



(12) 发明专利申请

(10) 申请公布号 CN 103650496 A

(43) 申请公布日 2014. 03. 19

(21) 申请号 201280034806. 1

H04N 19/176(2014. 01)

(22) 申请日 2012. 07. 11

H04N 19/593(2014. 01)

(30) 优先权数据

H04N 19/90(2014. 01)

61/506, 958 2011. 07. 12 US

H04N 19/11(2014. 01)

(85) PCT国际申请进入国家阶段日

2014. 01. 13

(86) PCT国际申请的申请数据

PCT/US2012/046287 2012. 07. 11

(87) PCT国际申请的公布数据

W02013/009896 EN 2013. 01. 17

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为

总部办公楼

(72) 发明人 高文 杨名远 宋锦 于浩平

(51) Int. Cl.

H04N 19/12(2014. 01)

H04N 19/103(2014. 01)

H04N 19/46(2014. 01)

H04N 19/154(2014. 01)

H04N 19/14(2014. 01)

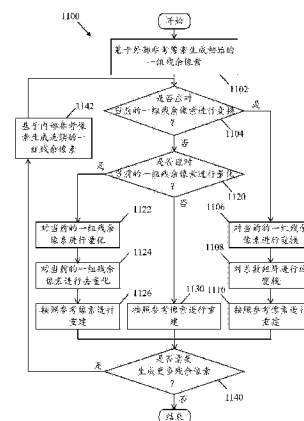
权利要求书3页 说明书13页 附图12页

(54) 发明名称

用于在 HEVC 中编码的基于像素的帧内预测

(57) 摘要

一种设备,其包括处理器,所述处理器用于:接收包括当前块的视频帧,其中所述当前块包括多个原始像素;基于所述原始像素计算多个重建像素;并且基于所述多个重建像素中的一组重建像素使用帧内预测模式来生成一组预测像素。



1. 一种设备,其包括:
处理器,所述处理器用于:
接收包括当前块的视频帧,其中所述当前块包括多个原始像素;
基于所述原始像素计算多个重建像素;并且
基于所述多个重建像素中的一组重建像素使用帧内预测模式来生成一组预测像素。
2. 根据权利要求1所述的设备,其中所述处理器进一步用于:
从多个可用帧内预测模式中选择所述帧内预测模式。
3. 根据权利要求2所述的设备,其中所述多个重建像素包括初始的一组重建像素,其中所述初始的一组重建像素是基于初始的一组预测像素确定的,其中所述帧内预测模式用来基于位于一个或多个邻近块中的一组外部参考像素生成所述初始的一组预测像素,并且其中所述一个或多个邻近块已在所述当前块之前进行编码。
4. 根据权利要求3所述的设备,其中所述处理器进一步用于:
计算所述一组预测像素中的每个预测像素与所述多个原始像素中对应的一组原始像素中的每个原始像素之间的差值,以生成一组残余像素,其中所述一组重建像素的计算是基于所述一组预测像素和所述一组残余像素完成的,其中所述一组重建像素用于生成第二组预测像素,其中根据所述帧内预测模式的预测顺序,所述一组预测像素中的每个预测像素位于所述第二组预测像素中的每个预测像素之前的一个位置中。
5. 根据权利要求4所述的设备,其中所述多个重建像素等于所述多个原始像素。
6. 根据权利要求4所述的设备,其中所述处理器进一步用于:
在所述一组残余像素上执行变换,以生成多个变换系数;
在所述多个变换系数上执行逆变换,以生成第二组残余像素;并且
将所述第二组残余像素与所述一组预测像素结合起来,以生成所述一组重建像素。
7. 根据权利要求6所述的设备,其中所述变换和逆变换是可逆的。
8. 根据权利要求7所述的设备,其中在包括所述一组残余像素的矩阵上执行所述变换,其中向不同于所述一组残余像素的所述矩阵的其他位置分配任意值。
9. 根据权利要求4所述的设备,其中所述处理器进一步用于:
对所述一组残余像素进行量化,以生成一组量化的残余像素;
对所述一组量化的残余像素进行去量化,以生成第二组残余像素;
将所述第二组残余像素与所述一组预测像素结合起来,以生成所述一组重建像素。
10. 根据权利要求9所述的设备,其中 $d(i, j)$ 表示所述一组残余像素,其中 i 和 j 是介于 0 和 $N-1$ 之间的整数,其中 N 是所述当前块的大小,其中 $d'(i, j)$ 表示所述第二组残余像素,其中 qp_scale 表示由量化参数(QP)确定的量化步骤,并且其中
$$d'(i, j) = \text{floor}(d(i, j) / qp_scale) * qp_scale.$$
11. 根据权利要求4所述的设备,其中所述块为预测单元(PU)。
12. 根据权利要求4所述的设备,其中所述帧内预测模式为垂直模式。
13. 根据权利要求4所述的设备,其中所述处理器进一步用于对所述一组残余像素进行编码,以生成编码数据。
14. 一种方法,其包括:
接收包括当前块的视频帧,其中所述当前块包括多个原始像素;

基于所述原始像素计算多个重建像素；并且

基于所述多个重建像素中的一组重建像素使用帧内预测模式来生成一组预测像素。

15. 根据权利要求 14 所述的方法，其进一步包括：

从多个可用帧内预测模式中选择所述帧内预测模式。

16. 根据权利要求 15 所述的方法，其中所述多个重建像素包括初始的一组重建像素，其中所述初始的一组重建像素是基于初始的一组预测像素确定的，其中所述帧内预测模式用来基于位于一个或多个邻近块中的一组外部参考像素生成所述初始的一组预测像素，并且其中所述一个或多个邻近块已在所述当前块之前进行编码。

17. 根据权利要求 16 所述的方法，其进一步包括：

计算所述一组预测像素中的每个预测像素与所述多个原始像素中对应的一组原始像素中的每个原始像素之间的差值，以生成一组残余像素，其中所述一组重建像素的计算是基于所述一组预测像素和所述一组残余像素完成的，其中所述一组重建像素用于生成第二组预测像素，其中根据所述帧内预测模式的预测顺序，所述一组重建像素中的每个重建像素位于所述第二组预测像素中的每个预测像素之前的一个位置中。

18. 根据权利要求 17 所述的方法，其中所述多个重建像素等于所述多个原始像素。

19. 根据权利要求 17 所述的方法，其进一步包括：

在所述一组残余像素上执行变换，以生成多个变换系数；

在所述多个变换系数上执行逆变换，以生成第二组残余像素；并且

将所述第二组残余像素与所述一组预测像素结合起来，以生成所述一组重建像素。

20. 根据权利要求 19 所述的方法，其中所述变换和逆变换是可逆的。

21. 根据权利要求 20 所述的方法，其中在包括所述一组残余像素的矩阵上执行所述变换，其中向不同于所述一组残余像素的所述矩阵的其他位置分配任意值。

22. 根据权利要求 17 所述的方法，其进一步包括：

对所述一组残余像素进行量化，以生成一组量化的残余像素；

对所述一组量化的残余像素进行去量化，以生成第二组残余像素；

将所述第二组残余像素与所述一组预测像素结合起来，以生成所述一组重建像素。

23. 根据权利要求 22 所述的方法，其中 $d(i, j)$ 表示所述一组残余像素，其中 i 和 j 是介于 0 和 $N-1$ 之间的整数，其中 N 是所述当前块的大小，其中 $d'(i, j)$ 表示所述第二组残余像素，其中 qp_scale 表示由量化参数 (QP) 确定的量化步骤，并且其中

$d'(i, j) = \text{floor}(d(i, j) / qp_scale) * qp_scale$ 。

24. 根据权利要求 17 所述的方法，其中所述块为预测单元 (PU)。

25. 根据权利要求 17 所述的方法，其中所述帧内预测模式为垂直模式。

26. 根据权利要求 17 所述的方法，其进一步包括对所述一组残余像素进行编码，以生成编码数据。

27. 一种设备，其包括：

处理器，所述处理器用于：

接收包括编码的残余块的比特流；

对所述编码的残余块进行解码，其中所述解码的残余块包括多个残余像素；并且

基于所述多个残余像素计算重建块，其中所述重建块包括多个重建像素，其中所述重

建块的计算包含基于所述重建块中的一组重建像素使用帧内预测模式来生成一组预测像素。

28. 根据权利要求 27 所述的设备,其中所述重建块包括初始的一组重建像素,其中所述初始的一组重建像素是基于初始的一组预测像素确定的,其中所述帧内预测模式用来基于位于一个或多个邻近块中的一组外部参考像素生成所述初始的一组预测像素,并且其中所述一个或多个邻近块已在所述重建块之前进行解码。

29. 根据权利要求 28 所述的设备,其中所述重建块的计算进一步包含将所述一组预测像素中的每个预测像素与对应的一组残余像素中的每个残余像素结合起来,以生成第二组重建像素,其中所述第二组重建像素用于生成第二组预测像素,其中根据所述帧内预测模式的预测顺序,所述一组预测像素中的每个预测像素位于所述第二组预测像素中的每个预测像素之前的位置中。

30. 根据权利要求 29 所述的设备,其中所述块为预测单元(PU)。

用于在 HEVC 中编码的基于像素的帧内预测

相关申请案的交叉参考

[0001] 本发明要求 2011 年 7 月 12 日由高文(Wen Gao)等人递交的发明名称为“用于复合视频的额外无损编码工具(Additional Lossless Coding Tools for Compound Video)”的第 61/506,958 号美国临时专利申请案的在先申请优先权,该在先申请的内容以引入的方式并入本文中,如全文再现一般。

关于由联邦政府赞助的研究或开发的声明

[0002] 不适用。

参考缩微胶片附录

[0003] 不适用。

技术领域

无

背景技术

[0004] 需要对大量的视频数据进行描述,即使在影片相对短小的情况下也是如此,当数据要在带宽容量有限的通信网络中流过或以其他方式传送时,这样可能会造成困难。因此,视频数据通常要先压缩然后在现代电信网络中传送。视频压缩装置通常在源处使用软件和/或硬件,以在传输之前对视频数据进行编码,从而减少用来表示数字视频图像所需的数据量。接着,压缩数据在目的地由视频解压装置接收,该视频解压装置用于对视频数据进行解码。在有限的网络资源以及对更高视频质量的需求不断增加的情况下,需要改进的压缩和解压技术,所述技术几乎无需以比特率的增加为代价就可以改进图像质量。

发明内容

[0005] 在一项实施例中,本发明包含一种设备,其包括处理器,所述处理器用于:接收包括当前块的视频帧,其中所述当前块包括多个原始像素;基于所述原始像素计算多个重建像素;并且基于所述多个重建像素中的一组重建像素使用帧内预测模式来生成一组预测像素。

[0006] 在另一项实施例中,本发明包含一种方法,所述方法包括:接收包括当前块的视频帧,其中所述当前块包括多个原始像素;基于所述原始像素计算多个重建像素;并且基于所述多个重建像素中的一组重建像素使用帧内预测模式来生成一组预测像素。

[0007] 在另一项实施例中,本发明包含一种设备,其包括处理器,所述处理器用于:接收包括编码的残余块的比特流;对所述编码的残余块进行解码,其中所述解码的残余块包括多个残余像素;并且基于所述多个残余像素计算重建块,其中所述重建块包括多个重建像素,其中计算所述重建块包含基于所述重建块中的一组重建像素使用帧内预测模式来生成一组预测像素。

[0008] 从结合附图和所附权利要求书进行的以下详细描述将更清楚地理解这些和其他

特征。

附图说明

[0009] 为了更完整地理解本发明,现在参考以下结合附图和详细描述进行的简要描述,其中相同参考标号表示相同部分。

[0010] 图 1 为多个帧内预测模式的图。

[0011] 图 2 为基于像素的帧内预测方案的一项实施例的图。

[0012] 图 3 为 4x4 的当前块和一些外部参考像素的一项实施例的图。

[0013] 图 4 为垂直模式中的帧内预测方案的一项实施例的图。

[0014] 图 5 为变换略过编码方案的一项实施例的示意图。

[0015] 图 6 为变换略过解码方案的一项实施例的示意图。

[0016] 图 7 为无量化变换编码方案的一项实施例的示意图。

[0017] 图 8 为无量化变换解码方案的一项实施例的示意图。

[0018] 图 9 为无变换量化编码方案的一项实施例的示意图。

[0019] 图 10 为无变换量化解码方案的一项实施例的示意图。

[0020] 图 11 为基于像素的帧内预测方法的一项实施例的流程图。

[0021] 图 12 为网络单元的一项实施例的示意图。

[0022] 图 13 为通用计算机系统的示意图。

具体实施方式

[0023] 首先应理解,尽管下文提供了一项或多项实施例的说明性实施方案,但可以使用任何数目的技术,不管是当前已知还是现有的,来实施所揭示的系统 and / 或方法。本发明决不应限于下文所说明的所述说明性实施方案、图式和技术,包含本文所说明并描述的示范性设计和实施方案,而是可以在所附权利要求书的范围及其等效物的完整范围内修改。

[0024] 通常,视频媒体涉及相对较快地连续显示一系列静态图像或帧,从而让观察者感觉到运动。每个帧可以包括多个图像元素或像素,每个图像元素或像素可以表示帧中的单个参考点。在数字处理期间,每个像素可以被分配一个整数值(例如,0、1、...或 255),所述整数值表示对应参考点处的图像质量或特征,例如,亮度或色度。在使用中,图像或视频帧可以包括大量的像素(例如,在 1920×1080 的帧中包括 2,073,600 个像素),因此对每个像素进行独立的编码和解码(下文中简称为编码)可能是繁琐且低效的。为了提高编码效率,通常将视频帧分成多个矩形块或宏块,每个矩形块或宏块可以作为预测、变换以及量化等处理的基本单元。例如,典型的 NxN 的块可以包括 N^2 个像素,其中 N 是大于 1 的整数,并且通常是四的倍数。

[0025] 在国际电信联盟(ITU)电信标准化部门(ITU-T)和国际标准化组织(ISO)/国际电工技术委员会(IEC)高性能视频编码(HEVC)的工作草案中,已经引入了新的块概念,其中高性能视频编码将准备作为未来视频标准。例如,编码单元(CU)可以指将视频帧分为相等大小或不同大小的矩形块的子分区。在 HEVC 中,CU 可以取代先前标准的宏块结构。根据帧间预测或帧内预测的模式,一个 CU 可以包括一个或多个预测单元(PU),每个预测单元可以用作预测的基本单元。例如,对于帧内预测,一个 64x64 的 CU 可以被对称地分为四个 32x32

的 PU。又例如,对于帧间预测,一个 64x64 的 CU 可以被非对称地分为一个 16x64 的 PU 和一个 48x64 的 PU。类似地,一个 PU 可以包括一个或多个变换单元(TU),每个变换单元可以用作变换和 / 或量化的基本单元。例如,一个 32x32 的 PU 可以被对称地分为四个 16x16 的 TU。一个 PU 的多个 TU 可以共用同一个预测模式,但是可以单独地进行变换。本文中,术语“块”通常可以指宏块、CU、PU 或 TU 中的任何一者。

[0026] 在视频帧或切片内,像素可以与相同帧内的其他像素相关,以使得一个块内的像素值或一些块内的像素值可能仅略有不同并且 / 或者呈现出重复的纹理(texture)。现代视频压缩方法使用各种技术来利用这些空间相关性(spatial correlation),这些技术可以统称为帧内部预测(或简称为帧内预测)。帧内部预测可以减少同一帧中邻近块之间的空间冗余,从而在没有大幅度降低图像质量的情况下对视频数据进行压缩。

[0027] 实际上,帧内部预测可以由视频编码器 / 解码器(编解码器)实施,以根据一个或多个先前编码 / 解码的邻近块插入预测块(或预测的块),从而形成近似的当前块。因此,编码器和解码器可以独立地插入预测块,从而使得帧和 / 或图像的绝大部分能够根据相对少量的参考块的通信进行重建,所述参考块例如,位于帧的左上角(并从此处延伸)的块。然而,单独的帧内部预测可能无法再现现代视频的质量足够好的图像,因此,残余消息等纠错消息可以在编码器与解码器之间传送,以纠正预测块与当前块之间的差异。例如,编码器可以从当前块中减去预测块或从预测块中减去当前块,从而得到残余块,接着所述残余块可以进行变换、量化以及扫描,然后被编码成编码数据流。在接收编码数据流之后,解码器可以将重建的残余块添加到单独生成的预测块上,以重建当前块。虽然重建的当前块可能是原始当前块的非完美版本,但是它们的差异可能是人眼几乎无法察觉的。因此,可以得到大量的位节约(bit saving),而不会显著降低重建图像的质量。

[0028] 残余块可以包括预测块与当前块之间的少量差异,因此,例如,在预测块与当前块相同和 / 或近乎相同的区域中,残余块的许多离散值,例如,像素数据,可以包括零和 / 或接近零的系数。此外,残余块的变换、量化和 / 或扫描可以从数据流中移除许多零和 / 或接近零的系数,从而引起对视频数据的进一步压缩。因此,对原始图像进行更精确的预测便可以得到编码效率。为了利用这些编码效率,传统视频 / 图像编码标准可以提高预测精确度,方式为在帧内预测期间使用多个预测模式,例如,每个预测模式可以生成独特的纹理。

[0029] 图 1 图示了多个帧内预测模式 100(下文中也称作帧内模式)的图,例如,所述帧内预测模式 100 可以用于 HEVC 统一帧内预测(UIP)方案中。对于亮度块,帧内预测模式 100 可以包括多达 35 个帧内预测模式,其可以包含 33 个定向模式以及 2 个非定向模式,例如,直流(DC)预测模式和平面预测模式。在使用中,编码器可以使用率失真优化(RDO)过程来选择能够生成每个当前块的最精确预测的预测模式。例如,在帧内预测模式 100 中可以计算每个模式的绝对误差之和(SAE),并且可以选择具有最小 SAE 的一个模式。通常情况下,大量的帧内预测模式可以产生更加精确的帧内预测。例如,最近的研究表明使用 35 个帧内部预测模式,例如,帧内预测模式 100,的约定与使用较少预测模式,例如,仅使用 9 个帧内预测模式的 ITU-T H. 264/ 高级视频编码(AVC)的约定相比,可以更加精确地预测复杂的纹理。尽管图 1 示出 35 个帧内预测模式,但是根据块的大小和 / 或帧内预测方案,也可以使用任何其他数目的预测模式。

[0030] 在当前帧内预测方案中,当前块周围的像素可以用作参考像素(或预测样本),以

生成预测块。帧内预测的质量可能受到块的大小和预测模式等因素的影响。例如,随着预测块的大小增加,预测块中距离参考像素较远的像素可能与参考像素具有较小的空间相关性,因此,较远像素的预测精确度可能降低。这种预测精确度的降低可能会产生更多残余数据,进而可能增加要进行编码、存储和 / 或传送的数据。

[0031] 本文中所揭示的是用于改进视频编码中的帧内预测的系统和方法。本发明提供一种帧内预测方案,该方案不仅将邻近块中的像素用作外部参考像素,而且将当前块内的像素用作内部参考像素。在一项实施例中,可以基于多个外部参考像素生成初始的一组预测像素。根据帧内预测模式,初始的一组预测像素可以位于当前预测块的行或列中或者同时位于当前预测块的行和列中。可以从当前块的初始的一组对应原始像素中减去所述初始的一组预测像素,或者从所述初始的一组预测像素中减去当前块的初始的一组对应原始像素,从而生成初始的一组残余像素。此外,为了生成连续的一组残余像素,可以使用多个内部参考像素,所述内部参考像素可以是原始像素的重建版本。例如,在帧内预测的垂直模式中,可以基于一行上方块对初始的一组当前块或上行的当前块进行预测。第二行可以根据顶行的重建版本进行预测,第三行可以根据相对于顶行的第二行的重建版本进行预测,并且依次类推。根据编码方案是否有损耗,重建像素可以是原始像素的无损版本或有损版本。在本发明中,内部参考像素可能直接地优于连续的一组预测像素,因此无论当前块的大小如何,预测距离可以始终保持为一。由于外部 / 内部参考像素和它们的预测像素的空间接近性,因此与当前帧内预测方案相比,所揭示的帧内预测系统和方法可以提供更精确的预测。本文中所揭示的基于像素的帧内预测方案可以在多种编码和解码方案中实施,例如,变换略过编码方案(即,略过变换步骤和量化步骤)、无量化变换编码方案(即,略过量化步骤)或无变换量化编码方案(即,略过变换步骤)。

[0032] 图 2 示出基于像素的帧内预测方案 200 的一项实施例。为了对 $N \times N$ 的当前块 210 进行预测,帧内预测方案 200 可以生成一个 $N \times N$ 的预测块。为了说明目的,假设当前块 210 的大小如图 2 所示为 8×8 。由于当前块 210 周围的像素可能已经在当前块 210 之前被编码,所以它们可以用于对当前块 210 进行预测。确切地说,位于当前块 210 左侧(以及与当前块 210 相邻)的列中,以及位于当前块 210 上方(以及与当前块 210 相邻)的行中的多个像素可以用作帧内预测的参考像素。因为这些像素位于当前块 210 的外部,所以它们在下文中可以称为外部参考像素 220,这些像素在图 2 中标记为向下阴影线的像素。由于可以使用多达 35 个帧内预测模式(例如,图 1 所示),因此对于 $N \times N$ 的预测块,外部参考像素 220 的数目可以多达 $4N+1$ 个。例如,如图 2 所示,对于 8×8 的当前块 210,可以具有多达 33 个外部参考像素 220。尽管图 2 示出正方形当前块 210,但是应理解,所揭示的基于像素的帧内预测方案也可以用于预测非正方形块,所述非正方形块可以用于短距离帧内预测(SDIP)方案中。在本发明中,由于所有的行都可以根据它们的相邻上方行进行预测,因此无论当前块的大小如何,预测像素与参考像素之间的距离可以始终保持为一。因此,可以提高帧内预测的精确度。

[0033] 与可以按并行方式生成当前块的所有预测像素(或预测的像素)的当前帧内预测方案相比,所揭示的帧内预测可以按顺序方式生成预测像素或一组接一组地生成预测像素。在一项实施例中,外部参考块 220 的一部分可以用于生成初始的一组预测像素。本文中的初始组可以指首先根据特定的帧内模式进行预测的一行像素和 / 或一列像素。例如,

在垂直模式(图 1 中标记为模式 1)中,初始组可以是当前块 210 的顶行。又例如,在水平模式(图 1 中标记为模式 2)中,初始组可以是当前块 210 的左列。再例如,在右向下对角线预测(图 1 中标记为模式 4)中,初始组可以是当前块 210 的顶行和左列。

[0034] 在获得初始的一组预测像素之后,可以通过从预测值中减去原始像素值生成初始的一组残余或差别像素,反之亦然。随后,连续的预测像素组可以基于位于当前块 210 内的参考像素来确定,所述参考像素在下文中可以指内部参考像素 230。应注意,为了避免潜在的解码偏差,应将相同的外部/内部参考像素用在编码器和解码器中。因此,重建像素而不是原始像素可以用作参考像素。由于重建像素是通过编码器和解码器中相同系列的程序从原始像素转化而来的,因此可以确保参考像素是相同的。

[0035] 连续的预测像素组可以基于初始的一组像素的重建版本而生成。例如,在垂直模式中,相对于顶行的第二行可以根据重建的顶行进行预测,相对于顶行的第三行可以根据重建的相对于顶行的第二行进行预测,并且依次类推。考虑到所有的预测模式,除了右下角的一个之外,当前块中的所有像素都可以潜在地用作内部参考像素。如图 2 所示,内部参考像素 230 标记为向上阴影线的像素,而右下角的像素 240 不标阴影线。

[0036] 图 3 示出 4x4 的当前块(例如,PU)和一些外部参考像素的一项实施例。当前块的原始像素值表示为 $Z(i, j)$, 其中 i 为介于 0 和 3 之间的行索引,并且 j 为介于 0 和 3 之间的列索引。针对 $j=0, \dots, 3$, 位于上方块的最后一行中的像素表示为 $X(j)$, 而针对 $i=0, \dots, 3$, 位于左侧块的右列中的像素表示为 $Y(i)$ 。第三邻近块的右下角的像素表示为 $XY(0)$ 。对于当前块,可以生成表示为 $P(i, j)$ 的预测像素,其中 $i=0, \dots, 3$ 并且 $j=0, \dots, 3$, 所述预测像素可以是原始像素 $Z(i, j)$ 的估计版本。帧内预测可以在多个帧内预测模式(例如,图 1 中的多个帧内模式 100)中的任一模式中实施。

[0037] 图 4 示出垂直模式(图 1 中标记为模式 1)中的帧内预测方案的一项实施例。在垂直模式中,初始的一组当前块(在这种情况下,根据 i 指示,为顶行或第 0 行)可以根据其上方块按照 $P(0, j)=X(j)$, 其中 $j=0, \dots, 3$ 进行预测。可以从原始像素 $Z(0, j)$ 中减去初始组,生成一组残余或差别像素(表示为 $D(0, j)$), 反之亦然。从数学角度而言, $D(0, j)=Z(0, j)-P(0, j)$ 或 $D(0, j)=P(0, j)-Z(0, j)$ 。所述一组残余像素可以进行处理并且用于生成一组重建像素(表示为 $R(0, j)$), 所述的一组重建像素可以为一组原始像素 $Z(0, j)$ 的精确版本或有损版本。如上文所述,重建像素而不是原始像素可以用于进一步预测,以避免编码偏差。连续的一组当前块(在这种情况下,为第 1 行至第 3 行中的任一行)可以根据上文重建的行按照 $P(i, j)=R(i-1, j)$, 其中 $i=1, 2, 3$ 以及 $j=0, \dots, 3$ 进行预测。这种帧内预测方案可以不同于当前帧内预测方案,其中当前块的所有行都可以根据其上方块按照 $P(i, j)=X(j)$, 其中 $i=0, \dots, 3$ 以及 $j=0, \dots, 3$ 进行预测。在本发明中,由于所有的行都可以根据它们的相邻上方行进行预测,因此无论当前块的大小如何,预测像素与参考像素之间的距离可以始终保持为一。因此,可以提高帧内预测的精确度。

[0038] 尽管图 4 中未示出,但是其他帧内预测模式可以使用类似于垂直模式的方案实施。例如,如果使用水平模式(图 1 中标记为模式 1),那么当前块的第 0 列可以根据其左邻近块按照 $P(i, 0)=Y(i)$, 其中 $i=0, \dots, 3$ 进行预测。当前块中的其他列(即,第 1 列至第 3 列)可以根据它们的左列按照 $P(i, j)=R(i, j-1)$, 其中 $i=0, \dots, 3$ 以及 $j=1, 2, 3$ 进行预测。在水平模式中,所揭示的基于像素的帧内预测方案可以不同于当前帧内预测方案,其中当前

块的所有列都可以根据左侧块按照 $P(i, j)=Y(i)$, 其中 $i=0, \dots, 3$ 以及 $j=0, \dots, 3$ 进行预测。又例如, 如果使用右向下对角线模式(图 1 中标记为模式 4), 那么包含第 0 行和第 0 列的初始的一组当前块可以根据三个邻近块中的像素按照 $P(0, 0)=XY(0)$ 、 $P(0, j+1)=X(j)$ 以及 $P(i+1, 0)=Y(i)$, 其中 $i=0, 1, 2$ 以及 $j=0, 1, 2$ 进行预测。当前块的其他行和其他列可以根据它们的左上方邻近像素按照 $P(i+1, j+1)=R(i, j)$, 其中 $i=0, 1, 2$ 以及 $j=0, 1, 2$ 进行预测。

[0039] 如上文所述, 本文中所揭示的基于像素的帧内预测方案可以在多种编码方案中实施。根据应用, 可以在视频编码器中实施有损(即, 存在失真或信息损失)和 / 或无损(即, 不存在失真或信息损失)编码。例如, 当对视频帧或切片进行编码时, 对于视频帧中的所有块或者一些块, 视频编码器可以略过或跳过变换步骤和 / 或量化步骤。本文中, 有损编码模式可以包含无变换量化编码(即, 略过变换步骤), 而无损编码模式可以包含变换略过编码(即, 略过变换步骤和量化步骤)以及无量化变换编码(即, 略过量化步骤)。类似地, 基于包含在接收的比特流中的信息, 视频解码器可以使用无损模式和 / 或有损模式对视频帧进行解码。有损解码模式可以包含无变换量化解码, 而无损解码模式可以包含变换略过解码和无量化变换解码。

[0040] 根据编码方案是否有损耗, 重建像素可以是原始像素的精确版本或有损版本。由于对于其他像素的帧内预测, 重建像素可以用作参考像素, 因此用无损 / 有损方案中帧内预测的精确度可能不同。此外, 由于用在无损和有损方案中的编码模块可能不同, 因此可以不同地实施所揭示的基于像素的帧内预测。为了简明起见, 在下文的段落中将描述有损和无损编码方案中的基于像素的帧内预测的各项实施例的应用。

[0041] 图 5 图示了可以在视频编码器中实施的变换略过编码方案 500 的一项实施例。变换略过编码方案 500 可以包括如图 5 所示进行布置的率失真优化(RDO)模块 510、预测模块 520、熵编码器 530 以及重建模块 540。在运行中, 包括视频帧(或切片)的序列的输入视频可以被编码器接收。本文中, 帧可以指预测帧(P 帧)、帧内编码帧(I 帧)或双向预测帧(B 帧)中的任一者。类似地, 切片可以指 P 切片、I 切片或 B 切片中的任一者。

[0042] RDO 模块 510 可以用于为一个或多个其他模块作出逻辑决策。例如, RDO 模块 510 可以通过从多个可用预测模式中确定当前块(例如, PU)的最佳帧内预测模式而与预测模块 520 配合。RDO 模块可以基于多种算法选择最佳帧内模式。例如, RDO 模块 510 可以针对每个预测模式来计算绝对误差之和(SAE), 并且选择产生最小 SAE 的预测模式。然而, 无论 RDO 使用的度量如何, 基于像素的帧内预测可以被视作帧内模式中的一者。

[0043] 基于由 RDO 模块 510 作出的逻辑决策, 预测模块 520 可以利用外部参考像素和内部参考像素来生成当前块的预测像素。可以从当前块中的对应原始像素中减去每个预测像素, 或者从每个预测像素中减去当前块的对应原始像素, 从而生成残余像素。为了有助于像素的连续编码, 残余块也可以被送入重建模块 540 中, 重建模块 540 可以生成重建像素, 以用作未来像素的帧内预测的参考像素。

[0044] 随后, 在针对当前块已生成所有残余像素之后, 可以对残余像素进行扫描, 并且熵编码器 530 可以将非零残余像素编码成编码比特流。熵编码器 530 可以采用任何熵编码方案, 例如, 内容适应性二进制算术编码(CABAC)编码、指数哥伦布编码, 或固定长度编码, 或其任意组合。在变换略过编码方案 500 中, 由于残余块是在没有变换步骤或量化步骤的情况下进行编码的, 因此在所述编码过程中不会引起任何信息损失。应注意, 图 5 可能是视

频编码器的简化图示,因此其可能只包含编码器中呈现的部分模块。其他模块(例如,滤波器、扫描仪,以及发射器)虽然未在图 5 中示出,但是也可以包含在内以协助视频编码。在从编码器处进行传输之前,编码比特流可以进一步配置成包含其他信息,例如视频分辨率、帧率、块分区信息(大小、坐标)、预测模式等,因此可以对视频帧的编码序列进行适当地解码。

[0045] 当通过变换略过编码方案 500 对 $N \times N$ 的当前块进行编码时,由于残余像素(即, $D(i, j)$, 其中 $i=0, \dots, N-1$ 以及 $j=0, \dots, N-1$) 可以直接添加到预测像素(即, $P(i, j)$) 中,以生成重建像素(例如, $R(i, j)=D(i, j)+P(i, j)$) 而无需任何额外过程,因此在所述重建过程中不会引起失真或信息损失。因此,重建像素($R(i, j)$) 可以与原始像素($Z(i, j)$) 完全相同。在一项实施例中,可以基于多个外部参考像素生成初始的一组预测像素,同时可以基于先前的一组内部参考像素生成连续的一组预测像素,所述连续的一组预测像素为重建像素。例如,在垂直帧内模式中,当前块的第 0 行可以根据上方块按照 $P(0, j)=X(j)$, 其中 $j=0, \dots, N-1$ 进行预测。可以从原始像素 $Z(0, j)$ 中减去第 0 行中的预测像素,生成一组残余像素 $D(0, j)$ 。此外,所述一组残余像素可以直接用于生成一组重建像素 $R(0, j)$ 。连续的行可以根据上文的重建的行按照 $P(i, j)=R(i-1, j)$, 其中 $i=1, \dots, N-1$ 以及 $j=0, \dots, N-1$ 进行预测。

[0046] 图 6 图示了可以在视频解码器中实施的变换略过解码方案 600 的一项实施例。变换略过解码方案 600 可以对应于变换略过编码方案 500, 并且可以包括如图 6 所示进行布置的熵解码器 610、预测模块 620 以及重建模块 630。在运行中,熵解码器 610 可以接收含有视频帧序列信息的编码比特流,该熵解码器可以将比特流解码成未压缩格式。熵解码器 610 可以采用任何熵解码算法,例如, CABAC 解码、指数哥伦布解码,或固定长度解码,或其任意组合。

[0047] 对于被解码的当前块而言,在使用熵解码器 610 之后,可以生成残余块。此外,包含有当前块的预测模式的信息也可以由熵解码器 610 进行解码。随后,基于预测模式以及位于一个或多个先前解码的邻近块中的多个外部参考像素,预测模块 620 可以生成初始的一组预测像素。随后,重建模块 630 可以将初始的一组预测像素与对应的一组残余像素结合起来以生成一组重建像素。重建像素也可以用作参考像素,用于对连续的像素组进行解码。具体而言,通过使用所述初始的一组重建像素,可以生成第二组预测像素。因此,第二组预测像素可以添加到第二组残余像素,以获得第二组重建像素。可以继续这种重复过程,直到获得当前块的所有重建像素为止。随后,解码器可以前进到重建下一个块。

[0048] 在使用中,如果使用变换略过编码方案 500 和变换略过解码方案 600 等无损方案对原始块进行编码和解码,那么在整个编码过程中可能不会引起任何信息损失。因此,除了在传输过程中引起的失真之外,重建块可以与原始块完全相同。

[0049] 在视频帧中的某些块的无损编码期间,有时可能需要将变换步骤包含到编码过程中。例如,对于文本区域的一些块而言,添加的变换步骤可以生成与变换略过编码方案相比更加短的比特流。图 7 图示了无量化变换编码方案 700 的一项实施例,所述编码方案可以包括 RDO 模块 710、预测模块 720、变换模块 730、熵编码器 740, 逆变换模块 750 以及重建模块 760。无量化变换编码方案 700 的一些方面可以与图 5 中的变换略过编码方案 500 相同或类似,因此为了简明起见将不会对类似的方面进行详细描述。

[0050] 无量化变换编码方案 700 可以在视频编码器中实施,所述视频编码器可以接收包

括视频帧序列的输入视频。RDO 模块 710 可以用于控制一个或多个其他模块，并且可以与图 5 中的 RDO 模块 510 相同或类似。基于由 RDO 模块 710 作出的逻辑决策，预测模块 720 可以利用外部参考像素和内部参考像素来生成当前块的预测像素。可以从当前块的对应原始像素中减去每个预测像素 ($P(i, j)$)，或者从每个预测像素 ($P(i, j)$) 中减去当前块的对应原始像素，从而生成残余像素。预测模块 720 可以与图 5 中的预测模块 520 相同或类似。

[0051] 预测模块 720 可以基于多个外部参考像素生成初始的一组残余像素。随后，所述初始的一组残余像素首先可以通过变换模块 730 从空间域变换到频域，这可以使用任何可逆变换来实施。例如，在垂直帧内模式中，第 0 行中表示为 $D(0, j)$ 的残余像素可以使用可逆变换来进行变换，其中 j 介于 0 和 $N-1$ 之间。此时，由于其他行中的残余像素还不能被预测模块 720 使用，因此其他行可以被任意地分配或填充或填补合适的像素值。例如，可以简单地向其他行可以分配零像素值。获得 $N \times N$ 的块之后，该块包括初始的一组残余像素以及任意分配的值，可逆变换可以应用到 $N \times N$ 的块上，以生成变换系数矩阵。

[0052] 为了有助于当前块中其他行的编码，变换系数矩阵也可以被送入逆变换模块 750 中，逆变换模块 750 可以执行变换模块 730 的逆变换，并且将变换系数从频域转化到空间域。逆变换模块 750 可以生成另一组残余像素，表示为 $D'(0, j)$ ，其中 j 介于 0 和 $N-1$ 之间。根据可逆变换是否完全可逆， $D'(0, j)$ 可以与 $D(0, j)$ 相同或与 $D(0, j)$ 略有不同。无论是哪一种情况，重建模块 760 都可以使用 $D'(0, j)$ 以按照 $R(0, j) = D'(0, j) + P(0, j)$ 来生成一组重建像素。随后，重建像素 $R(0, j)$ 的第 0 行可以用作内部参考像素，以按照 $P(1, j) = R(0, j)$ 生成第二组（即，第 1 行）预测像素。可以再次生成第二组残余像素 $D(1, j)$ 。其他行可以被分配任意值，并且可以对重建的 $N \times N$ 的块进行变换和逆变换，以生成含有 $D'(1, j)$ 的块。随后， $D'(1, j)$ 可以用作内部参考像素，用于继续帧内预测。可以继续这种重复过程，直到针对当前块生成所有残余像素为止。此种重复预测过程的某些方面可以类似于变换略过编码方案 500 中的过程，因此为了简明起见将不会对类似的方面进行详细描述。所述变换（例如，在变换模块 730 中）可以为垂直或水平方向上的一维“线变换”，并且该变换可以同时在一行或一列上执行。如果变换完全可逆，那么可以在无偏差的情况下，在无量化变换方案中执行基于像素的预测。

[0053] 在针对当前块已生成所有变换系数之后，变换系数可以被扫描，并且非零变换系数可以由熵编码器 740 编码成编码比特流。熵编码器 740 可以与熵编码器 530 相同或类似。在从编码器处进行传输之前，编码比特流可以进一步配置成包含其他信息，例如视频分辨率、帧率、块分区信息（大小、坐标）、预测模式等，因此可以对视频帧的编码序列进行适当地解码。

[0054] 图 8 图示了可以在视频解码器中实施的无量化变换解码方案 800 的一项实施例。变换略过解码方案 800 可以对应于无量化变换编码方案 700，并且可以包括如图 8 所示进行布置的熵解码器 810、逆变换模块 820、预测模块 830 以及重建模块 840。在运行中，熵解码器 810 可以接收含有视频帧序列信息的编码比特流，该熵解码器可以将比特流解码成未压缩格式。熵解码器 810 可以与图 6 中的熵解码器 610 相同或类似。在使用熵解码器 810 之后，可以生成变换系数的矩阵，该矩阵随后可以被送入逆变换模块 820 中。逆变换模块 820 可以将频域中的变换系数转化为空间域中的残余像素值。逆变换模块 820 可以与图 7 中的逆变换模块 750 相同或类似。

[0055] 对于被解码的当前块而言,在使用逆变换模块 820 之后,可以生成残余块。此外,包含有当前块的预测模式的信息也可以由熵解码器 810 进行解码。随后,基于预测模式以及位于一个或多个先前解码的邻近块中的多个外部参考像素,预测模块 830 可以生成初始的一组预测像素。随后,重建模块 840 可以将初始的一组预测像素与对应的一组残余像素结合起来以生成一组重建像素。重建像素也可以用作参考像素,用于对连续的像素组进行解码。具体而言,通过使用所述初始的一组重建像素,可以生成第二组预测像素。因此,第二组预测像素可以添加到第二组残余像素,以获得第二组重建像素。可以继续这种重复过程,直到获得当前块的所有重建像素为止。随后,解码器可以前进到重建下一个块。

[0056] 在使用中,有时可能需要将量化步骤而不是变换步骤包含到编码过程中。图 9 图例示了无变换量化编码方案 900 的一项实施例,所述编码方案可以包括 RDO 模块 910、预测模块 920、量化模块 930、熵编码器 940、去量化模块 950 以及重建模块 960。无变换量化编码方案 900 的一些方面可以与图 5 中的变换略过编码方案 500 或图 7 中的无量化变换编码方案 700 相同或类似,因此为了简明起见将不会对类似的方面进行详细描述。

[0057] 无变换量化编码方案 900 可以在视频编码器中实施,所述视频编码器可以接收包括视频帧序列的输入视频。RDO 模块 910 可以用于控制一个或多个其他模块,并且可以与图 5 中的 RDO 模块 510 相同或类似。基于由 RDO 模块 910 作出的逻辑决策,预测模块 920 可以利用外部参考像素和内部参考像素来生成当前块的预测像素。可以从当前块的对应原始像素中减去每个预测像素 ($P(i, j)$),或者从每个预测像素 ($P(i, j)$) 中减去当前块的对应原始像素,从而生成残余像素。预测模块 920 可以与图 7 中的预测模块 720 相同或者类似。

[0058] 预测模块 920 可以基于多个外部参考像素生成初始的一组残余像素。随后,所述初始的一组残余像素首先可以通过量化模块 930 进行量化或重调大小。根据应用,量化模块 930 可以采用任何合适的量化参数(QP)。例如,在垂直帧内模式中,第 0 行中表示为 $D(0, j)$ 的残余像素可以转化成表示为 $q_D(0, j)$ 的量化残余像素,其中 j 介于 0 和 $N-1$ 之间。量化可以使用方程式 $q_D(0, j) = \text{floor}(d(0, j) / qp_scale)$,其中 qp_scale 表示由 QP 确定的量化步骤。

[0059] 为了有助于当前块中其他行的编码,初始的一组量化残余像素可以被送入去量化模块 950 中,去量化模块 950 可以执行量化模块 930 的逆变换,并且恢复残余像素的标度。去量化模块 950 可以通过方程式 $D'(0, j) = q_D(0, j) * qp_scale = \text{floor}(d(0, j) / qp_scale) * qp_scale$ 生成另一组残余像素,表示为 $D'(0, j)$,其中 j 介于 0 和 $N-1$ 之间。重建模块 960 可以使用 $D(0, j)$ 的有损版本 $D'(0, j)$,以按照 $R(0, j) = D'(0, j) + P(0, j)$ 生成一组重建像素。随后,重建像素 $R(0, j)$ 的第 0 行可以用作内部参考像素,以按照 $P(1, j) = R(0, j)$ 生成第二组(即,第 1 行)预测像素。可以对第二组残余像素 $D(1, j)$ 进行再次生成、量化以及去量化,从而生成含有 $D'(1, j)$ 的块。随后, $D'(1, j)$ 可以用作内部参考像素,用于继续帧内预测。可以继续这种重复过程,直到针对当前块生成所有残余像素为止。此种重复预测过程的某些方面可以类似于变换略过编码方案 500 中的过程,因此为了简明起见将不会对类似的方面进行详细描述。

[0060] 在针对当前块已生成所有量化残余像素之后,量化残余像素可以被扫描,并且非零量化残余像素可以由熵编码器 940 编码成编码比特流。熵编码器 940 可以与熵编码器 530 相同或类似。在从编码器处进行传输之前,编码比特流可以进一步配置成包含其他信息,例

如视频分辨率、帧率、块分区信息(大小、坐标)、预测模式等,因此可以对视频帧的编码序列进行适当地解码。

[0061] 图 10 图示了可以在视频解码器中实施的无变换量化解码方案 1000 的一项实施例。变换略过解码方案 1000 可以对应于无变换量化编码方案 900,并且可以包括如图 10 所示进行布置的熵解码器 1010、去量化模块 1020、预测模块 1030 以及重建模块 1040。在运行中,熵解码器 1010 可以接收含有视频帧序列信息的编码比特流,该熵解码器可以将比特流解码成未压缩格式。熵解码器 1010 可以与图 6 中的熵解码器 610 相同或类似。在使用熵解码器 1010 之后,可以生成量化残余像素的块,该块随后可以被送入去量化模块 1020 中。去量化模块 1020 可以与图 9 中的去量化模块 950 相同或类似。

[0062] 对于被解码的当前块而言,在使用逆变换模块 1020 之后,可以生成残余块。此外,包含有当前块的预测模式的信息也可以由熵解码器 1010 进行解码。随后,基于预测模式以及位于一个或多个先前解码的邻近块中的多个外部参考像素,预测模块 1030 可以生成初始的一组预测像素。随后,重建模块 1040 可以将初始的一组预测像素与对应的一组残余像素结合起来以生成一组重建像素。重建像素也可以用作参考像素,用于对连续的像素组进行解码。具体而言,通过使用所述初始的一组重建像素,可以生成第二组预测像素。因此,第二组预测像素可以添加到第二组残余像素,以获得第二组重建像素。可以继续这种重复过程,直到获得当前块的所有重建像素为止。随后,解码器可以前进到重建下一个块。

[0063] 图 11 图示了基于像素的帧内预测方法 1100 的一项实施例,该预测方法可以为上述编码方案的一部分。方法 1100 可以从步骤 1102 开始,其中对于当前块,可以生成初始的一组或第一组残余像素。具体而言,可以基于位于邻近块中的多个外部参考像素生成初始的一组预测像素。可以从当前块的对应原始像素中减去所述初始的一组预测像素,或者从所述初始的一组预测像素中减去当前块的对应原始像素,从而生成所述初始的一组残余像素。根据帧内预测模式,初始组可以位于行中、列中或者这两者中。例如,在对 $N \times N$ 的块进行帧内预测的垂直模式中,所述初始的一组残余像素为 $D(0, j) = X(0, j) - Z(0, j)$ 或 $D(0, j) = Z(0, j) - X(0, j)$,其中 j 介于 0 和 $N-1$ 之间。接下来,在步骤 1104 中,方法 1100 可以确定是否应在所述一组残余像素上执行变换。如果满足方框 1104 中的条件,那么方法 1100 可以前进到步骤 1106。否则,方法 1100 可以前进到步骤 1120。

[0064] 在步骤 1106 中,使用可逆变换将当前的一组残余像素变换到频域中。由于此时还不能使用其他残余像素,因此可以将任意值分配到其他残余像素。例如,在垂直模式中,当变换所述当前的一组残余像素 $D(i, j)$ 时,可以简单地向其他行中的残余像素(即, $D(k, j)$, 其中 $k=0, \dots, N-1$ 并且 $k \neq i$) 分配零像素值(即, $D(k, j)=0$)。在垂直模式中,当 $i=0$ 时,当前组可以为初始组。通过重建 $N \times N$ 的块,该块包括所述当前的一组残余像素以及其他任意分配的值,可逆变换可以应用到 $N \times N$ 的块上,以生成变换系数矩阵。接下来,在步骤 1108 中,可以对变换系数矩阵进行逆变换,以生成可能与 $D(i, j)$ 相同或类似的另一组残余像素 $D'(i, j)$ 。接下来,在步骤 1110 中,例如,使用方程式 $R(i, j) = D'(i, j) + P(i, j)$ 可以将所述一组残余像素 $D'(i, j)$ 与预测像素结合起来以重建一组像素。在初始组中,当 $i=0$ 时,所述一组重建像素为初始组 $R(0, j) = D'(0, j) + P(0, j)$ 。

[0065] 在方框 1104 中,如果方法 1100 确定不应在所述一组残余像素上执行变换,那么接下来在步骤 1120 中,方法 1100 可以确定是否应在所述一组残余像素上执行量化。如果

满足方框 1120 中的条件,那么方法 1100 可以前进到步骤 1122。否则,方法 1100 可以前进到步骤 1130。在步骤 1122 中,所述一组残余像素可以进行量化或缩小标度,以生成一组量化的残余像素。量化可以具有确定量化步骤的预设 QP 。所述一组量化残余可以表示为 $q_D(i, j) = \text{floor}(d(i, j)/qp_scale)$ 。接下来,在步骤 1124 中,所述一组量化的残余像素可以进行去量化,以恢复残余像素的标度。恢复的残余像素 $D'(i, j)$ 可以表示为 $D'(i, j) = q_D(i, j) * qp_scale = \text{floor}(d(i, j)/qp_scale) * qp_scale$,这可能为 $D(i, j)$ 的有损版本。接下来,在步骤 1126 中,例如,使用方程式 $R(i, j) = D'(i, j) + P(i, j)$ 可以将所述一组残余像素 $D'(i, j)$ 与预测像素 $P(i, j)$ 结合起来以重建一组像素。在初始组中,当 $i=0$ 时,所述一组重建像素为初始组 $R(0, j) = D'(0, j) + P(0, j)$ 。

[0066] 在方框 1120 中,如果方法 1100 确定不应在所述一组残余像素上执行量化,那么接下来在步骤 1130 中,使用方程式 $R(i, j) = D(i, j) + P(i, j)$ 可以将所述一组残余像素 $D(i, j)$ 与预测像素 $P(i, j)$ 结合起来以重建一组像素。在这种情况下,由于在重建过程中不会引起失真或信息损失,因此 $R(i, j) = Z(i, j)$ 可能是正确的。具体而言,在初始组中,当 $i=0$ 时,所述一组重建像素为初始组 $R(0, j) = Z(0, j) = D(0, j) + P(0, j)$ 。

[0067] 接下来,在步骤 1140 中,方法 1100 可以确定是否会生成更多残余像素。如果满足方框 1120 中的条件,那么方法 1100 可以前进到步骤 1142。否则,方法 1100 可以结束。在步骤 1142 中,可以生成连续的一组或下一组残余像素。具体而言,可以基于多个内部参考像素生成连续的一组预测像素,所述多个内部参考像素可以为根据帧内预测模式的位于当前行和 / 或列之前的行和 / 或列中的重建像素。可以从当前块的对应原始像素中减去所述连续的一组预测像素,或者从所述连续的一组预测像素中减去当前块的对应原始像素,从而生成所述连续的一组残余像素。例如,在垂直模式中, $D(i, j) = R(i-1, j) - Z(i, j)$ 或 $D(i, j) = Z(i, j) - R(i-1, j)$ 。在生成所述连续的一组残余像素之后,方法 1100 可以返回到步骤 1104。

[0068] 最后,针对当前块获得所有残余像素,然后方法 1100 可以结束。应理解,方法 1100 可能仅包含所有必要编码步骤中的一部分,因而其他步骤,例如,扫描、传输和滤波也可以在必要的地方并入到编码过程中。此外,方法 1100 不一定需要实施列出的所有步骤来实现块的帧内预测。例如,如果需要,可以将决策块 1120 从方法 1100 中移除,因此方法 1100 可以配置成总是或从不执行残余像素的量化。

[0069] 图 12 图示了网络单元 1200 的一项实施例,网络单元 1200 可以包括编码器和解码器(编解码器),所述编码器和解码器可以如上文所述,例如,在网络或系统内对视频块进行处理。网络单元 1200 可以包括:多个输入端口 1210 和 / 或接收器单元(Rx) 1212,用于从其他网络单元或组件中接收数据;逻辑单元或处理器 1220,用于处理数据并且确定将数据发送到哪个网络单元;以及多个输出端口 1230 和 / 或发射器单元(Tx) 1232,用于将数据传输到其他网络单元。逻辑单元或处理器 1220 可以用于实施本文所描述的任何方案,例如,变换略过编码方案 500、无量化变换编码方案 700、无变换量化编码方案 900 以及基于像素的帧内预测方法 1100。逻辑单元 1220 可以使用硬件、软件或这两者来实施。

[0070] 上文所述的方案可以在任何通用网络组件上实施,例如,特定的计算机或网络组件,其具有足够的处理能力、存储资源以及网络吞吐能力以处理其上的必要工作量。图 13 图示了典型的通用网络组件或计算机系统 1300 的示意图,其适用于实施本文所揭示方法

的一项或多项实施例,例如,变换略过编码方案 500、无量化变换编码方案 700、无变换量化编码方案 900 以及基于像素的帧内预测方法 1100。通用网络组件或计算机系统 1300 包含处理器 1302(可以称为中央处理器单元或 CPU),所述处理器与包含以下项的存储装置通信:辅助存储器 1304、只读存储器(ROM)1306、随机存取存储器(RAM)1308、输入/输出(I/O)装置 1310,以及网络连接装置 1312。尽管处理器 1302 被图示为单个处理器,但是它并非受到此类限制而是可以包括多个处理器。处理器 1302 可以实施为一个或多个 CPU 芯片、核(例如,多核处理器)、现场可编程门阵列(FPGA)、专用集成电路(ASIC),和/或数字信号处理器(DSP),并且/或者可以是一个或多个 ASIC 的一部分。处理器 1302 可以用于实施本文所描述的任何方案,包括变换略过编码方案 500、无量化变换编码方案 700、无变换量化编码方案 900 以及基于像素的帧内预测方法 1100。处理器 1302 可以使用硬件、软件或这两者来实施。

[0071] 辅助存储器 1304 通常由一个或多个磁盘驱动器或磁带驱动器组成,并且用于数据的非易失性存储,而且如果 RAM1308 的容量不足以存储所有工作数据,那么所述辅助存储器则用作溢流数据存储装置。辅助存储器 1304 可以用于存储程序,当选择执行这些程序时,所述程序将加载到 RAM1308 中。ROM1306 用于存储在程序执行期间读取的指令以及可能的数据。ROM1306 为非易失性存储装置,其存储容量相对于辅助存储器 1304 的较大存储容量而言通常较小。RAM1308 用于存储易失性数据,还可能用于存储指令。对 ROM1306 和 RAM1308 二者的存取通常比对辅助存储器 1304 的存取快。

[0072] 揭示至少一项实施例,且所属领域的技术人员对所述实施例和/或所述实施例的特征的变化、组合和/或修改在本发明的范围内。因组合、整合和/或省略所述实施例的特征而产生的替代实施例也在本发明的范围内。在明确陈述数值范围或限制的情况下,应将此些表达范围或限制理解为包含属于明确陈述的范围或限制内的类似量值的重复范围或限制(例如,从约 1 到约 10 包含 2、3、4 等;大于 0.10 包含 0.11、0.12、0.13 等)。例如,每当揭示具有下限 R_1 和上限 R_0 的数值范围时,具体是揭示属于所述范围的任何数字。具体而言,特别揭示所述范围内的以下数字: $R=R_1+k*(R_0-R_1)$,其中 k 为从 1% 到 100% 范围内以 1% 递增的变量,即, k 为 1%、2%、3%、4%、5%、……、70%、71%、72%、……、95%、96%、97%、98%、99% 或 100%。此外,还特定揭示由如上文所定义的两个 R 数字定义的任何数值范围。除非另有说明,否则使用术语大约是指随后数字的 $\pm 10\%$ 。相对于权利要求的任一元素使用术语“任选地”意味着所述元素是需要的,或者所述元素是不需要的,两种替代方案均在所述权利要求的范围内。使用例如“包括”、“包含”和“具有”等较广术语应被理解为提供对例如“由…组成”、“基本上由…组成”以及“大体上由…组成”等较窄术语的支持。因此,保护范围不受上文所陈述的描述限制,而是由所附权利要求书界定,所述范围包含所附权利要求书的标的物的所有等效物。每一和每个权利要求作为进一步揭示内容并入说明书中,且所附权利要求书是本发明的实施例。所述揭示内容中的参考的论述并不是承认其为现有技术,尤其是具有在本申请案的在先申请优先权日期之后的公开日期的任何参考。本发明中所引用的所有专利、专利申请案和公开案的揭示内容特此以引用的方式并入本文本中,其提供补充本发明的示例性、程序性或其他细节。

[0073] 虽然本发明中已提供若干项实施例,但是应理解,在不脱离本发明的精神或范围的前提下,所公开的系统和方法可以按许多其他特定形式来体现。本发明的实例应被视为

说明性的而非限制性的,且本发明不限于本文所给出的细节。例如,各种元件或组件可以在另一系统中组合或集成,或某些特征可以省略或不实施。

[0074] 另外,在不脱离本发明的范围的情况下,各种实施例中描述和说明为离散或单独的技术、系统、子系统和方法可以与其他系统、模块、技术或方法组合或整合。展示或论述为彼此耦合或直接耦合或通信的其他项目也可以以电方式、机械方式或其他方式通过某一接口、装置或中间组件间接地耦合或通信。其他变化、替代和改变的实例可以由所属领域的技术人员确定,并且可以在不脱离本文本所揭示的范围和精神的情况下作出。

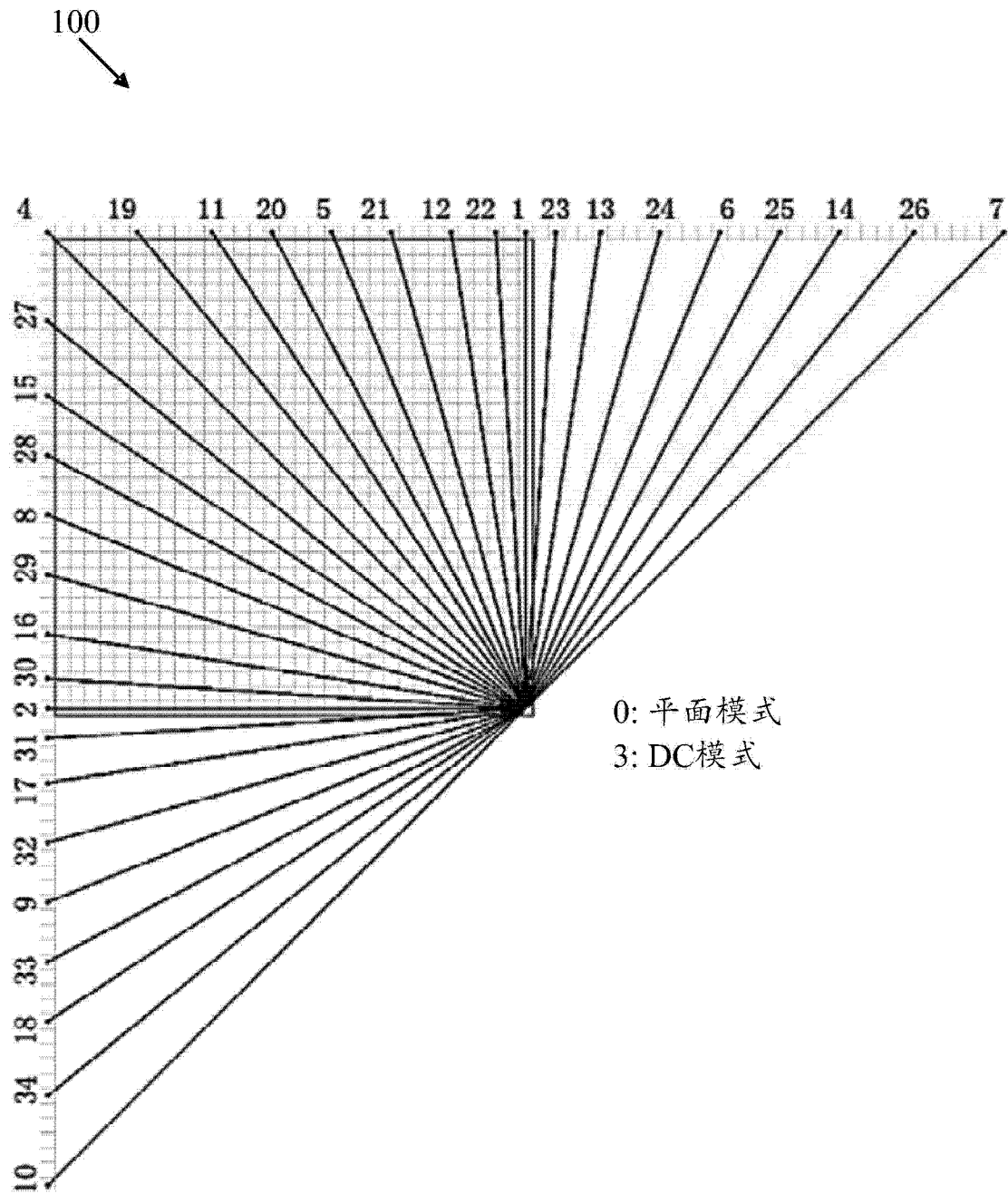


图 1

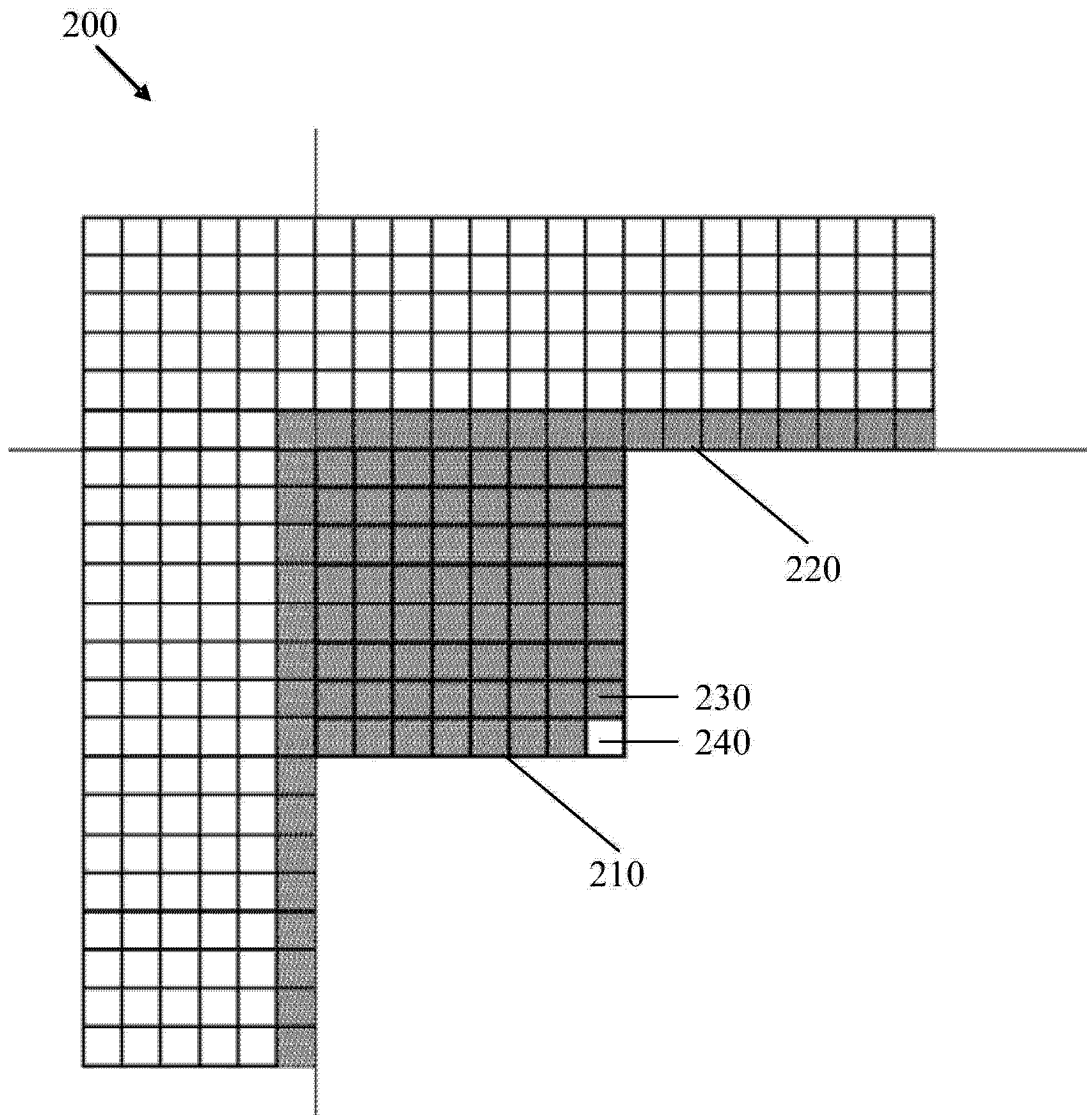


图 2

$XY(0)$	$X(0)$	$X(1)$	$X(2)$	$X(3)$
$Y(0)$	$Z(0, 0)$	$Z(0, 1)$	$Z(0, 2)$	$Z(0, 3)$
$Y(1)$	$Z(1, 0)$	$Z(1, 1)$	$Z(1, 2)$	$Z(1, 3)$
$Y(2)$	$Z(2, 0)$	$Z(2, 1)$	$Z(2, 2)$	$Z(2, 3)$
$Y(3)$	$Z(3, 0)$	$Z(3, 1)$	$Z(3, 2)$	$Z(3, 3)$

图 3

$XY(0)$	$X(0)$	$X(1)$	$X(2)$	$X(3)$
	↓	↓	↓	↓
$Y(0)$	↓	↓	↓	↓
	↓	↓	↓	↓
$Y(1)$	↓	↓	↓	↓
	↓	↓	↓	↓
$Y(2)$	↓	↓	↓	↓
	↓	↓	↓	↓
$Y(3)$	↓	↓	↓	↓

图 4

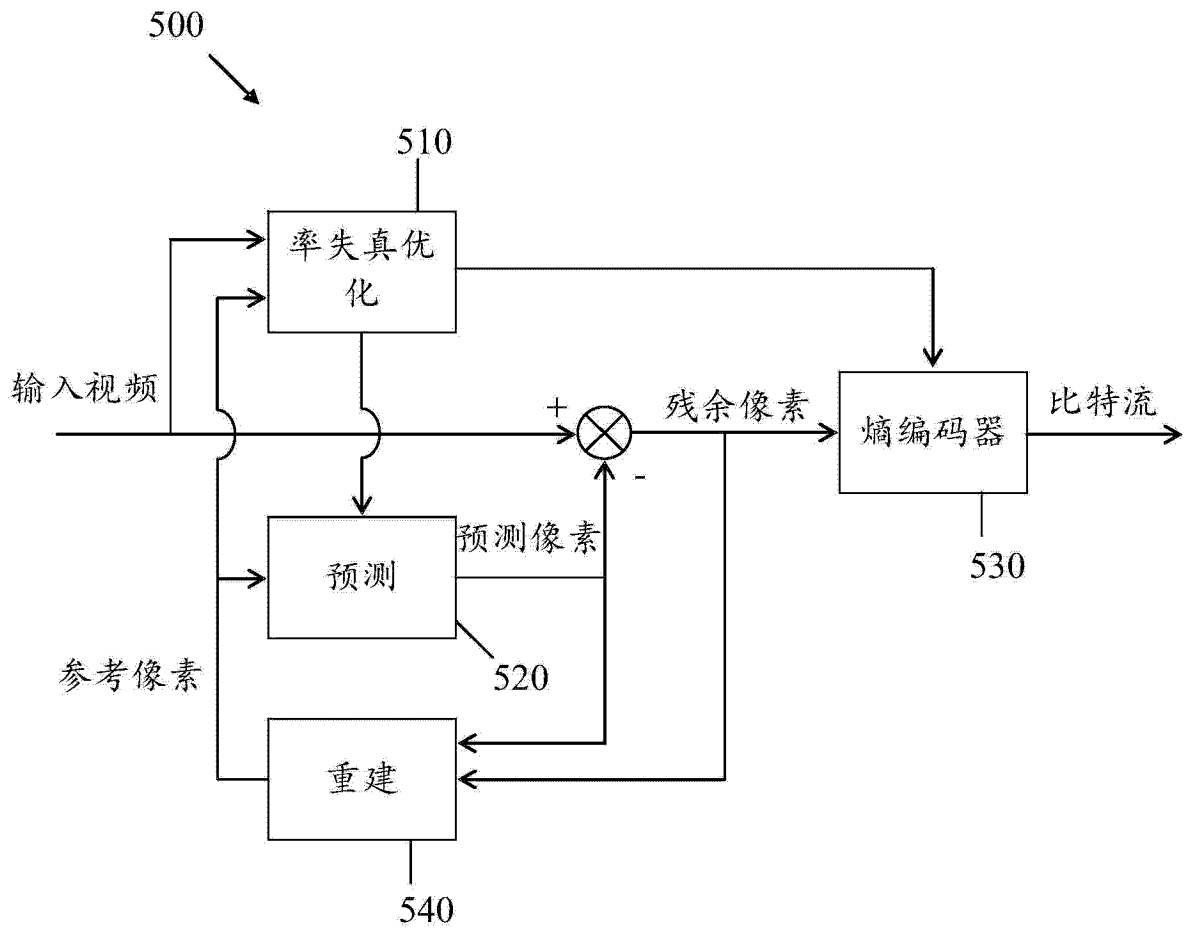


图 5

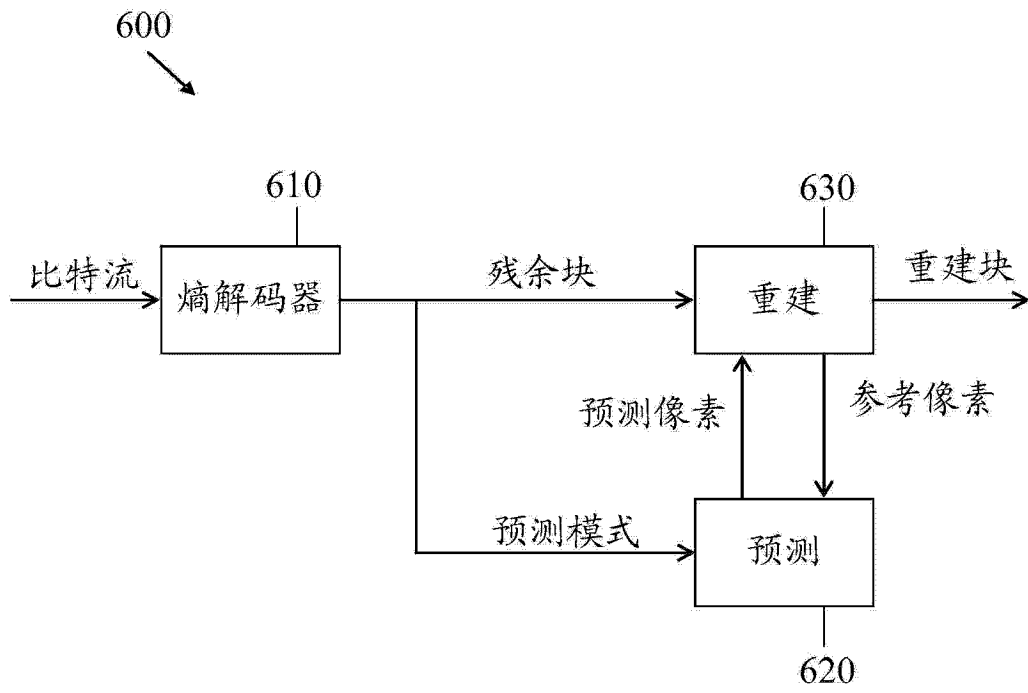


图 6

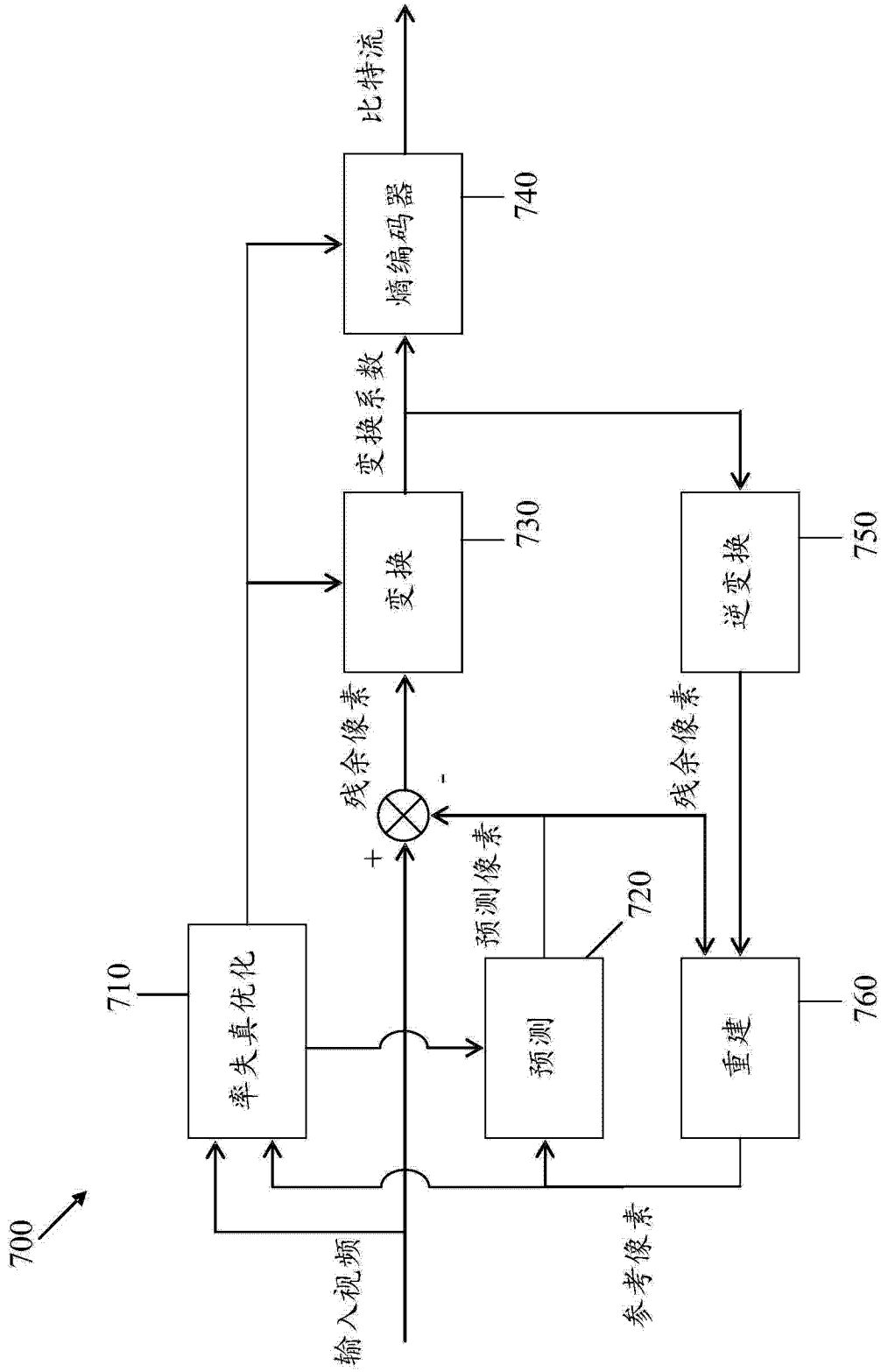


图 7

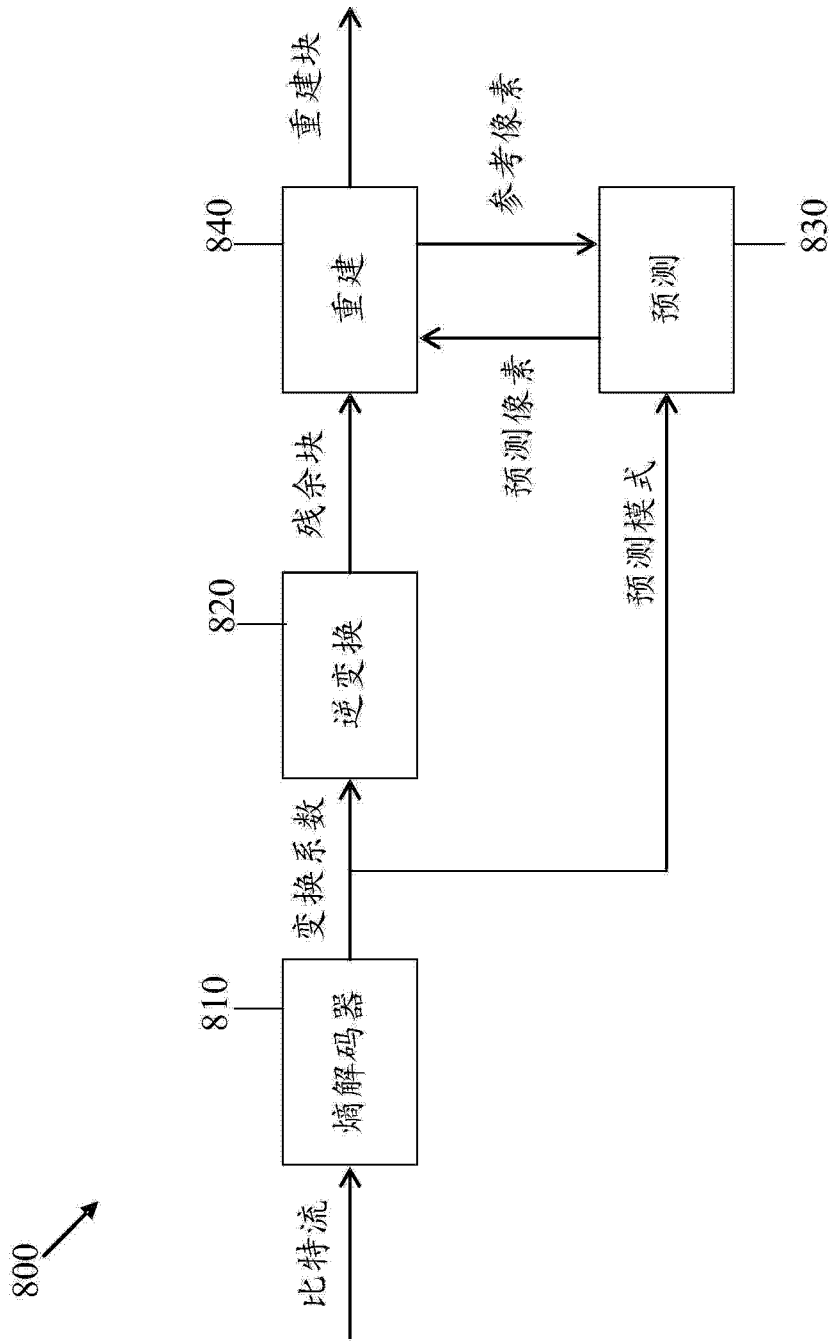


图 8

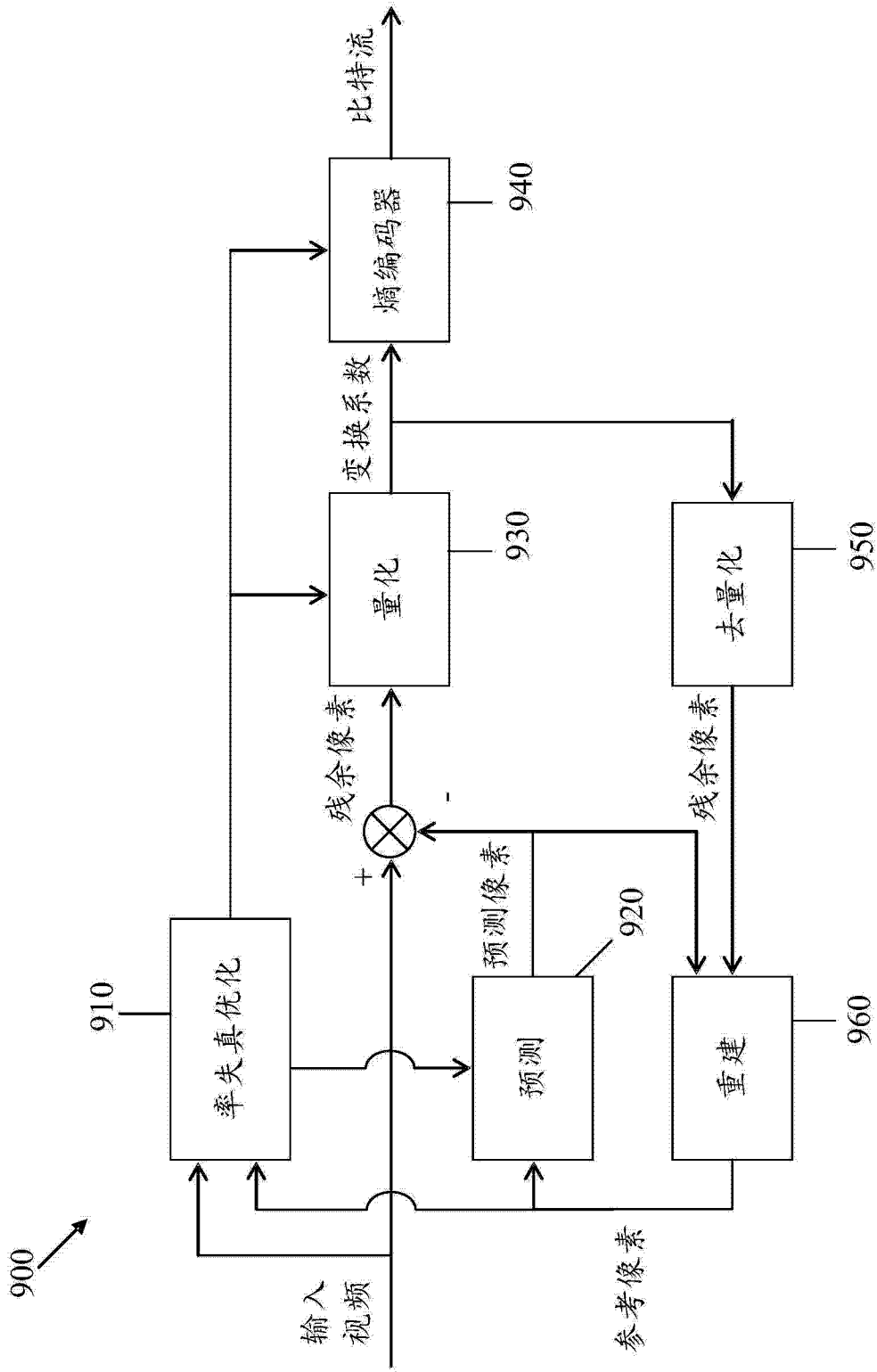


图 9

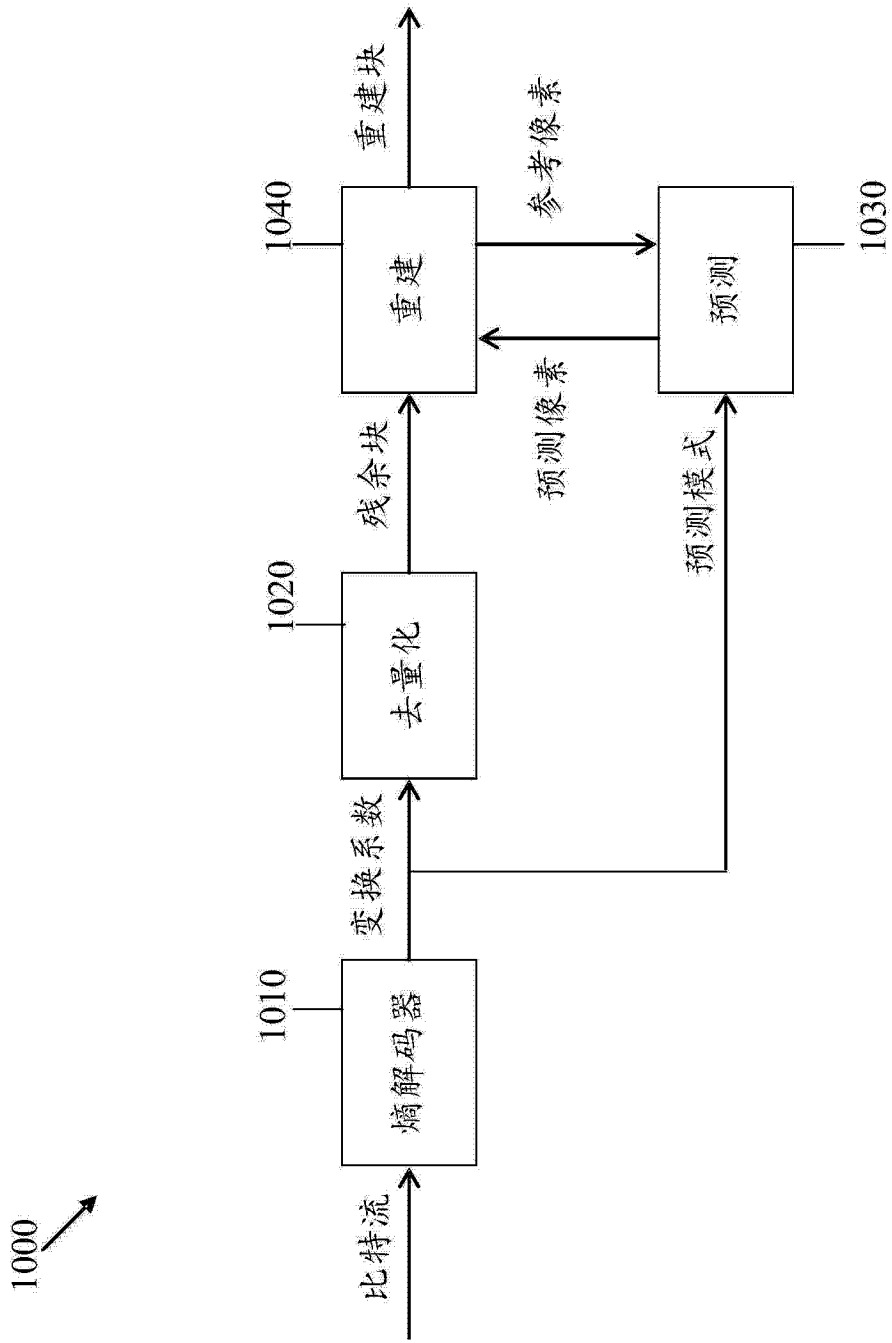


图 10

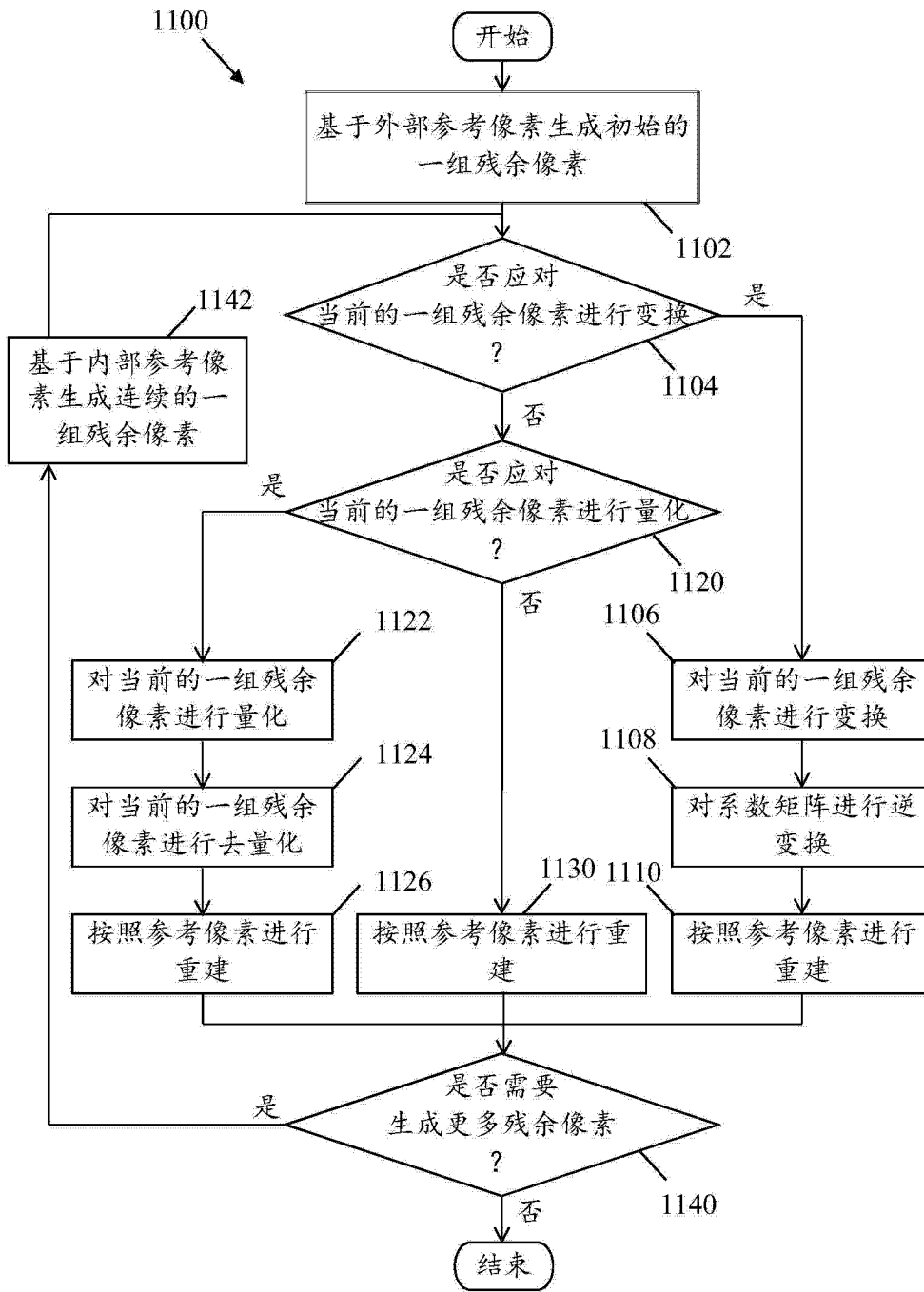


图 11

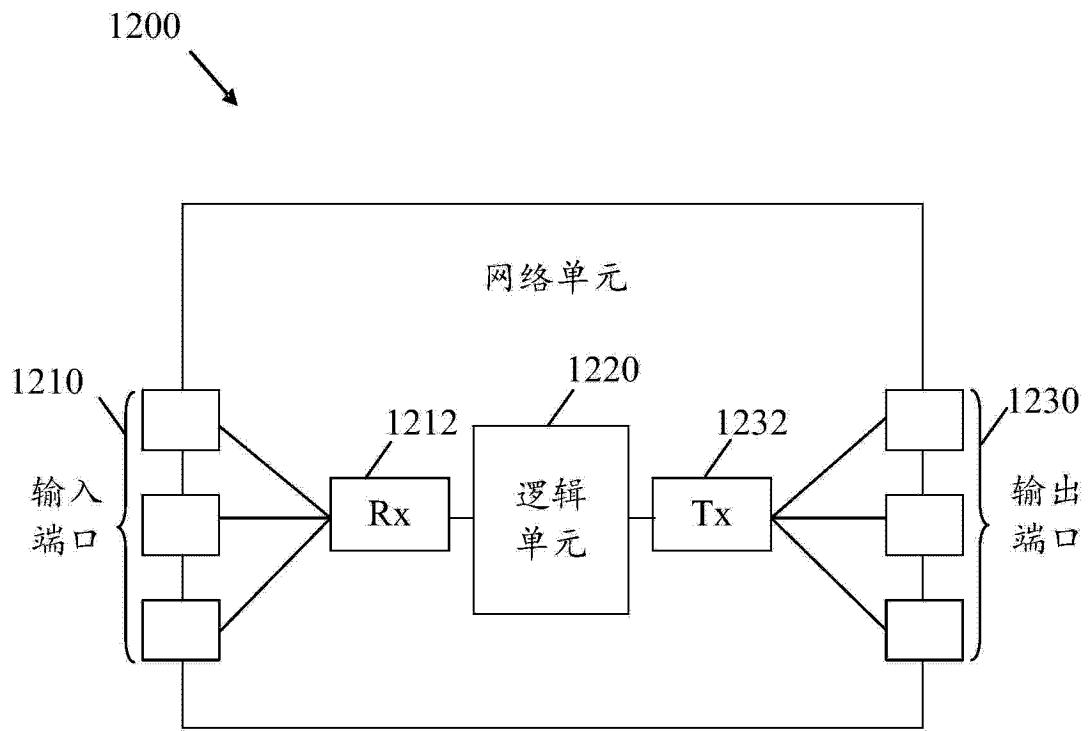


图 12

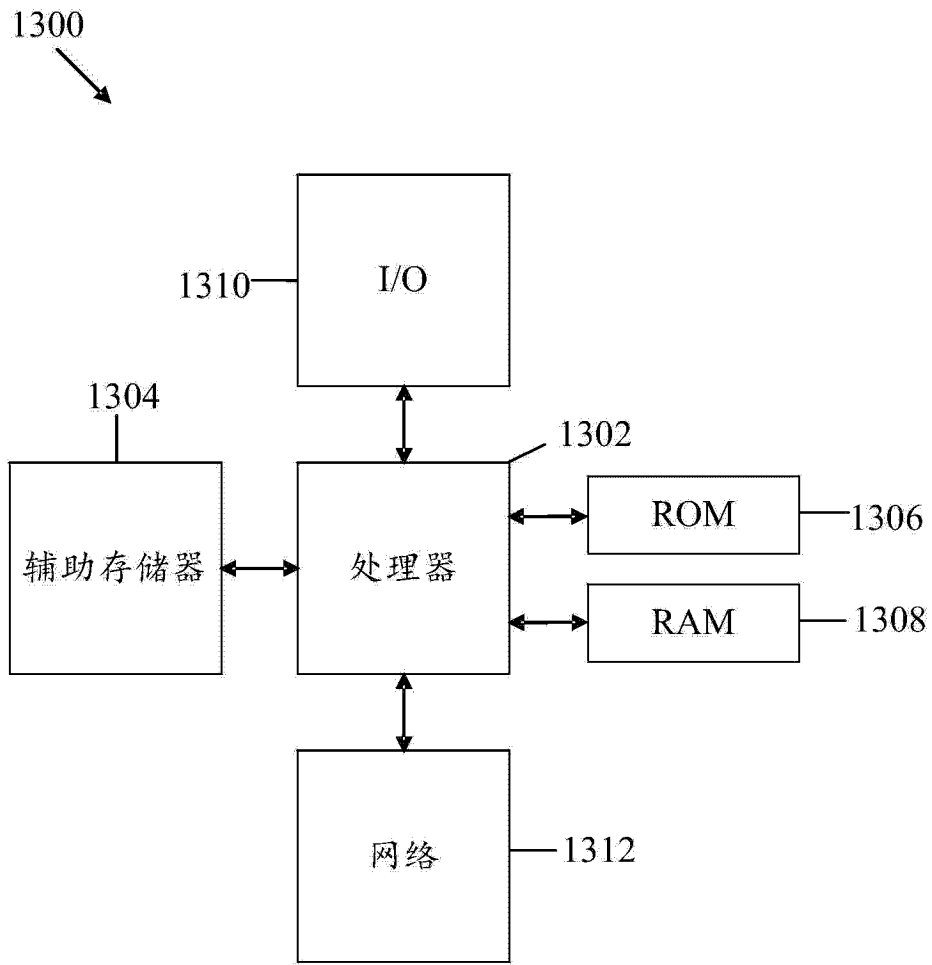


图 13