



(12) 发明专利申请

(10) 申请公布号 CN 114631312 A

(43) 申请公布日 2022. 06. 14

(21) 申请号 201980101590.8

(22) 申请日 2019.12.23

(30) 优先权数据

62/940,272 2019.11.26 US

(85) PCT国际申请进入国家阶段日

2022.04.22

(86) PCT国际申请的申请数据

PCT/US2019/068279 2019.12.23

(87) PCT国际申请的公布数据

W02021/107965 EN 2021.06.03

(71) 申请人 谷歌有限责任公司

地址 美国加利福尼亚州

(72) 发明人 山·李 克劳迪诺·科埃略

郑仁淑 阿基·库塞拉

(74) 专利代理机构 中原信达知识产权代理有限
责任公司 11219

专利代理师 李佳 周亚荣

(51) Int.Cl.

H04N 19/11 (2014.01)

H04N 19/176 (2014.01)

G06T 9/00 (2006.01)

H04N 19/147 (2014.01)

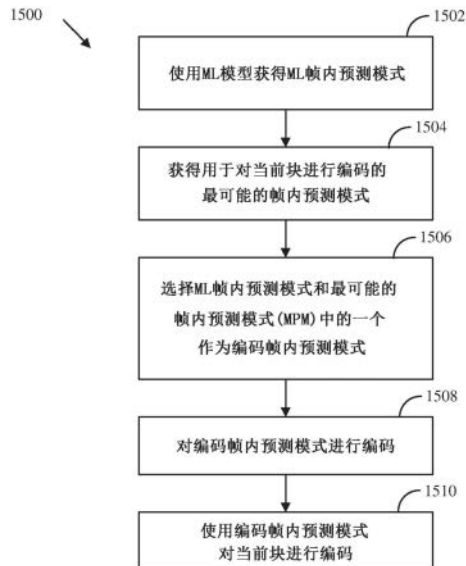
权利要求书3页 说明书26页 附图17页

(54) 发明名称

用于快速视频代码化的超轻模型和决策融合

(57) 摘要

一种使用帧内预测对视频的当前块进行编码的方法,所述方法包括:使用机器学习(ML)模型获得ML帧内预测模式;从用于对所述当前块进行编码的可用帧内预测模式当中获得最可能的帧内预测模式;选择所述ML帧内预测模式或所述最可能的帧内预测模式中的一个作为编码帧内预测模式;将所述编码帧内预测模式编码在压缩比特流中;以及使用所述编码帧内预测模式对所述当前块进行编码。所述选择是基于所述ML帧内预测模式和所述最可能的帧内预测模式的相对可靠性。



1. 一种用于使用帧内预测来对视频的当前块进行编码的方法,所述方法包括:
使用机器学习ML模型获得ML帧内预测模式;
从用于对所述当前块进行编码的可用帧内预测模式当中获得最可能的帧内预测模式;
选择所述ML帧内预测模式或所述最可能的帧内预测模式中的一个作为编码帧内预测模式,其中,所述选择是基于所述ML帧内预测模式和所述最可能的帧内预测模式的相对可靠性;
将所述编码帧内预测模式编码在压缩比特流中;以及
使用所述编码帧内预测模式对所述当前块进行编码。
2. 根据权利要求1所述的方法,其中,在所述ML帧内预测模式或所述最可能的帧内预测模式中的所述一个是最优帧内预测模式的更好预测者的情况下,所述ML帧内预测模式或所述最可能的帧内预测模式中的所述一个比所述ML帧内预测模式或所述最可能的帧内预测模式中的另一个更可靠。
3. 根据权利要求1或2所述的方法,其中,在训练阶段中,使用输入数据来训练所述ML模型以输出所述当前块的所述ML帧内预测模式,并且所述输入数据中的输入单个数据包括输入视频块和用于对所述输入视频块进行编码的对应的地面实况帧内预测模式。
4. 根据权利要求3所述的方法,其中,选择所述ML帧内预测模式或所述最可能的帧内预测模式中的一个作为所述编码帧内预测模式包括:
使用决策矩阵来选择所述编码帧内预测模式,其中:
所述决策矩阵是在所述训练阶段期间使用统计来生成的,所述统计将所述ML模型和编码器的相应性能进行比较,所述编码器选择与地面实况帧内预测模式相比的最可能的帧内预测模式,以及
所述决策矩阵指示是所述ML帧内预测模式还是所述最可能的帧内预测模式要被选择作为所述编码帧内预测模式。
5. 根据权利要求1至4中任一项所述的方法,其中,使用所述ML模型获得所述ML帧内预测模式包括:
使用所述当前块的平均值或所述当前块的方差中的至少一个作为所述ML模型的输入。
6. 根据权利要求1至5中任一项所述的方法,其中,使用所述ML模型获得所述ML帧内预测模式包括:
将内核应用于所述当前块以获得卷积块;
根据所述卷积块来获得标量;以及
使用所述标量作为所述ML模型的输入。
7. 根据权利要求6所述的方法,其中,所述内核是Sobel滤波器,所述Sobel滤波器对应于与所述可用帧内预测模式之一相关联的方向,并且所述标量是应用所述Sobel滤波器所得到的卷积块的绝对值之和。
8. 根据权利要求6所述的方法,其中,所述内核使用一个或多个卷积层而被应用于所述当前块,所述卷积层是在训练阶段期间进行训练的。
9. 根据权利要求1至8中任一项所述的方法,其中,使用所述ML模型获得所述ML帧内预测模式包括:使用量化参数的非线性函数作为所述ML模型的输入。
10. 根据权利要求1至9中任一项所述的方法,其中,使用所述ML模型获得所述ML帧内预

测模式包括：

针对所述可用帧内预测模式中的一些模式计算相应的预测块；
计算所述相应的预测块和所述当前块之间的相应的误差；以及
使用所述相应的误差作为所述ML模型的输入。

11. 根据权利要求1至10中任一项所述的方法，其中，选择所述ML帧内预测模式或所述最可能的帧内预测模式中的一个作为所述编码帧内预测模式包括：

计算与所述ML帧内预测模式相关联的第一成本；
选择所述ML帧内预测模式作为所述编码帧内预测模式；以及
将所述第一成本减小到第二成本。

12. 一种用于使用帧内预测对视频的当前块进行编码的方法，所述方法包括：

获得预先计算的特征，其中，所述预先计算的特征包括以下中的至少两项：

第一特征，所述第一特征是量化参数的非线性函数；

第二特征，所述第二特征是所述当前块和相应的预测块之间的相应的误差，其中，每个预测块对应于可用的帧内预测模式；

所述当前块的平均值和方差；以及

卷积块的绝对值之和，所述卷积块是从所述当前块获得的；

使用机器学习ML模型获得ML帧内预测模式，其中，所述ML模型接收所述预先计算的特征作为输入；

至少使用所述ML帧内预测模式来选择编码帧内预测模式；

将所述编码帧内预测模式编码在压缩比特流中；以及

使用所述编码帧内预测模式对所述当前块进行编码。

13. 根据权利要求12所述的方法，其中，所述预先计算的特征包括所述第一特征、所述第二特征、所述平均值和所述方差、以及所述卷积块的所述绝对值之和。

14. 根据权利要求12或13所述的方法，其中，至少使用所述ML帧内预测模式来选择所述编码帧内预测模式包括：

从用于对所述当前块进行编码的可用帧内预测模式当中获得最可能的帧内预测模式；
以及

使用决策矩阵来选择所述编码帧内预测模式，其中，所述编码帧内预测模式是所述ML帧内预测模式或所述最可能的帧内预测模式MPM之一。

15. 根据权利要求14所述的方法，其中：

所述决策矩阵是在所述训练阶段期间使用统计来生成的，所述统计将所述ML模型和编码器的相应性能进行比较，所述编码器选择与地面实况帧内预测模式相比的MPM，以及

所述决策矩阵指示是所述ML帧内预测模式还是所述最可能的帧内预测模式要被选择作为所述编码帧内预测模式。

16. 根据权利要求12至15中任一项所述的方法，其中，在训练阶段中，使用输入数据来训练所述ML模型以输出所述当前块的所述ML帧内预测模式，其中，所述输入数据中的输入单个数据包括输入视频块和用于对所述输入视频块进行编码的对应的最佳帧内预测模式。

17. 根据权利要求12至16中任一项所述的方法，其中，计算从所述当前块获得的所述卷积图像的所述绝对值之和包括：将Sobel滤波器应用于所述当前块以获得所述卷积块。

18. 根据权利要求12至16中任一项所述的方法,其中,计算从所述当前块获得的所述卷积图像的所述绝对值之和包括:使用卷积操作来获得所述卷积块。

19. 一种用于使用帧内预测来对视频的当前块进行编码的装置,其中,所述装置:

使用机器学习ML模型获得ML帧内预测模式;

从用于对所述当前块进行编码的可用帧内预测模式当中获得最可能的帧内预测模式;

选择所述ML帧内预测模式之一或所述最可能的帧内预测模式之一作为编码帧内预测模式;

将所述编码帧内预测模式编码在压缩比特流中;以及

使用所述编码帧内预测模式对所述当前块进行编码。

20. 根据权利要求19所述的装置,其中:

所述装置使用决策矩阵选择所述ML帧内预测模式之一或所述最可能的帧内预测模式之一作为所述编码帧内预测模式,

所述决策矩阵是在所述ML模型的训练阶段期间使用统计来生成的,所述统计将所述ML模型和编码器的相应性能进行比较,所述编码器选择与地面实况帧内预测模式相比的最可能的帧内预测模式,以及

所述决策矩阵指示是ML帧内预测模式还是最可能的帧内预测模式要被选择作为所述编码帧内预测模式。

21. 一种用于使用帧内预测对视频的当前块进行编码的装置,其中,所述装置:

获得预先计算的特征,其中,所述预先计算的特征包括以下中的至少两项:

第一特征,所述第一特征是量化参数的非线性函数;

第二特征,所述第二特征是所述当前块和对应的预测块之间的相应的误差,其中,每个预测块对应于可用的帧内预测模式;

所述当前块的平均值和方差;以及

卷积块的绝对值之和,所述卷积块是从所述当前块获得的;

使用机器学习ML模型获得ML帧内预测模式,其中,所述ML模型接收所述预先计算的特征作为输入;

至少使用所述ML帧内预测模式来选择编码帧内预测模式;

将所述编码帧内预测模式编码在压缩比特流中;以及

使用所述编码帧内预测模式对所述当前块进行编码。

用于快速视频代码化的超轻模型和决策融合

[0001] 相关申请的交叉引用

[0002] 本申请要求于2019年11月26日提交的美国临时专利申请No.62/940,272的优先权和权益,该申请的全部公开内容通过引用并入本文。

背景技术

[0003] 数字视频流可以使用一系列帧或静止图像来表示视频。数字视频可以用于各种应用,包括例如视频会议、高清视频娱乐、视频广告或用户生成的视频的共享。数字视频流可以包含大量数据并且消耗计算设备的大量计算或通信资源来处理、传输或存储视频数据。已经提出了各种各样的方法来减少视频流中的数据量,包括压缩和其他编码技术。

[0004] 多年来,视频编码器的代码化效率已经得到了改进。代码化效率可以指以最低的可能比特率(rate)对视频进行编码,同时最小化失真(即,同时保持一定水平的视频质量)。然而,改进的代码化效率导致了计算复杂性的增加。即,为了实现改进的代码化效率,编码器需要更多的计算时间。因此,需要以更少的计算时间(即,降低的计算复杂性)来获得改进的代码化效率。

发明内容

[0005] 所公开的实施方式的一个方面是使用帧内预测对当前视频块进行编码的方法。该方法包括:使用机器学习(ML)模型获得ML帧内预测模式;从用于对当前块进行编码的可用帧内预测模式当中获得最可能的帧内预测模式;选择ML帧内预测模式或最可能的帧内预测模式中的一个作为编码帧内预测模式;将编码帧内预测模式编码在压缩比特流中;以及使用编码帧内预测模式对当前块进行编码。选择是基于ML帧内预测模式和最可能的帧内预测模式的相对可靠性。

[0006] 另一个方面是使用帧内预测对当前视频块进行编码的方法。该方法包括:获得预先计算的特征;使用机器学习(ML)模型,获得ML帧内预测模式,其中,ML模型接收预先计算的特征作为输入;至少使用ML帧内预测模式选择编码帧内预测模式;将编码帧内预测模式编码在压缩比特流中;以及使用编码帧内预测模式对当前块进行编码。预先计算的特征可以包括以下至少两项:第一特征,该第一特征是量化参数的非线性函数;第二特征,该第二特征是当前块和相应的预测块之间的相应误差,其中,每个预测块都对应于可用帧内预测模式;当前块的平均值和方差;以及卷积块的绝对值之和,卷积块是从当前块获得的。

[0007] 另一个方面是用于使用帧内预测对当前视频块进行编码的装置。该装置使用机器学习(ML)模型获得ML帧内预测模式;从用于对当前块进行编码的可用帧内预测模式当中获得最可能的帧内预测模式;选择ML帧内预测模式或最可能的帧内预测模式中的一个作为编码帧内预测模式;将编码帧内预测模式编码在压缩比特流中;以及使用编码帧内预测模式对当前块进行编码。

[0008] 另一个方面是用于使用帧内预测对当前视频块进行编码的装置。该装置获得预先计算的特征;使用机器学习(ML)模型,获得ML帧内预测模式,其中,ML模型接收预先计算的

特征作为输入;至少使用ML帧内预测模式选择编码帧内预测模式;将编码帧内预测模式编码在压缩比特流中;以及使用编码帧内预测模式对当前块进行编码。预先计算的特征可以包括以下至少两项:第一特征,该第一特征是量化参数的非线性函数;第二特征,该第二特征是当前块和对应的预测块之间的相应误差,其中,每个预测块都对应于可用帧内预测模式;当前块的平均值和方差;以及卷积块的绝对值之和,卷积块是从当前块获得的。

[0009] 本公开的这些和其他方面在下面对实施例、所附权利要求书和附图的详细描述中进行了公开。

附图说明

[0010] 本文中的描述参考附图,其中,相同的附图标记在几个视图都指类似的部件。

[0011] 图1是视频编码和解码系统的示意图。

[0012] 图2是可以实施发送站或接收站的计算设备的示例的框图。

[0013] 图3是要被编码并且随后被解码的视频流的示意图。

[0014] 图4是根据本公开的实施方式的编码器的框图。

[0015] 图5是根据本公开的实施方式的解码器的框图。

[0016] 图6是根据本公开的实施方式的帧的部分的表示的框图。

[0017] 图7是根据本公开的实施方式的帧内预测模式的示例的示意图。

[0018] 图8是用于搜索对块进行代码化的最佳模式的过程的流程图。

[0019] 图9是估计使用预测模式对图像块进行代码化的速率和失真成本的示例的框图。

[0020] 图10A至图10B是用于模式决策的卷积神经网络(CNN)的示例的框图。

[0021] 图11是根据本公开的实施方式的、用于通过编码器使用帧内预测对图像块进行编码的过程的流程图。

[0022] 图12A是根据本公开的实施方式的全稠密网络结构的示例的示意图。

[0023] 图12B是根据本公开的实施方式的网络结构的示例的示意图。

[0024] 图13是根据本公开的实施方式的获得卷积块的绝对值之和的示例。

[0025] 图14图示了根据本公开的实施方式的决策矩阵的示例。

[0026] 图15是根据本公开的实施方式的、用于使用帧内预测对当前块进行编码的过程的流程图。

[0027] 图16是根据本公开的实施方式的、用于使用帧内预测对当前块进行编码的过程的流程图。

[0028] 图17是率失真性能比较的示例1700。

具体实施方式

[0029] 现代视频编解码器(例如,H.264(也被称为MPEG-4AVC)、VP9、H.265(也被称为HEVC)、AVS2和AV1)定义和使用大量工具和配置,这些工具和配置被用来改进代码化效率。代码化效率通常是按照速率和失真二者来测量的。速率是指编码(诸如对块、帧等进行编码)所需的比特数。失真测量在例如源视频块与源视频块的重构版本之间的质量损失。通过执行率失真优化(RDO)过程,视频编解码器根据对视频进行编码所需的速率来优化失真量。

[0030] 为了确定要使用的工具和配置(例如参数)的最优组合,视频编码器可以使用模式

决策过程。模式决策过程可以检查(例如,测试、评估等)至少一些有效的工具组合。在示例中,检查所有可能的组合。检查所有可能的组合或其子集在这里被称为蛮力(brute-force)方法。

[0031] 针对模式决策的蛮力方法可以计算所有可能的组合(或其子集)的率失真成本,然后为每个基本处理块选择具有最佳成本的组合。不同的编解码器使用不同的术语来指代基本处理块。基本处理块可以被称为超级块(即,在AV1中)、宏块(即,在HEVC中),等等。蛮力方法的计算量可能非常大,尤其是在所有组合都经过测试的情况下。随着可能的工具和参数的数量的增加,组合的数量也会增加,这又增加了确定最佳模式所需的时间。例如,AV1编解码器在VP9编解码器上包括大约160个附加工具,从而大大增加了最佳模式的搜索时间。

[0032] 假设第一参数组合产生第一速率(例如,速率=100)和第一失真(例如,失真=90),并且第二参数组合产生第二速率(例如,速率=120)和第二失真(例如,失真=80)。需要一个流程(例如,技术等)来评估第一组合和第二组合中哪一种是更好的参数组合。为了评估一个组合是否比另一个组合好,可以为每个经过检查的组合计算度量,并且将对应的度量进行比较。在一示例中,度量可以将速率和失真结合,以产生一个单一的标量值,如下所述。在本公开中,率失真成本用作这种标量值。

[0033] 模式决策过程的示例是帧内预测模式决策过程,该过程确定用于对代码化块进行代码化的最佳帧内预测模式。

[0034] 为了对基本处理块(该块可以有 64×64 、 128×128 的大小或一些其他大小)进行编码,编码器可以尝试确定(例如,提取、选择等)基本处理块的层次表示(即,分区),一直到 4×4 块,而带有不分割块的选项。(基本处理块或任何正方形子块的)可能的分区可以包括不对块进行分区、使用垂直或水平分割、使用正方形或矩形分割或它们的组合。

[0035] 对视频流或其部分(诸如帧或块)进行编码可以包括在视频流中使用时间和空间相似性来改进代码化效率。例如,视频流的当前块可以基于识别先前代码化的像素值之间或先前代码化的像素值的组合与当前块中的像素值之间的差(残差)进行编码。

[0036] 使用时间相似性的编码可以被称为帧间预测。帧间预测尝试使用来自一个或多个暂时邻近的帧(即,参考帧)的一个或多个可能移位的块来预测块的像素值。

[0037] 使用空间相似性的编码可以被称为帧内预测。帧内预测尝试使用块外围的像素来预测视频帧的块的像素值。块外围的像素可以包括:顶部像素,这些顶部像素是块上方的像素;左像素,这些左像素是块左边的像素;和/或一个或多个左上方像素。即,外围像素是与块在同一帧中但是在块外的像素。如何使用边缘(即,外围)像素来生成预测的机制被称为帧内预测模式。

[0038] 在HEVC编码器中,例如,对于大于或等于 4×4 的亮度块,35个帧内预测模式是可能的。VP9编码器使用10个帧内预测模式。AV1编码器可以说是包括58个帧内预测模式(两个无方向模式加上八个主方向角,其中,每个主角都有三个正的和负的偏移差量(delta)角)。每个帧内预测模式都决定如何确定对应的预测块。在这种情况下,模式决策过程可以为每个帧内预测模式确定对应的预测块,并且选择与最小率失真成本对应的帧内预测模式。换句话说,模式决策过程选择提供最佳率失真性能的帧内预测模式。帧内预测模式的示例是关于图7提供的。

[0039] 如上所述,最佳帧内预测模式(或简单地说,最佳模式)可以从许多可能的组合中

选择。

[0040] 因此,技术,诸如机器学习,可以被用来减少确定最佳帧内预测模式所需的时间。机器学习可以很好地适合于解决视频代码化中的计算复杂性问题。通常由编码器执行的蛮力运行中模式决策过程可以用训练机器学习(ML)模型取代,该模型可以推断用于对当前块进行编码的帧内预测模式决策。可以期望训练良好的ML模型在代码化效率上与蛮力方法非常相当,但是计算成本显著降低,或具有常规或面向数据流的计算成本。

[0041] 例如,使用蛮力方法进行模式决策(例如,帧内预测模式决策)可以生成大量的训练数据。即,训练数据可以由执行标准编码技术(诸如,关于图4和图8至图9描述的技术)的编码器获得。

[0042] 在机器学习的学习阶段,可以使用训练数据来获取(例如,学习、推断等)机器学习(ML)模型,该模型是(例如,定义、构成等)从输入数据到构成模式决策的输出(诸如,要被用于对当前块进行编码的帧内预测模式)的映射。因此,ML模型可以被用来取代蛮力、计算量大的编码过程(诸如,关于图4和图8至图9所描述的过程),从而降低模式决策的计算复杂性。

[0043] ML模型的预测能力(即,准确性)与用于训练机器学习模型的输入和被呈现给机器学习模型以预测结果的输入(例如,最佳模式)一样。因此,当机器学习用于视频编码时,关键是使用正确的输入集和正确的(例如,适当的、最优的,等等)形式的这种输入。一旦ML模型被训练,模型便计算输出作为其输入的确定性函数。因此,使ML模型拥有正确的架构可能是至关重要的。

[0044] 在本文中描述了一种有效率的为视频代码化设计的ML模型架构。在一示例中,机器学习模型可以是神经网络模型。在一示例中,神经网络模型可以是超轻的(即,包含数量相对较少的参数和/或操作),并且只包括稠密的操作(即,完全连接的层)。在另一个示例中,神经网络可以包括卷积层,但是它仍然是相对超轻的。

[0045] 众所周知的信息论的通用逼近定理表明,前馈神经网络可以用于逼近 n 维真正坐标空间 R^n 的紧致子集上的任何连续函数。注意,现有神经网络的固有线性性质意味着,如果神经网络的任务是(即,被训练以)逼近(例如,映射、解决、推断)非线性函数之外的线性函数(例如,映射),则可以实现更小的网络或更短的学习时间。还注意,视频块到模式决策的映射可以被表征为连续函数。

[0046] 通用逼近定理并没有表征学习阶段的可行性或时间和空间复杂性。即,虽然神经网络在理论上能够逼近非线性函数,但是神经网络学习使用线性函数逼近非线性函数可能需要不合理地大的(例如,在节点和/或层的数量方面)网络和/或不合理地长的训练时间。实际上,所需的不合理的大小和时间可能使学习、推断或两者都不可行。

[0047] 如上所述,可以期望训练良好的神经网络模型在代码化效率上与蛮力方法非常相当。然而,对于许多资源受限的编码平台(例如,移动设备,诸如手机、在线视频流等),编解码硬件区域大小和编码速度可能引起高度关注。

[0048] 因此,与大型模型相比,小型模型(即,具有小型参数集和较少的推断操作的模型)可能是首选。然而,小型模型可能不具有与深度(或大型)模型相同的准确性。

[0049] 为了提高或补充小型ML模型的潜在的低推断准确性,根据本公开的实施方式可以将ML模型与启发式结合(即,融合),该启发式可以在视频编解码器中使用在最佳模式选择

中(例如,帧内预测模式选择),以便保持编码效率并且显著降低了计算成本。使用正确的特征(作为ML模型的输入),并且使用启发式(例如,用于选择帧内预测模式的编解码器启发式)选择ML模型输出的正确融合策略,对于实现紧凑的深度学习/机器学习模型至关重要,该模型可以尝试在编码过程中分析帧内预测模式。

[0050] 小的ML模型大小可以使在功率/容量受限的移动平台上进行推断成为可能。根据本公开的实施方式的小的ML模型大小可以通过以下项获得:1) 预先计算方向特征集作为输入并且包括来自编解码器的现有特征,诸如块的平均值和方差、QP和基于SAD的模式成本,作为模型输入的部分;以及2) 将ML预测与基于SAD的模式决策合并以优化模式准确性。

[0051] 用于快速视频代码化的超轻模型和决策融合在本文中首先参考其中可以并入该教导的系统进行描述。

[0052] 注意,在这里省略了机器学习、神经网络的细节和/或本领域技术人员所知的细节。例如,本领域技术人员认识到,神经网络中节点(即,神经元)之间的连接的权重的值是在训练阶段确定的。因此,在这里不会对这些进行详细讨论。

[0053] 图1是视频编码和解码系统100的示意图。例如,发送站102可以是具有硬件的内部配置的计算机,诸如关于图2所描述的计算机。然而,发送站102的其他合适的实施方式是可能的。例如,发送站102的处理可以分布在多个设备当中。

[0054] 网络104可以将发送站102和接收站106连接以对视频流进行编码和解码。具体地,视频流可以在发送站102中进行编码,并且编码视频流可以在接收站106中进行解码。例如,网络104可以是互联网。网络104还可以是局域网(LAN)、广域网(WAN)、虚拟专用网络(VPN)、蜂窝电话网络或在该示例中将视频流从发送站102传输给接收站106的任何其他方式。

[0055] 在一个示例中,接收站106可以是具有硬件的内部配置的计算机,诸如,关于图2所描述的计算机。然而,接收站106的其他合适的实施方式是可能的。例如,接收站106的处理可以分布在多个设备之间。

[0056] 视频编码和解码系统100的其他实施方式是可能的。例如,实施方式可以省略网络104。在另一种实施方式中,视频流可以被编码,然后被存储以在以后的时间传输给接收站106或具有存储器的任何其他设备。在一种实施方式中,接收站106接收(例如,通过网络104、计算机总线和/或一些通信路径)编码视频流,并且存储视频流以便稍后进行解码。在示例实施方式中,实时传输协议(RTP)用于通过网络104传输编码视频。在另一种实施方式中,可以使用除RTP之外的传输协议(例如,基于HTTP的视频流协议)。

[0057] 当在视频会议系统中使用时,例如,发送站102和/或接收站106可以包括下述对视频流进行编码和解码的能力。例如,接收站106可以是视频会议参与者,该视频会议参与者接收来自视频会议服务器(例如,发送站102)的编码视频比特流以解码和查看,并且进一步对其自己的视频比特流进行编码并且将其传输给视频会议服务器,以供其他参与者解码和查看。

[0058] 图2是可以实施发送站或接收站的计算设备200的示例的框图。例如,计算设备200可以实施图1的发送站102和接收站106中的一个或两个。计算设备200可以是包括多个计算设备的计算系统的形式或单个计算设备的形式,例如,移动电话、平板计算机、膝上型计算机、笔记型计算机、桌上型计算机等。

[0059] 计算设备200中的CPU 202可以是中央处理单元。替选地,CPU202可以是现存或以

后开发的任何其他类型的设备或多个设备,这些设备能够操纵或处理信息。尽管所公开的实施方式可以用如图所示的单个处理器(例如,CPU 202)来实践,但是在速度和效率方面的优势可以通过使用一个以上的处理器来实现。

[0060] 在实施方式中,计算设备200中的存储器204可以是只读存储器(ROM)设备或随机存取存储器(RAM)设备。任何其他合适类型的存储设备都可以用作存储器204。存储器204可以包括由CPU 202使用总线212访问的代码和数据206。存储器204可以进一步包括操作系统208和应用程序210,应用程序210包括允许CPU 202执行本文中所描述的方法的至少一个程序。例如,应用程序210可以包括应用1至N,这些应用进一步包括执行本文中所描述的方法的视频代码化应用。计算设备200还可以包括辅助储存装置214,例如,该辅助储存装置214可以是与移动的计算设备200一起使用的内存卡。因为视频通信会话可以包含大量信息,所以它们可以全部或部分存储在辅助储存装置214中,并且根据需要被加载到存储器204中以进行处理。

[0061] 计算设备200还可以包括一个或多个输出设备,诸如显示器218。在一个示例中,显示器218可以是触敏显示器,该触敏显示器将显示器与可操作以感测触摸输入的触敏元件结合。显示器218可以通过总线212耦合到CPU 202。除了显示器218之外或作为显示器218的替选,可以提供允许用户编程或以其他方式使用计算设备200的其他输出设备。当输出设备是显示器或包括显示器时,显示器可以以各种方式实施,包括作为液晶显示器(LCD)、阴极射线管(CRT)显示器或发光二极管(LED)显示器,诸如有机LED(OLED)显示器。

[0062] 计算设备200还可以包括图像感测设备220(例如,摄像头)或能够感测图像(诸如,操作计算设备200的用户的图像)的现在存在或以后开发的任何其他图像感测设备220或与其进行通信。图像感测设备220可以被定位,以便其指向操作计算设备200的用户。在示例中,图像感测设备220的位置和光轴可以被配置,以便视场包括直接与显示器218相邻并且从其可见显示器218的区域。

[0063] 计算设备200还可以包括声音感测设备222,例如,麦克风或现存或以后开发的可以感测计算设备200附近的声音的任何其他声音感测设备,或与其通信。声音感测设备222可以被定位,以便其指向操作计算设备200的用户,并且可以配置为接收声音,例如,用户操作计算设备200时发出的语音或其他话语。

[0064] 尽管图2将计算设备200的CPU 202和存储器204描绘成被集成为单个单元,但是也可以利用其他配置。CPU 202的操作可以分布在多个机器(每个机器都具有一个或多个处理器)之间,这些机器可以直接地或跨局域网或其他网络被耦合。存储器204可以分布在多个机器之间,诸如基于网络的存储器或执行计算设备200的操作的多个机器中的存储器。尽管在这里被描绘为单个总线,但是计算设备200的总线212可以由多个总线组成。进一步地,二级储存装置214可以直接耦合到计算设备200的其他组件或可以通过网络访问,并且可以包括单个集成单元(诸如内存卡)或多个单元(诸如多个内存卡)。因此,计算设备200可以以各种配置实施。

[0065] 图3是要被编码并且随后被解码的视频流300的示例的示意图。视频流300包括视频序列302。在下一级别,视频序列302包括多个相邻帧304。尽管三个帧被描绘为相邻帧304,但是视频序列302可以包括任意数量的相邻帧304。相邻帧304然后可以被进一步细分为单独的帧,例如,帧306。在下一级别,帧306可以被分成一系列的段308或平面。例如,段

308可以是允许进行并行处理的帧的子集。段308还可以是可以将视频数据分成单独的颜色帧的子集。例如,颜色视频数据的帧306可以包括亮度平面和两个色度平面。段308可以以不同的分辨率进行采样。

[0066] 无论帧306是否被分成段308,帧306都可以被进一步细分成块310,该块310可以包含与例如帧306中的 16×16 像素对应的数据。块310也可以设置为包括来自像素数据的一个或多个段308的数据。块310也可以具有任何其他合适的大小,诸如 4×4 像素、 8×8 像素、 16×8 像素、 8×16 像素、 16×16 像素或更大。

[0067] 图4是根据本公开的实施方式的编码器400的框图。编码器400可以如上所述在发送站102中实施,诸如通过提供存储器(例如存储器204)中所存储的计算机软件程序。计算机软件程序可以包括机器指令,该机器指令在由处理器(诸如CPU 202)执行时使发送站102以本文中所描述的方式对视频数据进行编码。编码器400还可以实施为专用硬件,该专用硬件被包括在例如发送站102中。编码器400具有使用视频流300作为输入来执行正向路径(用实心连接线表示)中的各种功能以产生编码或压缩比特流420的以下阶段:帧内/帧间预测阶段402、变换阶段404、量化阶段406和熵编码阶段408。编码器400还可以包括用于重构帧以对未来的块进行编码的重构路径(用虚线表示)。在图4中,编码器400具有在重构路径中执行各种功能的以下阶段:解量化阶段410、逆变换阶段412、重构阶段414和环路滤波阶段416。编码器400的其他结构变型可以用于对视频流300进行编码。

[0068] 当视频流300被呈现以进行编码时,帧306可以以块为单位被处理。在帧内/帧间预测阶段402,块可以使用帧内预测(也称为帧内预测)或帧间预测(也称为帧间预测)或两者的组合进行编码。在任何情况下,预测块都可以形成。在帧内预测的情况下,预测块的全部或部分可以由当前帧中先前已被编码和重构的样本形成。在帧间预测的情况下,预测块的全部或部分可以由使用运动向量而被确定的一个或多个先前构造的参考帧中的样本形成。

[0069] 接下来,仍然参照图4,预测块在帧内/帧间阶段402可以从当前块减去以产生残差块(也称为残差)。变换阶段404使用基于块的变换在例如频域中将残差转变成变换系数。例如,这种基于块的变换(即,变换类型)包括离散余弦变换(DCT)和非对称离散正弦变换(ADST)。其他基于块的变换是可能的。进一步地,不同变换的组合都可以应用于单个残差。在变换应用的一个示例中,DCT将残差块转变成频域,其中,变换系数值是基于空间频率。最低频率(DC)系数在矩阵的左上方,并且最高频率系数在矩阵的右下方。值得注意的是,预测块的大小可能与变换块的大小不同,因此,由此产生的残差块可能与变换块的大小不同。例如,预测块可以被分成更小的块,单独的变换应用于这些更小的块。

[0070] 量化阶段406使用量化器值或量化水平将变换系数转变成离散量子值,这些离散量子值被称为量化变换系数。例如,变换系数可以除以量化器值并且被截断。然后,量化的变换系数通过熵编码阶段408进行熵编码。熵代码化可以使用任何数量的技术(包括令牌和二叉树)来执行。然后,将熵编码系数连同用于对块进行解码的其他信息(例如,可以包括所使用的预测类型、变换类型、运动向量和量化器值)输出给压缩比特流420。用于对块进行解码的信息可以被熵代码化成压缩比特流420中的块、帧、切片和/或节头(section header)。压缩比特流420还可以被称为编码视频流或编码视频比特流;这些术语在本文中将互换使用。

[0071] 图4中的重构路径(用虚线表示)可以用于确保编码器400和解码器500(如下所述)

使用相同的参考帧和块对压缩比特流420进行解码。重构路径执行与在解码过程中发生并且在下面更详细地讨论的功能类似的功能,包括在解量化阶段410对量化变换系数进行解量化和在逆变换阶段412对解量化变换系数进行逆变换以产生导数残差块(也被称为导数残差)。在重构阶段414,在帧内/帧间预测阶段402所预测的预测块可以添加到导数残差以创建重构块。环路滤波阶段416可以应用于重构块以减少失真,诸如构块(blocking)效应。

[0072] 编码器400的其他变型可以用于对压缩比特流420进行编码。例如,基于非变换的编码器400可以直接量化残差信号,而无需某些块或帧的变换阶段404。在另一种实施方式中,编码器400可以将量化阶段406和解量化阶段410结合成单个阶段。

[0073] 图5是根据本公开的实施方式的解码器500的框图。解码器500可以在接收站106中实施,例如,通过提供存储器204中所存储的计算机软件程序。计算机软件程序可以包括机器指令,这些机器指令在由处理器(诸如CPU 202)执行时使接收站106以下述方式对视频数据进行解码。解码器500还可以在硬件中实施,该硬件被包括在例如发送站102或接收站106中。

[0074] 类似于上述编码器400的重构路径,在一个示例中,解码器500包括执行各种功能以从压缩比特流420产生输出视频流516的以下阶段:熵解码阶段502、解量化阶段504、逆变换阶段506、帧内/帧间预测阶段508、重构阶段510、环路滤波阶段512和后滤波阶段514。解码器500的其他结构变型可以用于对压缩比特流420进行解码。

[0075] 当压缩比特流420被呈现用于进行解码时,压缩比特流420中的数据元素可以通过熵解码阶段502解码以产生一组量化变换系数。解量化阶段504对量化变换系数进行解量化(例如,通过将量化变换系数乘以量化器值),并且逆变换阶段506使用所选的变换类型对解量化变换系数进行逆变换,以产生导数残差,该导数残差可以与通过编码器400中的逆变换阶段412所创建的倒数残差相同。使用从压缩比特流420解码的报头信息,解码器500可以使用帧内/帧间预测阶段508来创建与在编码器400中所创建(例如,在帧内/帧间预测阶段402)的预测块相同的预测块。在重构阶段510,预测块可以被添加到导数残差,以创建重构块。环路滤波阶段512可以应用于重构块以减少构块效应。其他滤波可以应用于重构块。在示例中,后滤波阶段514应用于重构块以减少块失真,并且结果被输出作为输出视频流516。输出视频流516也可以被称为解码视频流;这些术语在本文中互换使用。

[0076] 解码器500的其他变型可以用于对压缩比特流420进行解码。例如,解码器500可以在没有后滤波阶段514的情况下产生输出视频流516。在解码器500的一些实施方式中,后滤波阶段514是在环路滤波阶段512之后应用的。环路滤波阶段512可以包括可选的去块滤波阶段。附加地或替代地,编码器400包括环路滤波阶段416中的可选的去块滤波阶段。

[0077] 编解码器可以使用多个变换类型。例如,变换类型可以是图4的变换阶段404用来生成变换块的变换类型。例如,变换类型(即,逆变换类型)可以是图5的解量化阶段504所使用的变换类型。可用变换类型可以包括一维离散余弦变换(1D DCT)或其近似、一维离散正弦变换(1D DST)或其近似、二维DCT(2D DCT)或其近似、二维DST(2D DST)或其近似以及恒等变换。其他变换类型可以是可用的。在示例中,一维变换(1D DCT或1D DST)可以在一维(例如,行或列)应用,并且恒等变换可以在其他维应用。

[0078] 在使用1D变换(例如,1D DCT、1D DST)(例如,1D DCT应用于变换块的列(或行,分别地))的情况下,量化系数可以使用逐行(即,光栅)扫描顺序或逐列扫描顺序进行代码化。

在使用2D变换(例如,2D DCT)的情况下,不同的扫描顺序可以用于对量化系数进行代码化。如上所述,不同的模板可以用于获取上下文,用于基于所使用的变换类型对非零映射的非零标记进行代码化。因此,在一实施方式中,模板可以是基于用于生成变换块的变换类型来选择的。如上所述,变换类型的示例包括:应用于行(或列)的1D DCT和应用于列(或行)的恒等变换;应用于行(或列)的1D DST和应用于列(或行)的恒等变换;应用于行(或列)的1D DCT和应用于列(或行)的1D DST;2D DCT;以及2D DST。其他变换组合可以包括变换类型。

[0079] 图6是根据本公开的实施方式的帧(诸如图3的帧306)的部分600的表示的框图。如图所示,帧的部分600包括四个 64×64 块610,这些块可以被称为超级块,在矩阵或笛卡尔平面中呈两行和两列。超级块可以具有更大或更小的大小。虽然图6是关于大小为 64×64 的超级块进行说明的,但是描述很容易扩展到更大(例如, 128×128)或更小的超级块大小。

[0080] 在一示例中,并且在不损失通用性的情况下,超级块可以是基本或最大代码化单元(CU)。每个超级块都可以包括四个 32×32 块620。每个 32×32 块620都可以包括四个 16×16 块630。每个 16×16 块630都可以包括四个 8×8 块640。每个 8×8 块640都可以包括四个 4×4 块650。每个 4×4 块650都可以包括16个像素,在笛卡尔平面或矩阵的每个对应的块中,这些像素都可以以四行和四列表示。像素可以包括表示在帧中捕捉的图像的信息,诸如亮度信息、颜色信息和位置信息。在一示例中,块(诸如如图所示的 16×16 像素块)可以包括:亮度块660,该亮度块660可以包括亮度像素662;以及两个色度块670/680,诸如U或Cb色度块670和V或Cr色度块680。色度块670/680可以包括色度像素690。例如,亮度块660可以包括 16×16 亮度像素662,并且每个色度块670/680都包括 8×8 色度像素690,如图所示。虽然示出了块的一种布置,但是任何布置都可以使用。虽然图6示出了 $N \times N$ 块,但是在一些实施方式中,可以使用 $N \times M$ 块,其中, $N \neq M$ 。例如,可以使用 32×64 块、 64×32 块、 16×32 块、 32×16 块或任何其他大小块。在一些实施方式中,可以使用 $N \times 2N$ 块、 $2N \times N$ 块或其组合。

[0081] 在一些实施方式中,视频代码化可以包括有序的块级代码化。有序的块级代码化可以包括按照一定顺序(诸如光栅扫描顺序)对帧的块进行代码化,其中,可以从帧的左上角的块或帧的部分开始识别和处理块,并且沿从左到右和从顶行到底行的行继续进行,依次识别每个块以进行处理。例如,在帧的顶行和左边列的超级块可以是被代码化的第一块,并且紧邻第一块的右边的超级块可以是被代码化的第二块。从顶部开始的第二行可以是被代码化的第二行,以便第二行的左边列中的超级块可以在第一行的最右边的列中的超级块之后进行代码化。

[0082] 在一示例中,对块进行代码化可以包括使用二叉树代码化,该二叉树代码化可以包括按照光栅扫描顺序对更小的块单元和块进行代码化。例如,在图6中所示的帧的部分的左下角所示的 64×64 块可以使用二叉树代码化进行代码化,在该二叉树代码化中,可以对左上方的 32×32 块进行代码化,然后可以对右上方的 32×32 块进行代码化,再然后可以对左下方的 32×32 块进行代码化,最后可以对右下方的 32×32 块进行代码化。每个 32×32 块都可以使用二叉树代码化进行代码化,在该二叉树代码化中,可以对左上方的 16×16 块进行代码化,然后可以对右上方的 16×16 块进行代码化,再然后对左下方的 16×16 块进行代码化,最后可以对右下方的 16×16 块进行代码化。每个 16×16 块都可以使用二叉树代码化进行代码化,在该二叉树代码化中,可以对左上方的 8×8 块进行代码化,然后可以对右上方的 8×8 块进行代码化,再然后对左下方的 8×8 块进行代码化,最后可以对右下方的 8×8 块

进行代码化。每个 8×8 块都可以使用二叉树代码化进行代码化,在该二叉树代码化中,可以对左上方的 4×4 块进行代码化,然后可以对右上方的 4×4 块进行代码化,再然后对左下方的 4×4 块进行代码化,最后可以对右下方的 4×4 块进行代码化。在一些实施方式中,对于 16×16 块, 8×8 块可以被省略,并且 16×16 块可以使用二叉树代码化进行代码化,在该二叉树代码化中,可以对左上方的 4×4 块进行代码化,然后可以按照光栅扫描顺序对 16×16 块中的其他 4×4 块进行代码化。

[0083] 在一示例中,视频代码化可以包括通过省略来自相应编码帧的原始帧中的一些信息,压缩原始或输入帧中所包括的信息。例如,代码化可以包括减少频谱冗余、减少空间冗余、减少时间冗余或其组合。

[0084] 在一示例中,减少频谱冗余可以包括使用基于亮度分量(Y)和两个色度分量(U和V或Cb和Cr)的颜色模型,该颜色模型可以被称为YUV或YCbCr颜色模型或颜色空间。使用YUV颜色模型可以包括使用相对大量的信息来表示帧的部分的亮度分量,并且使用相对少量的信息来表示帧的部分的每个相应的色度分量。例如,帧的部分可以由高分辨率亮度分量表示,该高分辨率亮度分量可以包括 16×16 像素块,并且由两个更低分辨率色度分量表示,每个更低分辨率色度分量都将帧的部分表示为 8×8 像素块。像素可以指示值(例如,范围从0到255的值),并且可以使用例如8个比特进行存储或发送。虽然本公开参照YUV颜色模型进行了描述,但是任何颜色模型都可以使用。

[0085] 减少空间冗余可以包括将块变换到频域,如上所述。例如,编码器的单元,诸如图4的熵编码单元408,可以基于空间频率使用变换系数值执行DCT。

[0086] 减少时间冗余可以包括使用帧之间的相似性,基于一个或多个参考帧使用相对少量的数据对帧进行编码,这些参考帧可以是视频流的先前被编码、解码和重构的帧。例如,当前帧的块或像素可以与参考帧的在空间上对应的块和像素类似。当前帧的块或像素可以与不同空间位置的参考帧的块或像素类似。因此,减少时间冗余可以包括生成指示空间差异的运动信息(例如当前帧中块或像素的位置与参考帧中块或像素的相应位置之间的平移)。

[0087] 减少时间冗余可以包括识别与当前帧的当前块或像素对应的参考帧中的块或像素或参考帧的部分。例如,可以在参考帧或参考帧的部分(可以存储在存储器中)中搜索最佳块或像素,以用于对当前帧的当前块或像素进行编码。例如,搜索可以识别参考帧的块,对于该块,参考块与当前块之间的像素值差被最小化,并且该搜索可以被称为运动搜索。搜索的参考帧的部分可以受到限制。例如,搜索的参考帧的部分(可以被称为搜索区域)可以包括参考帧的有限数量的行。在一示例中,识别参考块可以包括计算搜索区域中的块的像素与当前块的像素之间的成本函数,诸如绝对差之和(SAD)。

[0088] 参考帧中参考块的位置与当前帧中当前块之间的空间差可以被表示为运动向量。参考块与当前块之间的像素值差可以被称为差分数据、残差数据或被称为残差块。在一些实施方式中,生成运动向量可以被称为运动估计,并且当前块的像素可以使用笛卡尔坐标(诸如 $f_{x,y}$)基于位置来指示。类似地,参考帧的搜索区域的像素可以使用笛卡尔坐标(诸如 $r_{x,y}$)基于位置来指示。当前块的运动向量(MV)可以基于例如当前帧的像素与参考帧的相应像素之间的SAD来确定。

[0089] 如上所述,当前块可以使用帧内预测进行预测。帧内预测模式使用被预测的当前

块外围的像素。当前块外围的像素是当前块外部的像素。许多不同的帧内预测模式可以是可用的。

[0090] 一些帧内预测模式对使用至少一个外围像素而生成的预测块中的所有像素都使用单一值。作为示例,VP9编解码器包括帧内预测模式,被称为真运动(TM_PRED)模式,在该模式下,预测块的所有值都有值 $\text{predicted pixel}(x,y)$ = 对于所有的x和y的(顶部相邻者+左部相邻者-左上相邻者)。作为另一个示例,DC帧内预测模式(DC_PRED)使得预测块的每个像素都被设置为值 $\text{predicted pixel}(x,y)$ = 整个顶部行和左边列的平均值。

[0091] 其他帧内预测模式(可以被称为方向帧内预测模式)使得每一个都可以有相应的预测角。其他类型的帧内预测模式也可以是可用的。

[0092] 帧内预测模式可以由编码器选择作为率失真环路的部分。简而言之,各种帧内预测模式可以被测试,以确定哪种类型的预测对于给定的速率或要在编码视频比特流中传输的比特数目(包括比特流中所包括的开销比特以指示所使用的预测类型)具有最低失真。

[0093] 在示例编解码器中,以下13个帧内预测模式可以是可用的:DC_PRED、V_PRED、H_PRED、D45_PRED、D135_PRED、D117_PRED、D153_PRED、D207_PRED、D63_PRED、SMOOTH_PRED、SMOOTH_V_PRED和SMOOTH_H_PRED以及PAETH_PRED。13个帧内预测模式中的一个可以用于预测亮度块。

[0094] 图7是根据本公开的实施方式的帧内预测模式的示例的示意图。帧内预测模式710说明V_PRED帧内预测模式,该模式通常被称为垂直帧内预测模式。在这种模式下,第一列中的预测块像素设置为外围像素A的值;第二列中的预测块像素设置为像素B的值;第三列中的预测块像素设置为像素C的值;并且第四列中的预测块像素设置为像素D的值。

[0095] 帧内预测模式720说明H_PRED帧内预测模式,该模式通常被称为水平帧内预测模式。在这种模式下,第一行中的预测块像素设置为外围像素I的值;第二行中的预测块像素设置为像素J的值;第三行中的预测块像素设置为像素K的值;并且第四行中的预测块像素设置为像素L的值。

[0096] 帧内预测模式730例示了D117_PRED帧内预测模式,之所以称之为D117_PRED帧内预测模式是因为箭头的方向与水平方向成大约117°的角,外围像素将沿该箭头方向传播以生成形成对角线的预测块。即,在D117_PRED中,预测角为117°。帧内预测模式740说明D63_PRED帧内预测模式,该模式对应于63°的预测角。帧内预测模式750说明D153_PRED帧内预测模式,该模式对应于153°的预测角。帧内预测模式760说明D135_PRED帧内预测模式,该模式对应于135°的预测角。

[0097] 预测模式D45_PRED和D207_PRED(未示出)分别对应于预测角45°和207°。DC_PRED对应于这样的预测模式:其中所有的预测块像素都设置为单一值,该单一值是外围像素A-M的组合。

[0098] 在PAETH_PRED帧内预测模式下,像素的预测值如下确定:1) 计算基值作为一些外围像素的组合,以及2) 使用最接近基值的一些外围像素之一作为预测像素。PAETH_PRED帧内预测模式是使用像素712(在位置 $x=1, y=2$)作为示例来说明的。在一些外围像素的组合的示例中,基值可以计算为 $\text{base} = B + K - M$ 。即,基值等于:与要被预测的像素在同一行中的左部外围像素的值+与像素在同一列中的上部外围像素的值-左上角的像素的值。

[0099] 在SMOOTH_V帧内预测模式下,预测块的最底部的行的预测像素是用左边列中的最

后一个像素的值(即,在位置L的像素的值)来估计的。预测块的剩余像素是在垂直方向上通过二次插值计算的。

[0100] 在SMOOTH_H帧内预测模式下,预测块的最右边的列的预测像素是用左边行中的最后一个像素的值(即,在位置D的像素的值)来估计的。预测块的剩余像素是在水平方向上通过二次插值计算的。

[0101] 在SMOOTH_PRED帧内预测模式下,预测块的最底部的行的预测像素是用左边列中的最后一个像素的值(即,在位置L的像素的值)来估计的,并且预测块的最右边的列的预测像素是用最顶部的行中的最后一个像素的值(即,在位置D的像素的值)来估计的。预测块的剩余像素被计算为按比例加权的和。例如,在预测块的位置 (i, j) 的预测像素的值可以被计算为像素 L_j 、 R 、 T_i 和 B 的值的按比例加权的和。像素 L_j 是位于左列中并且与预测像素在同一行中的像素。像素 R 是SMOOTH_H所提供的像素。像素 T_i 是位于上部行并且与预测像素在同一列中的像素。像素 B 是SMOOTH_V所提供的像素。权值可以相当于水平和垂直方向的二次插值。

[0102] 编码器所选的帧内预测模式可以在比特流中传输给解码器。帧内预测模式可以使用上下文模型进行熵代码化(由编码器编码和/或由解码器解码)。

[0103] 一些编解码器使用左边和上面的邻居块的帧内预测模式作为用于对当前块的帧内预测模式进行代码化的上下文。使用图7作为示例,左边的邻居块可以是包含像素I-L,并且上面的邻居块可以是包含像素A-D的块。

[0104] 示意图700图示了VP9编解码器中可用的帧内预测模式。代码化的VP9支持一组块大小范围从 4×4 到 32×32 的10个帧内预测模式。这些帧内预测模式是DC_PRED、TM_PRED、H_PRED、V_PRED和6个斜向预测模式:D45_PRED、D63_PRED、D117_PRED、D135_PRED、D153_PRED、D207_PRED,大约对应于45度、63度、117度、135度、153度和207度的角(相对于水平轴逆时针方向测量)。

[0105] 图8是搜索对块进行代码化的最佳模式的过程800的流程图。过程800是使用蛮力方法确定最佳模式的模式决策过程的说明性高级过程。为了便于描述,过程800是关于选择帧内预测模式对预测块进行编码而被描述的。可以由类似于过程800的过程确定的最佳模式的其他示例包括确定变换类型和确定变换大小。过程800可以由编码器(诸如图4的编码器400)使用蛮力方法进行模式决策来实施。

[0106] 在802中,过程800接收图像块。由于过程800是关于确定帧内预测模式而被描述的,图像块可以是预测单元。如关于图7所描述的,每个叶节点代码化块(例如,块702-1、702-5、702-6、702-7、702-8、702-3或702-4)可以被划分为一个或多个预测单元。因此,图像块可以是一个这样的预测单元。

[0107] 在804中,过程800确定(例如,选择、计算、选定等)模式列表。模式列表可以包括 K 个模式,其中, K 是整数。模式列表可以被表示为 $\{m_1, m_2, \dots, m_k\}$ 。编码器可以具有可用的帧内预测模式列表。例如,可用的帧内预测模式列表可以是 $\{DC_PRED, TM_PRED, V_PRED, H_PRED, D45_PRED, D135_PRED, D117_PRED, D153_PRED, D207_PRED, D63_PRED\}$ 。在另一个示例中,可用的帧内预测模式列表可以是 $\{DC_PRED, V_PRED, H_PRED, D45_PRED, D135_PRED, D117_PRED, D153_PRED, D207_PRED, D63_PRED, SMOOTH_PRED, SMOOTH_V_PRED$ 和 $SMOOTH_H_PRED, PAETH_PRED\}$ 。这些帧内预测模式的描述被省略,因为该描述与对本公开的理解无关。

在804中确定的模式列表可以是可用的帧内预测模式列表的任何子集。

[0108] 在806中,过程800将BEST_COST变量初始化为高值(例如,INT_MAX,其可以等于2,147,483,647),并且将环路变量i初始化为1,其对应于要检查的第一模式。

[0109] 在808中,过程800为mode_i估算(例如,计算)RD_COST_i。在810中,过程800测试检查中的当前模式mode_i的RD成本RD_COST_i是否小于当前最佳成本BEST_COST。如果测试是肯定的,则在812中,过程800将最佳成本更新为当前模式的成本(即,BEST_COST=RD_COST_i),并且将当前最佳模式索引(BEST_MODE)设置为环路变量i(BEST_MODE=i)。过程800然后继续执行814,来增加环路变量i(即,i=i+1),以准备检查下一个模式(如果有的话)。如果测试是否定结果的,则过程800继续执行814。

[0110] 在816中,如果有更多的模式要检查,则过程800继续返回执行808;否则过程800继续执行816。在818中,过程800输出最佳模式BEST_MODE的索引。输出最佳模式可以指将最佳模式返回给过程800的调用方。输出最佳模式可以指使用最佳模式对图像进行编码。输出最佳模式可以有其他语义。过程800在输出最佳模式之后终止。

[0111] 图9是估计使用预测模式m_i对图像块X进行代码化的速率和失真成本的示例900的框图。过程900可以由编码器(诸如图4的编码器400)执行。过程900包括使用预测模式m_i对图像块X进行假设编码,以确定对块进行编码的RD成本。过程900可以由过程800在808中使用。

[0112] 假设编码过程是执行代码化步骤但不将比特输出到压缩比特流(诸如图4的压缩比特流420)的过程。由于目的是估计速率(也称为比特率),假设编码过程可以被视为或称为速率估计过程。假设编码过程估算对图像块X进行编码所需的比特数(RATE)。示例900还包括基于图像块X与图像块X的重构版本之间的差的失真(DISTORTION)。

[0113] 在904中,确定使用模式m_i的预测。预测可以如图4的帧内/帧间预测阶段402所描述的那样被确定。在906中,将残差确定为图像块902和预测之间的差。在908和910中,对残差进行变换和量化,诸如,分别关于图4的变换阶段404和量化阶段406所描述的。速率(RATE)由执行假设编码的速率估计器912计算。在一示例中,速率估计器912可以执行熵编码,诸如关于图4的熵编码阶段408所描述的。

[0114] 量化残差在914中进行解量化(诸如,例如关于图4的解量化阶段410所描述的),在916中进行逆变换(诸如,例如关于图4的逆变换阶段412所描述的),并且在918中重构(诸如,例如关于图4的重构阶段414所描述的),以生成重构块。失真估计器920计算图像块X和重构块之间的失真(即,视频质量损失)。在一示例中,失真可以是图像块X和重构块的像素值之间的均方误差。失真可以是图像块X和重构块的像素值之间的绝对差值误差的和。任何其他合适的失真测量都可以使用。

[0115] 然后,使用公式(5)中所示的拉格朗日乘子将速率RATE和失真DISTORTION结合成标量值(即,RD成本):

$$[0116] \quad \text{DISTORTION} + \lambda_{\text{mode}} \times \text{RATE}, \quad (5)$$

[0117] 公式5的拉格朗日乘子 λ_{mode} 可以根据执行示例900的操作的编码器来如上所述计算。

[0118] 图8和图9图示了模式决策的传统(即,蛮力)方法很大程度上是串行过程,该串行过程本质上是使用候选模式对图像块X进行代码化,以确定具有最佳成本的模式。在模式决

策中已经使用了技术来降低复杂性。例如，一旦满足某些条件，诸如，例如，率失真成本低于阈值，早期终止的技术便会用于终止图8的过程800的环路。

[0119] 其他技术包括选择(例如，基于启发式)可用候选模式的子集或使用候选模式上的多次(multi-passes)。因此，在大多数编解码器中，启发式用于从所有可能的模式中挑选前N个候选(即，N个候选帧内预测模式)，而率失真优化通常只在这前N个候选上使用，以挑选最佳候选。在典型的情况下，N个候选可以包括最有可能的候选。启发式可以包括度量，诸如简单的绝对差之和(SAD)、绝对变换差之和(SATD)、均方误差(MSE)或一些其他启发式度量。即，例如，当对当前块进行编码时并且考虑到候选帧内预测模式 m_1 ，确定预测块，并且在当前块和预测块之间计算一个或多个启发式度量(例如，SAD、SATD、MSE等)。

[0120] 这种容易获得的(例如，可计算的)启发式度量(即，启发式特征)可以包括在帧内预测模式选择的ML模型中，作为输入特征，并且有望改进ML模型的预测准确性。

[0121] 图10A至图10B是用于模式决策的卷积神经网络(CNN)的示例1000和1050的框图。

[0122] 图10A图示了典型的CNN网络或只是CNN的示例1000的高级框图。如上所述，CNN是机器学习模型的示例。在CNN中，特征提取部分通常包括一组卷积操作，这通常是用于基于滤波器(通常是大小k的平方，在不损失通用性的情况下)对输入图像进行滤波的一系列滤波器。例如，在机器视觉的场境(context)下，这些滤波器可以用于查找输入图像中的特征。例如，特征可以包括边缘、拐角、端点等。随着堆叠的卷积操作的数量增加，之后的卷积操作可以查找更高级的特征。

[0123] 在CNN中，分类部分通常是一组全连接(FC)层，这也可以被称为稠密操作。全连接层可以被认为查看图像的所有输入特征，以便生成高级分类器。几个阶段(例如，一系列)的高级分类器最终生成所需的分类输出。

[0124] 如上所述，典型的CNN网络由若干全连接层之后的若干卷积操作(例如，特征提取部分)组成。每个类型的操作数量以及其对应的大小通常是在机器学习的训练阶段确定的。如本领域技术人员认识到的，附加层和/或操作可以包括在每个部分中。例如，池化、最大池化、暂退法(dropout)、激活、归一化、批量归一化和其他操作的组合可以用卷积操作(即，在特征提取部分中)和/或全连接操作(即，在分类部分910中)来分组。全连接层可以被称为稠密操作。如本领域技术人员所认识到的，卷积操作可以使用SeparableConvolution2D或Convolution2D操作。

[0125] 如本公开中所使用的，卷积层可以是一组操作，从Convolution2D操作或SeparableConvolution2D操作开始，然后是零个或多个操作(例如，池化、暂退法、激活、归一化、批量归一化、其他操作或其组合)，直到到达另一个卷积层、稠密操作或CNN的输出。类似地，稠密层可以是一组操作或层，从稠密操作(即，全连接层)开始，然后是零个或多个操作(例如，池化、暂退法、激活、归一化、批量归一化、其他操作或其组合)，直到到达另一个卷积层、另一个稠密层或网络的输出。基于卷积网络的特征提取部分与使用稠密操作的特征分类部分之间的边界可以通过平坦化(Flatten)操作进行标记，该平坦化操作将多维矩阵从特征提取平坦化为向量。

[0126] 在典型的CNN中，每个卷积层都可以由一组滤波器构成。当每次将滤波器应用于输入数据的子集时，跨整个输入应用滤波器，诸如通过扫掠该输入。该层所执行的操作通常是线性/矩阵乘法。卷积滤波器的输出可以使用激活函数进行进一步滤波。激活函数可以是线

性函数或非线性函数(例如,sigmoid函数、arcTan函数、tanH函数、ReLu函数等)。

[0127] 每个全连接操作都是线性操作,其中,每个输入都通过权值连接到每个输出。因此,具有N个输入和M个输出的全连接层可以具有总共 $N \times M$ 个权值。如上所述,稠密操作之后通常可以有非线性激活函数来生成该层的输出。

[0128] 用于执行帧和超级块的分析(诸如用以推断本文中所描述的分区)的一些CNN网络架构可以包括:几个特征提取部分,这些特征提取部分以不同粒度(例如,以超级块的不同子块大小)提取特征;以及平坦化层(可以被称为连结(concatenation)层),该连结层接收每个提取部分的最后一个卷积层的输出。平坦化层将不同的特征提取部分所提取的所有特征都聚合成一个输入集。平坦化层的输出可以被馈送到分类部分的全连接层(即,用作其输入)。因此,整个网络的参数数量可以由特征提取部分(即,卷积层)和分类部分(即,全连接层)之间的接口处的参数数量支配(例如,定义、设置等)。即,网络的参数数量由平坦化层的参数决定。

[0129] 这种架构的机器学习模型往往具有大量参数和操作。在一些情况下,机器学习模型可以包括数百万个参数。这种大型模型可能不能被有效地或有效率地用于推断可能受到约束(例如,计算上的约束、能量约束和/或存储器约束)的设备(例如,移动设备)上的分类。即,一些设备可能没有足够的计算能力(例如,在速度方面)或内存存储(例如,RAM)来处理(例如,执行)这种大型模型。

[0130] 图10B图示了用于预测HEVC中的35个帧内预测模式之一的现有技术CNN的示例1050的高级框图。在CNN(诸如示例1050)中,卷积层用于提取特征,并且全连接层用作分类层。

[0131] 在示例1050中,要被编码的帧(例如,帧1052)的当前块(例如,当前块1054)可以通过一个或多个卷积层(例如,卷积层1056和1058)、一个或多个最大池化层(例如,池化层1060)和一个或多个全连接层(例如,全连接层1062和1064)进行馈送。

[0132] 选定模型(即,示例1050)的容量,以便可以在低训练误差和良好的泛化(generalization)之间实现良好的权衡。在最后一层(即,全连接层1064)中,执行将其分类为35类(即,帧内预测模式)。

[0133] 在分析架构(诸如,示例1050)的性能时,可以进行几次观察。

[0134] 第一观察结果是这种模型往往很大;即,这种模型具有大量参数和操作(例如,超过 32×32 块的6700万个参数)。大量参数和操作可以使这种模型无法在计算或大小受约束的平台(诸如移动平台)上操作(例如,用于推断)。

[0135] 第二观察结果是重要的特征(诸如量化参数(QP)和/或编解码器生成的启发式)可能没有被利用(即,没有被用作模型的输入)。这种特征可以对最优帧内预测模式的决策产生重大影响。

[0136] 第三观察结果是CNN的输出用于完全取代当前编解码器决策。无论ML在其预测中的置信度水平如何,都会忽视(例如,取代)编解码器决策。

[0137] 关于第一观察结果,希望ML模型尽可能小(例如,超轻)。关于第二观察结果,发明人已经发现,插入QP的非线性组合大大改进了ML模型的准确性。这在深度缩小的网络模型(例如,超轻模型)上尤其如此。关于第三观察结果,更好的方法是,根据ML预测的置信度,自适应地结合ML预测和现有的基于启发式的编解码决策,以优化模式选择准确性。在一示例

中,现有的基于启发式的编解码决策可以包括使用从一个或多个候选模式(诸如,最可能的帧内预测模式(MPM))获取的ML模型输入。

[0138] 图11是根据本公开的实施方式的、用于通过编码器(即,第一编码器)使用帧内预测对图像块进行编码的过程1100的流程图。过程1100使用输入数据对机器学习模型进行训练,以推断模式决策,诸如帧内预测模式决策(即,ML帧内预测模式)。然后,过程1100使用经过训练的机器学习模型来推断模式决策(例如,ML帧内预测模式),以用于对图像块(即,当前块)进行编码。当前块可以是亮度(Y)块、色度(U或V)块或要使用帧内预测进行编码的任何其他类型的图像块。图像块的大小可以是 $M \times N$,其中, M 可以等于 N 。即,图像块可以是能够使用帧内预测进行编码的任何类型和大小的块。

[0139] 在1102中,过程1100对机器学习(ML)模型进行训练。ML模型可以使用训练数据1112来训练。训练数据1112的每个训练单个数据(datum)都可以包括:视频块,该视频块通过传统的编码方法(例如,通过第二编码器)进行编码,诸如关于图4和图6至图9所描述的;地面实况(ground truth)帧内预测模式决策;预先计算的特征;以及零个或多个附加输入。地面实况帧内预测模式决策是第二编码器使用例如蛮力手段为该块确定的帧内预测模式。

[0140] 在训练阶段,生成ML模型的参数,以便对于至少一些训练数据,ML模型可以针对训练单个数据推断训练单个数据的结果帧内预测模式决策(即,ML帧内预测模式)。理想地,对于每个输入块,ML帧内预测模式(即,由ML模型推断出的模式)都与地面实况帧内预测模式决策相同。

[0141] 预先计算的特征被这样称呼(即,“预先计算”),是为了将它们与可以由深度学习神经网络(诸如卷积神经网络)提取(例如,获取、推断等)的特征进行区分。

[0142] 在一示例中,机器学习(ML)模型可以是小型的、全稠密的网络结构。即,ML模型只包括全连接层,但不包括卷积层和/或特征提取层。

[0143] 图12A是根据本公开的实施方式的全稠密网络结构的示例1200的示意图。输入层1202包括 N 个预先计算的特征,这些特征被输入到隐藏层1204。 M 个输出从输出层1206输出。在一示例中,隐藏层1204只包括一个带有50个节点的隐藏层。输出层1206的 M 个输出对应于编码器中的可用帧内预测模式的数量,在该编码器中使用示例1200/示例1200与该编码器一起使用。例如,如果示例1200要与VP9一起使用,该VP9包括10个可用帧内预测模式,则 $M=10$ 。

[0144] 在一示例中, N 个预先计算的特征可以是17个预先计算的特征。然而,可以使用更多或更少的特征。在一示例中,一个或多个预先计算的特征可以是标量值。现在关于当前块描述预先计算的特征。在训练阶段,当前块是训练块。在推断阶段(如下所述),当前块是需要推断帧内预测模式的块。

[0145] 本文中所描述的示例1200包括大约1600个参数,显著小于上述的CNN模型。

[0146] 第一预先计算的特征可以是量化参数的非线性函数。

[0147] 已知,视频编解码器中的量化参数可以用于控制速率和失真之间的权衡。通常,更大的量化参数指的是导致更低速率但更高失真的更高量化(诸如变换系数);并且更小的量化参数指的是导致更高速率但更低失真的更低量化。在本公开中,变量QP、 q 和 Q 可以被互换来用于指代量化参数。

[0148] 由于量化参数可以用于控制速率和失真之间的权衡,量化参数可以用于计算与每

个参数组合相关联的度量。度量可以结合编码参数组合的速率值和失真值。在一示例中,QP可以用于获取乘子,该乘子用于将速率值和失真值结合成一个度量。一些编解码器可以将乘子称为拉格朗日乘子(表示为 λ_{mode});其他编解码器可以使用被称为rdmult的类似乘子。每个编解码器都可以有不同的计算乘子的方法。例如,HEVC使用公式 $\lambda_{\text{mode}} = 0.85 \times 2^{(QP - 12)/3}$ 。例如,H.263使用公式 $\lambda_{\text{mode}} = 0.85 \times Q_{\text{H263}}^2$ 。例如,VP9使用公式 $\text{rdmult} = 88 \cdot q^2/24$ 。例如,AV1使用公式 $\lambda_{\text{mode}} = 0.12 \cdot Q_{\text{AV1}}^2/256$ 。

[0149] 在确定最佳模式(即,编码帧内预测模式)时,量化参数可以是重要特征。因此,在一示例中,QP应当用作ML模型的输入特征。从上面的公式中可以看出,乘子与量化参数具有非线性关系。因此,在另一个示例中,QP的非线性函数(例如,值)可以用作ML模型的输入。在一示例中,可以使用 $\log(\text{qp})$ 。非线性QP可以大大改进ML模型的准确性。此外, $\log(\text{qp})$ 在乘以任何所得的训练权值时可以很好地模拟具有高阶的QP多项式(例如,QP、QP²、QP³等)。按照 \log 函数,ML模型可以相应地学习 $n \cdot \log(\text{qp})$,其中,n对应于多项式指数。换句话说,ML模型可以学习量化参数的多项式表达式,诸如, $\log(\text{qp}) + \log(\text{qp})^2 + \log(\text{qp})^3$ 。在又一个示例中,可以使用非线性函数 $f(\text{QP}) = \text{QP}^\alpha$,其中, α 是不等于0或1的整数(即, $\alpha \neq 0$ 和 $\alpha \neq 1$)。在一示例中, $\alpha = 2$ 。在一般情况下,非线性函数与第二编码器为确定率失真计算中所使用的乘子而使用的函数具有相同类型,如上所述。

[0150] 第二预先计算的特征可以是当前块(I(x,y))和对应的预测块之间的对应误差。即,对于每个可用的帧内预测模式(mode_i),计算(例如,确定、获取、获得等)对应的预测块(即,I'(x,y)),并且计算当前块与对应的预测块之间的误差(例如,cost_i)。在一示例中,成本可以是绝对差之和(SAD)。因此,对于每个mode_i,将cost_i计算为 $\text{cost}_i = \sum_{\text{对于全部x和y}} |I'(x,y) - I(x,y)|$ 。可以使用误差/成本(诸如,平方差和或一些其他误差)的其他示例。因此,在VP9的情况下,其中,存在10个可用帧内预测模式,可以计算10个成本。这些基于块的SAD成本可以携带关于最优预测模式可能是什么的重要信息。

[0151] 在一示例中,第二预先计算的成成本可以是与可用帧内预测模式的子集相关联的成本。例如,成本可以是方向帧内预测模式计算的。因此,关于VP9,由于TM_PRED帧内预测模式和/或DC_PRED帧内预测模式的预先计算的特征,可以不计算成本。

[0152] 第三预先计算的特征可以是当前块的平均值。更具体地是当前块的像素值的平均值。第四预先计算的特征可以是当前块的方差。即,方差可以是当前块的像素值的方差。

[0153] 通过将当前块与一个或多个滤波器进行卷积,可以获得附加的预先计算的特征。对于每个滤波器,获得卷积块。附加的预先计算的特征是从卷积块获取的标量值。在一示例中,标量值可以是卷积块的值的绝对值之和。获得卷积块的值的绝对值之和的示例是关于图13进行描述的。

[0154] 图13是根据本公开的实施方式的、获得卷积块的绝对值之和的示例。示例1300包括图像块(即,当前块)的区域1302。为了该示例的目的,区域1302被示出为4×4区域。然而,要理解,卷积滤波器可以应用于任何大小块、超级块、图像区域或图像。

[0155] 在该示例中,使用大小为3×3的滤波器1304。然而,滤波器可以具有不同的大小。示例1300使用步幅为1的重叠卷积操作。然而,步幅可以大于1。例如,可以使用步幅等于滤波器大小的非重叠卷积操作。因此,在水平方向和垂直方向中的每一个上的步幅大小可以是3。

[0156] 卷积块1314是将滤波器1304与区域1302进行卷积的输出。

[0157] 滤波器1304首先与子区域1306进行卷积(例如,使用矩阵乘法运算)。因此,卷积块1314的值1316可以被计算为 $(230 \times 0.25 + 226 \times 0.5 + 10 \times 0.25 + 163 \times 0 + 124 \times 0 + 173 \times 0 + 201 \times (-0.25) + 104 \times (-0.5) + 150 \times (-0.25)) = 33.25$ 。滤波器1304然后与子区域1308进行卷积。因此,值1318可以被计算为 $(226 \times 0.25 + 10 \times 0.5 + 232 \times 0.25 + 124 \times 0 + 173 \times 0 + 110 \times 0 + 104 \times (-0.25) + 150 \times (-0.5) + 100 \times (-0.25)) = -6.5$ 。滤波器1304然后与子区域1310进行卷积。因此,值1320可以被计算为 $(163 \times 0.25 + 124 \times 0.5 + 173 \times 0.25 + 201 \times 0 + 104 \times 0 + 150 \times 0 + 100 \times (-0.25) + 81 \times (-0.5) + 91 \times (-0.25)) = 57.75$ 。滤波器1304然后与子区域1312进行卷积。因此,值1322可以被计算为 $(124 \times 0.25 + 173 \times 0.5 + 110 \times 0.25 + 104 \times 0 + 150 \times 0 + 100 \times 0 + 81 \times (-0.25) + 91 \times (-0.5) + 17 \times (-0.25)) = 75$ 。

[0158] 绝对值之和然后被计算为 $|33.25| + |-6.5| + |57.75| + |75| = 172.5$ 。

[0159] 返回图12。可以选择滤波器(也可以被称为内核)来测量当前块中的整体边缘方向。在一示例中,可以选择滤波器来测量沿可用帧内预测模式的主方向的整体边缘方向。

[0160] 主方向可以对应于通过蛮力方法所选择的最常见模式。例如,在VP9中,最常见模式是与水平方向(即,H_PRED帧内预测模式)、垂直方向(即,V_PRED帧内预测模式)、45度方向(即,D45_PRED帧内预测模式)和135度方向(即,D135_PRED帧内预测模式)相关联的模式。

[0161] 在一示例中,可以使用Sobel算子。Sobel算子对图像块执行2D空间梯度测量。因此,Sobel算子可以强调在图像块中与具有对应的方向的边缘相对应的高空间频率区域。

[0162] 可以使用任何数量的滤波器。在一示例中,可以使用四个滤波器。因此,可以获得四个卷积块和四个相应的标量值。在一示例中,每个标量值都可以是相应卷积块中所有值的绝对值之和。用于检测水平方向、垂直方向、135度方向和45度方向上的强边缘响应的四个滤波器分别由图3的滤波器1350(与滤波器1304相同)、1352、1354和1356给出。

[0163] 重申一下,对于每个滤波器(例如,Sobel滤波器),滤波器与当前块(例如, $N \times N$ 块)进行卷积,以生成 $M \times M$ 卷积块。 M 的值可以与步幅和卷积模式的不同值不同。每个方向都可以被计算为卷积图像中所有像素的绝对值的和。

[0164] 使用预先计算的特征,诸如上述特征(例如,当前块的像素值的平均值和方差、基于滤波器(例如,Sobel滤波器)的方向特征和基于SAD的启发式成本),可以提供对当前块的有效描述,并且包含重要信息,该重要信息可以导致对当前块的帧内预测模式的正确选择。附加地,这种预先计算的特征可以有效地取代对传统模型中的大特征提取层的需求,从而显著降低ML模型复杂性和大小。

[0165] 在另一个示例中,除了使用预先选择的内核权值(例如,滤波器1350-1356),内核(滤波器)权值可以使用一组卷积操作来学习。在一示例中,内核权值可以在使用卷积网络的离线训练中学习。学习到的内核可以代替滤波器1350-1356用来获得对应的卷积块。在另一个示例中,卷积操作可以是ML模型的部分,如图12B中所示。

[0166] 图12B是根据本公开的实施方式的网络结构的示例1250的示意图。示例1250与示例1200类似,除了示例1250包括用于生成滤波器权值的一组卷积层1252。从卷积层1252输出的滤波器权值可以用于获得对应的卷积块。对于每个对应的卷积块,计算绝对值之和。对应的卷积块的绝对值之和然后用作输入层1202的输入的部分。因此,滤波器的值(即,权值)在机器学习模型的训练阶段(诸如,在图11的1102中)可以由机器学习模型确定。一组卷积

层1252的输入可以是图像块的原始像素值。

[0167] 重申一下,在训练阶段(即,在1102中),ML模型学习(例如,训练、构建、获取等)接受块和预先计算的特征作为输入的映射(即,函数),并且输出ML帧内预测模式。

[0168] 在训练阶段,为了使学习到的函数可以尽可能有用,最好使用大范围的输入块和大范围的可能QP值(诸如代表真实世界应用的QP值)来训练ML模型。

[0169] 如上所述,经过训练的ML模型是小型模型,该模型可以容纳资源受约束的设备(例如,要在该设备上使用/由该设备使用)。利用模型复杂性和预测准确性之间的权衡,更小的ML模型通常意味着准确性下降。

[0170] 为了补偿下降的准确性,根据本公开的一些实施方式可以根据ML模型的预测质量自适应地将ML决策(即,ML帧内预测模式)与编码器的基于启发式的选择(即,最可能的帧内预测模式)合并(例如,融合、补充等)。

[0171] 在一示例中,最可能的帧内预测模式可以是基于SAD的模式决策。基于SAD的决策可以是具有最小SAD成本的帧内预测模式,如上所述。在另一个示例中,编码器可以通过一些其他方式选择最可能的帧内预测模式。例如,当前块的最可能的帧内预测模式可以基于当前块的一个或多个相邻块(例如,左边相邻块和上面相邻块)的帧内预测模式来选择。

[0172] 现在描述合并(即,融合)策略。

[0173] 让 x 表示最可能的帧内预测模式。让 m 表示ML帧内预测模式。即, m 是由ML模型预测的帧内预测模式。让 y 成为地面实况帧内预测模式决策。即, y 表示编码器使用对当前块进行编码的蛮力方法选择的帧内预测模式。

[0174] 在训练阶段,根据等式(1)和(2)计算两个概率。

$$[0175] \quad P_x[i] = p(y = i | x = i, m \neq i) \quad (1)$$

$$[0176] \quad P_m[i] = p(y = i | x \neq i, m = i) \quad (2)$$

[0177] 在等式(1)中,地面实况是模式 i , x 和 m 是不同的预测模式,并且 $P_x[i]$ 是 x 为正确模式的概率。即,ML帧内预测模式(即,通过ML模型推断的帧内预测模式)不等于蛮力选择的帧内预测模式;然而,启发式(即,最可能的)模式等于蛮力选择的帧内预测模式。

[0178] 在等式(2)中,地面实况是模式 i , x 和 m 是不同的预测模式,并且 $P_m[i]$ 是 m 为正确模式的概率。即,ML帧内预测模式(即,通过ML模型推断的帧内预测模式)等于蛮力选择的帧内预测模式;然而,启发式(即,最可能的)模式不等于蛮力选择的帧内预测模式。

[0179] 基于在训练过程中对所有测试数据累积的 $P_x[i]$ 和 $P_m[i]$ 概率,也可以生成决策矩阵。如果编码器包括 N 个可用帧内预测模式,则决策矩阵可以是 $N \times N$ 矩阵。例如,如果存在10个可用帧内预测模式(诸如在VP9中),则决策矩阵可以是 10×10 矩阵。

[0180] 在一示例中,决策矩阵 A 的条目可以如等式(3)中所示设置。

$$[0181] \quad A[i][j] = \begin{cases} 1 & \text{if } P_x[i] < P_m[j] \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

[0182] 在推断期间,并且在一示例中,决策矩阵可以如下使用:

[0183] 如果最可能的帧内预测模式(例如,基于SAD的模式)被预测为模式 i ,并且ML模型将模式预测为 j ,则评估ML模型预测和可能模型的模式预测的相对可靠性,并且选择最可靠的预测。同样,相对可靠性是在训练期间生成的决策矩阵的值。本文中所描述的自适应决策融合策略的作用可以显著提高小型模型的性能。

[0184] 在一示例中,在推断期间,如果选择了ML预测模式(即,ML帧内预测模式)(即,选择它是因为它比最可能的模式更可靠),并且ML帧内预测模式与最可能的帧内预测模式(例如,基于SAD的模式)不同,则与ML帧内预测模式相关联的成本可以设置为值,该值小于与最可能的帧内预测模式相关联的成本。众所周知,模式成本可以在编码器中用于决定例如最优分区、预测模式和/或变换大小。在一示例中,与ML帧内预测模式相关联的成本可以减少一定比例(例如,10%、25%或一些其他比例)。

[0185] 图14图示了根据本公开的实施方式的决策矩阵的示例。第一决策矩阵1402是在关于图12A所描述的第一ML模型的训练期间生成的。第一ML模型包括包含10个节点的一个隐藏层。第一ML模型大小为300个参数。第二决策矩阵1404也是在关于图12A所描述的第二ML模型的训练期间生成的。第二ML模型包括包含50个节点的一个隐藏层。第二ML模型大小为1600个参数。

[0186] 第一ML模型和第二ML模型是使用相同的输入数据来训练的。第一ML模型和第二ML模型被训练以与VP9解码器一起使用。因此,第一决策矩阵1402和第二决策矩阵1404的*i*和*j*具有从0到9的值,这些值分别对应于可用帧内预测模式DC_PRED、V_PRED、H_PRED、D45_PRED、D135_PRED、D117_PRED、D153_PRED、D207_PRED、D63_PRED和TM_PRED。

[0187] 在比较第一决策矩阵1402和第二决策矩阵1404时,可以观察到,对于较大的模型(即,第二ML模型),由于其预测准确性很高,决策矩阵(即,第二决策矩阵1404)几乎总是选择ML决策,而不是最可能的帧内预测决策;而对于较小的模型(即,第一ML模型)则不是如此。这种比较说明了将融合策略用于小型模型的重要性。当ML模型足够大时,这种融合并不重要,并且可能不是必要的,因为ML模型决策总是优先于最可能的模式决策。然而,对于较小的ML模型,低质量的ML决策可以通过最可能的帧内预测模式决策进行修改(例如,由其补充、与其融合,等等)。

[0188] 然后,在推断阶段,过程1100可以使用ML模型。推断阶段包括操作1104和1106。分隔1110表明训练阶段和推断阶段可以在时间上分离。因此,推断阶段可以由第一编码器执行,并且训练数据1112可以由第二编码器生成。在一示例中,第一编码器和第二编码器是相同的编码器。即,训练数据1112可以由执行推断阶段的相同编码器生成。在任何一种情况下,推断阶段使用经过训练的机器学习模型,如关于1102所描述的。如上所述,在一些示例中,决策矩阵也可以在推断阶段使用。

[0189] 在1104中,将输入呈现给ML模块。即,输入被呈现给这样的模块:该模块并入、包括、执行、实施ML模型,等等。ML模块可以是硬件实施的模块。ML模块可以作为可执行指令存储在存储器中,这些指令可以由处理器执行。

[0190] 输入可以包括图像块和预先计算的特征,如上面关于ML模型的训练阶段所描述的。

[0191] 在1106中,过程1100获得编码帧内预测模式。编码帧内预测模式要被用于对图像块(即,当前块)进行编码。

[0192] 在一示例中,编码帧内预测模式可以从ML模型获得的。即,ML帧内预测模式(即,ML模型的输出)可以是编码帧内预测模式。在另一个示例中,并且如上所述,在1104中,过程1100也可以使用决策矩阵。即,过程1100基于决策矩阵在最可能的帧内预测模式和ML帧内预测模式之间选择。

[0193] 在1108中,过程1100使用编码帧内预测模式对图像块进行编码。即,例如,并且如关于图4的帧内/帧间预测阶段402所描述的,可以使用编码帧内预测模式来获得预测块,然后,可以获得残差块,并且与图4的描述一致,最终将残差块熵编码在压缩比特流(诸如图4的比特流420)中,如关于熵编码阶段408所描述的。

[0194] 图15是根据本公开的实施方式并且与上面的描述一致的,用于使用帧内预测对当前块进行编码的过程1500的流程图。在编码器确定为当前块选择帧内预测模式之后,可以由编码器执行(即,进行)过程1500。过程1500使用机器学习(ML)模型和决策矩阵来选择(例如,获得、确定、推断等)用于对当前块进行编码的帧内预测模式。当前块可以是视频流的图像(例如,帧)的块。当前块可以是亮度块,诸如图6的亮度块660。当前块可以是色度块,诸如图6的U或Cb色度块670或V或Cr色度块680。当前块可以是超级块、宏块、任何大小为块的最大代码化单元或其子块。例如,图像可以是 128×128 亮度块、 64×64 色度块、 4×4 亮度块或一些其他大小的块。当前块可以是任何块,该块可以由执行过程1500的编码器使用帧内预测进行的预测。

[0195] 例如,过程1500可以实施为软件程序,该软件程序可以由计算设备(诸如,发送站102)执行。例如,软件程序可以包括机器可读指令,这些指令可以存储在存储器(诸如,存储器204或辅助储存装置214)中,并且在由处理器(诸如,处理器202)执行时可以使计算设备执行过程1500。过程1500可以使用专用硬件或固件来实施。例如,硬件组件可以配置为执行过程1500。如上所述,一些计算设备可以具有多个存储器或处理器,并且过程1500中所描述的操作可以使用多个处理器、存储器或两者来分布。在一示例中,过程1500可以作为帧内/帧间预测阶段(诸如图4的帧内/帧间预测阶段402)的部分执行。

[0196] 在1502中,过程1500使用机器学习(ML)模型获得ML帧内预测模式。ML模型可以是如上面关于图11所描述的那样进行训练的模式。因此,在一示例中,ML模型可以使用输入数据进行训练,以对当前块输出ML帧内预测模式。还如上所述,输入数据的输入单个数据可以包括输入视频块和用于对输入视频块进行编码的相应地面实况帧内预测模式。地面实况帧内预测模式是由使用蛮力方法进行模式预测的编码器选择的帧内预测模式。

[0197] 如上所述,可以将预先计算的特征输入到ML模型。因此,在一示例中,过程1500可以计算当前块的像素值的平均值和方差中的至少一个。平均值或方差中的至少一个可以用作ML模型的输入。

[0198] 在一示例中,预先计算的特征可以包括一个或多个基于方向的特征。每个基于方向的特征都可以是通过将对应的内核应用于当前块来获得卷积块,从卷积块中获得标量并且使用标量作为ML模型的输入来获得的。在一示例中,标量可以是卷积块的绝对值之和。还如上所述,在一示例中,滤波器可以是Sobel滤波器,该Sobel滤波器对应于与可用帧内预测模式相关联的方向。可用帧内预测模式是编码器可以用于对块进行帧内预测的可能的帧内预测模式之一。还如上所述,在一示例中,滤波器权值不是固定的。更确切地说,在一示例中,滤波器权值可以在ML模型的训练阶段学习。因此,卷积块可以通过使用当前块作为一个或多个卷积层的输入来获得。

[0199] 还如上所述,在一示例中,量化参数的非线性函数可以用作输入到ML模型的预先计算的特征。量化参数(即,量化参数的值)可以由编码器选择。

[0200] 还如上所述,在一示例中,与可用帧内预测模式的至少子集相关联的预先计算的

成本可以作为输入用于ML模型。在典型的视频编码器中,这种预先计算的成本在编码器中通常已经是可用的,因此这种预先计算的成本可以直接用作ML输入。即,过程1500可以计算与可用帧内预测模式的至少子集相关联的成本,并且可以将成本馈送给ML模型。因此,过程1500可以为可用帧内预测模式的一些模式计算对应的预测块,计算对应的预测块和当前块之间的对应误差,并且将对应误差作为输入用于ML模型。

[0201] 在1504中,过程1500从用于对当前块进行编码的可用帧内预测模式中获得最可能的帧内预测模式。在一示例中,最可能的帧内预测模式可以由编码器基于相邻块的帧内预测模式来选择。在一示例中,最可能的帧内预测模式可以是与上面所描述的预先计算的成本中的最小成本相对应的帧内预测模式。即,最可能的帧内预测模式可以是基于SAD的内部模式决策。因此,无论预先计算的成本是否用作ML模型的输入,过程1500都可以使用预先计算的成本来选择最可能的帧内预测模式。

[0202] 在1506中,过程1500选择ML帧内预测模式或最可能的帧内预测模式中的一个作为编码帧内预测模式。过程1500基于ML帧内预测模式和最可能的帧内预测模式的相对可靠性来选择编码帧内预测模式。

[0203] 在ML帧内预测模式或最可能的帧内预测模式中的一个是最优帧内预测模式的更好预测者的情况下,ML帧内预测模式或最可能的帧内预测模式中的一个比ML帧内预测模式或最可能的帧内预测模式中的另一个更可靠。最优帧内预测模式是执行最佳模式搜索的编码器所选择的模式。在一示例中,上面可以关于图8描述最佳模型。

[0204] 在一示例中,过程1500使用决策矩阵来选择编码帧内预测模式。在一示例中,并且如上所述,决策矩阵在训练阶段期间可以使用统计来生成,该统计将ML模型和编码器的对应性能进行比较,该编码器选择与地面实况帧内预测模式相比最可能的帧内预测模式。决策矩阵指示是ML帧内预测模式还是最可能的帧内预测模式被选择作为编码帧内预测模式。

[0205] 在一示例中,并且在选择了ML帧内预测模式的情况下,并且如上所述,过程1500可以减少与ML帧内预测模式相关联的成本。因此,选择ML帧内预测模式可以包括计算与ML帧内预测模式相关联的第一成本和将第一成本减少到第二成本。在一示例中,计算第一成本可能已经作为预先计算的成本的部分完成。

[0206] 在1508中,过程1500将编码帧内预测模式编码在压缩比特流中。压缩比特流可以是图4的压缩比特流420。

[0207] 在1510中,过程1500使用编码帧内预测模式对当前块进行编码。例如,过程1500可以执行以下一项或多项:生成预测块,生成预测块、生成残差块,对残差块进行变换以获得变换块,将变换块量化和编码在压缩比特流中,如上面关于图4所描述的。当对当前块进行解码时,解码器可以使用(即,解码)编码帧内预测模式,解码器可以使用该编码帧内预测模式来重构当前块,如上面关于图4所描述的。

[0208] 在一示例中,编码器可以使用前 $N (> 1)$ 个预测模式来估计和比较其最后的比特率。因此,在一示例中,本文中所描述的ML模型可以用于根据其对应的预测值获取(例如,输出、推断等) N 个最可能的模式。然后,这前 N 个模式可以与SAD获取的前 N 个模式结合(例如,融合,如本文中所述),以获得前 N 个模式的最终列表。然后, N 个模式之一可以例如基于其对当前块进行编码的对应成本来选择。

[0209] 因此,在一实施方式中, N 个ML帧内预测模式可以使用ML模型来获得; N 个最可能的

帧内预测模式(例如,基于SAD的最可能的帧内预测模式)可以从可用帧内预测模式中获得;N个ML帧内预测模式之一或N个最可能的帧内预测模式之一可以作为编码帧内预测模式被选择;编码帧内预测模式被编码在压缩比特流中;并且当前块然后可以使用编码帧内预测模式来进行编码。

[0210] 图16是根据本公开的实施方式并且与上面的描述一致的用于使用帧内预测对当前块进行编码的过程1600的流程图。在编码器确定为当前块选择帧内预测模式之后,可以由编码器执行(即,进行)过程1600。过程1600使用机器学习(ML)模型和决策矩阵来选择(例如,获得、确定、推断等)用于对当前块进行编码的帧内预测模式。当前块可以是视频流的图像(例如,帧)的块。当前块可以是亮度块,诸如图6的亮度块660。当前块可以是色度块,诸如图6的U或Cb色度块670或V或Cr色度块680。当前块可以是超级块、宏块、任何大小为块的最大代码化单元或其子块。例如,图像可以是 128×128 亮度块、 64×64 色度块、 4×4 亮度块或一些其他大小的块。当前块可以是任何能够由执行过程1600的编码器进行帧内预测的块。

[0211] 例如,过程1600可以实施为软件程序,该软件程序可以由计算设备(诸如,发送站102)执行。例如,软件程序可以包括机器可读指令,这些指令可以存储在存储器(诸如,存储器204或辅助储存装置214)中,并且在由处理器(诸如,处理器202)执行时可以使计算设备执行过程1600。过程1600可以使用专用硬件或固件来实施。例如,硬件组件可以配置为执行过程1600。如上所述,一些计算设备可以具有多个存储器或处理器,并且过程1600中所描述的操作可以使用多个处理器、存储器或两者进行分配。在一示例中,过程1600可以作为帧内/帧间预测阶段(诸如图4的帧内/帧间预测阶段402)的部分执行。

[0212] 在1602中,过程1600获得预先计算的特征。获得预先计算的特征可以指计算、选择、确定或以任何方式获得。预先计算的特征可以包括以下两项或多项:第一特征,该第一特征是量化参数的非线性函数;第二特征,该第二特征是当前块和对应的预测块之间的对应误差,其中,每个预测块都对应于可用帧内预测模式;当前块的平均值和方差;以及从当前块获得的卷积块的绝对值之和。每个预先计算的特征都可以如上所述。

[0213] 在一示例中,卷积图像的绝对值之和是通过将Sobel滤波器应用于当前块来获得卷积块,从当前块中获得的。在一示例中,卷积图像的绝对值之和是通过步骤从当前块获得的,这些步骤包括使用卷积操作获得卷积块。

[0214] 在一示例中,预先计算的特征可以包括以下全部项:第一特征、第二特征、平均值和方差以及卷积块的绝对值之和。

[0215] 在1604中,过程1600使用机器学习(ML)模型获得ML帧内预测模式。ML模型可以接收预先计算的特征作为输入。在训练阶段,ML模型可以使用输入数据进行训练,以输出当前块的ML帧内预测模式。输入数据(即,训练数据)的每个输入单个数据都可以包括输入视频块和用于对输入视频块进行编码的相应地面实况帧内预测模式。输入单个数据还可以包括相应的QP值,该值用于使用地面实况帧内预测模式对输入视频块进行编码。如上所述,QP值的非线性函数可以用于输入单个数据。

[0216] 在1606中,过程1600至少使用ML帧内预测模式来选择编码帧内预测模式。在一示例中,由ML模型输出的ML帧内预测模式可以是用于对当前块进行编码的编码帧内预测模式。

[0217] 在一示例中,至少使用ML帧内预测模式来选择编码帧内预测模式可以包括:从用

于对当前块进行编码的可用帧内预测模式中获得最可能的帧内预测模式;以及使用决策矩阵来选择编码帧内预测模式。编码帧内预测模式可以是ML帧内预测模式或最可能的帧内预测模式中的一个。最可能的帧内预测模式可以如上所述。在一示例中,决策矩阵在训练阶段可以使用统计来生成,该统计将ML模型和编码器的对应性能进行比较,该编码器选择与地面实况帧内预测模式相比最可能的帧内预测模式。决策矩阵指示是ML帧内预测模式还是最可能的帧内预测模式被选择作为编码帧内预测模式。

[0218] 在1608中,过程1600将编码帧内预测模式编码在压缩比特流中。压缩比特流可以是图4的压缩比特流420。在1610中,过程1600使用编码帧内预测模式对当前块进行编码。例如,过程1600可以执行以下一项或多项:生成预测块,生成残差块,对残差块进行变换以获得变换块,将变换块量化和编码在压缩比特流中,如上面关于图4所描述的。当对当前块进行解码时,解码器可以使用(即,解码)编码帧内预测模式,解码器可以使用该编码帧内预测模式来重构当前块,如上面关于图4所描述的。

[0219] 图17是率失真性能比较的示例1700。示例1700说明了使用多种类型来验证根据本公开的实施方式的有效性的实验的结果。将使用ML模型来推断要在VP9硬件编码器中仅用于关键帧编码的预测模式(表示为valid_mode 1712)所产生的率失真性能与使用最可能的帧内预测模式(表示为valid_vanil 1714)所产生的率失真性能进行比较。ML模型只包括1600个参数。小的模型大小使得在功率和/或容量受约束的平台(诸如移动设备)上进行推断成为可能。

[0220] 在该实验中,PSNR(峰值信噪比)用作失真矩阵。峰值信噪比(PSNR)用作失真矩阵。即,在图表1710和1720中,x轴指示对样本视频序列进行编码的数据速率,单位为kbps,而y轴指示PSNR质量,单位为分贝(dB)。

[0221] 图表1710说明实现了一致的性能增强。平均而言,当使用ML预测完全取代基于SAD的方法(例如,最可能的帧内预测模式)时,可以实现更高的率失真性能。BD-rate的性能大约比基于SAD的方法好0.9%。

[0222] 图表1720示出了根据上述自适应融合策略合并ML决策和基于SAD的决策时的率失真性能。当结合这两种决策时,平均而言,BD-rate(即,曲线1724)的性能大约比基于SAD的方法(例如,曲线1712)好1.2%,并且比只有ML的方法好0.3%。这表明了根据ML预测质量自适应地结合ML决策和基于SAD的决策的有效性。

[0223] 所使用的ML模型只有1600个参数。小的ML模型大小使得在功率/容量受约束的移动平台上进行推断成为可能。

[0224] 如上面关于图11的1108、图15的1508或图16的1608所描述的,使用机器学习模型(诸如,关于图12A所描述的ML模型或关于图12B所描述的ML模型)来推断图像块的模式决策参数的编码器可以将模式决策参数(即,编码帧内预测模式)编码在压缩比特流(诸如图4的比特流420)中。

[0225] 因此,解码器,诸如图5的解码器500,可以使用在压缩比特流中接收到的模式决策参数对图像块进行解码。

[0226] 因此,对图像块进行解码的过程可以包括:在压缩比特流(诸如图5的压缩比特流420)中接收编码帧内预测模式(例如,指示编码帧内预测模式的编码索引或符号);以及使用该编码帧内预测模式对图像块进行解码。

[0227] 如上所述,如上所述配置的ML模型可以由编码器(诸如图4的编码器400)用来推断帧内预测模式。因此,帧内预测模式并不是通过本领域中已知的蛮力方法获取出来的。在一示例中,ML模型可以由帧内/帧间预测阶段402使用。

[0228] 为了便于说明,过程800、1100、1500和1600分别被描绘和描述为一系列框、步骤或操作。然而,根据本公开的框、步骤或操作可以按照各种顺序和/或同时发生。附加地,可以使用本文中未呈现和描述的其他步骤或操作。此外,可能并不需要所有所示的步骤或操作来实施根据所公开的主题的技术。

[0229] 上述编码和解码方面说明了一些编码和解码技术。然而,要理解,“编码”和“解码”,如在权利要求书中所使用的那些术语,可以指对数据的压缩、解压缩、变换或任何其他处理或变更。

[0230] 词语“示例”或“实施方式”在本文中被用来指充当示例、实例或说明。在本文中被描述为“示例”或“实施方式”的任何方面或设计都不一定被视为比其他方面或设计优选或有利。确切地说,词语“示例”或“实施方式”的使用旨在以具体的方式呈现概念。如本申请中所使用的,术语“或”旨在指包括性的“或”,而不是排他性的“或”。即,除非上下文另有说明或明确说明,否则“X包括A或B”旨在指其任何自然的包括性排列。即,如果X包括A,X包括B或X包括A和B,则在任何上述情况下均满足“X包括A或B”。另外,本申请和所附权利要求中所使用的冠词“一”和“一个”通常应被视为指“一个或多个”,除非上下文另有明确规定其指的是单数形式。此外,除非如此描述,否则术语“实施方式”或“一种实施方式”并不旨在自始至终都指相同的实施例或实施方式。

[0231] 发送站102和/或接收站106的实施方式(以及存储在其上和/或由其执行的算法、方法、指令等,包括通过编码器400和解码器500)可以在硬件、软件或者其任何组合中实现。例如,硬件可以包括计算机、知识产权(IP)内核、专用集成电路(ASIC)、可编程逻辑阵列、光学处理器、可编程逻辑控制器、微代码、微控制器、服务器、微处理器、数字信号处理器或任何其他合适的电路。在权利要求书中,术语“处理器”应被理解为包括上述任何硬件,无论是单独地还是组合地。术语“信号”和“数据”可互换使用。进一步地,发送站102和接收站106的部分不一定以相同的方式实施。

[0232] 进一步地,在一个方面中,例如,发送站102或接收站106可以使用具有计算机程序的通用计算机或通用处理器来实施,该计算机程序在被执行时执行本文中所描述任何一种对应的方法、算法和/或指令。另外或备选地,例如,可以利用专用计算机/处理器,该专用计算机/处理器可以包含用于执行本文中所描述的任何方法、算法或指令的其他硬件。

[0233] 例如,发送站102和接收器106可以在视频会议系统中的计算机上实施。备选地,发送站102可以在服务器上实施,而接收站106可以在与服务器分离的设备(诸如,手持通信设备)上实施。在这种情况下,使用编码器400,发送站102可以将内容编码为编码视频信号,并且将编码视频信号发送给通信设备。反过来,通信设备然后可以使用解码器500对编码视频信号进行解码。备选地,通信设备可以对本地存储在通信设备上的内容(例如,不由发送站102发送的内容)进行解码。其他发送站102和接收站106实施方式是可用的。例如,接收站106可以是大体上固定的个人计算机,而不是便携式通信设备,和/或包括编码器400的设备还可以包括解码器500。

[0234] 进一步地,本公开的所有或部分实施方式可以采取计算机程序产品的形式,该计

计算机程序产品从例如有形的计算机可用或计算机可读介质是可访问的。计算机可用或计算机可读介质可以是任何设备,例如,该设备可以有形地包含、存储、传递或传输程序以供任何处理器使用或与其结合使用。例如,介质可以是电子、磁性、光学、电磁或半导体设备。其他合适的介质也是可用的。

[0235] 为了允许便于理解本公开并且不限制本公开,已经描述了上述实施例、实施方式和方面。相反,本公开旨在涵盖包括在所附权利要求书的范围内的各种修改和等同布置,该范围应被赋予最广泛的解释来涵盖法律允许的所有这样的修改和等同结构。

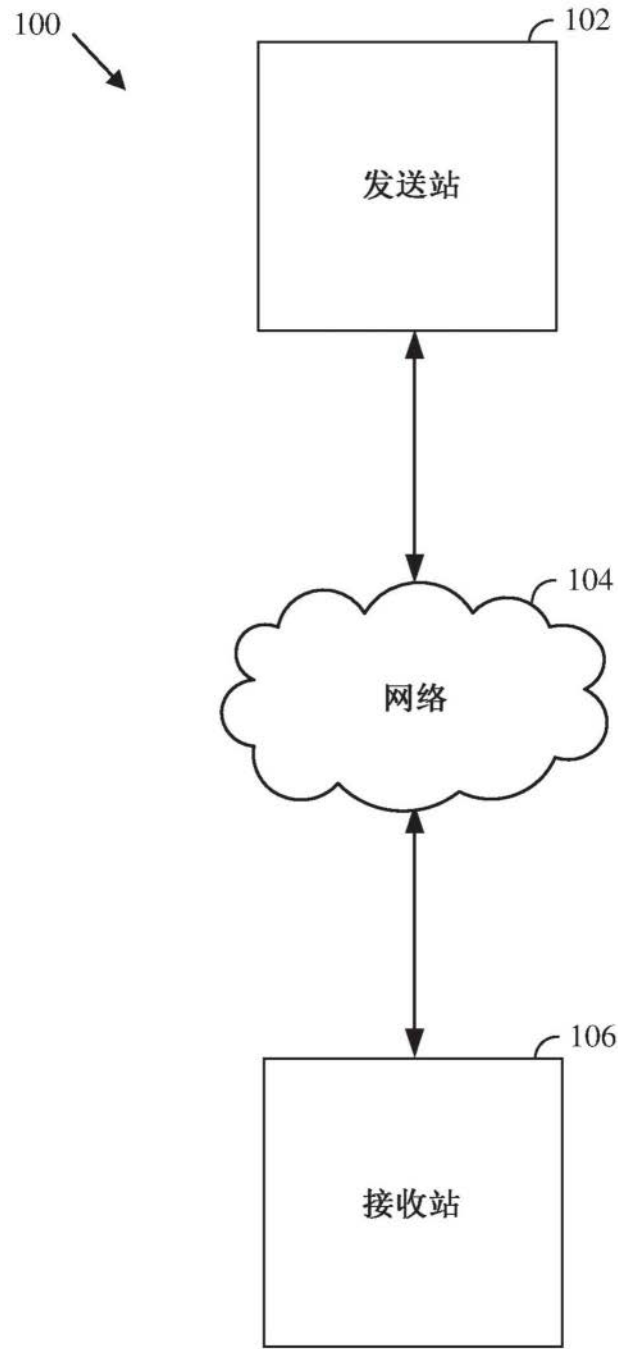


图1

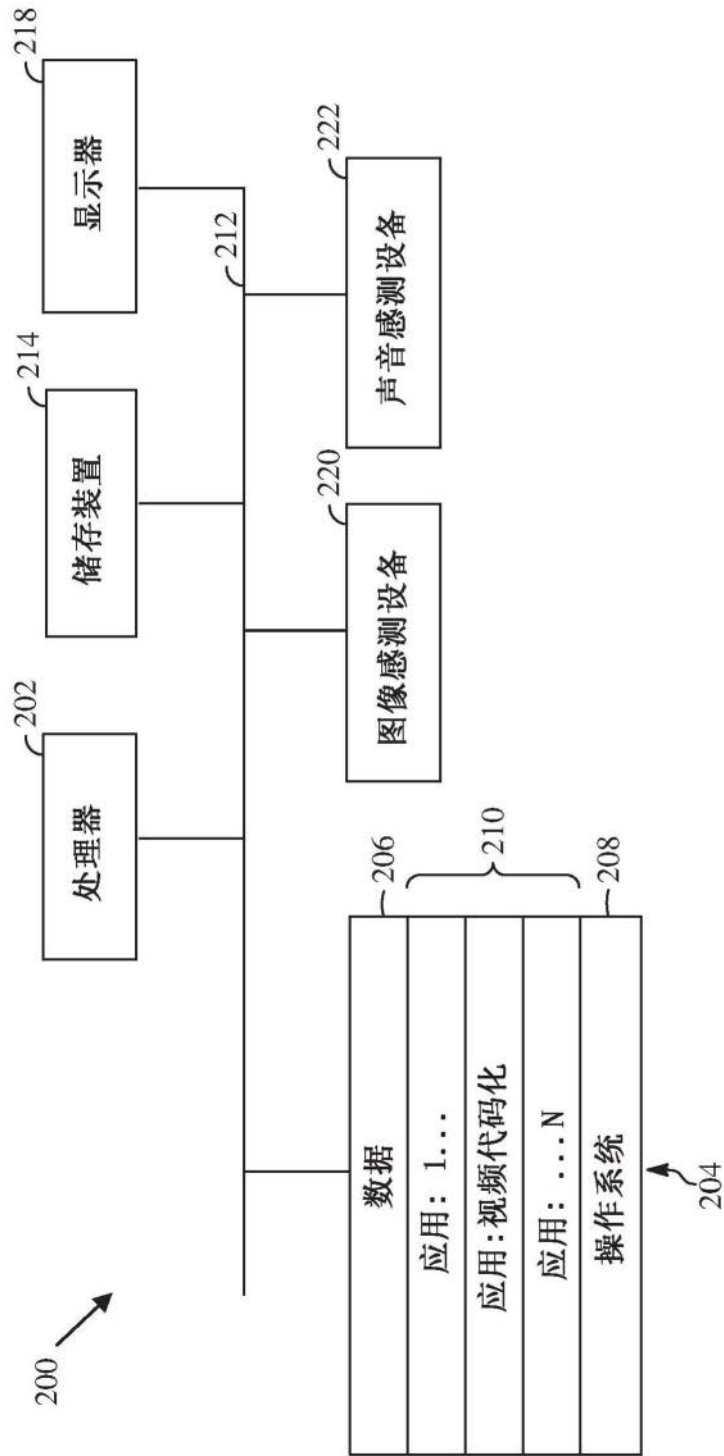


图2

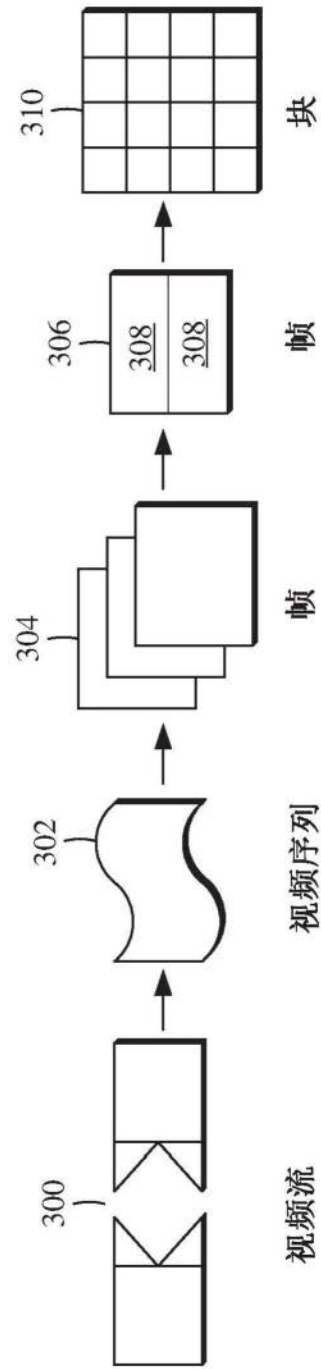


图3

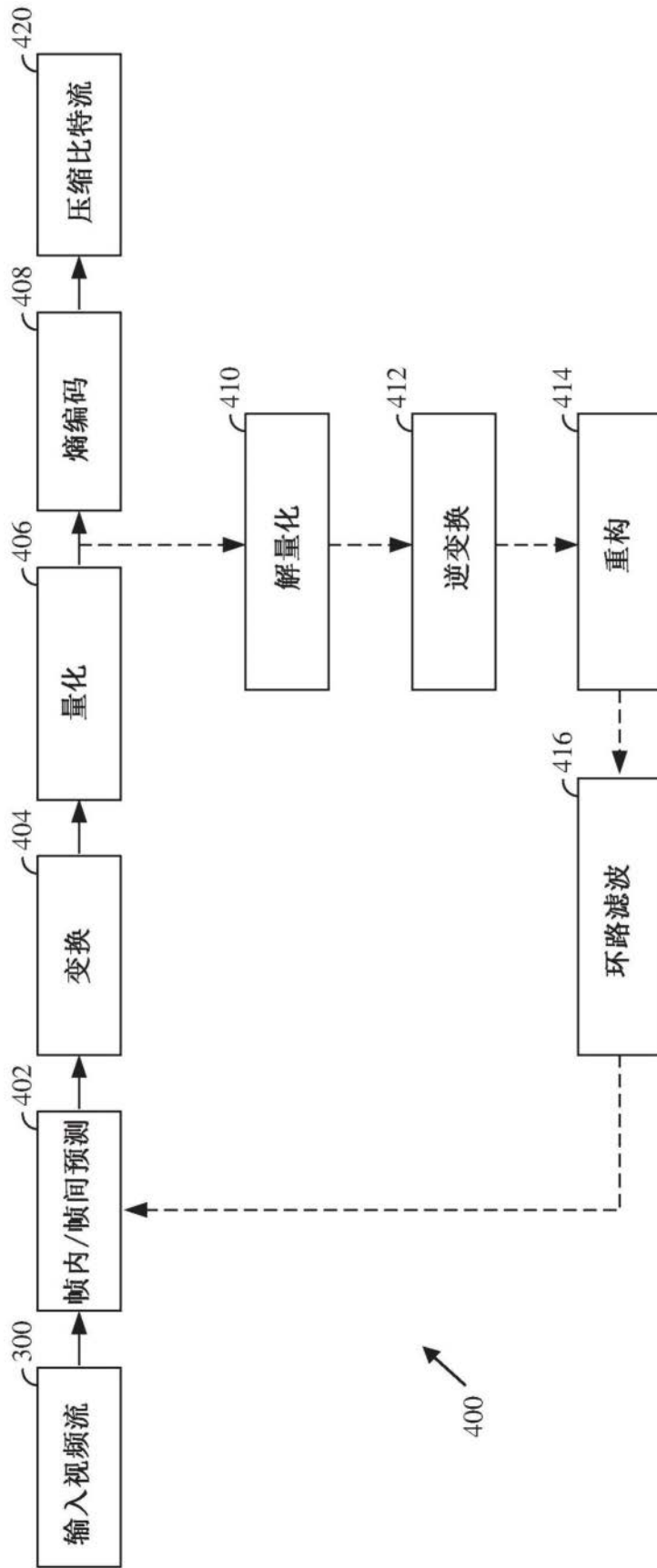


图4

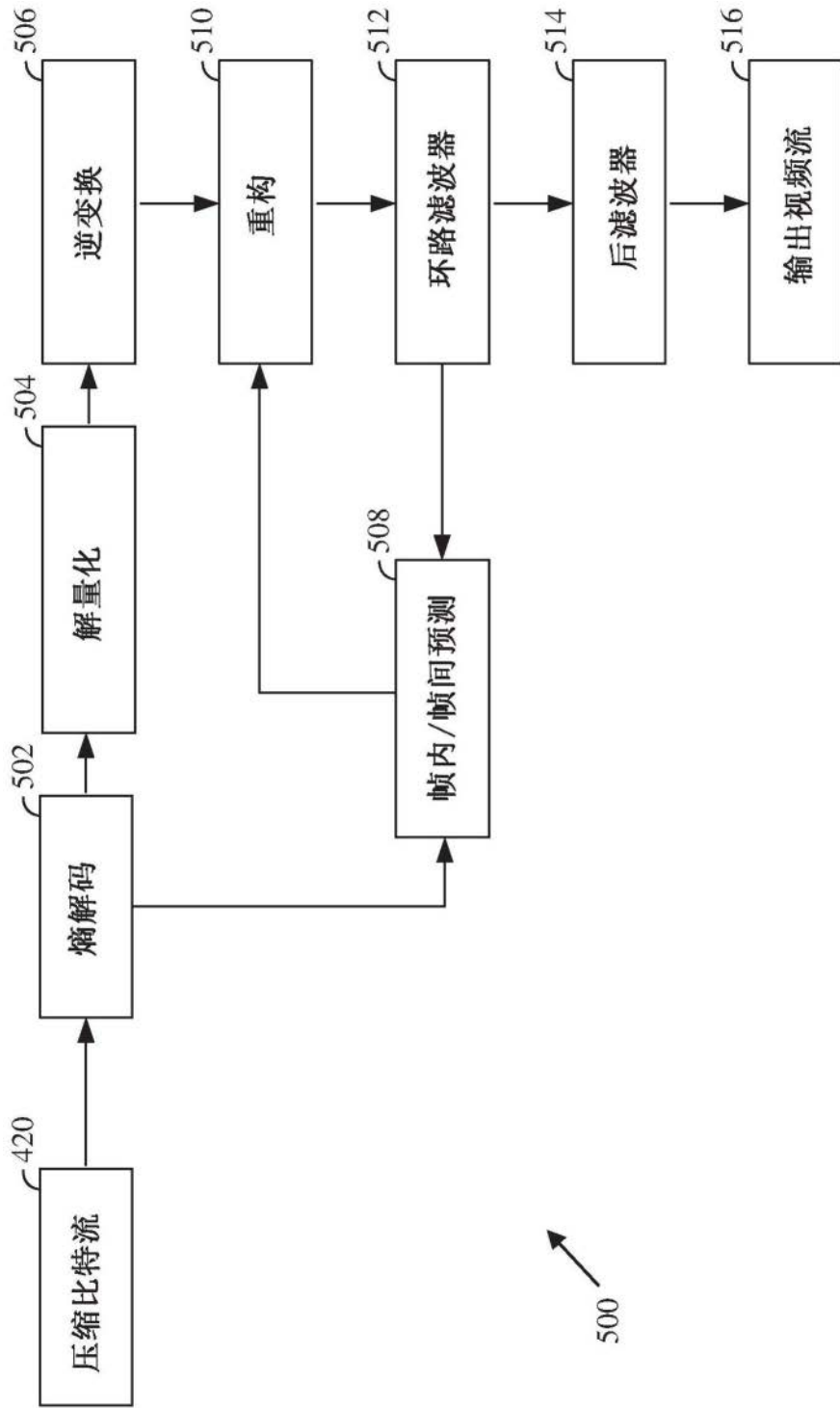


图5

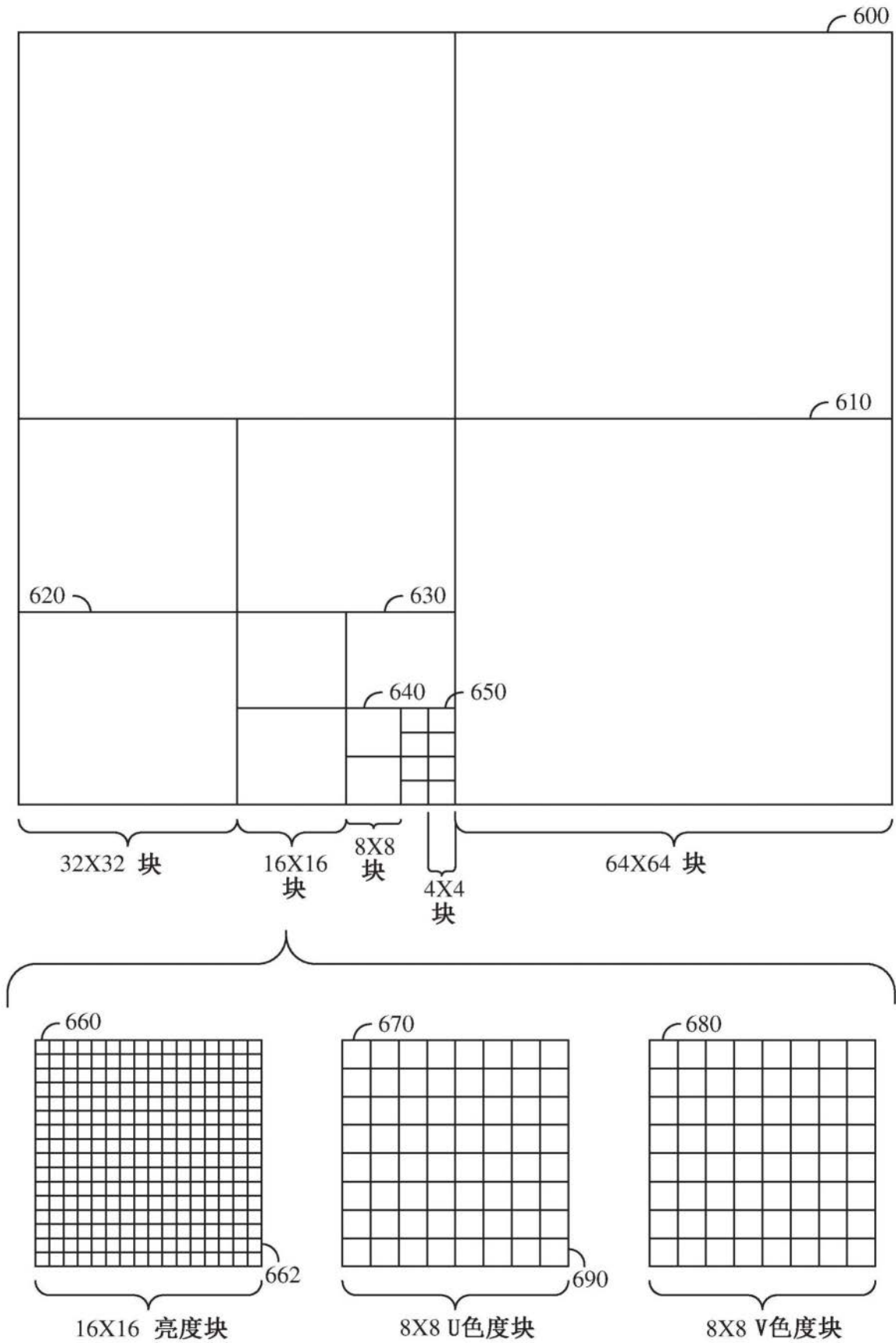


图6

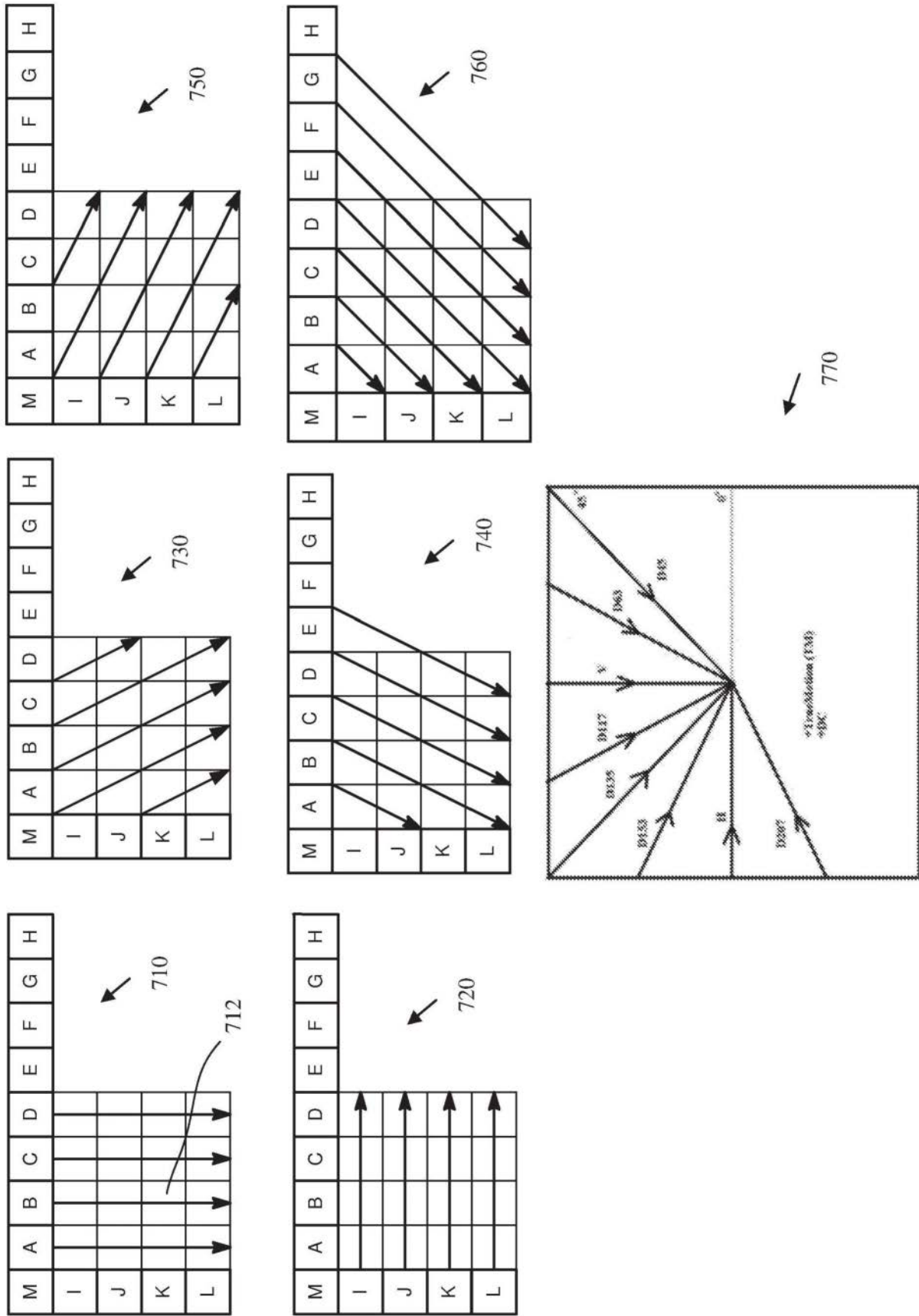


图7

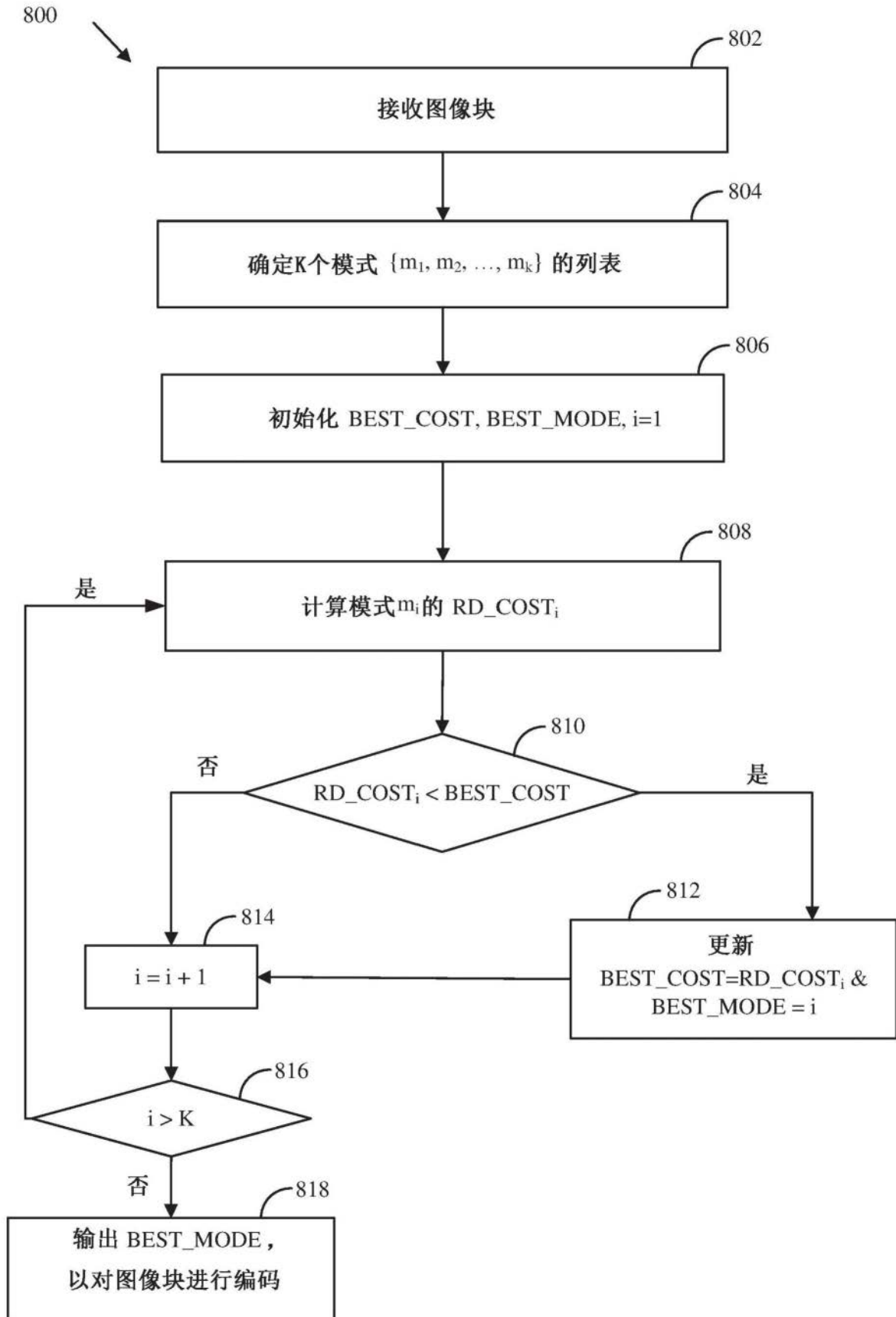


图8

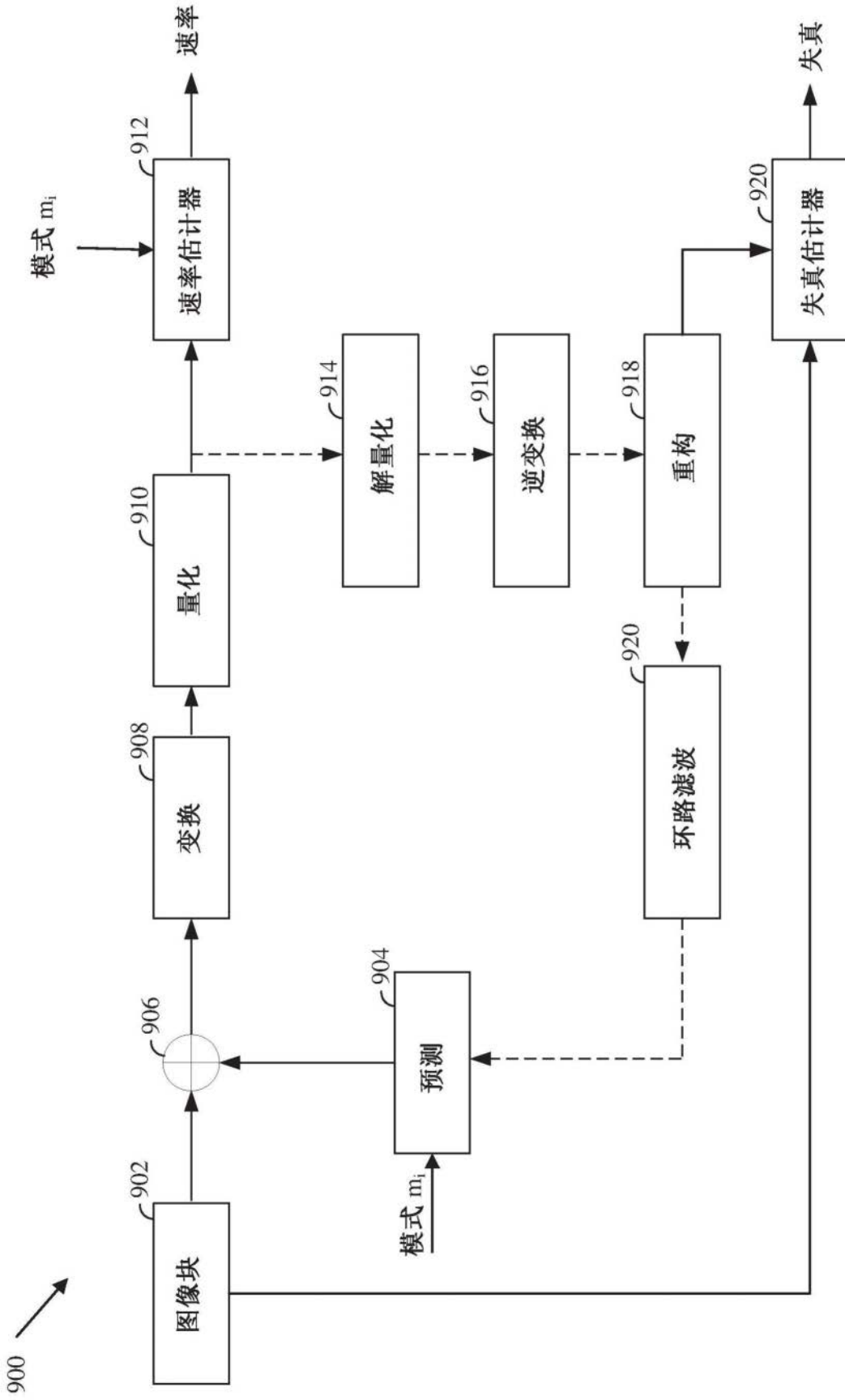
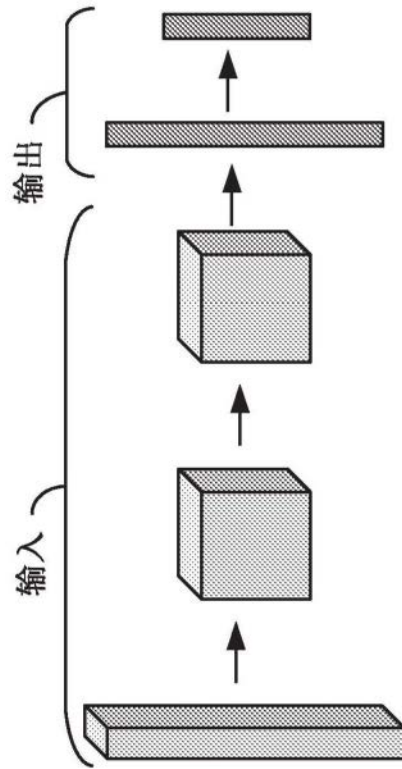


图9



1000 ↗

图10A

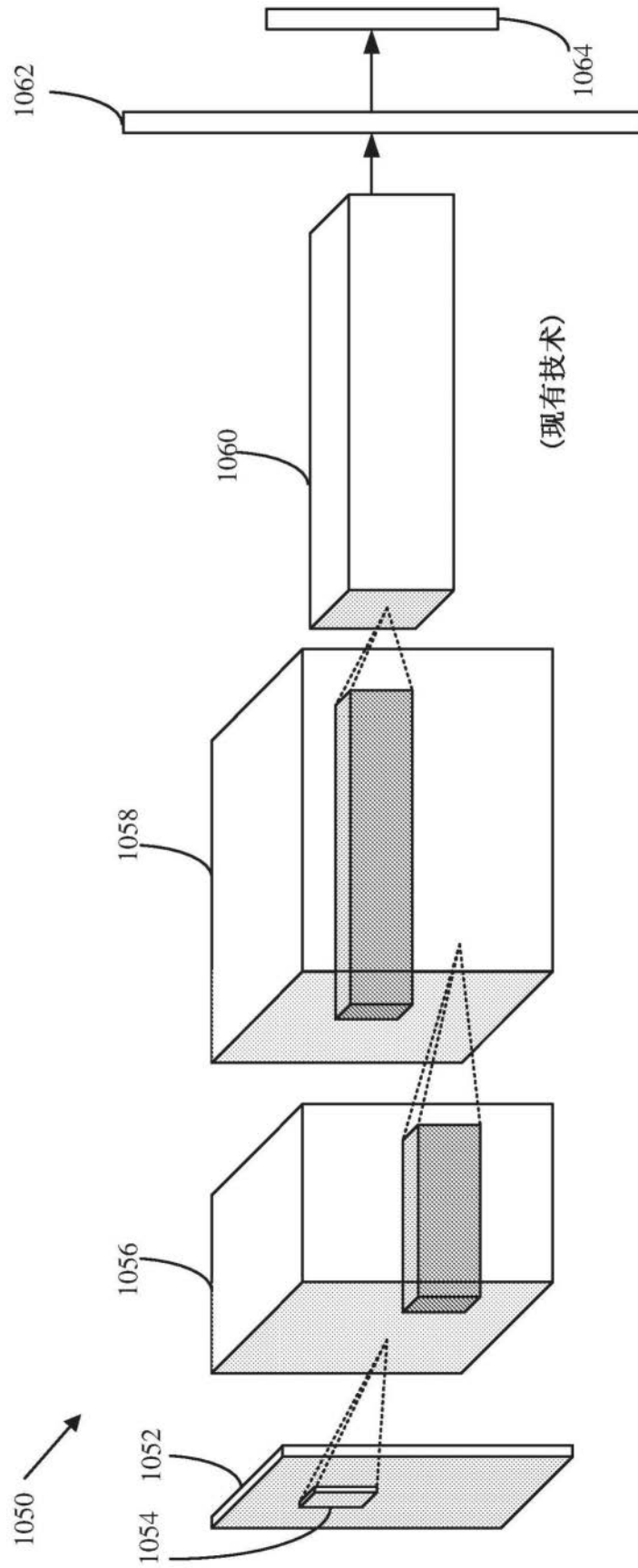


图10B

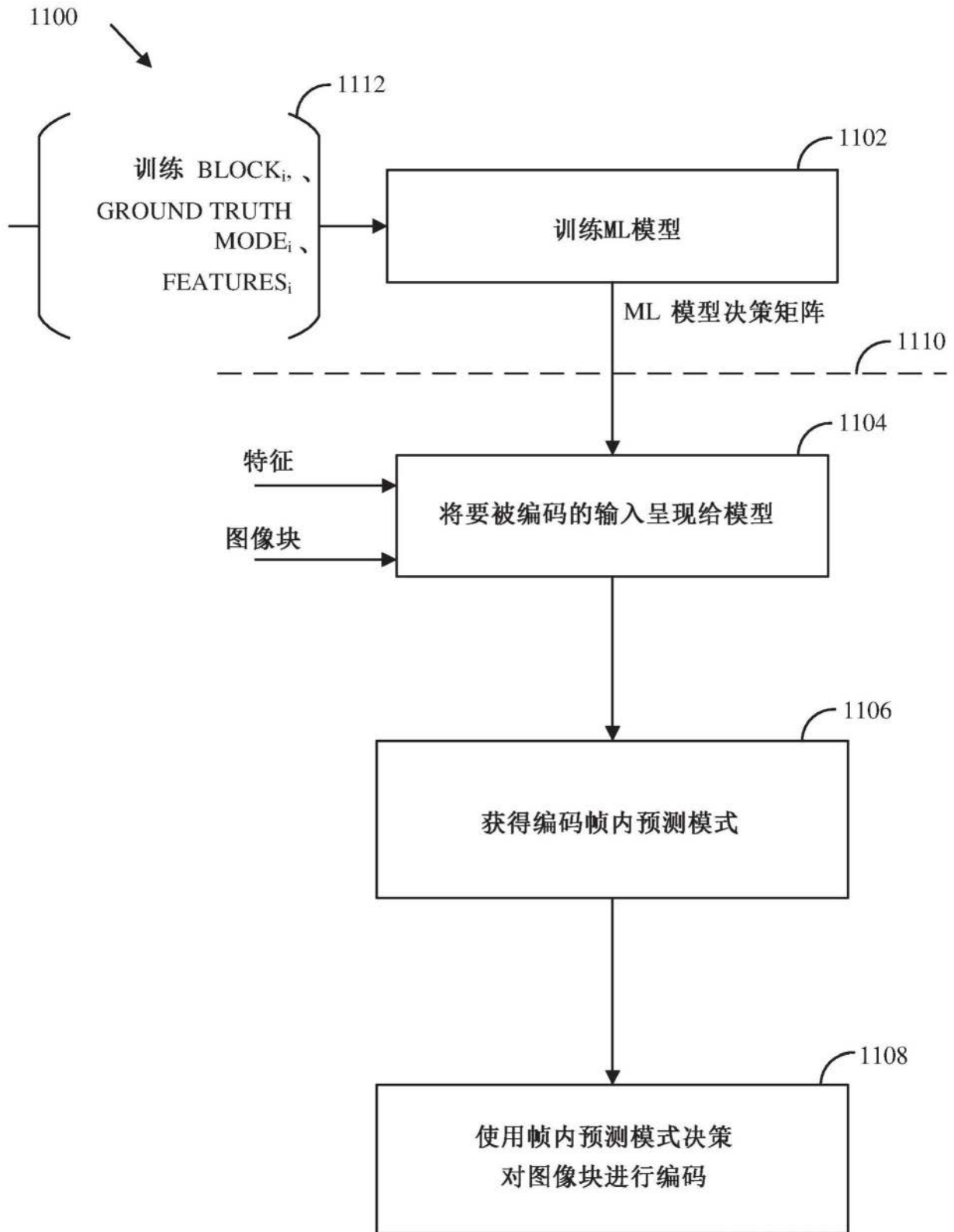


图11

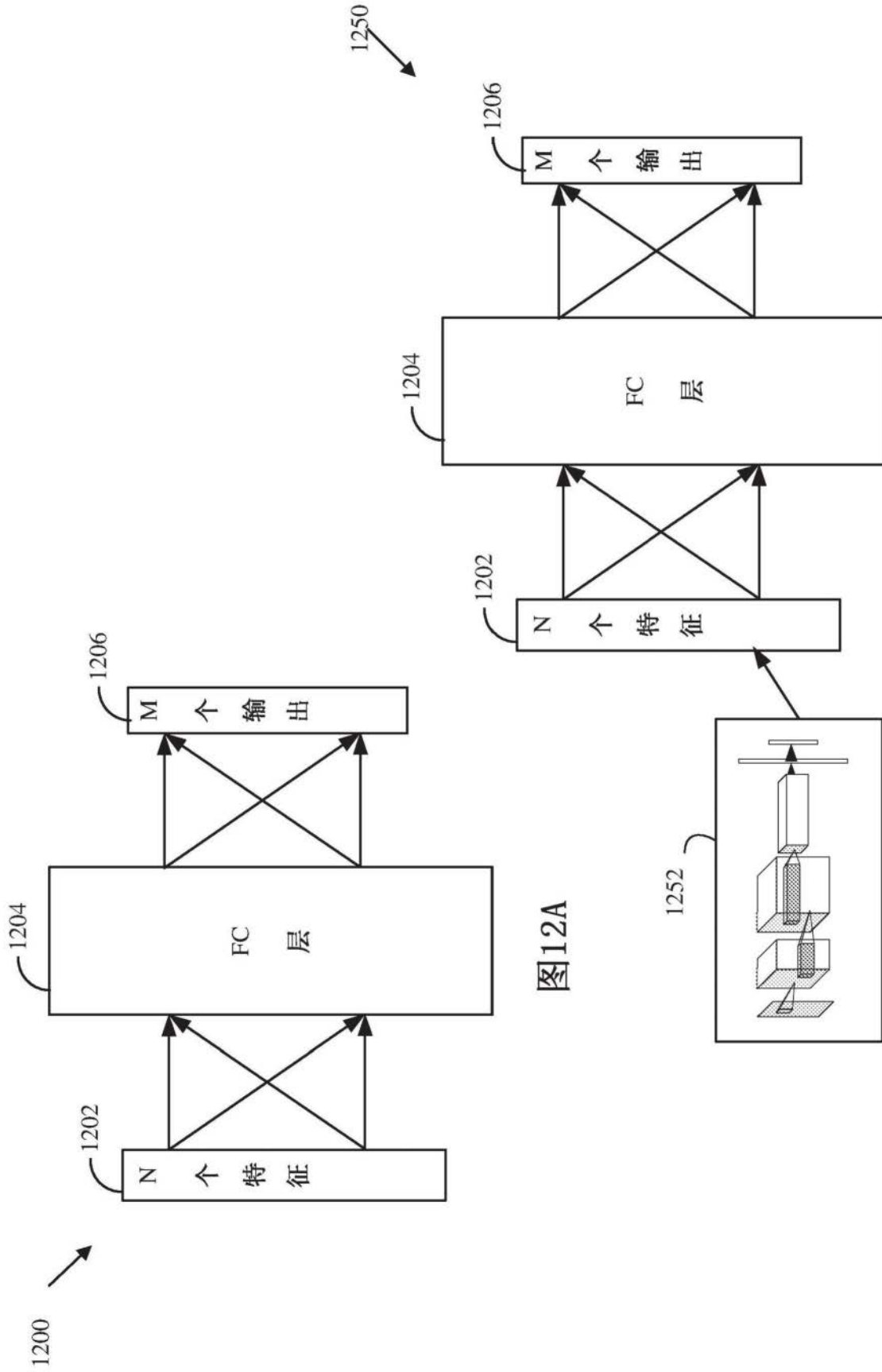


图12A

图12B

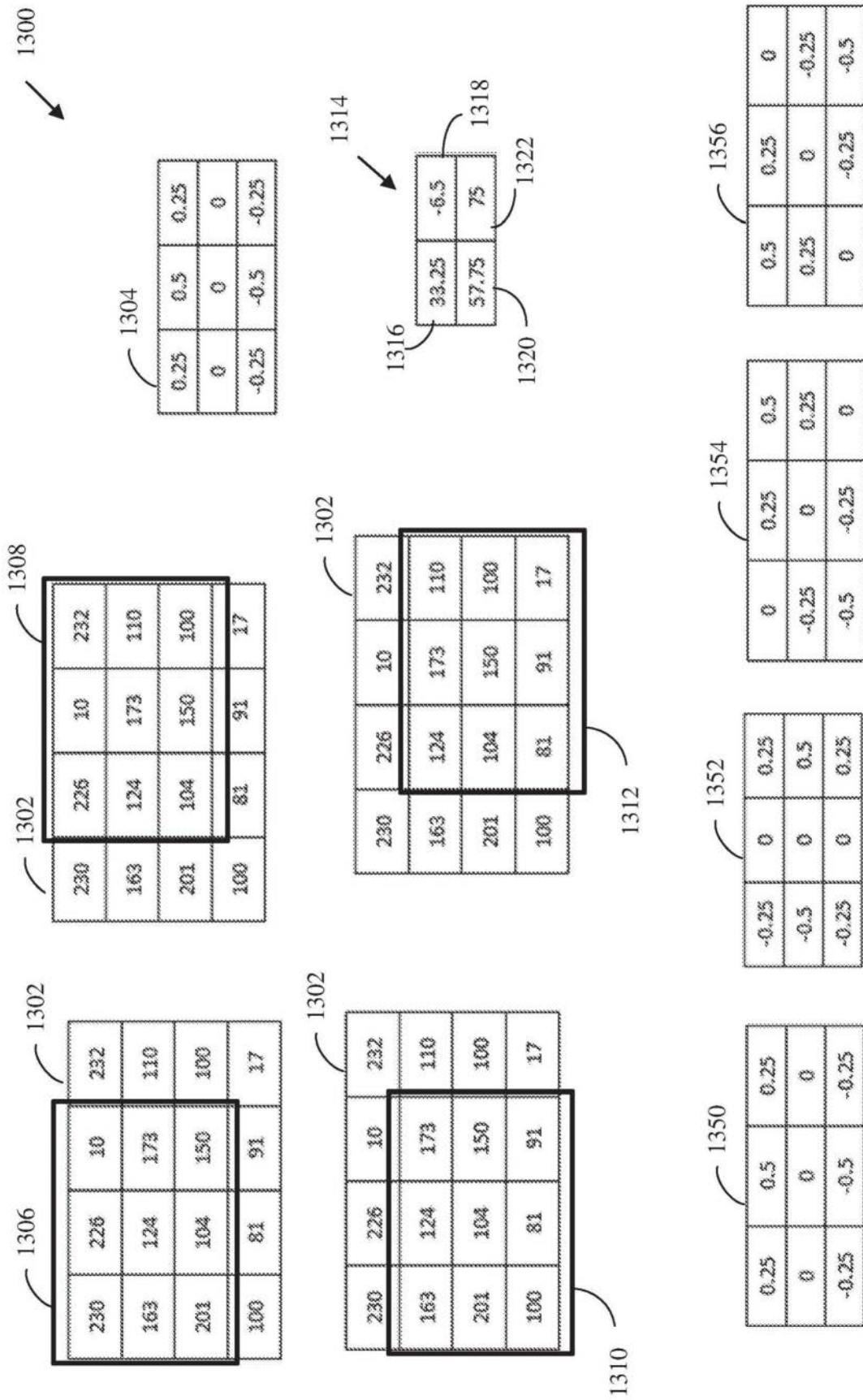


图13

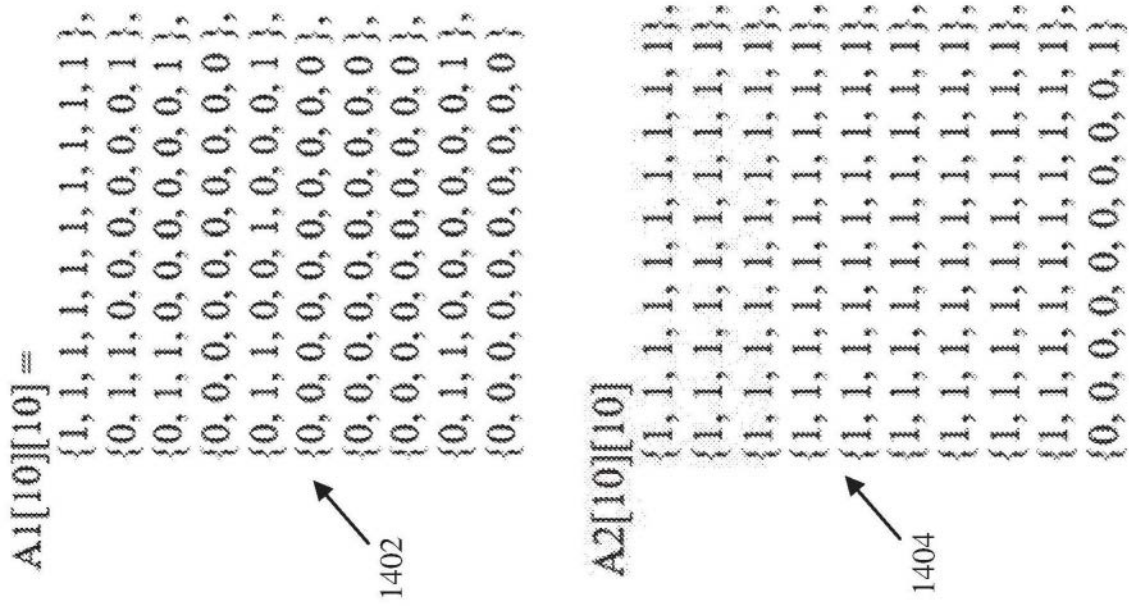


图14

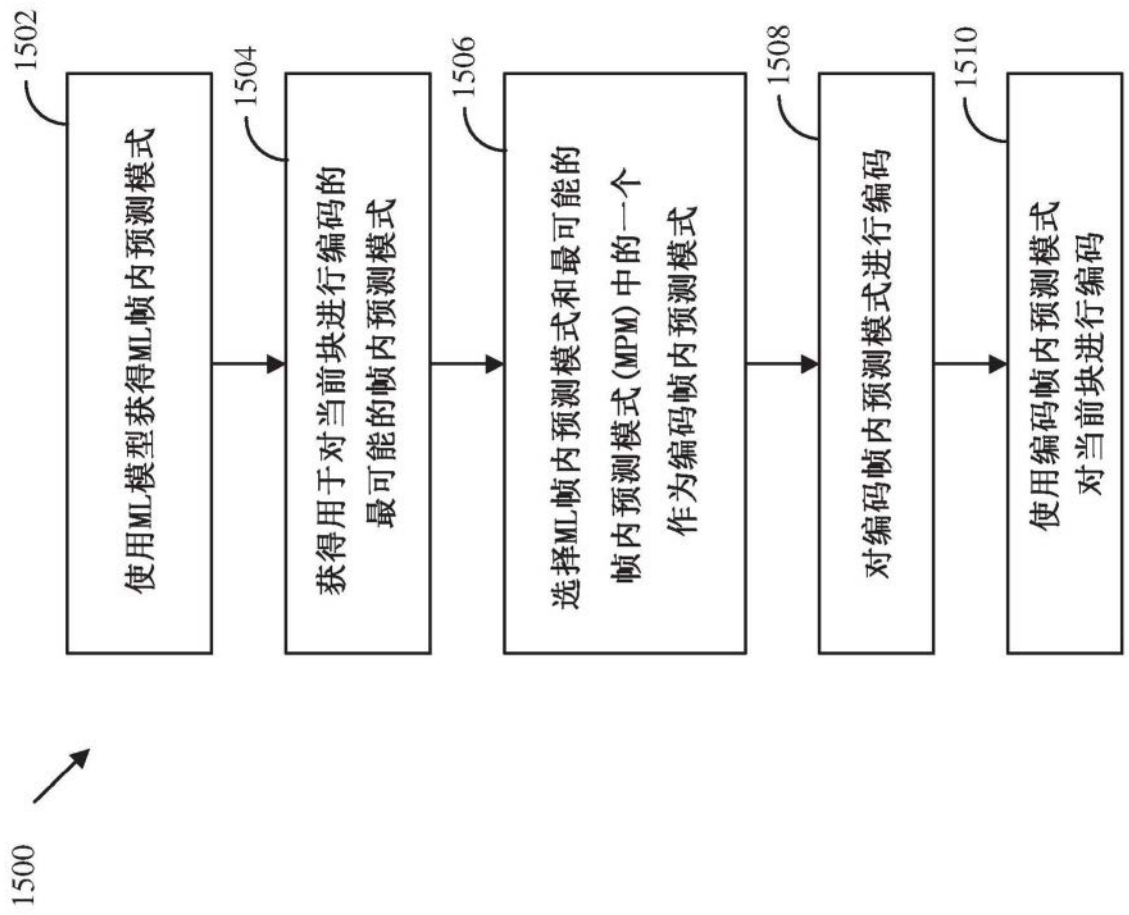


图15

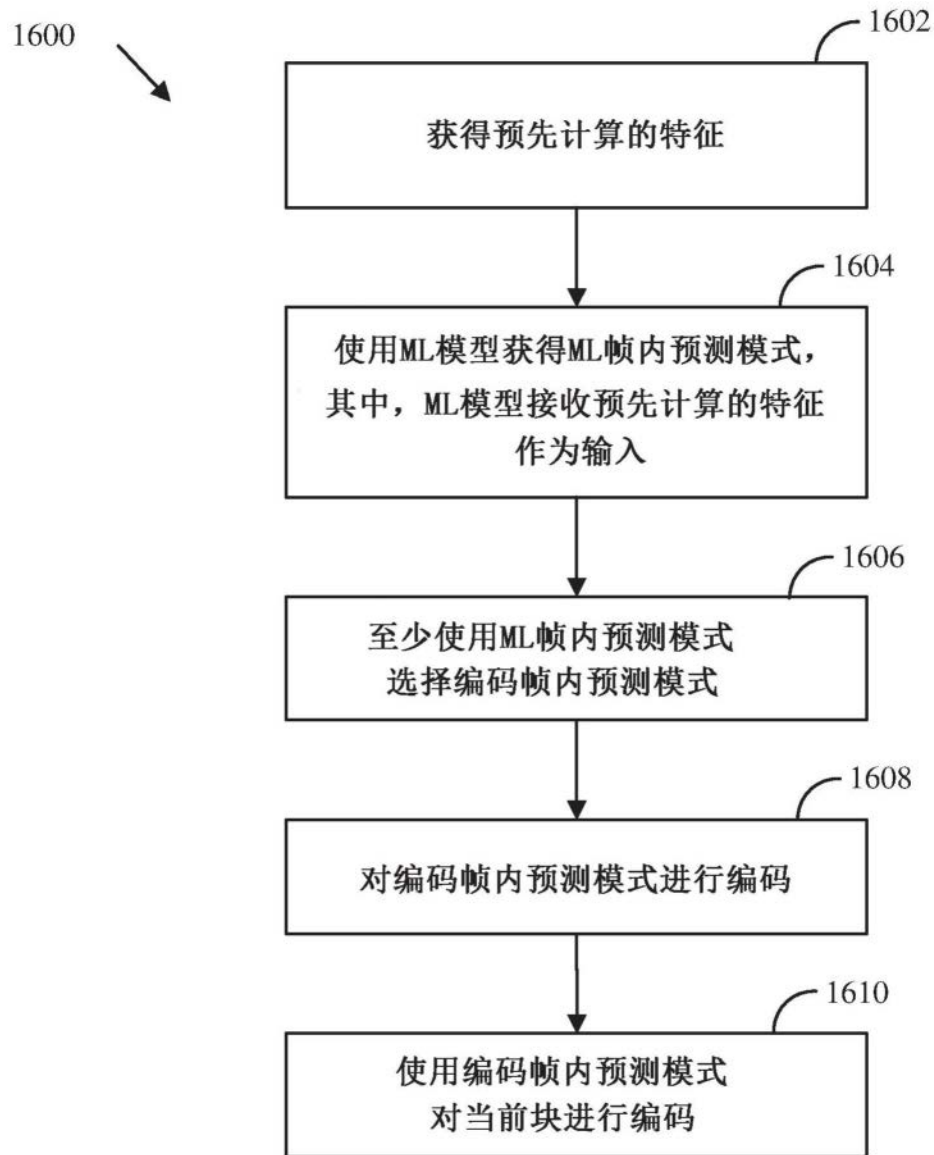


图16

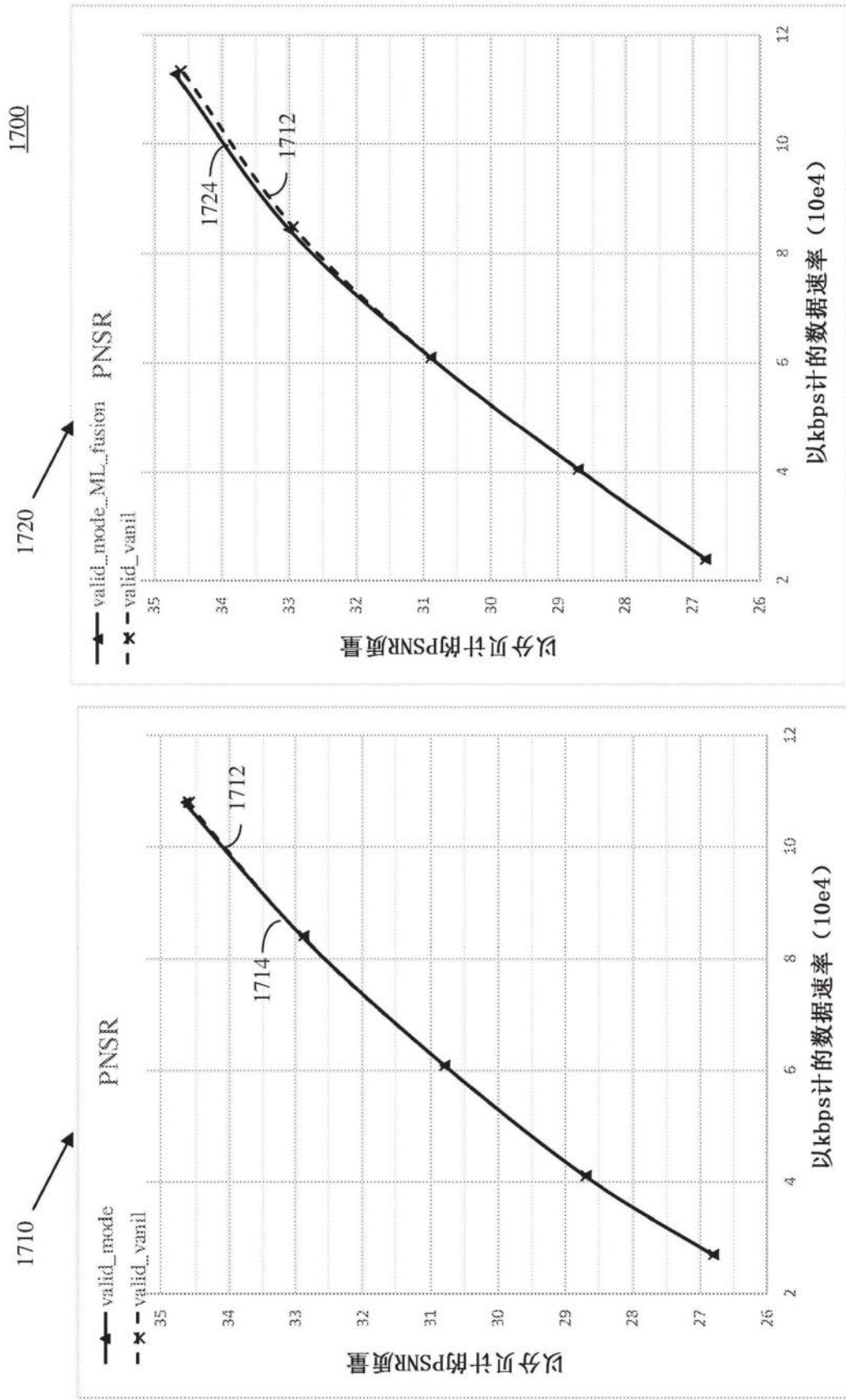


图17