

11398

申请公布号 CN 115362683 A

H04N 19/167 (2006.01)

H04N 19/172 (2006.01)

H04N 19/174 (2006.01)

(56) 对比文件

Robert Skupin等.AHG12: Sub-bitstream extraction/merging friendly slice address signalling.《Joint Video Experts Team (JVET)》.2018,第3章.

Rickard Sjöberg等.AHG12: On slice address signaling.《Joint Video Experts Team (JVET) 》.2019,参见第2章.

Jonatan Samuelsson等.AHG9: Picture Header in Slice Header.《Joint Video Experts Team (JVET)》.2020,参见第4章.

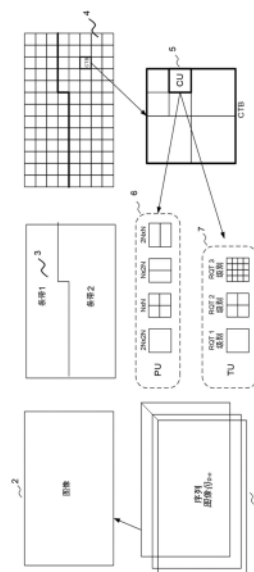
审查员 罗容

权利要求书2页 说明书46页 附图14页

(72)发明人 G·拉罗彻 内尔·奥德拉奥果  
P·乌诺

用于视频编码和解码的方法、解码器、编码器、计算机程序产品和计算机可读存储介质

提供一种从位流中解码视频数据的方法,所述位流包括与一个或多于一个条带相对应的视频数据。各个条带可以包括一个或多于一个区块。位流包括图片头部和条带头部,图片头部包括在对一个或多于一个条带进行解码时要使用的句法元素,条带头部包括要使用的句法元素。对条带进行解码包括对句法元素进行解析。在条带包含多个区块的情况下,如果句法元素被解析并指示在条带头部中用信号通知图片头部,则省略对指示条带的地址的句法元素的解析。使用所述句法元素对位流进行解码。



1. 一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中,各个条带能够包括一个或多个区块,

其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及

其中,所述方法包括:

对句法元素进行解析;

根据图片参数集中的标志的值从所述图片头部对加权预测参数进行解码,其中,当所述值为1时,所述标志指示所述加权预测参数能够存在于所述图片头部中;

在条带包括多个区块的情况下,如果被解析的第二句法元素指示在所述条带头部中存在图片头部,则省略对指示所述条带的地址的第一句法元素的解析;以及

使用被解析出的句法元素从所述位流中对所述视频数据进行解码。

2. 根据权利要求1所述的方法,其中,在要将光栅扫描条带模式用于对条带进行解码时,要进行所述省略。

3. 根据权利要求1或2所述的方法,其中,如果所述第二句法元素指示在所述条带头部中存在所述图片头部,则省略对表示从所述条带中的区块的数量减1的结果的第三句法元素的解析,并且无论所述条带中的区块的数量如何,都将所述第三句法元素的值推断为等于0。

4. 一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的所述视频数据,其中,各个条带能够包括一个或多个区块,

其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行编码时要使用的句法元素,以及

所述方法包括:

确定用于对所述视频数据进行编码的一个或多个句法元素;

根据图片参数集中的标志的值在所述图片头部中对加权预测参数进行编码,其中,当所述标志的值为1时,所述标志指示所述加权预测参数能够存在于所述图片头部中;

在条带包括多个区块的情况下,如果第二句法元素指示在所述条带头部中存在图片头部,则省略对指示所述条带的地址的第一句法元素的编码;以及

使用一个或多个句法元素对所述视频数据进行编码。

5. 根据权利要求4所述的方法,其中,在要将光栅扫描条带模式用于对条带进行编码时,要进行所述省略。

6. 根据权利要求4或5所述的方法,其中,如果所述第二句法元素指示在所述条带头部中存在所述图片头部,则省略对表示从所述条带中的区块的数量减1的结果的第三句法元素的编码,并且无论所述条带中的区块的数量如何,都将所述第三句法元素的值推断为等于0。

7. 一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的所述视频数据,其中,各个条带能够包括一个或多个区块,

其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条

带进行解码时要使用的句法元素,所述条带头部包括在对条带进行编码时要使用的句法元素,

所述位流被约束,使得:

在所述位流包括具有指示图片包括多个区块的值的句法元素、并且所述位流包括指示在所述条带头部中用信号通知图片头部的句法元素的情况下,所述位流还包括指示将不对指示条带的地址的句法元素进行解析的句法元素,所述方法包括使用所述句法元素对所述视频数据进行编码,并且所述方法包括:

根据图片参数集中的标志的值在所述图片头部中对加权预测参数进行编码,其中,当所述标志的值为1时,所述标志指示所述加权预测参数能够存在于所述图片头部中。

8. 一种用于从位流中解码视频数据的解码器,所述解码器被配置为进行根据权利要求1至3中任一项所述的方法。

9. 一种用于将视频数据编码到位流中的编码器,所述编码器被配置为进行根据权利要求4至7中任一项所述的方法。

10. 一种计算机程序产品,其包括计算机程序,所述计算机程序在执行时使得进行根据权利要求1至7中任一项所述的方法。

11. 一种计算机可读存储介质,其存储有计算机程序,所述计算机程序在执行时使得进行根据权利要求1至7中任一项所述的方法。

## 用于视频编码和解码的方法、解码器、编码器、计算机程序产品和计算机可读存储介质

### 技术领域

[0001] 本发明涉及视频编码和解码,并且具体地涉及用于位流中的高级句法。

### 背景技术

[0002] 近来,联合视频专家组(JVET)(由MPEG和ITU-T第16研究组VCEG组成的合作团队)开始研究一种称为多功能视频编码(VVC)的新视频编码标准。VVC的目标是在现有HEVC标准上提供压缩性能的显著改进(即,通常是以前的两倍)并在2020年完成。主要目标应用和服务包括但不限于360度和高动态范围(HDR)视频。总之,JVET使用独立测试实验室进行的正式主观测试来评价了来自32个组织的反馈。一些建议表明,当与使用HEVC相比时,压缩效率通常提高40%或更多。在超高清(UHD)视频测试材料上显示了特定的效果。因此,针对最终标准,我们可以预期压缩效率的提高将远远超过作为目标的50%。

[0003] JVET探索模型(JEM)使用所有HEVC工具且已引入数个新工具。这些改变需要改变位流的结构,特别是可能对位流的总位速率产生影响的高级句法。

### 发明内容

[0004] 本发明涉及对高级句法结构的改进,这使得复杂性降低而编码性能没有任何降低。

[0005] 在根据本发明的第一方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,所述方法包括:解析所述句法元素,以及在条带(或图片)包括多个区块的情况下,如果解析了指示在条带头部中用信号通知图片头部的句法元素,则省略对指示条带的地址的句法元素进行解析;以及使用所述句法元素对所述位流进行解码。在根据本发明的另一方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,所述方法包括:解析所述句法元素,并且在条带或图片包括多个区块的情况下,如果解析了指示在条带头部中用信号通知图片头部的句法元素,则省略对指示条带的地址的句法元素进行解析;以及使用所述句法元素对所述位流进行解码。在根据本发明的另一附加方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,所述位流被约束以使得在所述位流包括具有指示条带或图片包括多个区块

的值的句法元素并且所述位流包括指示在所述条带头部中用信号通知图片头部的句法元素的情况下,所述位流还包括指示将不解析指示条带的地址的句法元素的句法元素,所述方法包括使用所述句法元素对所述位流进行解码。

[0006] 因此,当图片头部在片头部中时,不解析条带地址,这降低了位速率,特别是对于低延迟和低位速率应用。此外,当在条带头部中用信号通知图片时,可以降低解析复杂性。

[0007] 在实施例中,(仅)在光栅扫描条带模式要被用于对条带进行解码时,才要进行省略。这降低了解析复杂性,但仍然允许一些位速率降低。

[0008] 省略还可以包括省略对指示条带中的区块的数量的句法元素进行解析。因此,可以实现位速率的进一步降低。

[0009] 在第二方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述解码包括:对一个或多个句法元素进行解析,并且在条带(或图片)包含多个区块的情况下,如果解析了指示在所述条带头部中用信号通知所述图片头部的句法元素,则省略对指示所述条带中的区块的数量的句法元素进行解析;以及使用所述句法元素对所述位流进行解码。在另一方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述解码包括:对一个或多个句法元素进行解析,并且在条带或图片包含多个区块的情况下,如果解析了指示在所述条带头部中用信号通知所述图片头部的句法元素,则省略对指示所述条带中的区块的数量的句法元素进行解析;以及使用所述句法元素对所述位流进行解码。在本发明的另一方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,所述位流被约束以使得在所述位流包括具有指示条带或图片包含多个区块的值的句法元素并且所述位流包括指示在所述条带头部中用信号通知所述图片头部的句法元素的情况下,所述位流还包括指示将不解析指示所述条带中的多个区块的句法元素的句法元素,所述方法包括使用所述句法元素对所述位流进行解码。

[0010] 因此,可以降低位速率,这特别有利于不需要发送多个区块的低延迟和低位速率应用。

[0011] 可以(仅)在光栅扫描条带模式要被用于对条带进行解码时,才进行省略。这降低了解析复杂性,但仍然允许一些位速率降低。

[0012] 该方法还可以包括:解析指示所述图片中的区块的数量的句法元素,并且基于所解析的句法元素所指示的所述图片中的区块的数量来确定所述条带中的区块的数量。这是有利的,因为允许在条带头部中用信号通知图片头部而不需进一步的用信号通知的情况下容易地预测条带中的区块的数量。

[0013] 省略还可以包括省略对指示条带的地址的句法元素进行解析。因此,可以进一步降低位速率。

[0014] 在本发明的第三方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述解码包括:解析一个或多个句法元素,并且在条带(或图片)包括多个区块的情况下,如果条带中的区块的数量等于图片中的区块的数量,则省略对指示条带地址的句法元素进行解析;以及使用所述句法元素对所述位流进行解码。这利用了以下见解:如果条带中的区块的数量等于图片中的区块的数量,则确保当前图片仅包含一个条带。因此,通过省略条带地址,可以改进位速率并且降低解析和/或编码的复杂性。

[0015] 可以(仅)在光栅扫描条带模式要被用于对条带进行解码时,才进行省略。因此,可以降低复杂性,同时仍然提供一些位速率降低。

[0016] 解码还可以包括:在条带中解析指示条带中的区块的数量的句法元素;以及在图片参数集中解析指示图片中的区块的数量的句法元素,其中省略对指示条带地址的句法元素进行解析是基于所解析的句法元素的。

[0017] 解码还可以包括:在用于用信号通知条带地址的一个或多个句法元素之前,解析条带中的指示条带中的区块的数量的句法元素。

[0018] 解码还可以包括:在条带中解析指示是否在条带头部中用信号通知图片头部的句法元素,并且如果所解析的句法元素指示在条带头部中用信号通知图片头部,则确定(推断)为条带中的区块的数量等于图片中的区块的数量。

[0019] 在第四方面,提供一种从位流中解码视频数据的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述解码包括:对一个或多个句法元素进行解析,并且在句法元素指示针对条带启用光栅扫描解码模式的情况下,从所述一个或多个句法元素解码条带地址和条带中的区块的数量其中至少之一,其中在针对条带启用光栅扫描解码模式的情况下从所述一个或多个句法元素解码条带地址和条带中的区块的数量其中至少之一不取决于图片中的区块的数量;以及使用所述句法元素对所述位流进行解码。因此,可以降低条带头部的解析复杂性。

[0020] 在根据本发明的第五方面,提供一种包括第一方面和第二方面的方法。

[0021] 在根据本发明的第六方面,提供一种包括第一方面、第二方面和第三方面的方法。

[0022] 根据本发明的第七方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中,各个条带可以包括一个或多个区块,其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行编码时要使用的句法元素,以及所述编码包括:确定用于对视频数据进行编码的一个或多个句法元素,并且在条带(或图片)包括多个区块的情况下,如果句法元素指示在条带头部中用信号通知图片头部,则省略对指示条带的地址的句法元素进行编码;以及使用所述句法元素对所述视频数

据进行编码。根据本发明的附加方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的所述视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行编码时要使用的句法元素,以及所述编码包括:确定用于对所述视频数据进行编码的一个或多个句法元素,并且在条带或图片包括多个区块的情况下,如果句法元素指示在条带头部中用信号通知图片头部,则省略对指示条带的地址的句法元素进行编码;以及使用所述句法元素对所述视频数据进行编码。根据本发明的附加补充方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行编码时要使用的句法元素,所述位流被约束以使得在所述位流包括具有指示条带或图片包括多个区块的值的句法元素并且所述位流包括指示在所述条带头部中用信号通知图片头部的句法元素的情况下,所述位流还包括指示将不解析指示条带的地址的句法元素的句法元素;所述方法包括使用所述句法元素对所述视频数据进行编码。

[0023] 在一个或多个实施例中,(仅)在光栅扫描条带模式被用于对条带进行编码时,才进行省略。

[0024] 省略还可以包括省略对指示条带中的区块的数量的句法元素进行编码。

[0025] 根据本发明的第八方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中,各个条带可以包括一个或多个区块,其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述编码包括:确定用于对所述视频数据进行编码的一个或多个句法元素,并且在条带包括多个区块的情况下,如果指示在所述条带头部中用信号通知所述图片头部的句法元素被确定用于编码,则省略对指示所述条带中的区块的数量的句法元素进行编码;以及使用所述句法元素对所述视频数据进行编码。根据本发明的另一附加方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述编码包括:确定用于对所述视频数据进行编码的一个或多个句法元素,并且在条带或图片包括多个区块的情况下,如果指示在条带头部中用信号通知图片头部的句法元素被确定用于编码,则省略对指示条带中的区块的数量的句法元素进行编码;以及使用所述句法元素对所述视频数据进行编码。根据本发明的另一补充方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中各个条带可以包括一个或多个区块,其中所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,所述位流被约束以使得在位流包括具有指示条带或图片包括多个区块的值的句法元素并且位流包括被确定用于编码的指示在条带头部中用信号通知图片头部的句法元素的情况下,位流

还包括指示将不解析指示条带中的区块的数量的句法元素的句法元素,所述方法包括使用所述句法元素对所述视频数据进行编码。

[0026] 在实施例中,(仅)在光栅扫描条带模式要被用于对条带进行编码时,才进行省略。

[0027] 编码还可以包括对指示图片中的区块的数量的句法元素进行编码,其中条带中的区块的数量是基于由所解析的句法元素指示的图片中的区块的数量的。

[0028] 省略还可以包括省略对指示条带的地址的句法元素进行编码。

[0029] 根据本发明的第九方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中,各个条带可以包括一个或多个区块,其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述编码包括:确定一个或多个句法元素,并且在条带(或图片)包括多个区块的情况下,如果条带中的区块的数量等于图片中的区块的数量,则省略对指示条带地址的句法元素进行编码;以及使用所述句法元素对所述视频数据进行编码。

[0030] 在一个或多个实施例中,(仅)在光栅扫描条带模式要被用于对条带进行解码时,才进行省略。

[0031] 编码还可以包括在条带中编码指示条带中的区块的数量的句法元素;以及在图片参数集中编码指示图片中的区块的数量的句法元素,其中省略或不省略对指示条带地址的句法元素进行编码是基于所编码的句法元素的值的。

[0032] 编码还可以包括:在用于用信号通知条带地址的一个或多个句法元素之前,在条带中编码指示条带中的区块的数量的句法元素。

[0033] 编码还可以包括在条带中编码指示是否在条带头部中用信号通知图片头部的句法元素,并且如果要编码的句法元素指示在条带头部中用信号通知图片头部,则确定为条带中的区块的数量等于图片中的区块的数量。

[0034] 根据本发明的第十方面,提供一种将视频数据编码到位流中的方法,所述位流包括与一个或多个条带相对应的视频数据,其中,各个条带可以包括一个或多个区块,其中,所述位流包括图片头部和条带头部,所述图片头部包括在对一个或多个条带进行解码时要使用的句法元素,所述条带头部包括在对条带进行解码时要使用的句法元素,以及所述编码包括:确定用于对所述视频数据进行编码的一个或多个句法元素,以及在被确定用于编码的句法元素指示针对条带启用光栅扫描解码模式的情况下,对指示条带地址和条带中的区块的数量其中至少之一的句法元素进行编码,其中在针对条带启用光栅扫描解码模式的情况下,从一个或多个句法元素中解码条带地址和条带中的区块的数量其中至少之一不取决于图片中的区块的数量;以及使用所述句法元素对所述位流进行编码。

[0035] 在根据本发明的第十一方面,提供一种包括第七方面和第八方面的方法。

[0036] 在根据本发明的第十二方面,提供一种包括第七方面、第八方面和第九方面的方法。

[0037] 根据本发明的第十三方面,提供一种从位流中解码视频数据的解码器,该解码器被配置为进行第一方面至第六方面中任一方面的方法。

[0038] 根据本发明的第十四方面,提供一种将视频数据编码到位流中的编码器,该编码



器被配置为进行第七方面至第十二方面中任一方面的方法。

[0039] 根据本发明的第十五方面,提供一种计算机程序,当该计算机程序被执行时使得进行第一方面至第十二方面中任一方面的方法。该程序可以单独提供,或者可以在载体介质上、由载体介质承载或在载体介质中承载。载体介质可以是非暂时性的,例如存储介质,特别是计算机可读存储介质。载体介质也可以是暂时性的,例如信号或其他传输介质。信号可以经由任何合适的网络(包括因特网)传输。

[0040] 本发明的一个方面中的任何特征可以以任何适当的组合应用于本发明的其他方面。特别地,方法方面可以应用于设备方面,反之亦然。

[0041] 此外,以硬件实现的特征可以以软件实现,反之亦然。本文对软件和硬件特征的任何引用均应据此解释。

[0042] 如本文所述的任何设备特征也可以被提供为方法特征,反之亦然。如本文中所使用的,部件加功能特征就其相应结构(诸如适当编程的处理器和相关联的存储器等)方面可以被替代性地表达。

[0043] 还应当理解,可以独立地实现、提供和/或使用在本发明的任何方面中描述和定义的各种特征的特定组合。

## 附图说明

[0044] 现在将通过示例的方式参考附图,在附图中:

[0045] 图1是用于说明HEVC和VVC中使用的编码结构的图;

[0046] 图2是示意性地示出可以实现本发明的一个或多于一个实施例的数据通信系统的框图;

[0047] 图3是示出可以实现本发明的一个或多于一个实施例的处理装置的组件的框图;

[0048] 图4是示出根据本发明实施例的编码方法的步骤的流程图;

[0049] 图5是示出根据本发明实施例的解码方法的步骤的流程图;

[0050] 图6示出示例性编码系统VVC中的位流的结构;

[0051] 图7示出示例性编码系统VVC中的位流的另一结构;

[0052] 图8示出亮度建模色度缩放(Luma Modelling Chroma Scaling(LMCS));

[0053] 图9示出LMCS的子工具;

[0054] 图10是当前VVC草案标准的光栅扫描条带模式和矩形条带模式的图;

[0055] 图11示出根据本发明实施例的包括编码器或解码器和通信网络的系统的图;

[0056] 图12是用于实现本发明的一个或多于一个实施例的计算装置的示意性框图;

[0057] 图13是示出网络照相机系统的图;以及

[0058] 图14是示出智能电话的图。

## 具体实施方式

[0059] 图1涉及在高效率视频编码(HEVC)视频标准中使用的编码结构。视频序列1由一系列数字图像*i*组成。各个这样的数字图像由一个或多于一个矩阵表示。矩阵系数表示像素。

[0060] 序列的图像2可以被分割成条带3。在一些情况下,一条带可以构成图像整体。这些条带被分割成非重叠编码树单元(CTU)。编码树单元(CTU)是高效率视频编码(HEVC)视频标

准的基本处理单元,并且概念性地在结构上与若干先前视频标准中使用的宏块单元相对应。CTU有时也被称为最大编码单元(LCU)。CTU具有亮度和色度分量部分,各个分量部分被称为编码树块(CTB)。这些不同的颜色分量未在图1中示出。

[0061] CTU通常大小为64像素×64像素。可以使用二叉树分解进而将各CTU迭代地分割成较小的可变大小编码单元(CU)5。

[0062] 编码单元是基本编码元素,并且由被称为预测单元(PU)和变换单元(TU)的两种子单元构成。PU或TU的最大大小等于CU大小。预测单元与CU的用于像素值的预测的分区相对应。将CU分区成PU的各种不同分区是可能的,如6所示,包括分成4个正方形PU的分区、以及分成2个矩形PU的两个不同分区。变换单元是使用DCT进行空间变换的基本单元。CU可以基于二叉树表示7分区成TU。

[0063] 各条带嵌入一个网络抽象层(NAL)单元中。另外,视频序列的编码参数存储在称为参数集的专用NAL单元中。在HEVC和H.264/AVC中,采用两种参数集NAL单元:第一,序列参数集(SPS)NAL单元,其收集在整个视频序列期间不变的所有参数。通常,它处理编码配置文件、视频帧的大小和其他参数。第二,图片参数集(PPS)NAL单元,其包括可以从序列的一个图像(或帧)改变为其他图像(或帧)的参数。HEVC还包括视频参数集(VPS)NAL单元,其包含描述位流的总体结构的参数。VPS是HEVC中定义的新类型的参数集,并且应用于位流的所有层。层可以包含多个时间子层,并且所有版本1的位流限定于单个层。HEVC具有用于可缩放性和多视图的某些分层扩展,并且这些扩展将允许具有向后兼容的版本1的基础层的多个层。

[0064] 在通用视频编码(VVC)的当前定义中,存在图片的分区的三种高级可能性:子图片、条带和区块。各自具有其自己的特征和有用性。分区成子图片以进行视频的区域的空间提取和/或合并。分区成条带是基于与先前标准相似的概念的,并且对应于用于视频传输的分包(即使其可以用于其他应用)。分区成区块在概念上是编码器并行化工具,因为其将图片拆分成图片的(几乎)相同大小的独立编码区域。但该工具也可以用于其他应用。

[0065] 由于可以一起使用图片分区的这三种高级可用的可能方式,因此存在针对其的使用的若干模式。如在VVC的当前草案规范中所定义,定义条带的两个模式。对于光栅扫描条带模式,条带包含图片的区块光栅扫描中的完整区块序列。当前VVC规范中的该模式在图10(a)中示出。如图中所示,图片包含示出为分区成12个条带和3个光栅扫描条带的18乘12个亮度CTU。

[0066] 对于第二个(矩形条带模式),条带包含共同来自图片的矩形区域的数个完整区块。当前VVC规范中的该模式在图10(b)中示出。在该示例中,图片具有示出为分区成24个区块和9个矩形条带的18乘12个亮度TU。

[0067] 图2例示可以实现本发明的一个或多于一个实施例的数据通信系统。数据通信系统包括传输装置(在这种情况下是服务器201),其可操作以经由数据通信网络200将数据流的数据包传输至接收装置(在这种情况下是客户端终端202)。数据通信网络200可以是广域网(WAN)或局域网(LAN)。这种网络可以是例如无线网络(Wifi/802.11a或b或g)、以太网网络、互联网网络或由若干不同网络组成的混合网络。在本发明的特定实施例中,数据通信系统可以是数字电视广播系统,其中服务器201将相同的数据内容发送到多个客户端。

[0068] 由服务器201提供的数据流204可以由表示视频和音频数据的多媒体数据组成。在

本发明的一些实施例中,音频和视频数据流可以分别由服务器201使用麦克风和照相机来捕获。在一些实施例中,数据流可以存储在服务器201上或由服务器201从其他数据提供商接收,或在服务器201处生成。服务器201设置有用对视频和音频流进行编码的编码器,特别是用以提供用于传输的压缩位流,该压缩位流是作为编码器的输入所呈现的数据的更紧凑表示。

[0069] 为了获得更好的传输数据的质量与传输数据的量的比率,可以例如根据HEVC格式或H.264/AVC格式来压缩视频数据。

[0070] 客户端202接收所传输的位流,并且解码重建的位流,以在显示装置上再现视频图像和利用扬声器再现音频数据。

[0071] 尽管在图2的示例中考虑了流式传输场景,但将认识到,在本发明的一些实施例中,可以使用例如介质存储装置(诸如光盘等)来进行编码器与解码器之间的数据通信。

[0072] 在本发明的一个或多个实施例中,视频图像与表示要应用到图像的重建像素的补偿偏移的数据一同传输,以在最终图像中提供经滤波的像素。

[0073] 图3示意性地例示被配置为实现本发明的至少一个实施例的处理装置300。处理装置300可以是诸如微计算机、工作站或轻型便携式装置等的装置。

[0074] 装置300包括通信总线313,其连接到:

[0075] -表示为CPU的中央处理单元311,诸如微处理器等;

[0076] -表示为ROM的只读存储器306,其用于存储实现本发明的计算机程序;

[0077] -用于存储本发明实施例的方法的可执行代码的表示为RAM的随机访问存储器312,以及适于记录变量和参数的寄存器,该变量和参数是根据本发明实施例实现对数字图像序列进行编码的方法和/或对位流进行解码的方法所需的;以及

[0078] -连接至通信网络303的通信接口302,通过该通信接口传输或接收要处理的数字数据。

[0079] 可选地,设备300还可以包括以下组件:

[0080] -诸如硬盘等的数据存储部件304,其用于存储实现本发明的一个或多个实施例的方法的计算机程序以及在实现本发明的一个或多个实施例期间所使用或产生的数据;

[0081] -用于盘306的盘驱动器305,该盘驱动器适于从盘306读取数据或将数据写入所述盘;

[0082] -屏幕309,其用于借助于键盘310或任何其他指示装置来显示数据和/或用作与用户交互的图形界面。

[0083] 设备300可以连接到诸如数字照相机320或麦克风308等的各种外围设备,其各自连接到输入/输出卡(未示出)以向设备300提供多媒体数据。

[0084] 通信总线提供设备300中所包括的或连接到设备300的各种元素之间的通信和互操作性。总线的表示不是限制性的,并且特别地,中央处理单元可操作地将指令直接或者借助于设备300的其他元素通信到设备300的任何元素。

[0085] 盘306可以由诸如可重写或不可重写的致密盘(CD-ROM)、ZIP盘或存储卡等的任何信息介质代替,并且一般而言,由微计算机或微处理器可以进行读取的信息存储部件代替,该盘306集成到或不集成到设备中、可能可移动并且适于存储其执行使得能够实现根据本

发明的对数字图像序列进行编码的方法和/或对位流进行解码的方法的一个或多个程序。

[0086] 可执行代码可以存储在只读存储器306中、硬盘304上或可移动数字介质(诸如,例如如前述的盘306等)上。根据一变型,程序的可执行代码可以经由接口302借助于通信网络303来接收,以在执行之前存储在设备300的存储部件之一(诸如硬盘304等)中。

[0087] 中央处理单元311适于控制和指导执行根据本发明的一个或多个程序的指令或软件代码的部分、存储在上述存储部件之一中的指令的执行。在通电时,存储在非易失性存储器中(例如,在硬盘304上或在只读存储器306中)的一个或多个程序被传递到随机访问存储器312中(其然后包含一个或多个程序的可执行代码)以及用于存储实现本发明所必需的变量和参数的寄存器。

[0088] 在该实施例中,设备是使用软件来实现本发明的可编程设备。然而,可替代地,本发明可以以硬件(例如,以专用集成电路或ASIC的形式)来实现。

[0089] 图4例示根据本发明的至少一个实施例的编码器的框图。编码器由所连接的模块表示,各模块适于例如以由装置300的CPU 311执行的编程指令的形式来实现根据本发明的一个或多个实施例的、用于实现对图像序列中的图像进行编码的至少一个实施例的方法的至少一个相应步骤。

[0090] 编码器400接收数字图像 $i_0$ 至 $i_n$ 的原始序列401作为输入。各数字图像由样本(称为像素)集表示。

[0091] 编码器400在实现编码处理之后输出位流410。位流410包括多个编码单元或条带,各条带包括用于对条带编码所用的编码参数的编码值进行传输的条带头部、以及包括编码视频数据的条带主体。

[0092] 模块402将输入数字图像 $i_0$ 至 $i_n$  401分割成像素块。块与图像部分相对应并且可以具有可变大小(例如, $4 \times 4$ 、 $8 \times 8$ 、 $16 \times 16$ 、 $32 \times 32$ 、 $64 \times 64$ 、 $128 \times 128$ 像素、并且还可以考虑若干矩形块大小)。针对各输入块选择编码模式。提供了两个编码模式族:基于空间预测编码(帧内预测)的编码模式和基于时间预测的编码模式(帧间编码、合并、跳过)。测试了可能的编码模式。

[0093] 模块403实现帧内预测处理,其中,通过根据要编码的给定块的相邻像素计算出的预测子来预测所述要编码的块。如果选择了帧内编码,则对所选择的帧内预测子以及给定块与其预测子之间的差的指示进行编码以提供残差。

[0094] 时间预测由运动估计模块404和运动补偿模块405实现。首先,选择来自参考图像集416的参考图像,并且由运动估计模块404选择参考图像的一部分(也被称为参考区域或图像部分),该部分是与要编码的给定块最接近的区域。然后运动补偿模块405使用所选择的区域来预测要编码的块。由运动补偿模块405计算所选择的参考区域与给定块(也称为残差块)之间的差。所选择的参考区域由运动矢量指示。

[0095] 由此,在这两个情况下(空间和时间预测),通过从原始块减去预测来计算残差。

[0096] 在由模块403实现的帧内预测中,对预测方向进行编码。在时间预测中,对至少一个运动矢量进行编码。在由模块404、405、416、418、417实现的帧间预测中,至少一个运动矢量或用于识别这种运动矢量的数据被编码用于时间预测。

[0097] 如果选择帧间预测,则对与运动矢量和残差块有关的信息进行编码。为了进一步

降低位速率,假设运动是同质的,通过相对于运动矢量预测子的差对运动矢量进行编码。由运动矢量预测和编码模块417从运动矢量场418获得运动信息预测子的集合中的运动矢量预测子。

[0098] 编码器400还包括选择模块406,该选择模块用于通过应用编码成本标准(诸如,率-失真标准等)来选择编码模式。为了进一步减少冗余,由变换模块407将变换(诸如DCT等)应用于残差块,然后,所获得的变换数据由量化模块408量化并且由熵编码模块409进行熵编码。最终,正被编码的当前块的编码后的残差块被插入位流410中。

[0099] 编码器400还进行编码图像的解码,以产生用于后续图像的运动估计的参考图像。这使得接收位流的编码器和解码器能够具有相同的参考帧。逆量化模块411进行量化数据的逆量化,之后是逆变换模块412的逆变换。逆帧内预测模块413使用预测信息来确定对于给定块使用哪个预测子,并且逆运动补偿模块414实际上将由模块412获得的残差添加到从参考图像集416获得的参考区域。

[0100] 然后,由模块415应用后滤波以对所重建的像素帧进行滤波。在本发明的实施例中,使用SA0环路滤波器,其中补偿偏移被添加到所重建图像的所重建像素的像素值。

[0101] 图5示出根据本发明实施例的解码器60的框图,解码器60可以用于从编码器接收数据。解码器由所连接的模块表示,各模块适于例如以要由装置300的CPU 311执行的编程指令的形式实现由解码器60实现的方法的相应步骤。

[0102] 解码器60接收包括编码单元的位流600,各编码单元由包含与经编码的参数有关的信息的头部和包含经编码的视频数据的主体组成。下文参考图6更详细地描述VVC中的位流的结构。如关于图4所说明的,针对给定块,在预定数量的位上,对经编码的视频数据进行熵编码,并且对运动矢量预测子的索引进行编码。所接收的经编码的视频数据由模块62进行熵解码。然后残差数据由模块63去量化,之后由模块64应用逆变换以获得像素值。

[0103] 用于指示编码模式的模式数据也被熵解码,并且基于该模式,对图像数据的编码块进行帧内类型解码或帧间类型解码。

[0104] 在帧内模式的情况下,帧内逆预测模块65基于在位流中指定的帧内预测模式来确定帧内预测子。

[0105] 如果模式是帧间,则从位流提取运动预测信息以找到由编码器使用的参考区域。运动预测信息由参考帧索引和运动矢量残差组成。运动矢量预测子被添加到运动矢量残差以由运动矢量解码模块70获得运动矢量。

[0106] 运动矢量解码模块70对通过运动预测编码的各当前块应用运动矢量解码。一旦已获得针对当前块的运动矢量预测子的索引,可以对与当前块相关联的运动矢量的实际值进行解码,并且该实际值用以通过模块66应用逆运动补偿。从参考图像68提取由经解码的运动矢量指示的参考图像部分以应用逆运动补偿66。利用经解码的运动矢量更新运动矢量场数据71,以用于后续解码运动矢量的逆预测。

[0107] 最终,获得经解码的块。后滤波由后滤波模块67应用。解码器60最终提供经解码的视频信号69。

[0108] 图6示出了如JVET\_Q2001-vD中所述的示例性编码系统VVC中的位流的组织。

[0109] 根据VVC编码系统的位流61由句法元素和经编码数据的有序序列组成。句法元素和经编码数据被放置到网络抽象层(NAL)单元601-608中。存在不同的NAL单元类型。网络抽

象层提供将位流封装到不同协议(如RTP/IP(代表实时协议/因特网协议)、ISO基本媒体文件格式等)中的能力。网络抽象层还提供用于抗包丢失的框架。

[0110] NAL单元被分割成视频编码层(VCL)NAL单元和非VCL NAL单元。VCL NAL单元包含实际的经编码视频数据。非VCL NAL单元包含附加信息。该附加信息可以是解码经编码视频数据所需的参数或者可增强经解码视频数据的可用性的补充数据。NAL单元606对应于条带且构成位流的VCL NAL单元。

[0111] 不同的NAL单元601-605对应于不同的参数集,这些NAL单元是非VCL NAL单元。解码器参数集(DPS)NAL单元301包含对于给定解码处理恒定的参数。视频参数集(VPS)NAL单元602包含针对整个视频且因此整个位流定义的参数。DPS NAL单元可以定义比VPS中的参数更静态的参数。换句话说,DPS的参数比VPS的参数更不频繁地改变。

[0112] 序列参数集(SPS)NAL单元603包含针对视频序列定义的参数。特别地,SPS NAL单元可定义视频序列的子图片布局及相关联参数。与各个子图片相关联的参数指定应用于子图片的编码约束。特别地,包括指示子图片之间的时间预测被限制为来自相同子图片的数据的标志。另一标志可以跨子图片边界启用或禁用环路滤波器。

[0113] 图片参数集(PPS)NAL单元604,PPS包含针对图片或图片组定义的参数。自适应参数集(APS)NAL单元605包含用于环路滤波器的参数,所述环路滤波器通常是自适应环路滤波器(ALF)或整形器模型(或具有色度缩放的亮度映射(LMCS)模型)或在条带级别使用的缩放矩阵。

[0114] 如在VVC的当前版本中提出的PPS的句法包括指定以亮度样本为单位的图片的大小以及各个图片以区块和条带的分区的句法元素。

[0115] PPS包含使得可以确定帧中的条带位置的句法元素。由于子图片在帧中形成矩形区域,因此可以根据参数集NAL单元确定属于子图片的条带集合、区块部分或区块。PPS与APS一样具有ID机制以限制相同PPS的发送的量。

[0116] PPS和图片头部之间的主要区别在于它的传输,与针对各个图像系统地发送的PH相比,PPS通常是针对图像组而发送的。因此,与PH相比,PPS包含对于若干图片可以是恒定的参数。

[0117] 位流还可以包含补充增强信息(SEI)NAL单元(图6中未表示)。这些参数集在位流中的出现周期是可变的。针对整个位流定义的VPS可以在位流中仅出现一次。相反,针对条带定义的APS可以针对各个图片中的各个条带出现一次。实际上,不同条带可依赖于同一APS,且因此通常存在比各个图片中的条带更少的APS。特别地,APS被定义在图片头部中。然而,可以在条带头部中细化ALF APS。

[0118] 访问单元定界符(Access Unit Delimiter(AUD))NAL单元607分离两个访问单元。访问单元是NAL单元的集合,其可以包括具有相同解码时间戳的一个或多于一个经编码图片。该任选NAL单元仅包含当前VVC规范中的一个句法元素:pic\_type,该句法元素指示slice\_type值用于AU中的经编码图片的所有条带。如果pic\_type被设置为等于0,则AU仅包含帧内(Intra)条带。如果等于1,则其包含P和I条带。如果等于2,则其包含B、P或帧内(Intra)条带。

[0119] 该NAL单元仅包含一个句法元素pic-type。

[0120] 表1句法AUD

[0121]	<b>access_unit_delimiter_rbsp() {</b>	<b>描述符</b>
	<b>pic_type</b>	u(3)
	<b>rbbsp_trailing_bits()</b>	
	<b>}</b>	

[0122] 在JVET-Q2001-vD中,pic-type定义如下:

[0123] “pic\_type指示包含AU定界符NAL单元的AU中的经编码图片的所有条带的slice\_type值是针对给定的pic\_type值在表2中列出的集合的成员。在符合该规范的该版本的位流中pic\_type的值应等于0、1或2。保留pic\_type的其他值以供ITUT|ISO/IEC将来使用。符合该规范的该版本的解码器将忽略pic\_type的保留值。”

[0124] **rbbsp\_trailing\_bits()**是添加位以与字节的结束对准的函数。因此,在该函数之后,所解析的位流的量是整数个字节。

[0125] 表2pic\_type的解释

[0126]	<b>pic_type</b>	<b>AU中可能存在的slice_type值</b>
	0	I
	1	P, I
	2	B, P, I

[0127] PH NAL单元608是图片头部NAL单元,其对于一个经编码图片的条带的集合的共同的参数进行分组。图片可以指一个或多个APSP以指示由图片的条带使用的AFL参数、整形器模型和缩放矩阵。

[0128] VCL NAL单元606各自包含条带。条带可以对应于整个图片或子图片、单个区块或多个区块或区块的片段。例如,图6的条带包含若干区块620。条带由条带头部610和原始字节序列有效载荷RBSP 611组成,RBSP 611包含编码为经编码块640的经编码像素数据。

[0129] 如在VVC的当前版本中提出的PPS的句法包括指定以亮度样本为单位的图片的大小以及以区块和条带为单位的各个图片的分区的句法元素。

[0130] PPS包含使得可以确定帧中的条带位置的句法元素。由于在帧中子图片形成矩形区域,因此可以从参数集NAL单元确定属于子图片的条带集合、区块部分或区块。

[0131] NAL单元条带

[0132] NAL单元条带层包含条带头部和条带数据,如表3所示。

[0133] 表3条带层句法

[0134]	<b>slice_layer_rbsp() {</b>	<b>描述符</b>
	<b>slice_header()</b>	
	<b>slice_data()</b>	
	<b>rbbsp_slice_trailing_bits()</b>	
	<b>}</b>	

[0135] APS

[0136] 自适应参数集(APS)NAL单元605在示出句法元素的表4中被定义。

[0137] 如表4中所描绘,存在由aps\_params\_type句法元素给出的3种可能类型的APS:

- [0138]     • ALF\_AP:用于ALF参数
- [0139]     • LMCS\_APS:用于LMCS参数
- [0140]     • SCALLING\_APS:用于缩放列表相关参数
- [0141]     表4自适应参数集句法

[0142]	adaptation_parameter_set_rbsp( ) {	描述符
	<b>adaptation_parameter_set_id</b>	u(5)
	<b>aps_params_type</b>	u(3)
	if( aps_params_type == ALF_APS )	
	alf_data( )	
	else if( aps_params_type == LMCS_APS )	
	lmcs_data( )	
	else if( aps_params_type == SCALING_APS )	
	scaling_list_data( )	
	<b>aps_extension_flag</b>	u(1)
	if( aps_extension_flag )	
	while( more_rbsp_data( ) )	
	<b>aps_extension_data_flag</b>	u(1)
	rbps_trailing_bits( )	
	}	

[0143]     下面依次讨论这三种类型的APS参数。

[0144]     ALF APS

[0145]     在自适应环路滤波器数据句法元素(表5)中描述ALF参数。首先,四个标志专用于指定是否针对亮度和/或针对色度发送ALF滤波器以及是否针对Cb分量和Cr分量启用CC-ALF(交叉分量自适应环路滤波)。如果启用亮度滤波器标志,则解码另一标志以知道是否用信号通知裁剪值(alf\_luma\_clip\_flag)。然后,使用alf\_luma\_num\_filters\_signalled\_minus1句法元素对用信号通知的滤波器的数量进行解码。如果需要,则针对各个经启用滤波器的各个系数的绝对值和符号。

[0146]     如果启用alf\_luma\_clip\_flag,则解码各个经启用滤波器的各个系数的裁剪索引。

[0147]     以相同方式,在需要时解码ALF色度系数。

[0148]     如果针对Cr或Cb启用CC-ALF,则对滤波器的数量进行解码(alf\_cc\_cb\_filters\_signalled\_minus1或alf\_cc\_cr\_filters\_signalled\_minus1)并且对相关系数进行解码(alf\_cc\_cb\_mapped\_coeff\_abs和alf\_cc\_cb\_coeff\_sign或者相应地alf\_cc\_cr\_mapped\_coeff\_abs和alf\_cc\_cr\_coeff\_sign)。



[0149] 表5自适应环路滤波器数据句法

[0150]	alf_data( ) {	描述符
	<b>alf_luma_filter_signal_flag</b>	u(1)
	<b>alf_chroma_filter_signal_flag</b>	u(1)
	<b>alf_cc_cb_filter_signal_flag</b>	u(1)
	<b>alf_cc_cr_filter_signal_flag</b>	u(1)
	if( alf_luma_filter_signal_flag ) {	
	<b>alf_luma_clip_flag</b>	u(1)
	<b>alf_luma_num_filters_signalled_minus1</b>	ue(v)
	if( alf_luma_num_filters_signalled_minus1 > 0 )	
	for( filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++ )	

[0151]

<b>alf_luma_coeff_delta_idx</b> [ filtIdx ]	u(v)
for( sflIdx = 0; sflIdx <= alf_luma_num_filters_signalled_minus1; sflIdx++ )	
for( j = 0; j < 12; j++ ) {	
<b>alf_luma_coeff_abs</b> [ sflIdx ][ j ]	ue(v)
if( alf_luma_coeff_abs[ sflIdx ][ j ] )	
<b>alf_luma_coeff_sign</b> [ sflIdx ][ j ]	u(1)
}	
if( alf_luma_clip_flag )	
for( sflIdx = 0; sflIdx <= alf_luma_num_filters_signalled_minus1; sflIdx++ )	
for( j = 0; j < 12; j++ )	
<b>alf_luma_clip_idx</b> [ sflIdx ][ j ]	u(2)
}	
if( alf_chroma_filter_signal_flag ) {	
<b>alf_chroma_clip_flag</b>	u(1)
<b>alf_chroma_num_alt_filters_minus1</b>	ue(v)
for( altIdx = 0; altIdx <= alf_chroma_num_alt_filters_minus1; altIdx++ ) {	
for( j = 0; j < 6; j++ ) {	
<b>alf_chroma_coeff_abs</b> [ altIdx ][ j ]	ue(v)
if( alf_chroma_coeff_abs[ altIdx ][ j ] > 0 )	
<b>alf_chroma_coeff_sign</b> [ altIdx ][ j ]	u(1)
}	
if( alf_chroma_clip_flag )	
for( j = 0; j < 6; j++ )	
<b>alf_chroma_clip_idx</b> [ altIdx ][ j ]	u(2)
}	
}	
if( alf_cc_cb_filter_signal_flag ) {	

[0152]

<b>alf_cc_cb_filters_signalled_minus1</b>	ue(v)
for( k = 0; k < alf_cc_cb_filters_signalled_minus1 + 1; k++ ) {	
for( j = 0; j < 7; j++ ) {	
<b>alf_cc_cb_mapped_coeff_abs[ k ][ j ]</b>	u(3)
if( alf_cc_cb_mapped_coeff_abs[ k ][ j ] )	
<b>alf_cc_cb_coeff_sign[ k ][ j ]</b>	u(1)
}	
}	
}	
if( alf_cc_cr_filter_signal_flag ) {	
<b>alf_cc_cr_filters_signalled_minus1</b>	ue(v)
for( k = 0; k < alf_cc_cr_filters_signalled_minus1 + 1; k++ ) {	
for( j = 0; j < 7; j++ ) {	
<b>alf_cc_cr_mapped_coeff_abs[ k ][ j ]</b>	u(3)
if( alf_cc_cr_mapped_coeff_abs[ k ][ j ] )	
<b>alf_cc_cr_coeff_sign[ k ][ j ]</b>	u(1)
}	
}	
}	
}	

[0153] 用于亮度映射和色度缩放这两者的LMCS句法元素

[0154] 下面的表6给出了当aps\_params\_type参数被设置为1时在自适应参数集 (APS) 句法结构中编码的所有LMCS句法元素 (LMCS\_APS)。在经编码视频序列中可以使用多达四个LMCS\_APS,然而,对于给定的图片,仅可以使用单个LMCS\_APS。

[0155] 这些参数用于构建用于亮度的前向和逆向映射函数以及用于色度的缩放函数。

[0156] 表6具有色度缩放数据句法的亮度映射

[0157]	lmcs_data () {	描述符
--------	----------------	-----

[0158]	<b>lmcs_min_bin_idx</b>	ue(v)
	<b>lmcs_delta_max_bin_idx</b>	ue(v)
	<b>lmcs_delta_cw_prec_minus1</b>	ue(v)
	for( i = lmcs_min_bin_idx; i <= LmcsMaxBinIdx; i++ ) {	
	<b>lmcs_delta_abs_cw[ i ]</b>	u(v)
	if( lmcs_delta_abs_cw[ i ] > 0 )	
	<b>lmcs_delta_sign_cw_flag[ i ]</b>	u(1)
	}	
	<b>lmcs_delta_abs_crs</b>	u(3)
	if( lmcs_delta_abs_crs > 0 )	
	<b>lmcs_delta_sign_crs_flag</b>	u(1)
	}	

[0159] 缩放列表APS

[0160] 缩放列表提供了更新用于量化的量化矩阵的可能性。在VVC中,该缩放矩阵在如缩放列表数据句法元素(表7缩放列表数据句法)中所描述的APS中用信号通知。第一句法元素基于标志scaling\_matrix\_for\_lfnst\_disabled\_flag指定缩放矩阵是否用于LFNST(低频不可分离变换)工具。如果缩放列表用于色度分量(scaling\_list\_chroma\_present\_flag),则指定第二个。然后,解码构建缩放矩阵所需的句法元素(scaling\_list\_copy\_mode\_flag、scaling\_list\_pred\_mode\_flag、scaling\_list\_pred\_id\_delta、scaling\_list\_dc\_coef、scaling\_list\_delta\_coef)。

[0161] 表7缩放列表数据句法

[0162]	scaling_list_data() {	<b>描述符</b>
	<b>scaling_matrix_for_lfnst_disabled_flag</b>	u(1)
	<b>scaling_list_chroma_present_flag</b>	u(1)
	for( id = 0; id < 28; id ++ )	
	matrixSize = (id < 2) ? 2 : ((id < 8) ? 4 : 8)	
	if( scaling_list_chroma_present_flag    (id % 3 == 2)    (id == 27) ) {	
	<b>scaling_list_copy_mode_flag[ id ]</b>	u(1)
	}	

	if( !scaling_list_copy_mode_flag[ id ] )	
	<b>scaling_list_pred_mode_flag[ id ]</b>	u(1)
	if( ( scaling_list_copy_mode_flag[ id ]    scaling_list_pred_mode_flag[ id ] ) && id != 0 && id != 2 && id != 8 )	
	<b>scaling_list_pred_id_delta[ id ]</b>	ue(v)
	if( !scaling_list_copy_mode_flag[ id ] ) {	
	nextCoef = 0	
	if( id > 13 ) {	
	<b>scaling_list_dc_coef[ id - 14 ]</b>	se(v)
	nextCoef += scaling_list_dc_coef[ id - 14 ]	
	}	
[0163]	for( i = 0; i < matrixSize * matrixSize; i++ ) {	
	x = DiagScanOrder[ 3 ][ 3 ][ i ][ 0 ]	
	y = DiagScanOrder[ 3 ][ 3 ][ i ][ 1 ]	
	if( !( id > 25 && x >= 4 && y >= 4 ) ) {	
	<b>scaling_list_delta_coef[ id ][ i ]</b>	se(v)
	nextCoef += scaling_list_delta_coef[ id ][ i ]	
	}	
	ScalingList[ id ][ i ] = nextCoef	
	}	
	}	
	}	
	}	
	}	

[0164] 图片头部

[0165] 在其他条带数据之前在各个图片的开始处发送图片头部。与标准的先前草案中的先前头部相比,这是非常大的。所有这些参数的完整描述可以在JVET\_Q2001-vD中找到。表9示出了当前图片头部解码句法中的这些参数。

[0166] 可以解码的相关句法元素涉及:

[0167] • 是否使用该图片、参考帧

[0168] • 图片的类型

[0169] • 输出帧

[0170] • 图片的数量

- [0171]     • 使用子图片(如果需要)
- [0172]     • 参考图片列表(如果需要)
- [0173]     • 颜色平面(如果需要)
- [0174]     • 分区更新(如果启用了覆写标志)
- [0175]     • 增量QP参数(如果需要)
- [0176]     • 运动信息参数(如果需要)
- [0177]     • ALF参数(如果需要)
- [0178]     • SAO参数(如果需要)
- [0179]     • 量化参数(如果需要)
- [0180]     • LMCS参数(如果需要)
- [0181]     • 缩放列表参数(如果需要)
- [0182]     • 图片头部扩展(如果需要)
- [0183]     • 等等
- [0184]     图片“类型”
- [0185]     第一标志是`grd_or_irap_pic_flag`,其指示当前图片是否是再同步图片(IRAP或GDR)。如果该标志为真,则解码`gdr_pic_flag`以知道当前图片是IRAP图片还是GDR图片。
- [0186]     然后对`ph_inter_slice_allowed_flag`进行解码以识别允许帧间条带。
- [0187]     当它们被允许时,对标志`ph_infra_slice_allowed_flag`进行解码以知道针对当前图片是否允许帧内条带。
- [0188]     然后对`non_reference_picture_flag`、指示PPS ID的`ph_pic_parameter_set_id`和图片顺序计数`ph_pic_order_cnt_lsb`进行解码。图片顺序计数给出了当前图片的编号。
- [0189]     如果图片是GDR或IRAP图片,则对标志`no_output_of_prior_pics_flag`进行解码。
- [0190]     并且如果图片是GDR,则对`recovery_poc_cnt`进行解码。然后,如果需要,则对`ph_poc_msb_present_flag`和`poc_msb_val`进行解码。
- [0191]     ALF
- [0192]     在描述关于当前图片的重要信息的这些参数之后,如果在SPS级别启用ALF并且如果在图片头部级别启用ALF,则解码ALF APS ID句法元素的集合。由于`sps_alf_enabled_flag`标志,在SPS级别启用ALF。并且由于`alf_info_in_ph_flag`等于1,在图片头部级别启用用信号通知ALF,否则(`alf_info_in_ph_flag`等于0),在条带级别用信号通知ALF。
- [0193]     `alf_info_in_ph_flag`定义如下:
- [0194]     “`alf_info_in_ph_flag`等于1指定ALF信息存在于PH句法结构中且不存在于参考不包含PH句法结构的PPS的条带头部中。`alf_info_in_ph_flag`等于0指定ALF信息不存在于PH句法结构中且可以存在于参考不包含PH句法结构的PPS的条带头部中。”
- [0195]     首先,对`ph_alf_enabled_present_flag`进行解码以确定是否应该解码`ph_alf_enabled_flag`。如果启用`ph_alf_enabled_present_flag`,则针对当前图片的所有条带启用ALF。
- [0196]     如果启用ALF,则使用`pic_num_alf_aps_ids_luma`句法元素来解码针对亮度的ALF APS ID的量。对于各个APS ID,解码针对亮度的APS ID值“`ph_alf_aps_id_luma`”。
- [0197]     对于色度,对句法元素`ph_alf_chroma_idc`进行解码以确定是否针对色度、仅针对

Cr或仅针对Cb启用ALF。如果启用,则使用ph\_alf\_aps\_id\_chroma句法元素来解码针对色度的APS ID的值。

[0198] 以这种方式,如果Cb和/或Cr分量需要,则解码针对CC-ALF方法的APS ID。

[0199] LMCS

[0200] 如果在SPS级别启用LMCS,则解码LMCS APS ID句法元素的集合。首先,对ph\_lmcs\_enabled\_flag进行解码以确定是否针对当前图片启用LMCS。如果启用LMCS,则解码ID值ph\_lmcs\_aps\_id。对于色度,仅对ph\_chroma\_residual\_scale\_flag进行解码以启用或禁用针对色度的方法。

[0201] 缩放列表

[0202] 如果在SPS级别启用缩放列表,则对缩放列表APS ID的集合进行解码。对ph\_scaling\_list\_present\_flag进行解码,以确定是否针对当前图片启用缩放矩阵。并且然后解码APS ID的值(ph\_scaling\_list\_aps\_id)。

[0203] 子图片

[0204] 当在SPS处启用子图片参数并且如果用信号通知子图片ID被禁用时,则启用子图片参数。还包含关于虚拟边界的一些信息。对于子图片参数,定义八个句法元素:

[0205] • ph\_virtual\_boundaries\_present\_flag

[0206] • ph\_num\_ver\_virtual\_boundaries

[0207] • ph\_virtual\_boundaries\_pos\_x[i]

[0208] • ph\_num\_hor\_virtual\_boundaries

[0209] • ph\_virtual\_boundaries\_pos\_y[i]

[0210] 输出标志

[0211] 这些子图片参数之后是pic\_output\_flag(如果存在)。

[0212] 参考图片列表

[0213] 如果在图片头部中用信号通知参考图片列表(由于rpl\_info\_in\_ph\_flag等于1),则解码参考图片列表的参数ref\_pic\_lists(),其包含以下句法元素:

[0214] • rpl\_sps\_flag[]

[0215] • rpl\_idx[]

[0216] • poc\_lsb\_lt[][]

[0217] • delta\_poc\_msb\_present\_flag[][]

[0218] • delta\_poc\_msb\_cycle\_lt[][]

[0219] 分区

[0220] 如果需要,则对分区参数集合进行解码,并且该分区参数集合包含以下句法元素:

[0221] • partition\_constraints\_override\_flag

[0222] • ph\_log2\_diff\_min\_qt\_min\_cb\_intra\_slice\_luma

[0223] • ph\_max\_mtt\_hierarchy\_depth\_intra\_slice\_luma

[0224] • ph\_log2\_diff\_max\_bt\_min\_qt\_intra\_slice\_luma

[0225] • ph\_log2\_diff\_max\_tt\_min\_qt\_intra\_slice\_luma

[0226] • ph\_log2\_diff\_min\_qt\_min\_cb\_intra\_slice\_chroma

[0227] • ph\_max\_mtt\_hierarchy\_depth\_intra\_slice\_chroma

- [0228]     • ph\_log2\_diff\_max\_bt\_min\_qt\_intra\_slice\_chroma
- [0229]     • ph\_log2\_diff\_max\_tt\_min\_qt\_intra\_slice\_chroma
- [0230]     • ph\_log2\_diff\_min\_qt\_min\_cb\_inter\_slice
- [0231]     • ph\_max\_mtt\_hierarchy\_depth\_inter\_slice
- [0232]     • ph\_log2\_diff\_max\_bt\_min\_qt\_inter\_slice
- [0233]     • ph\_log2\_diff\_max\_tt\_min\_qt\_inter\_slice
- [0234]     加权预测
- [0235]     如果在PPS级别启用加权预测方法并且如果在图片头部中用信号通知加权预测参数(wp\_info\_in\_ph\_flag等于1),则解码加权预测参数pred\_weight\_table()。
- [0236]     当启用双向预测加权预测时,pred\_weight\_table() 包含列表L0和列表L1的加权预测参数。如pred\_weight\_table() 句法表(表8)中所描绘的,当在图片头部中发送加权预测参数时,各个列表的权重的数量被显式地发送。
- [0237]     表8加权预测参数句法

[0238]	pred_weight_table() {	描述符
--------	-----------------------	-----



[0239]

<b>luma_log2_weight_denom</b>	ue(v)
if( ChromaArrayType != 0 )	
<b>delta_chroma_log2_weight_denom</b>	se(v)
if( wp_info_in_ph_flag )	
<b>num_l0_weights</b>	ue(v)
for( i = 0; i < NumWeightsL0; i++ )	
<b>luma_weight_l0_flag[ i ]</b>	u(1)
if( ChromaArrayType != 0 )	
for( i = 0; i < NumWeightsL0; i++ )	
<b>chroma_weight_l0_flag[ i ]</b>	u(1)
for( i = 0; i < NumWeightsL0; i++ ) {	
if( luma_weight_l0_flag[ i ] ) {	
<b>delta_luma_weight_l0[ i ]</b>	se(v)
<b>luma_offset_l0[ i ]</b>	se(v)
}	
if( chroma_weight_l0_flag[ i ] )	
for( j = 0; j < 2; j++ ) {	
<b>delta_chroma_weight_l0[ i ][ j ]</b>	se(v)
<b>delta_chroma_offset_l0[ i ][ j ]</b>	se(v)
}	
}	
if( pps_weighted_bipred_flag && wp_info_in_ph_flag )	
<b>num_l1_weights</b>	ue(v)
for( i = 0; i < NumWeightsL1; i++ )	
<b>luma_weight_l1_flag[ i ]</b>	u(1)
if( ChromaArrayType != 0 )	
for( i = 0; i < NumWeightsL1; i++ )	
<b>chroma_weight_l1_flag[ i ]</b>	u(1)
for( i = 0; i < NumWeightsL1; i++ ) {	
if( luma_weight_l1_flag[ i ] ) {	

[0240]

<b>delta_luma_weight_11[ i ]</b>	se(v)
<b>luma_offset_11[ i ]</b>	se(v)
}	
if( chroma_weight_11_flag[ i ] )	
for( j = 0; j < 2; j++ ) {	
<b>delta_chroma_weight_11[ i ][ j ]</b>	se(v)
<b>delta_chroma_offset_11[ i ][ j ]</b>	se(v)
}	
}	
}	

[0241] 增量QP

[0242] 当图片是帧内时,如果需要,则对ph\_cu\_qp\_delta\_subdiv\_intra\_slice和ph\_cu\_chroma\_qp\_offset\_subdiv\_intra\_slice进行解码。并且如果允许帧间条带,则在需要时对ph\_cu\_qp\_delta\_subdiv\_inter\_slice和ph\_cu\_chroma\_qp\_offset\_subdiv\_inter\_slice进行解码。最后,如果需要,则对图片头部扩展句法元素进行解码。

[0243] 在PPS中用信号通知所有参数alf\_info\_in\_ph\_flag、rpl\_info\_in\_ph\_flag、qp\_delta\_info\_in\_ph\_flag、sao\_info\_in\_ph\_flag、dbf\_info\_in\_ph\_flag、wp\_info\_in\_ph\_flag。

[0244] 表9图片头部结构

[0245]

picture_header_structure( ) {	<b>描述符</b>
<b>gdr_or_irap_pic_flag</b>	u(1)
if( gdr_or_irap_pic_flag )	
<b>gdr_pic_flag</b>	u(1)
<b>ph_inter_slice_allowed_flag</b>	u(1)
if( ph_inter_slice_allowed_flag )	
<b>ph_intra_slice_allowed_flag</b>	u(1)
<b>non_reference_picture_flag</b>	u(1)

[0246]

<b>ph_pic_parameter_set_id</b>	ue(v)
<b>ph_pic_order_cnt_lsb</b>	u(v)
if( gdr_or_irap_pic_flag )	
<b>no_output_of_prior_pics_flag</b>	u(1)
if( gdr_pic_flag )	
<b>recovery_poc_cnt</b>	ue(v)
for( i = 0; i < NumExtraPhBits; i++ )	
<b>ph_extra_bit[ i ]</b>	u(1)
if( sps_poc_msb_flag ) {	
<b>ph_poc_msb_present_flag</b>	u(1)
if( ph_poc_msb_present_flag )	
<b>poc_msb_val</b>	u(v)
}	
if( sps_alf_enabled_flag && alf_info_in_ph_flag ) {	
<b>ph_alf_enabled_flag</b>	u(1)
if( ph_alf_enabled_flag ) {	
<b>ph_num_alf_aps_ids_luma</b>	u(3)
for( i = 0; i < ph_num_alf_aps_ids_luma; i++ )	
<b>ph_alf_aps_id_luma[ i ]</b>	u(3)
if( ChromaArrayType != 0 )	
<b>ph_alf_chroma_idc</b>	u(2)
if( ph_alf_chroma_idc > 0 )	
<b>ph_alf_aps_id_chroma</b>	u(3)
if( sps_ccalf_enabled_flag ) {	
<b>ph_cc_alf_cb_enabled_flag</b>	u(1)
if( ph_cc_alf_cb_enabled_flag )	
<b>ph_cc_alf_cb_aps_id</b>	u(3)
<b>ph_cc_alf_cr_enabled_flag</b>	u(1)
if( ph_cc_alf_cr_enabled_flag )	
<b>ph_cc_alf_cr_aps_id</b>	u(3)

[0247]

}	
}	
}	
if( sps_lmcs_enabled_flag ) {	
<b>ph_lmcs_enabled_flag</b>	u(1)
if( ph_lmcs_enabled_flag ) {	
<b>ph_lmcs_aps_id</b>	u(2)
if( ChromaArrayType != 0 )	
<b>ph_chroma_residual_scale_flag</b>	u(1)
}	
}	
if( sps_scaling_list_enabled_flag ) {	
<b>ph_scaling_list_present_flag</b>	u(1)
if( ph_scaling_list_present_flag )	
<b>ph_scaling_list_aps_id</b>	u(3)
}	
if( sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag ) {	
<b>ph_virtual_boundaries_present_flag</b>	u(1)
if( ph_virtual_boundaries_present_flag ) {	
<b>ph_num_ver_virtual_boundaries</b>	u(2)
for( i = 0; i < ph_num_ver_virtual_boundaries; i++ )	
<b>ph_virtual_boundaries_pos_x[ i ]</b>	u(13)
<b>ph_num_hor_virtual_boundaries</b>	u(2)
for( i = 0; i < ph_num_hor_virtual_boundaries; i++ )	
<b>ph_virtual_boundaries_pos_y[ i ]</b>	u(13)
}	
}	
if( output_flag_present_flag )	
<b>pic_output_flag</b>	u(1)

[0248]

if( rpl_info_in_ph_flag )	
ref_pic_lists( )	
if( partition_constraints_override_enabled_flag )	
<b>partition_constraints_override_flag</b>	u(1)
if( ph_intra_slice_allowed_flag ) {	
if( partition_constraints_override_flag ) {	
<b>ph_log2_diff_min_qt_min_cb_intra_slice_luma</b>	ue(v)
<b>ph_max_mtt_hierarchy_depth_intra_slice_luma</b>	ue(v)
if( ph_max_mtt_hierarchy_depth_intra_slice_luma != 0 ) {	
<b>ph_log2_diff_max_bt_min_qt_intra_slice_luma</b>	ue(v)
<b>ph_log2_diff_max_tt_min_qt_intra_slice_luma</b>	ue(v)
}	
if( qtbtt_dual_tree_intra_flag ) {	
<b>ph_log2_diff_min_qt_min_cb_intra_slice_chroma</b>	ue(v)
<b>ph_max_mtt_hierarchy_depth_intra_slice_chroma</b>	ue(v)
if( ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0 ) {	
<b>ph_log2_diff_max_bt_min_qt_intra_slice_chroma</b>	ue(v)
<b>ph_log2_diff_max_tt_min_qt_intra_slice_chroma</b>	ue(v)
}	
}	
}	
if( cu_qp_delta_enabled_flag )	
<b>ph_cu_qp_delta_subdiv_intra_slice</b>	ue(v)
if( pps_cu_chroma_qp_offset_list_enabled_flag )	
<b>ph_cu_chroma_qp_offset_subdiv_intra_slice</b>	ue(v)
}	
if( ph_inter_slice_allowed_flag ) {	
if( partition_constraints_override_flag ) {	
<b>ph_log2_diff_min_qt_min_cb_inter_slice</b>	ue(v)
<b>ph_max_mtt_hierarchy_depth_inter_slice</b>	ue(v)

[0249]

if( ph_max_mtt_hierarchy_depth_inter_slice != 0 ) {	
<b>ph_log2_diff_max_bt_min_qt_inter_slice</b>	ue(v)
<b>ph_log2_diff_max_tt_min_qt_inter_slice</b>	ue(v)
}	
}	
if( cu_qp_delta_enabled_flag )	
<b>ph_cu_qp_delta_subdiv_inter_slice</b>	ue(v)
if( pps_cu_chroma_qp_offset_list_enabled_flag )	
<b>ph_cu_chroma_qp_offset_subdiv_inter_slice</b>	ue(v)
if( sps_temporal_mvp_enabled_flag ) {	
<b>ph_temporal_mvp_enabled_flag</b>	u(1)
if( ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag ) {	
<b>ph_collocated_from_l0_flag</b>	u(1)
if( ( ph_collocated_from_l0_flag && num_ref_entries[ 0 ][ RplsIdx[ 0 ] ] > 1 )    ( !ph_collocated_from_l0_flag && num_ref_entries[ 1 ][ RplsIdx[ 1 ] ] > 1 ) )	
<b>ph_collocated_ref_idx</b>	ue(v)
}	
}	
<b>mvd_l1_zero_flag</b>	u(1)
if( sps_fpel_mmvd_enabled_flag )	
<b>ph_fpel_mmvd_enabled_flag</b>	u(1)
if( sps_bdof_pic_present_flag )	
<b>ph_disable_bdof_flag</b>	u(1)
if( sps_dmvr_pic_present_flag )	
<b>ph_disable_dmvr_flag</b>	u(1)
if( sps_prof_pic_present_flag )	
<b>ph_disable_prof_flag</b>	u(1)
if( ( pps_weighted_pred_flag    pps_weighted_bipred_flag ) &&	

[0250]

wp_info_in_ph_flag )	
pred_weight_table( )	
}	
if( qp_delta_info_in_ph_flag )	
<b>ph_qp_delta</b>	se(v)
if( sps_joint_cbr_enabled_flag )	
<b>ph_joint_cbr_sign_flag</b>	u(1)
if( sps_sao_enabled_flag && sao_info_in_ph_flag ) {	
<b>ph_sao_luma_enabled_flag</b>	u(1)
if( ChromaArrayType != 0 )	
<b>ph_sao_chroma_enabled_flag</b>	u(1)
}	
if( sps_dep_quant_enabled_flag )	
<b>ph_dep_quant_enabled_flag</b>	u(1)
if( sps_sign_data_hiding_enabled_flag && !ph_dep_quant_enabled_flag )	
<b>pic_sign_data_hiding_enabled_flag</b>	u(1)
if( deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag ) {	
<b>ph_deblocking_filter_override_flag</b>	u(1)
if( ph_deblocking_filter_override_flag ) {	
<b>ph_deblocking_filter_disabled_flag</b>	u(1)
if( !ph_deblocking_filter_disabled_flag ) {	
<b>ph_beta_offset_div2</b>	se(v)
<b>ph_tc_offset_div2</b>	se(v)
<b>ph_cb_beta_offset_div2</b>	se(v)
<b>ph_cb_tc_offset_div2</b>	se(v)
<b>ph_cr_beta_offset_div2</b>	se(v)
<b>ph_cr_tc_offset_div2</b>	se(v)
}	
}	

[0251]	}	
	if( picture_header_extension_present_flag ) {	
	<b>ph_extension_length</b>	ue(v)
	for( i = 0; i < ph_extension_length; i++)	
	<b>ph_extension_data_byte[ i ]</b>	u(8)
	}	
	}	

[0252] 条带头部

[0253] 在各个条带的开始处发送条带头部。条带头部包含约65个句法元素。与早期视频编码标准中的先前条带头部相比,这是非常大的。可以在JVET-Q2001-vD中找到所有条带头部参数的完整描述。表10示出当前条带头部解码句法中的这些参数。

[0254] 表10部分条带头部

[0255]	slice_header() {	<b>描述符</b>
	<b>picture_header_in_slice_header_flag</b>	u(1)
	if( picture_header_in_slice_header_flag )	
	picture_header_structure()	
	if( subpic_info_present_flag )	
	<b>slice_subpic_id</b>	u(v)
	if( ( rect_slice_flag && NumSlicesInSubpic[ CurrSubpicIdx ] > 1 )    ( !rect_slice_flag && NumTilesInPic > 1 ) )	
	<b>slice_address</b>	u(v)
	for( i = 0; i < NumExtraShBits; i++ )	
	<b>sh_extra_bit[ i ]</b>	u(1)
	if( !rect_slice_flag && NumTilesInPic > 1 )	
	<b>num_tiles_in_slice_minus1</b>	ue(v)
	if( ph_inter_slice_allowed_flag )	
	<b>slice_type</b>	ue(v)
	if( sps_alf_enabled_flag && !alf_info_in_ph_flag ) {	



[0256]

<b>slice_alf_enabled_flag</b>	u(1)
if( slice_alf_enabled_flag ) {	
<b>slice_num_alf_aps_ids_luma</b>	u(3)
for( i = 0; i < slice_num_alf_aps_ids_luma; i++ )	
<b>slice_alf_aps_id_luma[ i ]</b>	u(3)
if( ChromaArrayType != 0 )	
<b>slice_alf_chroma_idc</b>	u(2)
if( slice_alf_chroma_idc )	
<b>slice_alf_aps_id_chroma</b>	u(3)
if( sps_ccalf_enabled_flag ) {	
<b>slice_cc_alf_cb_enabled_flag</b>	u(1)
if( slice_cc_alf_cb_enabled_flag )	
<b>slice_cc_alf_cb_aps_id</b>	u(3)
<b>slice_cc_alf_cr_enabled_flag</b>	u(1)
if( slice_cc_alf_cr_enabled_flag )	
<b>slice_cc_alf_cr_aps_id</b>	u(3)
}	
}	
}	
if( separate_colour_plane_flag == 1 )	
<b>colour_plane_id</b>	u(2)
if( !rpl_info_in_ph_flag && ( ( nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP )    sps_idr_rpl_present_flag ) )	
ref_pic_lists()	

[0257]

<pre> if( ( rpl_info_in_ph_flag    ( ( nal_unit_type != IDR_W_RADL &amp;&amp; nal_unit_type != IDR_N_LP )    sps_idr_rpl_present_flag ) ) &amp;&amp; ( ( slice_type != I &amp;&amp; num_ref_entries[ 0 ][ RplIdx[ 0 ] ] &gt; 1 )    ( slice_type == B &amp;&amp; num_ref_entries[ 1 ][ RplIdx[ 1 ] ] &gt; 1 ) ) { </pre>	
<b>num_ref_idx_active_override_flag</b>	u(1)
if( num_ref_idx_active_override_flag )	
for( i = 0; i < ( slice_type == B ? 2 : 1 ); i++ )	
if( num_ref_entries[ i ][ RplIdx[ i ] ] > 1 )	
<b>num_ref_idx_active_minus1[ i ]</b>	ue(v)
}	
if( slice_type != I ) {	
if( cabac_init_present_flag )	
<b>cabac_init_flag</b>	u(1)
if( ph_temporal_mvp_enabled_flag && !rpl_info_in_ph_flag ) {	
if( slice_type == B )	
<b>slice_collocated_from_l0_flag</b>	u(1)
if( ( slice_collocated_from_l0_flag && NumRefIdxActive[ 0 ] > 1 )    ( ! slice_collocated_from_l0_flag && NumRefIdxActive[ 1 ] > 1 ) )	
<b>slice_collocated_ref_idx</b>	ue(v)
}	
if( !wp_info_in_ph_flag && ( ( pps_weighted_pred_flag && slice_type == P )    ( pps_weighted_bipred_flag && slice_type == B ) ) )	
pred_weight_table( )	
}	
if( !qp_delta_info_in_ph_flag )	
<b>slice_qp_delta</b>	se(v)

[0258]

if( pps_slice_chroma_qp_offsets_present_flag ) {	
<b>slice_cb_qp_offset</b>	se(v)
<b>slice_cr_qp_offset</b>	se(v)
if( sps_joint_cbr_enabled_flag )	
<b>slice_joint_cbr_qp_offset</b>	se(v)
}	
if( pps_cu_chroma_qp_offset_list_enabled_flag )	
<b>cu_chroma_qp_offset_enabled_flag</b>	u(1)
if( sps_sao_enabled_flag && !sao_info_in_ph_flag ) {	
<b>slice_sao_luma_flag</b>	u(1)
if( ChromaArrayType != 0 )	
<b>slice_sao_chroma_flag</b>	u(1)
}	
if( deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag )	
<b>slice_deblocking_filter_override_flag</b>	u(1)
if( slice_deblocking_filter_override_flag ) {	
<b>slice_deblocking_filter_disabled_flag</b>	u(1)
if( !slice_deblocking_filter_disabled_flag ) {	
<b>slice_beta_offset_div2</b>	se(v)
<b>slice_tc_offset_div2</b>	se(v)
<b>slice_cb_beta_offset_div2</b>	se(v)
<b>slice_cb_tc_offset_div2</b>	se(v)
<b>slice_cr_beta_offset_div2</b>	se(v)
<b>slice_cr_tc_offset_div2</b>	se(v)
}	
}	
<b>slice_ts_residual_coding_disabled_flag</b>	u(1)
if( ph_lmcs_enabled_flag )	
<b>slice_lmcs_enabled_flag</b>	u(1)
if( ph_scaling_list_present_flag )	

[0259]	<b>slice_scaling_list_present_flag</b>	u(1)
	if( NumEntryPoints > 0 ) {	
	<b>offset_len_minus1</b>	ue(v)
	for( i = 0; i < NumEntryPoints; i++ )	
	<b>entry_point_offset_minus1[ i ]</b>	u(v)
	}	
	if( slice_header_extension_present_flag ) {	
	<b>slice_header_extension_length</b>	ue(v)
	for( i = 0; i < slice_header_extension_length; i++ )	
	<b>slice_header_extension_data_byte[ i ]</b>	u(8)
	}	
	<b>byte_alignment( )</b>	
	}	

[0260] 首先,对picture\_header\_in\_slice\_header\_flag进行解码,以知道在条带头部中是否存在picture\_header\_structure()。

[0261] 然后,如果需要,对slice\_subpic\_id进行解码以确定当前条带的子图片ID。然后对slice\_address进行解码以确定当前条带的地址。如果当前条带模式为矩形条带模式(rest\_slice\_flag等于1)并且如果当前子图片中的条带数量高于1,则对条带地址进行解码。如果当前条带模式为光栅扫描模式(rest\_slice\_flag等于0)且如果当前图片中的区块的数量高于基于PPS中定义的变量所计算出的1,则还可以对条带地址进行解码。

[0262] 如果当前图片中的区块的数量大于1并且如果当前条带模式不是矩形条带模式,则解码num\_tiles\_in\_slice\_minus1。在当前VVC草案规范中,num\_tiles\_in\_slice\_minus1定义如下:

[0263] “num\_tiles\_in\_slice\_minus1加1(当存在时)指定条带中的区块的数量。num\_tiles\_in\_slice\_minus1的值应在0到NumTilesInPic-1的范围内(包含端值)。”

[0264] 然后对slice\_type进行解码。

[0265] 如果在SPS级别启用ALF(sps\_alf\_enabled\_flag)并且如果在条带头部中用信号通知ALF(alf\_info\_in\_ph\_flag等于0),则解码ALF信息。这包括指示针对当前条带启用ALF的标志(slice\_alf\_enabled\_flag)。如果被启用,则解码针对亮度的APS ALF ID的数量(slice\_num\_alf\_aps\_ids\_luma),然后解码APS ID(slice\_alf\_aps\_id\_luma[i])。然后,解码slice\_alf\_chroma\_idc以知道是否针对色度分量启用ALF以及启用哪个色度分量。然后,如果需要,则解码针对色度的APS ID(slice\_alf\_aps\_id\_chroma)。以相同方式,如果需要,则解码slice\_cc\_alf\_cb\_enabled\_flag以知道是否启用CC ALF方法。如果启用CC ALF,如果针对Cr和/或Cb启用CC ALF,则解码针对Cr和/或Cb的相关APS ID。

[0266] 如果独立地发送颜色平面(separate\_colour\_plane\_flag等于1),则对colour\_plane\_id进行解码。

[0267] 当不在图片头部中发送参考图片列表(`rpl_info_in_ph_flag`等于0)时并且当NAL单元不是IDR时或者如果针对IDR图片发送参考图片列表(`sps_idr_rpl_present_flag`等于1),则对参考图片列表参数进行解码;这些类似于图片头部中的那些。

[0268] 如果在图片头部中发送参考图片列表(`rpl_info_in_ph_flag`等于1)或者NAL单元不是IDR、或者如果针对IDR图片发送参考图片列表(`sps_idr_rpl_present_flag`等于1),并且如果至少一个列表的参考数量高于1,则对覆写标志`num_ref_idx_active_override_flag`进行解码。

[0269] 如果启用该标志,则解码各个列表的参考索引。

[0270] 如果启用`num_ref_idx_active_override_flag`,则在需要时解码各个列表“i”的参考索引的数量`num_ref_idx_active_minus1[i]`。针对当前列表的参考索引覆写的数量应低于或等于在`ref_pic_lists()`中用信号通知的参考帧索引的数量。因此,覆写减小或不减小各个列表的参考帧的最大数量。

[0271] 当条带类型不是帧内时,并且如果需要,则解码`cabac_init_flag`。如果在条带头部中发送参考图片列表并且出现其他条件,则解码`slice_collocated_from_l0_flag`和`slice_collocated_ref_idx`。这些数据与CABAC编码和并置运动矢量有关。

[0272] 以相同的方式,当条带类型不是帧内时,对加权预测的参数`pred_weight_table()`进行解码。

[0273] 如果在条带头部中发送增量QP信息(`qp_delta_info_in_ph_flag`等于0),则对`slice_qp_delta`进行解码。如果需要,则对句法元素`slice_cb_qp_offset`、`slice_cr_qp_offset`、`slice_joint_cbr_qp_offset`、`cu_chroma_qp_offset_enabled_flag`进行解码。

[0274] 如果SAO信息在条带头部中发送(`sao_info_in_ph_flag`等于0)并且如果其在SPS级别启用(`sps_sao_enabled_flag`),则针对亮度和色度这两者来解码SAO的启用标志:`slice_sao_luma_flag`、`slice_sao_chroma_flag`。

[0275] 然后,如果在条带头部中用信号通知去块滤波器参数(`dbf_info_in_ph_flag`等于0),则对去块滤波器参数进行解码。

[0276] 对标志`slice_ts_residual_coding_disabled_flag`进行系统解码,以知道是否针对当前条带启用变换跳过残差编码方法。

[0277] 如果在图片头部中启用LMCS(`ph_lmcs_enabled_flag`等于1),则对标志`slice_lmcs_enabled_flag`进行解码。

[0278] 以相同的方式,如果在图片头部中启用缩放列表(`phpic_scaling_list_present_enabled_flag`等于1),则对标志`slice_scaling_list_present_flag`进行解码。

[0279] 然后,如果需要,则对其他参数进行解码。

[0280] 图片头部在条带头部中

[0281] 以特定的信号通知方式,如图7中所描绘,可以在条带头部(710)内用信号通知图片头部(708)。在这种情况下,不存在仅包含图片头部(608)的NAL单元。NAL单元701-707对应于图6中的相应NAL单元601-607。类似地,编码区块720和编码块740对应于图6的块620和640。因此,这里将不重复这些单元和块的说明。由于标志`picture_header_in_slice_header_flag`,可以在条带头部中启用。此外,当在条带头部内用信号通知图片头部时,图片应仅包含一个条带。因此,每个图片总是只有一个图片头部。此外,标志`picture_header`

in\_slice\_header\_flag对于CLVS(编码层视频序列)的所有图片应具有相同的值。这意味着包括第一IRAP的两个IRAP之间的所有图片针对每个图片仅具有一个条带。

[0282] 标志picture\_header\_in\_slice\_header\_flag定义如下:

[0283] “picture\_header\_in\_slice\_header\_flag等于1指定在条带头部中存在PH句法结构.picture\_header\_in\_slice\_header\_flag等于0指定在条带头部中不存在PH句法结构。

[0284] picture\_header\_in\_slice\_header\_flag的值在CLVS中的所有经编码条带中应相同是位流一致性的要求。

[0285] 当针对经编码条带、picture\_header\_in\_slice\_header\_flag等于1时,CLVS中不应存在nal\_unit\_type等于PH\_NUT的VCL NAL单元是位流一致性的要求。

[0286] 当picture\_header\_in\_slice\_header\_flag等于0时,当前图片中的所有经编码条带应使picture\_header\_in\_slice\_header\_flag等于0,并且当前PU应具有PH NAL单元。

[0287] picture\_header\_structure()包含除了填充位rbbsp\_trailing\_bits()之外的picture\_rbsp()的句法元素。”

[0288] 流式传输应用

[0289] 一些流式传输应用仅提取位流的某些部分。这些提取可以是空间的(作为子图片)或时间的(视频序列的子部分)。然后,这些提取的部分可以与其他位流合并。另一些帧通过仅提取一些帧来降低帧频。通常,这些流式传输应用的主要目的是使用最大允许带宽来为最终用户产生最高质量。

[0290] 在VVC中,为了帧速率降低,APS ID编号已经受到限制,以使得帧的新APS ID编号不能用于时间层级中的上层的帧。然而,对于提取位流的部分的流式传输应用,需要跟踪APS ID以确定对于位流的子部分应该保留哪些APS,因为帧(因为IRAP)不重置APS ID的编号。

[0291] LMCS(具有色度缩放的亮度映射)

[0292] 具有色度缩放的亮度映射(LMCS)技术是在如VVC的视频解码器中应用环路滤波器之前应用于块的样本值转换方法。

[0293] LMCS可以分成两个子工具。第一子工具应用于亮度块,而第二子工具应用于色度块,如下所述:

[0294] 1) 第一子工具是基于自适应分段线性模型的亮度分量的环内映射。亮度分量的环内映射通过跨动态范围重新分布码字来调整输入信号的动态范围以提高压缩效率。亮度映射利用到“映射域”中的前向映射函数和返回到“输入域”中的相应逆向映射函数。

[0295] 2) 第二子工具与应用亮度相关色度残差缩放的色度分量相关。色度残差缩放被设计为补偿亮度信号与其相应的色度信号之间的交互。色度残差缩放取决于当前块的上部和/或左侧重建的相邻亮度样本的平均值。

[0296] 如同视频编码器(如VVC)中的大多数其他工具,可以使用SPS标志在序列级别启用/禁用LMCS。还在条带级别用信号通知是否启用色度残差缩放。如果启用亮度映射,则用信号通知附加标志以指示是否启用亮度相关色度残差缩放。当不使用亮度映射时,亮度相关色度残差缩放被完全禁用。另外,对于大小小于或等于4的色度块,始终禁用亮度相关色度残差缩放。

[0297] 图8示出如上针对亮度映射子工具所描述的LMCS的原理。图8中的阴影块是新的

LMCS功能块,包括亮度信号的前向和逆向映射。重要的是要注意,当使用LMCS时,在“映射域”中应用一些解码操作。这些操作由该图8中的虚线块表示。它们通常对应于逆量化、逆变换、亮度帧内预测和重建步骤(其在于将亮度预测与亮度残差相加)。相反,图8中的实线块指示在原始(即,非映射)域中应用解码处理的位置,并且这包括诸如去块、ALF和SAO的环路滤波、运动补偿预测以及经解码图片作为参考图片(DPB)的存储。

[0298] 图9示出与图8类似的图,但是这次这是针对LMCS工具的色度缩放子工具。图9中的阴影块是新的LMCS功能块,其包括亮度相关色度缩放处理。然而,在色度方面,与亮度情况相比存在一些重要差异。这里,对于色度样本,仅在“映射域”中进行由虚线中的块表示的逆量化和逆变换。在原始域中进行帧内色度预测、运动补偿、环路滤波的所有其他步骤。如图9所示,对于亮度映射,仅存在缩放处理,并且不存在前向和逆向处理。

[0299] 使用分段线性模型的亮度映射

[0300] 亮度映射子工具使用分段线性模型。这意味着分段线性模型将输入信号动态范围分成16个相等的子范围,并且对于各个子范围,使用分配给该范围的码字的数量来表示其线性映射参数。

[0301] 亮度映射的语义

[0302] 句法元素`lmcs_min_bin_idx`指定在具有色度缩放的亮度映射(LMCS)的构建处理中使用的最小bin(区间)索引。`lmcs_min_bin_idx`的值应在0至15的范围内(包含端值)。

[0303] 句法元素`lmcs_delta_max_bin_idx`指定15与在具有色度缩放的亮度映射的构建处理中使用的最大bin索引`LmcsMaxBinIdx`之间的增量值。`lmcs_delta_max_bin_idx`的值应在0至15的范围内(包含端值)。`LmcsMaxBinIdx`的值被设置为等于 $15 - \text{lmcs\_delta\_max\_bin\_idx}$ 。`LmcsMaxBinIdx`的值应大于或等于`lmcs_min_bin_idx`。

[0304] 句法元素`lmcs_delta_cw_prec_minus1`加1指定用于表示句法`lmcs_delta_abs_cw[i]`的位的数量。

[0305] 句法元素`lmcs_delta_abs_cw[i]`指定第*i*个bin的绝对增量码字值。

[0306] 句法元素`lmcs_delta_sign_cw_flag[i]`指定变量`lmcsDeltaCW[i]`的符号。当`lmcs_delta_sign_cw_flag[i]`不存在时,推断其等于0。

[0307] 用于亮度映射的LMCS中间变量计算

[0308] 为了应用前向和逆向亮度映射处理,需要一些中间变量和数据阵列。

[0309] 首先,如下导出变量`OrgCW`:

[0310]  $\text{OrgCW} = (1 \ll \text{BitDepth}) / 16$

[0311] 然后,变量`lmcsDeltaCW[i]`(其中 $i = \text{lmcs\_min\_bin\_idx} \cdots \text{LmcsMaxBinIdx}$ )计算如下:

[0312]  $\text{lmcsDeltaCW}[i] = (1 - 2 * \text{lmcs\_delta\_sign\_cw\_flag}[i]) * \text{lmcs\_delta\_abs\_cw}[i]$

[0313] 新变量`lmcsCW[i]`导出如下:

[0314] -对于 $i = 0 \cdots \text{lmcs\_min\_bin\_idx} - 1$ ,`lmcsCW[i]`被设置为等于0。

[0315] -对于 $i = \text{lmcs\_min\_bin\_idx} \cdots \text{LmcsMaxBinIdx}$ ,应用以下:

[0316]  $\text{lmcsCW}[i] = \text{OrgCW} + \text{lmcsDeltaCW}[i]$

[0317] `lmcsCW[i]`的值应在 $(\text{OrgCW} \gg 3)$ 至 $(\text{OrgCW} \ll 3 - 1)$ 的范围内(包含端值)。

[0318] -对于 $i = \text{LmcsMaxBinIdx} + 1 \cdots 15$ ,`lmcsCW[i]`被设置为等于0。

[0319] 变量InputPivot[i] (其中 $i=0\cdots 16$ ) 导出如下:

[0320]  $\text{InputPivot}[i] = i * \text{OrgCW}$

[0321] 变量LmcsPivot[i] (其中 $i=0\cdots 16$ )、变量ScaleCoeff[i]和InvScaleCoeff[i] (其中 $i=0\cdots 15$ ) 如下计算:

$\text{LmcsPivot}[0] = 0;$

for( $i = 0; i \leq 15; i++$ ) {

$\text{LmcsPivot}[i+1] = \text{LmcsPivot}[i] + \text{lmcsCW}[i]$

$\text{ScaleCoeff}[i] = (\text{lmcsCW}[i] * (1 \ll 11) + (1 \ll (\text{Log2}(\text{OrgCW}) - 1))) \gg$

[0322]  $(\text{Log2}(\text{OrgCW}))$

if( $\text{lmcsCW}[i] == 0$ )

$\text{InvScaleCoeff}[i] = 0$

else

$\text{InvScaleCoeff}[i] = \text{OrgCW} * (1 \ll 11) / \text{lmcsCW}[i]$

[0323] 前向亮度映射

[0324] 如图8所示,当LMCS应用于亮度时,从预测样本 $\text{predSamples}[i][j]$ 获得称为 $\text{predMapSamples}[i][j]$ 的亮度重映射样本。

[0325]  $\text{predMapSamples}[i][j]$  计算如下:

[0326] 首先,从位置( $i, j$ )处的预测样本 $\text{predSamples}[i][j]$ 计算索引 $\text{idxY}$ 。

[0327]  $\text{idxY} = \text{predSamples}[i][j] \gg \text{Log2}(\text{OrgCW})$

[0328] 然后,通过使用部分0的中间变量 $\text{idxY}$ 、 $\text{LmcsPivot}[\text{idxY}]$ 和 $\text{InputPivot}[\text{idxY}]$ 如下导出 $\text{predMapSamples}[i][j]$ :

[0329]  $\text{predMapSamples}[i][j] = \text{LmcsPivot}[\text{idxY}]$

[0330]  $+ (\text{ScaleCoeff}[\text{idxY}] * (\text{predSamples}[i][j] - \text{InputPivot}[\text{idxY}]) + (1 \ll 10)) \gg 11$

[0331] 亮度重建样本

[0332] 从经预测亮度样本 $\text{predMapSample}[i][j]$ 和残差亮度样本 $\text{resiSamples}[i][j]$ 获得重建处理。

[0333] 通过如下将 $\text{predMapSample}[i][j]$ 相加到 $\text{resiSample}[i][j]$ 来简单地获得经重建亮度图片样本 $\text{recSamples}[i][j]$ :

[0334]  $\text{recSamples}[i][j] = \text{Clip1}(\text{predMapSamples}[i][j] + \text{resiSamples}[i][j])$

[0335] 在上述关系中,Clip 1函数是裁剪函数,以确保重建样本在0和 $1 \ll \text{BitDepth} - 1$ 之间。

[0336] 逆向亮度映射

[0337] 当应用根据图8的逆向亮度映射时,对正在处理的当前块的各个样本 $\text{recSample}[i][j]$ 应用以下操作:

[0338] 首先,从位置( $i, j$ )处的重建样本 $\text{recSamples}[i][j]$ 计算索引 $\text{idxY}$ 。

[0339]  $\text{idxY} = \text{recSamples}[i][j] \gg \text{Log2}(\text{OrgCW})$

[0340] 逆向映射亮度样本 $\text{invLumaSample}[i][j]$ 基于以下导出:

[0341]  $\text{invLumaSample}[i][j] =$

[0342]  $\text{InputPivot}[\text{idxYInv}] + (\text{InvScaleCoeff}[\text{idxYInv}] *$



[0343]  $(\text{recSample}[i][j] - \text{LmcsPivot}[\text{idxYInv}]) + (1 \ll 10) \gg 11$

[0344] 然后进行裁剪操作以获得最终样本:

[0345]  $\text{finalSample}[i][j] = \text{Clip1}(\text{invLumaSample}[i][j])$

[0346] 色度缩放

[0347] 用于色度缩放的LMCS语义

[0348] 表6中的句法元素`lmcs_delta_abs_crs`指定变量`lmcsDeltaCrs`的绝对码字值。`lmcs_delta_abs_crs`的值应在0与7的范围内(包含端值)。当不存在时,推断`lmcs_delta_abs_crs`等于0。

[0349] 句法元素`lmcs_delta_sign_crs_flag`指定变量`lmcsDeltaCrs`的符号。当不存在时,推断`lmcs_delta_sign_crs_flag`等于0。

[0350] 用于色度缩放的LMCS中间变量计算

[0351] 为了应用色度缩放处理,需要一些中间变量。

[0352] 变量`lmcsDeltaCrs`导出如下:

[0353]  $\text{lmcsDeltaCrs} = (1 - 2 * \text{lmcs\_delta\_sign\_crs\_flag}) * \text{lmcs\_delta\_abs\_crs}$

[0354] 变量`ChromaScaleCoeff[i]` (其中 $i = 0 \cdots 15$ ) 如下导出:

```

if( lmcsCW[ i ] == 0 )
    ChromaScaleCoeff[ i ] = ( 1 << 11 )
[0355] else
    ChromaScaleCoeff[ i ] = OrgCW * ( 1 << 11 ) / ( lmcsCW[ i ] + lmcsDeltaCrs )

```

[0356] 色度缩放处理

[0357] 在第一步骤中,导出变量`invAvgLuma`,以计算当前相应色度块周围的重建亮度样本的平均亮度值。平均亮度是从围绕相应色度块的左侧亮度块和上部亮度块计算的。

[0358] 如果没有样本可用,则变量`invAvgLuma`设置如下:

[0359]  $\text{invAvgLuma} = 1 \ll (\text{BitDepth} - 1)$

[0360] 基于部分0的中间阵列`LmcsPivot[ ]`,然后如下导出变量`idxYInv`:

```

For ( idxYInv = lmcs_min_bin_idx; idxYInv <= LmcsMaxBinIdx; idxYInv++ ) {
    if( invAvgLuma < LmcsPivot [ idxYInv + 1 ] ) break
[0361] }
IdxYInv = Min( idxYInv, 15 )

```

[0362] 变量`varScale`导出如下:

[0363]  $\text{varScale} = \text{ChromaScaleCoeff}[\text{idxYInv}]$

[0364] 当对当前色度块应用变换时,如下导出重建色度图片样本阵列`recSamples`:

[0365]  $\text{recSamples}[i][j] = \text{Clip1}(\text{predSamples}[i][j] +$

[0366]  $\text{Sign}(\text{resiSamples}[i][j]) * ((\text{Abs}(\text{resiSamples}[i][j]) * \text{varScale} + (1 \ll 10)) \gg$

[0367]  $11))$

[0367] 如果尚未对当前块应用变换,则应用以下:

[0368]  $\text{recSamples}[i][j] = \text{Clip1}(\text{predSamples}[i][j])$

[0369] 编码器考虑因素

[0370] LMCS编码器的基本原理是首先将更多的码字分配给那些动态范围段具有比平均方差更低的码字的范围。在这个的替代构想中,LMCS的主要目标是向具有比平均方差更高的码字的那些动态范围段分配更少的码字。以这种方式,图片的平滑区域将用比平均值更多的码字来编码,反之亦然。

[0371] 在编码器侧确定存储在APS中的LMCS工具的所有参数(参见表6)。LMCS编码器算法基于局部亮度方差的评估,并且根据上述基本原理优化LMCS参数的确定。然后进行优化以获得给定块的最终重建样本的最佳PSNR度量。

[0372] 实施例

[0373] 在不需要时避免条带地址句法元素

[0374] 在一个实施例中,当在条带头部中用信号通知图片头部时,即使区块的数量大于1,也推断条带地址句法元素(slice\_address)等于值0,。表11示出该实施例。

[0375] 该实施例的优点在于,当图片头部在条带头部中时,不解析条带地址,这降低了位速率,特别是对于低延迟和低位速率应用,并且当在条带头部中用信号通知图片时,降低了一些实现的解析复杂性。

[0376] 在实施例中,这仅应用于光栅扫描条带模式(rect\_slice\_flag等于0)。这降低了一些实现的解析复杂性。

[0377] 表11示出修改的部分条带头部

[0378]	slice_header() {	描述符
	picture_header_in_slice_header_flag	u(1)
	if( picture_header_in_slice_header_flag )	
	picture_header_structure( )	
	...	
	if( ( rect_slice_flag && NumSlicesInSubpic[ CurrSubpicIdx ] > 1 )    (( !rect_slice_flag && NumTilesInPic > 1 ) && !picture_header_in_slice_header_flag ) )	
	slice_address	u(v)
	...	

[0379] 在不需要时避免发送条带中的区块的数量

[0380] 在一个实施例中,当在条带头部中发送图片头部时,不发送条带中的区块的数量。表12示出该实施例,其中当标志picture\_header\_in\_slice\_header\_flag被设置为等于1时,不发送num\_tiles\_in\_slice\_minus1句法元素。该实施例的优点是位速率降低,特别是对于低延迟和低位速率应用,因为不需要发送区块的数量。

[0381] 在实施例中,这仅应用于光栅扫描条带模式(rect\_slice\_flag等于0)。这降低了一些实现的解析复杂性。

[0382] 表12示出修改的部分条带头部

[0383]

slice_header() {	描述符
<b>picture_header_in_slice_header_flag</b>	u(1)
if( picture_header_in_slice_header_flag )	
picture_header_structure()	
...	
if( ( !rect_slice_flag && NumTilesInPic > 1 )	
&& !picture_header_in_slice_header_flag )	
<b>num_tiles_in_slice_minus1</b>	ue(v)
...	

- [0384] 由PPS值NumTilesInPic(语义)预测
- [0385] 在一个附加实施例中,当在条带头部中发送图片头部时,推断当前条带中的区块的数量等于图片中的区块的数量。这可以通过在句法元素num\_tiles\_in\_slice\_minus1的语义中添加以下句子来设置:“当不存在时,变量num\_tiles\_in\_slice\_minus1被设置为等于NumTilesInPic-1”。
- [0386] 其中变量NumTilesInPic给出图片的区块的最大数量。基于在PPS中发送的句法元素来计算该变量。
- [0387] 在条带地址之前设置区块的数量并且避免slice\_address的不需要发送
- [0388] 在一个实施例中,专用于条带中的区块的数量的句法元素在条带地址之前被发送,并且其值用于知道是否需要解码条带地址。更确切地,将条带中的区块的数量与图片中的区块的数量进行比较,以知道是否需要解码条带地址。实际上,如果条带中的区块的数量等于图片中的区块的数量,则确保当前图片仅包含一个条带。
- [0389] 在实施例中,这仅应用于光栅扫描条带模式(rect\_slice\_flag等于0)。这降低了一些实现的解析复杂性。
- [0390] 表13示出该实施例。其中如果句法元素num\_tiles\_in\_slice\_minus1的值等于变量NumTilesInPic减1,则不解码句法元素slice\_address。当num\_tiles\_in\_slice\_minus1等于变量NumTilesInPic减1时,推断slice\_address等于0。
- [0391] 表13示出修改的部分条带头部

[0392]	slice_header() {	描述符
	<b>picture_header_in_slice_header_flag</b>	u(1)
	if( picture_header_in_slice_header_flag )	
	picture_header_structure()	
	...	
	if( !rect_slice_flag && NumTilesInPic > 1 )	
	<b>num_tiles_in_slice_minus1</b>	ue(v)
	if( ( rect_slice_flag && NumSlicesInSubpic[ CurrSubpicIdx ] > 1 )    (( !rect_slice_flag && NumTilesInPic > 1 ) && <u>num_tiles_in_slice_minus1 != NumTilesInPic-1</u> )	
	<b>slice_address</b>	u(v)
	...	

[0393] 该实施例的优点是当条件被设置为等于真时,位速率降低和解析复杂性降低,因为不发送条带地址。

[0394] 在一个实施例中,当在条带头部中发送图片头部时,不解码指示当前条带中的区块的数量的句法元素,并且推断条带中的区块的数量等于1。以及当条带中的区块的数量等于图片中的区块的数量时,推断条带地址等于0,并且不解码相关句法元素。表14示出该实施例。

[0395] 这增加了通过这两个实施例的组合获得的位速率降低。

[0396] 表14示出修改的部分条带头部

[0397]	slice_header() {	描述符
	<b>picture_header_in_slice_header_flag</b>	u(1)
	if( picture_header_in_slice_header_flag )	
	picture_header_structure()	
	...	
	if( ( !rect_slice_flag && NumTilesInPic > 1 ) && !picture_header_in_slice_header_flag )	
	<b>num_tiles_in_slice_minus1</b>	ue(v)
	if( ( rect_slice_flag && NumSlicesInSubpic[ CurrSubpicIdx ] > 1 )    (( !rect_slice_flag && NumTilesInPic > 1 ) && <u>num_tiles_in_slice_minus1 != NumTilesInPic-1</u> )	
	<b>slice_address</b>	u(v)
	...	

[0398] 移除不需要的条件NumTilesInPic>1

[0399] 在一个实施例中,当启用光栅扫描条带模式时,不需要测试当前图片中的区块的数量确实需要大于1的条件,以解码句法元素slice\_address和/或当前条带中的区块的数量。具体地,在当前图片中的区块的数量等于1时,推断rect\_slice\_flag值等于1。因此,在

这种情况下不能启用光栅扫描条带模式。表15示出该实施例。

[0400] 该实施例降低了条带头部的解析复杂性。

[0401] 表15示出修改的部分条带头部

[0402]	slice_header() {	描述符
	...	
	if( ( rect_slice_flag && NumSlicesInSubpic[ CurrSubpicIdx ] > 1 )    ( !rect_slice_flag && NumTilesInPic > 1 ) )	
	<b>slice_address</b>	u(v)
	for( i = 0; i < NumExtraShBits; i++ )	
	<b>sh_extra_bit[ i ]</b>	u(1)
	if( !rect_slice_flag && NumTilesInPic > 1 )	
	<b>num_tiles_in_slice_minus1</b>	ue(v)
	...	

[0403] 在一个实施例中,当在条带头部中发送图片头部时并且当启用光栅扫描条带模式时,不解码指示当前条带中的区块的数量的句法元素,并且推断条带中的区块的数量等于1。以及当条带中的区块的数量等于图片中的区块的数量时并且当启用光栅扫描条带模式时,推断条带地址等于0,并且不解码相关句法元素slice\_address。表16示出该实施例。

[0404] 优点是位速率降低和解析复杂性降低。

[0405] 表16示出修改的部分条带头部

[0406]	slice_header() {	描述符
	<b>picture_header_in_slice_header_flag</b>	u(1)
	if( picture_header_in_slice_header_flag )	
	picture_header_structure( )	
	...	
	if( !rect_slice_flag && !picture_header_in_slice_header_flag )	
	<b>num_tiles_in_slice_minus1</b>	ue(v)
	if( ( rect_slice_flag && NumSlicesInSubpic[ CurrSubpicIdx ] > 1 )    ( !rect_slice_flag && num_tiles_in_slice_minus1 != NumTilesInPic-1 ) )	
	<b>slice_address</b>	u(v)
	...	

[0407] 实现

[0408] 图11示出了根据本发明实施例的系统191、195,其包括编码器150或解码器100中的至少一个以及通信网络199。根据实施例,系统195用于处理并向用户提供内容(例如,用于显示/输出或流式传输视频/音频内容的视频和音频内容),用户例如通过包括解码器100的用户终端或可与解码器100通信的用户终端的用户界面访问解码器100。这样的用户终端可以是计算机、移动电话、平板电脑或能够向用户提供/显示(提供的/流式传输的)内容的

任何其他类型的装置。系统195经由通信网络199获得/接收位流101(以连续流或信号的形式(例如,在显示/输出较早的视频/音频时))。根据实施例,系统191用于处理内容并存储经处理的内容,例如经处理以供在稍后的时间显示/输出/流式传输的视频和音频内容。系统191获得/接收包括原始图像序列151的内容,该内容由编码器150接收和处理(包括利用根据本发明的去块滤波器进行滤波),并且编码器150生成将经由通信网络199传送到解码器100的位流101。然后,位流101以多种方式传送到解码器100,例如,可以由编码器150预先生成并作为数据存储在通信网络199中的存储设备中(例如,在服务器或云存储装置上),直到用户从存储设备请求内容(即,位流数据)为止,此时数据从存储设备传送/流式传输到解码器100。系统191还可以包括内容提供设备,以用于向用户提供/流式传输(例如,通过传送要在用户终端上显示的用户界面的数据)存储在存储设备中的内容的内容信息(例如,内容的标题和用于识别、选择和请求内容的其他元/存储位置数据),并且用于接收和处理用户对内容的请求,使得所请求的内容可以从存储设备传送/流式传输到用户终端。可替代地,编码器150生成位流101,并且在用户请求内容时将其直接传送/流式传输到解码器100。然后,解码器100接收位流101(或信号),并利用根据本发明的去块滤波器进行滤波,以获得/生成视频信号109和/或音频信号,然后用户终端使用视频信号109和/或音频信号来向用户提供所请求的内容。

[0409] 根据本发明的方法/处理的任何步骤或本文描述的功能可以用硬件、软件、固件或其任何组合来实现。如果以软件实施,则步骤/功能可以作为一个或多个指令或代码或程序或计算机可读介质而存储在一个或多个基于硬件的处理单元上或者经由一个或多个基于硬件的处理单元发送,并且由一个或多个基于硬件的处理单元执行,所述处理单元诸如为可编程计算机,其可以是PC(“个人计算机”)、DSP(“数字信号处理器”)、电路、电路系统、处理器和存储器、通用微处理器或中央处理单元、微控制器、ASIC(“专用集成电路”)、现场可编程逻辑阵列(FPGA)或其他等效集成或离散逻辑电路系统。因此,如本文中所使用的术语“处理器”可指前述结构或适合于实现本文中所描述的技术的任何其他结构中的一个。

[0410] 本发明的实施例还可以通过各种装置或设备来实现,包括无线手机、集成电路(IC)或JC集合(例如,芯片集)。本文中描述各种组件、模块或单元以说明被配置为进行这些实施例的装置/设备的功能方面,但未必需要由不同硬件单元实现。而是,各种模块/单元可组合于编解码器硬件单元中或由互操作硬件单元的集合提供,所述硬件单元包括结合合适软件/固件的一个或多个处理器。

[0411] 本发明的实施例可以通过读出并执行记录在存储介质上的计算机可执行指令(例如,一个或多个程序)以进行上述实施例中的一个或多个模块/单元/功能以及/或者包括用于进行上述实施例中的一个或多个的功能的一个或多个处理单元或电路的系统或设备的计算机来实现,并且可以通过由系统或设备的计算机进行的方法来实现,例如,从存储介质读出并执行计算机可执行指令以进行上述实施例中的一个或多个的功能和/或控制一个或多个处理单元或电路以进行上述实施例中的一个或多个的功能。计算机可以包括单独的计算机或单独的处理单元的网络,以读出并执行计算机可执行指令。计算机可执行指令可以例如经由网络或有形存储介质从诸如通信介质等的计算机可读介质提供给计算机。通信介质可以是信号/位流/载波。有形存储介质是“非暂时性

计算机可读存储介质”，其可以包括(例如)硬盘、随机访问存储器(RAM)、只读存储器(ROM)、分布式计算系统的存储装置、光盘(例如致密盘(CD)、数字多功能光盘(DVD)或蓝光光盘(BD)<sup>TM</sup>)、快闪存储器装置、存储卡等中的一个或多个。至少一些步骤/功能也可以由机器或专用组件(诸如FPGA(“现场可编程门阵列”)或ASIC(“专用集成电路”))在硬件中实现。

[0412] 图12是用于实现本发明的一个或多个实施例的计算装置2000的示意性框图。计算装置2000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2000包括连接到以下各项的通信总线：-中央处理单元(CPU)2001，诸如微处理器等；-用于存储本发明的实施例的方法的可执行代码的随机访问存储器(RAM)2002以及适于记录实现根据本发明的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的寄存器，其存储容量例如可以通过连接到扩展端口的可选RAM进行扩展；-用于存储用于实现本发明的实施例的计算机程序的只读存储器(ROM)2003；-网络接口(NET)2004，其通常连接至通信网络，要处理的数字数据通过该通信网络被传输或接收，网络接口(NET)2004可以是单个网络接口，或者由一组不同的网络接口(例如，有线和无线接口，或不同种类的有线或无线接口)组成，在运行在CPU 2001中的软件应用的控制下，数据包被写入网络接口用于传输或者从网络接口读取以进行接收；-用户接口(UI)2005，其可以用于从用户接收输入或向用户显示信息；-硬盘(HD)2006，其可以被设置为大容量存储装置；-输入/输出模块(IO)2007，其可以用于从/向外部装置(诸如视频源或显示器等)接收/发送数据。可执行代码可以存储在ROM 2003中、HD 2006上或诸如盘等的可移动数字介质上。根据变型，程序的可执行代码可以经由NET 2004借助于通信网络来接收，以在被执行之前存储在计算装置2000的存储部件(诸如HD 2006等)之一中。CPU 2001适于控制和指导根据本发明的实施例的一个或多个程序的软件代码的指令或部分的执行，该指令被存储在前述存储部件之一中。例如，在通电之后，CPU 2001能够执行来自从程序ROM 2003或HD 2006加载了指令之后的主RAM存储器2002的、与软件应用有关的那些指令。这种软件应用在由CPU 2001执行时使得进行根据本发明的方法的步骤。

[0413] 还应理解，根据本发明的其他实施例，在诸如计算机、移动电话(蜂窝电话)、平板或能够向用户提供/显示内容的任何其他类型的装置(例如，显示设备)等的用户终端中提供根据上述实施例的解码器。根据又一实施例，在图像捕获设备中提供根据上述实施例的编码器，该图像捕获设备还包括用于捕获和提供内容以供编码器进行编码的照相机、摄像机或网络照相机(例如，闭路电视或视频监视照相机)。以下参见图13和14提供两个这样的示例。

[0414] 网络照相机

[0415] 图13是例示包括网络照相机2102和客户端设备2104的网络照相机系统2100的图。

[0416] 网络照相机2102包括摄像单元2106、编码部2108、通信单元2110和控制单元2112。

[0417] 网络照相机2102和客户端设备2104经由网络200相互连接以能够彼此通信。

[0418] 摄像单元2106包括镜头和图像传感器(例如，电荷耦合器件(CCD)或互补金属氧化物半导体(CMOS))，并捕获对象的图像并基于该图像生成图像数据。该图像可以是静止图像或视频图像。

[0419] 编码部2108通过使用以上描述的所述编码方法来对图像数据进行编码。

[0420] 网络照相机2102的通信单元2110将由编码部2108编码的经编码的图像数据传输

至客户端设备2104。

[0421] 此外,通信单元2110接收来自客户端设备2104的命令。命令包括用于设置用于编码部2108的编码的参数的命令。

[0422] 控制单元2112根据通信单元2110接收到的命令来控制网络照相机2102中的其他单元。

[0423] 客户端设备2104包括通信单元2114、解码部2116和控制单元2118。

[0424] 客户端设备2104的通信单元2114向网络照相机2102传输命令。

[0425] 此外,客户端设备2104的通信单元2114从网络照相机2102接收经编码的图像数据。

[0426] 解码部2116通过使用以上描述的所述解码方法来对经编码的图像数据进行解码。

[0427] 客户端设备2104的控制单元2118根据由通信单元2114接收的用户操作或命令来控制客户端设备2104中的其他单元。

[0428] 客户端设备2104的控制单元2118控制显示设备2120以显示由解码部2116解码的图像。

[0429] 客户端设备2104的控制单元2118还控制显示设备2120以显示用于指定网络照相机2102的参数(包括用于编码部2108的编码的参数)的值的GUI(图形用户界面)。

[0430] 客户端设备2104的控制单元2118还根据对显示设备2120所显示的GUI的用户操作输入来控制客户端设备2104中的其他单元。

[0431] 客户端设备2104的控制单元2118根据对显示设备2120所显示的GUI的用户操作输入来控制客户端设备2104的通信单元2114,以将用于指定网络照相机2102的参数的值的命令传输至网络照相机2102。

[0432] 智能电话

[0433] 图14是例示智能电话2200的图。

[0434] 智能电话2200包括通信单元2202、解码部2204、控制单元2206、显示单元2208、图像记录装置2210和传感器2212。

[0435] 通信单元2202经由网络200接收经编码的图像数据。

[0436] 解码部2204对通信单元2202接收到的经编码的图像数据进行解码。

[0437] 解码部2204通过使用以上描述的所述解码方法来对经编码的图像数据进行解码。

[0438] 控制单元2206根据通信单元2202接收到的用户操作或命令控制智能电话2200中的其他单元。

[0439] 例如,控制单元2206控制显示单元2208以显示由解码部2204解码的图像。

[0440] 虽然已经参考实施例描述了本发明,但是应当理解,本发明不限于所公开的实施例。本领域技术人员将理解,在不脱离所附权利要求限定的本发明的范围的情况下,可以进行各种改变和修改。本说明书(包括任何所附权利要求、摘要和附图)中公开的所有特征、和/或所公开的任何方法或处理的所有步骤,可以以任何组合进行组合,除了这样的特征和/或步骤中的至少一些相互排斥的组合之外。除非另外明确说明,否则本说明书(包括任何所附权利要求、摘要和附图)中所公开的各个特征可以由用于相同、等同或相似目的的替代特征代替。因此,除非另有明确说明,否则所公开的各个特征仅为通用系列等效或类似特征的一个示例。



[0441] 还应理解,上述比较、确定、评估、选择、执行、进行或考虑的任何结果(例如,在编码或滤波处理期间作出的选择)可以在位流中的数据(例如,指示结果的标志或数据)中指示或可从位流中的数据确定/推断,使得所指示或确定/推断的结果可以用于处理而不是实际上例如在解码处理期间进行比较、确定、评估、选择、执行、进行或考虑。

[0442] 在权利要求中,词语“包括”不排除其他元素或步骤,并且不定冠词“a”或“an”不排除多个。仅仅在相互不同的从属权利要求中记载不同特征的事实并不指示这些特征的组合不能被有利地使用。

[0443] 权利要求中出现的附图标记仅作为说明,并且不应对权利要求的范围产生限定作用。

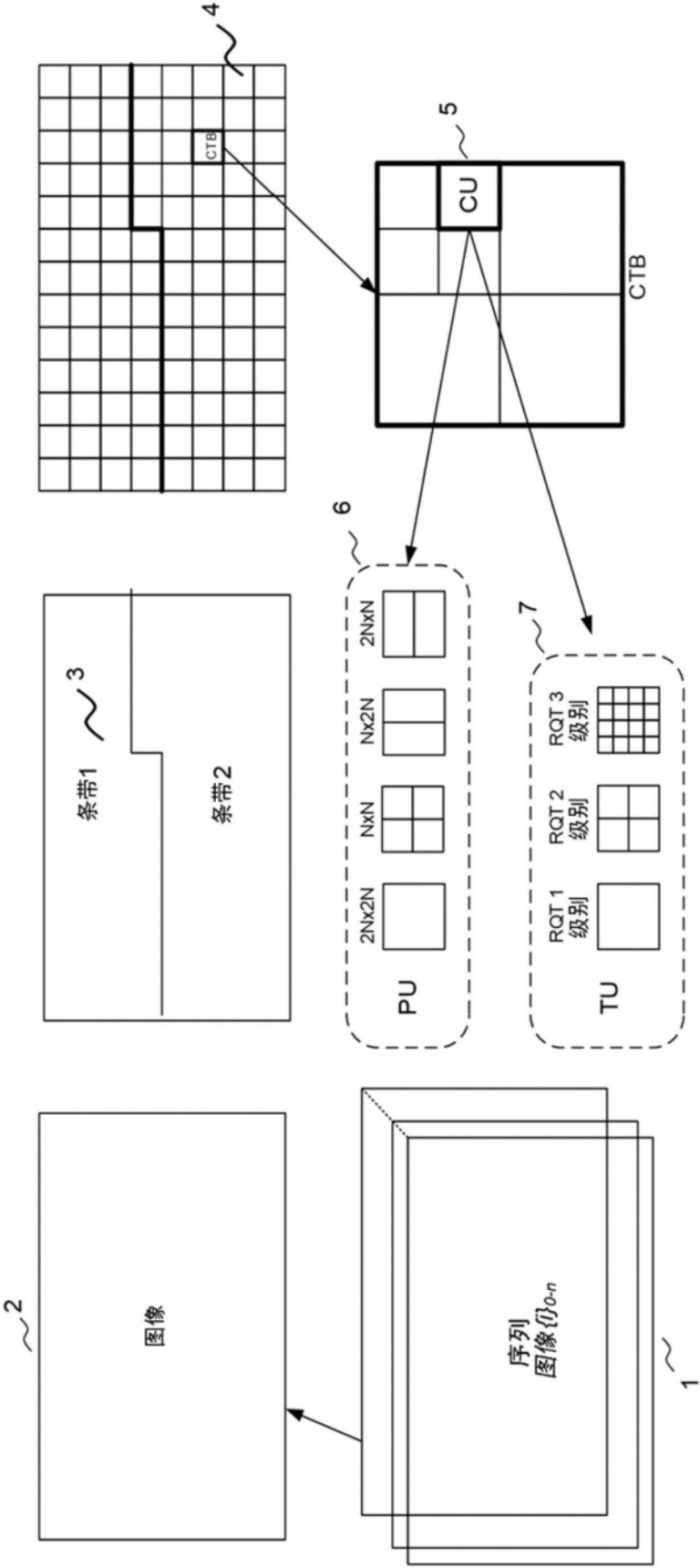


图1

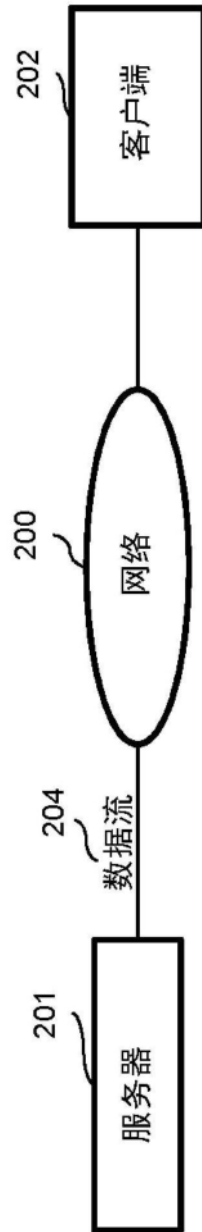


图2

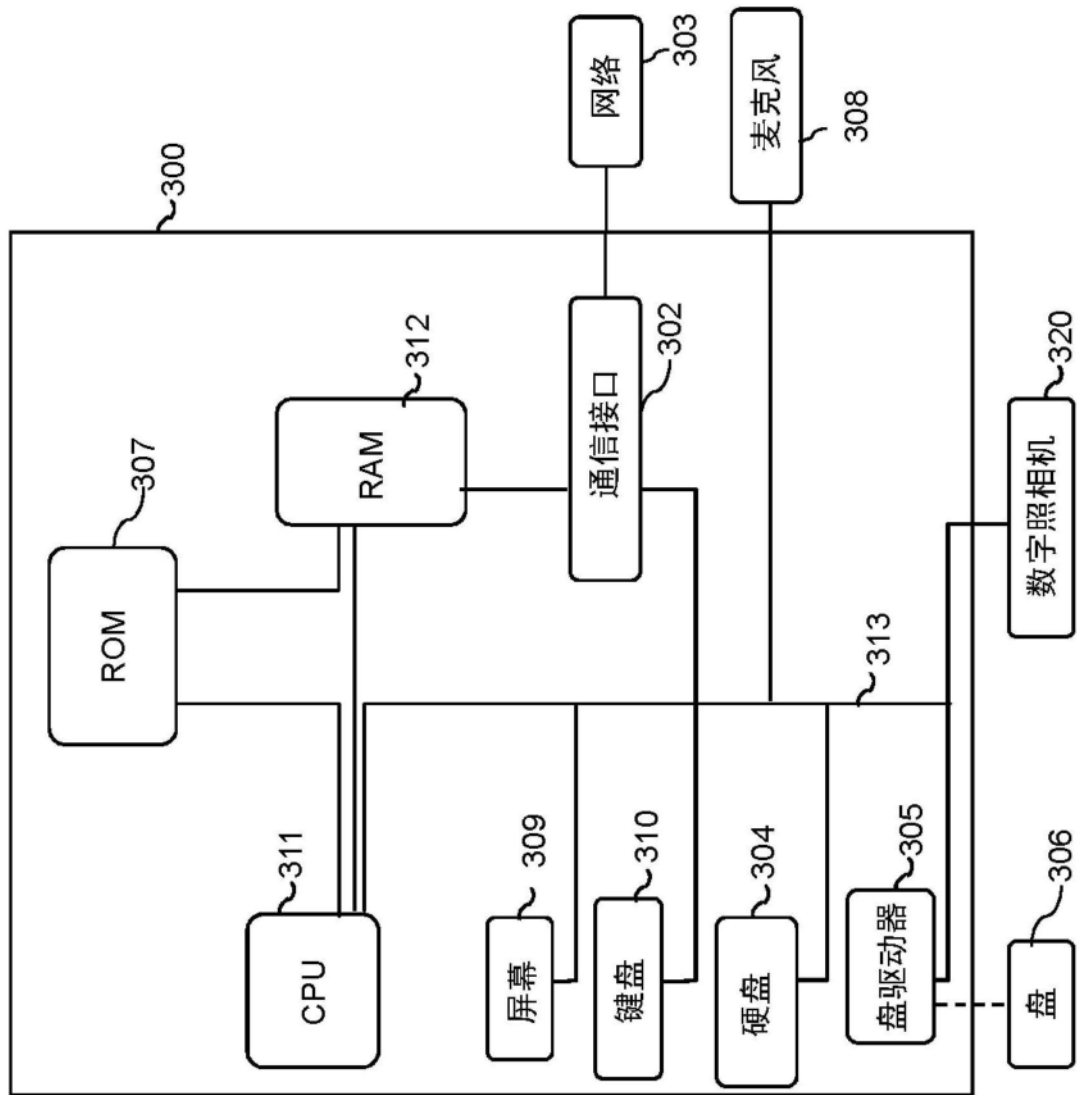


图3

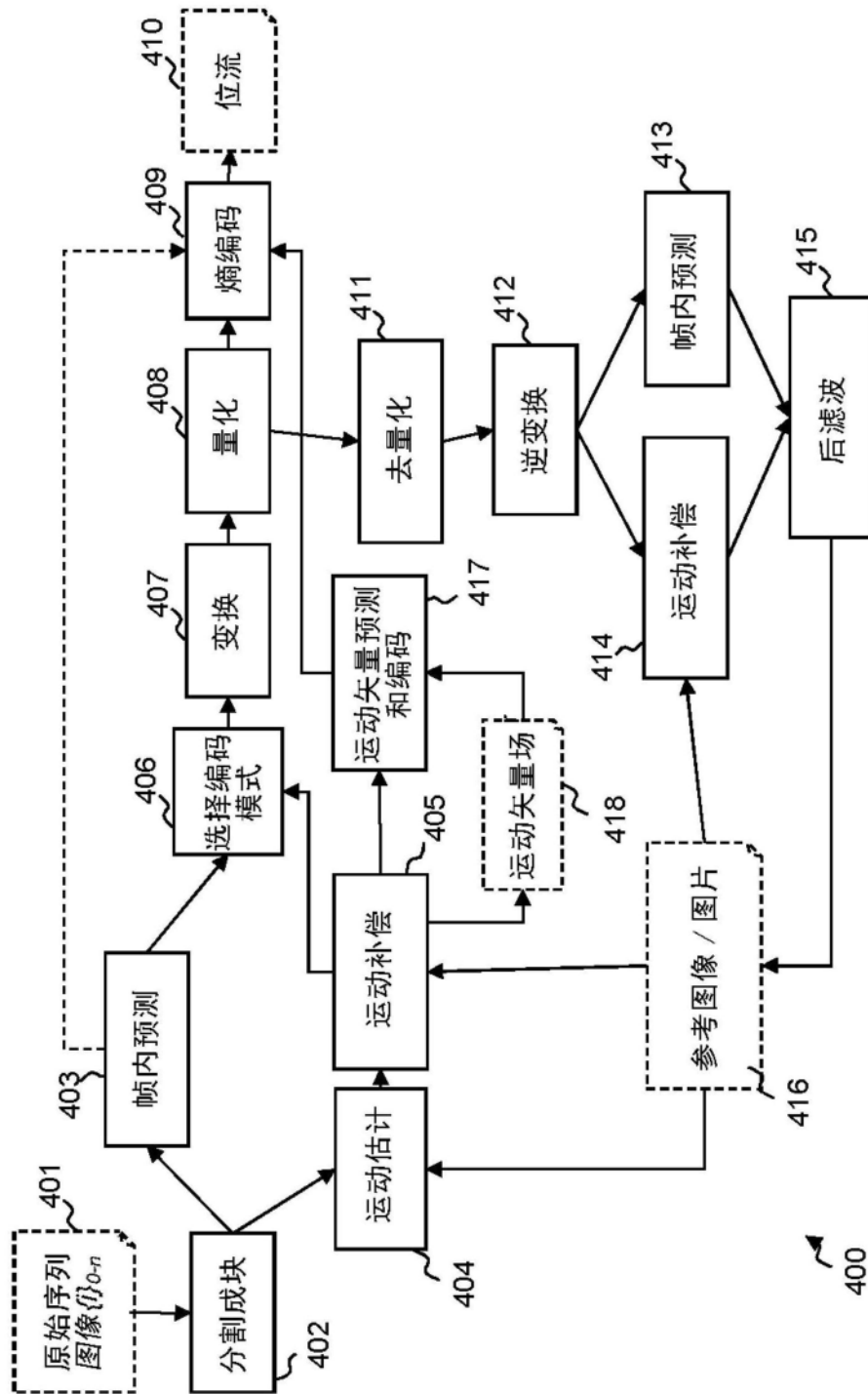


图4

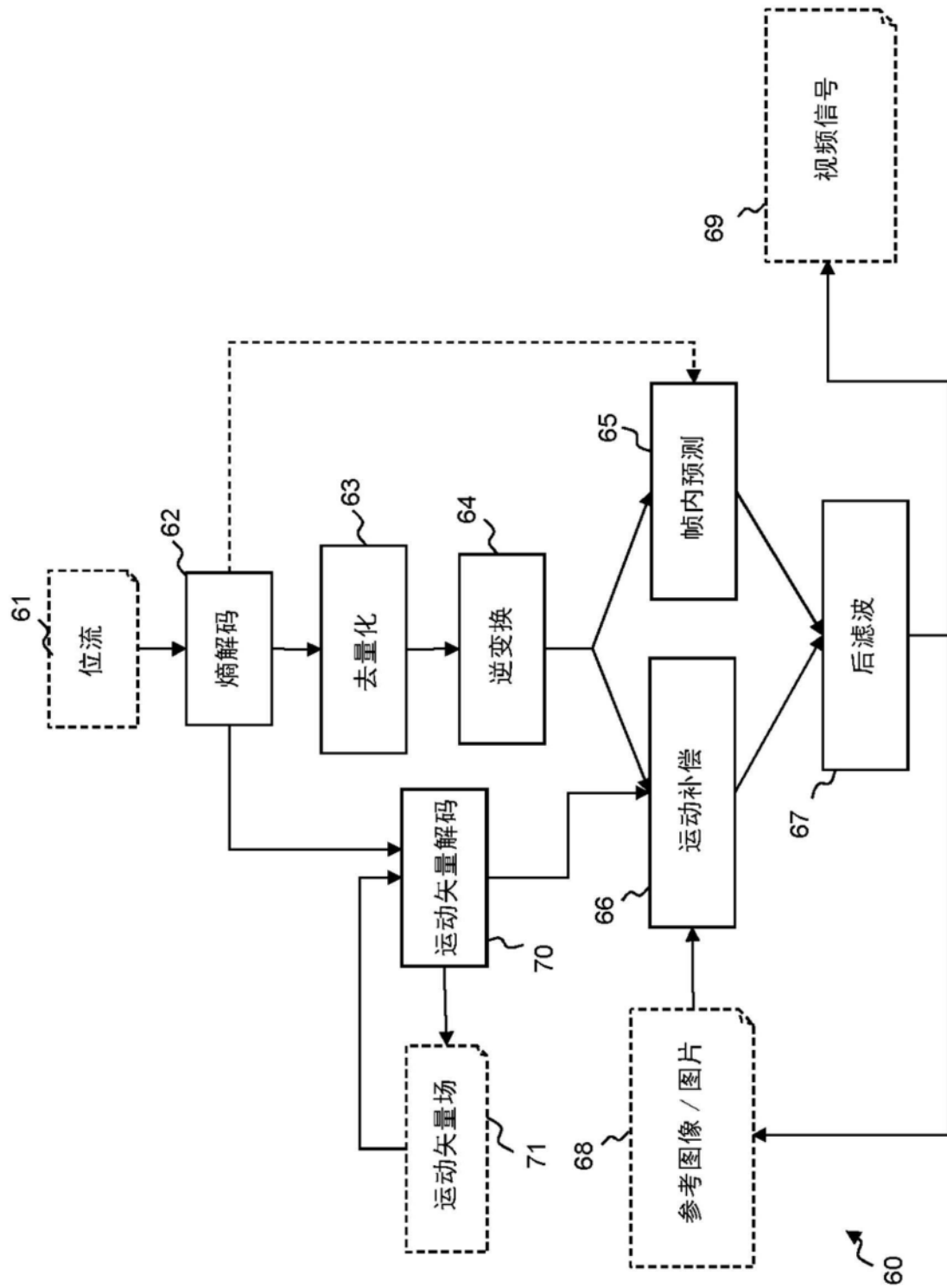


图5

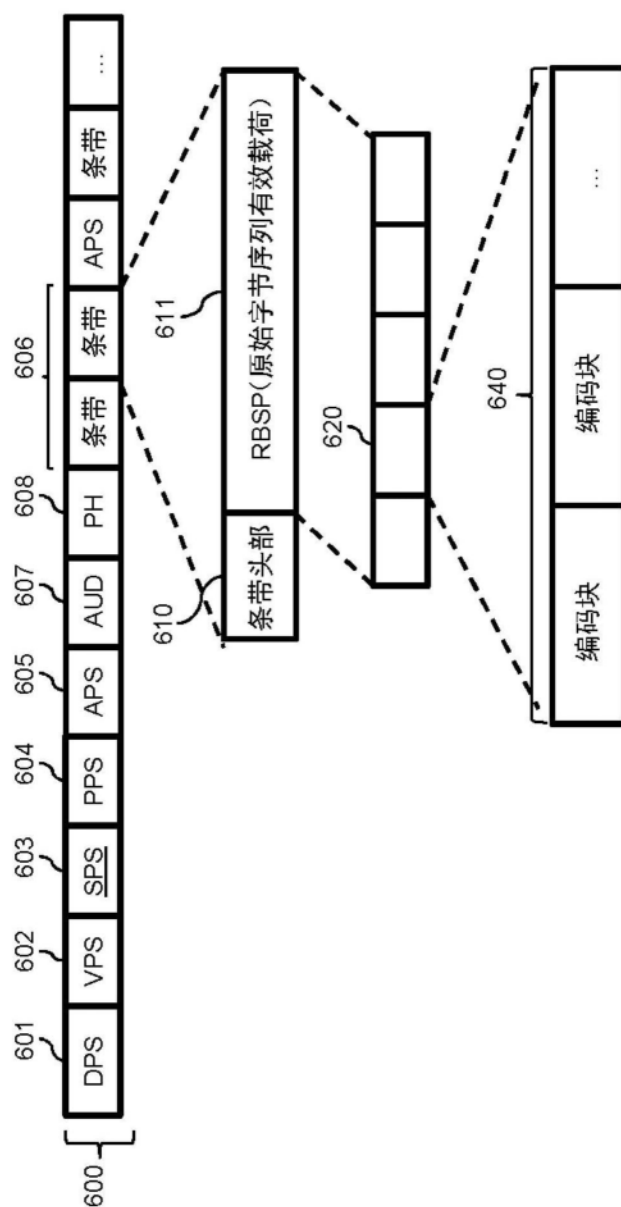


图6

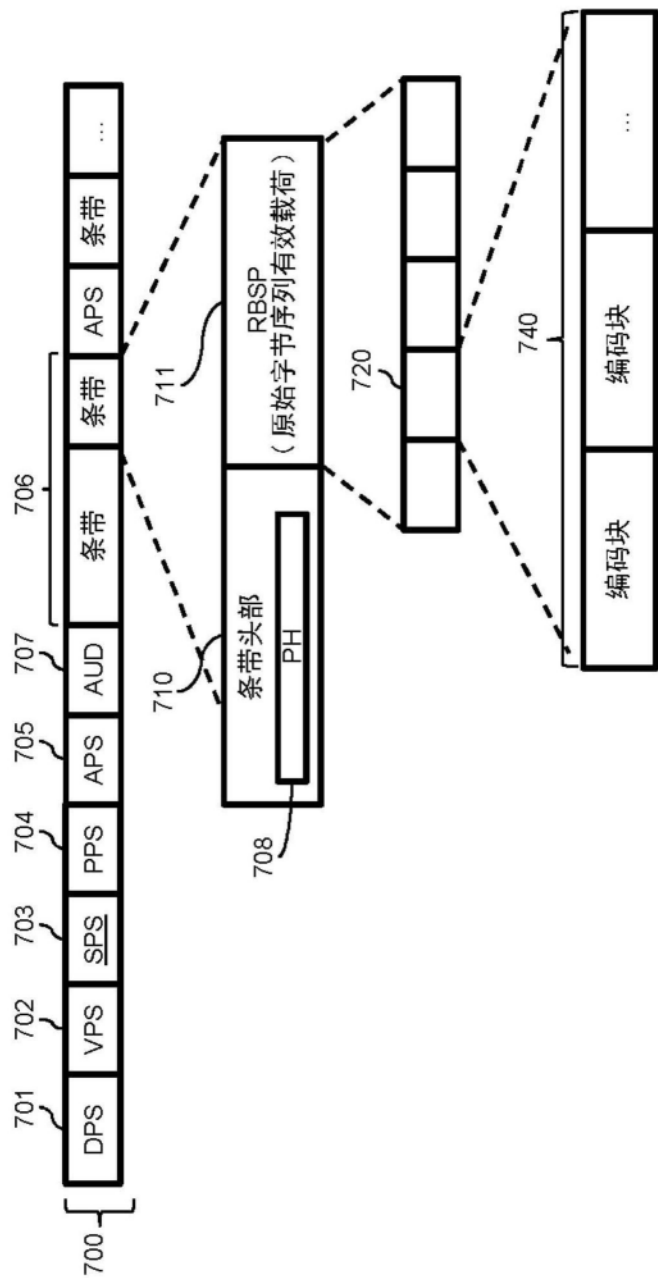


图7



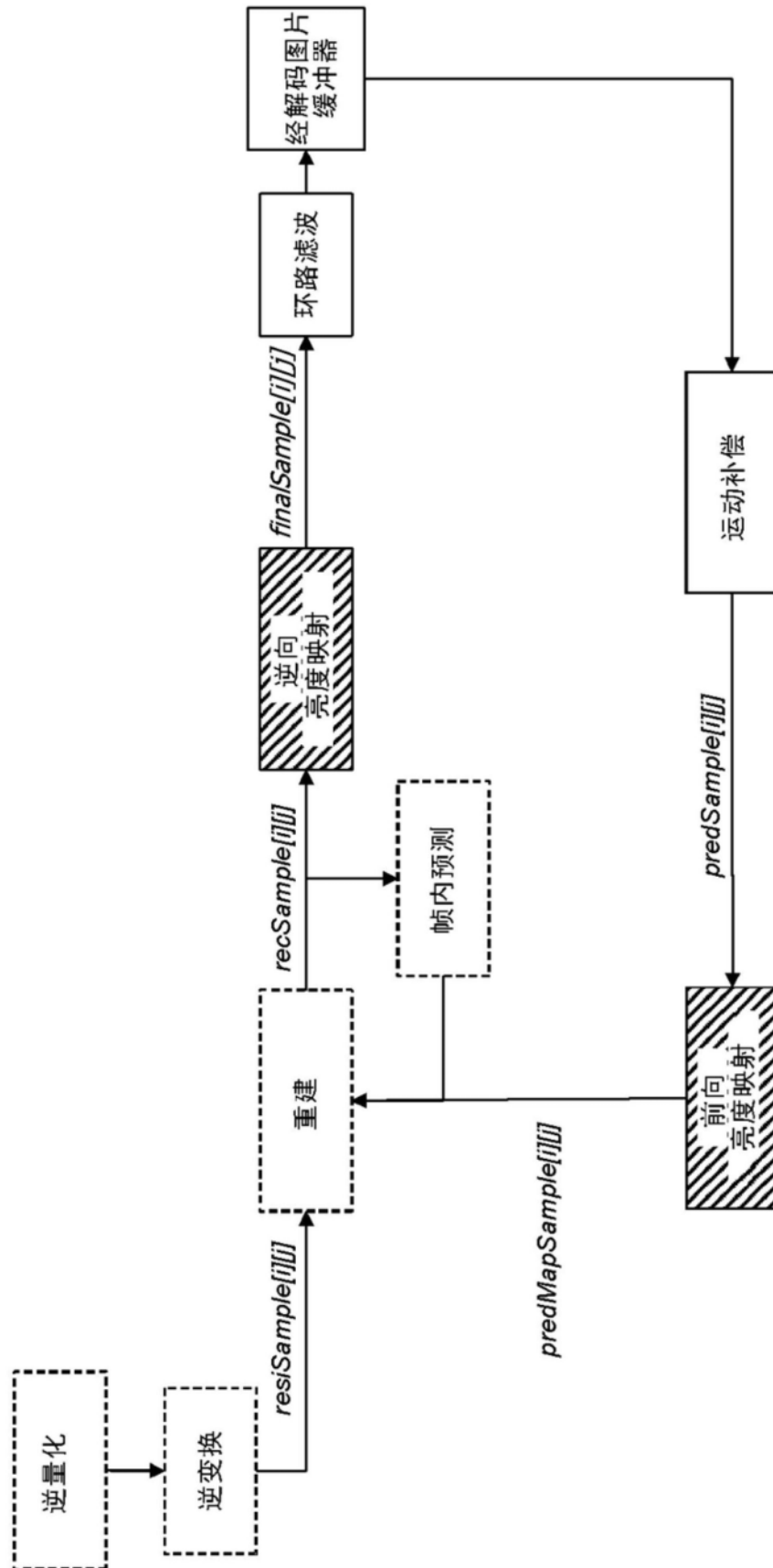


图8

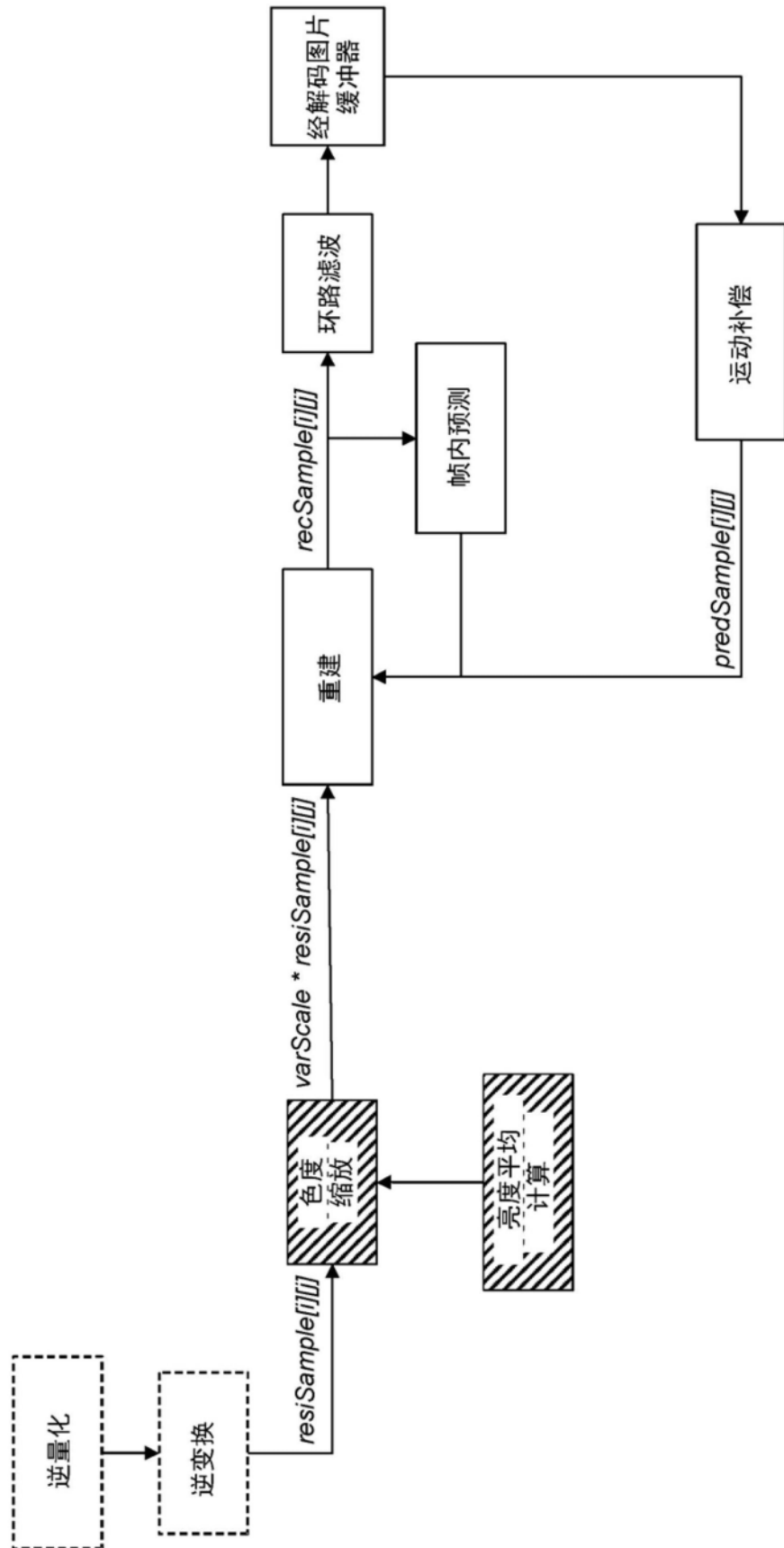


图9

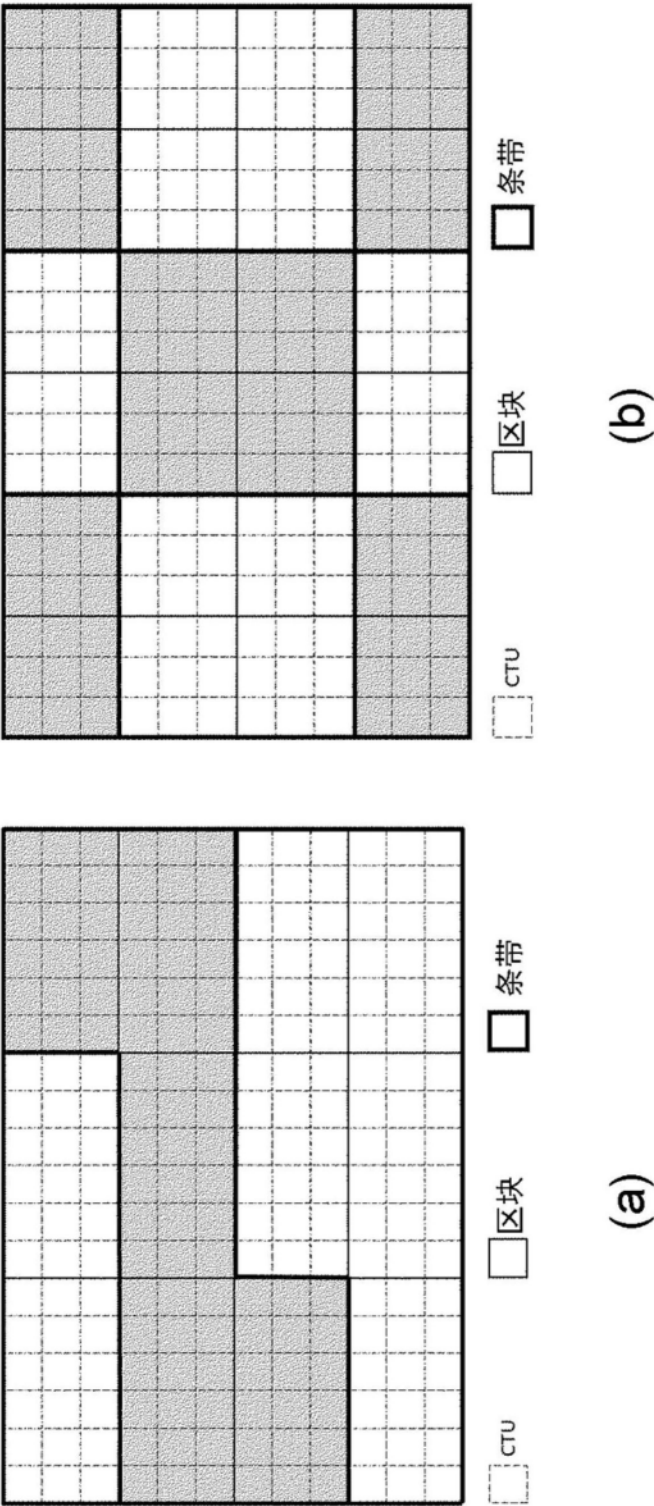


图10

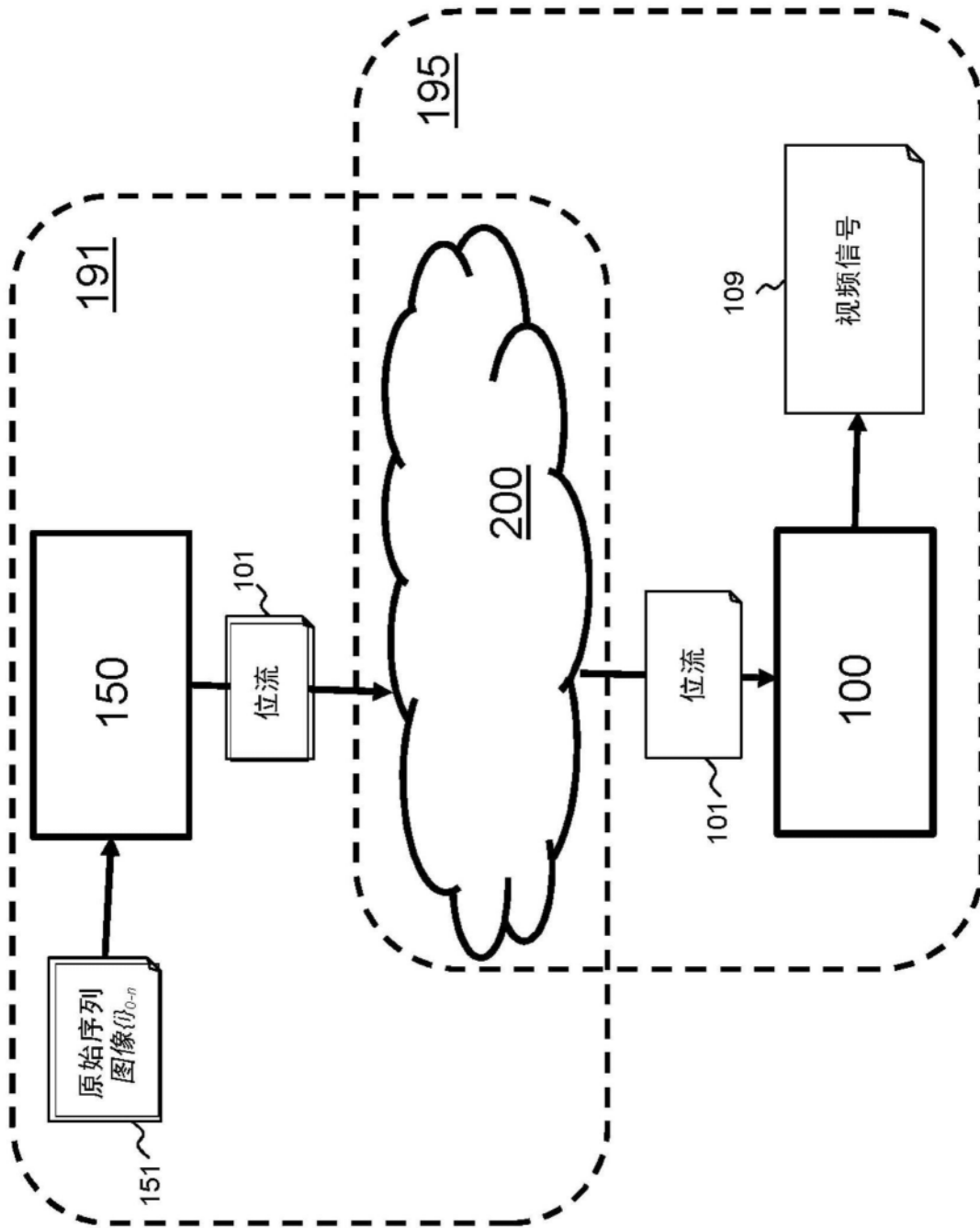


图11

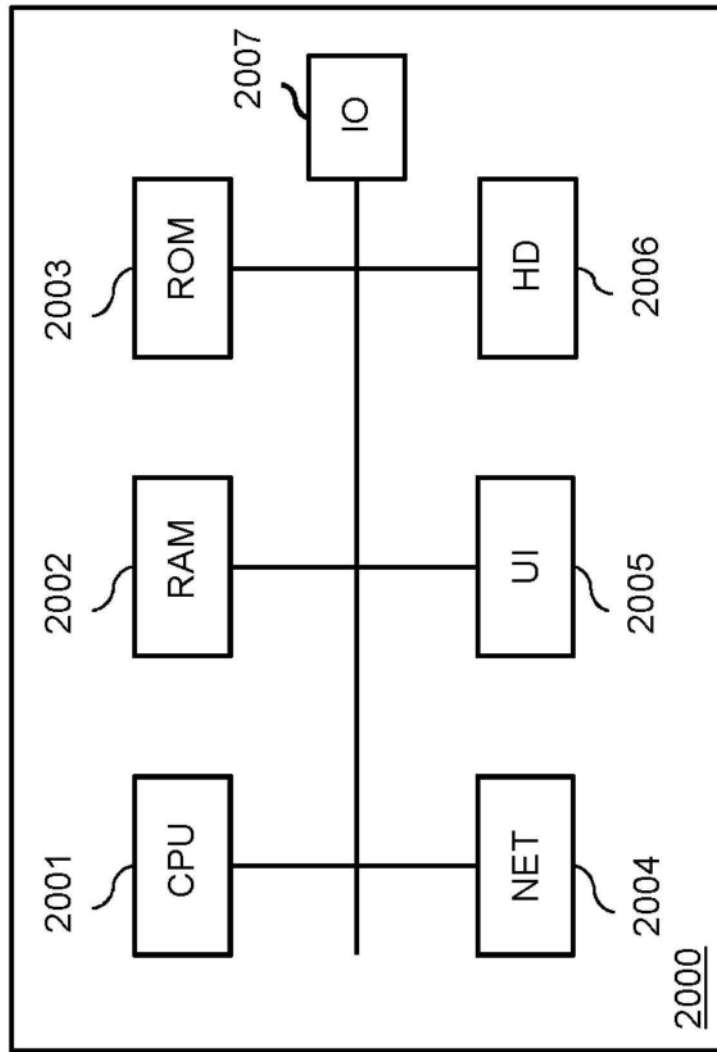


图12

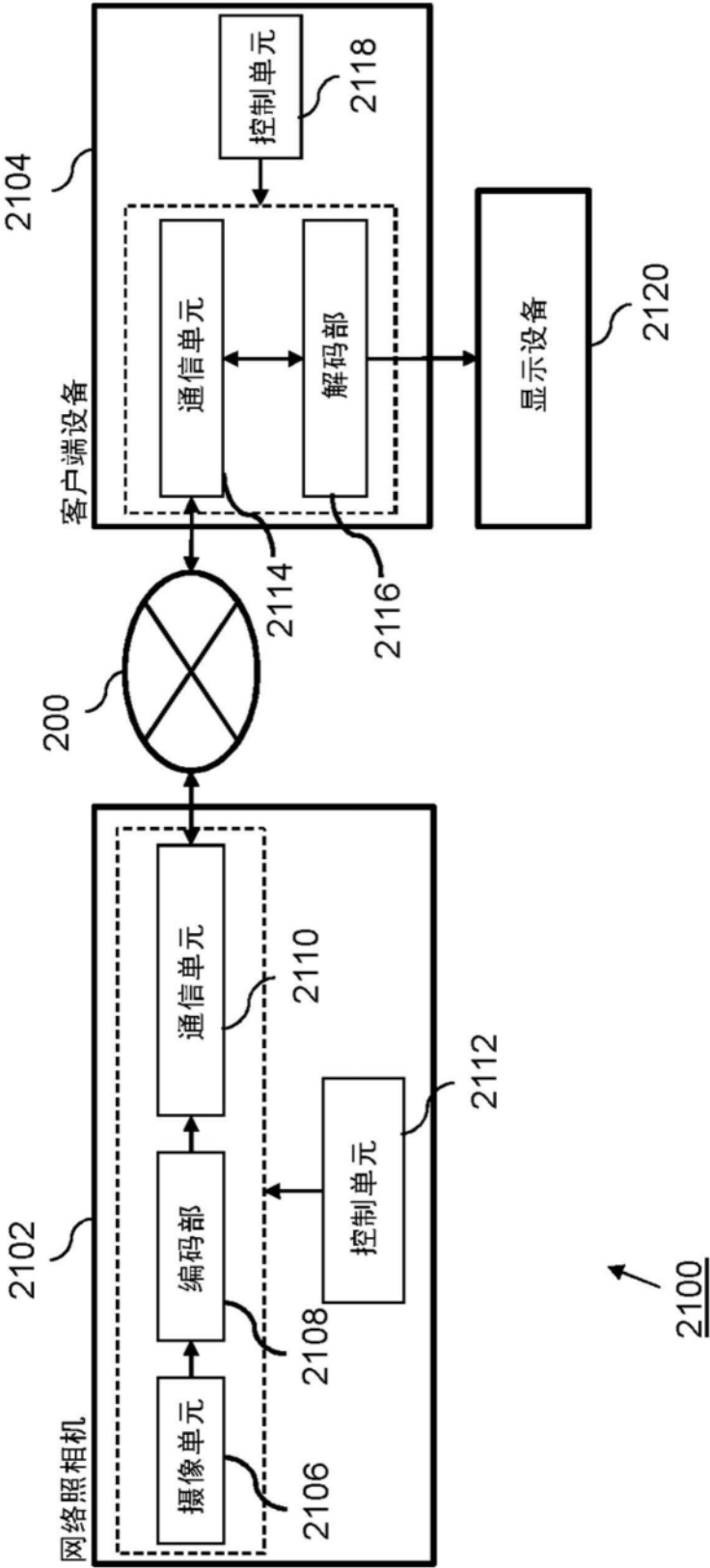


图13

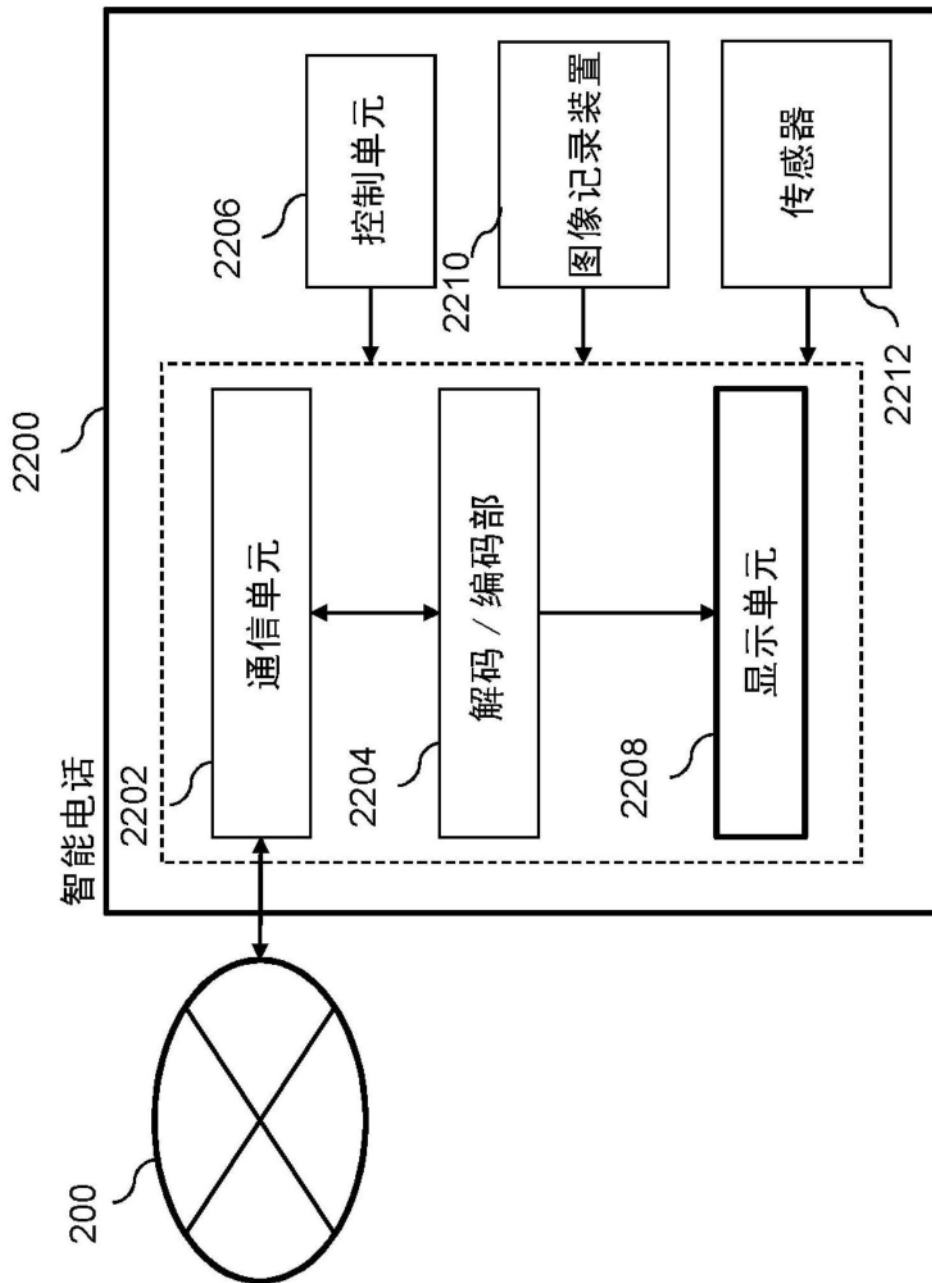


图14