/8等),并且计算相应的值β。

[0390] 在又一个实施例中,多个LM模式正在编码器侧竞争,并且在解码器侧语法元素可以在位流中表示所选择的LM模式。该信令可以是在条带级别(或PPS或SPS)以指示应使用哪些集合,或者至少提供用于块级别选择的候选。这些竞争的LM模式之间的差异至少之一是用于推导LM参数的两个点的集合。这两个点的集合和用于生成这两个点的方法定义了竞争中的不同LM模式。例如,对于一个LM模式,这两个点是基于最小和最大亮度值确定的,并且对于另一LM模式,这两个点是基于最大和最小色度值选择的。

[0391] 另一实施例涉及从如图5所示的可能位置定义多个集合。尽管四个这样的不同点可以得到多达十二个不同对,但可以优选这样的对:这些对在用于计算斜率参数α(或L)的等式中得到分子和分母的最大值。编码器建立对的列表,并且根据一些标准(诸如亮度分量中的距离、或使用亮度和色度分量这两者的笛卡尔距离)对这些对进行排序,从而可能地去除这些对中的一些对(即,该一些对的斜率彼此过于接近的情况),并由此建立可以选择和表示的参数的列表。

[0392] 这些实施例的优点是编码效率提高。

[0393] 这些实施例的说明提到了亮度和色度分量,但可以容易地适用于诸如色度分量或RGB分量这两者等的其它分量。根据实施例,本发明在根据第二色度分量预测第一色度分量样本值时使用。在另一实施例中,本发明在根据多于一个的分量的多于一个的样本值预测一个分量的样本值时使用。应理解,在这种情况下,线性模型是基于两个点/集合推导出的,各点/集合包含一个分量的样本值、以及多于一个的分量的多于一个的样本值。例如,如果使用两个分量的样本值来预测一个分量的样本值,则各点/集合可被表示为三维空间中的位置,并且线性模型是基于穿过三维空间中的与重建样本值的两个点/集合相对应的两个位置的直线。

[0394] 图13是用于实现本发明的一个或多个实施例的计算装置1300的示意框图。计算装置1300可以是诸如微计算机、工作站或轻型便携式装置等的装置。计算装置1300包括通信总线,其中该通信总线连接有以下组件:

[0395] -表示为CPU的诸如微处理器等的中央处理单元1301;

[0396] -表示为RAM的随机存取存储器1302以及寄存器,其中RAM 1302用于存储本发明实施例的方法的可执行代码,这些寄存器被适配为记录实现根据本发明实施例的用于对图像的至少一部分进行编码或解码的方法所需的变量和参数,其存储器容量例如可以利用连接至扩展端口的可选RAM来扩展;

[0397] -表示为ROM的只读存储器1303,用于存储实现本发明实施例所用的计算机程序;

[0398] -网络接口1304,其通常连接至发送或接收要处理的数字数据所经由的通信网络。网络接口1304可以是单个网络接口、或者包括不同的网络接口的集合(例如,有线接口和无线接口或者不同种类的有线接口或无线接口)。在CPU 1301中运行的软件应用的控制下,将数据包写入网络接口以供发送或者从网络接口读取数据以供接收;

[0399] -用户接口1305,其用于从用户接收输入或向用户显示信息;

[0400] -表示为HD的硬盘1306,其可被设置为大容量存储装置;

[0401] -I/O模块1307,其可用于相对于诸如视频源或显示器等的外部装置进行数据的接收/发送。

[0402] 可执行代码可以存储在只读存储器1303中、硬盘1306上或者例如盘等的可移除数字介质上。根据变形例,程序的可执行代码可以在执行之前利用通信网络经由网络接口1304来接收,从而存储在通信装置1300的诸如硬盘1306等的存储部件其中之一内。

[0403] 中央处理单元1301被适配为控制和引导根据本发明实施例的程序的指令或软件代码的一部分的执行,其中这些指令存储在上述存储部件其中之一内。在通电之后,CPU 1301例如能够在从程序ROM 1303或硬盘(HD)1306加载了来自主RAM存储器1302的与软件应用有关的指令之后,执行这些指令。这种软件应用在由CPU 1301执行的情况下,使得进行根据本发明的方法的各步骤。

[0404] 根据本发明的方法的任何步骤均可以通过利用可编程计算机器(诸如PC(“个人计算机”)、DSP(“数字信号处理器”)或微控制器等)执行一组指令或程序来以软件实现;或者通过机器或者专用组件(诸如特别是用于极小值和极大值选择的FPGA(“现场可编程门阵列”)、或ASIC(“专用集成电路”)等)以硬件实现。

[0405] 还应注意,尽管为了说明起见、一些示例基于HEVC,但本发明不限于HEVC。例如,本发明也可用在任何其它预测/估计处理中,其中可以利用模型来估计/预测两个或更多个分量的样本值之间的关系,其中该模型是基于至少从相关/关联分量样本值的所有可用集合中所选择的相关/关联分量样本值的至少两个集合所确定的近似模型。

[0406] 应理解,可以从阵列方面存储和/或处理与样本对(即,针对不同分量的关联样本值的集合)相对应的各点。例如,可以将各分量的样本值存储在阵列中,使得例如使用该样本值的索引,通过参考该阵列的元素可参考/可访问/可获得该分量的各样本值。可选地,阵列可用于存储并处理各样本对,这些样本对的各样本值是作为阵列的元素可访问/可获得的。

[0407] 还应理解,上述的比较、确定、评估、选择或考虑的任何结果(例如,在编码处理期间进行的选择)均可在位流的数据(例如,标志或表示结果的数据)中表示或者可从位流的数据确定,使得代替例如在解码期间实际进行比较、确定、评估、选择或考虑,可以将所表示或所确定的结果用在处理中。

[0408] 尽管以上已经参考具体实施例说明了本发明,但本发明不限于这些具体实施例,并且本领域技术人员将明白存在于本发明的范围内的变形。

[0409] 在参考仅以示例方式给出且并不意图限制本发明范围的前述例示实施例的情况下,许多其它修改和改变对本领域普通技术人员是不言自明的,其中所述范围仅由所附权利要求书来确定。特别地,在适当情况下,可以互换来自不同实施例的不同特征。

[0410] 以上所述的本发明的各实施例可以单独地或作为多个实施例的组合来实现。此外,在需要的情况下或在将各个实施例中的元件或特征组合成一个实施例有益的情况下,可以组合来自不同实施例的特征。

[0411] 除非另外明确说明,否则本说明书中公开的各特征均可被服务相同、等同或类似目的的替代特征所取代。因而,除非另外明确说明,否则所公开的各特征仅是一个通用系列的等同或相似特征的一个示例。

[0412] 在权利要求书中,词语“包括”没有排除其它元件或步骤,并且不定冠词“a”或“an”没有排除多个。在相互不同的其它方面中陈述不同的特征的仅有事实并不表明不能有利地使用这些特征的组合。

[0413] 以下编号的条款也定义了某些实施例:

[0414] 1.一种用于推导线性模型的方法,所述线性模型用于从同一帧中的第二分量块的第二分量样本获得第一分量块的关联的重建第一分量样本,所述方法包括:

[0415] 确定表示通过两个点的直线的线性方程的参数,各点由第一变量和第二变量这两个变量来定义,所述第一变量对应于第二分量样本值,所述第二变量对应于第一分量样本值;以及

[0416] 推导由直线参数定义的所述线性模型,

[0417] 其中,对所述参数的所述确定使用了整数运算。

[0418] 2.根据条款1所述的方法,其中,通过计算所述两个点的第一分量之间的差(diff)来估计与所述线性方程的斜率(α)相对应的参数,并且通过对所述差应用位移位来限制所述差的大小。

[0419] 3.根据条款2所述的方法,其中,所述位移位的值取决于样本的位深度。

[0420] 4.根据条款2或3所述的方法,其中,所述差的值的最大数量为256。

[0421] 5.根据条款4所述的方法,其中,所述位移位的值(pShift)通过下式与样本的位深度(BitDepth)有关:

[0422] $$pShift = \begin{cases} BitDepth - 8, & BitDepth > 8\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$。

[0423] 6.根据条款3至5中任一项所述的方法,其中,将与所述差相对应的值存储在表中,其中所述表中的条目的数量为256。

[0424] 7.根据条款3至5中任一项所述的方法,其中,将与所述差相对应的值存储在表中,其中所述表中的条目的数量在样本的位深度低于或等于阈值的情况下为256,并且在样本的位深度高于阈值的情况下为512。

[0425] 8.根据条款2或3所述的方法,其中,所述差的值的最大数量为128。

[0426] 9.根据条款8所述的方法,其中,所述位移位的值(pShift)通过下式与样本的位深度(BitDepth)有关:

[0427] $$pShift = \begin{cases} BitDepth - 7, & BitDepth > 7\text{的情况} \\ 0, & \text{其它情况} \end{cases}$$。

[0428] 10.根据条款8或9所述的方法,其中,将与所述差相对应的值存储在表中,其中所述表中的条目的数量为128。

[0429] 11.根据条款8或9所述的方法,其中,将与所述差相对应的值存储在表中,其中所述表中的条目的数量在样本的位深度低于或等于阈值的情况下为128,并且在样本的位深度高于阈值的情况下为256。

[0430] 12.根据条款7或11所述的方法,其中,所述阈值为10位。

[0431] 13.根据条款6至12中任一项所述的方法,其中,所述表中所存储的值通过包含Floor函数的函数对应于所述差。

[0432] 14.根据条款2至13中任一项所述的方法,其中,所述差被限制为整数。

[0433] 15.根据条款14所述的方法,其中,向所述差加上取决于被应用于所述差的位移位的值。

[0434] 16.根据条款1至15中任一项的方法,其中,确定与所述线性方程的斜率相对应的参数包括:确定相除,所述确定包括:通过对所述相除应用位移位来减小所述相除的幅值。

[0435] 17.根据条款16所述的方法,还包括:将所述位移位应用于表中所存储的与所述差相对应的值,以限制所述表中所存储的所述各条目的大小。

[0436] 18.根据条款16或17所述的方法,其中,减小所述幅值包括:将相除的值所使用的位数限制为一个字节。

[0437] 19.根据条款18所述的方法,其中,所述位数为6、7或8。

[0438] 20.根据条款18或19所述的方法,其中,所述位数取决于样本的位深度。

[0439] 21.根据条款1至20中任一项所述的方法,还包括:限制与所述线性方程的斜率相对应的参数的可能值的大小和/或数量。

[0440] 22.根据条款21所述的方法,还包括:将所述线性方程的斜率的值限幅在预定义范围内。

[0441] 23.根据条款22所述的方法,其中,在CCLM模式中,所述预定义范围对应于[-2.0,2.0]。

[0442] 24.根据条款17至23中任一项所述的方法,其中,限制与所述线性方程的斜率相对应的参数的可能值的大小和/或数量包括:通过对所述参数应用位移位来减小所述参数的幅值。

[0443] 25.根据在引用条款22的情况下的条款24所述的方法,其中,所述幅值的减小是在所述限幅之后进行的。

[0444] 26.根据条款24或25所述的方法,其中,减小所述幅值包括:限制所述参数的值所使用的位数。

[0445] 27.根据条款26所述的方法,其中,所述位数取决于样本的位深度。

[0446] 28.根据条款26所述的方法,其中,所述位数为5或6。

[0447] 29.一种用于对图像进行编码的装置,其中所述装置包括用于根据条款1至28中任一项来推导线性模型的部件。

[0448] 30.一种用于对图像进行解码的装置,其中所述装置包括用于根据条款1至28中任一项来推导线性模型的部件。

[0449] 31.一种用于可编程设备的计算机程序产品,所述计算机程序产品包括指令序列,所述指令序列用于在被载入所述可编程设备并由所述可编程设备执行时,实现根据条款1至28中任一项所述的方法。

[0450] 32.一种存储有程序的计算机可读介质,所述程序在由装置中的微处理器或计算机系统执行时,使得所述装置进行根据条款1至28中任一项所述的方法。

[0451] 33.一种计算机程序,其在执行时,使得进行根据条款1至28中任一项所述的方法。

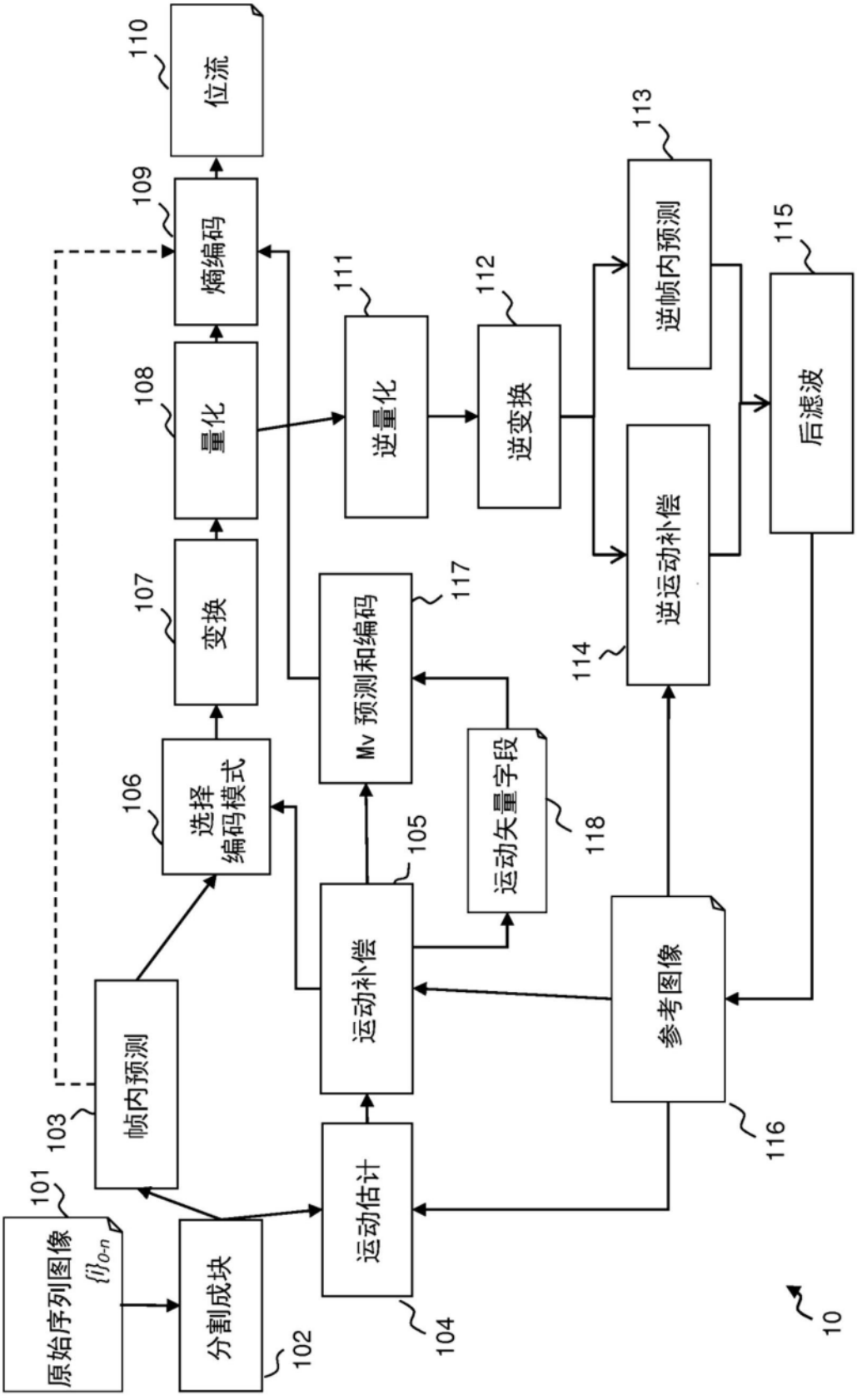

原始序列图像
$\{I_j\}_{0-n}$
101
分割成块
102
帧内预测
103
运动估计
104
运动补偿
105
选择编码模式
106
变换
107
量化
108
熵编码
109
位流
110
逆量化
111
逆变换
112
逆帧内预测
113
逆运动补偿
114
后滤波
115
参考图像
116
Mv 预测和编码
117
运动矢量字段
118
10

图1

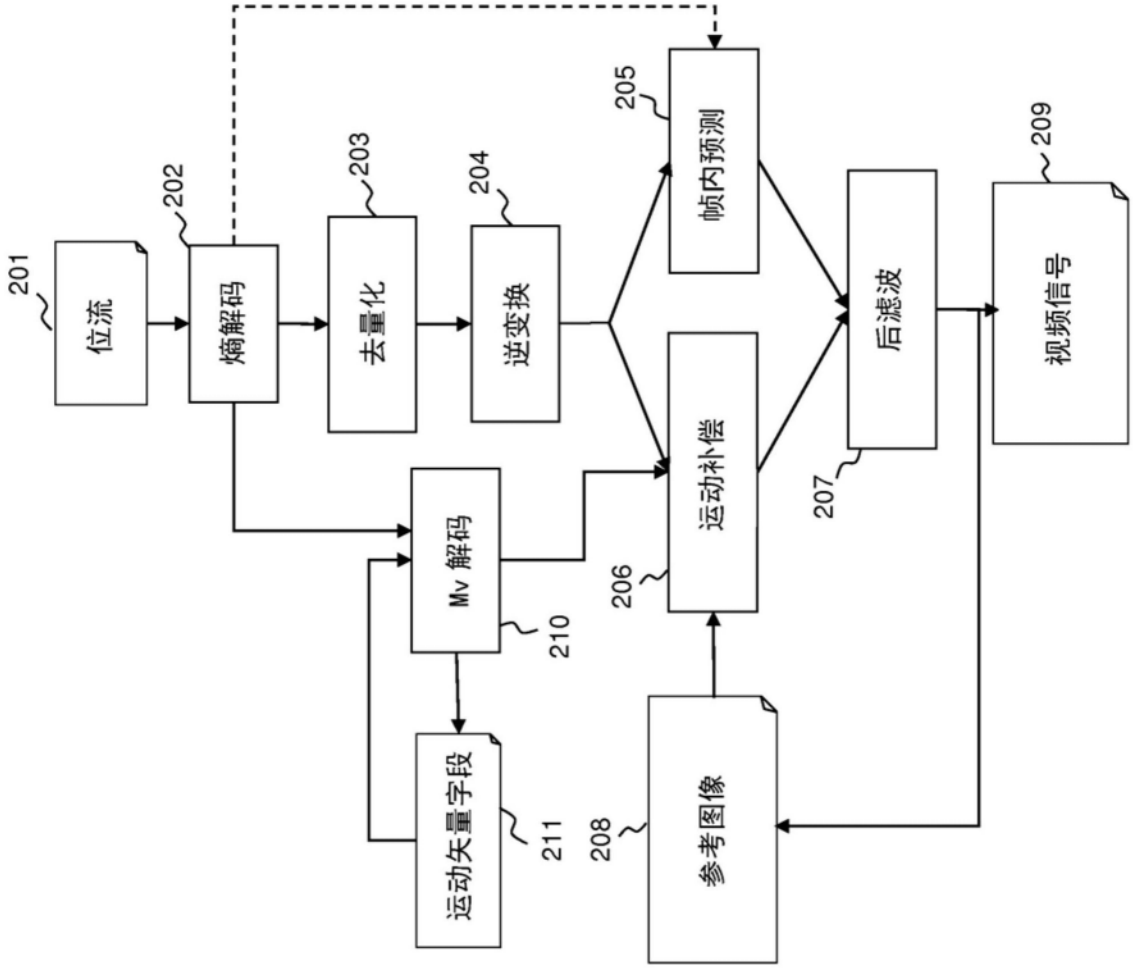

201
位流
202
熵解码
203
去量化
204
逆变换
205
帧内预测
209
视频信号
后滤波
207
运动补偿
206
Mv 解码
210
运动矢量字段
211
208
参考图像

图2

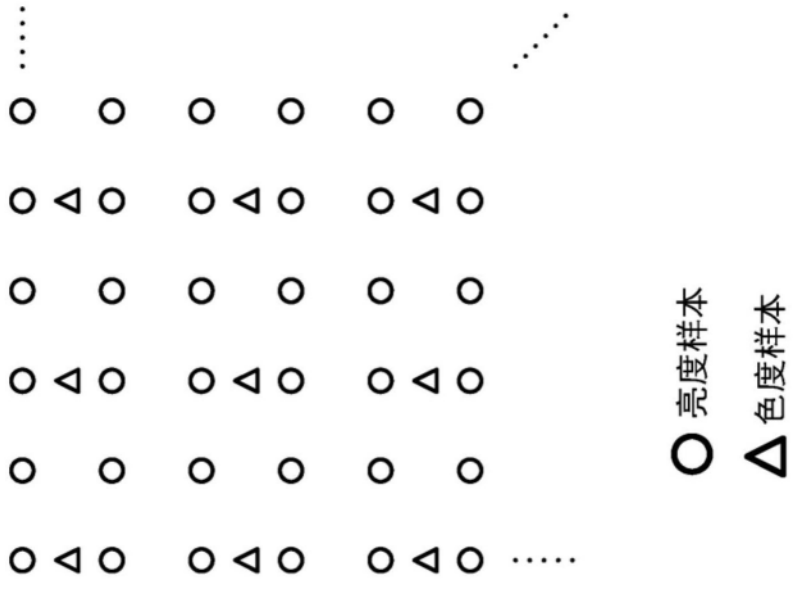

亮度样本
色度样本

图3

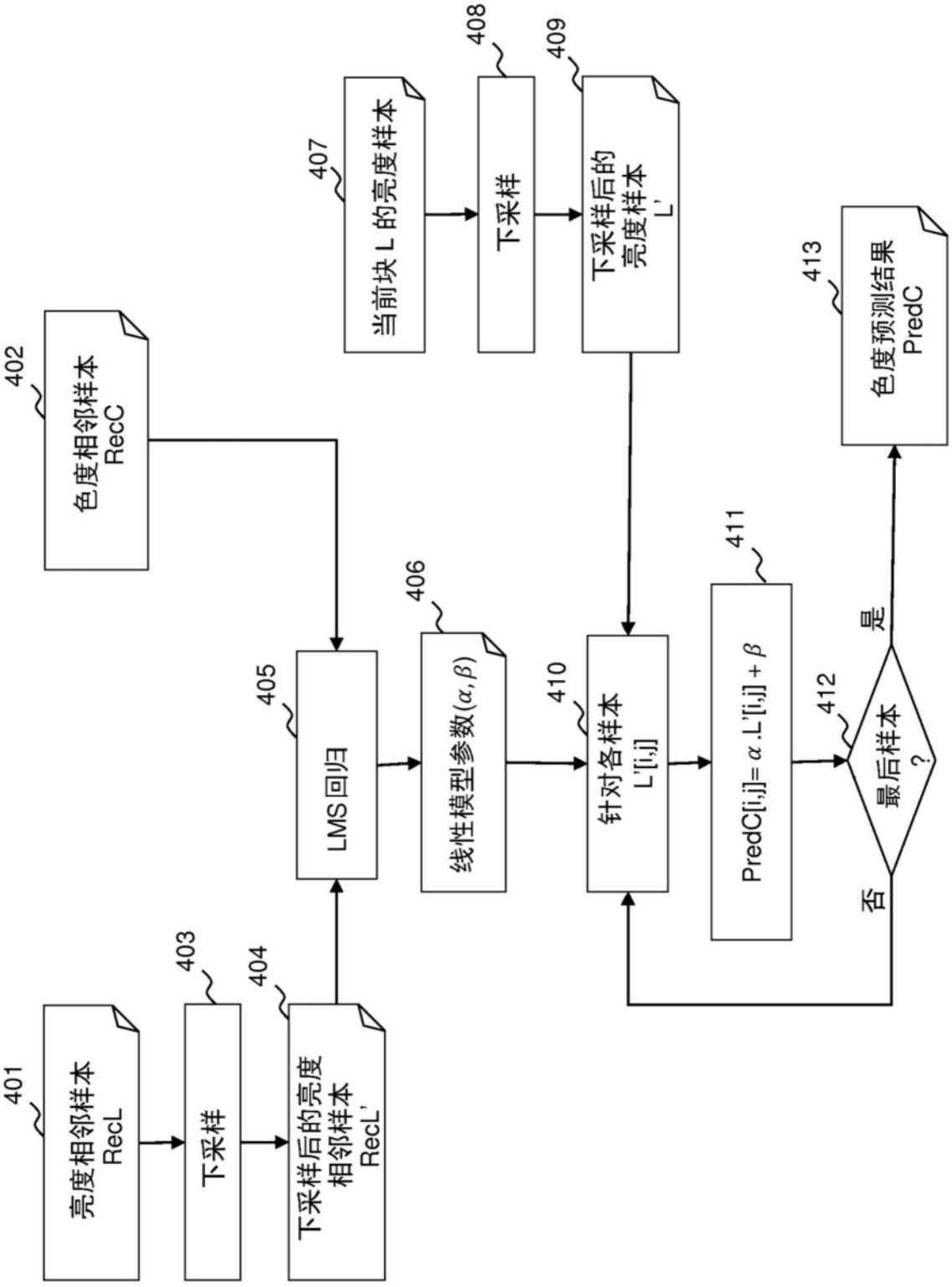

401
亮度相邻样本
RecL
403
下采样
404
下采样后的亮度
相邻样本
RecL'
405
LMS回归
406
线性模型参数(α,β)
402
色度相邻样本
RecC
407
当前块L的亮度样本
408
下采样
409
下采样后的
亮度样本
L'
410
针对各样本
L'[i,j]
411
PredC[i,j]= α .L'[i,j] + β
412
最后样本
?
是
否
413
色度预测结果
PredC

图4

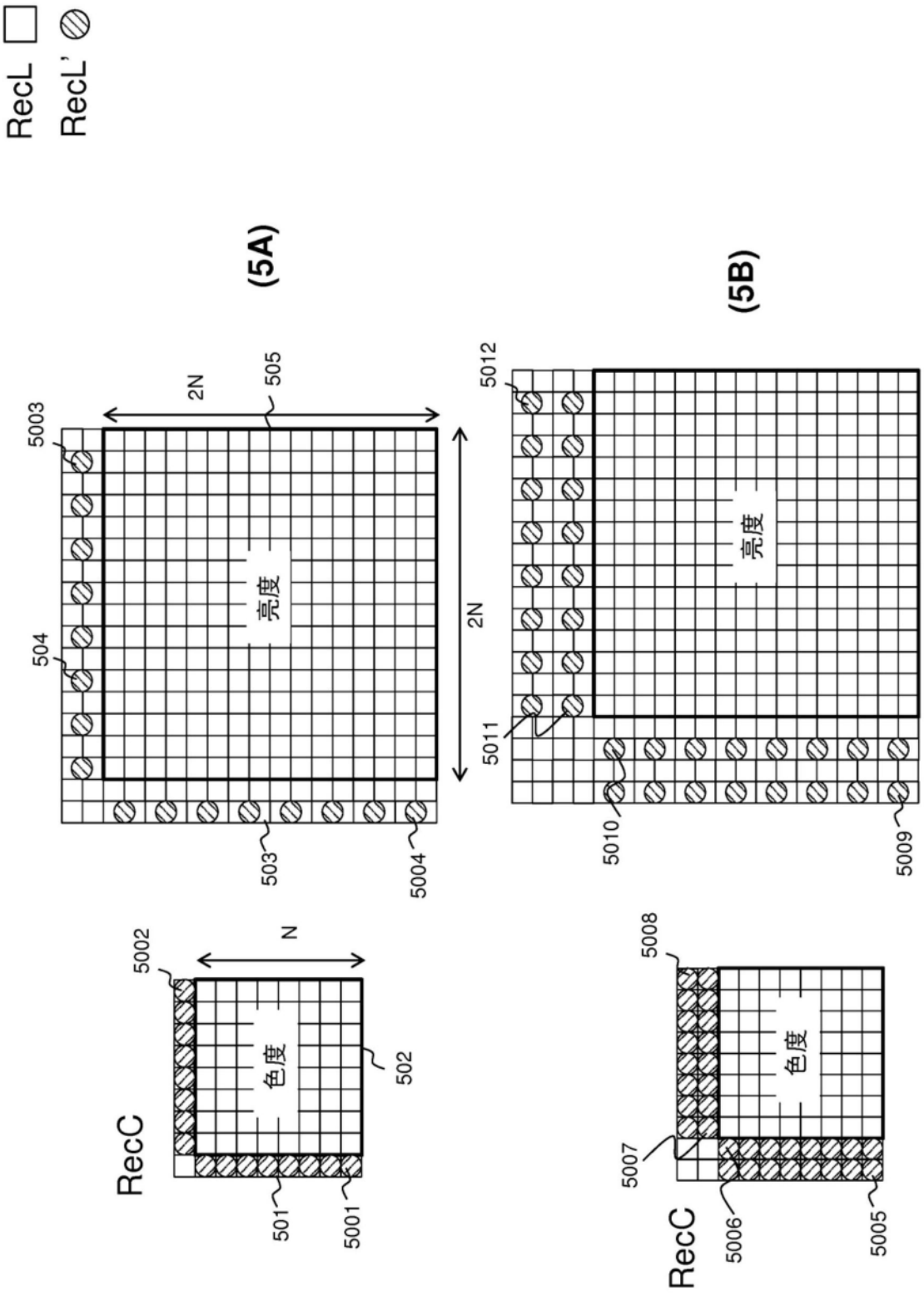

RecL
RecL'
(5A)
(5B)
2N
505
5003
504
亮度
2N
503
5004
5012
5011
5010
5009
N
5002
502
色度
RecC
501
5001
5008
5007
5006
5005

图5

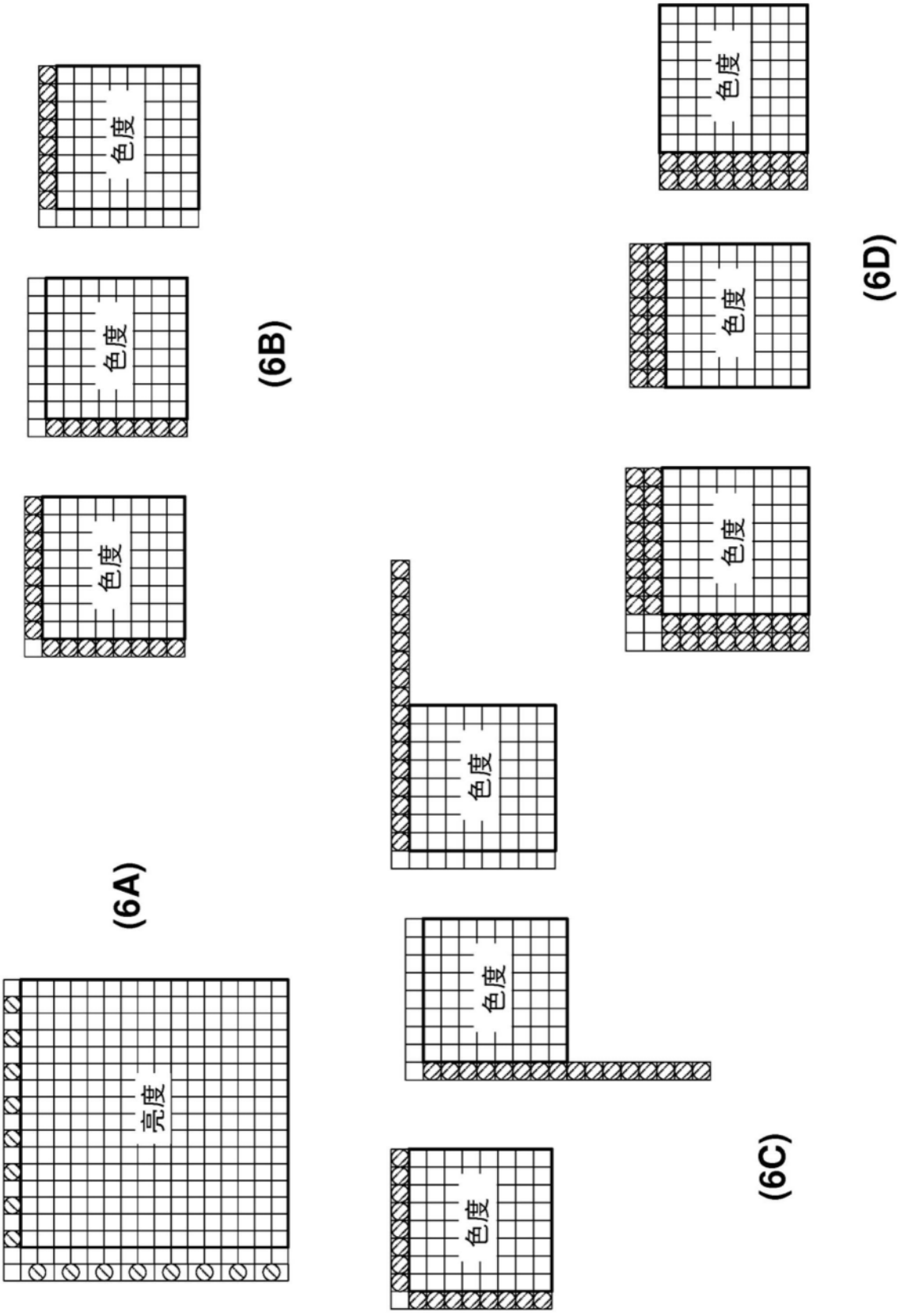


图6

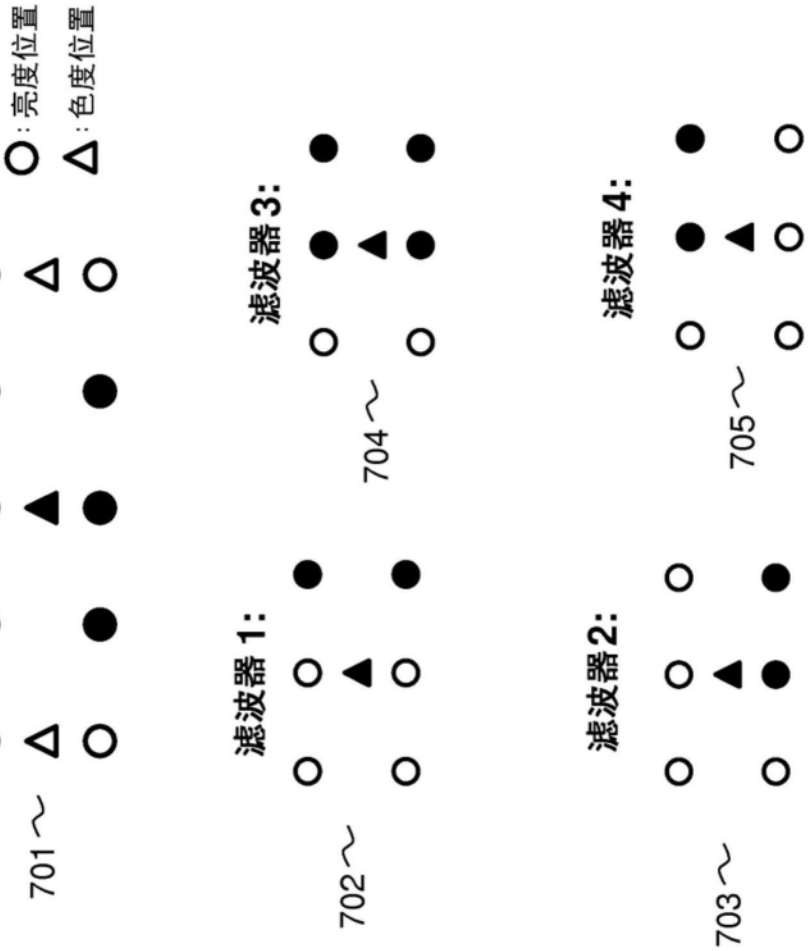


图7

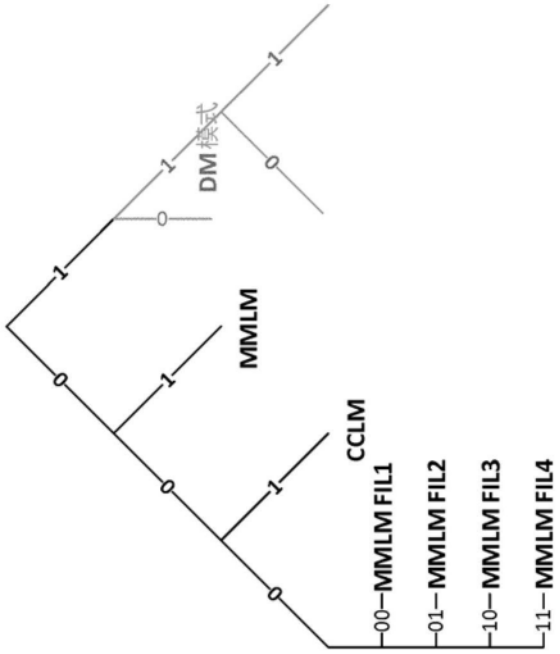

DM模式
MMLM
CCLM
00—MMLM FIL1
01—MMLM FIL2
10—MMLM FIL3
11—MMLM FIL4

图8

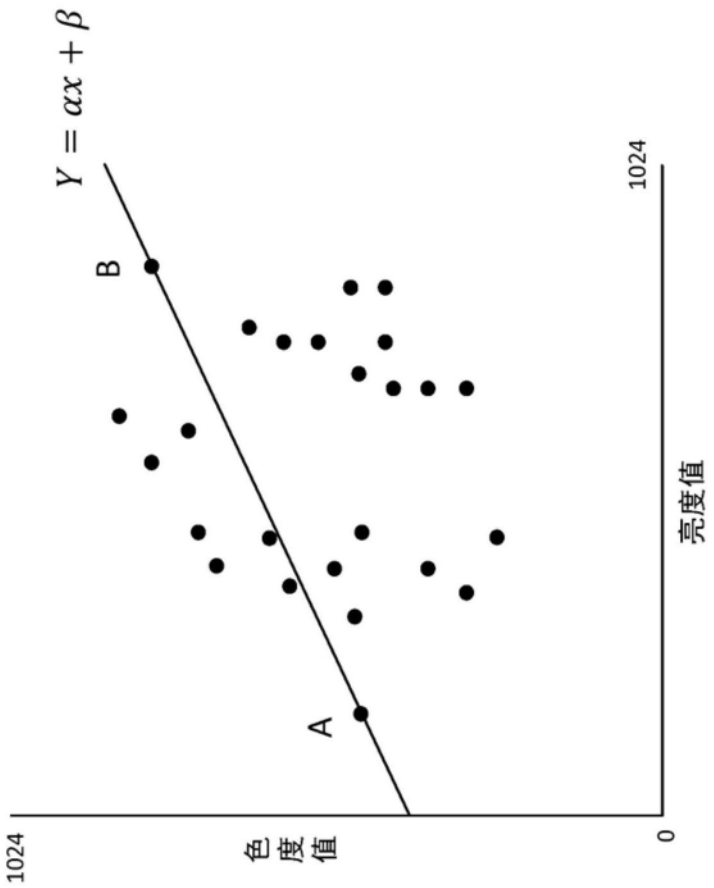

$Y = \alpha x + \beta$
B
A
1024
色度值
0
1024
亮度值

图9

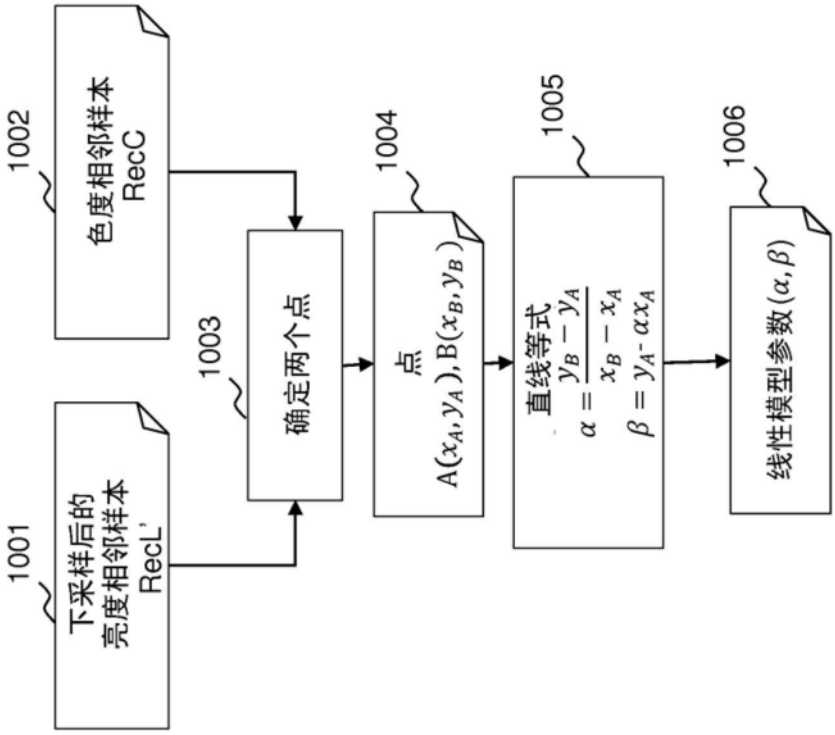

1001
下采样后的亮度相邻样本RecL'
1002
色度相邻样本RecC
1003
确定两个点
1004
点
$A(x_A, y_A), B(x_B, y_B)$
1005
直线等式
$\alpha = \frac{y_B - y_A}{x_B - x_A}$
$\beta = y_A - \alpha x_A$
1006
线性模型参数(α, β)

图10

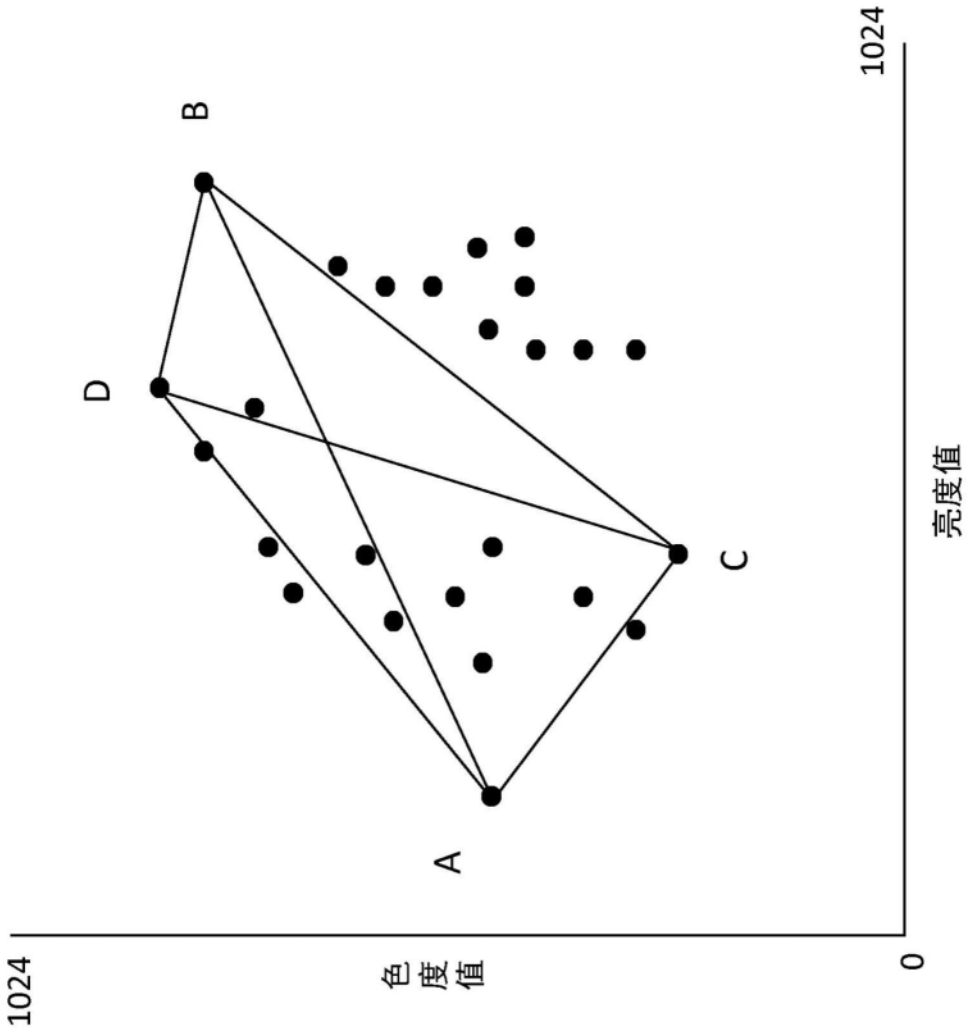


图11

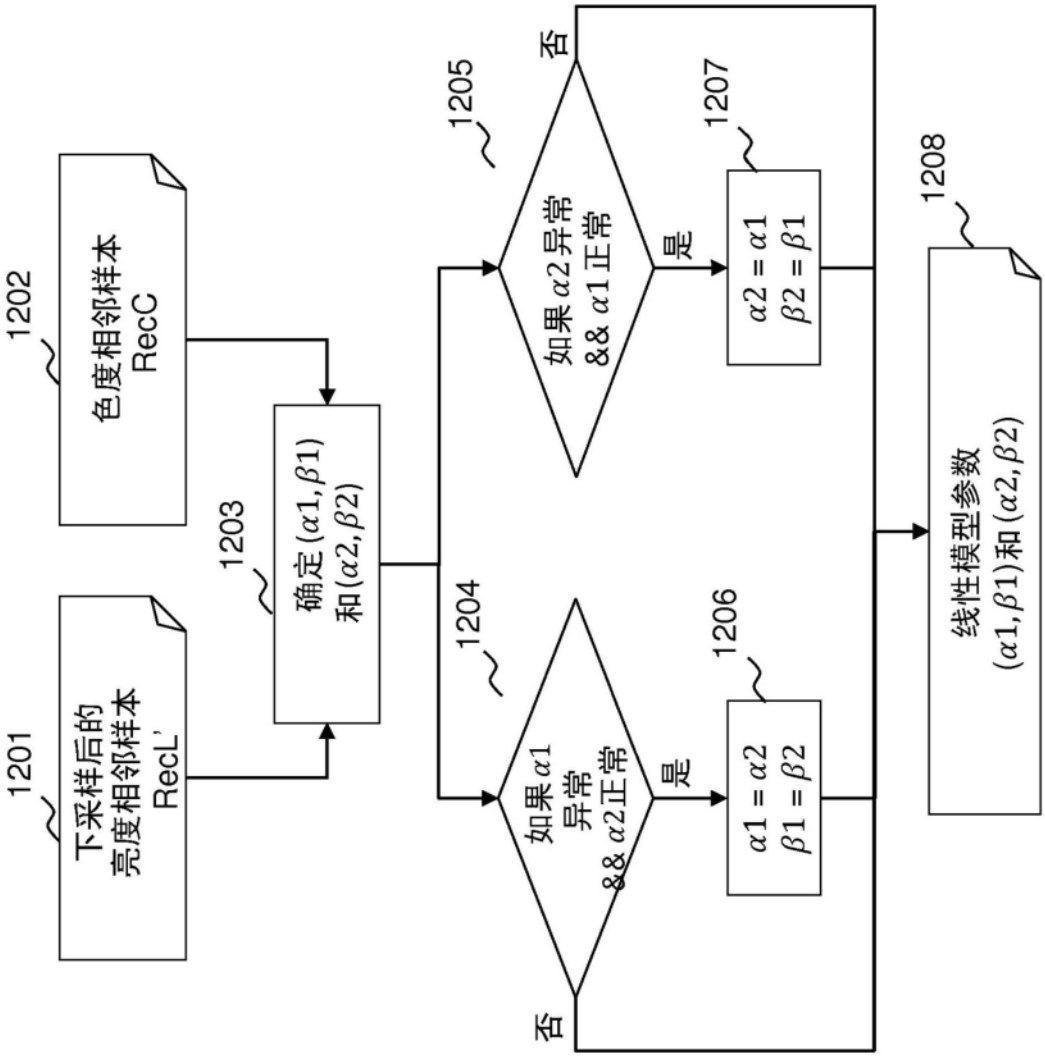

1201
下采样后的
亮度相邻样本
RecL'
1202
色度相邻样本
RecC
1203
确定(α1,β1)
和(α2,β2)
1204
如果α1
异常
&& α2正常
是
否
1205
如果α2异常
&& α1正常
是
否
1206
α1 = α2
β1 = β2
1207
α2 = α1
β2 = β1
1208
线性模型参数
(α1,β1)和(α2,β2)

图12

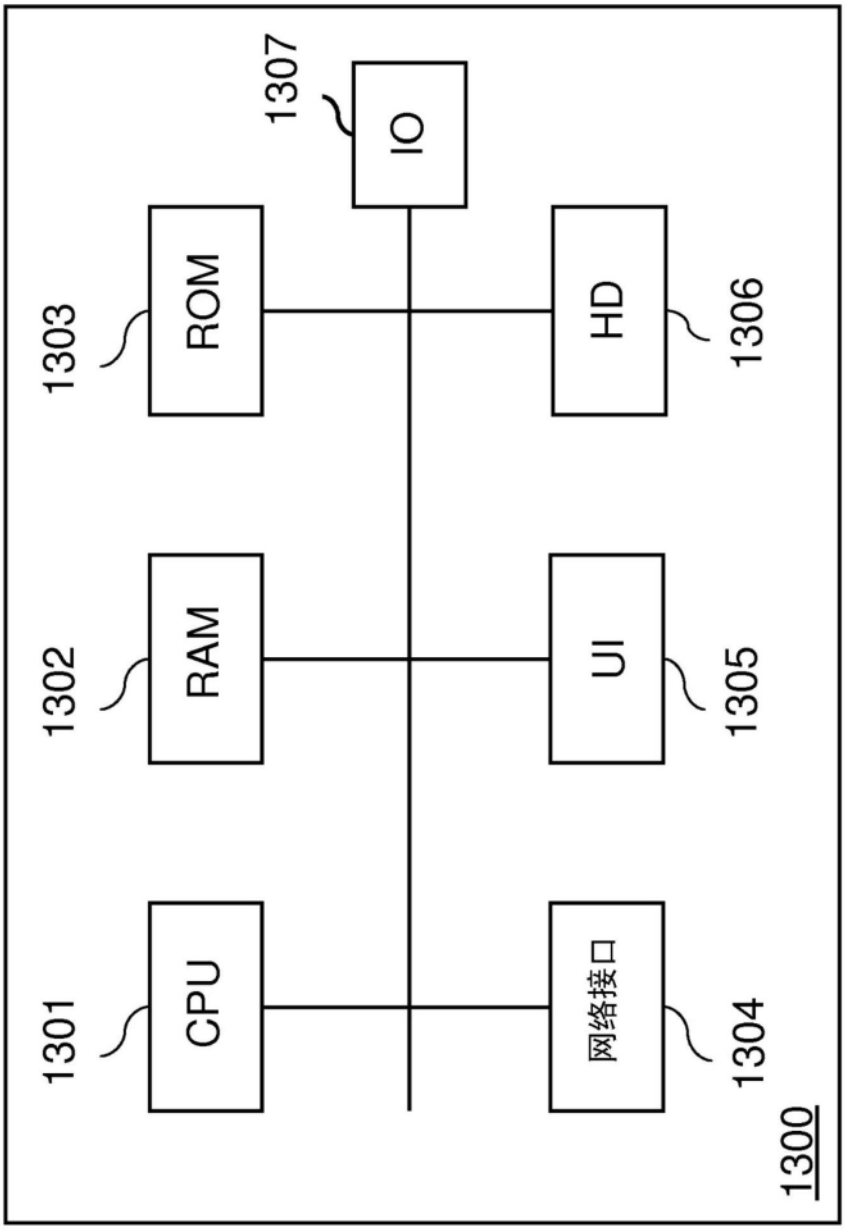

1307
IO
1303
ROM
HD
1306
1302
RAM
UI
1305
1301
CPU
网络接口
1304
1300

图13