



(12) 发明专利

(10) 授权公告号 CN 112233684 B

(45) 授权公告日 2024.03.19

(21) 申请号 202011242898.5

(22) 申请日 2016.03.08

(65) 同一申请的已公布的文献号

申请公布号 CN 112233684 A

(43) 申请公布日 2021.01.15

(30) 优先权数据

15158234.3 2015.03.09 EP

15172492.9 2015.06.17 EP

(62) 分案原申请数据

201680026823.9 2016.03.08

(73) 专利权人 弗劳恩霍夫应用研究促进协会

地址 德国慕尼黑

(72) 发明人 萨沙·迪克 弗洛里安·舒

尼古拉斯·里特尔博谢

托比亚斯·施威格勒

理查德·福格

约翰内斯·希尔珀特

马蒂亚斯·诺伊辛格

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

专利代理人 潘剑颖

(54) 发明名称

用于对多声道信号进行编码或解码的装置与方法

(57) 摘要

实施例提供了一种对具有至少三个声道的多声道信号进行编码的装置。该装置包括迭代处理器、声道编码器和输出接口。迭代处理器被配置为在第一迭代步骤中计算至少三个声道中的每对之间的声道间相关值，选择具有最高值或具有高于阈值的值的对，并且使用多声道处理操作处理所选择的对，以导出针对所选择的对的第一多声道参数并导出第一处理声道。迭代处理器被配置为在第二迭代步骤中通过使用处理声道的至少一个而执行所述计算、所述选择和所述处理，以导出第二多声道参数和第二处理声道。声

(51) Int.CI.

G10L 19/02 (2013.01)

G10L 19/008 (2013.01)

H04S 3/00 (2006.01)

(56) 对比文件

CN 101872618 A, 2010.10.27

CN 104364842 A, 2015.02.18

CN 102656628 A, 2012.09.05

CN 101133680 A, 2008.02.27

CN 101031959 A, 2007.09.05

WO 2009146734 A1, 2009.12.10

US 2006190247 A1, 2006.08.24

US 2007280485 A1, 2007.12.06

US 2005074127 A1, 2005.04.07

JP 2006323314 A, 2006.11.30

JP 2011501544 A, 2011.01.06

JP 2015011076 A, 2015.01.19

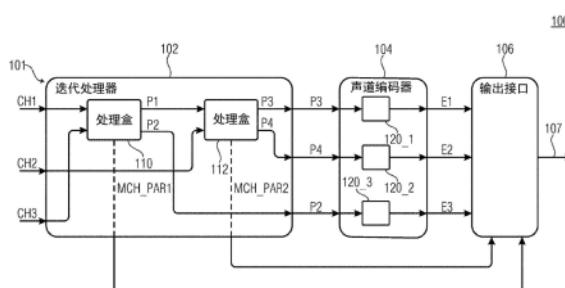
US 2011224993 A1, 2011.09.15

US 2013077793 A1, 2013.03.28

审查员 朱晓旻

权利要求书4页 说明书24页 附图6页

道编码器被配置为对根据迭代处理得到的声道进行编码以获得编码声道。输出接口被配置为产生具有编码声道以及第一多声道参数和第二多声道参数的编码多声道信号。



1. 一种用于对具有至少三个声道 (CH1 至 CH3) 的多声道信号 (101) 进行编码的装置 (100)，包括：

迭代处理器 (102)，用于在第一迭代步骤中计算所述至少三个声道 (CH1 至 CH3) 中的每对之间的声道间相关值，在第一迭代步骤中选择具有最高值或具有高于阈值的值的对，以及使用多声道处理操作 (110、112) 处理所选择的对，以导出针对所选择的对的第一多声道参数 (MCH_PAR1) 并导出第一处理声道 (P1、P2)，

其中，所述迭代处理器 (102) 被配置为在第二迭代步骤中使用处理声道 (P1) 中的至少一个执行所述计算、所述选择和所述处理，以导出第二多声道参数 (MCH_PAR2) 和第二处理声道 (P3、P4)；

声道编码器，用于对从根据所述迭代处理器 (104) 执行的迭代处理得到的声道 (P2 至 P4) 进行编码以获得编码声道 (E1 至 E3)；以及

输出接口 (106)，用于产生具有所述编码声道 (E1 至 E3) 以及第一多声道参数和第二多声道参数 (MCH_PAR1、MCH_PAR2) 的编码多声道信号 (107)，

其中，所述迭代处理器 (102) 被配置为针对第一帧导出多个选择对指示，并且所述输出接口 (106) 被配置为针对第一帧之后的第二帧在所述多声道信号 (107) 中包括保持指示符，所述保持指示符指示第二帧具有与第一帧相同的多个选择对指示。

2. 根据权利要求 1 所述的装置 (100)，其中，所述输出接口 (106) 被配置为将所述编码多声道信号 (107) 产生为串行比特流，使得第二多声道参数 (MCH_PAR2) 在所述编码多声道信号中位于第一多声道参数 (MCH_PAR1) 之前。

3. 根据权利要求 1 所述的装置 (100)，其中，所述迭代处理器 (102) 被配置为执行包括包含以下各项的组中的至少一项在内的立体声处理：使用根据所选择的对的旋转角度计算的旋转处理以及预测处理。

4. 根据权利要求 1 所述的装置 (100)，其中，第一多声道参数 (MCH_PAR1) 包括在第一迭代步骤中所选择的对中的声道的第一标识，并且第二多声道参数 (MCH_PAR2) 包括在第二迭代步骤中所选择的对中的声道的第二标识。

5. 根据权利要求 1 所述的装置 (100)，其中，所述迭代处理器 (102) 被配置为使用包括多个频带的每个声道的帧来计算声道间相关性，以便获得针对所述多个频带的单个声道间相关值，并且

其中，所述迭代处理器 (104) 被配置为针对所述多个频带中的每个频带执行所述多声道处理，从而针对所述多个频带中的每个频带获得第一多声道参数 (MCH_PAR1) 或第二多声道参数 (MCH_PAR2)。

6. 根据权利要求 1 所述的装置 (100)，其中，所述迭代处理器 (102) 被配置为仅在对的电平差小于阈值时才选择所述对，所述阈值小于 40dB 或 25dB 或 12dB，或者小于 6dB。

7. 根据权利要求 1 所述的装置 (100)，其中，所述迭代处理器 (102) 被配置为计算归一化相关值，并且所述迭代处理器 (102) 被配置为在所述相关值大于 0.2 并且优选为 0.3 时选择对。

8. 根据权利要求 1 所述的装置 (100)，其中，所述迭代处理器 (102) 被配置为在所述多声道处理中计算立体声参数，并且所述迭代处理器 (102) 被配置为仅在立体声参数高于由立体声参数量化器定义的量化到零阈值的频带中执行立体声处理。

9. 根据权利要求1所述的装置(100),其中,所述迭代处理器(102)被配置为在所述多声道处理中计算旋转角度,并且所述迭代处理器(102)被配置为仅在旋转角度高于解码器侧去量化到零阈值的频带中执行旋转处理。

10. 根据权利要求1所述的装置(100),其中,所述迭代处理器(102)被配置为在第二迭代步骤中不选择在第一迭代步骤中所选择的对,并且如果可行的话,在任何进一步的迭代步骤中也不选择该所选择的对。

11. 根据权利要求1所述的装置(100),其中,所述迭代处理器(102)被配置为执行迭代步骤直到达到迭代终止标准,其中,所述迭代终止标准是迭代步骤的最大数量等于所述多声道信号(101)的声道的总数量或比所述多声道信号(101)的声道的总数量大两个,或者所述迭代终止标准是当所述声道间相关值不具有大于所述阈值的值时。

12. 根据权利要求1所述的装置(100),其中,所述迭代处理器(102)被配置为在第一迭代步骤中使用所述多声道处理来处理所选择的对,使得所述处理声道(P1、P2)是中声道(P1)和侧声道(P2);以及

其中,所述迭代处理器(102)被配置为在第二迭代步骤中通过仅使用所述处理声道(P1、P2)的中声道(P1)作为所述处理声道(P1、P2)的所述至少一个来执行所述计算、所述选择和所述处理,以导出第二多声道参数(MCH_PAR2)和第二处理声道(P3、P4)。

13. 根据权利要求1所述的装置(100),其中,所述声道编码器包括用于对根据所述迭代处理得到的声道(P2至P4)进行编码的声道编码器(1201至1203),其中,所述声道编码器被配置为对声道(P2至P4)进行编码,使得与对具有较多能量的声道进行编码相比,使用较少的比特来对具有较少能量的声道进行编码。

14. 一种用于对具有编码声道(E1至E3)以及至少第一多声道参数和第二多声道参数(MCH_PAR1、MCH_PAR2)的编码多声道信号(107)进行解码的装置(200),包括:

声道解码器(202),用于对所述编码声道(E1至E3)进行解码以获得解码声道(D1至D3);以及

多声道处理器(204),用于使用通过第二多声道参数(MCH_PAR2)标识的解码声道(D1至D3)中的第二对并且使用第二多声道参数(MCH_PAR2)来执行多声道处理,以获得处理声道(P1*、P2*),以及用于使用通过第一多声道参数(MCH_PAR1)标识的声道(D1至D3、P1*、P2*)中的第一对并且使用第一多声道参数(MCH_PAR1)来执行进一步的多声道处理,其中,声道的第一对包括至少一个处理声道(P1*、P2*),

其中,所述编码多声道信号(107)包括针对第一帧的第一多声道参数和第二多声道参数(MCH_PAR1、MCH_PAR2)以及针对第一帧之后的第二帧的保持指示符,并且

其中,所述多声道处理器(204)被配置为在第二帧中对与在第一帧中所使用的声音的第二对和第一对相同的第二对和第一对执行所述多声道处理和所述进一步的多声道处理。

15. 根据权利要求14所述的装置(200),其中,所述多声道处理和所述进一步的多声道处理包括使用立体声参数的立体声处理,其中,针对解码声道(D1至D3)的单独比例因子频带或比例因子频带组,第一立体声参数包括在第一多声道参数(MCH_PAR1)中并且第二立体声参数包括在第二多声道参数(MCH_PAR2)中。

16. 根据权利要求14所述的装置(200),其中,第一多声道参数或第二多声道参数(MCH_PAR1、MCH_PAR2)包括多声道处理掩码,所述多声道处理掩码指示哪些比例因子频带经过多

声道处理以及哪些比例因子频带未经过多声道处理,并且

其中,所述多声道处理器(204)被配置为在由所述多声道处理掩码指示的比例因子频带中不执行多声道处理。

17.根据权利要求14所述的装置(200),其中,第一多声道参数和第二多声道参数(MCH_PAR1、MCH_PAR2)均包括声道对标识,并且

其中,所述多声道处理器(204)被配置为通过使用预定义解码规则或在编码多声道信号中指示的解码规则对所述声道对标识进行解码。

18.根据权利要求17所述的装置(200),其中,所述解码规则是霍夫曼解码规则,并且所述多声道处理器(204)被配置为对所述声道对标识执行霍夫曼解码。

19.根据权利要求14所述的装置(200),其中,所述编码多声道信号(107)包括多声道处理允许指示符,所述多声道处理允许指示符仅指示解码声道的允许进行所述多声道处理的子组并且指示不允许进行所述多声道处理的至少一个解码声道,并且

其中,所述多声道处理器(204)被配置为不对所述多声道处理允许指示符指示不允许进行所述多声道处理的所述至少一个解码声道执行任何多声道处理。

20.根据权利要求14所述的装置(200),其中,第一多声道参数和第二多声道参数(MCH_PAR1、MCH_PAR2)包括立体声参数,并且所述立体声参数被差分编码,并且所述多声道处理器(204)包括用于对经过差分编码的立体声参数进行差分解码的差分解码器。

21.根据权利要求14所述的装置,其中,所述编码多声道信号(107)是串行信号,其中,在解码器(200)处,第二多声道参数(MCH_PAR2)在第一多声道参数(MCH_PAR1)之前被接收,并且

其中,所述多声道处理器(204)被配置为按照所述解码器(200)接收多声道参数(MCH_PAR1、MCH_PAR2)的顺序来处理所述解码声道(D1至D3)。

22.一种用于对具有至少三个声道的多声道信号进行编码的方法(300),包括:

在第一迭代步骤中计算(302)至少三个声道中的每对之间的声道间相关值,在第一迭代步骤中选择具有最高值或具有高于阈值的值的对,并且使用多声道处理操作处理所选择的对,以导出针对所选择的对的第一多声道参数并导出第一处理声道;

在第二迭代步骤中使用所述处理声道中的至少一个来执行(304)所述计算、所述选择和所述处理,以导出第二多声道参数和第二处理声道;

对根据由迭代处理器执行的迭代处理得到的声道进行编码(306)以获得编码声道;以及

产生(308)具有编码声道以及第一多声道参数和第二多声道参数的编码多声道信号,

针对第一帧导出多个选择对指示,并且针对第一帧之后的第二帧在所述多声道信号中包括保持指示符,所述保持指示符指示第二帧具有与第一帧相同的多个选择对指示。

23.一种用于对具有编码声道以及至少第一多声道参数和第二多声道参数的编码多声道信号进行解码的方法(400),包括:

对编码声道进行解码(402)以获得解码声道;以及

使用通过第二多声道参数标识的解码声道的第二对并且使用第二多声道参数来执行(404)多声道处理,以获得处理声道,以及使用由第一多声道参数标识的声道的第一对并且使用第一多声道参数来执行进一步的多声道处理,其中,声道的第一对包括至少一个处理

声道，

其中，所述编码多声道信号包括针对第一帧的第一多声道参数和第二多声道参数以及针对第一帧之后的第二帧的保持指示符，并且

其中，在第二帧中对与在第一帧中所使用的声音的第二对和第一对相同的第二对和第一对执行所述多声道处理和所述进一步的多声道处理。

24. 一种计算机可读存储介质，其上存储有计算机程序，所述计算机程序用于当在计算机或处理器上运行时使所述计算机或处理器执行根据权利要求22所述的对具有至少三个声道的多声道信号进行编码的方法或根据权利要求23所述的对具有编码声道以及至少第一多声道参数和第二多声道参数的编码多声道信号进行解码的方法。

用于对多声道信号进行编码或解码的装置与方法

[0001] 本申请是申请日为2016年3月8日、国际申请号为PCT/EP2016/054900、中国申请号为“201680026823.9”、发明名称为“用于对多声道信号进行编码或解码的装置与方法”的申请的分案申请。

技术领域

[0002] 本发明涉及音频编码/解码,具体地,涉及利用声道间信号依赖性的音频编码。

背景技术

[0003] 音频编码是关于处理音频信号中的冗余和无关的压缩的领域。在MPEG USAC[ISO/IEC 23003-3:2012-Information technology-MPEG audio technologies Part 3: Unified speech and audio coding]中,两声道的联合立体声编码通过使用复预测、MPS 2-1-2或具有有限频带或全频带残差信号的统一立体声来进行。MPEG surround[ISO/IEC23003-1:2007-Information technology-MPEG audio technologies Part1:MPEG Surround]在有或没有残差信号传输的情况下针对多声道音频的联合编码分层地结合OTT与TTT盒。MPEG-H四声道元素分层地应用MPS 2-1-2立体声盒,之后是构建固定的4×4再混合树的复预测/MS立体声盒。AC4[ETSI TS 103 190V1.1.1 (2014-04) -Digital Audio Compression (AC-4) Standard]引入了新的3-声道、4-声道、5-声道元素,所述元素考虑经由被传输的混合矩阵与后续的联合立体声编码信息而对被传输声道进行再混合。此外,现有出版物建议使用正交变换,如针对增强的多声道音频编码的Karhunen-Loeve变换 (KLT) [Yang,Dai and Ai,Hongmei and Kyriakakis,Chris and Kuo,C.-C.Jay,2001:Adaptive Karhunen-Loeve Transform for Enhanced Multichannel Audio Coding,<http://ict.usc.edu/pubs/Adaptive%20Karhunen-Loeve%20Transform%20for%20Enhanced%20Multichannel%20Audio%20Coding.pdf>]。

[0004] 在3D音频上下文中,扬声器声道分布在多个高度层中,导致水平与竖直的声道对。如在USAC中所定义的仅两个声道的联合编码不足以考虑声道之间的空间和感知关系。MPEG环绕被应用于附加的预/后处理步骤,残差信号在没有联合立体声编码的可能性的情况下被单独传输,以例如利用左与右竖直残差信号之间的依赖性。在AC-4中,专用的N声道元素被引入以考虑联合编码参数的高效率编码,但无法考虑针对新的浸入回放场景 (7.1+4, 22.2) 而提出的具有更多声道的一般扬声器设定。MPEG-H四声道元素还被限制为仅仅4个声道并且无法动态地应用于任意的声道而只能应用于预配置且固定数量的声道。

发明内容

[0005] 本发明的目的在于提供改善的编码/解码构思。

[0006] 该目的通过根据一种用于对具有至少三个声道的多声道信号进行编码的装置而达到;或通过一种用于对具有编码声道以及至少第一和第二多声道参数的编码多声道信号进行解码的装置而达到;或通过一种用于对具有至少三个声道的多声道信号进行编码方法

而达到；或通过一种用于对具有编码声道以及至少第一和第二多声道参数的编码多声道信号进行解码的方法而达到；或通过一种计算机程序而达到。

[0007] 实施例提供了一种对具有至少三个声道的多声道信号进行编码的装置。所述装置包括迭代处理器、声道编码器和输出接口。迭代处理器被配置为在第一迭代步骤中计算在至少三个声道的每对之间的声道间相关值，在第一迭代步骤中选择具有最高值或具有高于阈值的值的对，并且通过使用多声道处理操作而处理所选择的对，以导出针对所选择的第一多声道参数并导出第一处理声道。此外，迭代处理器被配置为在第二迭代步骤中通过使用所述处理声道的至少一个而执行所述计算、所述选择和所述处理，以导出第二多声道参数和第二处理声道。声道编码器被配置为对根据通过迭代处理器执行的迭代处理得到的声道进行编码以获得编码声道。输出接口被配置为产生具有编码声道以及第一和第二多声道参数的编码多声道信号。

[0008] 另外，实施例提供了一种用于对编码多声道信号进行解码的装置，所述编码多声道信号具有编码声道以及至少第一和第二多声道参数。所述装置包括声道解码器以及多声道处理器。声道解码器被配置为对编码声道进行解码以获得解码声道。多声道处理器被配置为通过使用由第二多声道参数标识的解码声道的第二对和通过使用第二多声道参数来执行多声道处理，以获得处理声道，并通过使用由第一多声道参数标识的声道的第一对和使用第一多声道参数来执行进一步的多声道处理，其中声道的第一对包括至少一个处理声道。

[0009] 与使用固定信号路径(例如立体声编码树)的常见多声道编码构思相比，本发明的实施例使用动态信号路径，其适应多声道输入信号的至少三个输入声道的特性。具体地说，迭代处理器102可适于在第一迭代步骤中基于至少三个声道CH1至CH3中的每对之间的声道间相关值建立信号路径(例如立体声树)，以在第一迭代步骤中选择具有最高值或高于阈值的值的对，并且在第二迭代步骤中基于在所述至少三个声道的每对与相应的先前处理声道之间的声道间相关值，在第二迭代步骤中选择具有最高值或高于阈值的值的对。

[0010] 进一步的实施例提供了一种用于对具有至少三个声道的多声道信号进行编码的方法。该方法包括：

[0011] -在第一迭代步骤中计算至少三个声道中的每对之间的声道间相关值，在第一迭代步骤中选择具有最高值或具有高于阈值的值的对，并且通过使用多声道处理操作而处理所选择的对，以导出针对所选择的对的第一多声道参数并导出第一处理声道；

[0012] -在第二迭代步骤中通过使用所述处理声道的至少一个而执行所述计算、所述选择和所述处理，以导出第二多声道参数和第二处理声道；

[0013] -对根据通过迭代处理器执行的迭代处理得到的声道进行编码以获得编码声道；以及

[0014] -产生具有编码声道以及第一和第二多声道参数的编码多声道信号。

[0015] 进一步的实施例提供了一种用于对具有编码声道以及至少第一和第二多声道参数的编码多声道信号进行解码的方法。该方法包括：

[0016] -对编码声道进行解码以获得解码声道；和

[0017] -通过使用由第二多声道参数标识的解码声道的第二对和通过使用第二多声道参数来执行多声道处理，以获得处理声道，并通过使用由第一多声道参数标识的声道的第一

对和使用第一多声道参数来执行进一步的多声道处理,其中声道的第一对包括至少一个处理声道。

附图说明

[0018] 这里通过参考附图来描述本发明的实施例。

[0019] 图1示出了根据实施例的对具有至少三个声道的多声道信号进行编码的装置的示意性框图。

[0020] 图2示出了根据实施例的对具有至少三个声道的多声道信号进行编码的装置的示意性框图。

[0021] 图3示出了根据实施例的立体声盒的示意性框图。

[0022] 图4示出了根据实施例的对具有编码声道以及至少第一和第二多声道参数的编码多声道信号进行解码的装置的示意性框图。

[0023] 图5示出了根据实施例的对具有至少三个声道的多声道信号进行编码的方法的流程图。

[0024] 图6示出了根据实施例的对具有编码声道以及至少第一和第二多声道参数的编码多声道信号进行解码的方法的流程图。

具体实施方式

[0025] 具有相同或等效功能的相同或等效组件在下面的描述中由相同或等效的附图标记来表示。

[0026] 在下面的描述中,阐述多个细节以提供对本发明的实施例的更完全的解释。然而,本领域技术人员将清楚,可以在不具有这些具体细节的情况下实现本发明的实施例。在其他示例中,公知结构和设备以框图形式而不是详细地被示出,以避免模糊本发明的实施例。此外,在下文中描述的不同实施例可以彼此结合,除非另外详细说明。

[0027] 图1示出了对具有至少三个声道CH1至CH3的多声道信号101进行编码的装置(编码器)100的示意性框图。装置100包括迭代处理器102、声道编码器104以及输出接口106。

[0028] 迭代处理器102被配置为在第一迭代步骤中计算至少三个声道CH1至CH3中的每对之间的声道间相关值,在第一迭代步骤中选择具有最高值或具有高于阈值的值的对,并且通过使用多声道处理操作而处理所选择的对,以导出针对所选择的对的第一多声道参数MCH_PAR1并导出第一处理声道P1和P2。此外,迭代处理器102被配置为在第二迭代步骤中通过使用处理声道P1或P2的至少一个而执行该计算、该选择和该处理,以导出第二多声道参数MCH_PAR2与第二处理声道P3和P4。

[0029] 例如,如图1所示,迭代处理器102可在第一迭代步骤中计算至少三个声道CH1至CH3中的第一对之间的声道间相关值、至少三个声道CH1至CH3中的第二对之间的声道间相关值、和至少三个声道CH1至CH3中的第三对之间的声道间相关值,其中第一对由第一声道CH1与第二声道CH2组成,第二对由第二声道CH2与第三声道CH3组成,第三对由第一声道CH1与第三声道CH3组成。

[0030] 在图1中,假设由第一声道CH1与第三声道CH3组成的第三对包括最高声道间相关值,从而迭代处理器102在第一迭代步骤中选择具有最高声道间相关值的第三对并且通过

使用多声道处理操作来处理所选择的对,即第三对,以导出针对所选择的对的第一多声道参数MCH_PAR1并且导出第一处理声道P1和P2。

[0031] 此外,迭代处理器102是可被配置为在第二迭代步骤中计算在至少三个声道CH1至CH3的每对之间的声道间相关值与处理声道P1和P2,以用于在第二迭代步骤中选择具有最高声道间相关值或具有高于阈值的值的对。由此,迭代处理器102可被配置为在第二迭代步骤中(或在任何进一步的迭代步骤中)不选择在第一迭代步骤中所选择的对。

[0032] 参考图1所示的示例,迭代处理器102还可以计算声道的由第一声道CH1与第一处理声道P1所组成的第四对之间的声道间相关值、由第一声道CH1与第二处理声道P2所组成的第五对之间的声道间相关值、由第二声道CH2与第一处理声道P1所组成的第六对之间的声道间相关值、由第二声道CH2与第二处理声道P2所组成的第七对之间的声道间相关值、由第三声道CH3与第一处理声道P1所组成的第八对之间的声道间相关值、由第三声道CH3与第二处理声道P2所组成的第九对之间的声道间相关值以及由第一处理声道P1与第二处理声道P2所组成的第十对之间的声道间相关值。

[0033] 在图1中,假设在第二迭代步骤中由第二声道CH2与第一处理声道P1所组成的第六对具有最高声道间相关值,使得迭代处理器102在第二迭代步骤中选择第六对并且通过使用多声道处理操作来处理所选择的对,即第六对,以导出针对所选择的对的第二多声道参数MCH_PAR2并且导出第二处理声道P3和P4。

[0034] 迭代处理器102可被配置为仅在对的电平差(level difference) 小于阈值时选择该对,该阈值小于40dB、25dB、12dB、或小于6dB。由此,25dB或40dB的阈值对应于3度或0.5度的旋转角度。

[0035] 迭代处理器102可被配置为计算归一化(normalized)整数相关值,其中迭代处理器102可被配置为当整数相关值大于例如0.2或优选为0.3时选择对。

[0036] 此外,迭代处理器102可向声道编码器104提供由多声道处理所产生的声道。例如,参考图1,迭代处理器102可向声道编码器104提供由在第二迭代步骤中执行的多声道处理所产生的第三处理声道P3和第四处理声道P4,以及由在第一迭代步骤中执行的多声道处理所产生的第二处理声道P2。因此,迭代处理器102可仅向声道编码器104提供那些未在后续迭代步骤中被(进一步)处理的处理声道。如图1所示,第一处理声道P1未被提供给声道编码器104,这是因为它在第二迭代步骤中被另外处理。

[0037] 声道编码器104可被配置为对由迭代处理器102执行的迭代处理(或多声道处理)所产生的声道P2至P4进行编码,以获得编码声道E1至E3。

[0038] 例如,声道编码器104可被配置为使用单声道编码器(或单声道盒或单声道工具)120_1至120_3来对由迭代处理(或多声道处理)产生的声道P2至P4进行编码。单声道盒可被配置为编码声道,使得与对具有较高能量(或较高振幅)的声道编码相比,对具有较少能量(或较小振幅)的声道编码需要较少的比特。单声道盒120_1至120_3可例如为基于变换的音频编码器。此外,声道编码器104可被配置为使用立体声编码器(例如参数立体声编码器或损耗立体声编码器)来对由迭代处理(或多声道处理)产生的声道P2至P4进行编码。

[0039] 输出接口106可被配置为产生编码多声道信号107,所述编码多声道信号107具有编码声道E1至E3以及第一与第二多声道参数MCH_PAR1、MCH_PAR2。

[0040] 例如,输出接口106可被配置为将编码多声道信号107产生为串行信号(serial

signal)或串行比特流(serial bit stream),从而第二多声道参数MCH_PAR2在编码信号107内并位于第一多声道参数MCH_PAR1之前。因此,解码器(其实施例将在下文结合图4描述)将在第一多声道参数MCH_PAR1之前接收第二多声道参数MCH_PAR2。

[0041] 在图1中,迭代处理器102示例性地执行两个多声道处理操作,其中多声道处理操作在第一迭代步骤中,而另一多声道处理操作在第二迭代步骤中。自然地,迭代处理器102还可在后续的迭代步骤中执行其他的多声道处理操作。因此,迭代处理器102可被配置为执行迭代步骤直到达到了迭代终止标准。迭代终止标准可以是迭代步骤的最大数量等于多声道信号101的声音总数量或比多声道信号101的声音总数量大两个,或者其中迭代终止标准可以是当声道间相关值不具有大于阈值的值时,而阈值优选大于0.2或阈值优选等于0.3。在其他实施例中,迭代终止标准可以是迭代步骤的最大数量等于或大于多声道信号101的声音总数量,或者其中迭代终止标准是为当声道间相关值不具有大于阈值的值时,而阈值优选大于0.2或阈值优选等于0.3。

[0042] 为了说明目的,由迭代处理器102在第一迭代步骤与第二迭代步骤中执行的多声道处理操作示例性地由图1中的处理盒110、112来说明。处理盒110、112可以硬件或软件实现。处理盒110、112可例如为立体声盒。

[0043] 因此,声道间信号依赖性可通过分层地应用已知的联合立体声编码工具而被利用。与先前的MPEG方法相比,要被处理的信号不由固定信号路径(例如立体声编码树)预先确定,而是可被动态地改变以适应于输入信号特性。实际立体声盒的输入可为(1)未处理声道,例如声道CH1至CH3、(2)前立体声盒的输出,例如被处理的信号P1至P4、或(3)未处理声道与前立体声盒的输出的组合。

[0044] 在立体声盒110、112内的处理可为基于预测的(如在USAC中的复预测盒)或基于KLT/PCA的(输入声道在编码器中被旋转(例如经由 2×2 旋转矩阵)以最大化能量浓缩,即将信号能量集中于一个声道内,在解码器内被旋转信号将会再被变换到原始输入信号方向)。

[0045] 在编码器100的可能的实现中,(1)该编码器计算在各声道对之间的声道间相关性并从输入信号选择合适的信号对,并且将立体声工具应用于所选择声道;(2)该编码器再计算在所有声道(未处理声道以及被处理的中间输出声道)之间的声道间相关性并从输入信号选择合适的信号对,并且将立体声工具应用于所选择声道;以及(3)该编码器重复步骤(2)直到所有声道间相关性低于阈值或如果应用了最大数量的变换。

[0046] 如前所述,要被编码器100(更精确地说是迭代处理器102)处理的信号对不被固定信号路径(例如立体声编码树)所确定,而是可以动态地改变以适应于输入信号特性。因此,编码器100(或迭代处理器102)可被配置为根据多声道(输入)信号101的至少三个声道CH1至CH3而构建立体声树。换言之,编码器100(或迭代处理器102)可被配置为基于声道间相关性而构建立体声树(例如通过在第一迭代步骤中计算在至少三个声道CH1至CH3的每对之间的声道间相关值,以在第一迭代步骤中选择具有最高值或高于阈值的值的对,并且通过在第二迭代步骤中计算在至少三个声道的每对之间的声道间相关值以在第二迭代步骤中选择具有最高值或高于阈值的值的对)。根据单步骤方法,可针对可能包含所有在先前迭代中可能被处理的声道的相关性的每个迭代而计算相关矩阵。

[0047] 如前所述,迭代处理器102可被配置为在第一迭代步骤中导出针对所选择的对的第一多声道参数MCH_PAR1并在第二迭代步骤中导出针对所选择的对的第二多声道参数

MCH_PAR2。第一多声道参数MCH_PAR1可包括第一声道对标识(或索引),其标识(或标示(signal))在第一迭代步骤中被选择的声道对,其中第二多声道参数MCH_PAR2可包括第二声道对标识(或索引),其标识(或标示)在第二迭代步骤中选择的声道对。

[0048] 以下描述了输入信号的高效率索引。例如,声道对可通过根据声道的总数量使用针对每对的唯一索引而被有效地标示。例如,针对六声道的对的索引可以在下表中示出:

[0049]

	0	1	2	3	4	5
0		0	1	2	3	4
1			5	6	7	8
2				9	10	11
3					12	13
4						14
5						

[0050] 例如,在上表中,索引5可标示由第一和第二声道组成的对。相似地,索引6可标示由第一和第三声道组成的对。

[0051] 针对n个声道的可能声道对索引的总数量可如下计算:

[0052] $\text{numPairs} = \text{numChannels} \times (\text{numChannels}-1)/2$

[0053] 因此,需要用来标示声道对的比特数量为如下:

[0054] $\text{numBits} = \text{floor}(\log_2(\text{numPairs}-1))+1$

[0055] 此外,编码器100可使用声道掩码。多声道工具的配置可包含对该工具对于哪些声道是有效的加以指示的声道掩码。因此,在考虑更有效率的编码时,可从该声道对索引中移除LFE(LFE=低频效应/增强声道)。例如,对于11.1设定,在允许以6比特而非7比特进行标示的情况下,这可使声道对索引的数量从 $12 \times 11/2 = 66$ 减少到 $11 \times 10/2 = 55$ 。该机制还可被用于排除意图作为单声道对象(例如多语言音轨)的声音。关于对声道掩码的解码,可产生声道图(channelMap)以允许声道对索引再映射(re-mapping)至解码器声道。

[0056] 此外,迭代处理器102可被配置为导出针对第一帧的多个选择对指示,其中输出接口106可被配置为在多声道信号107中包括针对第一帧之后的第二帧的保持指示符,该指示符指示第二帧具有与第一帧相同的多个选择对指示。

[0057] 保持指示符或保持树标志可被配置为标示没有新的树被传送,但是最后的立体声树应被使用。如果声道相关性在一段较长时间维持不变,则这可用来避免相同的立体声树配置的多次传送。

[0058] 图2示出了立体声盒110、112的示意性框图。立体声盒110、112包括针对第一输入信号I1与第二输入信号I2的输入,并包括针对第一输出信号01与第二输出信号02的输出。如图2所示,来自输入信号I1、I2的输出信号01、02的依赖性可通过s-参数S1至S4来描述。

[0059] 迭代处理器102可使用(或包括)立体声盒110、112以执行对输入声道和/或处理声道的多声道处理操作,从而导出(进一步)处理声道。例如,迭代处理器102可被配置为使用一般性的、基于预测的或基于KLT(Karhunen-Loeve变换)的旋转立体声盒110、112。

[0060] 一般性的编码器(或编码器侧的立体声盒)可被配置为对输入信号I1、I2进行编码以获得基于下列方程式的输出信号01、02:

[0061]
$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ s_3 & s_4 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0062] 一般性的解码器(或解码器侧的立体声盒)可被配置为对输入信号I1、I2进行解码以获得基于下列方程式的输出信号O1、O2:

[0063]
$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ s_3 & s_4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0064] 基于预测的编码器(或编码器侧的立体声盒)可被配置为对输入信号I1、I2进行编码以获得基于下列方程式的输出信号O1、O2:

[0065]
$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} 1 & 1 \\ 1-p & -(1+p) \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix},$$

[0066] 其中p是为预测系数。

[0067] 基于预测的解码器(或解码器侧的立体声盒)可被配置为对输入信号I1、I2进行解码以获得基于下列方程式的输出信号O1、O2:

[0068]
$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} 1+p & 1 \\ 1-p & -1 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0069] 基于KLT的旋转编码器(或编码器侧的立体声盒)可被配置为对输入信号I1、I2进行编码以获得基于下列方程式的输出信号O1、O2:

[0070]
$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0071] 基于KLT的旋转解码器(或解码器侧的立体声盒)可被配置为对输入信号I1、I2进行解码以获得基于下列方程式的输出信号O1、O2(反向旋转):

[0072]
$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0073] 在下文中,描述了对基于KLT的旋转的旋转角度 α 的计算。

[0074] 针对基于KLT的旋转的旋转角度 α 可如下定义:

[0075]
$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{2c_{12}}{c_{11} - c_{22}} \right),$$

[0076] 其中 c_{xy} 是为非归一化相关矩阵的元素,其中 c_{11} 、 c_{22} 是为声道能量。

[0077] 这可通过使用atan2函数来实现,以考虑到在分子中的负相关之间的区别以及在分母中的负能量差:

[0078] $\text{alpha} = 0.5 \times \text{atan2}(2 \times \text{correlation}[\text{ch1}][\text{ch2}],$

[0079] $(\text{correlation}[\text{ch1}][\text{ch1}] - \text{correlation}[\text{ch2}][\text{ch2}]))$ 。

[0080] 此外,迭代处理器102可被配置为通过使用包括多个频带在内的每个声道的帧而计算声道间相关性,从而可获得针对多个频带的单个声道间相关值,其中迭代处理器102可被配置为针对多个频带中的每个频带执行多声道处理,使得可从每个频带获得第一或第二多声道参数。

[0081] 因此,迭代处理器102可被配置为计算在多声道处理中的立体声参数,其中迭代处理器102可被配置为仅在立体声参数高于由立体声量化器(例如基于KLT的旋转编码器)所

定义的量化到零阈值的频带中执行立体声处理。立体声参数可例如为MS开/关、或旋转角度、或预测系数。

[0082] 例如,迭代处理器102可被配置为计算在多声道处理中的旋转角度,其中迭代处理器102可被配置为仅在旋转角度高于由立体声量化器(例如基于KLT的旋转编码器)所定义的量化到零阈值的频带中执行旋转处理。

[0083] 因此,编码器100(或输出接口106)可被配置为发送变换/旋转信息,作为针对完整频谱(全频盒(full band box))的参数或针对部分频谱的多个频率依赖参数。

[0084] 编码器100可被配置为基于下表而产生比特流107:

[0085] 表1-mpegh3daExtElementConfig() 的语法

语法	比特数量	助记码
<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: /* No configuration element */ break; case ID_EXT_ELE_MPEGS: SpatialSpecificConfig(); break; } }</pre>	1	uimsbf

[0086]

[0087]

```

case ID_EXT_ELE_SAOC:
    SAOCspecificConfig();
    break;
case ID_EXT_ELE_AUDIOPREROLL:
    /* No configuration element */
    break;
case ID_EXT_ELE_UNI_DRC:
    mpegh3daUniDrcConfig();
    break;
case ID_EXT_ELE_OBJ_METADATA:
    ObjectMetadataConfig();
    break;
case ID_EXT_ELE_SAOC_3D:
    SAOCH3DSpecificConfig();
    break;
case ID_EXT_ELE_HOA:
    HOAConfig();
    break;
case ID_EXT_ELE_MCC: /* multi channel coding */
    MCCConfig(grp);
    break;
case ID_EXT_ELE_FMT_CNRVRTR
    /* No configuration element */
    break;
default:
    while (usacExtElementConfigLength--) {
        tmp;
    }
    break;
}
}

```

NOTE

8 uimsbf

注意：针对usacExtElementType的默认元素用于未知的extElementTypes，使得传统解码器可妥善处理未来扩展。

[0088]

表21-MCCConfig()的语法

语法	比特数量	助记码
<pre> MCCConfig(grp) { nChannels = 0 for(chan=0;chan < bsNumberOfSignals[grp]; chan++) chanMask[chan] if(chanMask[chan] > 0) { mctChannelMap[nChannels]=chan; nChannels++; } } } </pre>	1	

注意：相应的ID_USAC_EXT元素应先于特定信号组 grp 的任何音频元素。

[0090]

表32-MultichannelCodingBoxBandWise()的语法

语法	比特数量	助记码
<code>MultichannelCodingBoxBandWise()</code>		
<code>{</code>		
<code>for(pair=0; pair<numPairs;pair++) {</code>		
<code>if (keepTree == 0) {</code>		
<code> channelPairIndex[pair]</code>	<code>nBits</code> NOTE 1	
<code>}</code>		
<code>else {</code>		
<code> channelPairIndex[pair]=</code>		
<code> lastChannelPairIndex[pair];</code>		
<code>}</code>		
<code> hasMctMask</code>	1	
<code> hasBandwiseAngles</code>	1	
<code> if (hasMctMask hasBandwiseAngles) {</code>		
<code> isShort</code>	1	
<code> numMaskBands;</code>	5	
<code> if (isShort) {</code>		
<code> numMaskBands = numMaskBands*8</code>		
<code> }</code>		
<code> } else {</code>		NOTE 2)
<code> numMaskBands = MAX_NUM_MC_BANDS;</code>		
<code> }</code>		
<code> if (hasMctMask) {</code>		
<code> for(j=0;j<numMaskBands;j++) {</code>		
<code> msMask[pair][j];</code>	1	
<code> }</code>		
<code> } else {</code>		
<code> for(j=0;j<numMaskBands;j++) {</code>		
<code> msMask[pair][j] = 1;</code>		
<code> }</code>		
<code> }</code>		
<code> }</code>		
<code> if(indepFlag > 0) {</code>		
<code> delta_code_time = 0;</code>		
<code> } else {</code>		
<code> delta_code_time;</code>	1	
<code> }</code>		
<code> if (hasBandwiseAngles == 0) {</code>		
<code> hcoc_angle[dpcm_alpha[pair][0]];</code>	1..10	vlclbf
<code> }</code>		
<code> else {</code>		
<code> for(j=0;j< numMaskBands;j++) {</code>		
<code> if (msMask[pair][j] ==1) {</code>		
<code> hcoc_angle[dpcm_alpha[pair][j]];</code>	1..10	vlclbf
<code> }</code>		
<code> }</code>		
<code> }</code>		
<code> }</code>		
<code>}</code>		
[0092]	注意： 1) nBits = floor(log2(nChannels*(nChannels-1)/2 - 1))+1	

[0093] 表4-MultichannelCodingBoxFullband() 的语法

语法	比特数量	助记码
<pre>MultichannelCodingBoxFullband() { for (pair=0; pair<numPairs; pair++) { If(keepTree == 0) { channelPairIndex[pair] } else { numPairs = lastNumPairs; } alpha; } }</pre> <p>注意: 1) nBits = floor(log2(nChannels × (nChannels-1)/2 - 1))+1</p>	nBits NOTE 1)	8

[0095] 表5-MultichannelCodingFrame() 的语法

语法	比特数量	助记码
<pre>MultichannelCodingFrame() { MCCSignalingType keepTree if(keepTree==0) { numPairs } else { numPairs=lastNumPairs; } if(MCCSignalingType == 0) /* tree of standard stereo boxes */ for(i=0;i<numPairs;i++) { MCCBox[i] = StereoCoreToolInfo(0); } if(MCCSignalingType == 1) /* arbitrary mct trees */ MultichannelCodingBoxBandWise(); } if(MCCSignalingType == 2) /* transmitted trees */ } if(MCCSignalingType == 3) /* simple fullband tree */ MultichannelCodingBoxFullband(); }</pre>	2 1 5	

[0098] 表6-usacExtElementType的值

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_CNRVRTR	8
ID_EXT_ELE_MCC	9 或 10
/*为 ISO 使用预留*/	10-127
/*为了在 ISO 范围外使用预留*/	128 和更高

注意：特殊应用的 usacExtElementType 值被强制在为了 ISO 范围之外的使用所保留的空间。当解码器需要结构的最小值以跳过这些扩展时，这些被解码器跳过。

[0100] 表7-用于扩展有效载荷解码的数据块的解释

usacExtElementType	连接的usacExtElementSegmentData表示：
ID_EXT_ELE_FILL	Series of fill_byte
ID_EXT_ELE_MPEGS	SpatialFrame()
ID_EXT_ELE_SAOC	SaocFrame()
ID_EXT_ELE_AUDIOPREROLL	AudioPreRo ()
ID_EXT_ELE_UNI_DRC	uniDrcGain() as defined in ISO/IEC 23003-4
ID_EXT_ELE_OBJ_METADATA	object_metadata()
ID_EXT_ELE_SAOC_3D	SaoCH3DFrame()
ID_EXT_ELE_HOA	HOAFrame()
ID_EXT_ELE_FMT_CNRVRTR	FormatConverterFrame()
ID_EXT_ELE_MCC	MultichannelCodingFrame()
未知	未知数据，该数据块应被丢弃。

[0102] 图3示出了根据实施例的迭代处理器102的示意性框图。在图3所示的实施例中，多声道信号101是为5.1声道信号，其具有六个声道：左声道L、右声道R、左环绕声道Ls、右环绕声道Rs、中间声道C以及低频效果声道LFE。

[0103] 如图3所示，LFE声道不被迭代处理器102处理。这可能是由于在LFE声道与其他五个声道L、R、Ls、Rs、C中的每个之间的声道间相关值太小，或是由于声道掩码指示不要处理LFE声道，这种情况在下文中假设。

[0104] 在第一迭代步骤中，迭代处理器102计算五个声道L、R、Ls、Rs、C中的每对之间的声道间相关值，以在第一迭代步骤中选择具有最高值或具有高于阈值的值的对。在图3中，假设左声道L与右声道R具有最高值，使得迭代处理器102通过使用执行多声道操作处理操作的立体声盒(或立体声工具)110而处理左声道L与右声道R，以导出第一和第二处理声道P1和P2。

[0105] 在第二迭代步骤中，迭代处理器102计算五个声道L、R、Ls、Rs、C的每对之间与处理声道P1和P2之间的声道间相关值，以在第二迭代步骤中选择具有最高值或具有高于阈值的值的对。在图3中，假设左环绕声道Ls与右环绕声道Rs具有最高值，使得迭代处理器102通过

使用立体声盒(或立体声工具)112而处理左环绕声道Ls与右环绕声道Rs,以导出第三和第四处理声道P3和P4。

[0106] 在第三迭代步骤中,迭代处理器102计算五个声道L、R、Ls、Rs、C的每对之间与处理声道P1至P4之间的声道间相关值,以在第三迭代步骤中选择具有最高值或具有高于阈值的值的对。在图3中,假设第一处理声道P1与第三处理声道P3具有最高值,使得迭代处理器102通过使用立体声盒(或立体声工具)114而处理第一处理声道P1与第三处理声道P3,以导出第五和第六处理声道P5和P6。

[0107] 在第四迭代步骤中,迭代处理器102计算五个声道L、R、Ls、Rs、C的每对之间与处理声道P1至P6之间的声道间相关值,以在第四迭代步骤中选择具有最高值或具有高于阈值的值的对。在图3中,假设第五处理声道P5与中间声道C具有最高值,使得迭代处理器102通过使用立体声盒(或立体声工具)115而处理第五处理声道P5与第中间声道C,以导出第七和第八处理声道P7和P8。

[0108] 立体声盒110至116可以是MS立体声盒,即被配置为提供中声道与侧声道的中/侧立体声盒。中声道可以是立体声盒的输入声道的总和,其中侧声道可以是立体声盒的输入声道之间的差。此外,立体声盒110、116可以是旋转盒或立体声预测盒。

[0109] 在图3中,第一处理声道P1、第三处理声道P3与第五处理声道P5可以是中声道,其中第二处理声道P2、第四处理声道P4与第六处理声道P6可以是侧声道。

[0110] 另外,如图3所示,迭代处理器102可被配置为执行在第二迭代步骤中的计算、选择和处理,并且如果可行的话,在任何进一步的迭代步骤中使用输入声道L、R、Ls、Rs、C和处理声道的(仅仅)中声道P1、P3、P5。换言之,迭代处理器102可被配置为在第二迭代步骤中的计算、选择和处理中不使用处理声道的侧声道P1、P3、P5,并且如果可行的话,也不在任何进一步的迭代步骤中使用该所选择的对。

[0111] 图4示出了对具有编码声道E1至E3以及至少第一和第二多声道参数MCH_PAR1和MCH_PAR2的编码多声道信号107进行解码的装置(解码器)200的示意性框图。装置200包括声道解码器202与多声道处理器204。

[0112] 声道解码器202被配置为对编码声道E1至E3进行解码以获得解码声道D1至D3。

[0113] 例如,声道解码器202可包括至少三个单声道解码器(或单声道盒、或单声道工具)206_1至206_3,其中单声道解码器206_1至206_3中的每个可被配置为对至少三个编码声道E1至E3之一进行解码,以获得相应的解码声道E1至E3。单声道解码器206_1至206_3可例如为基于变换的音频解码器。

[0114] 多声道处理器204被配置为通过使用由第二多声道参数MCH_PAR2所标识的解码声道的第二对并通过使用第二多声道参数MCH_PAR2来执行多声道处理,以获得处理声道,并且通过使用由第一多声道参数MCH_PAR1所标识的声道的第一对并通过使用第一多声道参数MCH_PAR1来执行进一步的多声道处理,其中声道的第一对包括至少一个处理声道。

[0115] 如图4所示的示例,第二多声道参数MCH_PAR2可指示(或标示)解码声道的第二对由第一解码声道D1和第二解码声道D2组成。因此,多声道处理器204通过使用由第一解码声道D1和第二解码声道D2(由第二多声道参数MCH_PAR2标识)组成的解码声道的第二对并通过使用第二多声道参数MCH_PAR2来执行多声道处理,以获得处理声道P1*和P2*。第一多声道参数MCH_PAR1可指示解码声道的第一对由第一处理声道P1*和第三解码声道D3组成。因

此,多声道处理器204通过使用由第一处理声道P1*和第三解码声道D3(由第一多声道参数MCH_PAR1标识)组成的解码声道的第一对并通过使用第一多声道参数MCH_PAR1来执行多声道处理,以获得处理声道P3*和P4*。

[0116] 此外,多声道处理器204可提供第三处理声道P3*作为第一声道CH1、第四处理声道P4*作为第三声道CH3、以及第二处理声道P2*作为第二声道CH2。

[0117] 假设图4所示的解码器200从图1所示的编码器100接收编码多声道信号107,解码器200的第一解码声道D1可等效于编码器100的第三处理声道P3,其中解码器200的第二解码声道D2可等效于编码器100的第四处理声道P4,并且解码器200的第三解码声道D3可等效于编码器100的第二处理声道P2。另外,解码器200的第一处理声道P1*可等效于编码器100的第一处理声道P1。

[0118] 另外,编码多声道信号107可为串行信号,其中第二多声道参数MCH_PAR2在第一多声道参数MCH_PAR1之前在解码器200处被接收。在此情况下,多声道处理器204可被配置为按照多声道参数MCH_PAR1、MCH_PAR2被解码器接收的顺序来处理解码声道。在图4所示的示例中,解码器在第一多声道参数MCH_PAR1之前接收第二多声道参数MCH_PAR2,并且因此在通过使用由第一多声道参数MCH_PAR1标识的解码声道的第一对(由第一处理声道P1*和第三解码声道D3组成)来执行多声道处理之前,通过使用由第二多声道参数MCH_PAR2标识的解码声道的第二对(由第一和第二解码声道D1和D2组成)来执行多声道处理。

[0119] 在图4中,多声道处理器204示例性地执行两个多声道处理操作。为了说明目的,由多声道处理器204执行的多声道处理操作在图4中以处理盒208、210来说明。处理盒208、210可以硬件或软件来实现。处理盒208、210可例如为如上文关于编码器100所述的立体声盒,例如一般解码器(或解码器侧的立体声盒)、基于预测的解码器(或解码器侧的立体声盒)、或基于KLT的旋转解码器(或解码器侧的立体声盒)。

[0120] 例如,编码器100可使用基于KLT的旋转编码器(或编码器侧的立体声盒)。在此情况下,编码器100可导出第一和第二多声道参数MCH_PAR1、MCH_PAE2,使得第一和第二多声道参数MCH_PAR1、MCH_PAE2包括旋转角度。旋转角度可被差分编码。因此,解码器200的多声道处理器204可包括差分解码器以用于对被差分编码的旋转角度进行差分解码。

[0121] 装置200还可以包括输入接口212,其被配置为接收并处理编码多声道信号107,以向声道解码器202提供编码声道E1至E3并向多声道处理器204提供第一和第二多声道参数MCH_PAR1、MCH_PAR2。

[0122] 就如前述,保持指示符(或保持树标志)可被用来标示没有新的树要被传送,但是最后的立体声树应被使用。如果声道相关性在一段较长时间维持不变,则这可用来避免相同的立体声树配置的多次传送。

[0123] 因此,当编码多声道信号107包括针对第一帧的第一或第二多声道参数MCH_PAR1、MCH_PAR2以及针对第一帧之后的第二帧的保持指示符时,多声道处理器204可被配置为对如同在第一帧中所使用的声道的第二对或第一对执行在第二帧中的多声道处理或进一步的多声道处理。

[0124] 多声道处理与进一步的多声道处理可包括使用立体声参数的立体声处理,其中针对各个比例因子频带(scale factor band)或解码声道D1至D3的比例因子频带的组,第一立体声参数被包括在第一多声道参数MCH_PAR1中并且第二立体声参数被包括在第二多声

道参数MCH_PAR2中。因此,第一立体声参数与第二立体声参数可属于相同类型,例如旋转角度或预测系数。自然地,第一立体声参数与第二立体声参数可属于不同类型。例如,第一立体声参数可以是旋转角度,其中第二立体声参数可以是预测系数,反的亦可。

[0125] 另外,第一或第二多声道参数MCH_PAR1、MCH_PAR2可包括多声道处理掩码,其指示哪些比例因子频带经过多声道处理以及哪些比例因子频带未经过多声道处理。因此,多声道处理器204可被配置为不执行在由多声道处理掩码指示的比例因子频带中的多声道处理。

[0126] 第一与第二多声道参数MCH_PAR1、MCH_PAR2可分别包括声道对标识(或索引),其中多声道处理器204可被配置为通过使用预定义的解码规则或在编码多声道信号中所指示的解码规则而对该声道对标识(或索引)进行解码。

[0127] 例如,声道对可通过根据声道的总数量来使用针对每对的唯一索引而被有效地标识,如上文关于编码器100的描述。

[0128] 此外,解码规则可以是霍夫曼(Huffman)解码规则,其中多声道处理器204可被配置为对声道对标识执行霍夫曼解码。

[0129] 编码多声道信号107还可以包括多声道处理允许指示符,其指示多声道处理可被允许的解码声道的仅仅子组并且指示多声道处理不被允许的至少一个解码声道。因此,多声道处理器204可被配置为不对该至少一个解码声道执行任何多声道处理,对于该至少一个解码声道而言,如多声道处理允许指示符所指示的那样,多声道处理不被允许。

[0130] 例如,当多声道信号为5.1声道信号时,多声道处理允许指示符可指示多声道处理仅允许用于5声道,即右声道R、左声道L、右环绕声道Rs、左环绕声道Ls和中间声道C,其中多声道处理不被允许用于LFE声道。

[0131] 对于解码处理(声道对索引的解码),可使用下面的c代码。因此,对于所有的声道对,需要具有有效KLT处理的声音的数量(nChannels)以及当前帧的声道对的数量(numPairs)。

```
maxNumPairIdx = nChannels*(nChannels-1)/2 - 1;
```

```
numBits = floor(log2(maxNumPairIdx))+1;
```

[0132] pairCounter = 0;

```
for (chan1=1; chan1 < nChannels; chan1++) {
```

```
for (chan0=0; chan0 < chan1; chan0++) {  
    if (pairCounter == pairIdx) {  
        channelPair[0] = chan0;  
        channelPair[1] = chan1;  
        return;  
    }  
    else  
        pairCounter++;  
}  
}  
}
```

[0134] 为了对针对非分频段的角度的预测系数进行解码,可使用下面的c代码:

```
for(pair=0; pair<numPairs; pair++) {  
    mctBandsPerWindow = numMaskBands[pair]/windowsPerFrame;
```

```
    if(delta_code_time[pair] > 0) {  
        lastVal = alpha_prev_fullband[pair];  
    } else {  
        lastVal = DEFAULT_ALPHA;  
    }
```

[0135]

```
    newAlpha = lastVal + dpcm_alpha[pair][0];  
    if(newAlpha >= 64) {  
        newAlpha -= 64;  
    }  
  
    for (band=0; band < numMaskBands; band++) {  
        /* set all angles to fullband angle */
```

```
pairAlpha[pair][band] = newAlpha;

/* set previous angles according to mctMask */

if(mctMask[pair][band] > 0) {

    alpha_prev_frame[pair][band%mctBandsPerWindow] = newAlpha;

}

else {

    alpha_prev_frame[pair][band%mctBandsPerWindow] = DEFAULT_ALPHA;

}

}

alpha_prev_fullband[pair] = newAlpha;

for(band=bandsPerWindow ; band<MAX_NUM_MC_BANDS; band++) {

    alpha_prev_frame[pair][band] = DEFAULT_ALPHA;

}

}

[0136] [0137] 为了对针对非分频段的KLT角度的预测系数进行解码,可使用下面的c代码:
for(pair=0; pair<numPairs; pair++) {

    mctBandsPerWindow = numMaskBands[pair]/windowsPerFrame;

    for(band=0; band<numMaskBands[pair]; band++) {

        if(delta_code_time[pair] > 0) {

            lastVal = alpha_prev_frame[pair][band%mctBandsPerWindow];

        }

    }

}

[0138] else {

    if ((band % mctBandsPerWindow) == 0) {

        lastVal = DEFAULT_ALPHA;

    }

}
```

```

}

if (msMask[pair][band] > 0) {

    newAlpha = lastVal + dpcm_alpha[pair][band];

    if(newAlpha >= 64) {

        newAlpha -= 64;

    }

    pairAlpha[pair][band] = newAlpha;

    alpha_prev_frame[pair][band%mctBandsPerWindow] = newAlpha;

    lastVal = newAlpha;

}

[0139] else {

    alpha_prev_frame[pair][band%mctBandsPerWindow] = DEFAULT_ALPHA; /*

-45° */

}

/* reset fullband angle */

alpha_prev_fullband[pair] = DEFAULT_ALPHA;

}

for(band=bandsPerWindow ; band<MAX_NUM_MC_BANDS; band++) {

    alpha_prev_frame[pair][band] = DEFAULT_ALPHA;

}

}

```

[0140] 为了避免不同平台上的三角函数的浮点差异,可使用下面的用于将角度索引直接转换成sin/cos的查找表:

[0141] tabIndexToSinAlpha[64] = {
[0142] -1.000000f, -0.998795f, -0.995185f, -0.989177f, -0.980785f, -0.970031f, -
0.956940f, -0.941544f,
[0143] -0.923880f, -0.903989f, -0.881921f, -0.857729f, -0.831470f, -0.803208f, -
0.773010f, -0.740951f,
[0144] -0.707107f, -0.671559f, -0.634393f, -0.595699f, -0.555570f, -0.514103f, -
0.471397f, -0.427555f,
[0145] -0.382683f, -0.336890f, -0.290285f, -0.242980f, -0.195090f, -0.146730f, -
0.098017f, -0.049068f,

```
[0146] 0.000000f,0.049068f,0.098017f,0.146730f,0.195090f,0.242980f,  
0.290285f,0.336890f,  
[0147] 0.382683f,0.427555f,0.471397f,0.514103f,0.555570f,0.595699f,  
0.634393f,0.671559f,  
[0148] 0.707107f,0.740951f,0.773010f,0.803208f,0.831470f,0.857729f,  
0.881921f,0.903989f,  
[0149] 0.923880f,0.941544f,0.956940f,0.970031f,0.980785f,0.989177f,  
0.995185f,0.998795f  
[0150] } ;  
[0151] tabIndexToCosAlpha[64] = {  
[0152] 0.000000f,0.049068f,0.098017f,0.146730f,0.195090f,0.242980f,  
0.290285f,0.336890f,  
[0153] 0.382683f,0.427555f,0.471397f,0.514103f,0.555570f,0.595699f,  
0.634393f,0.671559f,  
[0154] 0.707107f,0.740951f,0.773010f,0.803208f,0.831470f,0.857729f,  
0.881921f,0.903989f,  
[0155] 0.923880f,0.941544f,0.956940f,0.970031f,0.980785f,0.989177f,  
0.995185f,0.998795f,  
[0156] 1.000000f,0.998795f,0.995185f,0.989177f,0.980785f,0.970031f,  
0.956940f,0.941544f,  
[0157] 0.923880f,0.903989f,0.881921f,0.857729f,0.831470f,0.803208f,  
0.773010f,0.740951f,  
[0158] 0.707107f,0.671559f,0.634393f,0.595699f,0.555570f,0.514103f,  
0.471397f,0.427555f,  
[0159] 0.382683f,0.336890f,0.290285f,0.242980f,0.195090f,0.146730f,  
0.098017f,0.049068f  
[0160] } ;  
[0161] 为了对多声道编码进行解码,可以使用下面的c代码以用于基于KLT旋转的方法:
```

```
decode_mct_rotation()
{
    for (pair=0; pair < self->numPairs; pair++) {

        mctBandOffset = 0;

        /* inverse MCT rotation */

        for (win = 0, group = 0; group < num_window_groups; group++) {

            for (groupwin = 0; groupwin < window_group_length[group]; groupwin++, win++) {
                [0162] *dmx = spectral_data[ch1][win];
                *res = spectral_data[ch2][win];
                apply_mct_rotation_wrapper(self, dmx, res, &alphaSfb[mctBandOffset],
                                           &mctMask[mctBandOffset], mctBandsPerWindow,
                                           alpha,
                                           totalsfb, pair, nSamples);
            }
        }

        mctBandOffset += mctBandsPerWindow;
    }
}

[0163] }
```

[0164] 为了分频段处理,可以使用下面的c代码:

```
apply_mct_rotation_wrapper(self, *dmx, *res, *alphaSfb, *mctMask,
mctBandsPerWindow,
                           alpha, totalsfb, pair, nSamples)

{

    sfb = 0;

    if (self->MCCSIGNALINGTYPE == 0) {

    }

    else if (self->MCCSIGNALINGTYPE == 1) {

        /* apply fullband box */

        if (!self->bHasBandwiseAngles[pair] && !self->bHasMctMask[pair]) {

            apply_mct_rotation(dmx, res, alphaSfb[0], nSamples);
[0165]
        }

        else {

            /* apply bandwise processing */

            for (i = 0; i < mctBandsPerWindow; i++) {

                if (mctMask[i] == 1) {

                    startLine = swb_offset [sfb];
                    stopLine = (sfb+2<totalsfb)? swb_offset [sfb+2] : swb_offset
[sfb+1];
                    nSamples = stopLine-startLine;

                    apply_mct_rotation(&dmx[startLine], &res[startLine],
alphaSfb[i], nSamples);
                }
            }
        }
    }
}
```

```

sfb += 2;

/* break condition */

if (sfb >= totalSfb) {

    break;

}

}

[0166] }

}

else if (self->MCCSIGNALINGTYPE == 2) {

}

else if (self->MCCSIGNALINGTYPE == 3) {

    apply_mct_rotation(dmx, res, alpha, nSamples);

}

}

[0167] 为了应用KLT旋转,可以使用下面的c代码:
apply_mct_rotation(*dmx, *res, alpha, nSamples)
{
    for (n=0;n<nSamples;n++) {

        L = dmx[n] * tabIndexToCosAlpha [alphaIdx] - res[n] *
        tabIndexToSinAlpha [alphaIdx];

        R = dmx[n] * tabIndexToSinAlpha [alphaIdx] + res[n] *
        tabIndexToCosAlpha [alphaIdx];

        dmx[n] = L;
        res[n] = R;
    }
}

```

[0168] 图5显示了用于对具有至少三个声道的多声道信号进行编码的方法300的流程图。方法300包括步骤302,在第一迭代步骤中计算在至少三个声道中的每对之间的声道间相关值,在第一迭代步骤中选择具有最高值或具有高于阈值的值的对,并且通过使用多声道处

理操作而处理所选择的对,以导出针对所选择的对的第一多声道参数并导出第一处理声道;步骤304,在第二迭代步骤中通过使用处理声道的至少一个而执行所述计算、所述选择和所述处理,以导出第二多声道参数和第二处理声道;步骤306,对根据通过迭代处理器执行的迭代处理得到的声道进行编码以获得编码声道;以及步骤308,产生具有编码声道以及第一和第二多声道参数的编码多声道信号。

[0169] 图6示出了对具有编码声道以及至少第一和第二多声道参数的编码多声道信号进行解码的方法400的流程图。方法400包括步骤402,对编码声道进行解码以获得解码声道;以及步骤404,通过使用由第二多声道参数标识的解码声道的第二对和通过使用第二多声道参数来执行多声道处理,以获得处理声道,并通过使用由第一多声道参数标识的声道的第一对和使用第一多声道参数来执行进一步的多声道处理,其中声道的第一对包括至少一个处理声道。

[0170] 尽管已在框图的背景(其中,所述块表示真实的或逻辑的硬件组件)下描述了本发明,但本发明也可以实现为计算机实施方法。在后者的情况下,块表示相应方法步骤,其中这些步骤代表由对应逻辑或实体硬件块执行的功能性。

[0171] 虽然已经在装置的上下文中描述了一些方面,但是将清楚的是,这些方面还表示对应方法的描述,其中,块或设备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描述的方面也表示对相应块或项或者相应装置的特征的描述。可以由(或使用)硬件装置(诸如,微处理器、可编程计算机或电子电路)来执行一些或全部方法步骤。在一些实施例中,可以由这种装置来执行最重要方法步骤中的某一个或多个方法步骤。

[0172] 本发明的经传输或编码的信号可以存储在数字存储介质上或可以在诸如无线传输介质的传输介质或诸如因特网的有线传输介质上传输。

[0173] 取决于某些实现要求,可以在硬件中或在软件中实现本发明的实施例。可以通过使用其上存储有电子可读控制信号的数字存储介质(例如,软盘、DVD、Blu-Ray、CD、ROM、PROM和EPROM、EEPROM或闪存)来执行所述实现方案,所述控制信号与可编程计算机系统合作(或能够与之合作),使得执行各个方法。因此,数字存储介质可以是计算机可读的。

[0174] 根据本发明的一些实施例包括具有电子可读控制信号的数据载体,该电子可读控制信号能够与可编程计算机系统协作从而执行本文所述的方法之一。

[0175] 通常,本发明的实施例可以实现为具有程序代码的计算机程序产品,程序代码可操作以在计算机程序产品在计算机上运行时执行方法之一。程序代码可以例如存储在机器可读载体上。

[0176] 其他实施例包括存储在机器可读载体上的计算机程序,该计算机程序用于执行本文所述的方法之一。

[0177] 换言之,本发明方法的实施例因此是具有程序代码的计算机程序,该程序代码用于在计算机程序在计算机上运行时执行本文所述的方法之一。

[0178] 因此,本发明方法的另一实施例是数据载体(或诸如数字存储介质或计算机可读介质的非暂时性存储介质),包含记录于其上的用于执行本文所述方法之一的计算机程序。数据载体、数字存储介质或记录的介质通常是有形的和/或非暂时性的。

[0179] 因此,本发明方法的另一实施例是表示用于执行本文所述的方法之一的计算机程序的数据流或信号序列。数据流或信号序列可以例如被配置为经由数据通信连接(例如,经

由互联网)传送。

[0180] 另一实施例包括处理装置,例如,配置为或适用于执行本文所述的方法之一的计算机或可编程逻辑器件。

[0181] 另一实施例包括其上安装有计算机程序的计算机,该计算机程序用于执行本文所述的方法之一。

[0182] 根据本发明的另一实施例包括被配置为向接收机(例如,以电子方式或以光学方式)传输计算机程序的装置或系统,该计算机程序用于执行本文所述的方法之一。接收机可以是例如计算机、移动设备、存储设备等。装置或系统可以例如包括用于向接收机传送计算机程序的文件服务器。

[0183] 在一些实施例中,可编程逻辑器件(例如,现场可编程门阵列)可以用于执行本文所述的方法的一些或全部功能。在一些实施例中,现场可编程门阵列可以与微处理器协作以执行本文所述的方法之一。通常,方法优选地由任意硬件装置来执行。

[0184] 上述实施例对于本发明的原理仅是说明性的。应当理解的是:本文所述的布置和细节的修改和变形对于本领域其他技术人员将是显而易见的。因此,旨在仅由所附专利权利要求的范围来限制而不是由借助对本文的实施例的描述和解释所给出的具体细节来限制。

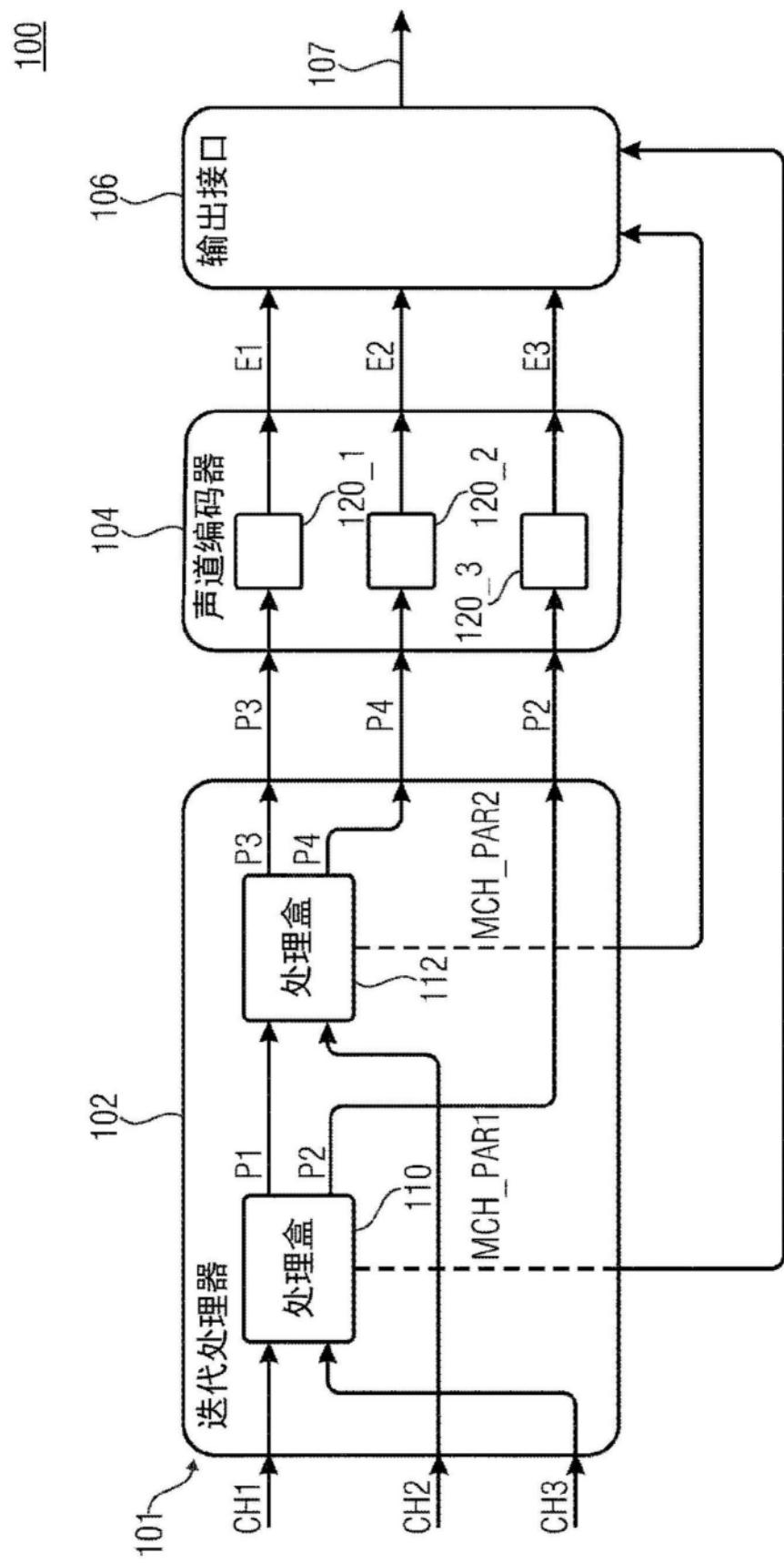


图1

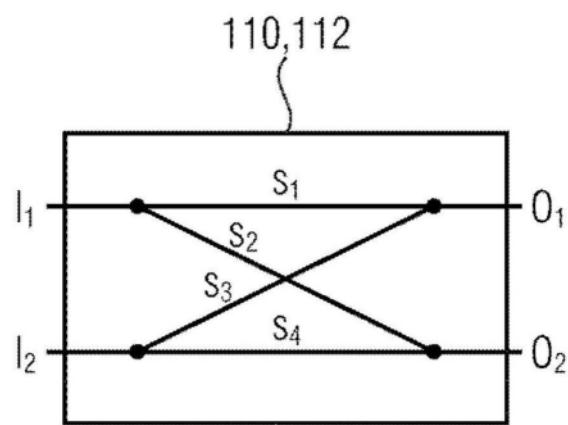


图2

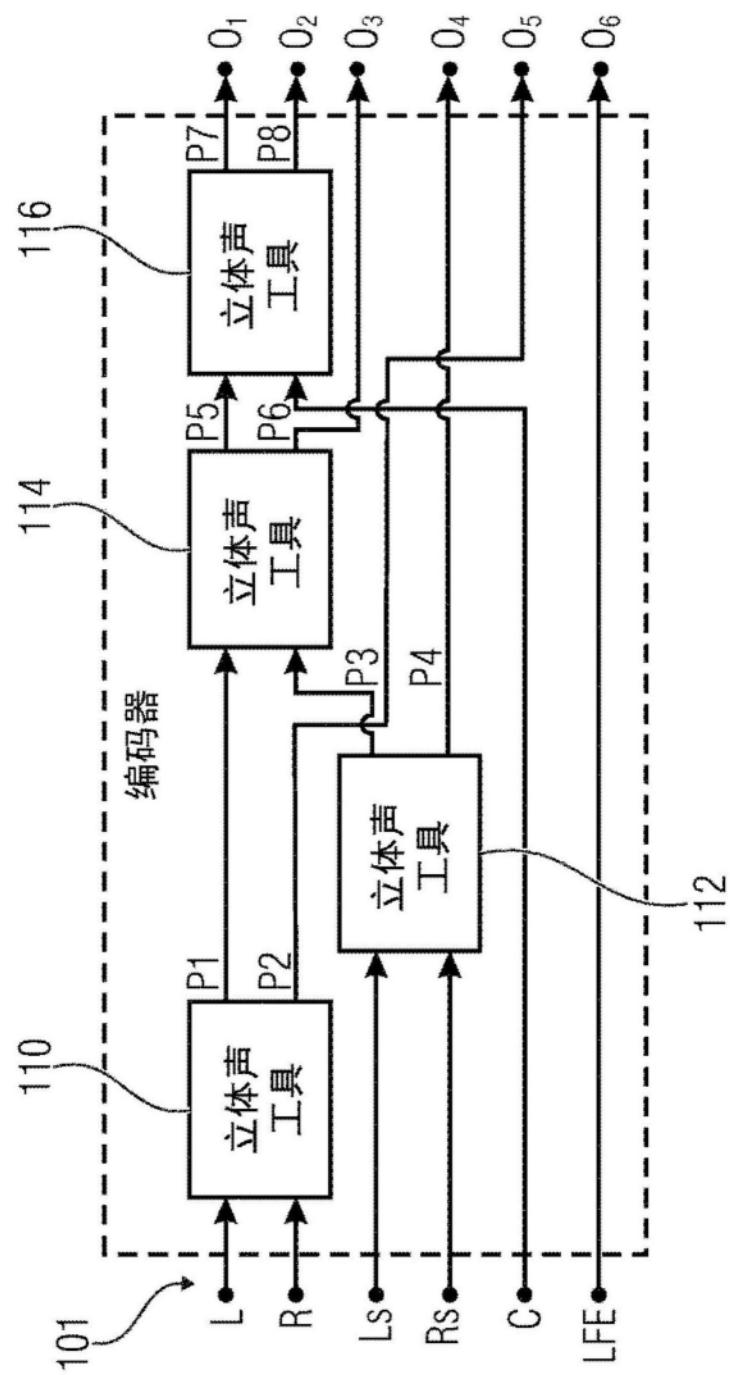


图3

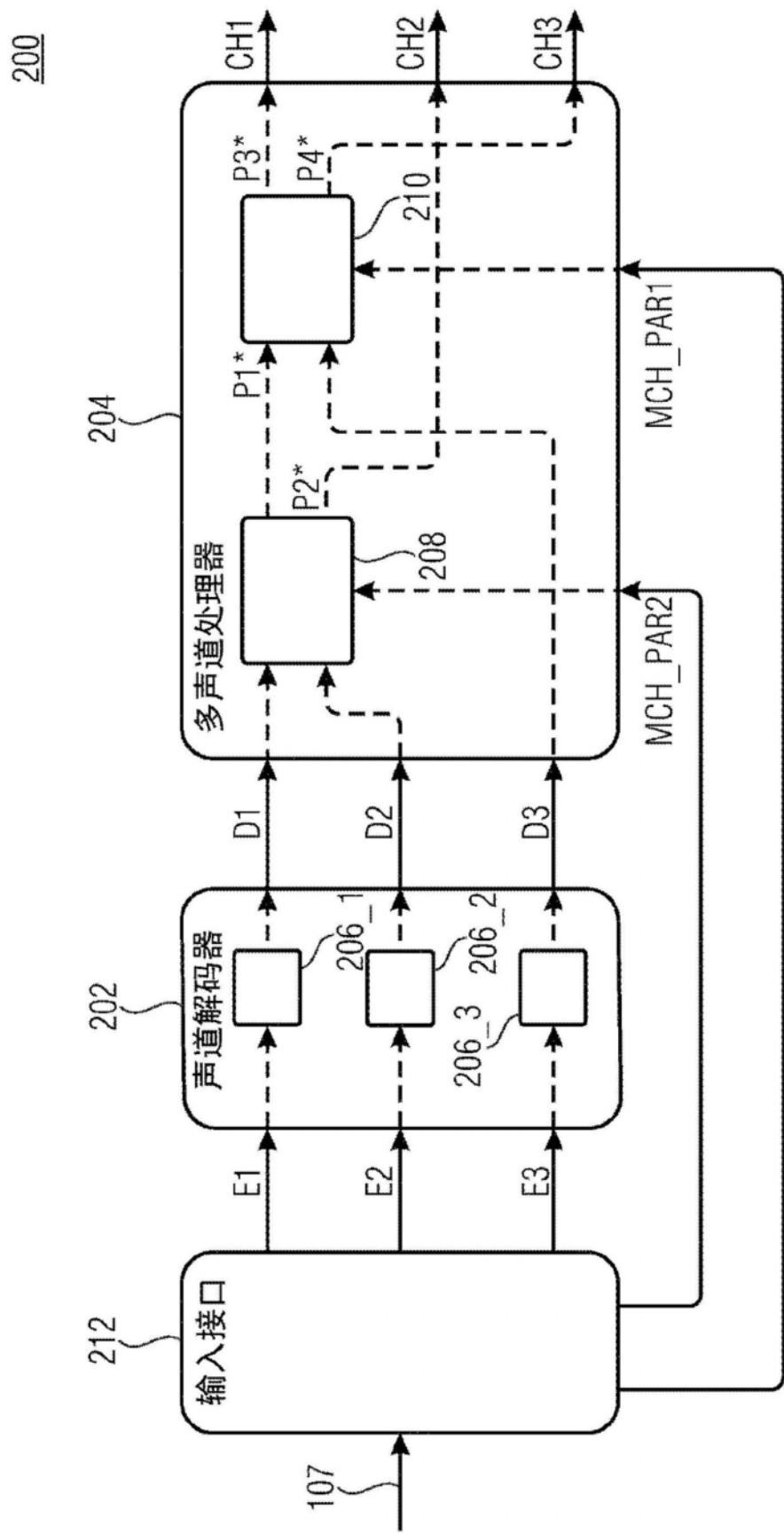


图4

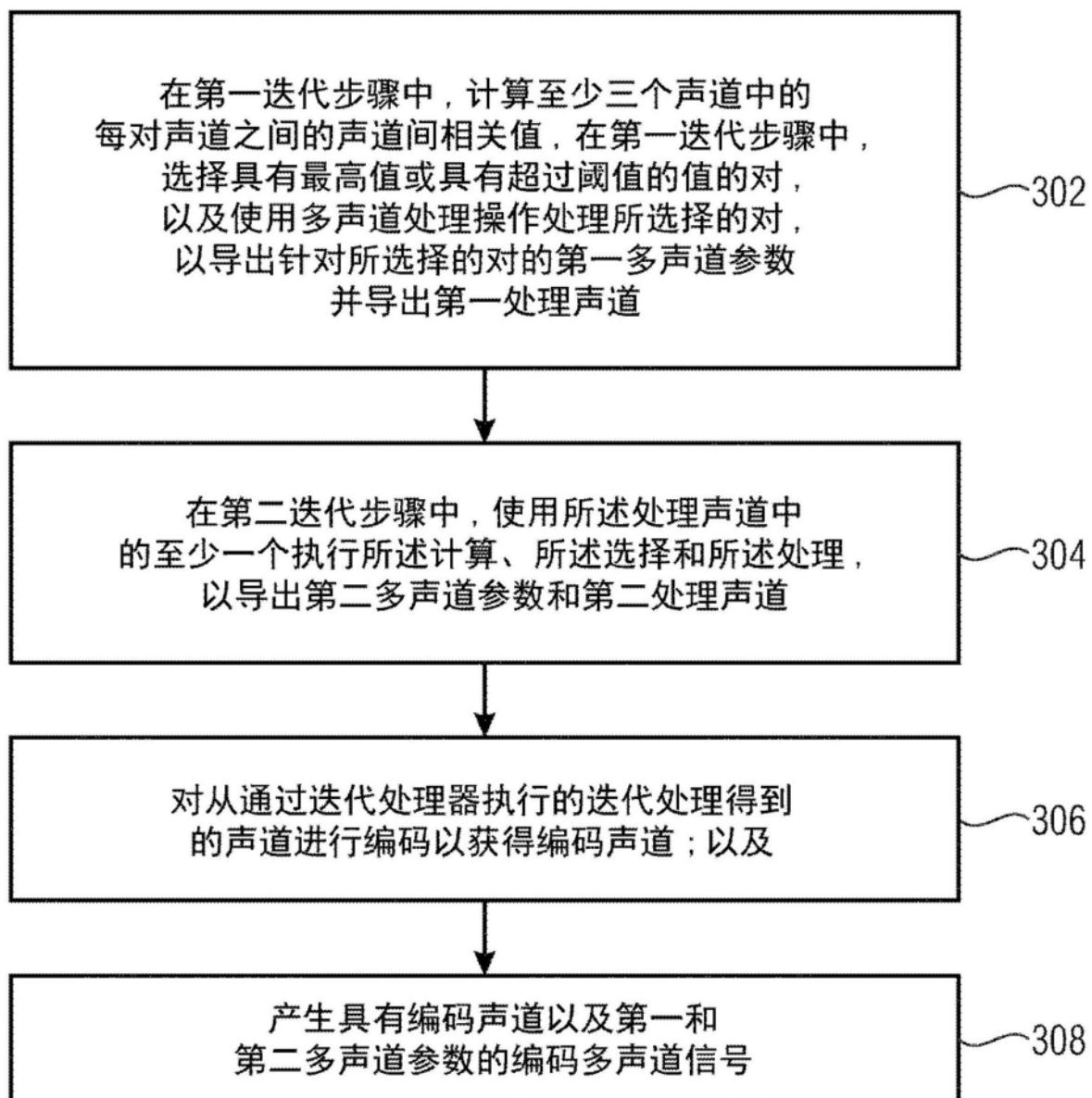


图5

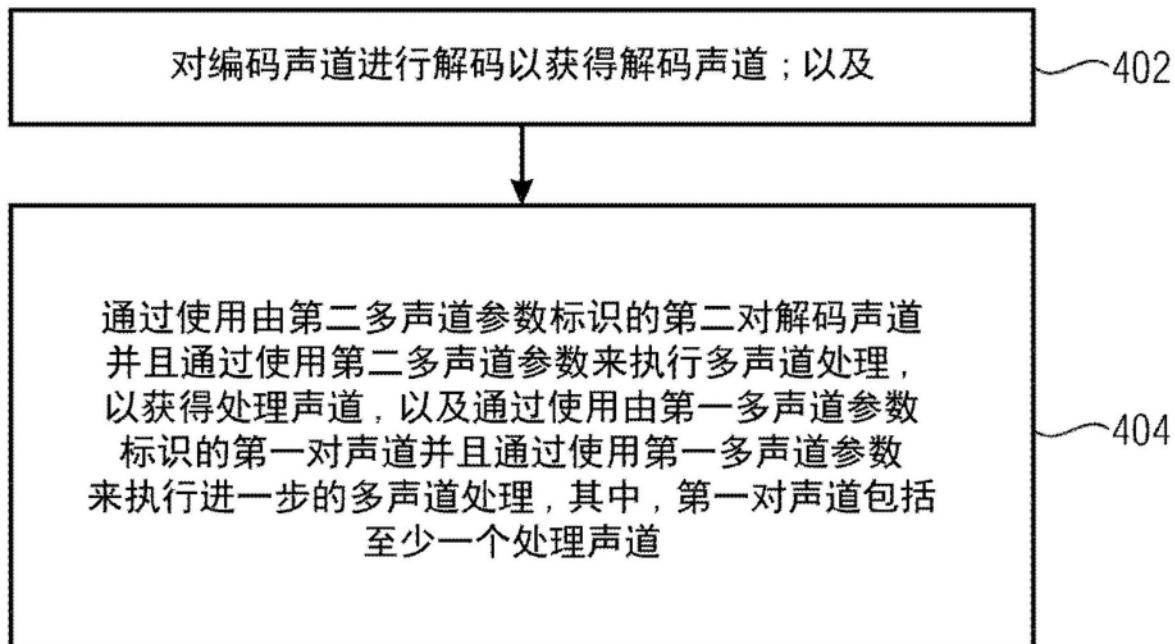
400

图6