



(21) 申请号 201680057989.7
(22) 申请日 2016.10.07
(65) 同一申请的已公布的文献号
申请公布号 CN 108140390 A
(43) 申请公布日 2018.06.08
(30) 优先权数据
15306591.7 2015.10.08 EP
62/361,863 2016.07.13 US
(85) PCT国际申请进入国家阶段日
2018.04.04
(86) PCT国际申请的申请数据
PCT/EP2016/073971 2016.10.07
(87) PCT国际申请的公布数据
W02017/060412 EN 2017.04.13

(73) 专利权人 杜比国际公司
地址 荷兰阿姆斯特丹
(72) 发明人 S·科顿 A·克鲁格
(74) 专利代理机构 中国贸促会专利商标事务所
有限公司 11038
专利代理师 宿小猛
(51) Int.Cl.
G10L 19/008 (2006.01)
审查员 陈凭

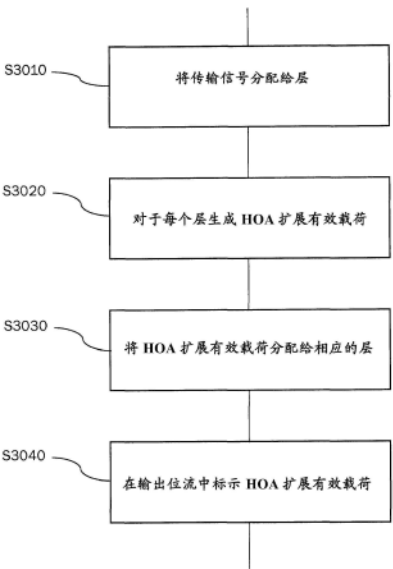
权利要求书2页 说明书41页 附图7页

(54) 发明名称

用于压缩高阶高保真度立体声响复制声音
或声场表示的分层编码和数据结构

(57) 摘要

本文档涉及一种声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧的分层编码方法。压缩HOA表示包括多个传输信号。该方法包括:将多个传输信号分配给多个分级层,多个分级层包括基本层和一个或多个分级增强层;对于每个层生成相应的HOA扩展有效载荷,该HOA扩展有效载荷包括用于参数化地增强从分配给相应层和低于相应层的任何层的传输信号可获得的重构的HOA表示的辅助信息,将生成的HOA扩展有效载荷分配给它们的相应的层,并且在输出位流中标示生成的HOA扩展有效载荷。本文档进一步涉及一种对声音或声场的压缩HOA表示的帧进行解码的方法、用于压缩HOA表示的分层编码的编码器和解码器、以及表示声音或声场的压缩HOA表示的帧的数据结构。



1. 一种对声音或声场的压缩高阶高保真度立体声响复制HOA表示进行解码的方法,该方法包括:

接收包含与多个分级层相对应的压缩HOA表示的位流,所述多个分级层包括基本层和一个或多个分级增强层,其中,多个层包括压缩HOA表示的分量,并且其中,对于每个层,所述位流包括相应的HOA扩展有效载荷,该HOA扩展有效载荷包括用于参数化地增强能够由分配给相应层和低于该相应层的任何层的传输信号获得的重构的HOA表示的辅助信息;

在所述多个层中确定用于解码的最高可用层,其中所述最高可用层是低于尚未被有效地接收到的最低层的层;

提取分配给最高可用层的HOA扩展有效载荷,其中,HOA扩展有效载荷包括用于参数化地增强与最高可用层相对应的重构的HOA表示的辅助信息,其中,与最高可用层相对应的重构的HOA表示能够基于分配给最高可用层和低于最高可用层的任何层的传输信号而获得;

基于层信息、分配给最高可用层和低于最高可用层的任何层的传输信号来对与最高可用层相对应的压缩HOA表示进行解码;以及

使用分配给最高可用层的HOA扩展有效载荷中所包括的辅助信息来参数化地增强解码的HOA表示。

2. 根据权利要求1所述的方法,其中,HOA扩展有效载荷包括用于HOA空间信号预测解码工具的位流元素。

3. 根据权利要求1所述的方法,其中,层信息包括增强信息,增强信息包括以下中的至少一个:空间信号预测、子带方向信号合成和参数化环境复制解码器。

4. 根据权利要求1所述的方法,进一步包括v矢量元素,对于与ContAddHoaCoeff集合中所包括的附加的HOA系数的索引相等的索引,不发送该v矢量元素。

5. 一种用于对声音或声场的压缩高阶高保真度立体声响复制HOA表示进行解码的装置,该装置包括:

接收器,该接收器被配置为接收包含与多个分级层相对应的压缩HOA表示的位流,所述多个分级层包括基本层和一个或多个分级增强层,其中,多个层包括压缩HOA表示的分量,该分量在相应的分量组中被分配给相应层,并且其中,对于每个层,所述位流包括相应的HOA扩展有效载荷,该HOA扩展有效载荷包括用于参数化地增强能够由分配给相应层和低于该相应层的任何层的传输信号获得的重构的HOA表示的辅助信息;

解码器,该解码器被配置为:

在所述多个层中确定用于解码的最高可用层,其中所述最高可用层是低于尚未被有效地接收到的最低层的层;

提取分配给该最高可用层的HOA扩展有效载荷,其中,

HOA扩展有效载荷包括用于参数化地增强与最高可用层相对应的重构的HOA表示的辅助信息,其中,与最高可用层相对应的重构的HOA表示能够基于分配给最高可用层和低于最高可用层的任何层的传输信号获得;

基于层信息、分配给最高可用层和低于最高可用层的任何层的传输信号来对与最高可用层相对应的压缩HOA表示进行解码;以及

使用分配给最高可用层的HOA扩展有效载荷中所包括的辅助信息来参数化地增强解码的HOA表示。

6. 根据权利要求5所述的装置,其中,H0A扩展有效载荷包括用于H0A空间信号预测解码工具的位流元素。

7. 根据权利要求5所述的装置,其中,层信息指示增强层的当前帧中的作用的方向信号的数量。

8. 根据权利要求5所述的装置,其中,层信息指示用于增强层的附加的环境H0A系数的总数量。

9. 根据权利要求5所述的装置,其中,层信息包括用于增强层的每个附加的环境H0A系数的H0A系数索引。

10. 根据权利要求5所述的装置,其中,层信息包括增强信息,增强信息包括以下中的至少一个:空间信号预测、子带方向信号合成和参数化环境复制解码器。

11. 根据权利要求5所述的装置,其中,如果在H0ADecoderConfig()中标示了等于1的CodedVVecLength,则使得压缩H0A表示适于用于基于H0A的内容的分层编码模式。

12. 根据权利要求5所述的装置,进一步包括v矢量元素,对于与ContAddHoaCoeff集合中所包括的附加的H0A系数的索引相等的索引,不发送该v矢量元素。

13. 根据权利要求12所述的装置,其中,该ContAddHoaCoeff集合是对于多个分级层中的每个单独地定义的。

14. 根据权利要求5所述的装置,其中,层信息包括NumLayers元素,其中,每个元素指示一直到第i层的所有层中所包括的传输信号的数量。

15. 根据权利要求5所述的装置,其中,层信息包括用于第k帧的所有实际使用的层的指示符。

16. 根据权利要求5所述的装置,其中,层信息指示:用于主导矢量的所有系数被指定。

17. 根据权利要求5所述的装置,其中,层信息指示:主导矢量的与大于MinNumOfCoeffsForAmbH0A的数值相对应的系数被指定。

18. 根据权利要求5所述的装置,其中,层信息指示:MinNumOfCoeffsForAmbH0A和在ContAddHoaCoeff[lay]中定义的所有元素不被发送,其中,lay是包含与矢量相对应的基于矢量的信号的层的索引。

19. 一种非暂态计算机可读介质,其上存储有指令,所述指令在被执行时使得执行根据权利要求1-4中任一项所述的方法。

20. 一种用于对声音或声场的压缩高阶高保真度立体声响复制H0A表示进行解码的设备,包括:

一个或多个处理器;以及

存储器,存储有指令,所述指令在被执行时使得所述一个或多个处理器执行根据权利要求1-4中任一项所述的方法。

21. 一种包括用于执行根据权利要求1-4中任一项所述的方法的部件的装置。

用于压缩高阶高保真度立体声响复制声音或声场表示的分层 编码和数据结构

[0001] 相关申请的交叉引用

[0002] 本申请要求2015年10月15日提交的欧洲专利申请No.15306653.5的优先权,该申请全文通过引用并入本文。

技术领域

[0003] 本文档涉及用于分层音频编码的方法和装置。具体地说,本文档涉及用于压缩高阶高保真度立体声响复制(HOA)声音(或声场)表示的帧的分层音频编码的方法和装置。本文档进一步涉及用于表示压缩HOA声音(或声场)表示的帧的数据结构(例如,位流)。

背景技术

[0004] 在HOA分层编码的目前定义中,创建用于HOA解码工具空间信号预测、子带方向信号合成和参数化环境复制(PAR)解码器的辅助信息(side information)来增强特定的HOA表示。也就是,在分层HOA编码的目前定义中,所提供的数据仅适当地扩展最高层(例如,最高增强层)的HOA表示。对于包括基本层的较低层,这些工具没有适当地增强部分重构的HOA表示。

[0005] 工具“子带方向信号合成和参数化环境复制解码器”是专门为低数据速率设计的,在低数据速率中,只有几个传输信号是可用的。但是,在HOA分层编码中,(部分)重构的HOA表示的适当增强尤其对于低位速率层(诸如基本层)是不可能的。从低位速率时的声音质量的观点来看,很清楚这是不期望的。

[0006] 另外,已经发现,如果在HOADecoderConfig()中标示(signal)了等于1的CodedVVecLength(即,如果矢量编码模式是作用的(active)),则处理基于矢量的信号的编码的V矢量元素的常规方式不导致适当的解码。在该矢量编码模式下,对于ContAddHoaCoeff集合中包括的HOA系数索引,不发送V矢量元素。该集合包括所有的HOA系数索引AmbCoeffIdx[i],该HOA系数索引具有等于零的AmbCoeffTransitionState。按照常规,不需要还添加加权的V矢量信号,因为用于这些索引的原始的HOA系数序列被显式地发送(被标示)。因此,对于这些索引,V矢量元素被设置为零。

[0007] 但是,在分层编码模式下,连续HOA系数索引集合依赖于传输信道,这些传输信道是当前作用的层的一部分。在较高层中发送的附加的HOA系数索引在较低层中可能缺失。那么,矢量信号不应该对HOA系数序列做出贡献的假定对于属于被包括在较高层中的HOA系数序列的HOA系数索引是错误的。

[0008] 因此,分层HOA编码中的V矢量可能不适合于最高层下面的任何层的解码。

[0009] 因此,需要适于声音或声场的压缩HOA表示的分层编码的编码方案和位流。

[0010] 本文档解决以上问题。具体地说,描述了用于压缩HOA声音或声场表示的帧的分层编码的方法和编码器/解码器以及用于表示压缩HOA声音或声场表示的帧的数据结构。

发明内容

[0011] 根据一方面,描述了一种声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧的分层编码方法。压缩HOA表示符合草案MPEG-H 3D音频标准和任何其他未来采用的或草案标准。压缩HOA表示可以包括多个传输信号。传输信号可以与单耳(monaural)信号有关,单耳信号例如表示HOA表示的系数序列或主导声音信号。该方法可以包括将多个传输信号分配给多个分级层。例如,传输信号可以被分布到多个层。该多个层可以包括基本层和一个或多个分级增强层。多个分级层可以按从基本层至第一增强层、第二增强层等、一直到总体最高增强层(总体最高层)的次序排序。该方法可以进一步包括对每个层生成相应的HOA扩展有效载荷,该HOA扩展有效载荷包括辅助信息(例如,增强辅助信息),该辅助信息用于参数化地增强从分配给相应层和低于该相应层的任何层的传输信号可获得的重构的HOA表示。用于较低层的重构的HOA表示可以被称为部分重构的HOA表示。该方法可以进一步包括将生成的HOA扩展有效载荷分配给它们的相应层。该方法还可以进一步包括在输出位流中标示生成的HOA扩展有效载荷。可以在HOAEnhFrame()有效载荷中标示HOA扩展有效载荷。因此,可以将辅助信息从HOAFrame()移动到HOAEnhFrame()。

[0012] 如上面所配置的,所提出的方法对压缩HOA表示(的帧)应用分层编码以便使得即使在低位速率时也能够对其进行高质量解码。具体地说,所提出的方法确保每个层包括合适的HOA扩展有效载荷(例如,增强辅助信息),该HOA扩展有效载荷用于增强从一直到当前层的任何层中的传输信号获得的(部分)重构的声音表示。其中,一直到当前层的层被理解为包括例如基本层、第一增强层、第二增强层等、一直到当前层。其中,一直到当前层的层被理解为包括例如基本层、第一增强层、第二增强层等、一直到当前层。例如,将启用解码器来参考分配给基本层的HOA扩展有效载荷增强从基本层获得的(部分)重构的声音表示。在常规的方法中,只有最高增强层的重构的HOA表示才可以通过HOA扩展有效载荷被增强。因此,不管实际的最高可用层如何(例如,是低于尚未被有效地接收到的最低层的层,使得低于最高可用层的所有层和最高可用层本身已经被有效地接收到),即使(部分)重构的声音表示可能不同于完整的(例如,整个的)声音表示,都将启用解码器来改进或增强重构的声音表示。具体地说,不管实际的最高可用层如何,对于解码器来说只对单个层(即,对最高可用层)解码HOA扩展有效载荷以改进或增强(部分)重构的声音表示就足够了,该(部分)重构的声音表示可以基于被包括在一直到实际的最高可用层的层中的所有的传输信号而获得。解码较高或较低层的HOA扩展有效载荷不是必需的。另一方面,所提出的方法允许充分地利用当应用分层编码时可以实现的所需带宽的减小。

[0013] 在实施例中,该方法可以进一步包括发送用于多个层的具有相应的错误保护水平的数据有效载荷。数据有效载荷可以包括相应的HOA扩展有效载荷。基本层可以具有最高的错误保护,一个或多个增强层可以具有依次降低的错误保护。从而,可以确保至少数个较低层被可靠地发送,同时另一方面通过对较高层不应用过大的错误保护来减小总体所需带宽。

[0014] 在实施例中,HOA扩展有效载荷可以包括用于HOA空间信号预测解码工具的位流元素。另外地或可替代地,HOA扩展有效载荷可以包括用于HOA子带方向信号合成解码工具的位流元素。另外地或可替代地,HOA扩展有效载荷可以包括用于HOA参数化环境复制解码工具的位流元素。

[0015] 在实施例中,HOA扩展有效载荷可以具有为ID_EXT_ELE_HOA_ENH_LAYER的usacExtElementType。

[0016] 在实施例中,该方法可以进一步包括生成HOA配置扩展有效载荷,该HOA配置扩展有效载荷包括用于配置HOA空间信号预测解码工具、HOA子带方向信号合成解码工具和/或HOA参数化环境复制解码工具的位流元素。HOA配置扩展有效载荷可以被包括在HOADecoderEnhConfig()中。该方法可以进一步包括在输出的位流中标示HOA配置扩展有效载荷。

[0017] 在实施例中,该方法可以进一步包括生成HOA解码器配置有效载荷,该HOA解码器配置有效载荷包括指示HOA扩展有效载荷对于多个层的分配的信息。该方法可以进一步包括在输出位流中标示HOA解码器配置有效载荷。

[0018] 在实施例中,该方法可以进一步包括确定矢量编码模式是否作用。该方法可以进一步包括如果矢量编码模式作用,则对于每个层基于分配给相应层的传输信号来确定连续HOA系数索引集合。连续HOA系数索引集合中的HOA系数索引可以是被包括在集合ContAddHOACoeff中的HOA系数索引。该方法可以进一步包括对于每个传输信号基于对被分配了相应的传输信号的层确定的连续HOA系数索引集合来生成V矢量,以使得生成的V矢量包括用于分配给比被分配了相应的传输信号的层高的层的任何传输信号的元素。该方法可以进一步包括在输出位流中标示生成的V矢量。

[0019] 根据另一方面,描述了一种声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧的分层编码方法。压缩HOA表示可以包括多个传输信号。传输信号可以与单耳信号有关,单耳信号例如表示HOA表示的系数序列或主导声音信号。该方法可以包括将多个传输信号分配给多个分级层。例如,传输信号可以被分布到多个层。该多个层可以包括基本层和一个或多个分级增强层。该方法可以进一步包括确定矢量编码模式是否作用。该方法可以进一步包括如果矢量编码模式作用,则对于每个层基于分配给相应层的传输信号来确定连续HOA系数索引集合。连续HOA系数索引集合中的HOA系数索引可以是被包括在集合ContAddHOACoeff中的HOA系数索引。该方法可以进一步包括对于每个传输信号基于对被分配了相应的传输信号的层确定的连续HOA系数索引集合来生成V矢量,以使得生成的V矢量包括用于分配给比被分配了相应的传输信号的层高的层的任何传输信号的元素。该方法可以进一步包括在输出位流中标示生成的V矢量。

[0020] 照此配置,所提出的方法确保在矢量编码模式下,合适的V矢量对属于一直到最高可用层的层的每一个传输信号是可用的。具体地说,所提出的方法排除了与较高层中的传输信号相对应的V矢量的元素没有被显式地标示的情况。因此,一直到最高可用层的层中所包括的信息对于对属于一直到最高可用层的层的任何传输信号进行解码是足够的。从而,即使较高层可能还未被解码器有效地接收,用于较低层(低位速率层)的相应的重构的HOA表示也被适当地解压缩。另一方面,所提出的方法允许充分地利用当应用分层编码时可以实现的所需带宽的减小。

[0021] 根据另一方面,描述了一种声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧的解码方法。压缩HOA表示可以被编码在多个分级层中。该多个分级层可以包括基本层和一个或多个分级增强层。该方法可以包括接收与压缩HOA表示的帧相关的位流。该方法可以进一步包括提取用于该多个层的有效载荷。每个有效载荷可以包括分配给相应层的

传输信号。该方法可以进一步包括在该多个层中确定用于解码的最高可用层。该方法可以进一步包括提取分配给最高可用层的HOA扩展有效载荷。该HOA扩展有效载荷可以包括用于参数化地增强与最高可用层相对应的(部分)重构的HOA表示的辅助信息。与最高可用层相对应的(部分)重构的HOA表示可以基于分配给最高可用层和低于最高可用层的任何层的传输信号获得。该方法可以进一步包括基于分配给最高可用层和低于最高可用层的任何层的传输信号来生成与最高可用层相对应的(部分)重构的HOA表示。该方法还可以进一步包括使用分配给最高可用层的HOA扩展有效载荷中所包括的辅助信息来增强(例如,参数化地增强)(部分)重构的HOA表示。结果,可以获得增强的重构的HOA表示。

[0022] 照此配置,通过尽最大可能程度地使用可用的(例如,有效地接收的)信息,所提出的方法确保最终的(例如,增强的)重构的HOA表示具有最佳的质量。

[0023] 在实施例中,HOA扩展有效载荷可以包括用于HOA空间信号预测解码工具的位流元素。另外地或可替代地,HOA扩展有效载荷可以包括用于HOA子带方向信号合成解码工具的位流元素。另外地或可替代地,HOA扩展有效载荷可以包括用于HOA参数化环境复制解码工具的位流元素。

[0024] 在实施例中,HOA扩展有效载荷可以具有为ID_EXT_ELE_HOA_ENH_LAYER的usacExtElementType。

[0025] 在实施例中,该方法可以进一步包括通过对位流进行解析来提取HOA配置扩展有效载荷。HOA配置扩展有效载荷可以包括用于配置HOA空间信号预测解码工具、HOA子带方向信号合成解码工具和/或HOA参数化环境复制解码工具的位流元素。

[0026] 在实施例中,该方法可以进一步包括提取分别分配给多个层的HOA扩展有效载荷。每个HOA扩展有效载荷可以包括用于参数化地增强与其相应的被分配的层相对应的(部分)重构的HOA表示的辅助信息。与其相应的被分配的层相对应的(部分)重构的HOA表示可以从分配给该层和低于该层的任何层的传输信号获得。HOA扩展有效载荷对于相应层的分配可以从位流中所包括的配置信息知道。

[0027] 在实施例中,确定最高可用层可以涉及确定指示尚未被有效地接收的层的无效层索引集合。它可以进一步涉及将最高可用层确定为是低于由无效层索引集合中的最小(最低)索引指示的层的一层。基本层可以具有最低层索引(例如,层索引1),分级增强层可以具有依次提高的层索引。从而,所提出的方法确保最高可用层是以对来自最高可用层和低于最高可用层的任何层的(部分)重构的HOA表示进行解码所需的所有信息都可用的方式来选择。

[0028] 在实施例中,确定最高可用层可以涉及确定指示尚未被有效地接收的层的无效层索引集合。它可以进一步涉及确定当前帧前面的先前(previous)帧的最高可用层。它还可以进一步涉及将最高可用层确定为以下层中的较低的一个:先前帧的最高可用层、以及作为低于由无效层索引集合中的最小索引指示的层的一层的层。从而,即使当前帧已经被相对于前一(preceding)帧不同地编码,用于当前帧的最高可用层也是以对来自最高可用层和低于最高可用层的任何层的(部分)重构的HOA表示进行解码所需的所有信息都可用的方式来选择。

[0029] 在实施例中,该方法可以进一步包括:如果当前帧的最高可用层低于先前帧的最高可用层,并且如果当前帧已经被相对于先前帧不同地编码,则决定不使用分配给最高可

用层的HOA扩展有效载荷中所包括的辅助信息来执行(部分)重构的HOA表示的参数化增强。从而,在当前帧(包括分配给最高可用层的HOA扩展有效载荷中所包含的辅助信息)已经被相对于前一帧不同地编码的情况下,可以对重构的HOA表示进行无错误地解码。

[0030] 在实施例中,可以通过评估对应的HOA扩展有效载荷的有效性标志来确定无效层索引集合。如果用于分配给相应层的HOA扩展有效载荷的有效性标志没有被设置,则可以将给定层的层索引添加到无效层索引集合。从而,可以以高效的方式确定无效层索引集合。

[0031] 根据另一方面,描述了一种表示声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧的数据结构(例如,位流)。压缩HOA表示可以包括多个传输信号。数据结构可以包括与多个分级层中的相应的分级层相对应的多个HOA帧有效载荷。HOA帧有效载荷可以包括相应的传输信号。多个传输信号可以被分配给(例如,被分布到)多个层。该多个层可以包括基本层和一个或多个分级增强层。数据结构可以进一步包括用于每个层的相应的HOA扩展有效载荷,该HOA扩展有效载荷包括用于参数化地增强从分配给相应层和低于相应层的任何层的传输信号可获得的(部分)重构的HOA表示的辅助信息。

[0032] 在实施例中,用于多个层的HOA帧有效载荷和HOA扩展有效载荷可以设有相应的错误保护水平。基本层可以具有最高的错误保护,一个或多个增强层可以具有依次降低的错误保护。

[0033] 在实施例中,HOA扩展有效载荷可以包括用于HOA空间信号预测解码工具的位流元素。另外地或可替代地,HOA扩展有效载荷可以包括用于HOA子带方向信号合成解码工具的位流元素。另外地或可替代地,HOA扩展有效载荷可以包括用于HOA参数化环境复制解码工具的位流元素。

[0034] 在实施例中,HOA扩展有效载荷可以具有为ID_EXT_ELE_HOA_ENH_LAYER的usacExtElementType。

[0035] 在实施例中,数据结构可以进一步包括HOA配置扩展有效载荷,该HOA配置扩展有效载荷包括用于配置HOA空间信号预测解码工具、HOA子带方向信号合成解码工具和/或HOA参数化环境复制解码工具的位流元素。

[0036] 在实施例中,数据结构可以进一步包括HOA解码器配置有效载荷,该HOA解码器配置有效载荷包括指示HOA扩展有效载荷对于多个层的分配的信息。

[0037] 在实施例中,方法和装置涉及对声音或声场的压缩高阶高保真度立体声响复制(HOA)表示进行解码。该装置可以被配置为或者该方法可以包括:接收位流,该位流包含与多个分级层相对应的压缩HOA表示,该多个分级层包括基本层和一个或多个分级增强层,其中,多个层被分配有声音或声场的基本压缩声音表示的分量,这些分量在相应的分量组中被分配给相应层,在该多个层中确定用于解码的最高可用层;提取分配给最高可用层的HOA扩展有效载荷,其中,HOA扩展有效载荷包括用于参数化地增强与最高可用层相对应的重构的HOA表示的辅助信息,其中,与最高可用层相对应的重构的HOA表示可以基于分配给最高可用层和低于最高可用层的任何层的传输信号获得;基于层信息、分配给最高可用层和低于最高可用层的任何层的传输信号来对与最高可用层相对应的压缩HOA表示进行解码;并且使用分配给最高可用层的HOA扩展有效载荷中所包括的辅助信息来参数化地增强解码的HOA表示。

[0038] HOA扩展有效载荷可以包括用于HOA空间信号预测解码工具的位流元素。层信息可

以指示增强层的当前帧中的作用方向信号的数量。

[0039] 层信息可以指示用于增强层的附加环境HOA系数的总数量。层信息可以包括用于增强层的每个附加环境HOA系数的HOA系数索引。层信息可以包括增强信息,该增强信息包括空间信号预测、子带方向信号合成和参数化环境复制解码器中的至少一个。如果在HOADecoderConfig()中标示了等于1的CodedVVecLength,则使得压缩HOA表示适于基于HOA的内容的分层编码模式。此外,对于与ContAddHoaCoeff集合中所包括的附加HOA系数的索引相等的索引,可以不发送v矢量元素。可以对多个分级层中的每个单独地定义ContAddHoaCoeff集合。层信息包括NumLayers元素,其中,每个元素指示一直到第i层的所有层中包括的传输信号的数量。层信息可以包括对第k帧的所有实际使用的层的指示符。层信息还可以指示用于主导矢量的所有系数被指定。层信息可以指示与大于MinNumOfCoeffsForAmbHOA的数值相对应的主导矢量的系数被指定。层信息可以指示MinNumOfCoeffsForAmbHOA和ContAddHoaCoeff[lay]中定义的所有元素不被发送,其中,lay是包含与矢量相对应的基于矢量的信号的层的索引。

[0040] 根据另一方面,描述了一种用于声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧的分层编码的编码器。压缩HOA表示可以包括多个传输信号。该编码器可以包括处理器,该处理器被配置为执行根据第一上述方面和第二上述方面的方法的方法步骤中的一些或全部。

[0041] 根据另一方面,描述了一种用于对声音或声场的压缩高阶高保真度立体声响复制(HOA)表示的帧进行解码的解码器。压缩HOA表示可以被编码在多个分级层中,该多个分级层包括基本层和一个或多个分级增强层。该解码器可以包括处理器,该处理器被配置为执行根据第三上述方面的方法的方法步骤中的一些或全部。

[0042] 根据另一方面,描述了一种软件程序。该软件程序可以适于在处理器上执行并且适于当在计算设备上被执行时执行本文档中概述的方法步骤中的一些或全部。

[0043] 根据又一方面,描述了一种存储介质。该存储介质可以包括软件程序,该软件程序适于在处理器上执行并且适于当在计算设备上被执行时执行本文档中概述的方法步骤中的一些或全部。

[0044] 如技术人员将意识到的,要意识到关于上面任何方面或其实施例所做的陈述也适用于相应的其他方面或它们的实施例。由于简洁的原因,省略了对每一个方面或实施例重复这些陈述。

[0045] 应注意,如本文档所概述的包括其优选实施例的方法和装置可以独立地使用,或者与本文档所公开的其他方法和系统组合使用。此外,本文档中所概述的方法和装置的所有方面可以被任意地组合。具体地说,权利要求的特征可以以任意的方式相互组合。

[0046] 应进一步注意,方法步骤和装置特征可以以许多方式互换。具体地说,如技术人员将意识到的,所公开的方法的细节可以实现为适于执行该方法的步骤中的一些或全部的装置,反之亦然。

附图说明

[0047] 下面参照附图以示例性的方式对本发明进行说明,其中:

[0048] 图1是示意性地示出在编码器端对于基本层和M-1个增强层的有效载荷分配的框

图;

[0049] 图2是示意性地示出接收器和解压缩级的例子的框图;

[0050] 图3是示出根据本公开的实施例的压缩HOA表示的帧的分层编码方法的例子的流程图;

[0051] 图4是示出根据本公开的实施例的压缩HOA表示的帧的分层编码方法的另一例子的流程图;

[0052] 图5是示出根据本公开的实施例的压缩HOA表示的帧的解码方法的例子的流程图;

[0053] 图6是示意性地示出根据本公开的实施例的编码器的硬件实现的例子的框图;以及

[0054] 图7是示意性地示出根据本公开的实施例的解码器的硬件实现的例子的框图。

具体实施方式

[0055] 首先,将描述根据本公开的方法和编码器/解码器可以适用的压缩声音(或声场)表示。

[0056] 为了在具有时变条件的发送信道上流传输压缩声音(或声场)表示,分层编码是使接收的声音表示的质量适应发送条件、特别是避免无用信号衰落(dropout)的手段。

[0057] 为了分层编码,通常将压缩声音(或声场)表示细分为具有相对小的大小的高优先级基本层和具有减量优先级和任意大小的附加增强层。每个增强层通常被假定为包含补充所有较低层的信息的增量信息以便改进压缩声音(或声场)表示的质量。构思然后根据各个层的优先级来控制用于这些层的发送的错误保护的量。具体地说,基本层设有高错误保护,由于其大小低,因此这是合理的且负担得起的。

[0058] 在下面假定完整的压缩声音(或声场)表示一般包括以下三个分量:

[0059] 1.本身包括数个补充分量的基本压缩声音(或声场)表示,这些补充分量占完整的压缩声音(或声场)表示的区别性地最大的百分比。

[0060] 2.对基本压缩声音表示进行解码所需要的基本辅助信息,该基本辅助信息被假定为与基本压缩声音(或声场)表示相比大小小得多。它被进一步假定为其最大部分包含以下两个分量,这两个分量指定基本压缩声音表示的仅一个特定分量的解压缩:

[0061] a) 第一分量包含独立于其他补充分量描述基本压缩声音(或声场)表示的各个补充分量的辅助信息。

[0062] b) 第二(可选的)分量包含依赖于其他补充分量描述基本压缩声音(或声场)表示的各个补充分量的辅助信息。具体地说,依赖性具有以下性质:

[0063] ●用于基本压缩声音(或声场)表示的每个单个的补充分量的相关辅助信息在没有其他的某些补充分量被包含在基本压缩声音(或声场)表示中的情况下实现其最大程度。

[0064] ●在附加的某些补充分量被添加到基本压缩声音(或声场)表示的情况下,用于所考虑的单个的补充分量的相关辅助信息变为原始的相关辅助信息的子集,从而缩小其大小。

[0065] 3.改进基本压缩声音(或声场)表示的可选的增强辅助信息。其大小也被假定为比基本压缩声音(或声场)表示的大小小得多。

[0066] 这样的类型的完整的压缩声音(或声场)表示的一个突出的例子由如MPEG-H 3D音

频标准的初级版本指定的压缩HOA声场表示给出。

[0067] 1. 其基本压缩声场表示可以用数个量化的单耳信号来标识, 这些单耳信号表示所谓的主导声音信号或所谓的环境HOA声场分量的系数序列。

[0068] 2. 基本辅助信息除了别的之外还对于这些单耳信号中的每个描述它在空间上如何对声场做出贡献。该信息可以被进一步划分为以下两个不同的分量:

[0069] (a) 与特定的各个单耳信号相关的辅助信息, 该辅助信息是独立于其他单耳信号的存在。这样的辅助信息可以例如指定表示具有某个入射方向的方向信号(意指一般的平面波)的单耳信号。可替代地, 单耳信号可以被指定为具有某个索引的原始HOA表示的系数序列。

[0070] (b) 与特定的各个单耳信号相关的辅助信息, 该辅助信息是依赖于其他单耳信号的存在。例如, 如果单耳信号被指定为所谓的基于矢量的信号(这意味着它们按方向分布在声场内, 其中, 方向分布是借助于矢量指定的), 则这样的辅助信息出现。在某个模式(即, CodedVVecLength=1)下, 该矢量的特定分量被隐式地设置为零, 并且不是压缩矢量表示的一部分。这些分量是作为基本压缩声场表示的一部分的、具有与原始HOA表示的系数序列的索引相等的索引的那些分量。这意味着, 如果矢量的各个分量被编码, 则它们的总数量依赖于基本压缩声场表示, 特别是依赖于它包含原始HOA表示的哪些系数序列。

[0071] 如果原始HOA表示的系数序列都不包含在基本压缩声场表示中, 则用于每个基于矢量的信号的依赖基本辅助信息包括所有的矢量分量, 并且具有其最大大小。在具有某些索引的原始HOA表示的系数序列被添加到基本压缩声场表示的情况下, 从用于每个基于矢量的信号的辅助信息移除具有那些索引的矢量分量, 从而缩小用于基于矢量的信号的依赖基本辅助信息的大小。

[0072] 3. 增强辅助信息包括以下分量:

[0073] ●与用于从方向信号(线性地)预测声场的缺失部分的所谓的(宽带)空间预测相关的参数。

[0074] ●与所谓的子带方向信号合成和参数化环境复制相关的参数, 子带方向信号合成和参数化环境复制是允许对要被空间分布的附加的单耳信号进行频率相关的参数化预测以便补充目前空间上不完整的或有缺陷的压缩HOA表示的压缩工具。该预测基于基本压缩声场表示的系数序列。重要的方面是, 所提及的对于声场的补充贡献在压缩HOA表示内不是借助于附加的量化信号、而是借助于相比而言大小小得多的额外的辅助信息来表示的。因此, 所提及的两个编码工具尤其适合于低数据速率时的HOA表示的压缩。

[0075] 具有上述结构的单耳信号的压缩表示的第二个例子可以包括以下分量:

[0076] 1. 用于一直到某个上频率的不相交(disjoint)的频带的一些编码的频谱信息, 这可以被认为是基本压缩表示。

[0077] 2. (按例如编码的频带的数量和宽度)指定编码的频谱信息的一些基本辅助信息。

[0078] 3. 包括所谓的频带复制(SBR)的参数的一些增强辅助信息, 其描述如何根据基本压缩表示来参数化地重构在基本压缩表示中没有被考虑的较高频带的频谱信息。

[0079] 接着, 将描述具有上述结构的完整的压缩声音(或声场)表示的分层编码方法。

[0080] 从压缩提供在连续的时间间隔(例如相等大小的时间间隔)内的压缩表示(例如, 以数据包(packet)或等同地帧有效载荷的形式)的意义上讲, 假定压缩是基于帧的。这些

数据包被假定为包含有效性标志(指示它们的大小的值)以及实际的压缩表示数据。在以下整个描述中,将主要集中于单个帧的处理,因此帧索引将被省略。

[0081] 所考虑的完整的压缩声音(或声场)表示1100的每个帧有效载荷被假定为包含J个数据包,每个数据包用于基本压缩声音(或声场)表示的一个分量1110-1, ..., 1110-J,这些分量用 $BSRC_j$, $j=1, \dots, J$ 来表示。此外,它被假定为包含用 BSI_I 表示的具有独立基本辅助信息1120的包,该包独立于其他分量指定基本压缩声音表示的特定分量 $BSRC_j$ 。可选地,它另外还被假定为包含用 BSI_D 表示的具有依赖基本辅助信息的包,该包依赖于其他分量指定基本压缩声音表示的特定分量 $BSRC_j$ 。两个数据包 BSI_I 和 BSI_D 中包含的信息可以可选地被分组到一个单个数据包BSI中。

[0082] 最后,它包括用ESI表示的增强辅助信息有效载荷,该增强辅助信息有效载荷具有如何改进从完整的基本压缩表示重构的声音(或声场)的描述。

[0083] 所描述的分层编码方案解决了使得能够既实现压缩部分(包括对数据包进行打包以用于发送)、又能够实现接收器和解压缩部分的所需步骤。下面将详细地描述每个部分。

[0084] 接着,将描述用于发送的压缩和打包。在分层编码(假定总共有M个层,即,一个基本层和M-1个增强层)的情况下,如下对完整的压缩声音(或声场)表示1100的每个分量进行处理:

[0085] ●将基本压缩声音(或声场)表示细分为要被分配给各个层的部分。不失一般性地,分组可以用M+1个数 J_m , $m=0, \dots, M$ (其中 $J_0=1$ 且 $J_M=J+1$)来描述,以使得对于 $J_{m-1} \leq j < J_m$, $BSRC_j$ 被分配给第m层。

[0086] ●由于其大小小,将完整的基本辅助信息分配给基本层以避免其不必要的碎片化是合理的。虽然独立基本辅助信息 BSI_I 被对于分配保持不变,但是依赖基本辅助信息需要被特别地处理以用于分层编码,以便一方面允许在接收器端进行正确的解码,另一方面缩小要被发送的相关辅助信息的大小。提出了将它分解为用 $BSI_{D,m}$, $m=1, \dots, M$ 表示的M个部分1130-1, ..., 1130-M,其中,第m部分包含用于分配给第m层的基本压缩声音表示的分量 $BSRC_j$, $J_{m-1} \leq j < J_m$ 中的每个的相关辅助信息,如果相应的相关辅助信息存在的话。在相应的相关辅助信息不存在的情况下, $BSI_{D,m}$ 被假定为是空的。辅助信息 $BSI_{D,m}$ 依赖于一直到第m层的所有层中包含的所有分量 $BSRC_j$, $1 \leq j < J_m$ 。

[0087] ●在分层编码的情况下,重要的是意识到需要对于每个层额外地计算增强辅助信息,因为它的意图是增强初步解压缩的声音(或声场),不过这依赖于可用于解压缩的层。因此,压缩需要提供用 ESI_m , $m=1, \dots, M$ 表示的M个单个的增强辅助信息数据包1140-1、.....、1140-M,其中,计算第m数据包 ESI_m 中的增强辅助信息,诸如以便增强从基本层和具有低于m的索引的增强层中包含的所有数据获得的声音(或声场)表示。

[0088] 总而言之,在压缩级,必须提供具有以下成分的用FRAME表示的帧数据包:

[0089] $FRAME = [BSRC_1 \dots BSRC_J, BSI_I, BSI_{D,1} \dots BSI_{D,M}, ESI_1 \dots ESI_M]$ (1)

[0090] 理解的是,具有帧数据包的各个有效载荷的排序一般是任意的。

[0091] 各个有效载荷对于基本层和增强层的已经描述的分配是通过所谓的传输层打包器来实现的,并且在图1中被示意性地示出。

[0092] 接着,将描述接收和解压缩。图2中示出了对应的接收器和解压缩级。

[0093] 首先,对各个数据包1200、1300-1、.....、1300-(M-1)进行复用以提供完整的压缩

声音(或声场)表示的接收的帧包:

[0094] $[BSI_1 BSI_{D,1} \cdots BSI_{D,M} ESI_1 BSRC_1 \cdots BSRC_{(J1)-1} \cdots ESI_M BSRC_{J(M-1)} \cdots BSRC_J] (2)$

[0095] 该帧包然后被传递到解压缩器2100。假定如果单个的层的发送已经是无错误的,则至少所包含的增强辅助信息有效载荷的有效性标志被设置为“真”。在由于单个的层的发送导致错误的情况下,至少该层中的增强辅助信息有效载荷内的有效性标志被设置为“假”。因此,可以从所包含的增强辅助信息有效载荷的有效性确定层包的有效性。

[0096] 在解压缩器2100中,首先对接收的帧包进行解复用。为了这个目的,可以利用关于每个有效载荷的大小的信息来避免不必要地解析各个有效载荷的数据。

[0097] 在下一步中,选择实际上要被用于基本声音表示的解压缩的最高层的编号 N_B 。实际上要被用于基本声音表示的解压缩的最高增强层由 N_B-1 给出。因为每个层包含正好一个增强辅助信息有效载荷,所以从每个增强辅助信息有效载荷知道包含层是否有效。因此,可以使用所有的增强辅助信息有效载荷 $ESI_m, m=1, \cdots, M$ 来实现选择。另外,确定要被用于解压缩的增强辅助信息有效载荷的索引 N_E ,该索引总是等于 N_B 或等于零。这意味着增强要么总是根据基本声音表示来实现的,要么根本不实现。下面进一步给出选择的较详细的描述。

[0098] 连续地,将基本压缩声音表示分量 $BSRC_1, \cdots, BSRC_J$ 的有效载荷与所有的基本辅助信息有效载荷(即, BSI_1 和 $BSI_{D,m}, m=1, \cdots, M$)和值 N_B 一起传递到基本表示解压缩处理单元2200,基本表示解压缩处理单元2200仅使用最低的 N_B 个层(即,基本层和 N_B-1 个增强层)内包含的那些基本压缩声音表示分量来重构基本声音(或声场)表示。关于基本压缩声音(或声场)表示的哪些分量被包含在各个层中的所需信息被假定为对于解压缩器2100来说是从具有配置信息的数据包知道的,该数据包被假定为是在帧数据包之前发送和接收的。每个单个的依赖基本辅助信息有效载荷 $BSI_{D,m}, m=1, \cdots, N_B$ 的实际解码可以如下分为两个部分:

[0099] 1.通过利用每个有效载荷 $BSI_{D,m}, m=1, \cdots, N_B$ 对头 m 个层中包含的头 J_m-1 个基本压缩声音表示分量 $BSRC_1, \cdots, BSRC_{(Jm)-1}$ 的依赖性来对该有效载荷进行初步解码,该依赖性是在编码级被假定的。

[0100] 2.通过考虑到基本声音分量最后是从头 $N_B > m$ 个层中包含的头 $J_{N_B}-1$ 个基本压缩声音表示分量 $BSRC_1, \cdots, BSRC_{(J_{N_B})-1}$ 重构的来对每个有效载荷 $BSI_{D,m}, m=1, \cdots, N_B$ 进行连续地校正,这些基本压缩声音表示分量是比对于初步解码采取的分量多的分量。因此,可以通过丢弃陈旧的信息来实现校正,由于依赖基本辅助信息的最初假定的性质(即,如果某些补充分量被添加到基本压缩声音(或声场)表示,则用于每个单个的补充分量的依赖基本辅助信息变为原始的依赖基本辅助信息的子集),这是可能的。

[0101] 最后,重构的基本声音(或声场)表示与所有的增强辅助信息有效载荷 ESI_1, \cdots, ESI_M 、基本辅助信息有效载荷 BSI_1 和 $BSI_{D,m}, m=1, \cdots, M$ 以及值 N_E 一起被提供给增强表示解压缩处理单元2300,增强表示解压缩处理单元2300通过仅使用增强辅助信息有效载荷 ESI_{N_E} 并且丢弃所有的其他的增强辅助信息有效载荷来计算最终的增强声音(或声场)表示。如果 N_E 的值等于零,则所有的增强辅助信息有效载荷都被丢弃,并且重构的最终的增强声音(或声场)表示等于重构的基本声音(或声场)表示。

[0102] 接着,将描述层选择。在所有的帧数据包可以被彼此独立地解压缩的情况下,实际上要被用于基本声音表示的解压缩的最高层的编号 N_B 和要被用于解压缩的增强辅助信息

有效载荷的索引 N_E 这二者被设置为有效的增强辅助信息有效载荷的最高编号 L ,该最高编号 L 本身可以通过评估增强辅助信息有效载荷内的有效性标志来确定。通过利用对每个增强辅助信息有效载荷的大小的知悉,可以避免为确定有效载荷的有效性而对它们的实际数据进行复杂的解析。

[0103] 在采用具有帧间依赖性的差分解压缩的情况下,需要另外考虑得自先前帧的决策。就差分解压缩而言,以规律的时间间隔发送独立的帧数据包以便允许从这些时刻开始解压缩,其中,值 N_B 和 N_E 的确定变为帧独立的,并且被如上所述那样执行。

[0104] 为了详细地说明帧依赖决策,我们首先对于第 k 帧

[0105] ●用 $L(k)$ 表示有效的增强辅助信息有效载荷的最高编号

[0106] ●用 $N_B(k)$ 表示要被选择并且要被用于基本声音表示的解压缩的最高层编号

[0107] ●用 $N_E(k)$ 表示要被用于解压缩的增强辅助信息有效载荷的编号

[0108] 通过使用该记号,根据以下方程来计算由 $N_B(k)$ 表示的要被用于基本声音表示的解压缩的最高层编号:

$$[0109] \quad N_B(k) = \min(N_B(k-1), L(k)) \quad (3)$$

[0110] 通过选择不大于 $N_B(k-1)$ 和 $L(k)$ 的 $N_B(k)$,确保基本声音表示的差分解压缩所需的所有信息是可用的。

[0111] 根据以下方程来确定要被用于解压缩的增强辅助信息有效载荷的编号 $N_E(k)$:

$$[0112] \quad N_E(k) = \begin{cases} N_B(k) & \text{如果 } N_B(k) = N_B(k-1) \\ 0 & \text{否则} \end{cases} \quad (4)$$

[0113] 这特别是意味着,只要要被用于基本声音表示的解压缩的最高层编号 $N_B(k)$ 没有改变,相同的对应的增强层编号就被选择。但是,在 $N_B(k)$ 改变的情况下,通过将 $N_E(k)$ 设置为零来禁用增强。由于对增强辅助信息采取差分解压缩,它根据 $N_B(k)$ 改变是不可能的,因为它将需要对先前帧处的对应的增强辅助信息层进行解压缩,而该解压缩被假定为尚未被执行。

[0114] 可替代地,如果在解压缩时具有一直到 $N_E(k)$ 的编号的所有的增强辅助信息有效载荷被并行地解压缩,则可以用以下方程来取代选择规则(4):

$$[0115] \quad N_E(k) = N_B(k) \quad (5)$$

[0116] 最后,要注意的是,对于差分解压缩,最高的被用层的编号只可以在独立的帧数据包处增大,而在每一个帧处减小是可能的。

[0117] 接着,将针对压缩HOA表示的情况来描述本公开的涉及压缩声音表示的帧的分层编码以及表示编码的压缩声音表示的帧的数据结构(例如,位流)的实施例。具体地说,将描述所提出的对于压缩HOA表示的分层编码方案的改变。

[0118] 作为用于基于HOA的内容的分层编码模式的校正,定义新的usacExtElementType以使HOA解码工具空间信号预测、子带方向信号合成和参数化环境复制(PAR)解码器的配置和帧有效载荷更好地适应对应的HOA增强层。如果用于基于HOA的内容的分层编码模式被激活(这用SingleLayer=0标示),则提出将这些工具的对应的位流元素移动到用于每个层(包括基本层和一个或多个增强层)的新类型的一个附加的HOA扩展有效载荷。

[0119] 因为用于这些工具的辅助信息是为增强特定的HOA表示而创建的,所以需要进行扩展。在分层HOA编码的目前定义中,所提供的数据仅适当地扩展了最高层的HOA表示。对于

较低层,这些工具没有适当地增强部分重构的HOA表示。

[0120] 因此,将较好的是对每个层提供这些工具的辅助信息以更好地使它们适应对应层的重构的HOA表示。

[0121] 另外,工具子带方向信号合成和参数化环境复制解码器是专门为低数据速率设计的,在低数据速率中,只有几个传输信号是可用的。所提出的扩展因此将提供使这些工具的辅助信息最佳地适应层中的传输信号的数量能力。因此,用于低位速率层(例如,基本层)的重构的HOA表示的声音质量与现有的分层方法相比可以被显著地提高。

[0122] 此外,如果在HOADecoderConfig()中标示了等于1的CodedVVecLength,则需要使得用于基于矢量的信号的编码的V矢量元素的位流语法适于HOA分层编码。在该矢量编码模式下,对于ContAddHoaCoeff集合中包括的HOA系数索引,不发送V矢量元素。该集合包括所有的HOA系数索引AmbCoeffIdx[i],这些HOA系数索引具有等于零的AmbCoeffTransitionState。因为用于这些索引的原始HOA系数序列被显式地发送,所以不需要还添加加权的V矢量信号。因此,对于这些索引,常规方法中的V矢量元素被设置为零。

[0123] 但是,在分层编码模式下,连续HOA系数索引集合依赖于传输信道,这些传输信道是当前作用的层的一部分。这意味着在较高层中发送的附加的HOA系数索引在较低层中是缺失的。那么,矢量信号不应该对HOA系数序列做出贡献的假定对于属于被包括在较高层中的HOA系数序列的HOA系数索引是错误的。因此,提出了对这些缺失的系数索引(显式地)标示V矢量元素。

[0124] 因此,提出了对每个层定义ContAddHoaCoeff集合并且使用其中添加了V矢量信号(V矢量信号的传输信号所属于)的层的集合来选择作用的V矢量元素。不过,提出了V矢量数据留在HOAFrame()中并且不被移动到HOAEnhFrame()。

[0125] 接着,将描述到MPEG-H位流语法中的集成。将参照图3来描述根据本公开的实施例的对应的编码方法(例如,声音或声场的压缩HOA表示的帧的分层编码方法)。下面将在附录中描述所提出的对于MPEG-H 3D位流的改变。

[0126] 在分层编码模式下,HOADecoderConfig()中的标志SingleLayer是不作用的(SingleLayer==0),并且定义层的数量和它们的被分配的HOA传输信号的对应数量。一般来说,压缩HOA表示可以包括多个传输信号。

[0127] 因此,在图3中的S3010,将多个传输信号分配给多个分级层。换句话说,传输信号被分布到多个层。每个层可以被说成是包括分配给该层的相应的传输信号。每个层可以具有分配给该层的多于一个的传输信号。多个层可以包括基本层和一个或多个分级增强层。这些层可以按从基本层、至增强层、一直到总体最高增强层(总体最高层)的次序排序。

[0128] 提出了将具有新定义的usacExtElementType ID_EXT_ELE_HOA_ENH_LAYER的附加的HOA配置扩展有效载荷和HOA帧扩展有效载荷添加到MPEG-H位流中以对每个HOA增强层(包括基本层)发送空间信号预测、子带方向信号合成和PAR解码器数据的一个有效载荷。这些额外的有效载荷将直接跟在mpegH3daExtElementConfig()中的、对应地在mpegH3daFrame()中的类型ID_EXT_ELE_HOA的有效载荷的后面。

[0129] 因此,提出了在SingleLayer==0的情况下,将用于空间信号预测、子带方向信号合成和PAR解码器的配置元素从HOADecoderConfig()移动到新定义的HOADecoderEnhConfig(),并且对应地将HOAPredictionInfo()、

HOADirectionalPredictionInfo() 和HOAParInfo() 从HOAFrame() 移动到新定义的HOAEnhFrame()。

[0130] 因此,在S3020,对每个层生成相应的HOA扩展有效载荷。所生成的HOA扩展有效载荷可以包括用于参数化地增强从分配给相应层和低于相应层的任何层的(例如,被包括在相应层和低于相应层的任何层中的)传输信号可获得的重构的HOA表示的辅助信息。如上面所指示的,HOA扩展有效载荷可以包括用于HOA空间信号预测解码工具、HOA子带方向信号合成解码工具和HOA参数化环境复制解码工具中的一个或多个的位流元素。此外,HOA扩展有效载荷可以具有为ID_EXT_ELE_HOA_ENH_LAYER的usacExtElementType。

[0131] 在S3030,将生成的HOA扩展有效载荷分配给它们的相应的层。

[0132] 此外(在图3中未示出),可以生成包括用于配置HOA空间信号预测解码工具、HOA子带方向信号合成解码工具和/或HOA参数化环境复制解码工具的位流元素的HOA配置扩展有效载荷。

[0133] 此外(在图3中未示出),可以生成HOA解码器配置有效载荷,该HOA解码器配置有效载荷包括指示HOA扩展有效载荷对于多个层的分配的信息。

[0134] 接着,将描述分层位流(例如,MPEG-H位流)的发送。因为MPEG-H位流的所有的扩展有效载荷是字节对齐的并且它们的大小是显式地標示的,所以如果假定elementLengthPresent标志等于1,则拆包器可以对MPEG-H位流进行解析,并且提取用于高于一(one)层的层的有效载荷,并且通过不同的发送信道分别发送它们。基本层包括排除用于较高层的数据的MPEG-H位流(例如,由排除用于较高层的数据的MPEG-H位流组成)。缺失的扩展有效载荷被标示为空的或不作用的。对于类型ID_USAC_SCE、ID_USAC_CPE和ID_USAC_LFE的有效载荷,空的有效载荷是用为零的elementLength标示的,其中,elementLengthPresent需要被设置为一。可以通过将usacExtElementPresent标志设置为零(假)来标示类型ID_USAC_EXT的空的有效载荷。

[0135] 因此,在S3040,在输出位流中标示(例如,发送或输出)生成的HOA扩展有效载荷。一般来说,在输出位流中标示(例如,发送或输出)多个层和分配给它们的有效载荷。此外,可以在输出位流中标示(例如,发送或输出)HOA解码器配置有效载荷和/或HOA配置扩展有效载荷。

[0136] 假定HOA基本层(等于1的层索引)是以最高的错误保护发送的,并且具有相对小的位速率。用于后面的层(一个或多个HOA增强层)的错误保护根据增强层的位速率提高而稳步地降低。由于不佳的发送条件和较低的错误保护,较高层的发送可能失败,并且在最坏的情况下,只有基本层被正确地发送。假定对于一个层的所有有效载荷应用组合的错误保护。因此如果一层的发送失败,则对应层的所有有效载荷缺失。

[0137] 换句话说,可以以相应的错误保护水平发送用于多个层的数据有效载荷,其中,基本层具有最高的错误保护,一个或多个增强层具有依次降低的错误保护。

[0138] 除非步骤需要某些其他步骤作为先决条件,否则前述步骤可以按任何次序执行,并且图3所示的示例性次序被理解为非限制性的。

[0139] 如上面所指示的,如果在HOADecoderConfig()中标示了等于1的CodedVVecLength,则需要使得用于基于矢量的信号的编码的V矢量元素的位流语法适于HOA分层编码。将参照图4来描述根据本公开的实施例的对应的编码方法(例如,声音或声场

的压缩HOA表示的帧的分层编码方法)。

[0140] 在图4中的S4010,将多个传输信号分配给多个分级层。可以以与上述S3010相同的方式执行该步骤。

[0141] 在S4020,确定矢量编码模式是否作用。这可以涉及确定是否CodedVVecLength==1。

[0142] 如上面所指示的,在常规的方法中,在矢量编码模式下,对于ContAddHoaCoeff集合中包括的HOA系数索引,不发送V矢量元素。该集合包括所有的HOA系数索引AmbCoeffIdx[i],这些HOA系数索引具有等于零的AmbCoeffTransitionState。因为用于这些索引的原始HOA系数序列被显式地发送,所以不需要还添加加权的V矢量信号。因此,对于这些索引,常规方法中的V矢量元素被设置为零。

[0143] 但是,在分层编码模式下,连续HOA系数索引集合依赖于传输信道,这些传输信道是当前作用的层的一部分。这意味着在较高层中发送的附加的HOA系数索引在较低层中是缺失的。那么,矢量信号不应该对HOA系数序列做出贡献的假定对于属于被包括在较高层中的HOA系数序列的HOA系数索引是错误的。

[0144] 因此,如果矢量编码模式是作用的,则在S4030,对于每个层基于分配给相应层的传输信号来确定(例如,限定)连续HOA系数索引(例如,ContAddHoaCoeff)集合。

[0145] 如果矢量编码模式是作用的,则在S4040,对于每个传输信号,基于用于被分配了相应的传输信号的层的确定的连续HOA系数索引集合来生成V矢量。每个生成的V矢量可以包括用于被分配给比被分配了相应的传输信号的层高的层的任何传输信号的元素。该步骤可以涉及使用已经针对其中添加了V矢量信号的层(V矢量信号的传输信号所属的层)确定的连续HOA系数索引集合来选择作用的V矢量元素。不过,提出了:V矢量数据留在HOAFrame()中并且不被移到HOAEnhFrame()。

[0146] 然后,在S4050,在输出位流中标示生成的V矢量(V矢量信号)。这可以涉及(显式地)标示用于前述缺失的系数索引的V矢量元素。

[0147] 图4中的步骤S4040至S4050也可以在图3所示的编码方法的上下文中被采用,例如,在S3010之后被采用。在这种情况下,可以将S3040和S4050组合为单个标示步骤。

[0148] 除非步骤需要某些其他的步骤作为先决条件,否则前述步骤可以按任何次序执行,并且图4所示的示例性次序被理解为是非限制性的。

[0149] 在接收器端,MPEG-H位流打包器可以重新将正确接收的有效载荷插入到基本层MPEG-H位流中,并且将它传递到MPEG-H 3D音频解码器。

[0150] 接着,将描述HOA解码初始化(配置)。类型ID_EXT_ELE_HOA和ID_EXT_ELE_HOA_ENH_LAYER的HOA配置有效载荷(具有它们的对应的以字节为单位的大小)被输入到HOA解码器以用于其初始化。HOA编码工具是根据HOAConfig()中定义的位流元素配置的,HOAConfig()是从类型ID_EXT_ELE_HOA的有效载荷解析的。此外,该有效载荷包含分层编码模式的使用、层的数量以及每一层的传输信号的对应数量。然后,如果分层编码被激活(SingleLayer==0),则从类型ID_EXT_ELE_HOA_ENH_LAYER的有效载荷解析HOAEnhConfig()以配置每个层的对应的空间信号预测、子带方向信号合成和参数化环境复制解码器。

[0151] 来自HOAEnhConfig()的元素LayerIdx与mpeg3daExtElementConfig()中的HOA增强层配置有效载荷的次序一起指示HOA增强层的次序。mpeg3daFrame()中的类型ID_

EXT_ELE_HOA_ENH_LAYER的HOA增强层帧有效载荷的次序与mpeg3daExtElementConfig()中的配置有效载荷的次序是相同的以清楚地将帧有效载荷分配给对应的层。

[0152] 在SingleLayer==1(单层编码)的情况下,忽视类型ID_EXT_ELE_HOA_ENH_LAYER的有效载荷,并且空间信号预测、子带方向信号合成和参数化环境复制解码器使用来自HOADecoderConfig()的对应数据以用于它们的配置。

[0153] 接着,将描述分层模式下的HOA帧解码。将参照图5来描述根据本公开的实施例的对应的解码方法(例如,对声音或声场的压缩HOA表示的帧进行解码的方法)。理解的是,压缩HOA表示(例如,上述图3或图4的方法的输出)已经被编码在多个分级层中,该多个分级层包括基本层和一个或多个增强层。

[0154] 在图5中的S5010,接收与压缩HOA表示的帧相关的位流。

[0155] 3D音频核心解码器对正确发送的HOA传输信号进行解码,并且对于对应的无效的有效载荷用等于零的采样来创建传输信号。解码的传输信号与usacExtElementPresent标志、类型ID_EXT_ELE_HOA和ID_EXT_ELE_HOA_ENH_LAYER的HOA有效载荷的数据和大小一起被输入到HOA解码器。来自类型ID_USAC_EXT的具有被设置为假的usacExtElementPresent标志的扩展有效载荷需要向HOA解码器标示其为缺失的有效载荷以保证有效载荷对于对应层的分配。

[0156] 在S5020,提取用于多个层的有效载荷。每个有效载荷可以包括分配给相应层的传输信号。

[0157] 在该步骤,HOA解码器可以从类型ID_EXT_ELE_HOA的有效载荷解析HOAFrame()。

[0158] 随后,通过评估有效载荷的对应的usacExtElementPresent标志来确定类型ID_EXT_ELE_HOA_ENH_LAYER的有效的有效载荷和类型ID_EXT_ELE_HOA_ENH_LAYER的无效的有效载荷,其中,无效的有效载荷由等于假的usacExtElementPresent标志指示,并且HOA增强有效载荷对于增强层索引的分配是从HOA解码器配置知道的。

[0159] 在S5030,在多个层中确定用于解码的最高可用层。

[0160] 因为就传输信号而言,层是彼此依赖的,所以HOA解码器只有在具有更低索引的所有层被正确地接收到时才可以对一层进行解码。可以在该步骤选择最高可用层以使得一直到最高可用层的所有层已经被正确地接收到。下面将描述该步骤的细节。

[0161] 在S5040,提取分配给最高可用层的HOA扩展有效载荷。如上面所指示的,HOA扩展有效载荷可以包括用于参数化地增强与最高可用层相对应的重构的HOA表示的辅助信息。其中,与最高可用层相对应的重构的HOA表示可以基于分配给最高可用层和低于最高可用层的任何层的传输信号获得。

[0162] 另外,可以提取分别分配给多个层中的其余的层的HOA扩展有效载荷。每个HOA扩展有效载荷可以包括用于参数化地增强与其相应的被分配的层相对应的重构的HOA表示的辅助信息。与其相应的被分配的层相对应的重构的HOA表示可以从分配给该层和低于该层的任何层的传输信号获得。

[0163] 此外(在图5中未示出),解码方法可以包括提取HOA配置扩展有效载荷的步骤。这可以通过对位流进行解析来进行。HOA配置扩展有效载荷可以包括用于配置HOA空间信号预测解码工具、HOA子带方向信号合成解码工具和/或HOA参数化环境复制解码工具的位流元素。

[0164] 在S5050,基于分配给最高可用层和低于最高可用层的任何层的传输信号来生成与最高可用层相对应的(部分)重构的HOA表示。

[0165] 实际上使用的传输信号的数量 $I_{ADD,LAY}(k)$ 是根据最高可用层(的索引 $M_{LAY}(k)$)设置的,并且第一初步HOA表示是从HOAFrame()以及从该层和任何更低的层的对应的传输信号解码的。

[0166] 然后,在S5060,使用分配给最高可用层的HOA扩展有效载荷中所包括的辅助信息来增强(例如,参数化地增强)重构的HOA表示。

[0167] 也就是说,然后通过空间信号预测、子带方向信号合成和参数化环境复制解码器使用从当前作用的层 $M_{LAY}(k)$ (即,最高可用层)的类型ID_EXT_ELE_HOA_ENH_LAYER的HOA增强层扩展有效载荷解析的HOAEnhFrame()数据来增强在S5050中获得的HOA表示。

[0168] 在步骤S5020-S5060使用的信息可以被称为层信息。

[0169] 除非步骤需要某些其他步骤作为先决条件,否则前述步骤可以按任何次序执行,并且图5所示的示例性次序被理解为非限制性的。

[0170] 接着,将描述S5030中的最高可用层的确定(例如,选择)的细节。

[0171] 如上面所指示的,HOA解码器只有在具有更低索引的所有层被正确地接收到时才可以对一层进行解码,因为就传输信号而言,层是彼此依赖的。

[0172] 为了选择最高可解码层,HOA解码器可以创建无效层索引集合,其中,该集合中的最小索引减一得到最高可解码增强层的索引 M_{LAY} 。可以通过评估对应的HOA扩展有效载荷的有效性标志来确定无效层索引集合。

[0173] 换句话说,确定最高可用层可以涉及确定指示尚未被有效地接收到的层的无效层索引集合。它可以进一步涉及将最高可用层确定为是由无效层索引集合中的最小索引指示的层下面的一层的层。从而,确保最高可用层下面的所有层已经被有效地接收到。

[0174] 在对帧进行差分编码的情况下,将需要考虑先前帧(例如,紧邻的前一帧)的最高可用层的索引。首先,将描述先前帧(例如,前一帧)的最高可用层的索引被保持的情形。

[0175] 如果用于当前帧的最高可用层(例如,最高可解码层)的索引等于先前帧的层索引 $M_{LAY}(k-1)$,则将当前帧的层索引 $M_{LAY}(k)$ 设置为 $M_{LAY}(k-1)$ 。

[0176] 然后,如上面所指示的,根据 $M_{LAY}(k)$ 来设置实际上使用的传输信号的数量 $I_{ADD,LAY}(k)$,并且从HOAFrame()以及从该层和任何更低的层的对应的传输信号解码第一初步HOA表示。如上面所指示的,然后通过空间信号预测、子带方向信号合成和参数化环境复制解码器使用从当前作用的层 $M_{LAY}(k)$ 的类型ID_EXT_ELE_HOA_ENH_LAYER的HOA增强层扩展有效载荷被解析的HOAEnhFrame()数据来增强该HOA表示。

[0177] 接着,将描述它被切换到比先前帧(例如,前一帧)的最高可用层的索引低的索引的情形。也就是,在当前帧的最高可解码层的索引小于先前帧的层的索引 $M_{LAY}(k-1)$ 的情况下,HOA解码器将 $M_{LAY}(k)$ 设置为当前帧的最高可解码层的索引。用于新的层的空间信号预测、子带方向信号合成和参数化环境复制解码器的有效载荷的解码只有在具有等于1的hoaIndependencyFlag的下一个HOA帧处才可以开始。在不执行空间信号预测、子带方向信号合成和参数化环境复制解码器的情况下重构索引 $M_{LAY}(k)$ 的层的HOA表示,一直到这样的HOAFrame()已经被接收到。这意味着实际上使用的传输信号的数量 $I_{ADD,LAY}(k)$ 是根据 $M_{LAY}(k)$ 设置的,并且只有第一初步HOA表示是从HOAFrame()以及从该层和任何更低的层的对

应的传输信号解码的。然后,如果具有等于1的`hoaIndependencyFlag`的`H0AFrame()`已经被接收到,则对用于空间信号预测、子带方向信号合成和参数化环境复制解码器的有效载荷进行解析和解码以增强初步HOA表示,以使得对该帧提供当前作用的层的全面质量。

[0178] 因此,所提出的方法可以包括(在图5中未示出):如果当前帧的最高可用层低于先前帧的最高可用层(如果当前帧已经被相对于先前帧不同地编码),则决定不使用分配给最高可用层的HOA扩展有效载荷中所包括的辅助信息来执行重构的HOA表示的参数化增强。

[0179] 一般来说,确定当前帧的最高可用层可以涉及对于当前帧确定指示尚未被有效地接收到的层的无效层索引的集合。它可以进一步包括确定当前帧前面的先前帧的最高可用层。它还可以进一步包括将最高可用层确定为以下层中的较低的一个:先前帧的最高可用层、以及作为由无效层索引集合中的最小索引指示的层下面的一个层的层(如果当前帧已经被相对于先前帧不同地编码)。

[0180] 替代解决方案可以总是并行地对所有的有效的增强层有效载荷(例如,HOA扩展有效载荷)进行解析,即使它们当前是不作用的。这将使得能够以全面的质量直接切换到具有更低索引的层,其中,空间信号预测、子带方向信号合成和参数化环境复制(PAR)解码器可以被直接应用于切换到的帧处。

[0181] 接着,将描述它被切换到比先前帧(例如,前一帧)的最高可用层的索引高的索引的情形。到具有更高索引的层的这个切换只有在`mpegh3daFrame()`具有等于1的`usacIndependencyFlag`(例如,帧是独立帧)时才可以被应用,因为先前帧的所有的对应的有效载荷或解码级是缺失的。因此,HOA解码器使HOA层索引 $M_{LAY}(k)$ 保持等于 $M_{LAY}(k-1)$,一直到包含用于更高的可解码层的有效数据的、具有等于1的`usacIndependencyFlag`的`mpegh3daFrame()`(例如,独立帧)已经被接收到为止。然后,将 $M_{LAY}(k)$ 设置为当前帧的最高可解码层索引,并且相应地,确定实际上使用的传输信号的数量 $I_{ADD,LAY}(k)$ 。从`H0AFrame()`和对应的传输信号解码该层的初步HOA表示,并且通过空间信号预测、子带方向信号合成和参数化环境复制解码器使用从当前作用的层 $M_{LAY}(k)$ 的类型`ID_EXT_ELE_HOA_ENH_LAYER`的HOA增强层扩展有效载荷解析的`H0AEnhFrame()`来增强该初步HOA表示。

[0182] 理解的是,所提出的压缩声音表示的分层编码方法可以通过用于压缩声音表示的分层编码的编码器来实现。这样的编码器可以包括适于执行上述各步骤的各单元。图6中示意性地示出了这样的编码器6000的例子。例如,这样的编码器6000可以包括适于执行前述S3010的传输信号分配单元6010、适于执行前述S3020的HOA扩展层有效载荷生成单元6020、适于执行前述S3030的HOA扩展有效载荷分配单元6030以及适于执行前述S3040的标示单元或输出单元6040。进一步理解的是,这样的编码器的各单元可以由计算设备的处理器6100实施,处理器6100适于执行由所述各单元中的每个执行的处理,即,适于执行图3中示意性地示出的所提出的编码方法的前述步骤中的一些或全部。另外地或可替代地,处理器6100可以适于执行图4中示意性地示出的编码方法的每个步骤。为了这个目的,处理器6100可以适于实现编码器的各单元。编码器或计算设备可以进一步包括处理器6100可访问的存储器6200。

[0183] 进一步理解的是,所提出的对被编码在多个分级层中的压缩声音表示进行解码的方法可以通过用于对编码在多个分级层中的压缩声音表示进行解码的解码器来实现。这样的解码器可以包括适于执行上述各步骤的各单元。图7中示意性地示出了这样的解码器

7000的例子。例如,这样的解码器7000可以包括适于执行前述S5010的接收单元7010、适于执行前述S5020的有效载荷提取单元7020、适于执行前述S5030的最高可用层确定单元7030、适于执行前述S5040的HOA扩展有效载荷提取单元7040、适于执行前述S5050的重构HOA表示生成单元7050以及适于执行前述S5060的增强单元7060。进一步理解的是,这样的解码器的各单元可以由计算设备的处理器7100实施,处理器7100适于执行由所述各单元中的每个执行的处理,即,适于执行所提出的解码方法的前述步骤中的一些或全部。解码器或计算设备可以进一步包括处理器7100可访问的存储器7200。

[0184] 接着,将描述用于在分层编码模式下容纳(例如,表示)压缩HOA表示的数据结构(例如,位流)。这样的数据结构可以起因于采用所提出的编码方法,并且可以通过使用所提出的解码方法解码(例如,解压缩)。

[0185] 该数据结构可以包括与多个分级层中的相应的分级层相对应的多个HOA帧有效载荷。多个传输信号可以被分配给(例如,可以属于)多个层中的相应的层。数据结构可以包括相应的HOA扩展有效载荷,该HOA扩展有效载荷包括用于参数化地增强从分配给相应层和低于相应层的任何层的传输信号可获得的重构的HOA表示的辅助信息。如上面所指示的,用于多个层的HOA帧有效载荷和HOA扩展有效载荷可以设有相应的错误保护水平。此外,HOA扩展有效载荷可以包括上面指示的位流元素,并且可以具有ID_EXT_ELE_HOA_ENH_LAYER的usacExtElementType。该数据结构还可以进一步包括HOA配置扩展有效载荷和/或HOA解码器配置有效载荷,该HOA配置扩展有效载荷和/或HOA解码器配置有效载荷包括上面指示的位流元素。

[0186] 应注意到,描述和附图仅示出了所提出的方法和装置的原理。因此将意识到,本领域技术人员将能够想出各种布置,这些布置尽管在本文中没有被明确地描述或示出,但是体现了本发明的原理,并且包括在本发明的精神和范围内。此外,本文所记载的所有例子主要明确地意图仅用于教学的目的以帮助读者理解所提出的方法和装置的原理以及发明人对推进现有技术所贡献的构思,并且要被解释为不限于这样的具体记载的例子和条件。而且,本文的记载本发明的原理、方面和实施例以及其特定例子的所有陈述意图涵盖其等同物。

[0187] 本文档所描述的方法和装置可以实现为软件、固件和/或硬件。某些组件可以例如实现为在数字信号处理器或微处理器上运行的软件。其他组件可以例如实现为硬件和/或专用集成电路。在所描述的方法和装置中遇到的信号可以存储在介质(诸如随机存取存储器或光学存储介质)上。它们可以经由网络(诸如无线网络、卫星网络、无线网络或有线网络(例如,互联网))传送。

[0188] 附录:

[0189] 所提出的MPEG-H 3D位流改变

[0190] 通过用灰色高亮来标记改变:

[0191] 表1——mpeg3daExtElementConfig()的语法

	语法	位数	助记符
	<pre>mpegh3daExtElementConfig() { usacExtElementType = escapedValue(4, 8, 16); usacExtElementConfigLength = escapedValue(4, 8, 16); if (usacExtElementDefaultLengthPresent) { usacExtElementDefaultLength = escapedValue(8, 16, 0) + 1; } else { usacExtElementDefaultLength = 0; } usacExtElementPayloadFrag; switch (usacExtElementType) { case ID_EXT_ELE_FILL: /* 无配置元素 */ break; case ID_EXT_ELE_MPEGS: SpatialSpecificConfig(); break; case ID_EXT_ELE_SAOC: SAOCSpecificConfig(); break; case ID_EXT_ELE_AUDIOPREROLL: /*无配置元素*/ }</pre>	1	uimbsf
[0192]		1	uimbsf

[0193]

<pre> break; case ID_EXT_ELE_UNI_DRC: mpeggh3daUniDrcConfig(); break; case ID_EXT_ELE_OBJ_METADATA: ObjectMetadataConfig(); break; case ID_EXT_ELE_SAOC_3D: SAOC3DSpecificConfig(); break; case ID_EXT_ELE_HOA: HOAConfig(); break; case ID_EXT_ELE_HOA_ENH_LAYER: HOAEnhConfig(); break; case ID_EXT_ELE_FMT_CNVTR /*无配置元素*/ break; default: while (usacExtElementConfigLength--) { tmp; } break; } }</pre>			
	NOTE		
	8	uimbsf	
注意：usacExtElementType的默认条目用于未知的extElementTypes以使得传统解码器可以处理未来的扩展。			

[0194] 表2——usacExtElementType的值

[0195]

usacExtElementType	值
ID_EXT_ELE_FILL	0
ID_EXT_ELE_MPEGS	1
ID_EXT_ELE_SAOC	2
ID_EXT_ELE_AUDIOPREROLL	3
ID_EXT_ELE_UNI_DRC	4
ID_EXT_ELE_OBJ_METADATA	5
ID_EXT_ELE_SAOC_3D	6
ID_EXT_ELE_HOA	7
ID_EXT_ELE_FMT_CNVTRTR	8
ID_EXT_ELE_HOA_ENH_LAYER	9
/* 保留以用于 ISO 使用 */	10-127
/* 保留以用于在 ISO 范围外部使用 */	128 和更高
注意：应用特定的 usacExtElementType 值被要求在被保留在 ISO 范围外使用的空间中。这些被解码器跳过，因为最小结构是解码器跳过这些扩展所需要的。	

[0196] 表3——用于扩展有效载荷解码的数据块的解释

[0197]

usacExtElementType	连结的 usacExtElementSegmentData 表示:
ID_EXT_ELE_FILL	fill_byte 系列
ID_EXT_ELE_MPEGS	如 ISO/IEC 23003-1 中定义的 SpatialFrame()
ID_EXT_ELE_SAOC	如 ISO/IEC 23003-2 中定义的 SAOCFrame()
ID_EXT_ELE_AUDIOPREROLL	AudioPreRoll()
ID_EXT_ELE_UNI_DRC	如 ISO/IEC 23003-4 中定义的 uniDrcGain()
ID_EXT_ELE_OBJ_METADATA	object_metadata()
ID_EXT_ELE_SAOC_3D	Saoc3DFrame()
ID_EXT_ELE_HOA	HOAFrame()
ID_EXT_ELE_HOA_ENH_LAYER	HOAEnhFrame()

[0198]

ID_EXT_ELE_FMT_CNVTR	FormatConverterFrame()
未知的	未知数据。该数据块应被丢弃。

[0199] 表4——HOADecoderConfig()的语法

[0200]

语法	位数	助记符
HOADecoderConfig(numHOATransportChannels)		
{		
MinAmbHoaOrder = escapedValue(3,5,0) - 1;	3,8	uimsbf
MinNumOfCoeffsForAmbHOA = (MinAmbHoaOrder + 1)^2;		
NumOfAdditionalCoders =		
numHOATransportChannels - MinNumOfCoeffsForAmbHOA;		
if(SingleLayer == 0){	1	bslbf
HOALayerChBits = ceil(log2(NumOfAdditionalCoders));		
NumHOAChannelsLayer[0] = codedLayerCh +	HOALayerCh	uimsbf
MinNumOfCoeffsForAmbHOA;	Bits	
remainingCh = numHOATransportChannels -		
NumHOAChannelsLayer[0];		
NumLayers = 1;		
while (remainingCh>1) {		
HOALayerChBits = ceil(log2(remainingCh));		
NumHOAChannelsLayer[NumLayers] =	HOALayerCh	uimsbf
NumHOAChannelsLayer[NumLayers-1] +	Bits	
codedLayerCh + 1;		
remainingCh = remainingCh -		
NumHOAChannelsLayer[NumLayers];		
NumLayers++;		
}		
if (remainingCh) {		

[0201]

```

        NumHOAChannelsLayer[NumLayers] =
            NumOfAdditionalCoders;
        NumLayers++;
    }
}

MaxNoOfDirSigsForPrediction = MaxNoOfDirSigsForPrediction + 1;      2      uimsbf
NoOfBitsPerScalefactor = NoOfBitsPerScalefactor + 1;                4      uimsbf
    CodedSpatialInterpolationTime;                                       3      uimsbf
    SpatialInterpolationMethod;                                           1      bslbf
    CodedVVecLength;                                                       2      uimsbf
    MaxGainCorrAmpExp;                                                     3      uimsbf
    HOAFrameLengthIndicator;                                              2      uimsbf
    MaxHOAOrderToBeTransmitted = escapedValue(2,5,0) +
        MinAmbHoaOrder;
    MaxNumOfCoeffsToBeTransmitted =
        (MaxHOAOrderToBeTransmitted + 1)^2;
    MaxNumAddActiveAmbCoeffs =
        MaxNumOfCoeffsToBeTransmitted
        - MinNumOfCoeffsForAmbHOA;
    VqConfBits = ceil( log2( ceil( log2(NumOfHoaCoeffs))) )
    NumVVecVqElementsBits++;                                              VqConfBi  uimsbf
                                                                    ts
    if( MinAmbHoaOrder == 1 ) {
        UsePhaseShiftDecorr;                                             1      bslbf
    }
    if(SingleLayer==1) {
        HOADecoderEnhConfig();                                           2      uimsbf
    }
    AmbAssignmBits = ceil( log2( MaxNumAddActiveAmbCoeffs ) );
    ActivePredIdsBits = ceil( log2( NumOfHoaCoeffs ) );

```

[0202]

```

    i = 1;
    while( i * ActivePredIdsBits + ceil( log2( i ) ) < NumOfHoaCoeffs ){
        i++;
    }
    NumActivePredIdsBits = ceil( log2( max( 1, i - 1 ) ) );
    GainCorrPrevAmpExpBits = ceil( log2( ceil( log2(
        1.5 * NumOfHoaCoeffs ) )
        + MaxGainCorrAmpExp + 1 ) );
    for (i=0; i<NumOfAdditionalCoders; ++i){
        AmbCoeffTransitionState[i] = 3;
    }
}

```

注意：MinAmbHoaOrder = 30 ... 37 被保留。HOAFrameLengthIndicator = 3 被保留。

[0203] 新表?——HOAEnhConfig() 的语法

[0204]

语法	位数	助记符
HOAEnhConfig()		
{		
LayerIdx	HOALayerC	uimsbf
	hBits	
HOADecoderEnhConfig();		
}		

[0205] 新表?——HOADecoderEnhConfig() 的语法

[0206]

语法	位数	助记符
HOADecoderEnhConfig()		
{		
MaxNoOfDirSigsForPrediction = MaxNoOfDirSigsForPrediction + 1;	2	uimsbf

[0207]

```

NoOfBitsPerScalefactor = NoOfBitsPerScalefactor + 1;          4          uimsbf

if(PredSubbandsIdx < 3) {
    NumOfPredSubbands =
        NumOfPredSubbandsTable[PredSubbandsIdx];
    PredSubbandWidths =
        PredSubbandWidthTable[PredSubbandsIdx];
}
else {
    CodedNumberOfSubbands          5          uimsbf
    NumOfPredSubbands = CodedNumberOfSubbands+1;
    PredSubbandWidths =
        getSubbandWidths(NumOfPredSubbands);
}

if ( NumOfPredSubbands > 0 ) {
    FirstSBRSubbandIdxBits = ceil( log2 (NumOfPredSubbands+1));
    FirstSBRSubbandIdx;          FirstSBRSubbandIdx  uimsbf
                                ubbandIdx
                                xBits

    MaxNumOfPredDirs = 2^( MaxNumOfPredDirsLog2);          3          uimsbf
    MaxNumOfPredDirsPerBand = escapedValue(3,2,5) + 1;
    NumOfBitsPerDirIdx =          2          uimsbf
        NumOfBitsPerDirIdxTable[DirGridTableIdx];
}

if( ParSubbandTableIdx < 3 ) {          2          uimsbf
    NumOfParSubbands =
        NumOfParSubbandsTable[ParSubbandTableIdx];
    ParSubbandWidths =
        ParSubbandWidthTable[ParSubbandTableIdx];
}

```

[0208]

else {		
CodedNumberOfSubbands	5	uimsbf
NumOfParSubbands = CodedNumberOfSubbands+1;		
ParSubbandWidths =		
getSubbandWidths(NumOfParSubbands);		
}		
if(NumOfParSubbands > 0) {		
LastFirstOrderSubbandIdxBits =		
ceil(log2(NumOfParSubbands + 1));		
LastFirstOrderSubbandIdx;	LastFirst	uimsbf
	OrderSub	
	bandIdxB	
	its	
for (idx = 0; idx < NumOfParSubbands; idx++) {		
UseRealCoeffsPerParSubband[idx];	1	bslbf
}		
for (idx = 0; idx < LastFirstOrderSubBandIdx; idx++) {		
UpmixHoaOrderPerParSubband[idx] = 1;		
MaxNumOfDecoSigs[idx] =		
(UpmixHoaOrderPerParSubband[idx] + 1)^2;		
}		
for (idx = LastFirstOrderSubBandIdx;		
idx < NumOfParSubbands; idx++) {		
UpmixHoaOrderPerParSubband[idx] = 2;		
MaxNumOfDecoSigs[idx] =		
(UpmixHoaOrderPerParSubband[idx] + 1)^2;		
}		
}		
}		
}		

[0209] 表5——HOAFrame的语法

[0210]

语法	位数	助记符
<pre>HOAFrame() { NumOfDirSigs = 0; for(lay=0; (lay< NumLayers) & !SingleLayer; ++lay){ NumOfDirSigsPerLayer[lay] = 0; NumOfAddHoaChansPerLayer[lay] = 0; NumOfContAddHoaChans[lay] = 0; } NumOfVecSigs = 0; NumOfAddHoaChans = 0; NumOfAddVVecValCoeffIdx = 0; hoaindependencyFlag; for(i=0; i< NumOfAdditionalCoders; ++i){ ChannelSideInfoData(i); HOAGainCorrectionData(i); switch ChannelType[i] { case 0: DirSigChannelIds[NumOfDirSigs] = i + 1; NumOfDirSigs++; for(lay=0; (lay< NumLayers) & !SingleLayer; ++lay){ if((MinNumOfCoeffsForAmbHOA + i) < NumHOAChannelsLayer[lay]){ NumOfDirSigsPerLayer[lay]++; } } break; case 1:</pre>	1	bslbf

[0211]

```

VecSigChannelIdx[NumOfVecSigs] = i + 1;
VecSigLayerIdx[NumOfVecSigs] = 0;
if (SingleLayer == 0) {
    lay = 0;
    while( (MinNumOfCoeffsForAmbHOA + i )
           ≥ NumHOAChannelsLayer[lay]) {
        lay ++;
    }
    VecSigLayerIdx[NumOfVecSigs] = lay;
}
NumOfVecSigs++;
break;
case 2:
    if (AmbCoeffTransitionState[i] == 0) {
        for(lay=0; (lay< NumLayers); ++lay){
            if( (MinNumOfCoeffsForAmbHOA + i ) <
                NumHOAChannelsLayer[lay]){
                ContAddHoaCoeff[lay]
                    [NumOfContAddHoaChans[lay]] =
                    = AmbCoeffIdx[i];
                NumOfContAddHoaChans[lay]++;
            }
        }
        AddHoaCoeff[NumOfAddHoaChans] = AmbCoeffIdx[i];
        for(lay=0; (lay< NumLayers) & (SingleLayer == 0); ++lay){
            if( (MinNumOfCoeffsForAmbHOA + i ) <
                NumHOAChannelsLayer[lay]){
                AddHoaCoeffPerLayer[lay][NumOfAddHoaChans]
                    = AmbCoeffIdx[i];
                NumOfAddHoaChansPerLayer[lay]++;
            }
        }
    }
}

```

[0212]

```
        }  
    }  
    NumOfAddHoaChans++;  
    break;  
}  
}  
  
for ( i= NumOfAdditionalCoders;  
      i< NumHOATransportChannels; ++i){  
    HOAGainCorrectionData(i);  
}  
  
for(i=0; i< NumOfVecSigs; ++i){  
    VVectorData ( VecSigChannelIds(i) );  
}  
  
if(SingleLayer==1) {  
    HOAEnhFrame();  
}
```

}

注意：如果 `usacIndependencyFlag`（参见 `mpegh3daFrame()`）被设置为 1，则编码器应将 `hoaIndependencyFlag` 设置为 1。

注意：如果 `SingleLayer == 1`，设置 `NumLayers = 1`。

[0213] `NumOfDirSigsPerLayer[lay]` 该元素确定在HOA增强层`lay`中实际使用的当前的 `HOAFrame()` 中的作用的力向信号的数量。

[0214] `AddHoaCoeffPerLayer[lay]` 该阵列包含用于在HOA增强层`lay`中实际使用的每个附加的环境HOA系数的HOA系数索引。`NumOfAddHoaChansPerLayer[lay]` 该元素标示在HOA增强层`lay`中实际使用的附加的环境HOA系数的总数。

[0215] 添加该表

[0216] 新表?——HOAEnhFrame的语法

[0217]

语法	位数	助记符
<pre>HOAEnhFrame() { if(((SingleLayer==1) & (NumOfDirSigs > 0)) ((SingleLayer==0) & (NumOfDirSigsPerLayer[lay]) > 0)){ HOAPredictionInfo() } if(NumOfPredSubbands > 0) { HOADirectionalPredictionInfo(); } if(NumOfParSubbands > 0) { HOAParInfo(); } }</pre>		
注意：lay 是当前作用的 HOA 增强层的索引。		

[0218] 更新该表：

[0219] 表6——VVectorData() 的语法

[0220]

语法	位数	助记符
<pre>VVectorData(i) { if (CodedVVecLength == 1) { VVecLengthUsed = VVecLength[i]; VVecCoeffIdUsed = VVecCoeffId[i]; } }</pre>		

[0221]

```

} else {
    VVecLengthUsed = VVecLength;
    VVecCoeffIdUsed = VVecCoeffId;
}
if (NbitsQ(k)[i] == 4){
    if (NumVvecIndices(k)[i] == 1) {
        VecIdx[o] = VecIdx + 1;           10      uimsbf
        WeightVal[o] = ((SgnVal*2)-1);    1      uimsbf
    } else {
        WeightIdx;                        8      uimsbf
        nbitsIdx = ceil(log2(NumOfHoaCoeffs));
        for (j=0; j< NumVvecIndices(k)[i]; ++j) {
            VvecIdx[j] = VvecIdx + 1;      nbitsIdx uimsbf
            WeightVal[j] = ((SgnVal*2)-1) * 1      uimsbf
            WeightValCdbk[CodebkIdx(k)[i]][WeightIdx[j]];
        }
    }
}
else if (NbitsQ(k)[i] == 5) {
    for (m=0; m< VVecLengthUsed; ++m){
        aVal[i][m] = (VecVal / 128.0) - 1.0; 8      uimsbf
    }
}
else if (NbitsQ(k)[i] >= 6) {
    for (m=0; m< VVecLengthUsed; ++m){
        huffIdx = huffSelect(VVecCoeffIdUsed[m], PFlag[i],
            CbFlag[i]);
        cid = huffDecode(NbitsQ[i], huffIdx, huffVal);  dynamic huffDecode
        aVal[i][m] = 0.0;
        if ( cid > 0 ) {

```

[0222]

aVal[i][m] = sgn = (sgnVal * 2) - 1;	1	bslbf
if (cid > 1) {		
aVal[i][m] = sgn * (2.0^(cid - 1) + intAddVal);	cid-1	uimsbf
}		
}		
}		
}		
}		
注意：计算VVecLength参见错误！未找到引用源。		

[0223] 表7——HOAPredictionInfo (DirSigChannelIds, NumOfDirSigs) 的语法

[0224]

语法	位数	助记符
HOAPredictionInfo() { if(SingleLayer==1){ PredIdsBits = ceil(log2(NumOfDirSigs + 1)); } else{ PredIdsBits = ceil(log2(NumOfDirSigsPerLayer[lay] + 1)); } if(PSPredictionActive){ NumActivePred = 0; if(KindOfCodedPredIds){ NumActivePred = NumActivePredIds + 1; i=0; while(i < NumActivePred){ PredIds[i] = PredIds[i] + 1; 	1	bslbf
	1	bslbf
	NumActivePredIdsBits	uimsbf
	ActivePredIdsBits	uimsbf

[0225]

```
        i++;
    }
}
else{
    for (i=0; i<((HoaOrder +1)^2; i++) {
        if(ActivePred[i]) {
            NumActivePred ++;
        }
    }
}
NumOfGains=0;
for (i=0; i<NumActivePred * MaxNoOfDirSigsForPrediction; i++) {
    if (PredDirSigIds[i] > 0) {
        PredIdsBits
        uimbsf
        PredDirSigIds[i] =
            DirSigChannelIds[PredDirSigIds[i] - 1];
        NumOfGains++;
    }
}
n=0;
for (i=0; i< NumOfGains; i++) {
    if (PredDirSigIds[i]>0) {
        PredGains[n];
        NoOfBitsPerScalefa
        ctor
        n++;
    }
}
}
```

注意：lay 是当前作用的 HOA 增强层的索引

[0226] 表AMD1.2——HOADirectionalPredictionInfo()的语法

[0227]

语法	位数	助记符
HOADirectionalPredictionInfo ()		
{		
if(UseDirectionalPrediction) {	1	bslbf
if (!hoalIndependencyFlag) {		
KeepPreviousGlobalPredDirsFlag;	1	bslbf
}		
else{		
KeepPreviousGlobalPredDirsFlag = 0;		
}		
if(!KeepPreviousGlobalPredDirsFlag) {		
NumOfGlobalPredDirs = NumOfGlobalPredDirs + 1;	MaxNumOf	bslbf
	PredDirsLog	
	2	
NumBitsForRelDirGridIdx = ceil(
log2(NumOfGlobalPredDirs));		
for (idx=0; idx < NumOfGlobalPredDirs; idx++) {		
GlobalPredDirsIds[idx];	NumOfBitsP	uimsbf
	erDirIdx	
}		
}		
else{		
/* 保持来自用于 NumOfGlobalPredDirs 和		
GlobalPredDirsIds 的先前的		
HOADirectionalPredictionInfo		
有效载荷的值*/		
}		
if(SingleLayer==1){		

[0228]

```

        SortedAddHoaCoeff = sort(AddHoaCoeff,
                                'ascend');

        NumOfAddHoaChansUsed = NumOfAddHoaChans;
    }
    else{
        SortedAddHoaCoeff = sort(AddHoaCoeffPerLayer[lay],
                                'ascend');

        NumOfAddHoaChansUsed =
            NumOfAddHoaChansPerLayer[lay];
    }

    for ( band = 0; band < NumOfPredSubbands; band++ ) {
        for ( dir = 0; dir < MaxNumOfPredDirsPerBand; dir++ ) {
            for ( hoaldx = 0;
                hoaldx < MinNumOfCoeffsForAmbHOA;
                hoaldx++ ) {
                DecodedMagDiff[band][dir][hoaldx] = 0;
                DecodedAngleDiff[band][dir][hoaldx] = 0;
            }
        }
    }

    for ( band = 0; band < NumOfPredSubbands; band++ ) {
        if (!hoalndependencyFlag) {
            KeepPreviousDirPredMatrixFlag[band];           1           bslbf
        }
        else{
            KeepPreviousDirPredMatrixFlag[band] = 0;
        }
        if (!KeepPreviousDirPredMatrixFlag[band]) {
            UseHuffmanCodingDiffMag;                       1           bslbf

```

[0229]

```

        if( band < FirstSBRSubbandIdx ) {

            UseHuffmanCodingDiffAngle;                1          bslbf

        for ( dir = 0; dir < MaxNumOfPredDirsPerBand; dir++) {

            if ( DirlsActive[band][dir] ) {                1          bslbf

                RelDirGridIdx;                            NumBitsFor  uimsbf

                                                            RelDirGridId

                                                            x

                PredDirGridIdx[band][dir] =

                    GlobalPredDirIds[RelDirGridIdx];

                for ( hoalidx = 0;

                    hoalidx < MinNumOfCoeffsForAmbHOA;

                    hoalidx++ ) {

                    readDirPredDiffValues (band, dir, hoalidx,

                        UseHuffmanCodingDiffAbs,

                        UseHuffmanCodingDiffAngle,

                        FirstSBRSubbandIdx);

                }

                for ( idx = 0;

                    idx < NumOfAddHoaChansUsed;

                    idx++ ) {

                    readDirPredDiffValues (band, dir,

                        SortedAddHoaCoeff[idx] -1,

                        UseHuffmanCodingDiffAbs,

                        UseHuffmanCodingDiffAngle,

                        FirstSBRSubbandIdx);

                }

            }

        }

    }

}

```

[0230]	{	
	{	
	注意：lay 是当前作用的 HOA 增强层的索引	

[0231] 表8——SingleLayer定义

[0232]	值	意义
	0	HOA 信号被设置在多个层中；使得能够标示 HOA 传输信道到不同层中的分布
	1	HOA 信号被设置在单个层中

[0233] codedLayerCh该元素对于第一（即，基本）层指示包括的传输信号的数量，该数量由codedLayerCh+MinNumOfCoeffsForAmbHOA给出。对于较高的（即，增强）层，该元素指示与下一较低层相比包括到增强层中的附加信号的数量，该数量由codedLayerCh+1给出。

[0234] HOALaverChBits该元素指示用于读取codedLayerCh的位数。

[0235] NumLavers该元素指示位流内的总层数（在读取HOADecoderConfig（）之后）。

[0236] NUmHOAChannelsLaver该元素是包括NumLayers元素的数组，其中第i元素指示一直到第i层的所有层中包括的传输信号的数量。

[0237] 12.4.1X帧和用户依赖参数

[0238] $M_{LAY}(k)$ 在解码器端用于第k帧（将被指定）的实际使用的层的数量。在分层编码（由SingleLayer==0指示）的情况下，该数量必须小于或等于位流中存在的总层数，即， $M_{LAY} \leq NumLayers$ 。在单层编码（由SingleLayer==1指示）的情况下， M_{LAY} 被设置为1。

[0239] 依赖于 $M_{LAY}(k)$ 的选择，如下计算实际上用于空间HOA解码的附加传输信道（即，除了总是被隐式地使用的 0_{MIN} 个信道之外）的数量 $I_{ADD,LAY}(k)$ ：if (SingleLayer | (!SingleLayer& $M_{LAY}(k) == NumLayers$))

[0240] {

[0241] $I_{ADD,LAY}(k) = NumOfAdditionalCoders$;

[0242] }

[0243] else

[0244] {

[0245] $I_{ADD,LAY}(k) = NumHOACannelsLayer[M_{LAY}(k) - 1] - MinNumOfCoeffsForAmbHOA$;

[0246] }

[0247] VVecLength和VVecCoeffId

[0248] codedVVecLength字指示：

[0249] 0) 完整矢量长度（NumOfHoaCoeffs元素）。指示用于主导矢量的所有系数（NumOfHoaCoeffs）被指定。

[0250] 1) 矢量元素1至MinNumOfCoeffsForAmbHOA和索引为lay=0...NumLayers-1的当前作用的层的ContAddHoaCoeff[lay]中定义的所有元素没有被发送。对于单层模式

SingleLayer==1,变量NumLayers需要被设置为等于1。指示主导矢量的只有与大于MinNumOfCoeffsForAmbH0A的数量相对应的那些系数被指定。此外,ContAddAmbH0aChan[lay]中标识的那些NumOfContAddAmbH0aChan[lay]系数被去掉。列表ContAddAmbH0aChan[lay]指定与超过次序(order)MinAmbH0aOrder的次序相对应的附加信道。

[0251] 2) 矢量元素1至MinNumOfCoeffsForAmbH0A没有被发送。指示主导矢量的与大于MinNumOfCoeffsForAmbH0A的数量相对应的那些系数被指定。

[0252] 在codedVVecLength==1的情况下,VVecLength[i]阵列和VVecCoeffId[i][m]2D阵列这二者对于索引i的VVector是有效的,在其他情况下,VVecLength元素和VVecCoeffId[m]阵列这二者对于H0AFrame内的所有VVector是有效的。对于下面的分配算法,如下定义助手函数。

[0253] switch CodedVVecLength{

```
case 0:
    VVecLength = NumOfHoaCoeffs;
    for (m=0; m<VVecLength; ++m) {
        VVecCoeffId[m] = m;
    }
    break;
case 1:
    for (i=0; i < NumOfVecSigs; ++i) {
        lay = VecSigLayerIdx[i];
        VVecLength[i] = NumOfHoaCoeffs
            - MinNumOfCoeffsForAmbHOA
            - NumOfContAddHoaChans[lay];
        CoeffIdx = MinNumOfCoeffsForAmbHOA+1;
        for (m=0; m<VVecLength[i]; ++m) {
            blsInArray = isMemberOf(CoeffIdx,
                ContAddHoaCoeff[lay],
                NumOfContAddHoaChans[lay]);
            while (blsInArray) {
                CoeffIdx++;
                blsInArray = isMemberOf(CoeffIdx,
                    ContAddHoaCoeff[lay],
                    NumOfContAddHoaChans[lay]);
            }
            VVecCoeffId[i][m] = CoeffIdx-1;
        }
    }
    break;
case 2:
    VVecLength = NumOfHoaCoeffs - MinNumOfCoeffsForAmbHOA;
    for (m=0; m<VVecLength; ++m) {
```

[0254]

VVecCoeffId[m] = m + MinNumOfCoeffsForAmbHOA;

[0255] }

}

[0256] 具有三种情况(情况0-2)的第一开关语句从而提供了确定就数量而言的主导矢量长度(VVecLength)以及系数的索引(VVecCoeffId)的方式。

[0257] 12.4.1X到VVec元素的转换

[0258] V矢量的去量化的种类是用字NbBitsQ标示的。NbBitsQ值为4指示矢量量化。当NbBitsQ等于5时,执行均匀的8位标量去量化。相反,NbBitsQ值大于或等于6指示应用标量量化的V矢量的哈夫曼解码。预测模式被表示为PFlag,而CbFlag表示哈夫曼表信息位。

if (CodedVVecLength == 1) {

VVecLengthUsed = VVecLength[i];

VVecCoeffIdUsed = VVecCoeffId[i];

} else {

VVecLengthUsed = VVecLength;

VVecCoeffIdUsed = VVecCoeffId;

}

if (NbBitsQ(k)[i] == 4) {

if (NumVVecIndices == 1) {

[0259] for (m=0; m< VVecLengthUsed; ++m) {

idx = VVecCoeffIdUsed[m];

$v_{idx}^{(i)}(k) = \text{WeightVal}[o] * \text{VecDict}[goo].[\text{VvecIdx}[o]][idx];$

}

} else {

cdbLen = O_i ;

if (N==4) {

cdbLen = 32;

}

for (m=0; m< O_i ; ++m) {

```

        TmpVVec[m] = 0;
        for (j=0; j< NumVVecIndices; ++j) {
            TmpVVec[m] += WeightVal[j] * VecDict[cdbLen].[VVecIdx[j]][m];
        }
    }
    FNorm = 0.0;
    for (m=0; m<O; ++m) {
        FNorm += TmpVVec[m] * TmpVVec[m];
    }
    FNorm = (N+1)/sqrt(FNorm);
    for (m=0; m< VVecLengthUsed; ++m) {
        idx = VVecCoeffIdxUsed[m];
         $v_{idx}^{(i)}(k) = \text{TmpVVec}[idx] * \text{FNorm};$ 
    }
[0260] }
}
elseif (NbitsQ(k)[i] == 5) {
    for (m=0; m< VVecLengthUsed; ++m) {
         $v_{VVecCoeffIdxUsed[m]}^{(i)}(k) = (N+1) * aVal[i][m];$ 
    }
}
elseif (NbitsQ(k)[i] >= 6) {
    for (m=0; m< VVecLengthUsed; ++m) {
         $v_{VVecCoeffIdxUsed[m]}^{(i)}(k) = (N+1) * (2^{(16 - NbitsQ(k)[i])} * aVal[i][m]) / 2^{15};$ 
        if (PFlag(k)[i] == 1) {
             $v_{VVecCoeffIdxUsed[m]}^{(i)}(k) += v_{VVecCoeffIdxUsed[m]}^{(i)}(k - 1);$ 
        }
    }
}
}

```

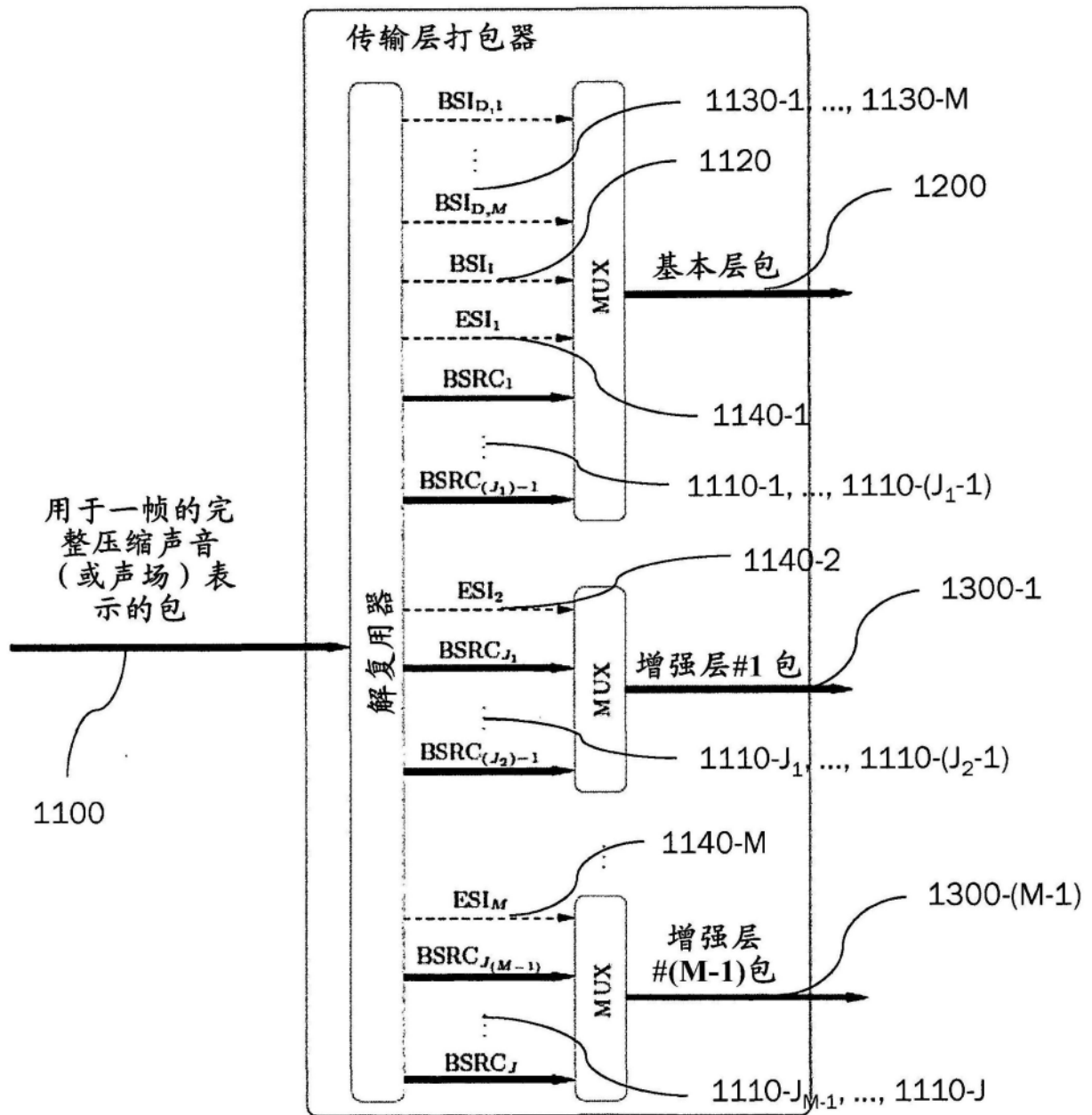


图1

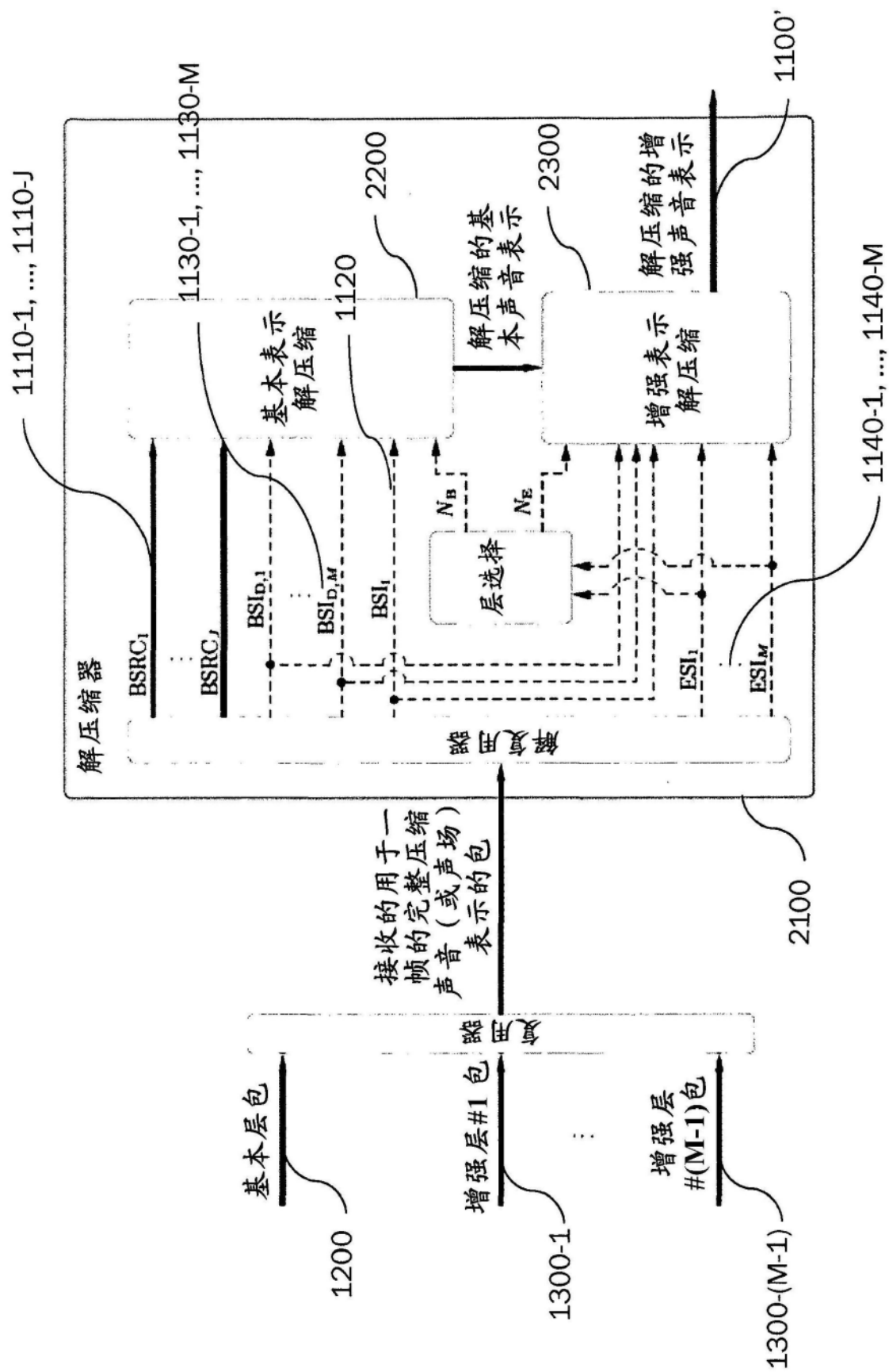


图2

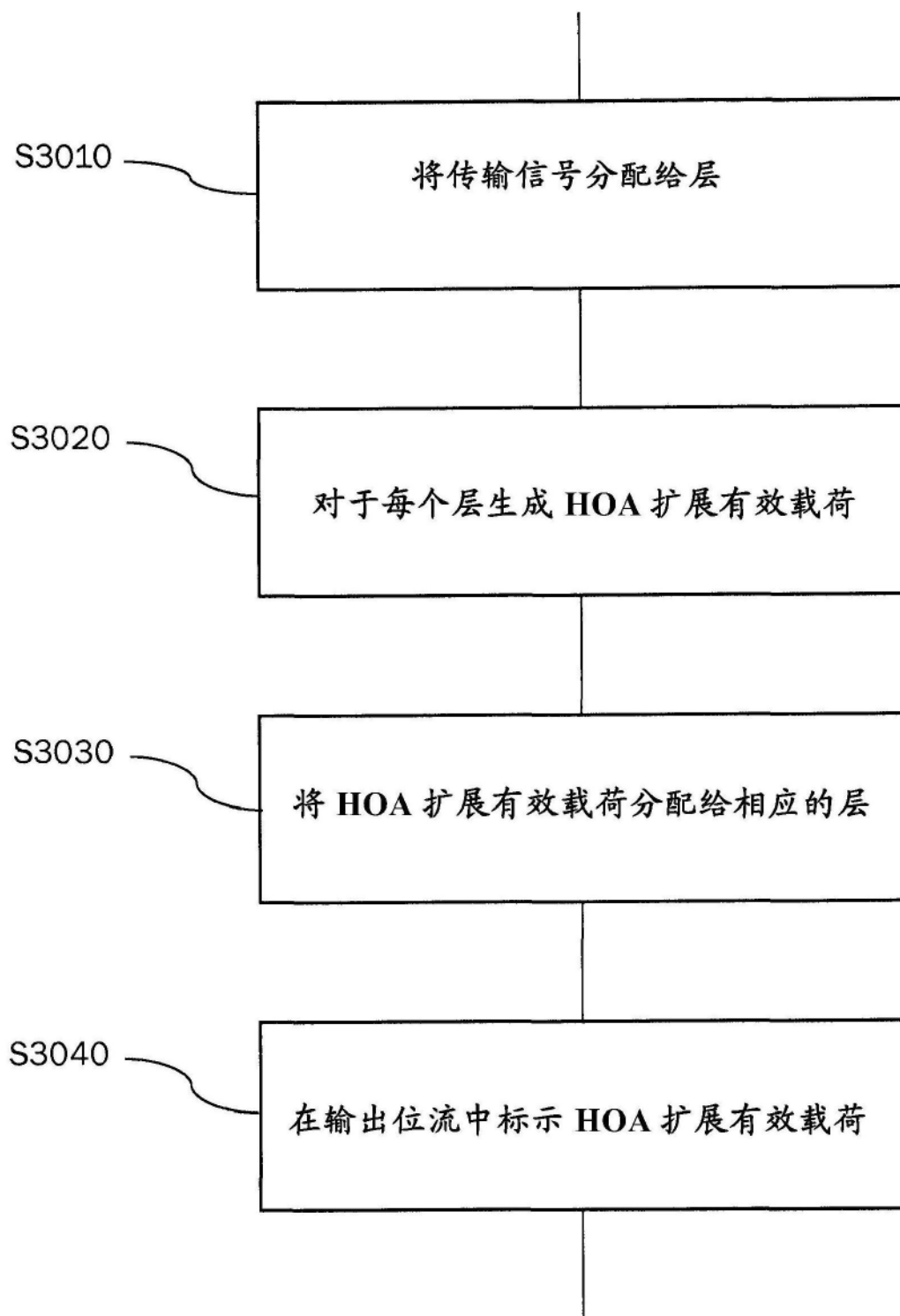


图3

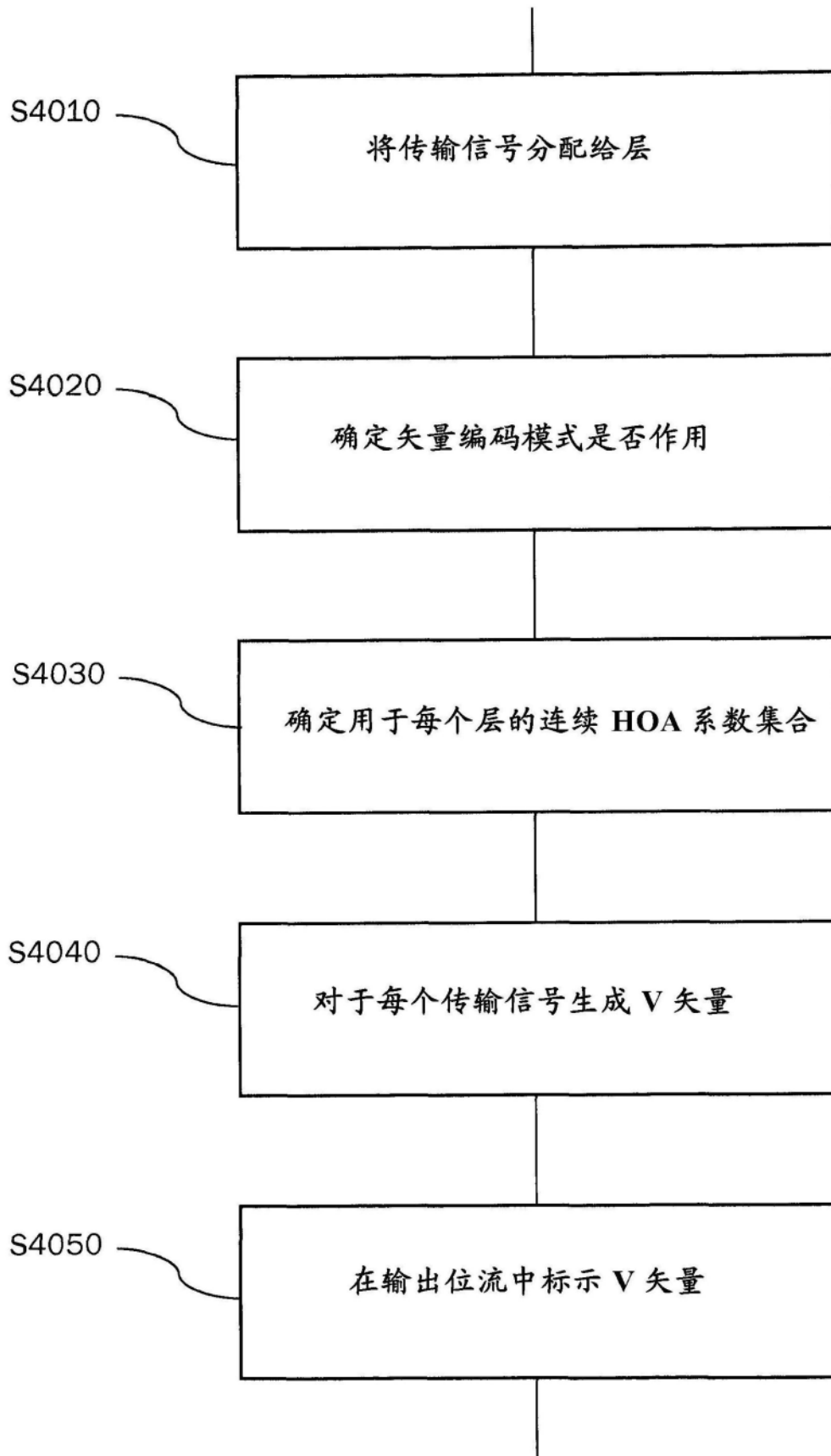


图4

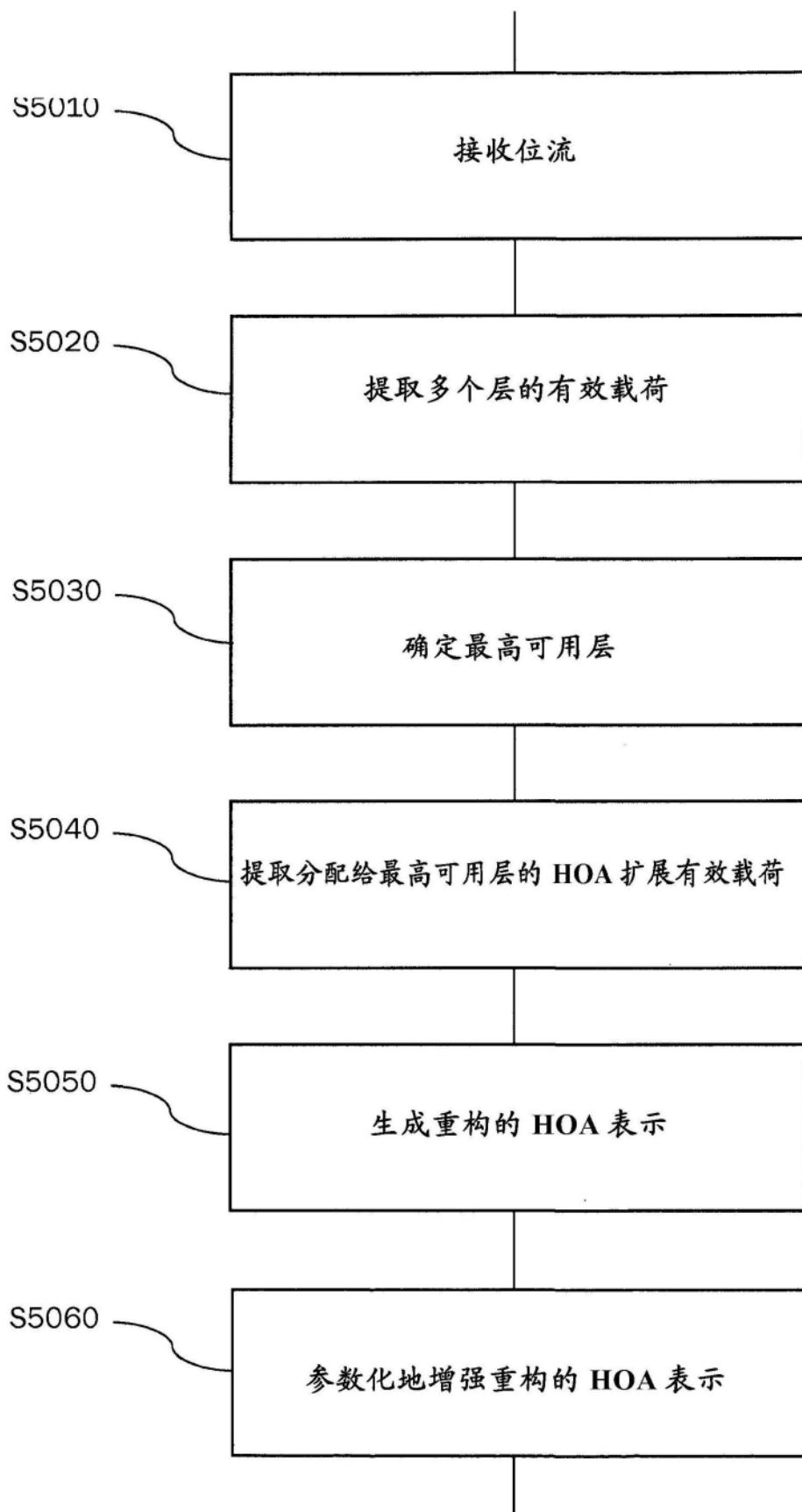


图5

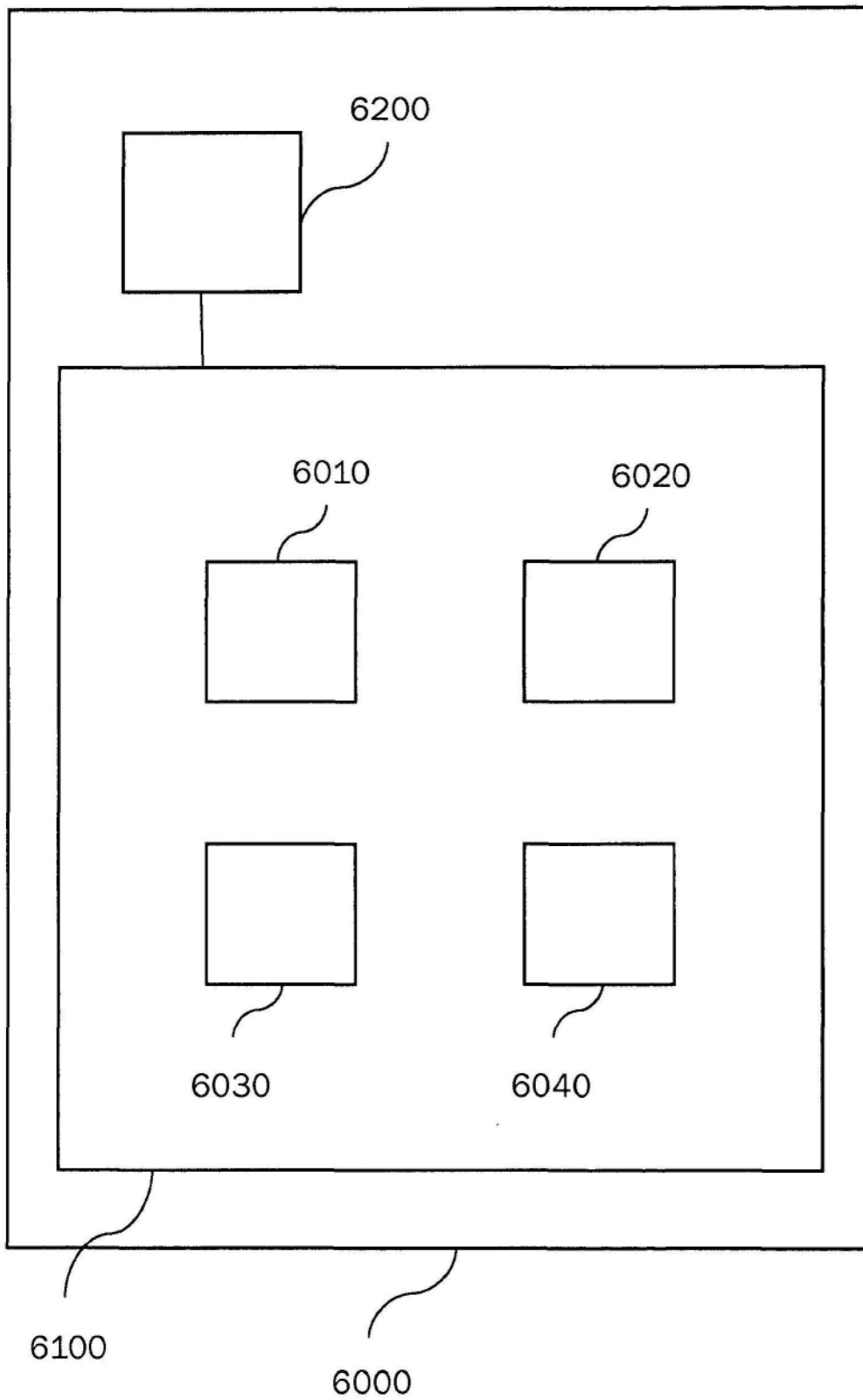


图6

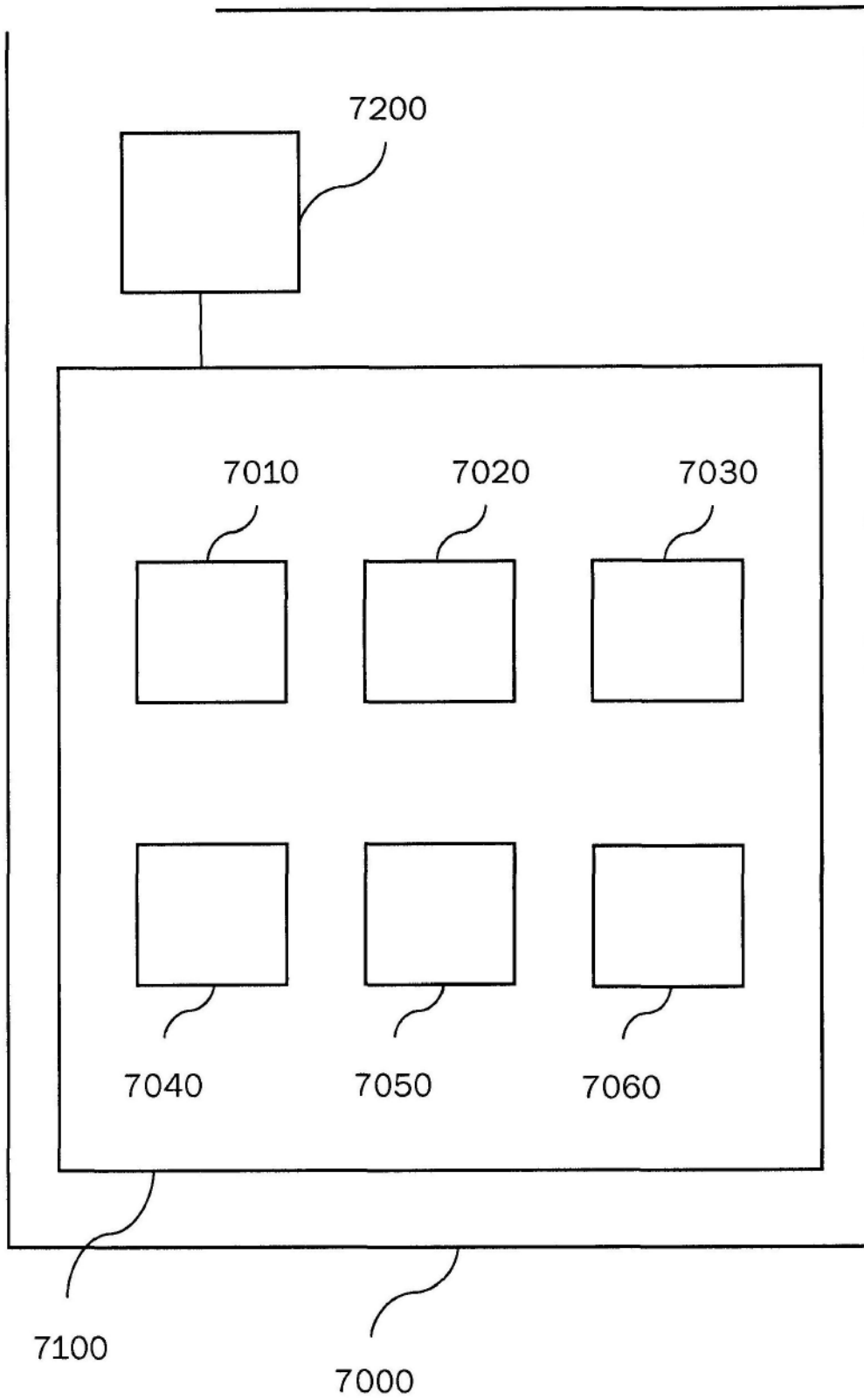


图7