



(10) 授权公告号 CN 111883148 B

(45) 授权公告日 2024. 08. 02

(21) 申请号 202010303989.9

(22) 申请日 2014.07.16

(65) 同一申请的已公布的文献号
申请公布号 CN 111883148 A

(43) 申请公布日 2020.11.03

(30) 优先权数据

EP13177365 2013.07.22 EP

EP13177367 2013.07.22 EP

EP13177378 2013.07.22 EP

EP13189279 2013.10.18 EP

(62) 分案原申请数据

201480041461.1 2014.07.16

(73) 专利权人 弗朗霍夫应用科学研究促进协会
地址 德国慕尼黑

(72) 发明人 克里斯蒂安·鲍斯

克里斯蒂安·埃特尔

约翰内斯·希勒佩特

(74) 专利代理机构 北京清亦华知识产权代理事
务所(普通合伙) 11201
专利代理师 宋融冰

(51) Int.Cl.

G10L 19/008 (2013.01)

G10L 19/16 (2013.01)

G10L 19/005 (2013.01)

G10L 19/02 (2013.01)

H04S 3/00 (2006.01)

H04S 3/02 (2006.01)

(56) 对比文件

W0 2013006325 A1, 2013.01.10

审查员 陈宗琪

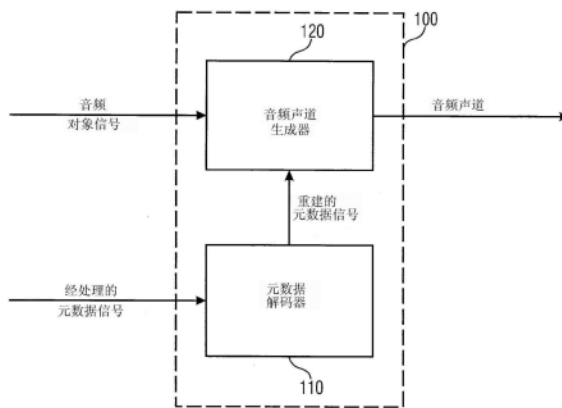
权利要求书4页 说明书22页 附图17页

(54) 发明名称

用于低延迟对象元数据编码的装置及方法

(57) 摘要

提供一种用于生成一个或多个音频声道的装置。该装置包括：根据控制信号从一个或多个经处理的元数据信号生成一个或多个重建的元数据信号的元数据解码器，其中一个或多个重建的元数据信号中的每个指示与一个或多个音频对象信号中的音频对象信号相关联的信息，其中元数据解码器用于通过确定用于一个或多个重建的元数据信号中的每个的多个重建的元数据样本，生成一个或多个重建的元数据信号。此外该装置包括用于根据一个或多个音频对象信号以及根据一个或多个重建的元数据信号生成一个或多个音频声道的音频声道生成器。



1. 一种用于生成一个或多个重建的元数据信号的装置(100), 其中所述装置包括:

元数据解码器(110;901), 用于根据控制信号从一个或多个经处理的元数据信号生成一个或多个重建的元数据信号, 其中所述一个或多个重建的元数据信号中的每个指示与一个或多个音频对象信号中的音频对象信号相关联的信息, 其中所述元数据解码器(110;901) 用于通过确定用于所述一个或多个重建的元数据信号中的每个的多个重建的元数据样本以生成所述一个或多个重建的元数据信号,

其中所述元数据解码器(110;901) 用于接收所述一个或多个经处理的元数据信号中的每个的多个经处理的元数据样本,

其中所述元数据解码器(110;901) 用于接收所述控制信号,

其中所述元数据解码器(110;901) 用于确定所述一个或多个重建的元数据信号中的每个重建的元数据信号的所述多个重建的元数据样本中的每个重建的元数据样本, 以使得当所述控制信号指示第一状态时, 所述重建的元数据样本为所述一个或多个经处理的元数据信号中的一个的经处理的元数据样本中的一个与所述重建的元数据信号的另一个已生成的重建的元数据样本的和, 并使得当所述控制信号指示不同于所述第一状态的第二状态时, 所述重建的元数据样本为所述一个或多个经处理的元数据信号中的所述一个的经处理的元数据样本中的所述一个。

2. 根据权利要求1所述的装置(100),

其中所述元数据解码器(110;901) 用于接收所述经处理的元数据信号中的两个或更多, 并用于生成所述重建的元数据信号中的两个或更多,

其中所述元数据解码器(110;901) 包括两个或更多元数据解码器子单元(911, ..., 91N),

其中所述两个或更多元数据解码器子单元(911, ..., 91N) 中的每个(91i;91i') 被配置为包括加法器(910) 和选择器(930),

其中所述两个或更多元数据解码器子单元(911, ..., 91N) 中的每个(91i;91i') 用于接收所述两个或更多经处理的元数据信号中的一个的所述多个经处理的元数据样本并用于生成所述两个或更多重建的元数据信号中的一个,

其中所述元数据解码器子单元(91i;91i') 的所述加法器(910) 用于将所述两个或更多经处理的元数据信号中的所述一个的所述经处理的元数据样本中的一个与所述两个或更多重建的元数据信号中的所述一个的另一个已生成的重建的元数据样本相加, 以获得总和值, 以及

其中所述元数据解码器子单元(91i;91i') 的所述选择器(930) 用于接收所述经处理的元数据样本中的一个、所述总和值以及所述控制信号, 且其中所述选择器(930) 用于确定所述重建的元数据信号的所述多个元数据样本中的一个, 以使得当所述控制信号指示所述第一状态时, 所述重建的元数据样本为所述总和值, 并使得当所述控制信号指示所述第二状态时, 所述重建的元数据样本为所述经处理的元数据样本中的所述一个。

3. 根据权利要求1所述的装置(100),

其中所述一个或多个重建的元数据信号中的至少一个指示所述一个或多个音频对象信号中的一个的位置信息。

4. 根据权利要求1所述的装置(100),

其中所述一个或多个重建的元数据信号中的至少一个指示所述一个或多个音频对象信号中的一个的音量。

5. 一种用于生成包括一个或多个编码的音频信号和一个或多个经处理的元数据信号的编码的音频信息的装置 (250), 其中所述装置包括:

元数据编码器 (210; 801; 802), 用于接收一个或多个原始元数据信号, 以及用于确定所述一个或多个经处理的元数据信号, 其中所述一个或多个原始元数据信号中的每个包括多个原始元数据样本, 其中所述一个或多个原始元数据信号中的每个的所述原始元数据样本指示与一个或多个音频对象信号中的音频对象信号相关联的信息,

其中所述元数据编码器 (210; 801; 802) 用于确定所述一个或多个经处理的元数据信号中的每个经处理的元数据信号的多个经处理的元数据样本中的每个经处理的元数据样本, 以使得当控制信号指示第一状态时, 所述经处理的元数据样本指示所述一个或多个原始元数据信号中的一个的多个原始元数据样本中的一个与所述经处理的元数据信号的另一个已生成的经处理的元数据样本之间的差值; 并使得当所述控制信号指示不同于所述第一状态的所述第二状态时, 所述经处理的元数据样本为所述一个或多个原始元数据信号中的所述一个的所述原始元数据样本中的所述一个或为所述原始元数据样本中的所述一个的量化表示。

6. 根据权利要求5所述的装置 (250),

其中所述元数据编码器 (210; 801; 802) 用于接收所述原始元数据信号中的两个或更多, 以及用于生成所述经处理的元数据信号中的两个或更多,

其中所述元数据编码器 (210; 801; 802) 包括两个或更多DPCM编码器 (811, ..., 81N),

其中所述两个或更多DPCM编码器 (811, ..., 81N) 中的每个用于确定所述两个或更多原始元数据信号中的一个的所述原始元数据样本中的一个与所述两个或更多经处理的元数据信号中的一个的另一个已生成的经处理的元数据样本之间的差值, 以获得差值样本, 以及

其中所述元数据编码器 (210; 801; 802) 还包括选择器 (830), 所述选择器 (830) 用于确定所述经处理的元数据信号的所述多个经处理的元数据样本中的一个, 以使得当所述控制信号指示所述第一状态时, 所述经处理的元数据样本为所述差值样本, 并使得当所述控制信号指示所述第二状态时, 所述经处理的元数据样本为所述原始元数据样本中的所述一个或为所述原始元数据样本中的所述一个的量化表示。

7. 根据权利要求5所述的装置 (250),

其中所述一个或多个原始元数据信号中的至少一个指示所述一个或多个音频对象信号中的一个的位置信息, 以及

其中所述元数据编码器 (210; 801; 802) 用于根据指示所述位置信息的所述一个或多个原始元数据信号中的至少一个, 生成所述一个或多个经处理的元数据信号中的至少一个。

8. 根据权利要求5所述的装置 (250),

其中所述一个或多个原始元数据信号中的至少一个指示所述一个或多个音频对象信号中的一个的音量, 以及

其中所述元数据编码器 (210; 801; 802) 用于根据指示所述音量的所述一个或多个原始元数据信号中的至少一个, 生成所述一个或多个经处理的元数据信号中的至少一个。

9. 根据权利要求5所述的装置(250),

其中所述元数据编码器(210;801;802)用于在所述控制信号指示所述第一状态时,利用第一数目的比特数对所述一个或多个经处理的元数据信号中的一个的所述经处理的元数据样本中的每个进行编码;在所述控制信号指示所述第二状态时,利用第二数目的比特数对所述一个或多个经处理的元数据信号中的一个的所述经处理的元数据样本中的每个进行编码;其中所述第一数目的比特数小于所述第二数目的比特数。

10. 一种音频系统,其包括:

根据权利要求6所述的装置(250),所述装置(250)用于生成一个或多个经处理的元数据信号,以及

装置(100),所述装置(100)用于根据所述一个或多个经处理的元数据信号生成一个或多个重建的元数据信号,

其中所述装置包括:

元数据解码器,用于根据控制信号从一个或多个经处理的元数据信号中生成一个或多个重建的元数据信号,其中所述一个或多个重建的元数据信号中的每个指示与一个或多个音频对象信号中的音频对象信号相关联的信息,其中所述元数据解码器用于通过确定用于所述一个或多个重建的元数据信号中的每个的多个重建的元数据样本,以生成所述一个或多个重建的元数据信号,

其中所述元数据解码器用于接收所述一个或多个经处理的元数据信号中的每个的多个经处理的元数据样本,

其中所述元数据解码器用于接收所述控制信号,

其中所述元数据解码器用于确定所述一个或多个重建的元数据信号中的每个重建的元数据信号的所述多个重建的元数据样本中的每个重建的元数据样本,以使得当所述控制信号指示第一状态时,所述重建的元数据样本为所述一个或多个经处理的元数据信号中的一个的所述经处理的元数据样本中的一个与所述重建的元数据信号的另一个已生成的重建的元数据样本的和,并使得当所述控制信号指示不同于所述第一状态的所述第二状态时,所述重建的元数据样本为所述一个或多个经处理的元数据信号中的所述一个的所述经处理的元数据样本中的所述一个。

11. 一种用于生成一个或多个重建的元数据信号的方法,其中所述方法包括:

根据控制信号从一个或多个经处理的元数据信号生成一个或多个重建的元数据信号,其中所述一个或多个重建的元数据信号中的每个指示与一个或多个音频对象信号中的音频对象信号相关联的信息,其中通过确定用于所述一个或多个重建的元数据信号中的每个的多个重建的元数据样本以生成所述一个或多个重建的元数据信号,

其中通过接收所述一个或多个经处理的元数据信号中的每个的多个经处理的元数据样本,通过接收所述控制信号,以及通过确定所述一个或多个重建的元数据信号中的每个重建的元数据信号的所述多个重建的元数据样本中的每个重建的元数据样本,以执行生成所述一个或多个重建的元数据信号,以使得当所述控制信号指示第一状态时,所述重建的元数据样本为所述一个或多个经处理的元数据信号中的一个的经处理的元数据样本中的一个与所述重建的元数据信号的另一个已生成的重建的元数据样本的和,并使得当所述控制信号指示不同于所述第一状态的所述第二状态时,所述重建的元数据样本为所述一个或多个

经处理的元数据信号中的所述一个的经处理的元数据样本中的所述一个。

12. 一种用于生成一个或多个经处理的元数据信号的方法,其中所述方法包括:

接收一个或多个原始元数据信号,以及

确定所述一个或多个经处理的元数据信号,

其中所述一个或多个原始元数据信号中的每个包括多个原始元数据样本,其中所述一个或多个原始元数据信号中的每个的所述原始元数据样本指示与一个或多个音频对象信号中的音频对象信号相关联的信息,以及

其中确定所述一个或多个经处理的元数据信号包括:确定所述一个或多个经处理的元数据信号中的每个经处理的元数据信号的多个经处理的元数据样本中的每个经处理的元数据样本,以使得当控制信号指示第一状态时,所述经处理的元数据样本指示所述一个或多个原始元数据信号中的一个的多个原始元数据样本中的一个与所述经处理的元数据信号的另一个已生成的经处理的元数据样本之间的差值或量化差值,并使得当所述控制信号指示不同于所述第一状态的第二状态时,所述经处理的元数据样本为所述一个或多个原始元数据信号中的所述一个的所述原始元数据样本中的所述一个或为所述原始元数据样本中的所述一个的量化表示。

13. 一种非暂时性数字存储介质,具有存储于其上的计算机可读代码,用于在执行于计算机或信号处理器上时执行如权利要求11所述的方法。

14. 一种非暂时性数字存储介质,具有存储于其上的计算机可读代码,用于在执行于计算机或信号处理器上时执行如权利要求12所述的方法。

用于低延迟对象元数据编码的装置及方法

[0001] 本申请是申请人为弗朗霍夫应用科学研究促进协会、申请日为2014年7月16日、申请号为201480041461.1、发明名称为“用于低延迟对象元数据编码的装置及方法”的分案申请。

技术领域

[0002] 本发明涉及音频编码/解码,特别涉及空间音频编码和空间音频对象编码,更特别涉及用于有效的对象元数据编码的装置和方法。

背景技术

[0003] 在本技术领域,空间音频编码工具是众所周知的并且,例如,在环绕MPEG标准中已被标准化。空间音频编码从诸如在再现装备(setup)中通过其布置而识别的五个或七个声道(即左声道、中间声道、右声道、左环绕声道、右环绕声道以及低频增强声道)的原始输入声道开始。空间音频编码器通常从原始声道得到一个或多个降混合声道,以及此外,得到关于空间线索(cues)的参数化数据,例如在声道相干数值中的声道间水平差异、声道间相位差异、声道间时间差异等等。一个或多个降混合声道与指示空间线索的参数化边信息一起被传输至空间音频解码器,该空间音频解码器对降混合声道和相关联的参数化数据进行解码以最终获得输出声道,该输出声道为原始输入声道的近似版本。声道在输出装备中的布置通常为固定的且为,例如5.1声道格式或7.1声道格式等。

[0004] 此种基于声道的音频格式被广泛用于储存或传输多声道音频内容,其中每个声道涉及在给定位置处的特定扬声器。这些种类的格式的忠实再现需要扬声器装备,其中扬声器放置在与音频信号产生期间所使用的扬声器相同的位置。而增加扬声器数量可改进真正拟真的三维音频场景的再现,但是达到此要求变得越来越困难,尤其是在诸如客厅的家庭环境中。

[0005] 通过基于对象的方法可以克服对特定扬声器装备的需求,在基于对象的方法中特别针对播放装备渲染扬声器信号。

[0006] 例如,空间音频对象编码工具在此技术领域是众所周知的且在MPEG SAOC(SAOC=空间音频对象编码)标准中被标准化。相比于空间音频编码从原始声道开始,空间音频对象编码从非自动专用于特定渲染再现装备的音频对象开始。另外,音频对象在再现场景中的布置是灵活的,且可由用户通过将特定渲染信息输入至空间音频对象编码解码器来确定。可选地或此外,渲染信息,即在再现装备中特定音频对象通常随时间待被放置的位置处的信息,可被传输作为附加的边信息或元数据。为了获得特定的数据压缩,通过SAOC编码器对多个音频对象进行编码,SAOC编码器通过根据特定的降混合信息对对象进行降混合以从输入对象计算一个或多个传输声道。此外,SAOC编码器计算表示对象间线索的参数化边信息,例如对象水平差异(OLD)、对象相干数值等等。当在空间音频编码(SAC)中,针对个别的时间/频率平铺(time/frequency tiles)(即,针对包括例如1024或2048个样本的音频信号的特定帧,24、32或64等)计算对象间参数化数据,考虑频带以使得最后对于每个帧和每

个频带都存在参数化数据。作为示例,当音频片具有20个帧且每个帧被细分成32个频带时,时间/频率平铺的数量为640。

[0007] 在基于对象的方法中,通过离散音频对象描述声场。此需要对象元数据,其描述每个声源在3D空间中的时变位置。

[0008] 现有技术中的第一元数据编码概念为空间声音描述交换格式(SpatDIF),尚在发展中的音频场景描述格式[1]。音频场景描述格式被设计为用于基于对象的声音场景的交换格式,且其并没有提供任何用于对象轨迹的压缩方法。SpatDIF使用基于文本的开放式声音控制(OSC)格式以构造对象元数据[2]。然而,简单的基于文本的表示并不是用于对象轨迹的经压缩的传输的选项。

[0009] 现有技术中的另一个元数据概念为音频场景描述格式(ASDF)[3],其和基于文本的解决方案具有相同的缺点。由同步多媒体集成语言(SMIL)的扩展构造数据,该同步多媒体集成语言(SMIL)为可扩展标记语言(XML)[4,5]的子集。

[0010] 现有技术中的另一个元数据概念为用于场景的音频二进制格式(AudioBIFS),二进制格式是MPEG-4规范[6,7]的一部分。它与基于XML的虚拟现实建模语言(VRML)密切相关,基于XML的虚拟现实建模语言被开发用于音频虚拟3D场景的描述和交互式虚拟现实应用[8]。复杂的AudioBIFS规范使用场景图以指定对象移动的路径。AudioBIFS的主要缺点在于,它并非被设计用于要求有限的系统延迟和随机访问数据流的实时操作。此外,对象位置的编码没有运用听者的有限定位能力。对于音频虚拟场景中的固定的听者位置,可以利用较低数目的比特[9]来量化对象数据。因此,应用于AudioBIFS的对象元数据的编码对于数据压缩是无效的。

[0011] 因此,如果能提供改进的有效的对象元数据编码概念,将会获得高度的赞赏。

发明内容

[0012] 本发明的目的在于提供用于对象元数据编码的改进技术。

[0013] 提供一种用于生成一个或多个音频声道的装置,该装置包括:元数据解码器,用于根据控制信号(b)从一个或多个经处理的元数据信号(z_1, \dots, z_N)生成一个或多个重建的元数据信号(x'_1, \dots, x'_N),其中一个或多个重建的元数据信号(x'_1, \dots, x'_N)中的每个指示与一个或多个音频对象信号的音频对象信号相关联的信息,其中元数据解码器用于通过确定用于一个或多个重建的元数据信号(x'_1, \dots, x'_N)中的每个的多个重建的元数据样本($x'_1(n), \dots, x'_N(n)$)以生成一个或多个重建的元数据信号(x'_1, \dots, x'_N)。此外,该装置包括:音频声道生成器,用于根据一个或多个音频对象信号以及根据一个或多个重建的元数据信号(x'_1, \dots, x'_N)生成一个或多个音频声道。元数据解码器用于接收一个或多个经处理的元数据信号(z_1, \dots, z_N)中的每个的多个经处理的元数据样本($z_1(n), \dots, z_N(n)$)。此外,元数据解码器用于接收控制信号(b)。此外,元数据解码器用于确定一个或多个重建的元数据信号(x'_1, \dots, x'_N)中的每个重建的元数据信号(x'_i)的多个重建的元数据样本($x'_i(1), \dots, x'_i(n-1), x'_i(n)$)中的每个重建的元数据样本($x'_i(n)$),以使得当控制信号(b)指示第一状态($b(n)=0$)时,所述重建的元数据样本($x'_i(n)$)为一个或多个经处理的元数据信号中的一个(z_i)的经处理的元数据样本中的一个($z_i(n)$)与所述重建的元数据信号(x'_i)的另一个已生成的重建的元数据样本($x'_i(n-1)$)的和,并使得当控制信号指示不同于第一状态的第二状

态 ($b(n) = 1$) 时,所述重建的元数据样本 ($x_i'(n)$) 为一个或多个经处理的元数据信号 (z_1, \dots, z_N) 中的所述一个 (z_i) 的经处理的元数据样本 ($z_i(1), \dots, z_i(n)$) 中的所述一个 ($z_i(n)$)。

[0014] 此外,提供一种用于生成包括一个或多个编码的音频信号和一个或多个经处理的元数据信号的编码的音频信息的装置。此装置包括:元数据编码器,用于接收一个或多个原始元数据信号,以及用于确定一个或多个经处理的元数据信号,其中一个或多个原始元数据信号中的每个包括多个原始元数据样本,其中一个或多个原始元数据信号中的每个的原始元数据样本指示与一个或多个音频对象信号中的音频对象信号相关联的信息。

[0015] 此外,该装置包括:音频编码器,用于对一个或多个音频对象信号进行编码以获得一个或多个编码的音频信号。

[0016] 元数据编码器用于确定一个或多个经处理的元数据信号 (z_1, \dots, z_N) 中的每个经处理的元数据信号 (z_i) 的多个经处理的元数据样本 ($z_i(1), \dots, z_i(n-1), z_i(n)$) 中的每个经处理的元数据样本 ($z_i(n)$),以使得当控制信号 (b) 指示第一状态 ($b(n) = 0$) 时,所述重建的元数据样本 ($z_i(n)$) 指示一个或多个原始元数据信号中的一个 (x_i) 的多个原始元数据样本中的一个 ($x_i(n)$) 与所述经处理的元数据信号 (z_i) 的另一个已生成的经处理的元数据样本之间的差值或量化差值;并使得当控制信号指示不同于第一状态的第二状态 ($b(n) = 1$) 时,所述经处理的元数据样本 ($z_i(n)$) 为一个或多个经处理的元数据信号中的所述一个 (x_i) 的原始元数据样本 ($x_i(1), \dots, x_i(n)$) 中的所述一个 ($x_i(n)$) 或为原始元数据样本 ($x_i(1), \dots, x_i(n)$) 中的所述一个 ($x_i(n)$) 的量化表示 ($q_i(n)$)。

[0017] 根据实施例,提供用于对象元数据的数据压缩概念,其利用有限的数据速率实现用于多个传输声道的有效压缩机制。编码器和解码器没有引入额外的延迟。此外,可实现用于纯方位角变化(例如,照相机旋转)的良好的压缩速率。此外,所提供的概念支持不连续的轨迹,例如位置的跳跃。此外,实现了低解码复杂度。此外,实现了具有有限的重新初始化时间的随机访问。

[0018] 此外,提供一种用于生成一个或多个音频声道的方法,该方法包括:

[0019] -根据控制信号 (b) 从一个或多个经处理的元数据信号 (z_1, \dots, z_N) 中生成一个或多个重建的元数据信号 (x_1', \dots, x_N'), 其中一个或多个重建的元数据信号 (x_1', \dots, x_N') 中的每个指示与一个或多个音频对象信号中的音频对象信号相关联的信息,其中通过确定用于一个或多个重建的元数据信号 (x_1', \dots, x_N') 中的每个的多个重建的元数据样本 ($x_1'(n), \dots, x_N'(n)$), 以执行生成一个或多个重建的元数据信号 (x_1', \dots, x_N'); 以及

[0020] -根据一个或多个音频对象信号以及根据一个或多个重建的元数据信号 (x_1', \dots, x_N'), 生成一个或多个音频声道。

[0021] 通过接收一个或多个经处理的元数据信号 (z_1, \dots, z_N) 中的每个的多个经处理的元数据样本 ($z_1(n), \dots, z_N(n)$)、通过接收控制信号 (b) 以及通过确定一个或多个重建的元数据信号 (x_1', \dots, x_N') 中的每个重建的元数据信号 (x_i') 的多个重建的元数据样本 ($x_i'(1), \dots, x_i'(n-1), x_i'(n)$) 中的每个重建的元数据样本 ($x_i'(n)$), 以执行生成一个或多个重建的元数据信号 (x_1', \dots, x_N'), 以使得当控制信号 (b) 指示第一状态 ($b(n) = 0$) 时,所述重建的元数据样本 ($x_i'(n)$) 为一个或多个经处理的元数据信号中的一个 (z_i) 的经处理的元数据样本中的一个 ($z_i(n)$) 与所述重建的元数据信号 (x_i') 的另一个已生成的重建的元数据样本 (x_i'

(n-1))的和,并使得当控制信号指示不同于第一状态的第二状态($b(n)=1$)时,所述重建的元数据样本($x_i'(n)$)为一个或多个经处理的元数据信号(z_1, \dots, z_N)中的所述一个(z_i)的经处理的元数据样本($z_i(1), \dots, z_i(n)$)中的所述一个($z_i(n)$)。

[0022] 此外,提供一种用于生成包括一个或多个编码的音频信号和一个或多个经处理的元数据信号的编码的音频信息的方法,此方法包括:

[0023] -接收一个或多个原始元数据信号;

[0024] -确定一个或多个经处理的元数据信号;以及

[0025] -对一个或多个音频对象信号进行编码以获得一个或多个编码的音频信号。

[0026] 一个或多个原始元数据信号中的每个包括多个原始元数据样本,其中一个或多个原始元数据信号中的每个的原始元数据样本指示与一个或多个音频对象信号中的音频对象信号相关联的信息。确定一个或多个经处理的元数据信号包括:确定一个或多个经处理的元数据信号(z_1, \dots, z_N)中的每个经处理的元数据信号(z_i)的多个经处理的元数据样本($z_i(1), \dots, z_i(n-1), z_i(n)$)中的每个经处理的元数据样本($z_i(n)$),以使得当控制信号(b)指示第一状态($b(n)=0$)时,所述重建的元数据样本($z_i(n)$)指示一个或多个原始元数据信号中的一个(x_i)的多个原始元数据样本中的一个($x_i(n)$)与所述经处理的元数据信号(z_i)的另一个已生成的经处理的元数据样本之间的差值或量化差值,并使得当控制信号指示不同于第一状态的第二状态($b(n)=1$)时,所述经处理的元数据样本($z_i(n)$)为一个或多个经处理的元数据信号中的所述一个(x_i)的原始元数据样本($x_i(1), \dots, x_i(n)$)中的所述一个($x_i(n)$)或为原始元数据样本($x_i(1), \dots, x_i(n)$)中的所述一个($x_i(n)$)的量化表示($q_i(n)$)。

[0027] 此外,提供一种计算机程序,当其执行于计算机或信号处理器上时,其用于实现上述方法。

附图说明

[0028] 以下将参考附图详细地描述本发明的实施例,其中:

[0029] 图1示出根据实施例的用于生成一个或多个音频声道的装置;

[0030] 图2示出根据实施例的用于生成编码的音频信息的装置;

[0031] 图3示出根据实施例的系统;

[0032] 图4示出通过方位角、仰角以及半径表示的音频对象在三维空间中离原点的位置。

[0033] 图5示出音频声道生成器假定的音频对象和扬声器装备的位置;

[0034] 图6示出差分脉冲码调制编码器;

[0035] 图7示出差分脉冲码调制解码器;

[0036] 图8A示出根据实施例的元数据编码器;

[0037] 图8B示出根据另一实施例的元数据编码器;

[0038] 图9A示出根据实施例的元数据解码器;

[0039] 图9B示出根据实施例的元数据解码器子单元;

[0040] 图10示出3D音频编码器的第一实施例;

[0041] 图11示出3D音频解码器的第一实施例;

[0042] 图12示出3D音频编码器的第二实施例;

[0043] 图13示出3D音频解码器的第二实施例;

[0044] 图14示出3D音频编码器的第三实施例;以及

[0045] 图15示出3D音频解码器的第三实施例。

具体实施方式

[0046] 图2示出根据实施例的用于生成编码的音频信息的装置250,该编码的音频信息包括一个或多个编码的音频信号以及一个或多个经处理的元数据信号。

[0047] 装置250包括用于接收一个或多个原始元数据信号并用于确定一个或多个经处理的元数据信号的元数据编码器210,,其中一个或多个原始元数据信号中的每个包括多个原始元数据样本,其中一个或多个原始元数据信号中的每个的原始元数据样本指示与一个或多个音频对象信号中的音频对象信号相关联的信息。

[0048] 此外,装置250包括用于对一个或多个音频对象信号进行编码以获得一个或多个编码的音频信号的音频编码器220。

[0049] 元数据编码器210用于确定一个或多个经处理的元数据信号 (z_1, \dots, z_N) 中的每个经处理的元数据信号 (z_i) 的多个经处理的元数据样本 $(z_i(1), \dots, z_i(n-1), z_i(n))$ 中的每个经处理的元数据样本 $(z_i(n))$, 以使得当控制信号 (b) 指示第一状态 $(b(n)=0)$ 时,所述重建的元数据样本 $(z_i(n))$ 指示一个或多个原始元数据信号中的一个 (x_i) 的多个原始元数据样本中的一个 $(x_i(n))$ 与所述经处理的元数据信号 (z_i) 的另一个已生成的经处理的元数据样本之间的差值或量化差值;并使得当控制信号指示不同于第一状态的第二状态 $(b(n)=1)$ 时,所述经处理的元数据样本 $(z_i(n))$ 为一个或多个经处理的元数据信号中的所述一个 (x_i) 的原始元数据样本 $(x_i(1), \dots, x_i(n))$ 中的所述一个 $(x_i(n))$ 或为原始元数据样本 $(x_i(1), \dots, x_i(n))$ 中的所述一个 $(x_i(n))$ 的量化表示 $(q_i(n))$ 。

[0050] 图1示出根据实施例的用于生成一个或多个音频声道的装置100。

[0051] 装置100包括用于根据控制信号 (b) 从一个或多个经处理的元数据信号 (z_1, \dots, z_N) 生成一个或多个重建的元数据信号 (x_1', \dots, x_N') 的元数据解码器110,其中一个或多个重建的元数据信号 (x_1', \dots, x_N') 中的每个指示与一个或多个音频对象信号的音频对象信号相关联的信息,其中元数据解码器110用于通过确定用于一个或多个重建的元数据信号 (x_1', \dots, x_N') 中的每个的多个重建的元数据样本 $(x_1'(n), \dots, x_N'(n))$ 以生成一个或多个重建的元数据信号 (x_1', \dots, x_N') 。

[0052] 此外,装置100包括用于根据一个或多个音频对象信号以及根据一个或多个重建的元数据信号 (x_1', \dots, x_N') 生成一个或多个音频声道的音频声道生成器120。

[0053] 元数据解码器110用于接收一个或多个经处理的元数据信号 (z_1, \dots, z_N) 中的每个的多个经处理的元数据样本 $(z_1(n), \dots, z_N(n))$ 。此外,元数据解码器110用于接收控制信号 (b) 。

[0054] 此外,元数据解码器110用于确定一个或多个重建的元数据信号 (x_1', \dots, x_N') 中的每个重建的元数据信号 (x_i') 的多个重建的元数据样本 $(x_i'(1), \dots, x_i'(n-1), x_i'(n))$ 中的每个重建的元数据样本 $(x_i'(n))$, 以使得当控制信号 (b) 指示第一状态 $(b(n)=0)$ 时,所述重建的元数据样本 $(x_i'(n))$ 为一个或多个经处理的元数据信号中的一个 (z_i) 的经处理的元数据样本中的一个 $(z_i(n))$ 与所述重建的元数据信号 (x_i') 的另一个已生成的重建的元数据样本 $(x_i'(n-1))$ 的和,并使得当控制信号指示不同于第一状态的第二状态 $(b(n)=1)$ 时,所

述重建的元数据样本($x_i'(n)$)为一个或多个经处理的元数据信号(z_1, \dots, z_N)中的所述一个(z_i)的经处理的元数据样本($z_i(1), \dots, z_i(n)$)中的所述一个($z_i(n)$)。

[0055] 当提及元数据样本时,应当注意的是,元数据样本的特征在于其元数据样本值以及与其相关的时间点。例如,此时间点可与音序列或其类似的起始相关。例如,索引 n 或 k 可识别元数据信号中的元数据样本的位置,并借此指示出(相关的)时间点(与起始时间相关)。应当注意的是,当两个元数据样本与不同的时间点相关时,即使它们的元数据样本值是相同的(有时可能会出现这样的情况),该两个元数据样本也是不同的元数据样本。

[0056] 上述实施例基于此发现:与音频对象信号相关联的(由元数据信号包括的)元数据信息常常缓慢地改变。

[0057] 例如,元数据信号可指示音频对象的位置信息(例如,定义音频对象的位置的方位角、仰角或半径)。可以假设,在大部分时间,音频对象的位置不会改变或仅缓慢地改变。

[0058] 或,元数据信号可以,例如指示音频对象的音量(例如,增益),并且也可以假设,在大部分时间,音频对象的音量缓慢地改变。

[0059] 基于此原因,无需在每个时间点传输(完整的)元数据信息。

[0060] 相反地,根据一些实施例,例如,可以仅在特定时间点传输(完整的)元数据信息,例如周期性地,如在每第 N 个时间点,如在时间点 $0, N, 2N, 3N$ 等。

[0061] 例如,在实施例中,三个元数据信号指定音频对象在3D空间中的位置。元数据信号中的第一个可以,例如指定音频对象的位置的方位角。元数据信号中的第二个可以,例如指定音频对象的位置的仰角。元数据信号中的第三个可以,例如指定关于音频对象的距离的半径。

[0062] 方位角、仰角以及半径明确地定义出音频对象在3D空间中离原点的位置,将参考图4示出此。

[0063] 图4示出通过方位角、仰角以及半径表示的音频对象在三维(3D)空间中离原点400的位置410。

[0064] 仰角指定,例如从原点到对象位置的直线与此直线在 xy 平面(由 x 轴和 y 轴定义的平面)上的正交投影之间的角度。方位角定义,例如 x 轴与所述正交投影之间的角度。通过指定方位角和仰角,可定义出通过原点400和音频对象的位置410的直线415。通过更进一步地指定半径,可定义出音频对象的精确位置410。

[0065] 在实施例中,方位角的范围被定义为: $-180^\circ < \text{方位角} \leq 180^\circ$,仰角的范围被定义为: $-90^\circ \leq \text{仰角} \leq 90^\circ$,半径可以,例如被定义为以米[m](大于或等于0m)为单位。

[0066] 在另一实施例中,例如,可假设,在 xyz 坐标系中的音频对象位置的所有 x 值都大于或等于零,方位角的范围可被定义为: $-90^\circ \leq \text{方位角} \leq 90^\circ$,仰角的范围可被定义为: $-90^\circ \leq \text{仰角} \leq 90^\circ$,以及半径可以,例如被定义为以米[m]为单位。

[0067] 在另一实施例中,可调整元数据信号以使得方位角的范围被定义为: $-128^\circ < \text{方位角} \leq 128^\circ$ 、仰角的范围被定义为: $-32^\circ \leq \text{仰角} \leq 32^\circ$ 以及半径可以,例如被定义在对数标度上。在一些实施例中,原始元数据信号、经处理的元数据信号以及重建的元数据信号分别可以包括一个或多个音频对象信号中的一个的位置信息的缩放表示和/或音量的缩放表示。

[0068] 音频声道生成器120可以,例如用于根据一个或多个音频对象信号以及根据重建的元数据信号生成一个或多个音频声道,其中重建的元数据信号可以,例如指示音频对象

的位置。

[0069] 图5示出音频声道生成器假设的音频对象和扬声器装备的位置。示出xyz坐标系的原点500。此外,示出第一音频对象的位置510和第二音频对象的位置520。此外,图5示出音频声道生成器120为四个扬声器生成四个音频声道的方案。音频声道生成器120假设四个扬声器511、512、513及514位于图5所示的位置处。

[0070] 在图5中,第一音频对象位于接近于扬声器511和512的假定位置的位置510处,并远离扬声器513和514。因此,音频声道生成器120可生成四个音频声道,以使得第一音频对象510由扬声器511和512而不由扬声器513和514再现。

[0071] 在其他实施例中,音频声道生成器120可生成四个音频声道,以使得第一音频对象510由扬声器511和512以高音量再现,并由扬声器513和514以低音量再现。

[0072] 此外,第二音频对象位于接近于扬声器513和514的假定位置的位置520处,并远离扬声器511和512。因此,音频声道生成器120可生成四个音频声道,以使得第二音频对象520由扬声器513和514而不由扬声器511和512再现。

[0073] 在其他实施例中,音频声道生成器120可生成四个音频声道,以使得第二音频对象520由扬声器513和514以高音量再现,并由扬声器511和512以低音量再现。

[0074] 在可选实施例中,仅使用两个元数据信号指定音频对象的位置。例如,当假设所有音频对象位于单个平面内时,例如可以仅指定方位角和半径。

[0075] 在其他实施例中,对于每个音频对象,仅将单个元数据信号编码并传输作为位置信息。例如,仅将方位角指定为音频对象的位置信息(例如可假设,所有音频对象位于具有距中心点相同距离的相同平面,因此被假设为具有相同半径)。方位角信息可以,例如足以确定音频对象位于接近于左扬声器并远离右扬声器的位置。在此情况下,音频声道生成器120可以,例如生成一个或多个音频声道,以使得音频对象由左扬声器而不由右扬声器再现。

[0076] 例如,可以应用基于矢量的幅度平移(Vector Base Amplitude Panning,VBAP)以确定扬声器的音频声道中的每个内的音频对象信号的权重(例如,参见[11])。例如关于VBAP,假设音频对象与虚拟源相关。

[0077] 在实施例中,另一元数据信号可指定每个音频对象的音量,例如增益(例如,以分贝[dB]表示)。

[0078] 例如,在图5中,第一增益值可由用于位于位置510处的第一音频对象的其他元数据信号指定,第二增益值由用于位于位置520处的第二音频对象的另一其他元数据信号指定,其中第一增益值大于第二增益值。在此情况下,扬声器511和512可以再现第一音频对象,其再现第一音频对象的音量高于扬声器513和514再现第二音频对象的音量。

[0079] 实施例也假设,音频对象的此增益值常常缓慢地改变。因此,无需在每个时间点传输此元数据信息。相反地,仅在特定时间点传输元数据信息。在中间的时间点处,例如,可以使用被传输的在先元数据样本和随后元数据样本来近似元数据信息。例如,线性内插法可用于中间值的近似。例如,可以针对时间点近似音频对象中的每个的增益、方位角、仰角和/或半径,其中不传输此元数据。

[0080] 通过此方法,可以实现元数据的传输速率的可观的节省。

[0081] 图3示出根据实施例的系统。

[0082] 该系统包括如上所述的装置250,其用于生成包括一个或多个编码的音频信号和一个或多个经处理的元数据信号的编码的音频信息。

[0083] 此外,该系统包括如上所述的装置100,其用于接收一个或多个编码的音频信号和一个或多个经处理的元数据信号,并用于根据一个或多个编码的音频信号以及根据一个或多个经处理的元数据信号生成一个或多个音频声道。

[0084] 例如,当用于编码的装置250使用SAOC编码器对一个或多个音频对象进行编码时,通过应用根据现有技术的SAOC解码器,用于生成一个或多个音频声道的装置100可对一个或多个编码的音频信号进行解码,以获得一个或多个音频对象信号。

[0085] 实施例基于此发现,可以扩展差分脉冲码调制的概念,然后此扩展的概念适于对用于音频对象的元数据信号进行编码。

[0086] 差分脉冲码调变(DPCM)方法针对缓慢变化的时间信号而建立,其借由差分传输[10]通过量化和冗余减少不相关。图6中示出DPCM编码器。

[0087] 在图6的DPCM编码器中,输入信号 x 的实际输入样本 $x(n)$ 被馈入相减单元610。在相减单元的另一输入处,另一个数值被馈入相减单元。可以假设,此另一个数值为先前所接收的样本 $x(n-1)$,尽管量化错误或其他错误可能导致在另一输入处的值不完全等于先前的样本 $x(n-1)$ 。由于偏离 $x(n-1)$ 的此可能偏差,减法器的另一输入可被称作 $x^*(n-1)$ 。相减单元从 $x(n)$ 减去 $x^*(n-1)$ 以获得差值 $d(n)$ 。

[0088] 然后在量化器620中量化 $d(n)$ 以获得输出信号 y 的另一输出样本 $y(n)$ 。一般来说, $y(n)$ 等于 $d(n)$ 或为接近于 $d(n)$ 的值。

[0089] 此外, $y(n)$ 被馈入加法器630。此外, $x^*(n-1)$ 被馈入加法器630。用于 $d(n)$ 从减法 $d(n) = x(n) - x^*(n-1)$ 中得到,且 $y(n)$ 为等于或至少接近于 $d(n)$ 的值,加法器630的输出 $x^*(n)$ 等或或至少接近于 $x(n)$ 。

[0090] 在单元640内 $x^*(n)$ 被保留一个采样周期,然后继续处理下一个样本 $x(n+1)$ 。

[0091] 图7示出对应的DPCM解码器。

[0092] 在图7中,来自DPCM编码器的输出信号 y 的样本 $y(n)$ 被馈入加法器710。 $y(n)$ 表示将被重建的信号 $x(n)$ 的差值。在加法器710的另一输入处,先前所重建的样本 $x'(n-1)$ 被馈入加法器710。从加法 $x'(n) = x'(n-1) + y(n)$ 得到加法器的输出 $x'(n)$ 。由于 $x'(n-1)$ 大体等于或至少接近于 $x(n-1)$,且 $y(n)$ 大体等于或接近于 $x(n) - x(n-1)$,加法器710的输出 $x'(n)$ 大体等于或接近于 $x(n)$ 。

[0093] 在单元740内 $x'(n)$ 被保留一个采样周期,然后继续处理下一个样本 $y(n+1)$ 。

[0094] 当DPCM压缩方法实现大多数的先前阐述的所需特征时,它不允许随机访问。

[0095] 图8A示出根据实施例的元数据编码器801。

[0096] 图8A的元数据编码器801所应用的编码方法为典型的DPCM编码方法的扩展。

[0097] 图8A的元数据编码器801包括一个或多个DPCM编码器811, ..., 81N。例如,当元数据编码器801用于接收N个原始元数据信号时,元数据编码器801可以,例如正好包括N个DPCM编码器。在实施例中,如关于图6所描述地实现N个DPCM编码器中的每个。

[0098] 在实施例中,N个DPCM编码器中的每个用于接收N个原始元数据信号 x_1, \dots, x_N 中的一个的元数据样本 $x_i(n)$,并生成用于所述原始元数据信号 x_i 的元数据样本 $x_i(n)$ 中的每个的作为元数据差值信号 y_i 的差值样本 $y_i(n)$ 的差值,该差值被馈入所述DPCM编码器。在实施

例中,可以例如,如参考6图所述地执行生成差值样本 $y_i(n)$ 。

[0099] 图8A的元数据编码器801还包括选择器830(“A”),其用于接收控制信号 $b(n)$ 。

[0100] 此外,选择器830用于接收N个元数据差值信号 $y_1 \cdots y_N$ 。

[0101] 此外,在图8A的实施例中,元数据编码器801包括量化器820,其量化N个原始元数据信号 x_1, \cdots, x_N 以获得N个量化的元数据信号 q_1, \cdots, q_N 。在此实施例中,量化器可用于将N个量化的元数据信号馈入选择器830。

[0102] 选择器830可用于从量化的元数据信号 q_i 以及从取决于控制信号 $b(n)$ 的DPCM编码的差值元数据信号 y_i ,生成经处理的元数据信号 z_i 。

[0103] 例如,当控制信号 b 处于第一状态(例如, $b(n) = 0$)时,选择器830可用于输出元数据差值信号 y_i 的差值样本 $y_i(n)$ 作为经处理的元数据信号 z_i 的元数据样本 $z_i(n)$ 。

[0104] 当控制信号 b 处于不同于第一状态的第二状态(例如, $b(n) = 1$)时,选择器830可用于输出量化的元数据信号 q_i 的元数据样本 $q_i(n)$ 作为经处理的元数据信号 z_i 的元数据样本 $z_i(n)$ 。

[0105] 图8B示出根据另一实施例的元数据编码器802。

[0106] 在图8B的实施例中,元数据编码器802不包括量化器820,并将N个原始元数据信号 x_1, \cdots, x_N 而非N个量化的元数据信号 q_1, \cdots, q_N 直接地馈入选择器830。

[0107] 在此实施例中,例如,当控制信号 b 处于第一状态(例如, $b(n) = 0$)时,选择器830可用于输出元数据差值信号 y_i 的差值样本 $y_i(n)$ 作为经处理的元数据信号 z_i 的元数据样本 $z_i(n)$ 。

[0108] 当控制信号 b 处于不同于第一状态的第二状态(例如, $b(n) = 1$)时,选择器830可用于输出原始元数据信号 x_i 的元数据样本 $x_i(n)$ 作为经处理的元数据信号 z_i 的元数据样本 $z_i(n)$ 。

[0109] 图9A示出根据实施例的元数据解码器901。根据图9A的元数据编码器与图8A和图8B的元数据编码器相对应。

[0110] 图9A的元数据解码器901包括一个或多个元数据解码器子单元 $911, \cdots, 91N$ 。元数据解码器901用于接收一个或多个经处理的元数据信号 z_1, \cdots, z_N 。此外,元数据解码器901用于接收控制信号 b 。元数据解码器用于根据控制信号 b 从一个或多个经处理的元数据信号 z_1, \cdots, z_N 生成一个或多个重建的元数据信号 x'_1, \cdots, x'_N 。

[0111] 在实施例中,N个经处理的元数据信号 z_1, \cdots, z_N 中的每个被馈入元数据解码器子单元 $911, \cdots, 91N$ 中的不同者。此外,根据实施例,控制信号 b 被馈入元数据解码器子单元 $911, \cdots, 91N$ 中的每个。根据实施例,元数据解码器子单元 $911, \cdots, 91N$ 的数目等于元数据解码器901所接收的经处理的元数据信号 z_1, \cdots, z_N 的数目。

[0112] 图9B示出根据实施例的图9A的元数据解码器子单元 $911, \cdots, 91N$ 中的元数据解码器子单元(91i)。元数据解码器子单元91i用于针对单个经处理的元数据信号 z_i 进行解码。元数据解码器子单元91i包括选择器930(“B”)和加法器910。

[0113] 元数据解码器子单元91i用于根据控制信号 $b(n)$ 从所接收的经处理的元数据信号 z_i 生成重建的元数据信号 x'_i 。

[0114] 例如,其可被实现如下:

[0115] 重建的元数据信号 x'_i 的最后一个重建的元数据样本 $x'_i(n-1)$ 被馈入加法器910。

此外,经处理的元数据信号 z_i 的实际元数据样本 $z_i(n)$ 也被馈入加法器910。加法器用于将最后一个重建的元数据样本 $x_i'(n-1)$ 与实际元数据样本 $z_i(n)$ 相加以获得总和值 $s_i(n)$,并将该总和值馈入选择器930。

[0116] 此外,实际元数据样本 $z_i(n)$ 也被馈入加法器930。

[0117] 选择器用于根据控制信号 b 选择来自加法器910的总和值 $s_i(n)$ 或实际元数据样本 $z_i(n)$ 作为重建的元数据信号 $x_i'(n)$ 的实际元数据样本 $x_i'(n)$ 。

[0118] 例如,当控制信号 b 位于第一状态(例如, $b(n)=0$)时,控制信号 b 指示,实际元数据样本 $z_i(n)$ 为差值,故总和值 $s_i(n)$ 为重建的元数据信号 x_i' 的正确的实际元数据样本 $x_i'(n)$ 。当控制信号处于第一状态(当 $b(n)=0$)时,选择器830用于选择总和值 $s_i(n)$ 作为重建的元数据信号 x_i' 的实际元数据样本 $x_i'(n)$ 。

[0119] 当控制信号 b 处于不同于第一状态的第二状态(例如, $b(n)=1$)时,控制信号 b 指示,实际元数据样本 $z_i(n)$ 并非为差值,故实际元数据样本 $z_i(n)$ 为重建的元数据信号 x_i' 的正确的实际元数据样本 $x_i'(n)$ 。当控制信号 b 处于第二状态(当 $b(n)=1$)时,选择器830用于选择实际元数据样本 $z_i(n)$ 作为重建的元数据信号 x_i' 的实际元数据样本 $x_i'(n)$ 。

[0120] 根据实施例,元数据解码器子单元91i还包括单元920,该单元920用于在采样周期的持续时间内保留重建的元数据信号的实际元数据样本 $x_i'(n)$ 。在实施例中,此确保了当 $x_i'(n)$ 被生成时,所生成的 x' (n)不会被过早地反馈,以使得当 $z_i(n)$ 为差值时,实际上基于 $x_i'(n-1)$ 生成 $x_i'(n)$ 。

[0121] 在图9B的实施例中,选择器930可根据控制信号 $b(n)$ 从所接收的信号分量 $z_i(n)$ 以及延迟的输出分量(重建的元数据信号的已生成的元数据样本)与所接收的信号分量 $z_i(n)$ 的线性组合中生成元数据样本 $x_i'(n)$ 。

[0122] 以下,DPCM编码的信号被表示为 $y_i(n)$,且B的第二输入信号(和信号)被表示为 $s_i(n)$ 。对于仅取决于对应的输入分量的输出分量,编码器和解码器输出被给定如下:

[0123] $z_i(n) = A(x_i(n), v_i(n), b(n))$

[0124] $x_i'(n) = B(z_i(n), s_i(n), b(n))$

[0125] 根据上述的用于一般方法的实施例的解决方案使用 $b(n)$ 以在DPCM编码的信号与量化的输入信号之间切换。为简便起见,忽略时间索引 n ,则功能区块A及B被给定如下:

[0126] 在元数据编码器801和802中,选择器830(A)选择:

[0127] $A: z_i(x_i, y_i, b) = y_i$, 如果 $b=0$ (z_i 指示差值)

[0128] $A: z_i(x_i, y_i, b) = x_i$, 如果 $b=1$ (z_i 不指示差值)

[0129] 在元数据解码器子单元91i和91i'中,选择器930(B)选择:

[0130] $B: x_i'(z_i, s_i, b) = s_i$, 如果 $b=0$ (z_i 指示差值)

[0131] $B: x_i'(z_i, s_i, b) = z_i$, 如果 $b=1$ (z_i 不指示差值)

[0132] 每当 $b(n)$ 等于1时,这允许传输量化的输入信号,而每当 $b(n)$ 为0时,则允许传输DPCM信号。在后者的情况下,解码器变成DPCM解码器。

[0133] 当被应用于对象元数据的传输时,此机制被用于规则地传输未经压缩的对象位置,解码器可使用该机制用于随机访问。

[0134] 在优选的实施例中,用于对差值进行编码的比特数少于用于对元数据样本进行编码的比特的数目。这些实施例基于此发现,(例如, N 个)随后的元数据样本在大部分时间内

仅稍微的变化。例如,如果一种元数据样本被编码,如以8个比特,这些元数据样本可呈现256个差值中的一个。一般来说,由于(例如,N个)随后的元数据值的稍微改变,可认为仅以,例如5个比特,便足以对差值进行编码。因此,即使差值被传输,可减少传输的比特的数目。

[0135] 在实施例中,元数据编码器210用于在控制信号指示第一状态($b(n)=0$)时,利用第一数目的比特对一个或多个经处理的元数据信号(z_1, \dots, z_N)中的一个 $z_i()$ 的经处理的元数据样本($z_i(1), \dots, z_i(n)$)中的每个进行编码;在控制信号指示第二状态($b(n)=1$)时,利用第二数目的比特对一个或多个经处理的元数据信号(z_1, \dots, z_N)中的一个 $z_i()$ 的经处理的元数据样本($z_i(1), \dots, z_i(n)$)中的每个进行编码;其中第一数目的比特小于第二数目的比特。

[0136] 在优选实施例中,一个或多个差值被传输,并且利用比元数据样本中的每个较少的比特对一个或多个差值中的每个进行编码,其中差值中的每个为整数。

[0137] 根据实施例,元数据编码器110用于利用第一数目的比特对一个或多个经处理的元数据信号中的一个的元数据样本中的一个或多个进行编码,其中一个或多个经处理的元数据信号中的所述一个的元数据样本中的所述一个或多个中的每个指示整数。此外,元数据编码器(110)用于利用第二数目的比特对差值中的一个或多个进行编码,其中差值中的所述一个或多个中的每个指示整数,其中第二数目的比特小于第一数目的比特。

[0138] 例如,在实施例中,考虑元数据样本可以表示以8个比特编码的方位角,例如方位角可以为 $-90 \leq \text{方位角} \leq 90$ 之间的整数。因此,方位角可呈现181个不同的值。然而,如果可假设,(例如,N个)随后的方位角样本仅相差不多于,例如 ± 15 ,则5个比特($2^5=32$)可足以对差值进行编码。如果差值可被表示为整数,则确定差值自动地将待被传输的额外的值变换到适当的值域。

[0139] 例如,考虑第一音频对象的第一方位角值为 60° 且其随后的值在从 45° 至 75° 的范围内变化的情况。此外,考虑第二音频对象的第二方位角值为 -30° 且其随后的值在从 -45° 至 -15° 的范围内变化。通过确定用于第一音频对象的两个随后的值和用于第二音频对象的两个随后的值的差值,第二方位角值和第一方位角值的差值均介于 -15° 至 $+15^\circ$ 的值域内,从而使得5个比特足以对差值中的每个进行编码且使得对差值进行编码的比特序列对于第一方位角的差值和第二方位角的差值具有相同含义。

[0140] 以下,描述根据实施例的对象元数据帧和根据实施例的符号表示。

[0141] 编码的对象元数据在帧中传输。这些对象元数据帧可包含内编码的对象数据或动态对象数据,其中后者包含自最后一次传输的帧的改变。

[0142] 用于对象元数据帧的以下语法的一些或全部部分可以,例如被应用:

		比特数	助记符
[0143]	object_metadata()		
	{		
	has_intracoded_object_metadata;	1	bslbf

[0144]

```

    if (has_intracoded_object_metadata) {
        intracoded_object_metadata ();
    }
    else {
        dynamic_object_metadata();
    }
}

```

[0145] 以下,描述根据实施例的内编码的对象数据。

[0146] 通过内编码的对象数据 (“I-Frames”) 实现编码的对象元数据的随机访问,该内编码的对象数据 (“I-Frames”) 包含在规则网格 (例如,长度为1024的每32个帧) 上采样的量化值。这些I-Frames可以,例如具有以下语法,其中position_azimuth、position_elevation、position_radius以及gain_factor指定当前的量化值。

[0147]

	比特数	助记符
intracoded_object_metadata()		
{		
if (num_objects>1) {		
fixed_azimuth;	1	bslbf
if (fixed_azimuth) {		
default_azimuth;	8	tcimsbf
}		
else {		
common_azimuth;	1	bslbf
if (common_azimuth) {		
default_azimuth;	8	tcimsbf
}		
else {		
for (o=1:num_objects) {		
position_azimuth[o];	8	tcimsbf
}		
}		
fixed_elevation;	1	bslbf
if (fixed_azimuth) {		
default_elevation;	6	tcimsbf
}		
else {		
common_elevation;	1	bslbf
if (common_azimuth) {		
default_elevation;	6	tcimsbf
}		
}		

[0148]	else {		
	for (o=1:num_objects) {		
	position_azimuth[o];	6	tcimsbf
	}		
	}		
	fixed_radius;	1	bslbf
	if (fixed_azimuth) {		
	default_radius;	4	tcimsbf
	}		
	else {		
	common_radius;	1	bslbf
	if (common_azimuth) {		
	default_radius;	4	tcimsbf
	}		
	else {		
	for (o=1:num_objects) {		
	position_radius[o];	4	tcimsbf
	}		
	}		
	fixed_gain;	1	bslbf
[0149]	if (fixed_azimuth) {		
	default_gain;	7	tcimsbf
	}		
	else {		
	common_gain;	1	bslbf
	if (common_azimuth) {		
	default_gain;	7	tcimsbf
	}		
	else {		
	for (o=1:num_objects) {		
	gain_factor[o];	7	tcimsbf
	}		
	}		
	}		
	else {		
	position_azimuth;	8	tcimsbf
	position_elevation;	6	tcimsbf
	position_radius;	4	tcimsbf
	gain_factor;	7	tcimsbf
	}		
	}		

[0150] 以下,描述根据实施例的动态对象数据。

[0151] 例如,在动态对象帧中传输的DPCM数据可具有以下语法:

	比特数	助记符
dynamic_object_metadata()		
{		
flag_absolute;	1	bslbf
for (o=1:num_objects) {		
has_object_metadata;	1	bslbf
if (has_object_metadata) {		
single_dynamic_object_metadata(flag_absolute);		
}		
}		
}		

	比特数	助记符
single_dynamic_object_metadata (flag_absolute) {		
if (flag_absolute) {		
if (!fixed_azimuth*) {		
position_azimuth;	8	tcimsbf
}		
if (!fixed_elevation*) {		
position_elevation;	6	tcimsbf
}		
if (!fixed_radius*) {		
position_radius;	4	tcimsbf
}		
if (!fixed_gain*) {		
gain_factor;	7	tcimsbf
}		
}		
else {		
nbits;	3	uimsbf
if (!fixed_azimuth*) {		
flag_azimuth;	1	bslbf
if (flag_azimuth) {		
position_azimuth_difference ;	num_bits	tcimsbf
}		

[0154]	}		
	if (!fixed_elevation*) {		
	flag_elevation;	1	bslbf
	if (flag_elevation) {		
	position_elevation_difference ;	min(num_bits,7)	tcimsbf
	}		
	}		
	if (!fixed_radius*) {		
	flag_radius;	1	bslbf
	if (flag_radius) {		
	position_radius_difference ;	min(num_bits,5)	tcimsbf
	}		
	}		
	if (!fixed_gain*) {		
	flag_gain;	1	bslbf
	if (flag_gain) {		
	gain_factor_difference ;	min(num_bits,8)	tcimsbf
	}		
	}		
Note: num_bits = nbits + 2;			
Footnote *: Given by the preceding			
intracoded_object_data()-frame			

[0155] 特别的,在实施例中,以上宏指令可以,例如具有以下含义:

[0156] 根据实施例的object_data()的参数定义:

[0157] has_intracoded_object_metadata指示帧是否是内编码的或差分编码的。

[0158] 根据实施例的intracoded_object_metadata()的参数定义:

[0159] fixed_azimuth 指示方位角值是否对于所有对象为固定的且不在dynamic_object_metadata()中传输的旗标。

[0160] default_azimuth 定义固定或共同方位角的值。

[0161] common_azimuth 指示共同方位角是否用于所有对象。

[0162] position_azimuth 如果没有共同方位角值,则传输用于每个对象的值。

[0163] fixed_elevation 指示仰角值是否对于所有对象为固定的且不在dynamic_object_metadata()中传输的旗标。

[0164] default_elevation 定义固定或共同仰角的值。

[0165] common_elevation 指示共同仰角值是否用于所有对象。

[0166] position_elevation 如果没有共同仰角值,则传输用于每个对象的值。

[0167] fixed_radius 指示半径是否对于所有对象为固定的且不在dynamic_object_metadata()中传输的旗标。

[0168] default_radius 定义共同半径的值。

[0169] common_radius 指示共同半径值是否用于所有对象。

[0170] position_radius 如果没有共同半径值,则传输用于每个对象的值。

[0171] fixed_gain 指示增益因数是否对于所有对象为固定的且不在

- dynamic_object_metadata() 中传输的旗标。
- [0172] default_gain 定义固定或共同增益因数的值。
- [0173] common_gain 指示共同增益因数值是否用于所有对象。
- [0174] gain_factor 如果没有共同增益因数值,则传输用于每个对象的值。
- [0175] position_azimuth 如果仅存在一个对象,则此为其方位角。
- [0176] position_elevation 如果仅存在一个对象,则此为其仰角。
- [0177] position_radius 如果仅存在一个对象,则此为其半径。
- [0178] gain_factor 如果仅存在一个对象,则此为其增益因数。
- [0179] 根据实施例的dynamic_object_metadata()的参数的定义:
- [0180] flag_absolute 指示分量的值是否被差分地传输或以绝对值传输。
- [0181] has_object_metadata 指示是有有对象数据出现在比特流中。
- [0182] 根据实施例的single_dynamic_object_metadata()的参数的定义:
- [0183] position_azimuth 方位角的绝对值,如果值为非固定的。
- [0184] position_elevation 仰角的绝对值,如果值为非固定的。
- [0185] position_radius 半径的绝对值,如果值为非固定的。
- [0186] gain_factor 增益因数的绝对值,如果值为非固定的。
- [0187] nbits 需要多少比特来表示差值。
- [0188] flag_azimuth 指示方位角值是否改变的每个对象的旗标。
- [0189] position_azimuth_difference 在先值与活跃值之间的差值。
- [0190] flag_elevation 指示仰角值是否改变的每个对象的旗标。
- [0191] position_elevation_difference 在先值与活跃值之间的差值的值。
- [0192] flag_radius 指示半径是否改变的每个对象的旗标。
- [0193] position_radius_difference 在先值与活跃值之间的差值。
- [0194] flag_gain 指示增益半径是否改变的每个对象的旗标。
- [0195] gain_factor_difference 在先值与活跃值之间的差值。
- [0196] 在现有技术中,不存在一方面结合声道编码另一方面结合对象编码以便以低比特速率获得可接受的音频品质的灵活技术。
- [0197] 通过3D音频编解码器系统克服此限制。在此,描述3D音频编解码器系统。
- [0198] 图10示出根据本发明的实施例的3D音频编码器。该3D音频编码器用于对音频输入数据101进行编码以获得音频输出数据501。3D音频编码器包括输入接口,该输入接口用于接收由CH所指示的多个音频声道和由OBJ所指示的多个音频对象。此外,如图10所示,输入接口1100额外地接收与多个音频对象OBJ中的一个或多个相关的元数据。此外,3D音频编码器包括混合器200,该混合器200用于混合多个对象和多个声道以获得多个预混合的声道,其中每个预混合的声道包括声道的音频数据和至少一个对象的音频数据。
- [0199] 此外,3D音频编码器包括:核心编码器300,用于对核心编码器输入数据进行核心编码;以及元数据压缩器400,用于压缩与多个音频对象中的一个或多个相关的元数据。
- [0200] 此外,3D音频编码器可包括模式控制器600,其用于在一些操作模式中的一个下控制混合器、核心编码器和/或输出接口500,其中在第一模式下,核心编码器用于对多个音频声道以及由输入接口1100所接收的多个音频对象进行编码而未受到混合器的任何影响(即

没有经过混合器200的任意混合)。然而,在第二模式下混合器200是活跃的,核心编码器对多个混合的声道(即由区块200所生成的输出)进行编码。在后者的情况下,优选地,不再对任何对象数据进行编码。相反地,指示音频对象的位置的元数据已被混合器200用于将对象渲染至元数据所指示的声道上。换句话说,混合器200使用与多个音频对象相关的元数据以预渲染音频对象,然后预渲染的音频对象与声道混和以在混合器的输出处获得混合的声道。在此实施例中,可以不必传输任何对象,此也请求作为区块400的输出的经压缩的元数据。然而,如果并非输入至接口1100的所有对象都被混合而仅特定数量的对象被混合,则仅维持未被混合的对象以及相关联的元数据仍分别被传输至核心编码器300或元数据压缩器400。

[0201] 在图10中,元数据压缩器400为根据上述实施例中的一个的用于生成编码的音频信息的装置250的元数据编码器210。此外,在图10中,混合器200和核心编码器300一起形成根据上述实施例中的一个的用于生成编码的音频信息的装置250的音频编码器220。

[0202] 图12示出3D音频编码器的另一实施例,3D音频编码器额外地包括SAOC编码器800。该SAOC编码器800用于从空间音频对象编码器输入数据生成一个或多个传输声道和参数化数据。如图12所示,空间音频对象编码器输入数据为尚未经由预渲染器/混合器处理的对象。可选地,提供如在单独的声道/对象编码是活跃的模式一之下的预渲染器/混合器已被旁路,SAOC编码器800对输入至输入接口1100的所有对象进行编码。

[0203] 此外,如图12所示,优选地,核心编码器300被实现为USAC编码器,即作为如MPEG-USAC标准(USAC=联合语音和音频编码)中所定义和标准化的编码器。图12中示出的整个3D音频编码器的输出为具有用于单独的数据类型的容器状结构的MPEG 4数据流。此外,元数据被指示为“OAM”数据,且图10中的元数据压缩器400与OAM编码器400相对应,以获得输入至USAC编码器300的经压缩的OAM数据,如从图12中可看出的,USAC编码器300额外地包括输出接口,以获得具有编码的声道/对象数据和具有经压缩的OAM数据的MP4输出数据流。

[0204] 在图12中,OAM编码器400为根据上述实施例中的一个的用于生成编码的音频信息的装置250的元数据编码器210。此外,在图12中,SAOC编码器800和USAC编码器300一起形成根据上述实施例中的一个的用于生成编码的音频信息的装置250的音频编码器220。

[0205] 图14示出3D音频编码器的另一实施例,其中相对于图12,SAOC编码器可用于利用SAOC编码算法对在于此模式下不活跃的预渲染器/混合器200处所提供的声道进行编码,或,可选地,对加入对象的预渲染的声道进行SAOC编码。因此,在图14中,SAOC编码器800可对三种不同种类的输入数据进行操作,即不具有任何预渲染的对象的声道、声道和预渲染的对象,或独自的对象。此外,优选地,在图14中提供附加的OAM解码器420,以使得SAOC编码器800使用与在解码器侧上相同的数据(即通过有损压缩而获得的数据,而非原始的OAM数据)用于其处理。

[0206] 图14的3D音频编码器可在一些单独的模式下操作。

[0207] 除了在图10的上下文中所描述的第一模式和第二模式之外,图14的3D音频编码器可额外地在第三模式下操作,在此模式下,当预渲染器/混合器200不活跃时,核心编码器从单独的对象生成一个或多个传输声道。可选地或此外地,在此第三模式下,当对应于图10的混合器200的预渲染器/混合器200不活跃时,SAOC编码器800从原始声道生成一个或多个可选的或额外的传输声道。

[0208] 最后,当3D音频编码器用于第四模式下时,SAOC编码器800可对加入由预渲染器/混合器所生成的预渲染的对象的声道进行编码。因此,由于在第四模式下声道和对象已被完全地变换至单独的SAOC传输声道且不必传输如在图3和5中被指示为“SAOC-SI”的相关联的边信息,以及此外地任何经压缩的元数据的事实,在此第四模式下最低比特速率应用将提供良好的品质。

[0209] 在图14中,0AM编码器400为根据上述实施例中的一个的用于生成编码的音频信息的装置250的元数据编码器210。此外,在图14中,SAOC编码器800和USAC编码器300一起形成根据上述实施例中的一个的用于生成编码的音频信息的装置250的音频编码器220。

[0210] 根据实施例,提供一种用于对音频输入数据101进行编码以获得音频输出数据501的装置,用于对音频输入数据101进行编码的装置包括:

[0211] -输入接口1100,用于接收多个音频声道、多个音频对象以及与多个音频对象中的一个或多个相关的元数据;

[0212] -混合器200,用于混合多个对象和多个声道以获得多个预混合的声道,每个预混合的声道包括声道的音频数据和至少一个对象的音频数据;以及

[0213] -装置250,用于生成编码的音频信息,其包括如上所述的元数据编码器和音频编码器。

[0214] 用于生成编码的音频信息的装置250的音频编码器220为核心编码器(300),其用于对核心编码器输入数据进行核心编码。

[0215] 用于生成编码的音频信息的装置250的元数据编码器210为用于对与多个音频对象中的一个或多个相关的元数据进行压缩的元数据压缩器400。

[0216] 图11示出根据本发明的实施例的3D音频解码器。3D音频解码器接收编码的音频数据(即图10的数据501)作为输入。

[0217] 3D音频解码器包括元数据解压缩器1400、核心解码器1300、对象处理器1200、模式控制器1600以及后处理器1700。

[0218] 具体地,3D音频解码器用于对编码的音频数据进行解码,且输入接口用于接收编码的音频数据,编码的音频数据包括多个编码的声道和多个编码的对象以及在特定的模式下与多个对象相关的经压缩的元数据。

[0219] 此外,核心解码器1300用于对多个编码的声道和多个编码的对象进行解码,以及,此外地,元数据解压缩器用于对经压缩的元数据进行解压缩。

[0220] 此外,对象处理器1200用于使用经解压缩的元数据对由核心解码器1300所生成的多个解码的对象进行处理,以获得包括对象数据和解码的声道的预定数目的输出声道。如在1205处所指示的这些输出声道之后被输入后处理器1700。后处理器1700用于将多个输出声道1205转换成特定输出格式,该特定的输出格式可以为双声道输出格式或扬声器输出格式,如5.1、7.1等输出格式。

[0221] 优选地,3D音频解码器包括模式控制器1600,该模式控制器1600用于分析编码的数据以检测模式指示。因此,模式控制器1600连接到图11中的输入接口1100。然而,可选地,模式控制器在此并非为必要的。相反地,可通过任何其他种类的控制数据(如用户输入或任何其他控制)预设置灵活的音频解码器。优选地,由模式控制器1600控制的图11中的3D音频解码器用于旁路对象处理器并将多个解码的声道馈入后处理器1700。即当模式2已被应用

于图10的3D音频编码器时,此为模式2下的操作,即其中仅接收到预渲染的声道。可选地,当模式1已被应用于3D音频编码器时,即当3D音频编码器已执行单独的声道/对象编码时,则对象处理器1200不会被旁路,而多个解码的声道和多个解码的对象与由元数据解压缩器1400所生成的经解压缩的元数据一起被馈入对象处理器1200。

[0222] 优选地,是否将应用模式1或模式2的指示被包含在编码的音频数据中,然后模式控制器1600分析编码的数据以检测模式指示。当模式指示表示编码的音频数据包括编码的声道和编码的对象时,使用模式1;而当模式指示表示编码的音频数据不包含任何音频对象(即仅包含由图10的3D音频编码器的模式2获得的预渲染的声道)时,使用模式2。

[0223] 在图11中,元数据解压缩器1400为根据上述实施例中的一个的用于生成一个或多个音频声道的装置100的元数据解码器110。此外,在图11中,核心解码器1300、对象处理器1200以及后处理器1700一起形成根据上述实施例中的一个的用于生成一个或多个音频声道的装置100的音频解码器120。

[0224] 图13示出相对于图11的3D音频解码器的优选实施例,且图13的实施例与图12的3D音频编码器相对应。除了图11的3D音频解码器的实施方式之外,图13中的3D音频解码器包括SAOC解码器1800。此外,图11的对象处理器1200被实施为分离的对象渲染器1210和混合器1220,而取决于模式,对象渲染器1210的功能也可由SAOC解码器1800来实施。

[0225] 此外,后处理器1700可被实施为双声道渲染器1710或格式转换器1720。可选地,也可如1730所示地实施图11的数据1205的直接输出。因此,为了具有灵活性以及在需要较小的格式时的之后的后处理,优选地在解码器内对最高数目的(例如22.2或32)的声道执行处理。然而,当从一开始就清楚仅需要小格式(例如5.1格式)时,为了避免不必要的升混合操作以及随后的降混合操作,则优选地,如图11或6的简化操作1727所示,可施加跨越SAOC解码器和/或USAC解码器的特定控制。

[0226] 在本发明的优选实施例中,对象处理器1200包括SAOC解码器1800,且该SAOC解码器1800用于对核心解码器所输出的一个或多个传输声道以及相关参数化数据进行解码,并使用经解压缩的元数据以获得多个渲染的音频对象。至此,OAM输出连接至方块1800。

[0227] 此外,对象处理器1200用于渲染由核心解码器所输出的解码的对象,其并未被编码于SAOC传输声道,而被单独地编码于如对象渲染器1210所指示的典型单个的声道元件。此外,解码器包括与输出1730相对应的用于将混合器的输出输出至扬声器的输出接口。

[0228] 在另一实施例中,对象处理器1200包括空间音频对象编码解码器1800,用于对一个或多个传输声道以及表示编码的音频信号或编码的音频声道的相关参数化边信息进行解码,其中空间音频对象编码解码器用于将相关参数化信息以及经解压缩的元数据转码成可用于直接地渲染输出格式的经转码的参数化边信息,例如在SAOC的早期版本中所定义的。后处理器1700用于使用解码的传输声道和经转码的参数化边信息计算输出格式的音频声道。后处理器所执行的处理可类似于MPEG环绕处理或可以为任何其他的处理,如BCC处理等。

[0229] 在另一实施例中,对象处理器1200包括空间音频对象编码解码器1800,其用于使用(由核心解码器)解码的传输声道和参数化边信息直接地升混合并渲染用于输出格式的声道信号。

[0230] 此外,重要的是,图11的对象处理器1200额外地包括混合器1220,当存在与声道混

合的预渲染的对象时(即当图10的混合器200活跃时),混合器1220直接地接收USAC解码器1300所输出的数据作为输入。此外,混合器1220从执行对象渲染的对象渲染器接收未经SAOC解码的数据。此外,混合器接收SAOC解码器输出数据,即SAOC渲染的对象。

[0231] 混合器1220连接到输出接口1730、双声道渲染器1710以及格式转换器1720。双声道渲染器1710用于使用头部相关的传递函数或双耳空间脉冲响应(BRIR)将输出声道渲染成两个双耳声道。格式转换器1720用于将输出声道转换成输出格式,该输出格式具有比混合器的输出声道1205较少的数目的声道,且格式转换器1720需要再现布局(例如5.1扬声器等)的信息。

[0232] 在图13中,OAM解码器1400为根据上述实施例中的一个的用于生成一个或多个音频声道的装置100的元数据解码器110。此外,在图13中,对象渲染器1210、USAC解码器1300以及混合器1220一起形成根据上述实施例中的一个的用于生成一个或多个音频声道的装置100的音频解码器120。

[0233] 图15的3D音频解码器与图13的3D音频解码器的不同之处在于,SAOC解码器不仅能生成渲染的对象也能生成渲染的声道,且此为这样的情况:图14的3D音频编码器已被使用且在声道/预渲染的对象与SAOC编码器800的输入接口之间的连接900是活跃的。

[0234] 此外,基于矢量的幅度平移(VBAP)级1810用于从SAOC解码器接收再现布局的信息,并将渲染矩阵输出至SAOC解码器,以使得SAOC解码器最终能以1205(即32个扬声器)的高声道格式来提供渲染的声道,而无需混合器的任何其他操作。

[0235] 优选地,VBAP方块接收解码的OAM数据以得到渲染矩阵。更一般的,优选地需要再现布局和输入信号应被渲染到再现布局的位置的几何信息。此几何输入数据可以为用于对象的OAM数据或用于声道的声道位置信息,其已使用SAOC而被传输。

[0236] 然而,如果仅需要特定的输出接口,则VBAP状态1810已经提供用于例如5.1输出的所需的渲染矩阵。然后SAOC解码器1800执行来自SAOC传输声道、相关联的参数化数据以及经解压缩的元数据的直接渲染,无需混合器1220的任何互相作用直接渲染成所需的输出格式。然而,当应用模式之间的特定混合时,即对一些声道而非所有声道进行SAOC编码;或对一些对象而非所有对象进行SAOC编码;或当仅对特定数量的具有声道的预渲染的对象进行SAOC解码而对剩余声道不进行SAOC处理时,则混合器将来自单独的输入部分,即直接来自核心解码器1300、来自对象渲染器1210以及来自SAOC解码器1800的数据放在一起。

[0237] 在图15中,OAM解码器1400为根据上述实施例中的一个的用于生成一个或多个音频声道的装置100的元数据解码器110。此外,在图15中,由对象渲染器1210、USAC解码器1300以及混合器1220一起形成根据上述实施例中的一个的用于生成一个或多个音频声道的装置100的音频解码器120。

[0238] 提供一种对编码的音频数据进行解码的装置。对编码的音频数据进行解码的装置包括:

[0239] -输入接口1100,用于接收编码的音频数据,此编码的音频数据包括多个编码的声道、或多个编码的对象、或与多个对象有关的压缩元数据;以及

[0240] -如上所述的装置100,其用于生成一个或多个音频声道,包括元数据解码器110和音频声道生成器120。

[0241] 用于生成一个或多个音频声道的装置100的元数据解码器110为用于对经压缩的

元数据进行解压缩的元数据解压缩器400。

[0242] 用于生成一个或多个音频声道的装置100的音频声道生成器120包括用于对多个编码的声道和多个编码的对象进行解码的核心解码器1300。

[0243] 此外,音频声道生成器120还包括对象处理器1200,其使用经解压缩的元数据处理多个解码的对象,以从对象和解码的声道中获得包括音频数据的多个输出声道1205。

[0244] 此外,音频声道生成器120还包括后处理器1700,其用于将多个输出声道1205转换成输出格式。

[0245] 尽管已在装置的上下文中描述一些方面,显然的是,这些方面也表示对应方法的描述,其中区块或装置对应于方法步骤或方法步骤的特征。类似地,在方法步骤的上下文中所描述的方面也表示对应装置的对应区块或项目或特征的描述。

[0246] 本发明的经分解的信号可储存在数字存储介质上或可在传输介质上(例如无线传输介质或有线传输介质(例如因特网))上传输。

[0247] 取决于特定的实施要求,本发明的实施例可以硬件或软件实施。可使用具有存储于其上的电子可读控制信号的数字存储介质,例如软性磁盘、DVD、CD、ROM、PROM、EPROM、EEPROM或闪存,执行实施方案,这些电子可读控制信号与可编程计算机系统协作(或能够协作)以使得执行各个方法。

[0248] 根据本发明的一些实施例包括具有电子可读控制信号的非暂时性数据载体,这些电子可读控制信号能够与可编程计算机系统协作,使得执行本文中所描述的方法中的一个。

[0249] 通常,本发明的实施例可被实施为具有程序代码的计算机程序产品,当计算机程序产品执行于计算机上时,程序代码操作性地用于执行这些方法中的一个。程序代码可(例如)储存于机器可读载体上。

[0250] 其他实施例包括储存于机器可读载体上的用于执行本文中所描述的方法中的一个的计算机程序。

[0251] 换言之,因此,本发明方法的实施例为具有程序代码的计算机程序,当计算机程序执行于计算机上时,该程序代码用于执行本文中所描述的方法中的一个。

[0252] 因此,本发明方法的另一实施例为包括记录于其上的,用于执行本文中所描述的方法中的一个的计算机程序的数据载体(或数字存储介质,或计算机可读介质)。

[0253] 因此,本发明方法的另一实施例为表示用于执行本文中所描述的方法中的一个的计算机程序的数据流或信号序列。数据流或信号序列可例如用于经由数据通信连接(例如,经由因特网)而传送。

[0254] 另一实施例包括用于或经调适以执行本文中所描述的方法中的一个的处理构件,例如,计算机或可编程逻辑器件。

[0255] 另一实施例包括安装有用于执行本文中所描述的方法中的一个的计算机程序的计算机。

[0256] 在一些实施例中,可编程逻辑器件(例如,场可编程门阵列)可用于执行本文中所描述的方法的功能性中的一些或所有。在一些实施例中,场可编程门阵列可与微处理器协作,以便执行本文中所描述的方法中的一个。大体而言,优选地由任何硬件装置执行这些方法。

[0257] 上文所描述的实施例仅仅说明本发明的原理。应理解,对本文中所描述的配置及细节的修改及变型对本领域技术人员而言将是显而易见。因此,仅意欲由待决专利的权利要求的范围限制,而不由通过本文的实施例的描述及解释而提出的特定细节限制。

[0258] 参考文献

[0259] [1]Peters,N.,Lossius,T.and Schacher J.C.,“SpatDIF:Principles, Specification,and Examples”,9th Sound and Music Computing Conference, Copenhagen,Denmark,Jul.2012.

[0260] [2]Wright,M.,Freed,A.,“Open Sound Control:A New Protocol for Communicating with Sound Synthesizers”,International Computer Music Conference,Thessaloniki,Greece,1997.

[0261] [3]Matthias Geier,Jens Ahrens,and Sascha Spors. (2010) ,“Object-based audio reproduction and the audio scene description format”,Org.Sound,Vol.15, No.3,pp.219-227,December 2010.

[0262] [4]W3C,“Synchronized Multimedia Integration Language (SMIL 3.0)”, Dec.2008.

[0263] [5]W3C,“Extensible Markup Language(XML)1.0(Fifth Edition)”,Nov.2008.

[0264] [6]MPEG,“ISO/IEC International Standard 14496-3-Coding of audio-visual objects,Part 3Audio”,2009.

[0265] [7]Schmidt,J.;Schroeder,E.F. (2004) ,“New and Advanced Features for Audio Presentation in the MPEG-4Standard”,116th AES Convention,Berlin, Germany,May 2004

[0266] [8]Web3D,“International Standard ISO/IEC 14772-1:1997-The Virtual Reality Modeling Language (VRML) ,Part 1:Functional specification and UTF-8encoding”,1997.

[0267] [9]Sporer,T. (2012) ,“Codierung **räumlicher** Audiosignale mit leichtgewichtigen Audio-Objekten”,Proc.Annual Meeting of the German Audiological Society (DGA) ,Erlangen,Germany,Mar.2012.

[0268] [10]Cutler,C.C. (1950) ,“Differential Quantization of Communication Signals”,US Patent US2605361,Jul.1952.

[0269] [11]Ville Pulkki,“Virtual Sound Source Positioning Using Vector Base Amplitude Panning”;J.Audio Eng.Soc.,Volume 45,Issue 6,pp.456-466,June 1997.

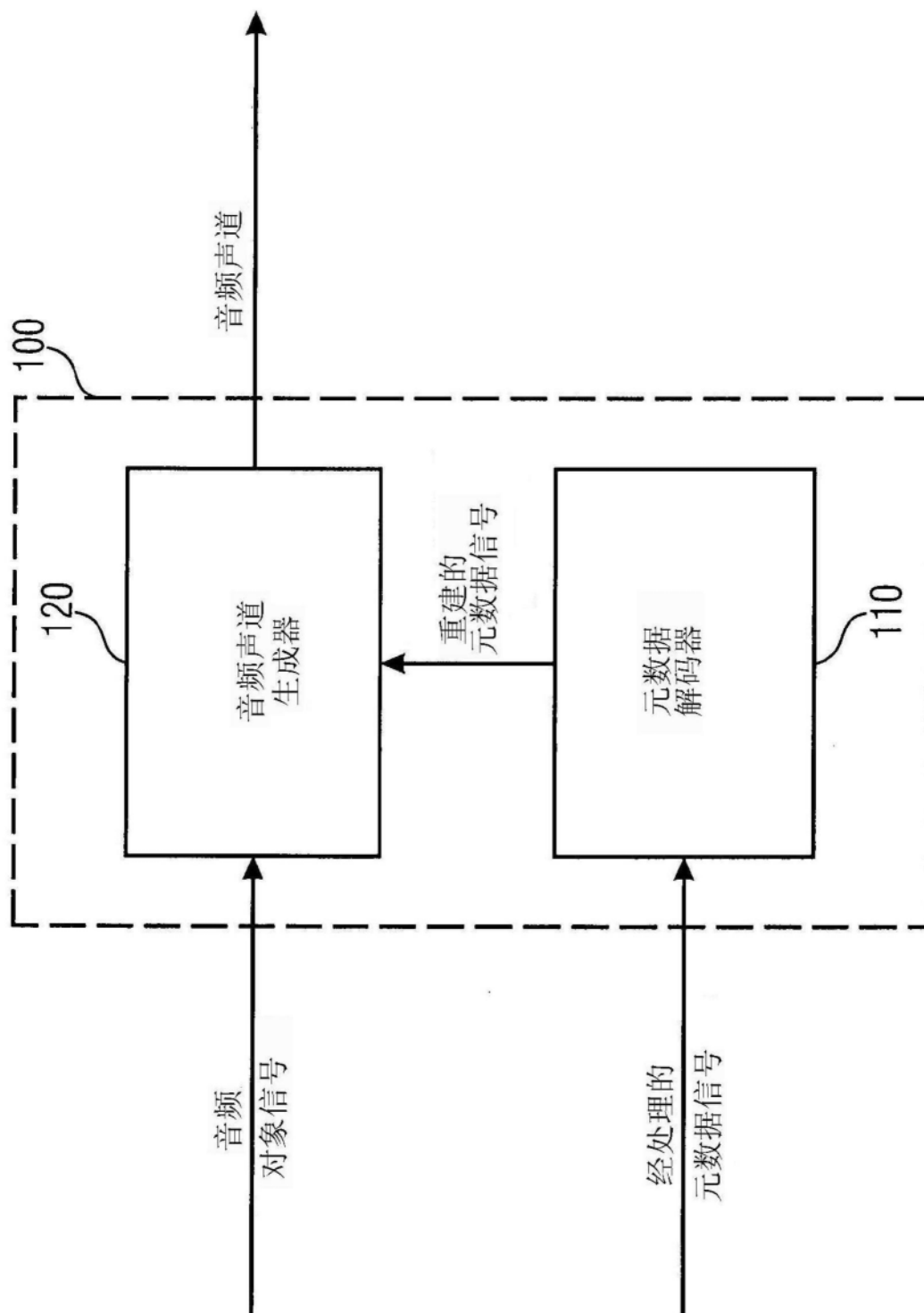


图1

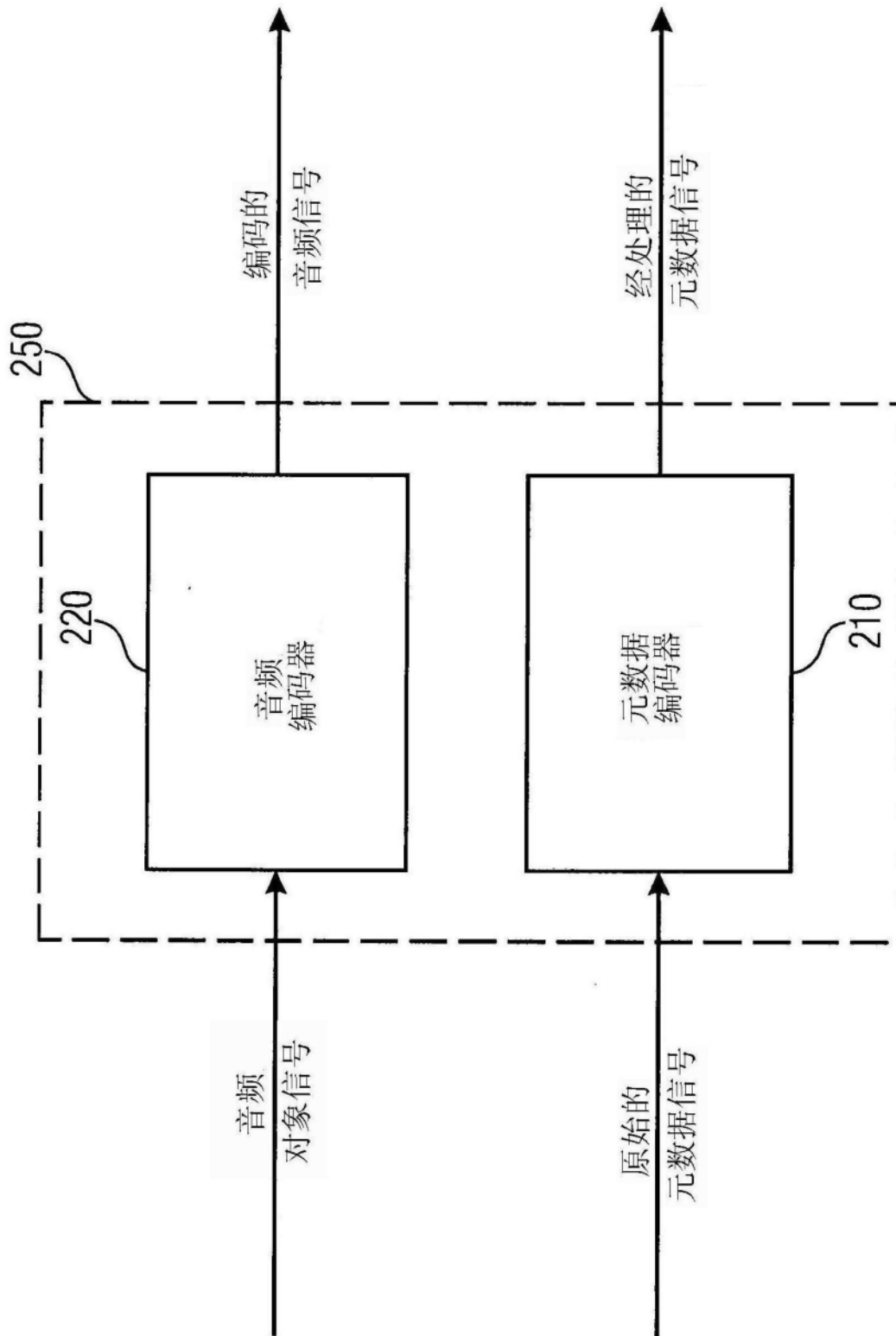


图2

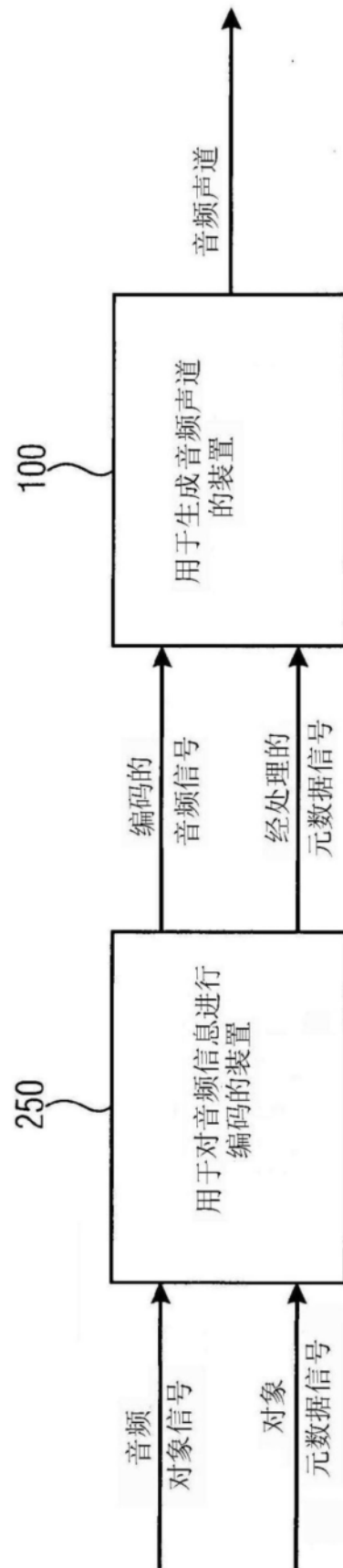


图3

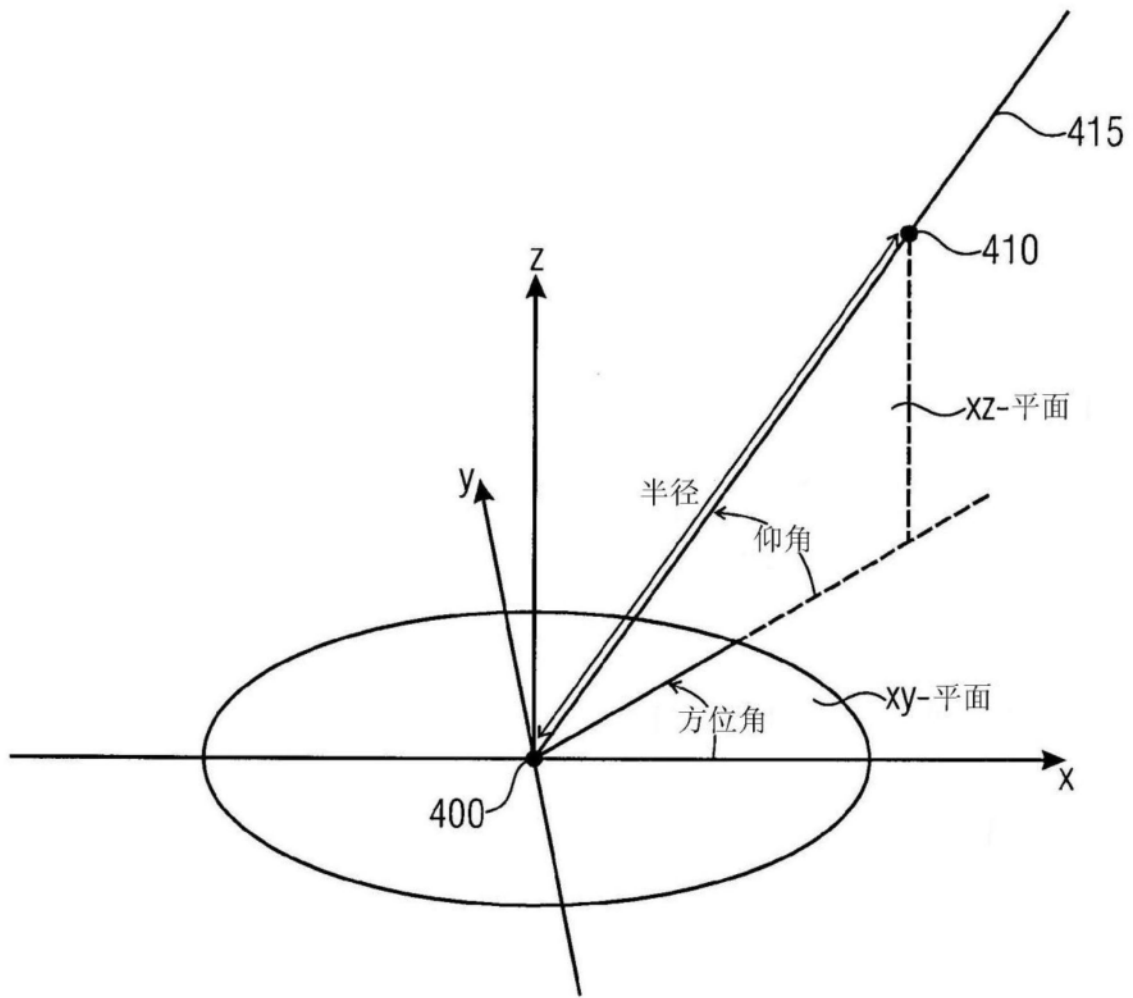


图4

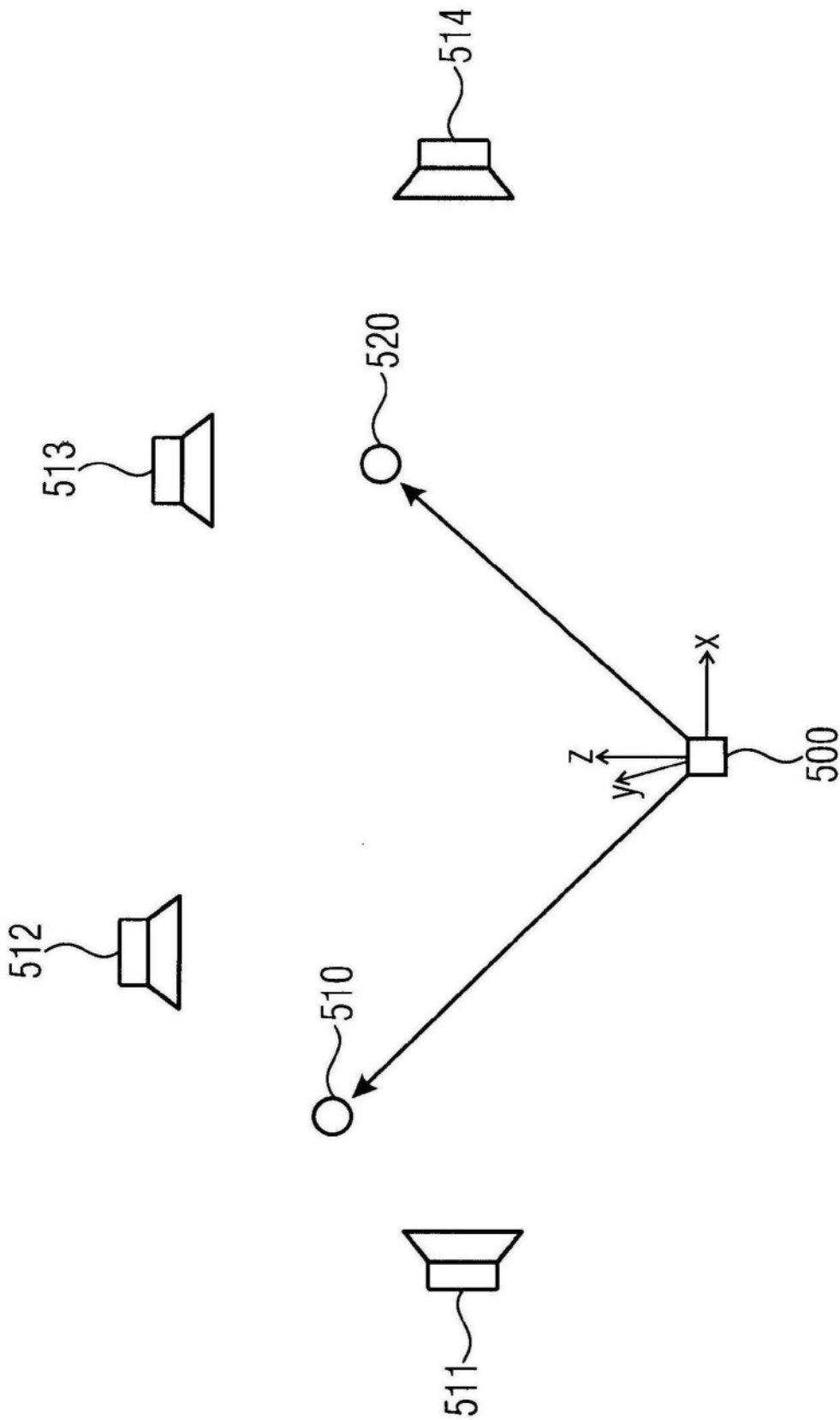


图5

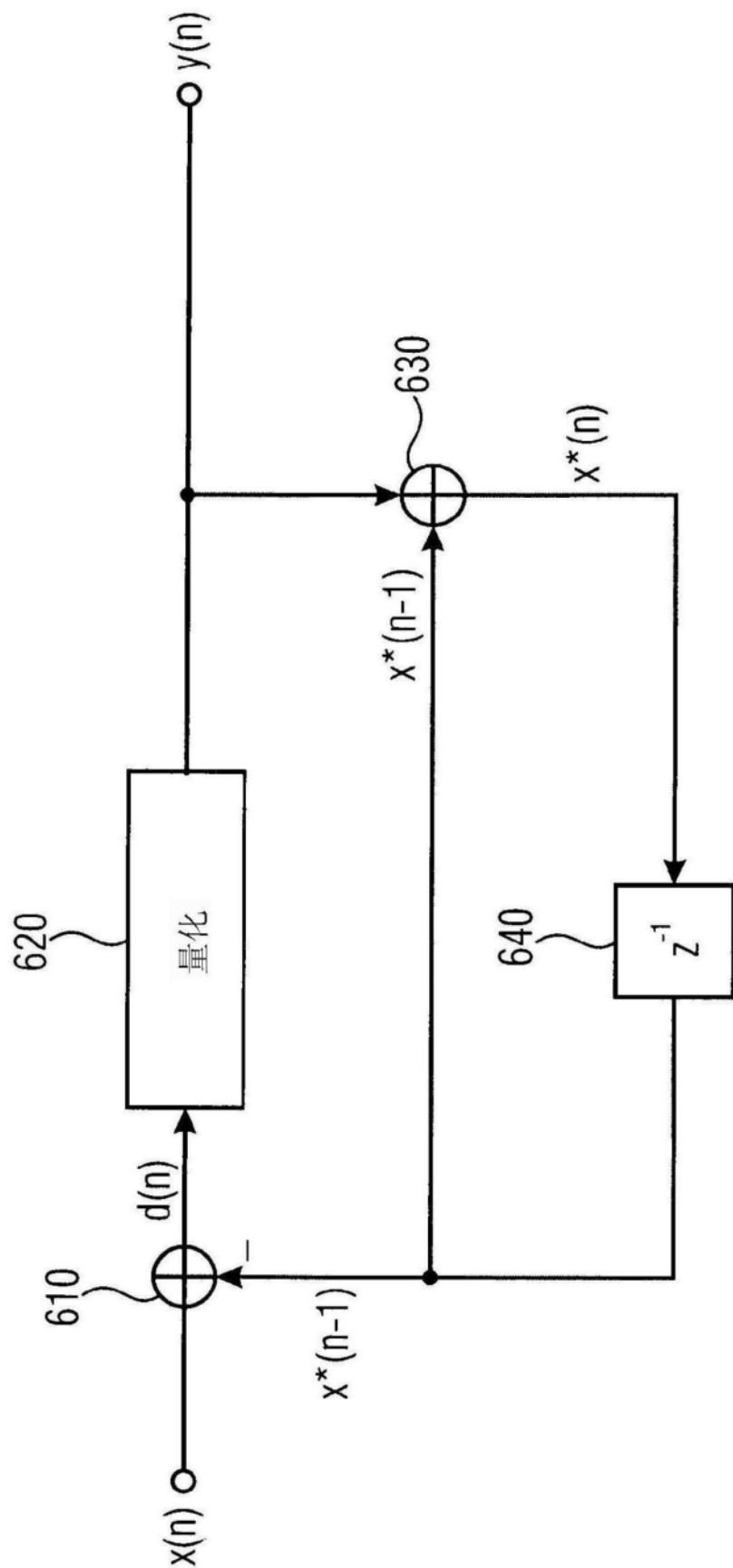


图6

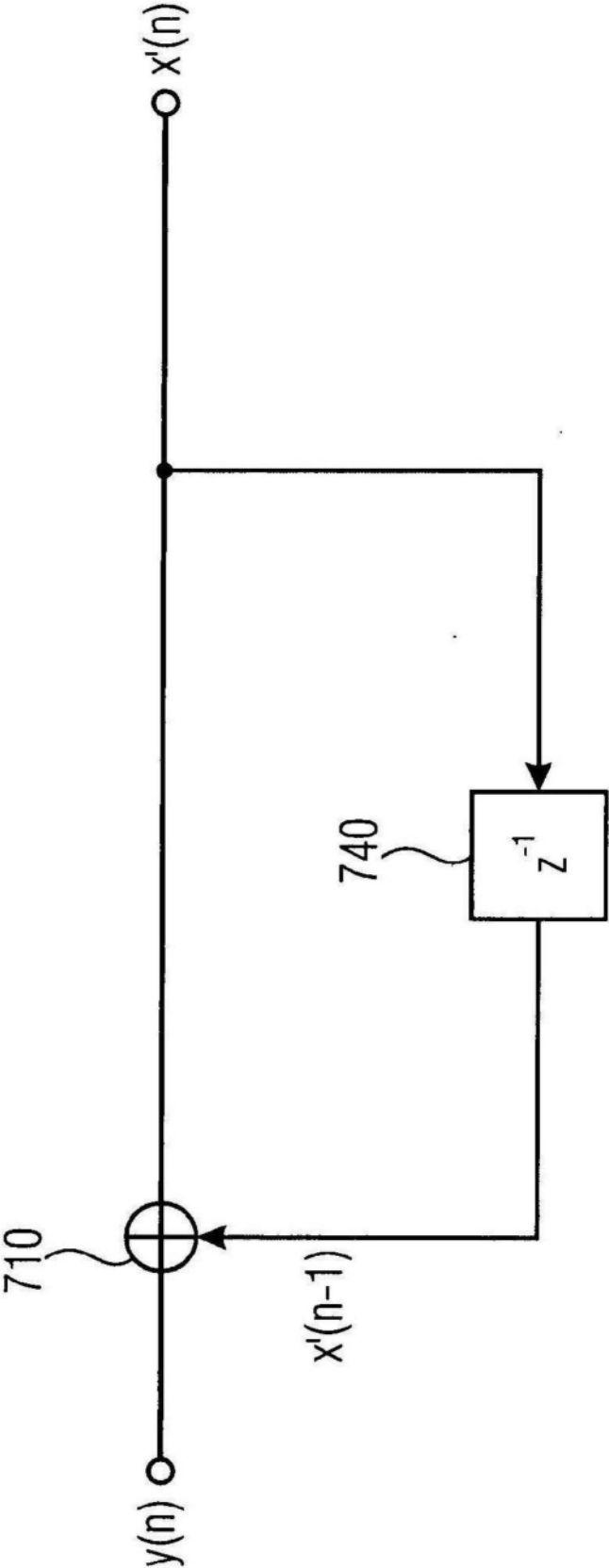


图7

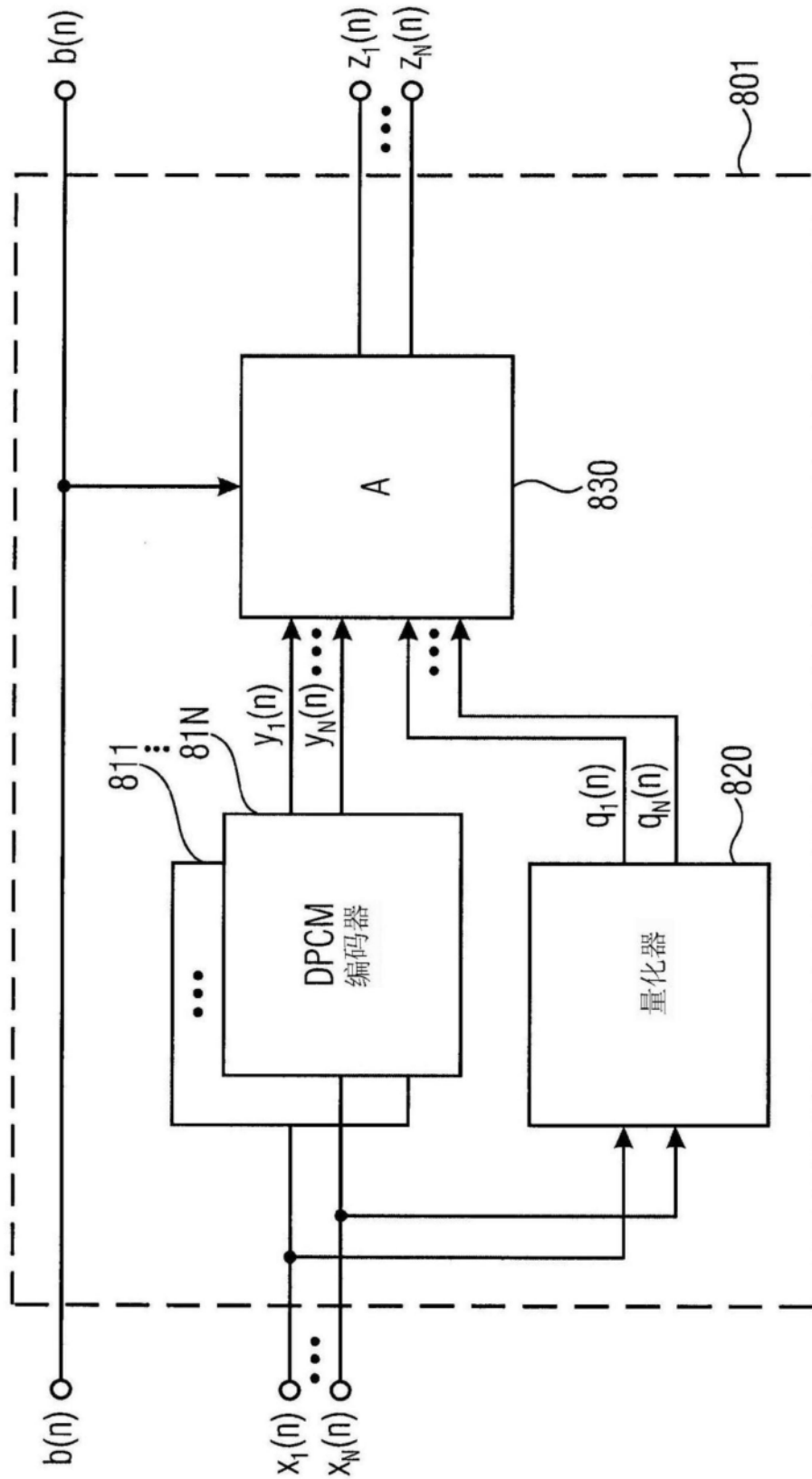


图8A

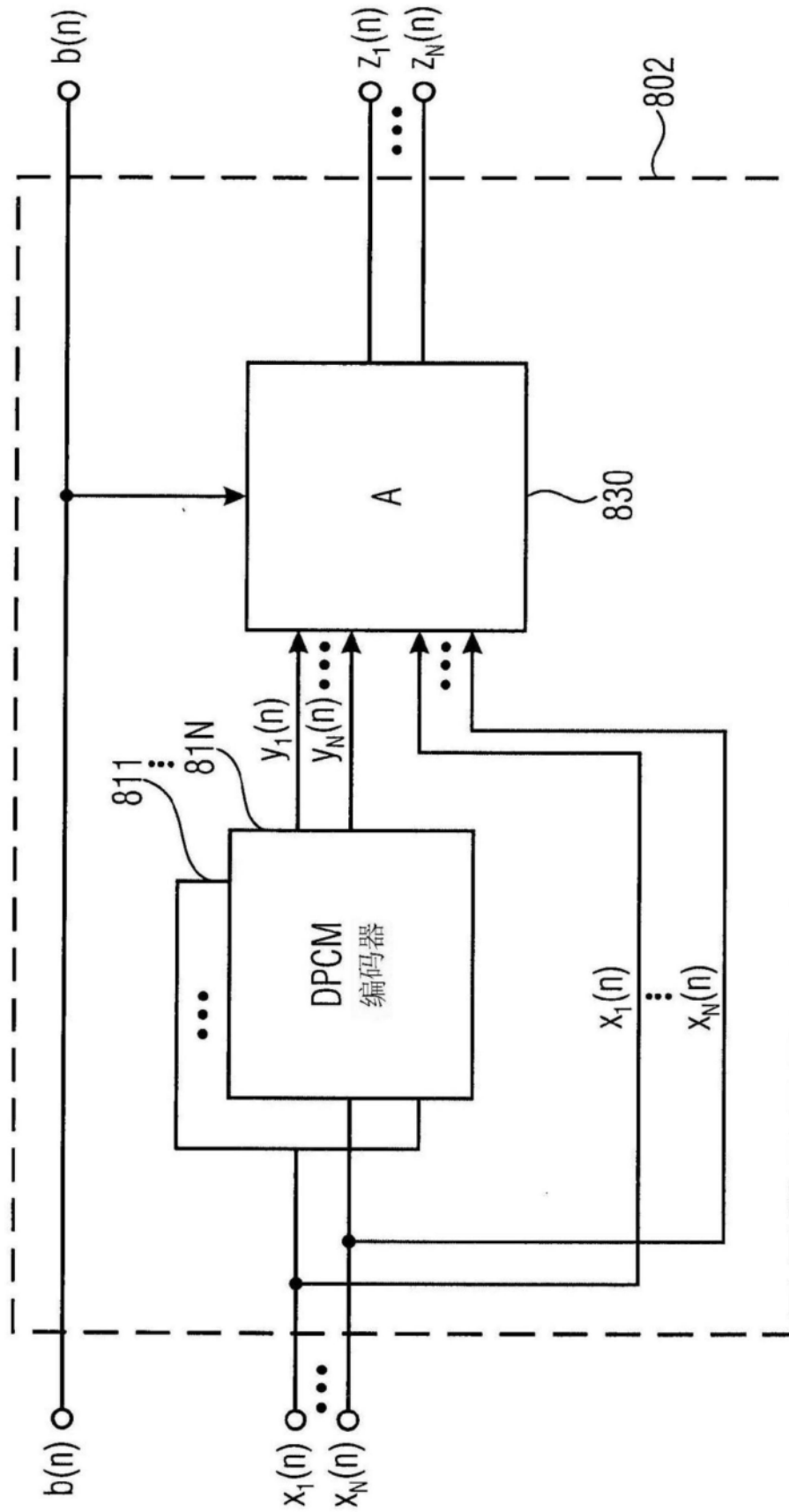


图8B

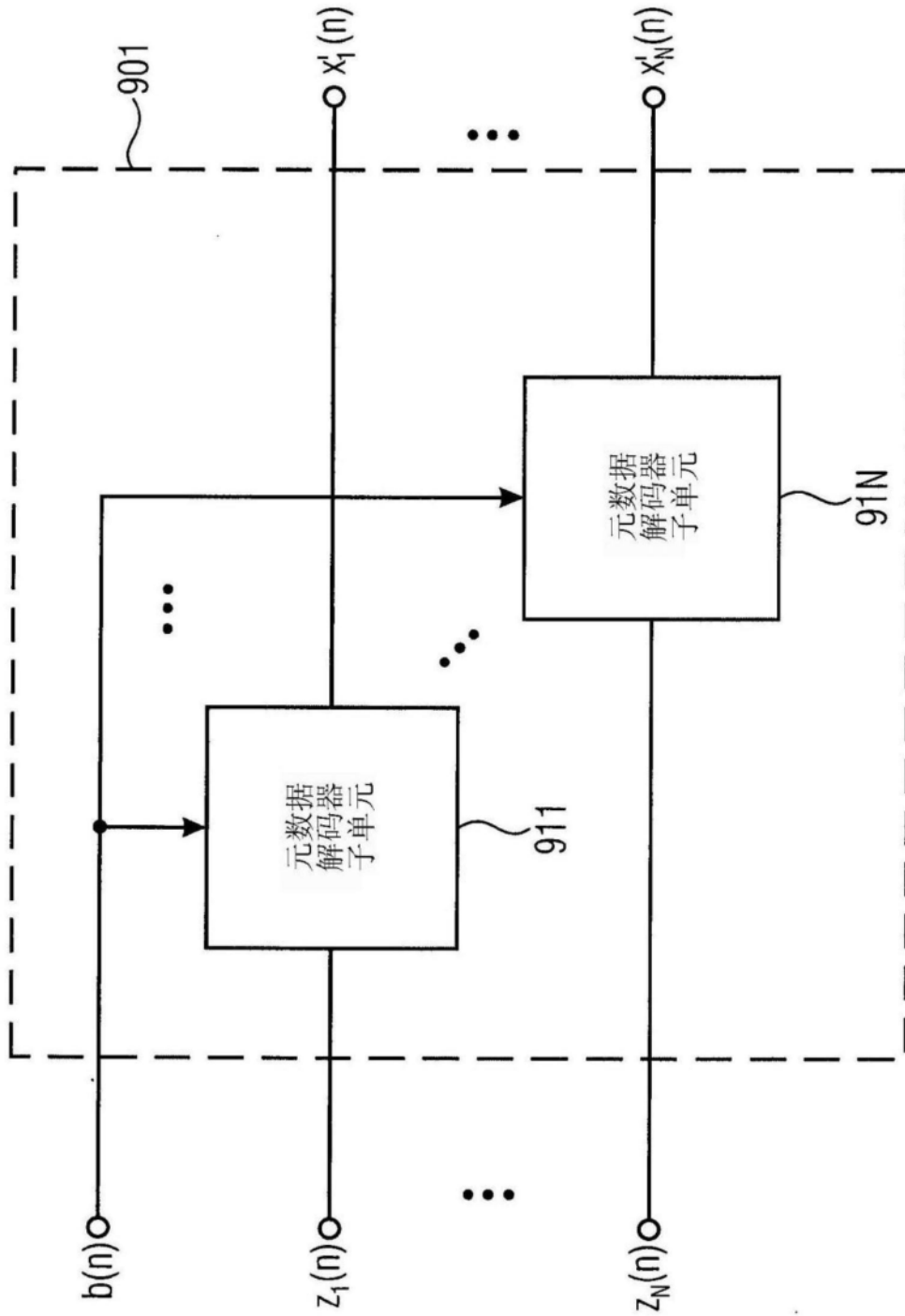


图9A

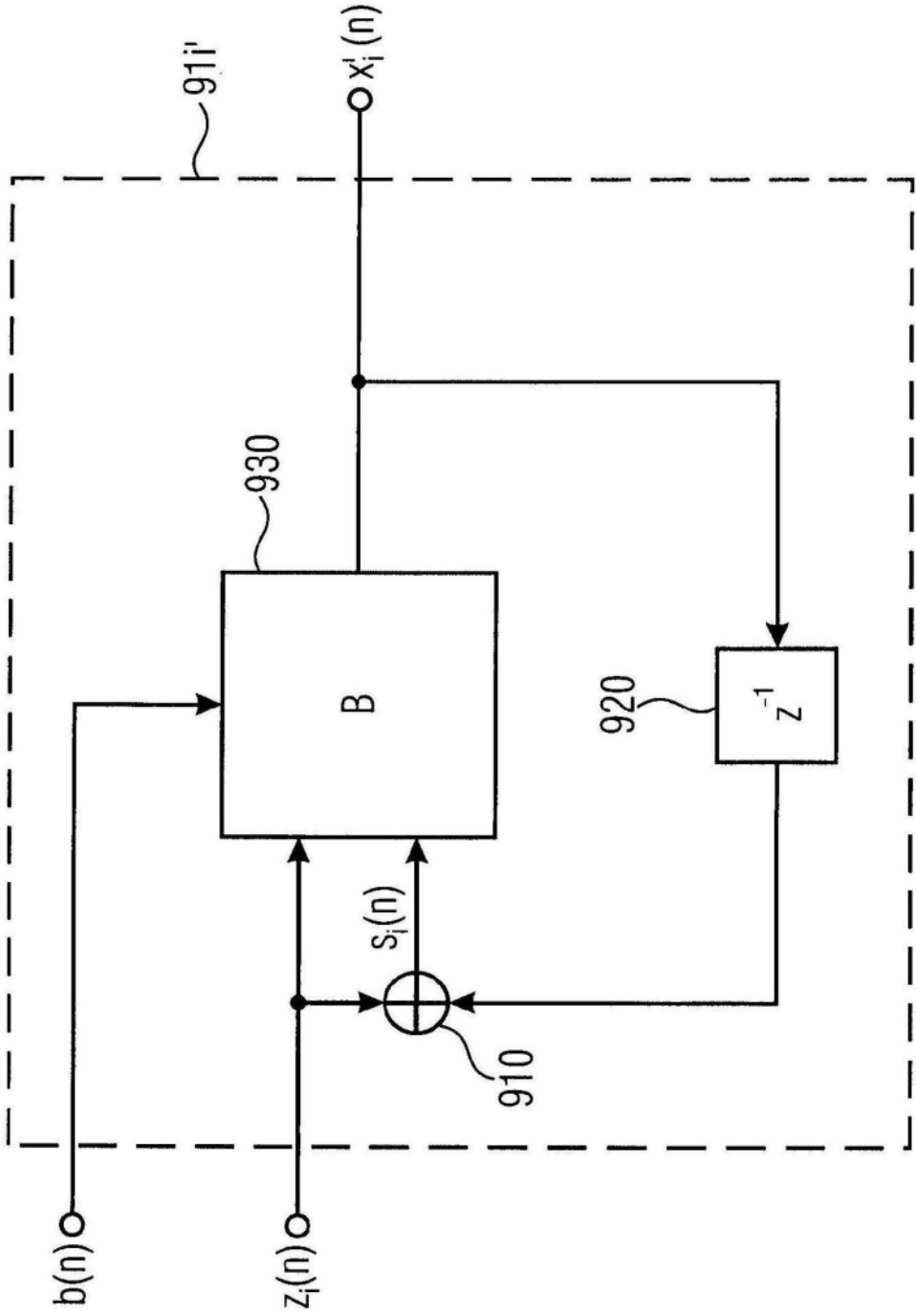


图9B

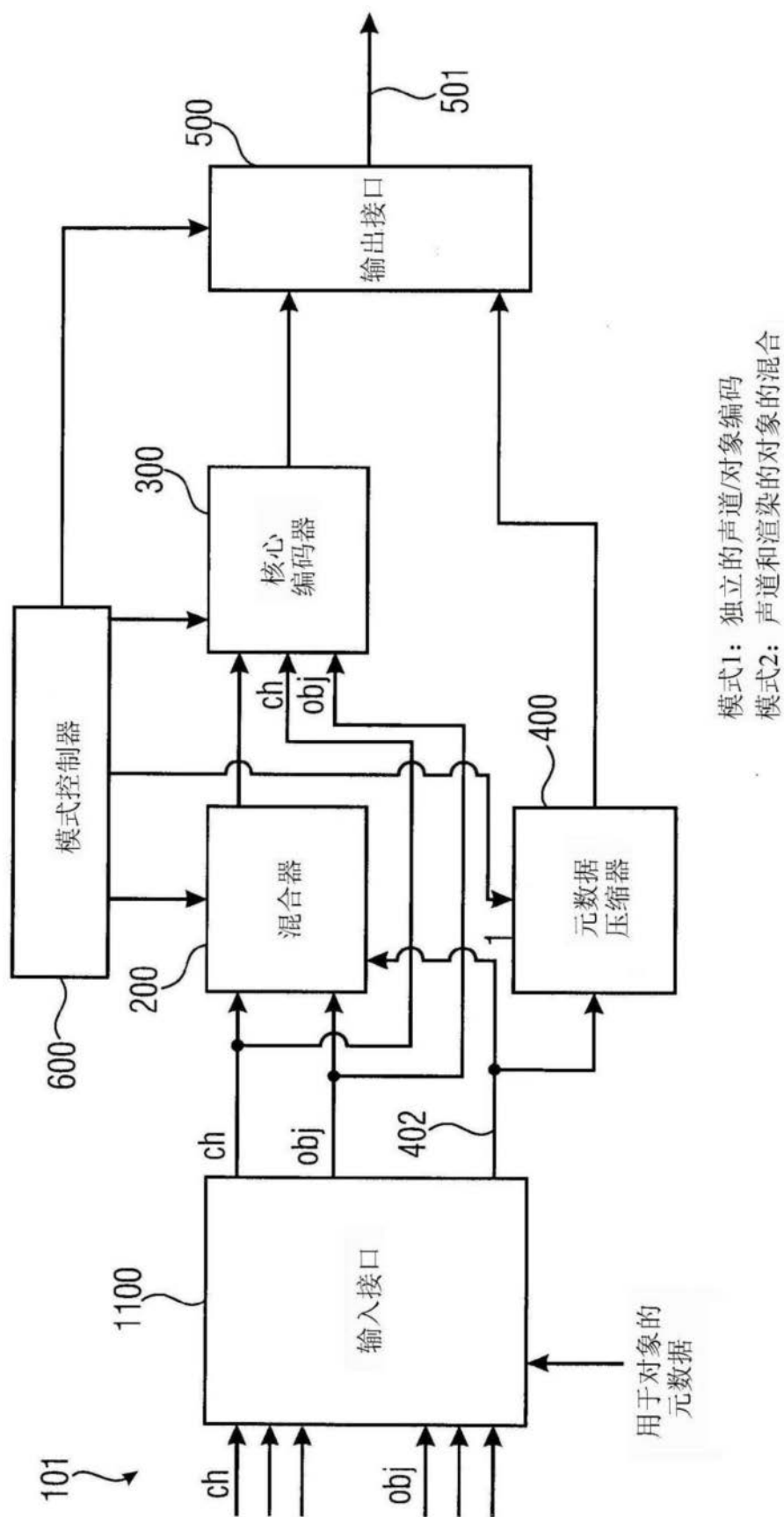


图10(编码器)

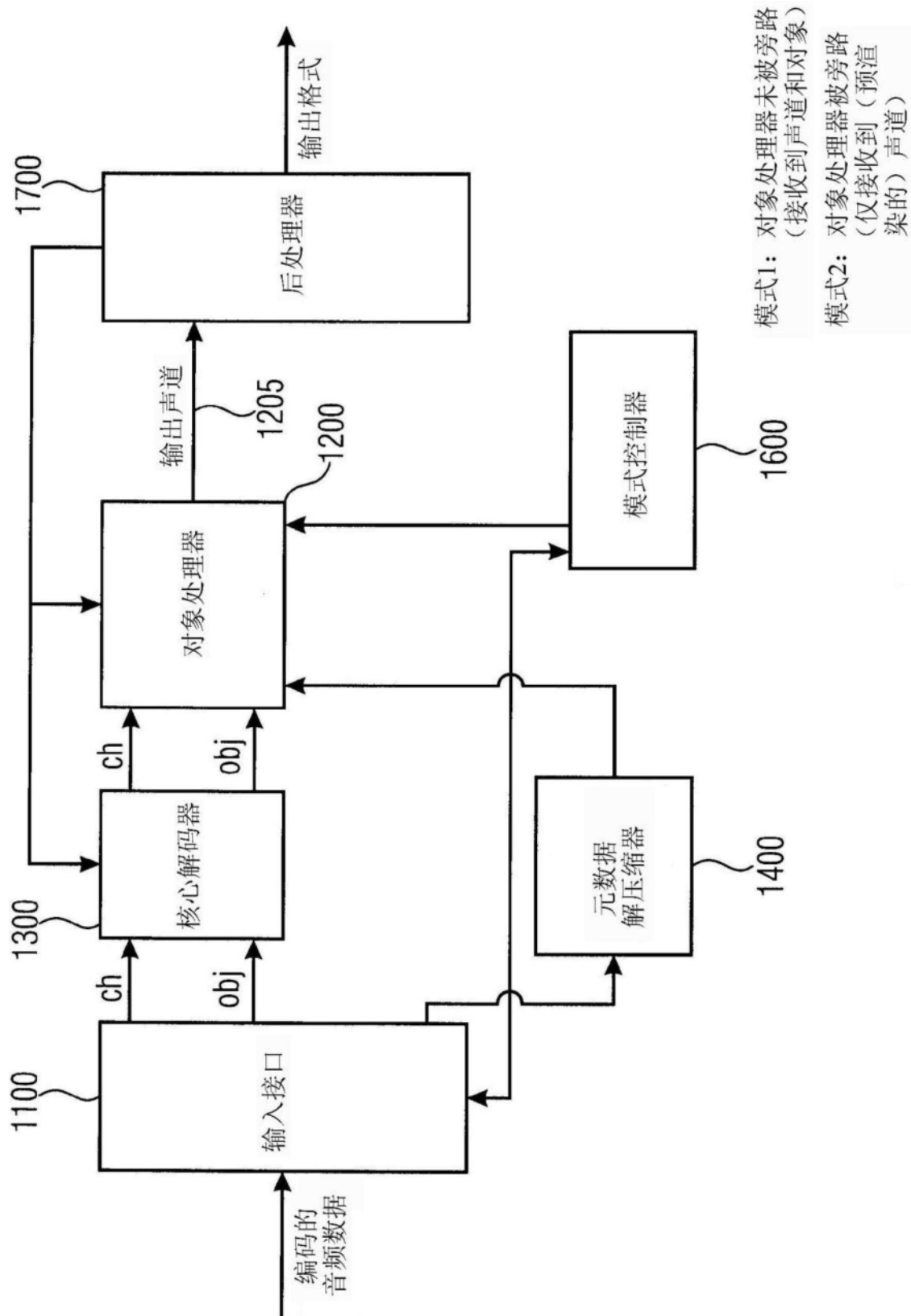


图11 (解码器)

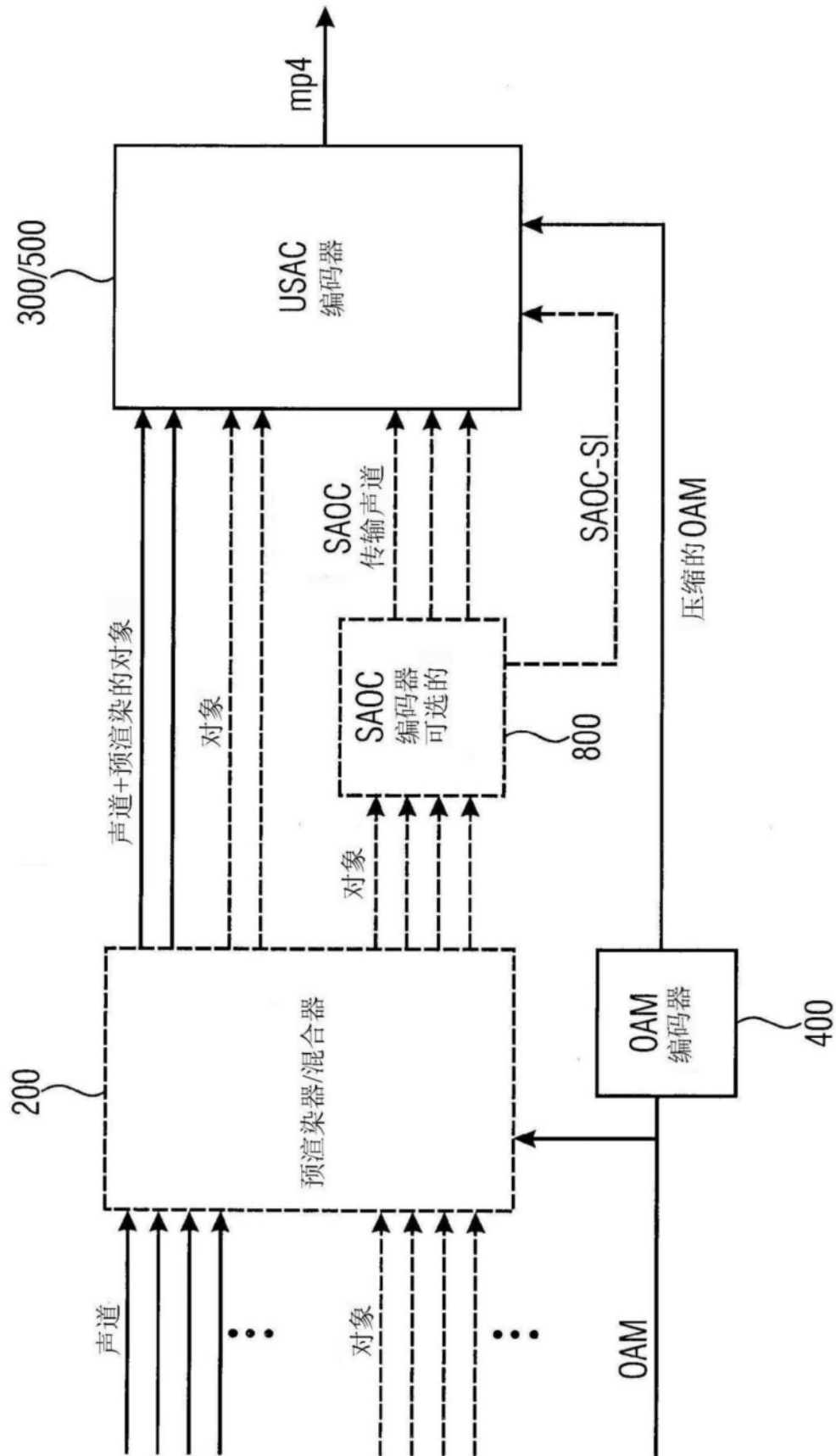
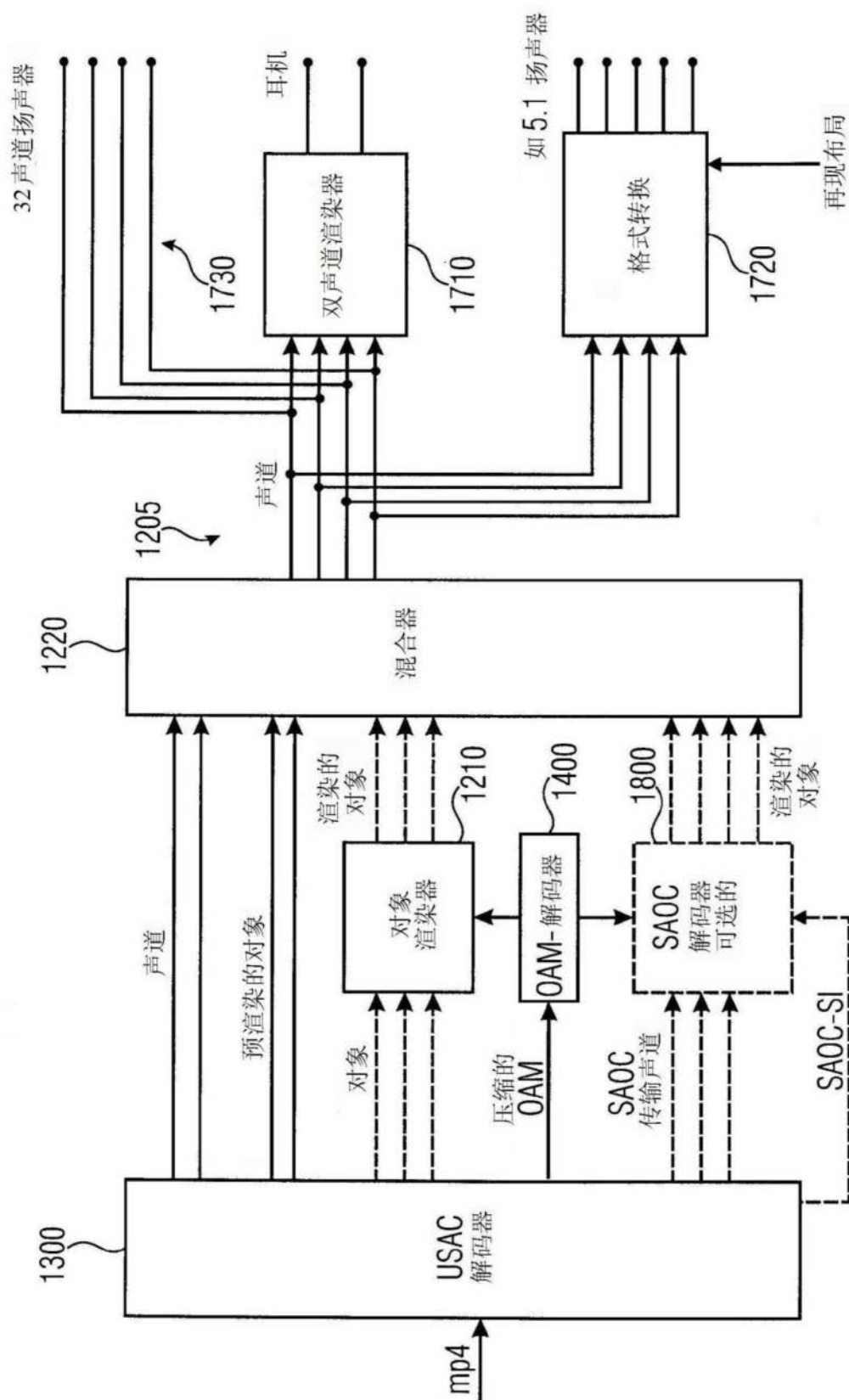


图12(编码器)



以SBR模式在双声道渲染器、格式转换器、SAOC解码器、USAC解码器中直接的QMF域处理的

图13(解码器)

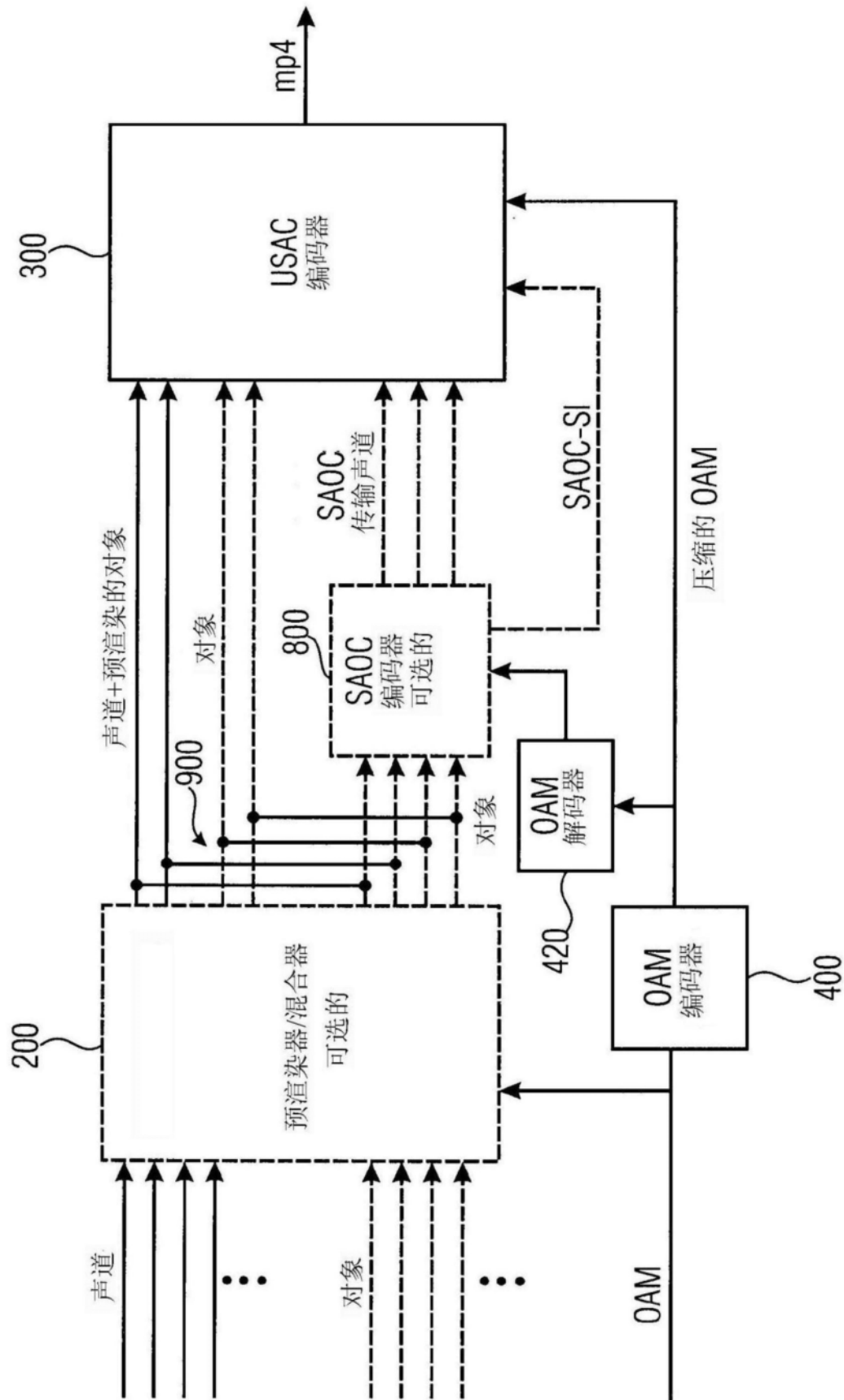


图14(编码器)

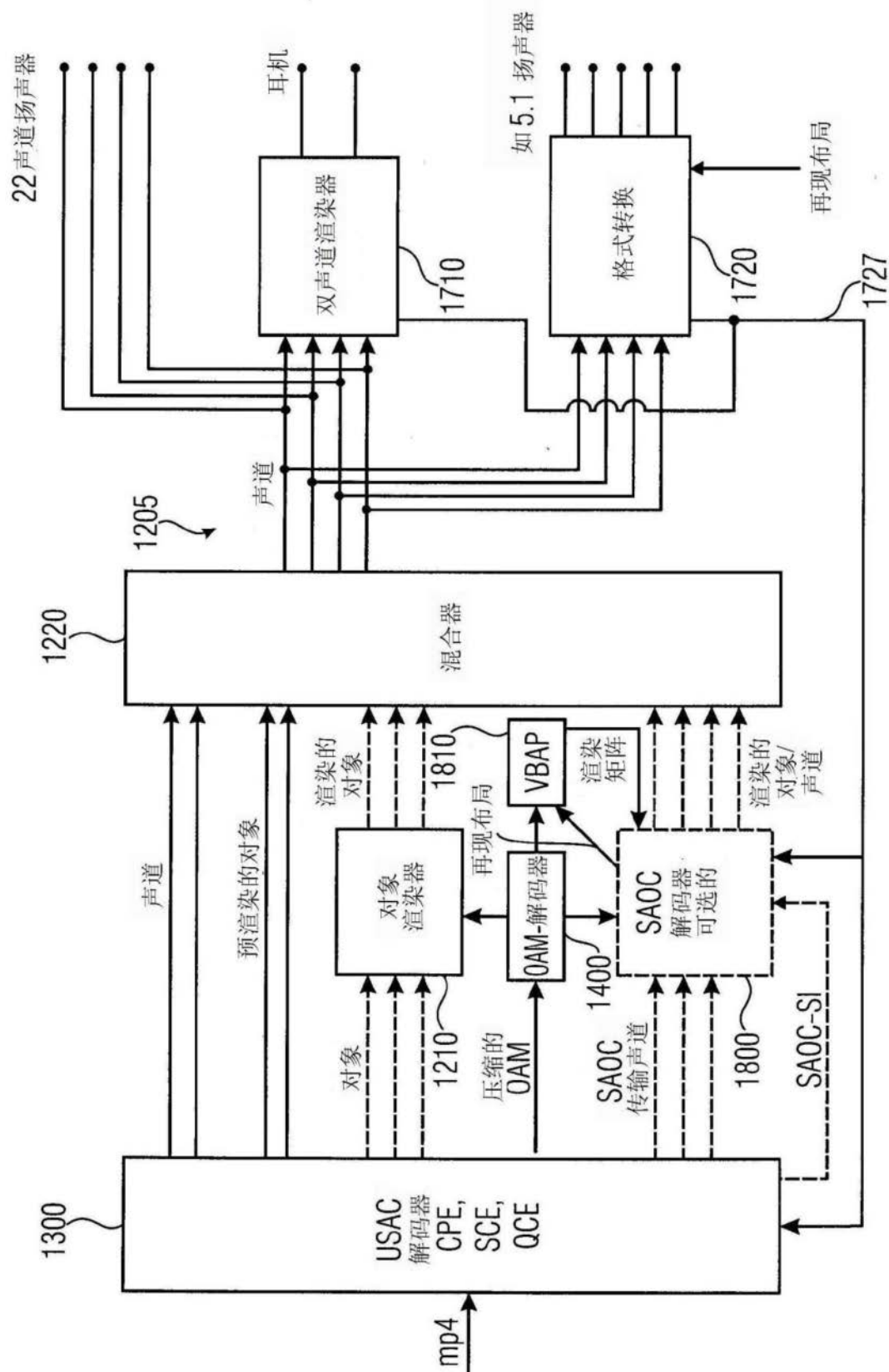


图15(解码器)