



(12) 发明专利申请

(10) 申请公布号 CN 102461171 A

(43) 申请公布日 2012. 05. 16

(21) 申请号 201080029616. 1

(51) Int. Cl.

(22) 申请日 2010. 04. 30

H04N 7/26 (2006. 01)

H04N 13/00 (2006. 01)

(30) 优先权数据

61/215, 154 2009. 05. 01 US

61/215, 874 2009. 05. 11 US

61/310, 497 2010. 03. 04 US

(85) PCT申请进入国家阶段日

2011. 12. 30

(86) PCT申请的申请数据

PCT/US2010/001291 2010. 04. 30

(87) PCT申请的公布数据

W02010/126612 EN 2010. 11. 04

(71) 申请人 汤姆森特许公司

地址 法国伊西莱穆利诺

(72) 发明人 田栋 赖旺霖

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 吕晓章

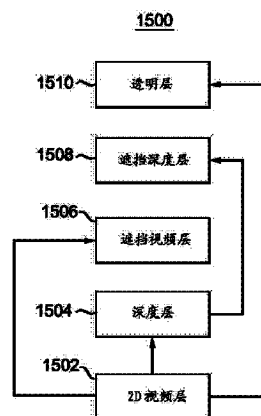
权利要求书 2 页 说明书 50 页 附图 32 页

(54) 发明名称

三维视频的参考画面列表

(57) 摘要

若干实现方式涉及编码 3DV 内容的参考画面列表的构建。本发明的一或多个示范性实施例包括依据优化的层间依赖性结构来构建参考画面列表。另外,可以依据参考画面相对于构建参考画面列表的画面具有的冗余程度使参考画面列表具有优先级。在一个实施方式中,对于画面的层间参考确定优先级。相对于该画面的一个或更多其它参考确定该优先级。在画面的基于所述优先级排序的参考列表中包括该层间参考。



1. 一种方法,包括
确定画面的层间参考相对于该画面的一个或者多个其它参考的优先级;以及
在该画面的基于该优先级排序的参考列表中包含所述层间参考。
2. 根据权利要求 1 所述的方法,还包括基于该画面的依赖性信息确定所述画面的层间参考。
3. 根据权利要求 1 所述的方法,还包括在涉及该画面的编解码操作中使用该层间参考。
4. 根据权利要求 3 所述的方法,其中编解码操作包括至少部分地基于所述层间参考编码该画面。
5. 根据权利要求 3 所述的方法,其中编解码操作包括至少部分地基于所述层间参考解码该画面。
6. 根据权利要求 2 所述的方法,其中从接收的语法元素中访问画面的依赖性信息。
7. 根据权利要求 1 所述的方法,其中所述包含还包括:在排序列表中包含时间上的参考或者视图间参考中的至少一个。
8. 根据权利要求 4 所述的方法,其中编码的画面以及一个或更多其它画面描述在给定时间用于给定视图的三维(3D)信息,并且所述方法还包括:
编码所述一个或更多其它画面;
生成指示怎样将编码的画面和一个或更多其它编码的画面装入支持 3D 处理的结构中的语法元素,所述结构定义画面的内容类型;以及
生成包含所述编码的画面、一个或更多其它编码的画面,以及所述语法元素的比特流;
语法元素的包含在编码的比特流级别上提供结构中的所述编码的画面和编码的一个或更多其它画面之间的关系的指示。
9. 根据权利要求 5 所述的方法,其中所述画面以及一个或更多其它画面描述在给定时间用于给定视图的三维(3D)信息,并且所述方法还包括:
从比特流中访问所述画面的编码和所述一个或更多其它画面的编码,多个画面描述在给定的时间用于给定视图的不同的三维(3D)信息;
从所述比特流中访问语法元素,所述语法元素为所述画面以及所述一个或更多其它画面指示怎样将所述画面和所述一个或更多其它画面装入支持 3D 处理的结构中,所述结构提供所述画面和所述一个或更多其它画面之间的定义关系;
解码一个或更多其它编码的画面;以及
以指示解码的画面和一个或更多其它解码的画面之间的定义关系的输出格式提供解码的画面和解码的一个或更多画面。
10. 根据权利要求 4 所述的方法,还包括:
生成包含所述编码的画面的比特流。
11. 根据权利要求 1 所述的方法,其中所述画面和所述层间参考是与包含二维(2D)视频画面、深度画面、遮挡视频画面、遮挡深度画面,或者透明画面的集合不同的画面。
12. 根据权利要求 1 所述的方法,其中所述画面是遮挡视频画面并且所述层间参考是二维(2D)视频画面。
13. 根据权利要求 12 所述的方法,其中所述包含还包括:在遮挡视频画面的排序的参

考列表的开始处包含所述层间参考。

14. 根据权利要求 1 所述的方法,其中所述画面是遮挡深度画面并且所述层间参考是深度画面。

15. 根据权利要求 14 所述的方法,其中所述包含还包括:在遮挡深度画面的排序的参考列表的开始处包含所述层间参考。

16. 根据权利要求 1 所述的方法,其中所述画面是透明画面并且所述层间参考是二维(2D) 视频画面。

17. 根据权利要求 16 所述的方法,其中所述包含步骤还包括:在透明画面的排序的参考列表的结尾处包含所述层间参考。

18. 根据权利要求 1 所述的方法,其中所述画面是深度画面并且所述层间参考是二维(2D) 视频画面,以及其中所述包含步骤还包括:在深度画面的排序的参考列表的结尾处包含所述层间参考。

19. 根据权利要求 1 所述的方法,其中:

所述画面包含被彼此单独编码的多个部分,以及

所述层间参考的优先级基于多频繁地将所述层间参考用作画面的部分的参考。

20. 根据权利要求 2 所述的方法,其中层间参考的确定还包括:

确定所述画面是否是二维(2D) 视频画面,并且如果所述画面是 2D 视频画面,则在画面的排序的参考列表中排除任何深度画面作为层间参考。

21. 根据权利要求 1 所述的方法,其中在一个或更多编码器或解码器中执行确定优先级和包含层间参考。

22. 一种装置,包括:

用于确定画面的层间参考相对于该画面的一个或者多个其它参考的优先级的部件;以及

用于在该画面的基于该优先级排序的参考列表中包含所述层间参考的部件。

23. 一种处理器可读的介质,在其上存储用于促使处理器至少执行以下步骤的指令:

确定画面的层间参考相对于该画面的一个或者多个其它参考的优先级;以及
在该画面的基于该优先级排序的参考列表中包含所述层间参考。

24. 一种装置,包括被配置为至少执行以下步骤的处理器:

确定画面的层间参考相对于该画面的一个或者多个其它参考的优先级;以及
在该画面的基于该优先级排序的参考列表中包含所述层间参考。

25. 一种装置,包括:

三维视频(3DV) 参考缓冲器,被配置为确定画面的层间参考相对于该画面的一个或者多个其它参考的优先级以及配置为在该画面的基于该优先级排序的参考列表中包含所述层间参考。

26. 根据权利要求 25 所述的装置,还包括一个或更多编码器,被配置为在涉及该画面的编解码操作中使用该层间参考,以及

其中 3DV 参考缓冲器还被配置为基于该画面的依赖性信息确定所述画面的层间参考。

三维视频的参考画面列表

[0001] 相关申请的交叉引用

[0002] 本申请要求 2009 年 5 月 1 日提交的、标题为“3D Video Coding Formats”的、序列号 No 61/215, 154 的美国临时申请的提交日, 2009 年 5 月 11 日提交的、标题为“Reference Pictures for 3D Video”的、序列号 No 61/215, 874 的美国临时申请的提交日, 以及 2010 年 3 月 4 日提交的、标题为“Extended SPS for 3DV sequences”的、序列号 No 61/310, 497 的美国临时申请的提交日的权益, 在此针对所有目的通过引用以其整体并入它们每一个的内容。

技术领域

[0003] 描述有关于编解码系统的实现方式。各种特定的实现方式涉及三维 (3D) 视频编解码方案。

背景技术

[0004] 为了方便新的视频应用, 诸如三维电视 (3DTV) 和自由视点视频 (free-viewpoint video, FVV), 可以利用包括常规的 2D 视频和深度图二者的 3D 视频 (3DV) 数据格式使得可以在用户端呈现附加的视图。这种 3DV 格式的示例包括: 包含二维的 (2D) 视频及其对应的深度图的 2D 加深度 (2D+Z), 以及包含 2D+Z 和遮挡 (occlusion) 视频加遮挡深度的分层的深度视频 (LDV)。这种 3DV 格式的其它示例包括多视图加深度 (MVD) 和视差增强的立体 (DES)。MVD 是 2D+Z 的扩展, 这是由于其包括来自不同视点的多个 2D+Z。继而, DES 由来自两个不同视点的两个 LDV 组成。另一示例 3DV 格式是层深度视频加右视图 (LDV+R), 其由左视图的一个 LDV 和右视图的 2D 视频组成。怎样在所有这些格式中传递 (编码和传输) 数据是具有挑战的问题, 这是由于在用户端共同使用不同的分量 (component) 来解码 3DV 内容。

发明内容

[0005] 根据一个总的方面, 对画面的层间 (inter-layer) 参考确定优先级。相对于该画面的一个或更多其它参考确定该优先级。在画面的基于所述优先级排序的参考列表中包括该层间参考。

[0006] 在附图和以下描述中提出一个或更多实现方式的细节。即使以一个特定方式来描述, 但应清楚可以以各种方式来配置或体现所述实现方式。例如, 实现方式可以被执行为方法, 或者被体现为装置, 诸如, 例如被配置为执行一组操作的装置, 或者存储用于执行一组操作的指令的装置, 或者在信号中体现。在连同附图和权利要求书考虑的以下详细描述中, 其它方面和特征将变得明显。

附图说明

[0007] 图 1 是深度图的示例。

- [0008] 图 2 是示出 LDV 格式的四个分量的示例。
- [0009] 图 3 是 3DV 编码器的实现方式的框图 / 流程图。
- [0010] 图 4 是 3DV 解码器的实现方式的框图 / 流程图。
- [0011] 图 5 是 3DV 层编码器的实现方式的框图 / 流程图。
- [0012] 图 6 是 3DV 层解码器的实现方式的框图 / 流程图。
- [0013] 图 7 是视频传输系统的实现方式的框图 / 流程图。
- [0014] 图 8 是视频接收系统的实现方式的框图 / 流程图。
- [0015] 图 9 是视频处理装置的实现方式的框图 / 流程图。
- [0016] 图 10 是 3DV 编码结构的示例的图。
- [0017] 图 11 是网络抽象层 (NAL) 单元流的第一示例的框图 / 流程图。
- [0018] 图 12 是 NAL 单元流的第二示例的框图 / 流程图。
- [0019] 图 13 是解码 3DV 内容的方法的示例的流程图。
- [0020] 图 14 是编码 3DV 内容的方法的示例的流程图。
- [0021] 图 15 是图示层间依赖性结构的示例的框图。
- [0022] 图 16 是构建用于编码处理的参考画面列表的方法的示例的流程图。
- [0023] 图 17 是构建用于解码处理的参考画面列表的方法的示例的流程图。
- [0024] 图 18 是编码 3DV 内容的扩展的序列参数集的 NAL 单元的方法的示例的流程图。
- [0025] 图 19 是解码 3DV 内容的扩展的序列参数集的 NAL 单元的方法的示例的流程图。
- [0026] 图 20 是编码具有扩展的序列参数集的方法的示例的流程图。
- [0027] 图 21 是解码具有扩展的序列参数集的方法的示例的流程图。
- [0028] 图 22 是编码 3DV 内容的层间依赖性结构的序列参数子集的第一方法的示例的框图 / 流程图。
- [0029] 图 23 是解码 3DV 内容的层间依赖性结构的序列参数子集的第一方法的示例的框图 / 流程图。
- [0030] 图 24 是编码 3DV 内容的层间依赖性结构的序列参数子集的第二方法的示例的框图 / 流程图。
- [0031] 图 25 是解码 3DV 内容的层间依赖性结构的序列参数子集的第二方法的示例的框图 / 流程图。
- [0032] 图 26 是编码 3DV 内容的方法的示例的流程图。
- [0033] 图 27 是解码 3DV 内容的方法的示例的流程图。
- [0034] 图 28 是构建编解码操作的参考画面列表的方法的示例的流程图。
- [0035] 图 29 是可以在图 28 的方法中实现的用于处理 2D 视频层画面的方法的示例的流程图。
- [0036] 图 30 是编码 3DV 内容并且传递层间依赖性结构的方法的示例的流程图。
- [0037] 图 31 是解码 3DV 内容并且传递层间依赖性结构的方法的示例的流程图。
- [0038] 图 32 是 NAL 单元流的示例的框图 / 流程图。
- [0039] 图 33 是通过采用层间依赖性结构来管理网络业务量的系统的示例的框图 / 流程图。
- [0040] 图 34 是通过采用层间依赖性结构来管理网络业务量的方法的示例的流程图。

具体实施方式

[0041] 如在本领域中所理解的,3D 视频 (3DV) 的基本原则是典型地向用户的每只眼睛提供场景或者对象的不同视图使得用户能够感知到场景或者对象的深度。另外,为了增强用户体验,可以呈现除了正在被传输的视图之外的虚拟视图,例如来调整不同的感知深度范围的基线 (baseline) 距离。为了实现这些目标中的一个或者多个,如上注意到,3D 视频 (3DV) 表示格式可以包括各种层,诸如视频、深度,以及或许更多的补充信息,诸如 2D+Z (MVD) 和 LDV (DES)。为了更好地说明 3DV 内容的深度和其它补充信息的概念,参照图 1 和图 2 来进行。

[0042] 图 1 提供了对应于常规的视频的深度图 100 的示例。另外,图 2 包括 LDV 格式中的四个分量的示例:2D 视频 202 加上深度 (Z) 204 以及同一场景的遮挡视频 206 和伴随的遮挡深度 208。对上述的数据格式进行编码和传输在许多方面中是具有挑战的。例如,除了编码效率之外,还应该最好提供诸如同步和向后兼容性 (用于常规的单视觉 (monoscopic) 2D 视频) 之类的功能使得传统的高级视频编码 (AVC) / 多视图编码 (MVC) 解码器可以从比特流中提取可观看的视频。

[0043] 可以解决这些问题中的至少一些的一种解决方案是进行联播 (simulcast),其中独立地编码并传输每个视图和 / 或层。这种方法可以使用多个编码器和解码器来分别编码和解码单独的视图 / 层,并且在系统级别或者应用级别上将视图 / 层同步到可观看的图像。例如,移动画面专家组 (MPEG)-C 部分 3 (国际标准化组织 (ISO) / 国际电工委员会 (IEC) 23002-3) 规定了用于 2D+Z 的系统框架。典型的实现方式在系统级别上使用视频和深度之间的同步。可以使用任何现有的视频编码标准来编码视频和深度。然而,在典型的实现方式中,视频和深度的编码被解耦。由此,联播的成本被典型地乘以所传输的视图和 / 或层的数量。此外,因为单独地编码不同视图和 / 或层,典型地不以任何方式利用视图和 / 或层之间的任何冗余性而实现更高的编码效率。

[0044] 相反,除了 AVC/MVC 系统的向后兼容性之外,在此描述的一个或更多实现方式可以允许层间编码以利用层之间的冗余性,并且由此实现更高的编码效率。具体地,一个或更多实现方式提供了在编码级别上对视图和 / 或层进行同步的手段以便获得这些益处中的至少一些。例如,在这里描述的至少一个实现方式中,提出了 AVC 的 NAL 单元设计上的新的 3DV 前缀网络抽象层 (NAL) 单元和新的 3DV NAL 单元首标扩展以高效地使得能够进行层间编码和视图 / 层的同步。高级语法发信号通知可以怎样从诸如 AVC 和可分级视频编码 (SVC) / MVC 比特流之类的比特流中提取 3DV 分量。由此,该方法具有的优点在于,不需要在系统级别上对不同 3DV 分量之间进行同步,这是由于可以在编码的比特流中耦合 3DV 分量 (诸如 SVC 层,或者 MVC 视图)。另一潜在的益处在于,当以此方式进行编码时,可以移除层间或者视图间 (inter-view) 的冗余性。另外,新的 NAL 单元设计可以与 MVC 兼容并且还可以允许与任何将来的封装编码技术兼容以实现增强的压缩效率。

[0045] 如在以下在此讨论的,为了使得能够在编码级别而不是系统级别上对不同视图 / 层同步,一个或更多实现方式将 3DV NAL 单元设计与 3DV 视图标识符 (ID) 和 3DV 层 ID 相关联。另外,为了更好地利用视图间 / 层间冗余性,与利用交织方法的 AVC 相比较,采用视图间 / 层间预测来提供更高的编码效率。此外,3DV 补充层的 NAL 单元设计可以实现完全的

向后兼容性,同时使得能够开发新的编码模式/工具而不影响 2D 视图层与 MVC/AVC 的兼容性。

[0046] 各种实施例针对参考列表的配置以允许通过采用多参考预测来编码和解码包括 3DV 内容的比特流。例如,对于 3DV 编码结构,可以存在至少三种可能的参考画面类型,例如包括:时间参考画面、视图间参考画面,以及来自不同 3DV 层的参考画面。来自不同 3DV 层的参考画面可以包括,例如,用于深度层的参考的 2D 视频层。在本申请中描述的至少一个实施例提供了怎样在参考画面列表中布置三种类型的参考画面的构思和实现方式。例如,当以预测模式编码宏块 (MB) 时,编码器可以发信号通知哪个画面或者哪些画面用作可得到的多个参考画面中的参考。这里,列表中的索引可以指示使用哪个参考画面。如以下在此进一步讨论的,一个或更多实施例可以在列表中提供一个或更多层间参考画面以便使得能够进行层间预测。

[0047] 如以上注意的,一个或更多实施例提供许多优点,其中之一是与 MVC 的潜在的兼容性。也就是,当根据这些实施例之一的 3DV 比特流被馈入传统的 MVC 解码器时,可以解码并输出 2D 视频(例如,以下被指定为层 0)。为了进一步有助于与 MVC 的兼容性,同时允许使用各种层的 3DV 内容的高效编码,各种实施例附加地针对序列参数集 (SPS) 的构建和信令。如本领域普通技术人员所理解的,SPS 可以指定画面序列的画面之间共享的公共属性。这样的公共属性可以包括,例如,画面大小、可选的采用的编码模式,以及宏块到码片组的映射,它们中的每一个可以在序列中的画面之间可选地共享。对于至少一个实施例,采用 SPS 的扩展来发信号通知用于编码和解码 3DV 内容的新的序列参数。此外,可以利用单独且新的 NAL 单元类型用于扩展的 SPS。扩展的 SPS 可以用于诸如路由器之类的网络设备以适配于 3DV 内容的流传输的比特率,如以下在此进一步讨论的。

[0048] 在详细地讨论实施例之前,提供对采用的术语的一些讨论以便于理解所描述的构思。

[0049] 术语:

[0050] 的“2D 视频”在此通常用于指代传统的视频信号。

[0051] “深度”层在此通常用于指代指示场景对象的距离信息的数据。“深度图”是深度层的典型示例。

[0052] “遮挡视频”层在此通常用于指代从某个视点遮挡的视频信息。遮挡视频层典型地包括 2D 视频层的背景信息。

[0053] “遮挡深度”层在此通常用于指代从某个视点遮挡的深度信息。遮挡深度层典型地包括深度层的背景信息。

[0054] “透明”层通常用于指代用以指示深度不连续性或者深度边界的画面。典型的透明层具有二进制信息,其中大于特定阈值的两个值中之一指示深度关于邻近深度值具有不连续性的位置。

[0055] “3D 视图”在此被定义为来自于一个视图位置的数据集,其与 MVC 中使用的“视图”不同。例如,3DV 视图可以包括比 MVC 视图更多的数据。对于 2D+Z 格式,3DV 视图可以包括两个层:2D 视频加上其深度图。对于 LDV 格式,3DV 视图可以包括四个层:2D 视图、深度图、遮挡视频,以及遮挡深度图。另外,透明图可以是 3DV 视图中除其它之外的另一层的数据类型。

[0056] “3DV 层”被定义为 3DV 视图的层之一。3DV 层的示例例如是 2D 视图或者视频、深度、遮挡视频、遮挡深度,以及透明图。除了 2D 视图或者视频之外的层还被定义为“3DV 补充层”。在一个或更多实施例中,3DV 解码器可以被配置为识别层并且使用 3dv_layer_id 将该层与其它进行区分。在一个实施例中,如在表 1 中定义 3dv_layer_id。然而,应注意,如本领域普通技术人员在考虑到这里提供的教导时所理解的,可以以其它方式定义并识别层。

[0057] 表 1-3DV 层

[0058]

3dv_layer_id 的值	描述
0	2D 视频
1	深度
2	遮挡视频
3	遮挡深度
4	透明图
≥ 5	保留

[0059]

[0060] 图 3 和图 4 分别图示了高级通用的 3DV 编码器 300 和解码器 400。编码器 300/ 解码器 400 由层编码器 / 解码器和 3DV 参考缓冲器组成。例如,将可能包括例如 2D 视图、深度、遮挡视图、遮挡深度以及透明图各层的 3DV 内容信号 302 输入给如图 3 所示的各个层编码器。具体地,编码器系统 / 装置 300 包括:2D 层编码器 304,被配置为编码 2D 层,其可以是 AVC 兼容的;增强的 2D 层编码器 306,配置为编码增强的 2D 层;深度层编码器 308,配置为编码深度层;遮挡视图层编码器 310,配置为编码遮挡视图层;遮挡深度层编码器 312,配置为编码遮挡深度层;以及透明层编码器 314,配置为编码透明层。由此,可以使用不同的编码器和 / 或编码技术来编码每个层。

[0061] 增强的 2D 层通常在这里用于将这种层和与 AVC、MVC、SVC,或者一些其它底层标准兼容的层区分开。例如,增强的 2D 层典型地不与 MVC 兼容,这是因为这种层允许诸如例如使用层间参考的新的编码工具。因此,这种层通常不向后兼容 MVC。

[0062] 注意到,术语“增强的 2D 层”(或者补充层)还可以用于指代可以利用 MVC 编码的、但是不希望被显示的并且因此典型地不被描述为利用 MVC 所编码的层。例如,一系列深度层可以被 MVC 视为一系列的画并且可以被 MVC 编码。然而,典型地不能显示深度层,因此,通常期望具有识别和编码这种层的不同的方式,除了通过使用 MVC。

[0063] 每个层还可以使用不同的参考。参考可以来自于除了正编码(解码)的画面/块之外的不同的层。来自不同层的参考可以从 3DV 参考缓冲器 316(3DV 参考 / 输出缓冲器 414)中获得。如图 3 中所示,每个层编码器与 3DV 参考缓冲器 316 进行信号通信以允许编码输入信号 302 以生成输出信号 318 的各种模式。

[0064] 通过利用 3DV 参考缓冲器 316,3DV 格式的每个层可以使用来自其自身的层的参考

和 / 或使用各个层之间的层间预测来进行编码,所述参考诸如是例如具有运动和 / 或视差补偿的相同层内的时间参考和 / 或视图间参考。例如,层间预测可以重用运动信息,诸如例如,来自另一层的运动矢量、参考索引等来编码当前层,其也被称为运动跳跃模式。以此方式,输出信号 318 可以与用于一个或更多 3DV 视图的各种层信息进行交织。层间预测可以是基于其它层的访问的任何种类的技术。

[0065] 关于解码器系统 / 装置 400,系统 400 包括各种层解码器,如图 4 所示,信号 318 可以被输入到所述各种层解码器。具体地,解码器系统 / 装置 400 包括:2D 层解码器 402,其可以是 AVC 兼容的、被配置为解码 2D 层;增强的 2D 层解码器 404,配置为解码增强的 2D 层;深度层解码器 406,配置为解码深度层;遮挡视图层解码器 408,配置为解码遮挡视图层;遮挡深度层解码器 410,配置为解码遮挡深度层;和 / 或透明层解码器 412,配置为解码透明层。

[0066] 如图 4 中图示的,每个层解码器与 3DV 参考 / 输出缓冲器 414 进行信号通信,3DV 参考 / 输出缓冲器 414 可以被配置为解析从层解码器接收的解码的层信息并且确定输入信号中包括的层怎样适合支持 3D 处理的结构。这种 3D 处理可以包括例如如这里描述的 3D 层的编码或者在接收器或者显示单元处的附加画面的呈现(合成)。呈现可以使用例如深度画面来弯曲(warp)2D 视频和 / 或遮挡画面以便利用背景信息来填充呈现的画面中的洞(hole)。

[0067] 另外,3DV 参考 / 输出缓冲器 414 可以被配置为生成用于向用户呈现的 3DV 兼容格式的输出信号 416。格式化的 3DV 内容信号 416 当然可以包括例如,2D 视图、深度、遮挡视图、遮挡深度,以及透明图各层。如图 4 所示的,可以与参考缓冲器一起实现输出缓冲器,或者可替换地,在其它实施例中,参考缓冲器和输出缓冲器可以分离。

[0068] 编码器 300 和解码器 400 的其它实现方式可以使用更多或者更少的层。另外,可以使用与所示的那些层不同的层。

[0069] 应清楚,如在 3DV 参考缓冲器 316 和在 3DV 参考 / 输出缓冲器 414 中所使用的,术语“缓冲器”是智能缓冲器。这种缓冲器可以用于例如存储画面、提供参考(或者参考的一部分),以及重新排序输出的画面。另外,这种缓冲器可以用于例如执行各种其它处理操作,诸如例如假设的参考解码器测试、处理标记(marking)命令(例如,AVC 中的存储管理控制操作),以及解码的画面缓冲器管理。

[0070] 图 5 和图 6 分别描绘了通用的 3DV 层编码器 500 和解码器 600 的高级框图 / 流程图,其可以分别用于实现层编码器 304-314 中的任何一个或更多以及层解码器 402-412 中的任何一个或更多。应注意,如例如图 5 中描绘的,层编码器 304-314 中的每一个可以关于它们对应的各层的相同的一般方式来设计以利于特定的目的。相反,如鉴于在此提供的教导所理解的,层编码器可以被不同地配置以更好地利用它们独特的特征。类似地,如例如图 6 中描绘的,解码器 402-412 可以关于它们对应的各层的相同的一般方式来设计。相反,层编码器可以被不同地配置以更好地利用它们独特的特征。

[0071] 应注意,关于 MVC 编码器,输入由多个视图组成。每个视图是传统的 2D 视频。由此,与 AVC 编码器相比较,典型的 MVC 编码器包括诸如视差估计块、视差补偿块之类附加块,以及视图间参考缓冲器。类似地,图 5 和图 6 包括用于 3DV 参考和层间预测的块。在 3DV 编码器的情况下,输入由多个 3D 视图组成。如上所述,每个 3D 视图可以包括若干层。相应

地,用于每个层的编码方法可以被不同地设计以利用它们独特的特性。结果,3DV 编码器可以被划分为层编码器,如图 3 所示。然而,层编码器还可以被紧耦合。在层编码器中使用的技术可以被调整 (tailor) 为给定系统所期望的。由于每个层表现为视频信号,所以层可以具有与图 5 所示的高级处的类似的结构。应注意,在较低的、更具体的级别,层编码器可以被不同地设计。当然,一个实施例还可以使用被配置为编码所有层的单个编码器。

[0072] 关于图 5 所图示的高级图,3DV 层编码器 500 可以包括层分区器 504,被配置为针对输入信号 502 内的 3DV 视图 i 接收并且将 3DV 视图层彼此分区。分区器 504 与加法器或者组合器 506、与移位 (运动 / 视差) 补偿模块 508、以及移位 (运动 / 视差) 估计模块 510 进行信号通信,其中每一个从分区器 504 接收一组分区的层。加法器 506 的另一输入是通过开关 512 接收的各种可能的参考画面信息之一。

[0073] 例如,如果与开关 512 进行信号通信的模式判定模块 536 确定编码模式应该是参考当前正被编码的相同块或者码片的帧内预测,则加法器从帧内预测模块 530 接收其输入。可替换地,如果模式判定模块 536 确定编码模式应该是移位补偿和估计,则加法器从移位补偿模块 508 接收其输入,如图 5 中所示,所述移位补偿和估计参考当前正在被处理的相同帧或者 3DV 视图或者 3DV 层的块或者码片,或者是与当前正在被编码的块或者码片不同的、之前处理的另一帧或者 3DV 视图或者 3DV 层的块或码片。此外,如果模式判定模块 536 确定编码模式应该是 3DV 层间预测,则加法器从与 3DV 参考缓冲器 532 进行信号通信的 3DV 层间预测模块 534 接收其输入,所述 3DV 层间预测参考当前正在被处理的相同帧或者 3DV 视图的 3DV 层,或者是与当前正在被处理的层不同的之前处理的另一帧或者 3DV 视图的 3DV 层。

[0074] 加法器 506 向变换模块 514 提供包括 (多个) 3DV 层和预测、补偿,和 / 或估计信息的信号,变换模块 514 被配置为变换其输入信号并且将变换的信号提供给量化模块 516。量化模块 516 被配置为对其接收的信号执行量化并且向熵编码器 518 输出量化的信息。熵编码器 518 被配置为对其输入信号执行熵编码以生成比特流 520。逆量化模块 522 被配置为从量化模块 516 接收量化的信号并且对量化的信号执行逆量化。继而,逆变换模块 524 被配置为从模块 522 接收逆量化的信号并且对其接收的信号执行逆变换。模块 522 和 524 重建或者重构从加法器 506 输出的信号。

[0075] 加法器或者组合器 526 对从逆变换模块 524 和开关 512 接收的信号进行相加 (组合) 并且将结果信号输出给帧内预测模块 530 和去块滤波器 528。此外,帧内预测模块 530 使用其接收的信号执行帧内预测,如以上讨论的。类似地,去块滤波器 528 对从加法器 526 接收的信号进行滤波并且将滤波的信号提供给 3DV 参考缓冲器 532。

[0076] 3DV 参考缓冲器 532 继而解析其接收的信号。3DV 参考缓冲器 532 通过元件 534、508 和 510 辅助层间和移位补偿 / 估计编码,如以上讨论的。3DV 参考缓冲器 532 例如提供各种 3DV 层中的全部或者一部分。

[0077] 再次参照图 6,3DV 层解码器 600 可以被配置为使用比特流接收器 602 接收比特流 318,比特流接收器 602 继而与比特流解析器 604 进行信号通信并且向解析器 604 提供比特流。比特流解析器 604 可以被配置为向熵解码器 606 传输残差比特流 605、向模式选择模块 622 传输控制语法元素 607、向移位补偿 (运动 / 视差) 模块 618 传输移位 (运动 / 视差) 矢量信息 609,以及向 3DV 层间预测模块 620 传输来自除了当前被解码的 3DV 层之外的 3DV

层的编码信息 611。逆量化模块 608 可以被配置为对从熵解码器 606 接收的熵解码信号执行逆量化。另外,逆变换模块 610 可以被配置为对从逆量化模块 608 接收的逆量化的信号执行逆变换并且向加法器或者组合器 612 输出逆变换的信号。

[0078] 加法器 612 可以接收依赖于采用的解码模式的各种其它信号之一。例如,模式判定模块 622 可以通过解析并分析控制语法元素 607 来判定是否编码器 500 对当前处理的块执行了 3DV 层间预测、移位补偿或帧内预测编码。依赖于所确定的模式,模式选择控制模块 622 可以基于控制语法元素 607 访问并控制开关 623,使得加法器 612 可以从 3DV 层间预测模块 620、移位补偿模块 618 或者帧内预测模块 614 接收信号。

[0079] 这里,帧内预测模块 614 可以被配置例如执行帧内预测,所述帧内预测使用对当前正在被解码的相同块或者码片的参考来解码块或者码片。继而,移位补偿模块 618 可以被配置例如执行移位补偿,所述移位补偿使用对当前正在被处理的相同帧或者 3DV 视图或者 3DV 层的块或者码片,或者对与当前正在被解码的块或者码片不同的、之前处理的另一帧或者 3DV 视图或者 3DV 层的块或码片的参考来解码块或者码片。此外,3DV 层间预测模块 620 可以被配置例如执行 3DV 层间预测,所述 3DV 层间预测使用对当前正在被处理的相同帧或者 3DV 视图的 3DV 层,或者对与当前正在被处理的层不同的之前处理的另一帧或者 3DV 视图的 3DV 层的参考来解码块或者码片。

[0080] 在接收预测或者补偿信息信号之后,加法器 612 可以将预测或者补偿信息信号与逆变换的信号相加,用于传输给去块滤波器 602。去块滤波器 602 可以被配置为对其输入信号进行滤波并且输出解码的画面。加法器 612 还可以将相加的信号输出给帧内预测模块 614 用于帧内预测。此外,去块滤波器 602 可以将滤波的信号传输给 3DV 参考缓冲器 616。3DV 参考缓冲器 616 可以被配置为解析其接收的信号以通过元件 618 和 620 来允许并辅助层间和移位补偿解码,3DV 参考缓冲器 616 将解析的信号提供给元件 618 和 620 中的每一个。这种解析的信号可以是例如各种 3DV 层的全部或者一部分。

[0081] 应理解,系统/装置 300、400、500 和 600 可以被不同地配置并且可以包括本领域普通技术人员鉴于在此公开的教导所理解的不同元件。

[0082] 现在参照图 7,图 7 图示了依据一实现方式的可以应用在此描述的方面的视频传输系统/装置 700。视频传输系统 700 可以是例如一种使用各种介质中任一种来传输信号的首端(head-end)或者传输系统,所述各种介质诸如例如是卫星、电缆、电话线,或者地面广播。可以通过因特网或者其它一些网络来提供该传输。

[0083] 视频传输系统 700 能够生成并递送例如视频内容和深度,以及伴随其它 3DV 补充层。这是通过以下方式来实现的:生成包括 3DV 补充层信息或者能够用于在可能例如具有解码器的接收器端合成 3DV 补充层信息的信息的(多个)编码信号。

[0084] 视频传输系统 700 包括编码器 710 和能够传输编码的信号的传输器 720。编码器 710 接收视频信息并且基于视频信息和/或 3DV 层信息生成(多个)编码信号。编码器 710 可以是例如以上详述的编码器 300。编码器 710 可以包括例如包含用于接收各种信息并且将其组装为用于存储或传输的结构化格式的组装单元的子模块。各种信息可以包括例如,编码或者未编码的视频、编码或者未编码的深度信息,以及编码或者未编码的元素,诸如例如运动矢量、编码模式指示符,以及语法元素。

[0085] 传输器 720 可以例如被适配为传输具有一个或者多个比特流的节目信号 750,所

述一个或更多比特流表示与之有关的编码画面和 / 或信息。典型的传输器执行以下一个或者多个功能 : 诸如例如, 提供纠错编码、交织信号中的数据、随机化信号中的能量, 以及使用调制器 722 将信号调制到一个或更多载波。传输器 720 可以包括, 或者连接到天线 (未示出)。此外, 传输器 720 的实现方式可以包括或者限于调制器。

[0086] 参照图 8, 图 8 图示了依据一实现方式的可以应用在此描述的方面的视频接收系统 / 装置 800。视频接收系统 800 可以被配置为通过各种介质来接收信号, 所述各种介质诸如例如是卫星、电缆、电话线, 或者地面广播。可以通过因特网或者其它一些网络来接收该信号。

[0087] 视频接收系统 800 可以是例如蜂窝电话、计算机、机顶盒、电视机, 或者是接收编码的视频并且提供解码的视频用于向用户显示或者用于存储的其它设备。由此, 视频接收系统 800 可以将其输出提供给例如电视机、计算机监视器、计算机 (用于存储、处理或者显示), 或者其它一些存储、处理或者显示设备的屏幕。

[0088] 视频接收系统 800 能够接收并处理包括视频信息的视频内容。视频接收系统 800 包括 : 接收器 810, 能够接收编码的信号, 诸如例如在本申请的实现方式中描述的信号 ; 以及解码器 820, 能够解码接收的信号。

[0089] 接收器 810 可以例如被适配为接收具有多个代表编码画面的比特流的节目信号。典型的接收器执行以下一个或更多功能 : 诸如例如, 接收调制和编码的数据信号, 使用解调器 822 从一个或更多载波中解调数据信号, 对信号中的能量进行去随机化, 对信号中的数据进行解交织, 以及对信号进行纠错解码。接收器 810 可以包括, 或者连接到天线 (未示出)。接收器 810 的实现方式可以包括或者限于解调器。

[0090] 解码器 820 输出包括视频信息和深度信息的视频信号。解码器 820 可以是例如以上详述的解码器 400。

[0091] 系统 700 的输入在图 7 中被列为“(多个) 输入视频”, 来自系统 800 的输出在图 8 中被列为“输出视频”。应清楚, 至少在这些实现方式中, 这些指代包括多个层的 3D 视频。

[0092] 参照图 9, 图 9 图示了依据一实现方式的可以应用在此描述的方面的视频处理设备 900。视频处理设备 900 可以是例如机顶盒或者接收编码的视频并且提供例如解码的视频用于向用户显示或者用于存储的其它设备。由此, 视频处理设备 900 可以将其输出提供给电视机、计算机监视器、计算机或者其它处理设备。

[0093] 视频处理设备 900 包括前端 (FE) 设备 905 和解码器 910。前端设备 905 可以例如是适配为接收具有多个表示编码的画面的比特流的节目信号并且从多个比特流中选择一个或更多比特流用于解码的接收器。典型的接收器执行以下一个或更多功能 : 诸如例如, 接收调制和编码的数据信号, 解调该数据信号, 解码数据信号的一个或者多个编码 (例如, 信道编码和 / 或源编码), 和 / 或对数据信号进行纠错。前端设备 905 可以从例如天线 (未示出) 接收节目信号。前端设备 905 将接收的数据信号提供给解码器 910。

[0094] 解码器 910 接收数据信号 920。数据信号 920 可以包括例如以下中的一个或更多 : 高级视频编码 (AVC)、可分级视频编码 (SVC), 或者多视图视频编码 (MVC) 兼容的流。

[0095] AVC 更具体地指代现有的国际标准化组织 / 国际电工委员会 (ISO/IEC) 运动画面专家组 -4 (MPEG-4) 第十部分高级视频编码 (AVC) 标准 / 国际电信联盟电信分部 (ITU-T) H. 264 推荐 (下文的“H. 264/MPEG-4AVC 标准”或者其变型, 诸如“AVC 标准”或者简称

“AVC”)。

[0096] MVC更具体地指代AVC标准的多视图视频编码(“MVC”)扩展(附录H),称为H. 264/MPEG-4AVC, MVC扩展(“MVC扩展”或者简称“MVC”)。

[0097] SVC更具体地指代AVC标准的可分级视频编码(“SVC”)扩展(附录G),称为H. 264/MPEG-4AVC, SVC扩展(“SVC扩展”或者简称“SVC”)。

[0098] 解码器910解码接收的信号920的全部或者一部分,并且提供解码的视频信号930作为输出。解码的视频930被提供给选择器950。设备900还包括接收用户输入970的用户接口960。用户接口960基于用户输入970向选择器950提供画面选择信号980。画面选择信号980和用户输入970指示用户期望显示多个画面、序列、可分级版本、视图,或者可得到的解码数据的其它选择中的哪一个。选择器950提供(多个)选择的画面作为输出990。选择器950使用画面选择信息980来在解码视频930中选择哪些画面以便提供为输出990。

[0099] 在各种实现方式中,选择器950包括用户接口960,而在其它实现方式中,不需要用户接口960,这是因为选择器直接接收用户输入970而不执行单独的接口功能。例如,选择器950可以被实现为软件或者集成电路。在一个实现方式中,选择器950与解码器910合并,在另一实现方式中,解码器910、选择器950和用户接口960都被集成。

[0100] 在一个应用中,前端905接收各种电视节目的广播并且选择一个用于处理。一个节目的选择是基于期望观看的节目的用户输入。尽管在图9中未示出前端设备905的用户输入,但前端设备905接收用户输入970。前端905接收广播并且通过解调广播频谱的相关部分以及解码解调的节目的任何外部编码而处理期望的节目。前端905向解码器910提供解码的节目。解码器910是包括设备960和950的集成单元。解码器910由此接收用户输入,所述用户输入是用户提供的观看节目中的期望视图的指示。解码器910解码选择的视图,以及来自其它视图的任何需要的参考画面,并且提供解码的视图990用于在电视机(未示出)上显示。

[0101] 继续上述应用,用户可能期望切换被显示的视图并且可能然后向解码器910提供新的输入。在接收到来自用户的“视图改变”之后,解码器910解码旧视图和新视图,以及在旧视图和新视图之间的任何视图。也就是,解码器910解码在物理上位于拍摄旧视图的相机和拍摄新视图的相机之间的、相机拍摄的任何视图。前端设备905还接收识别旧视图、新视图,以及其间的视图的信息。这种信息可以例如由具有关于视图的位置的信息的控制器(在图9中未示出)或者解码器910来提供。其它实现方式可以使用具有与前端设备集成在一起的控制器的前端设备。

[0102] 解码器910提供所有这些解码的视图作为输出990。后处理器(在图9中未示出)在视图之间进行内插以提供从旧视图到新视图的平滑转换,并且向用户显示该转换。在转换到新视图之后,后处理器通知(通过一个或者多个通信链路,未示出)解码器910和前端设备905仅需要新视图。此后,解码器910仅提供新视图作为输出990。

[0103] 系统/装置900可以用于接收图像序列的多个视图、并且表现单个视图用于显示,以及以平滑的方式在各个视图之间切换。平滑的方式可能涉及在视图之间进行内插以移动到另一视图。另外,系统900可以允许用户旋转对象或者场景,或者以其它方式观看对象或者场景的三维表示。对象的旋转例如可以对应于:从视图到视图的移动,并且在视图之间进

行内插以获得视图之间的平滑转换或者简单地获得三维表示。也就是,用户可以“选择”内插的视图作为要显示的“视图”。

[0104] 应清楚,视频传输系统 700、视频接收系统 800 和视频处理设备 900,都可以被适配用于在本申请中描述的各种实现方式。例如,系统 700、800 和 900 可以被适配为以讨论的 3DV 格式之一操作数据以及操作相关联的信令信息。

[0105] 实施例 1:3DV 前缀 NAL 单元

[0106] 在本实施例中,引入了新的 NAL 单元类型并且被称为表示为“16”的“3DV 前缀 NAL 单元”,其可以在特定 3DV 视图或者 3DV 层的视频编码层 (VCL) NAL 单元或者 MVC 前缀 NAL 单元 (具有被表示为 14 的 nal_unit_type) 之前。在 2009 年 1 月-2 月,瑞士日内瓦,Gary Sullivan 等人的“Editors’ draft revision to ITU-T Rec.H.264|ISO/IEC 14496-10 Advanced Video Coding”,JVT-AD007(下文简称“AVC 草案”)中描述了 VCL NAL 单元和 MVC 前缀单元,在此通过引用并入,其与所提出的 AVC 标准有关。可以在 AVC 草案中找到所使用的没有在此明确定义的许多术语和缩写的含义,并且可由相关技术领域的人员理解。表示 3DV 前缀 NAL 单元的“16”的使用是任意的,并且可以被选择为 AVC 草案中任何被保留的 NAL 单元类型。

[0107] 以下提供的表 2 是用于 nal_unit_type 代码的 AVC 草案中的表 7-1 的修改的版本并且定义 3DV 前缀 NAL 单元 16。AVC 草案中的表 7-1 在以下被再现为表 3。应注意,表 2 还包括以下更详细地讨论的实施例 3 的修改。3DV 前缀 NAL 单元 16 允许 MVC 兼容的解码器解码所有传输的 3DV 层,包括 3DV 补充层,并且还允许在编码级别同步 3DV 视图和层。表 2 的 2-5 行 (NAL 单元类型 16-23) 反映了对表 3 的语法改变。

[0108] 表 2-NAL 单元类型代码、语法元素类别,以及 NAL 单元类型分类

[0109]

[0110]

nal_unit_type	NAL 单元的内容和 RBSP 语法结构	C	附录 A	附录 G 和附录 H NAL 单元类型分类
0..15	如 AVC 草案中表 7-1 定义的			
16	3DV 前缀 NAL 单元		非 VCL	非 VCL
17..20	保留的			
21	编码的 3DV 码片扩展 3dv_slice_layer_extension_rbsp()	2,3,4	非 VCL	VCL
22..23	保留的		非 VCL	非 VCL
24..31	如 AVC 草案中表 7-1 定义的		非 VCL	非 VCL

[0111] 表 3-NAL 单元类型代码、语法元素类别,以及 NAL 单元类型分类

[0112]

nal_unit_type	NAL 单元的内容和 RBSP 语法结构	C	附录 A NAL 单元 类型 分类	附录 G 和 附 录 H NAL 单元 类型 分类
0	未指定		非 VCL	非 VCL
1	编码的非 IDR 画面的码片 slice_layer_without_partitioning_rbsp()	2,3,4	VCL	VCL
2	编码的码片数据分区 A slice_data_partition_a_layer_rbsp()	2	VCL	不 可 应用
3	编码的码片数据分区 B slice_data_partition_b_layer_rbsp()	3	VCL	不 可 应用
4	编码的码片数据分区 C slice_data_partition_c_layer_rbsp()	4	VCL	不 可 应用
5	编码的 IDR 画面的码片	2,3	VCL	VCL

[0113]

	slice_layer_without_partitioning_rbsp()			
6	补充增强信息 (SEI) sei_rbsp()	5	非 VCL	非 VCL
7	序列参数集 seq_parameter_set_rbsp()	0	非 VCL	非 VCL
8	画面参数集 pic_parameter_set_rbsp()	1	非 VCL	非 VCL
9	访问单元定界符 access_unit_delimiter_rbsp()	6	非 VCL	非 VCL
10	序列的结束 end_of_seq_rbsp()	7	非 VCL	非 VCL
11	流的结束 end_of_stream_rbsp()	8	非 VCL	非 VCL
12	填充数据 filler_data_rbsp()	9	非 VCL	非 VCL
13	序列参数集扩展 seq_parameter_set_extension_rbsp()	10	非 VCL	非 VCL
14	前缀 NAL 单元 Prefix_nal_unit_rbsp()	2	非 VCL	后缀 非独立
15	子集序列参数集 Subset_seq_parameter_set_rbsp()	0	非 VCL	非 VCL
16..18	保留的		非 VCL	非 VCL
19	编码的没有分区的辅助编码画面的码片 slice_layer_without_partitioning_rbsp()	2,3,4	非 VCL	非 VCL
20	编码的码片扩展 Slice_layer_extension_rbsp()	2,3,4	非 VCL	VCL
21..23	保留的		非 VCL	非 VCL

[0114]

24..31	未指定的		非 VCL	非 VCL
--------	------	--	----------	----------

[0115] 以下在表 4 中示出了提出的 3DV 前缀 NAL 单元的更详细的描述

[0116] 表 4-3DV 前缀 NAL 单元

[0117]

3dv_prefix_nal_unit() {	C	描述符
3dv_view_id	All	u(7)
3dv_layer_id	All	u(3)
reserved_bits	All	u(6)
}		

[0118] 如表 4 中所图示的,3DV 前缀 NAL 单元可以包括 3dv_view_id 和 3dv_layer_id。3dv_view_id 指定与 3DV 视图相关联的帧的 3DV 视图 ID 号。另外,3dv_layer_id 指定相关联的帧的 3DV 层 ID 号。reserved_bit 允许将 NAL 单元字节对齐。应理解,用于每个语法元素的比特的数目和它们的编码方法仅作为示例而提供。还应注意,NAL 单元 16 的首标可以包括标准的第一字节,如以下表 9 中的前三个元素中的。在该实施例中,NAL 单元 16 可以包括首标和扩展的首标并且不需要包括净荷(payload)。例如,可以在每个 3DV 层帧之前或者在 3DV 层帧的每个码片之前传输 NAL 单元 16。

[0119] 为了更好地说明可以怎样采用 3DV 前缀 NAL 单元,参照图 10 进行,图 10 示出了包括 3DV 视图 1002、1004 和 1006 的结构 1000 的 3DV 内容的示例。这里视图 1002、1004 和 1006 提供了相同场景或对象的不同透视。在该示例中,每个 3DV 视图进一步由两个层组成: 2DV 视图 1010 加上其深度 1008。图 10 中的箭头示出了不同视图和层之间的编码依赖性。例如,出于编码目的,B 视图 1004(双向预测的视图)依赖于并且参考基本视图 1002 和 P 视图 1006(预测视图)。类似地,P 视图 1006 依赖于并且参考基本视图 1002。这里,每个 3DV 视图的深度层 1008 参考对应的 3DV 视图的 2DV 视图层 1010。应注意,本领域普通技术人员鉴于在这里提供的教导的情况下可以将 3DV 视图和依赖性扩展到具有附加的 3DV 补充层的 3DV 内容,诸如依据 MVD、LDV、DES 格式的那些。还应注意,图 10 中提供的依赖性仅仅是示例并且 3DV 前缀 NAL 的使用允许各种其它的依赖性。

[0120] 在图 11 中图示了根据该实施例的图 10 中的 3DV 内容的 NAL 单元流。具体地,图 11 提供了视频表示的不同时间 T0 1102 和 T1 1110 的 NAL 单元 1100 的流。这里,视图 1104 和视图 1112(3DV 视图 0) 分别对应于时间 T0 和 T1 处的基本视图 1002,在于它们与作为基本视图 1002 的相同透视或者视点相关联。类似地,视图 1106 和视图 1114(3DV 视图 2) 分别对应于时间 T0 和 T1 的 P 视图 1006,而视图 1108 和视图 1116(3DV 视图 1) 分别对应时间 T0 和 T1 的 B 视图 1004。

[0121] 如图 11 中所示,每个 3DV 视图由 2D 视图层和深度层组成。然而,应理解,可以在其它实施例中采用附加补充层。在此,视图 1104 由 2D 视图层 1118 和深度层 1120 组成。2D

视图层 1118 自身由 NAL 单元 16 (1126)、14 (1128), 和 5 (1130) 组成, 而深度层 1120 由 NAL 单元 16 和 NAL 单元 20 (1132) 组成。继而, 视图 1106 的 2D 视图层 1122 和深度层 1124 它们自身由 NAL 单元 16 和 NAL 单元 20 组成, 如图 11 所图示的。视图 1112 由包括 NAL 单元 16 和 20 的深度层以及包括 NAL 单元 16、14 和 1 (1134) 的 2D 视图层 1136 二者组成。

[0122] 图 11 的箭头指示 NAL 单元的传输顺序。例如, NAL 单元 16 (1126) 在 NAL 单元 14 (1128) 之前传输, NAL 单元 14 (1128) 其自身在 NAL 单元 5 (1130) 之前传输, 等等。在表 2 和 4 中定义 NAL 单元 16, 而在表 3 中定义图 11 中图示的其它 NAL 单元。例如, NAL 单元 5 包括瞬时解码刷新 (IDR) 画面的编码码片的视频数据, 所述瞬时解码刷新 (IDR) 画面由仅仅帧内码片或者 SI 码片组成, 如 AVC 草案中定义的。通常, IDR 画面是仅仅使用帧内预测或者仅仅使用帧内预测和预测样本的量化而编码的。此外, NAL 单元 1 包括非 IDR 画面的编码码片的视频数据, 诸如双向 (B) 编码的画面或者预测 (P) 编码的画面, 其继而可以参考其它画面、3DV 层或者 3DV 视图。继而, 如例如图 10 中所指示的, NAL 单元 20 是可以参考另一层或者另一 3DV 视图的编码的码片扩展。还应注意, 图 11 中所示的 NAL 单元 1、5 和 20 是许多这种单元的代表并且已经为了易于表示而被截断 (truncate)。例如, 对 2D 视图 1118 已经传输了前缀单元 16 和 14 之后, 可以传输若干 NAL 单元 5 (1130), 直到已经发送了对应帧的所有码片为止。类似地, 在对于深度视图已经传输了前缀 NAL 单元 16 之后, 可以传输组成深度层帧的多个 NAL 单元 20。图 11 中的 NAL 单元 1 类似于对应于 2D 视图层 1136 的帧的被截断的码片代表。

[0123] 如上所述, 每个 NAL 单元 14 是指示其对应层的 MVC 视图 ID 的前缀 NAL 单元。例如, NAL 单元 14 包括其对应 2D 视图层 1118 的 MVC 视图 ID。类似地, NAL 单元 20 也包括其对应 3DV 层的 MVC 视图 ID。在该实施例中, 每个 3DV 层被编码为单独的 MVC 视图并且因此在其编码期间被分配了唯一的 MVC view_id。编码器, 诸如如上讨论的编码 300, 可以使用 MVC view_id 来指示序列参数集 (SPS) 中层和 / 或帧之间的依赖性, 如以下关于实施例 5-7 进一步所讨论的, 并且可以指定前缀 NAL 单元 16 中的对应 3dv_view_id 和 3dv_layer_id, 使得诸如解码器 400 之类的解码器可以使用 3DV 前缀 NAL 单元以正确的方式解释并且解码帧。

[0124] 作为示例, 可以如表 5 中设置每个 3DV 层的 MVC view_id。因此, 在实施例 1 的架构中, 在具有 MVC view_id 等于 4 的任何 NAL 单元之前可以是具有 3dv_view_id 被设置为 2 和 3dv_layer_id 被设置为 0 的前缀 NAL 单元 16。此处分配的实际值是任意的并且可以变化, 只要其中每一个与不同透视或者观看点对应的不同 3DV 视图被唯一地识别并且它们对应的 3DV 层被充分地识别和传达。还应注意, 表 5 中的值在不同时间是一致的。例如, 视图 1104 和 1112 共享相同 MVC 视图、3DV 视图和 3DV 层 ID。

[0125] 表 5- 实施例 1 中的 MVC view_id 的示例

[0126]

MVC view_id	3dv_view_id	3dv_layer_id	描述
0	0	0	2D 视频
1	0	1	深度

2	1	0	2D 视频
3	1	1	深度
4	2	0	2D 视频
5	2	1	深度

[0127] 应理解,实施例 1 中定义的比特流是 MVC 兼容的并且每个 3DV 视图及其所有的层可以由常规的 MVC 解码器解码。因此,3DV 前缀 NAL 单元 16 允许 MVC 兼容的解码器解码所有传输的 3DV 层,包括 3DV 补充层。然而,尽管常规的 MVC 解码器可能不会意识到怎样将解码数据组织为 3DV 格式,但 NAL 单元 16 的使用允许由实施例在编码级别上对 3DV 视图和层的同步。例如,依据前面公开的教导,图 3 中图示的编码器 300 的 3DV 参考缓冲器 316 可以在比特流 318 中包括合适的 3DV 前缀单元,而图 4 的解码器 400 的 3DV 参考缓冲器 414 可以解释比特流 318 中的 NAL 单元并且相应地使用 NAL 单元构建和格式化 3DV 内容,使得它们符合以上关于图 10 和图 11 讨论的结构。

[0128] 应注意,实现了 MVC 向后兼容性在于可以依据 MVC 通过常规的 MVC 解码器来解码并且格式化 3DV 视图的每个 2DV 视图层。然而,因为深度层和其它 3DV 补充层将包括它们自身唯一的 MVC 视图 ID,所以 3DV 补充层将被 MVC 解码器解释为单独的 MVC 视图。由此,如果依据 MVC 来格式化并且显示 3DV 补充层,则显示的图像将一般不具有三维效果。同样,用户可以彻底搜索并且试图显示 MVC 视图,直到呈现可观看的 3D 图像为止。此处,无论何时两个 2D 视图层被选择 / 显示并且呈现给用户的每只眼睛,将呈现可观看的 3D 视图。

[0129] 另外,如果用户的显示器被配置为接受例如使用实施例 1 所传输的 3DV 补充层并且产生 3D 图像,则用户还可以能够观看 3D 图像。例如,用户的显示器可以接受 LDV 格式化的输入并且从该输入产生 3D 图像。在这种情形下,用户例如可以在用户的显示器上选择一模式以指示输入是 LDV 格式。

[0130] 实施例 2 :3DV 下重用 MVC view_id。

[0131] 依据实施例 2,作为实施例 1 的可替换实现方式,提出了对 NAL 单元首标的新的编码和解码处理。此处,以上关于实施例 1 提供的细节应用于实施例 2,除了采用涉及 MVC view_id 的特定的编号方法使得避免对 3DV 前缀单元 16 的使用。例如,由于 MVC view_id 被定义为具有 10 比特,所以 MVCview_id 的 3 个最低有效比特可以指示 3dv_layer_id,而 MVC view_id 的 7 个最高有效比特可以指示 3dv_view_id。结果,表 5 中的 MVC view_id 可以如以下的表 6 所设置。因此,图 11 中提供的 3DV 内容对于实施例 2 将相同,除了在实施例 2 中将不存在 NAL 单元 16,并且解码器可以存储并使用表 6,通过交叉参考提取的 MVC 视图 ID 与 3DV 视图 ID 和 3DV 层 ID 而从提取的比特流中的 MVC 视图 ID 中确定 3DV 视图 ID 和 3DV 层 ID。相应地,NAL 前缀单元 14 和 / 或 NAL 单元 20 可以依据涉及 MVC 视图 ID 的编号方法而被配置。此处,如所讨论的,可以采用 MVC 视图 ID 来传达 3DV 视图 ID 和 3DV 层 ID 以允许在编码级别上同步并且格式化 3DV 内容。

[0132] 表 6- 实施例 2 中的 MVC view_id 的示例

[0133]

MVC view_id	3dv_view_id	3dv_layer_id	描述
0	0	0	2D 视频
1	0	1	深度
8	1	0	2D 视频
9	1	1	深度
16	2	0	2D 视频
17	2	1	深度

[0134] 实施例 3 :3DV NAL 单元扩展

[0135] 在实施例 1 和 2 中,使用某些 MVC 编码技术来编码所有的 3DV 层,并且同样,所有的 3DV 层通过常规的 MVC 解码器是可解码的。然而,实现当前 MVC 标准的常规的 MVC 解码器不将各种 3DV 层中的每一个构成为 3DV 格式,如以上讨论的。在实施例 3 中,提出了一种编码框架,其允许引入不是当前 MVC 标准的一部分的、以及可以应用到某些 3DV 视图和 / 或某些 3DV 层的附加的编码技术。

[0136] 为了实现该目标,可以采用以上表 2 中所示的,在此被称为“21”的新颖的新 NAL 单元类型。类似于 NAL 单元 16,实施例 3 的新的 NAL 单元的选择的参考编号可以是表 3 中 AVC 草案保留的任何编号。此处,不需要 MVC 解码器解码的任何 3DV 视图和 / 或 3DV 层可以使用 NAL 单元类型 21 来解码 3DV 内容。另外,可以由 MVC 解码器解码并适当解释的所有 2D 视图层可以以常规 NAL 单元类型来编码,诸如如以上讨论的 1、5 和 20,并且它们被称为 MVC 兼容的 2D 视图。MVC 兼容的 2D 视图之前可以是 3DV 前缀 NAL 单元,诸如 NAL 单元 16,如关于实施例 1 描述的;或者可以指定 MVC view_id 编号方法,以便避免 3DV 前缀 NAL 单元,如关于实施例 2 所描述的。

[0137] 类似于以下在表 7 中提供的 AVC 草案 MVC NAL 单元首标扩展,以下在表 8 中提出并且提供了新的 3DV NAL 单元首标扩展。

[0138] 表 7-NAL 单元首标 MVC 扩展

[0139]

nal_unit_header_mvc_extension() {	C	描述符
non_idr_flag	All	u(1)
priority_id	All	u(6)
view_id	All	u(10)
temporal_id	All	u(3)

anohor_pic_flag	All	u(1)
inter_view_flag	All	u(1)
reserved_one_bit	All	u(1)
}		

[0140] 表 8-NAL 单元首标 3DV 扩展

[0141]

nal_unit_header_3dv_extension() {	C	描述符
non_idr_flag	All	u(1)
priority_id	All	u(6)
3dv_view_id	All	u(7)
3dv_layer_id	All	u(3)
temporal_id	All	u(3)
anchor_pic_flag	All	u(1)
inter_view_flag	All	u(1)
reserved_one_bit	All	u(1)
}		

[0142] 如表 7 和表 8 中所示,3DV NAL 单元首标扩展可以包括与 MVC NAL 单元首标扩展相同的语法元素,除了 MVC NAL 单元首标扩展的语法元素 view_id 在 3DV NAL 单元首标扩展中的两个语法元素 3dv_view_id 和 3dv_layer_id 取代。此处,在实施例 3 中,3dv_view_id 指定相关联的帧的 3DV 视图 ID 编号。相同的 3dv_view_id 在来自相同视图位置的 3DV 视图层之间共享。继而,3dv_layer_id 指定相关联的帧的 3DV 层 ID 编号。在表 9 中示出了对 nal_unit_header_3dv_extension() 的调用。

[0143] 表 9-NAL 单元语法

[0144]

nal_unit(NumBytesInNALunit)	C	描述符
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)

nal_unit_type	All	u(5)
NumBytesInRBSP = 0		
nalUnitHeaderBytes = 1		
If(nal_unit_type == 14 nal_unit_type == 20) {		
svc_extension_flag	All	u(1)
If(svc_extension_flag)		
nal_unit_header_svc_extension()	All	
Else		
nal_unit_header_mvc_extension()	All	
nalUnitHeaderBytes+ = 3		
}		
If(nal_unit_type == 21) {		
nal_unit_header_3dv_extension()		
nalUnitHeaderBytes+ = 3		
}		
for(I = nalUnitHeaderBytes ; i < NumBytesInNALunit ; i++) {		
If(i+2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp_byte[NumBytesInRBSP++]	All	b(8)
rbsp_byte[NumBytesInRBSP++]	All	b(8)
i+ = 2		
emulation_prevention_three_byte/* 等于 0x03 */	All	f(8)
} else		
rbsp_byte[NumBytesInRBSP++]	All	b(8)

}		
}		

[0145] 此处, 已经将 if(nal_unit_type == 21) {...} 语句添加到 AVC 草案中描述的 NAL 单元语法中。

[0146] 在图 12 中提供了依据实施例 13 的 NAL 单元流 1200 的示例, 其中采用新的 NAL 单元类型 21。此处, 避免 3DV 前缀 NAL 单元类型的使用, 这是由于在 NAL 单元首标解析处理中指定了 view_id 编号。NAL 单元流 1200 是将实施例 3 应用于图 10 中提供的 3DV 内容示例的说明。如上讨论的, 在 3DV 视图和 3DV 层之间的依赖性的不同变型以及所使用的 3DV 层的不同变型可以依据各种实现方式而不同。

[0147] 类似于流 1100, 流 1200 可以包括不同时间的不同的视图集合, 视图 1204、1206 和 1208 对应于 T0 (1202), 视图 1212、1214 和 1216 对应于时间 T1 (1210)。视图 1204 和视图 1212 (3DV 视图 0) 分别对应于时间 T0 和 T1 处的基本视图 1002, 在于它们与作为基本视图 1002 的相同透视或者视点相关联。类似地, 视图 1206 和视图 1214 (3DV 视图 2) 分别对应于时间 T0 和 T1 的 P 视图 1006, 而视图 1208 和视图 1216 (3DV 视图 1) 分别对应时间 T0 和 T1 的 B 视图 1004。每个 3DV 视图由 2D 视图层和深度层组成。至于流 1100, 应理解, 可以在其它实施例中采用附加补充层。视图 1204 由 2D 视图层 1218 和深度层 1220 组成。继而, 2D 视图层 1218 由 NAL 单元 14 (1226) 和 5 (1230) 组成, 而深度层 1220 由 NAL 单元 21 (1230) 组成。此外, 视图 1206 由 2D 视图 1222 和深度视图 1224 组成, 2D 视图 1222 包括 NAL 单元 20, 深度视图 1224 由 NAL 单元 21 组成。另外, 视图 1212 的 2D 视图 1236 由 NAL 单元 14 和 1 组成。

[0148] 以上已经关于图 11 描述了 NAL 单元 1、5、14 和 20。与 NAL 单元 14 和 20 使用的表 7 的 MVC NAL 单元首标扩展相反, NAL 单元 21 采用表 8 的 3DV NAL 单元首标控制。使用新颖的 3DV NAL 单元首标扩展使得能够在编码级别上将 3DV 层同步到 3DV 内容格式同时允许应用新的编码方法。与 NAL 单元 16 不同, NAL 单元 21 可以包括对应的视频数据的净荷。更一般地, 该净荷可以包括通常被称为对应的编码画面的数据的画面数据。该画面数据可以来自与任何层, 诸如, 例如, 2D 视频、深度、遮挡视频、遮挡深度, 或者透明性。

[0149] 还应注意, 类似于图 11, 图 12 的箭头指示 NAL 单元的传输顺序。此外, 以与其中截断图 11 的 NAL 单元 1、5 和 20 的方式相同的方式截断图 12 中的 NAL 单元 1、5、20 和 21。另外, 实施例 3 是 MVC 可兼容的, 这是在于, 2D 视图层可以通过常规的解码器来解码并且依据 MVC 来组合以允许生成并显示 3D 内容。

[0150] 现在转到图 13 和图 14, 分别图示了解码和编码依据实施例 3 例示的 3DV 内容流的方法 1300 和 1400。应注意, 可以在图 4 的解码器 400 中执行并实现方法 1300, 而可以在图 3 的编码器 300 中执行并实现方法 1400。方法 1300 和 1400 二者采用以上表 9 中提供的语法。

[0151] 方法 1300 可以在步骤 1302 开始, 其中解码器 400 可以读取所接收的 NAL 单元中的以上在表 9 中并且也在 AVC 草案中描述的 nal_ref_idc。

[0152] 在步骤 1304, 解码器 400 可以读取 NAL 单元类型。

[0153] 在步骤 1306, 解码器 400 可以确定 NAL 单元类型是否是 14。如果 NAL 单元类型是 14, 则解码器 400 可以前进到 1308 并且解析当前处理的 NAL 单元的剩余部分以获得 MVC 视图 ID。在实施例 3 的该具体实现方式中, 例如, 如以上关于第二实施例描述的, 通过 MVC 视图 ID 来指示 3DV 视图 ID 和 3DV 层 ID。

[0154] 由此, 在步骤 1310, 解码器 400 可以从 MVC 视图 ID 中获得 3DV 视图 ID 和 3DV 层 ID, 如以上例如关于实施例 2 所讨论的。

[0155] 在步骤 1312, 解码器 400 可以读取并解析所接收的下一 NAL 单元。下一 NAL 单元应该是类型 1 或者类型 15。由此, 如果解码器确定下一 NAL 单元不是类型 1 或者类型 15, 则已经发生了错误。

[0156] 在步骤 1314, 解码器 400 可以解码当前处理的 NAL 单元的当前码片数据。

[0157] 在步骤 1316, 解码器 400 可以确定处理的 NAL 单元是否对应于当前帧的结尾。如果处理的 NAL 单元不对应于当前帧的结尾, 则可以通过解码器 400 重复步骤 1312-1316。

[0158] 在到达当前帧的结尾时, 该方法然后可以前进到步骤 1318, 其中解码器 400 可以将解码帧与其 3DV 视图 ID 及其 3DV 层 ID 发送給其输出缓冲器, 诸如, 例如, 3DV 参考 / 输出缓冲器 414, 3DV 参考 / 输出缓冲器 414 继而可以以 3DV 格式来配置该帧用于显示, 如上所讨论的。

[0159] 在步骤 1320, 解码器 400 可以确定是否已经到达比特流或者序列的结尾。如果还没有到达比特流或者序列的结尾, 则该方法可以前进到步骤 1302 并且解码器 400 可以重复方法 1300。如果到达了比特流或者序列的结尾, 则方法 1300 可以结束。

[0160] 返回到步骤 1306, 如果解码器 400 确定当前处理的 NAL 单元的 NAL 单元类型不是类型 14, 则该方法可以前进到步骤 1322, 其中解码器 400 可以确定当前处理的 NAL 单元的 NAL 单元类型是否是 20。如果当前处理的 NAL 单元是类型 20, 则该方法可以前进到步骤 1324, 其中解码器 400 可以解析当前处理的 NAL 单元的剩余部分以获得 MVC 视图 ID。在实施例 3 的该特定实现方式中, 3DV 视图 ID 和 3DV 层 ID 由 MVC 视图 ID 指示, 例如, 如以上关于实施例 2 所描述的。

[0161] 相应地, 在步骤 1326, 解码器 400 可以从 MVC 视图 ID 中获得 3DV 视图 ID 和 3DV 层 ID, 如以上例如关于实施例 2 所讨论的。

[0162] 在步骤 1328, 解码器 400 可以解码当前处理的 NAL 单元的当前码片数据。

[0163] 在步骤 1330, 解码器 400 可以确定处理的 NAL 单元是否对应于当前帧的结尾。如果处理的 NAL 单元不对应于当前帧的结尾, 则该方法可以前进到步骤 1332, 其中解码器 400 可以读取并解析所接收的下一 NAL 单元。该下一 NAL 单元应是类型 20。因此, 如果解码器确定下一 NAL 单元不是类型 20, 则已经发生了错误。其后, 可以由解码器 400 重复步骤 1326-1330。

[0164] 如果在步骤 1330, 解码器 400 确定到达了当前帧的结尾, 则该方法可以前进到步骤 1318, 其中解码器 400 可以将解码的帧与其 3DV 视图 ID 及其 3DV 层 ID 发送給其输出缓冲器, 如上所讨论的。其后, 该方法可以前进到步骤 1320, 并且可以被重复或者终止, 如上所讨论的。

[0165] 返回到步骤 1322, 如果解码器 400 确定当前处理的 NAL 单元不是类型 20, 则该方法可以前进到步骤 1334, 其中解码器确定当前处理的 NAL 单元是否是类型 21。如果当前处

理的 NAL 单元是类型 21, 则该方法可以前进到步骤 1336, 其中解码器 400 可以解析当前处理的 NAL 单元的剩余部分并且获得由 3DV NAL 单元首标扩展所提供的 3DV 视图 ID 和 3DV 层 ID。

[0166] 在步骤 1338, 解码器 400 可以解码当前处理的 NAL 单元的当前码片数据。

[0167] 在步骤 1340, 解码器 400 可以确定处理的 NAL 单元是否对应于当前帧的结尾。如果处理的 NAL 单元不对应于当前帧的结尾, 则该方法可以前进到步骤 1342, 其中解码器 400 可以读取并解析所接收的下一 NAL 单元。该下一 NAL 单元应是类型 21。因此, 如果解码器确定下一 NAL 单元不是类型 21, 则已经发生了错误。其后, 可以由解码器 400 重复步骤 1338-1340。

[0168] 如果在步骤 1340, 解码器 400 确定到达了当前帧的结尾, 则该方法可以前进到步骤 1318, 其中解码器 400 可以将解码的帧与其 3DV 视图 ID 及其 3DV 层 ID 发送到其输出缓冲器, 如上所讨论的。其后, 该方法可以前进到步骤 1320, 并且可以被重复或者终止, 如上所讨论的。

[0169] 返回到步骤 1334, 如果解码器 400 在步骤 1334 确定当前处理的 NAL 单元不是类型 21, 则该方法可以前进到步骤 1344, 其中解析当前处理的 NAL 单元的剩余部分, 其可以意图用于序列参数集 (SPS)、画面参数集 (PPS) 或者其它目的。其后, 该方法可以前进到步骤 1320 并且可以被重复或者终止, 如以上讨论的。

[0170] 再次参照图 14, 用于编码依据实施例 3 的 3DV 内容流的方法 1400 可以开始于步骤 1402, 其中编码器 300 可以读取其配置简档 (profile)。

[0171] 在步骤 1404, 编码器 300 可以写入 SPS 和 / 或 PPS NAL 单元。

[0172] 在步骤 1406, 编码器 300 可以读取要编码的下一帧。

[0173] 在步骤 1408, 编码器 300 可以确定当前处理的帧是否应是 AVC 兼容的视图。如果当前处理的帧应是 AVC 兼容的视图, 则该方法可以前进到步骤 1410, 其中编码器 300 可以编码当前帧的下一码片。

[0174] 在步骤 1412, 如果当前帧的当前处理的码片是当前帧的第一码片, 如编码器 300 所确定的, 则编码器 300 可以写入具有 NAL 单元类型例如为 14 的 MVC 前缀 NAL 单元。

[0175] 在步骤 1414, 编码器 300 可以将当前码片封装到 NAL 单元, 诸如例如, 类型 1 或 5 的 NAL 单元。

[0176] 在步骤 1416, 编码器 300 可以写入其中在步骤 1414 中封装了当前码片的 NAL 单元。

[0177] 在步骤 1418, 编码器 300 可以确定其是否已经到达当前帧的结尾。如果编码器还没有到达当前帧的结尾, 则该方法可以前进到步骤 1410 并且编码器 300 可以重复步骤 1410-1418。如果编码器已经到达了当前帧的结尾, 则该方法可以前进到步骤 1420, 其中编码器 300 可以确定对于序列或者比特流是否已经处理了所有的帧。如果已经处理了所有的帧, 则该方法可以结束。否则, 该方法可以前进到步骤 1406 并且编码器可以重复步骤 1406 和 1408。

[0178] 返回到步骤 1408, 如以上介绍的, 如果编码器 300 确定当前处理的帧不需要是 AVC 兼容的视图, 则该方法可以前进到步骤 1422, 其中编码器 300 可以确定当前处理的帧是否应是 MVC 兼容的视图。如果当前处理的帧应是 MVC 兼容的视图, 则该方法可以前进到步骤

1424,其中编码器 300 可以编码当前处理的帧的下一码片。

[0179] 在步骤 1426,编码器可以将当前码片封装到具有 NAL 单元类型例如为 20 的 NAL 单元。

[0180] 在步骤 1428,编码器 300 可以写入其中在步骤 1426 中当前码片被封装到的 NAL 单元。

[0181] 在步骤 1430,编码器 300 可以确定其是否已经到达当前帧的结尾。如果编码器还没有到达当前帧的结尾,则该方法可以前进到步骤 1424 并且编码器 300 可以重复步骤 1424-1430。如果编码器已经到达了当前帧的结尾,则该方法可以前进到步骤 1420,其中编码器 300 可以确定对于序列或者比特流是否已经处理了所有的帧。如果已经处理了所有的帧,则该方法可以结束。否则,该方法可以前进到步骤 1406 并且编码器可以重复步骤 1406 和 1408。

[0182] 返回到步骤 1422,如果编码器 300 确定当前处理的帧不需要是 MVC 兼容的视图,则该方法可以前进到步骤 1432,其中编码器 300 可以编码当前帧的下一码片。

[0183] 在步骤 1434,编码器可以将当前码片封装到具有 NAL 单元类型例如为 21 的 NAL 单元。

[0184] 在步骤 1436,编码器 300 可以写入其中在步骤 1434 中当前码片被封装到的 NAL 单元。

[0185] 在步骤 1440,编码器 300 可以确定其是否已经到达当前帧的结尾。如果编码器还没有到达当前帧的结尾,则该方法可以前进到步骤 1432 并且编码器 300 可以重复步骤 1432-1440。如果编码器已经到达了当前帧的结尾,则该方法可以前进到步骤 1420,其中编码器 300 可以确定对于序列或者比特流是否已经处理了所有的帧。如果已经处理了所有的帧,则该方法可以结束。否则,该方法可以前进到步骤 1406 并且编码器可以重复步骤 1406 和 1408。

[0186] 应理解,编码步骤 1410、1424 和 1432 以及解码步骤 1314、1328 和 1338 可以依据各种不同的编码方法和标准来执行,所述不同编码方法和标准允许符合以上关于例如图 10 和 12 讨论的实施例的结构和特征。

[0187] 另外,在引入用于 3DV 层的新的 NAL 单元类型 21 的情况下,可以为利用它们的不同特性的不同 3DV 层定义特定的编码技术。例如,在深度图用于寻找参考画面中的预测块时,2D 视图的解码可以依赖于其深度图的解码。此外,如上讨论的,可以采用其它这样的依赖性。

[0188] 还应注意,利用新颖的 NAL 单元类型 21,可以利用如表 10 中的 3dv_slice_layer_extension_rbsp() 来编码 3DV 视图 / 层,其中 3dv_slice_header() 和 3dv_slice_data() 可以包括修改的 slice_header() 和 slice_data()。

[0189] 表 10-3DV 码片层

[0190]

3dv_slice_layer_extension_rbsp() {	C	描述符
3dv_slice_header()	2	
3dv_slice_data()	2 3 4	
_rbsp_slice_trailing_bits()	2	
}		

[0191] 还应理解, 尽管已经单独地描述了实施例 1-3, 但如相关技术领域的技术人员鉴于在此提供的教导所理解的, 可以以各种方式来组合这些实施例中的一个或更多。例如, 可以以不同方式来编码相同帧的不同码片。例如, 可以根据实施例 1 和 / 或 2 以 MVC 兼容的方式来编码帧的某些码片, 而依据实施例 3 使用非 MVC 编码模式来编码其它码片。另外, 可以采用根据实施例 1 和 / 或 2 的 MVC 来编码 3DV 视图的某些层, 诸如例如, 2D 视图, 而可以应用根据实施例 3 的非 MVC 模式来编码 3DV 视图的其它层, 诸如例如, 遮挡视图。此处, 可以将具有 NAL 单元 1 和 / 或 5 的 NAL 单元 16 应用于一个或更多 3DV 视图的某些层, 而可以将 NAL 单元 21 应用于一个或更多 3DV 视图的其它层。

[0192] 第四实施例: 参考画面列表构建

[0193] 如上所指示的, 实施例可以针对参考画面列表构建处理。在以下此处讨论的实施例中, 每个画面具有其自身的参考画面列表。然而, 其它实现方式可以提供特定于 (并且用于) 多个画面的参考画面列表。例如, 参考画面列表可以被分配给时间上的整个画面序列, 或者时间上的给定点的跨越多个视图上的整个画面组, 或者画面的子集。例如, 画面的子集可以由码片或者单个宏块或者子宏块组成。该参考画面列表构建处理的输入是来自 NAL 单元首标的 inter_view_flag 和从序列参数集解码的视图依赖性信息。应理解, 可以将图 3 的编码器 300 和图 4 的解码器 400 二者配置为通过采用以下在此描述的教导来分别构建编码和解码比特流的参考画面列表。

[0194] 在该处理的第一阶段, 时间的参考画面和视图间参考画面可以被插入到初始的参考画面列表 RefPicListX (X 是 0 或者 1), 如例如在 AVC 或者 MVC 系统中可以进行的。如在 AVC 草案中定义的 RefPicListX 可以用作示例的初始参考画面列表。例如, RefPicListX, X 为 0, 可以用于编码或解码预测编码的画面的任何类型, 而 RefPicListX, X 为 1, 可以用于编码或解码双向编码的画面或者 B 画面。由此, B 画面可以具有两个参考画面列表, RefPicList0 和 RefPicList1, 而预测编码的画面的其它类型可以仅仅具有一个参考画面列表 RefPicList0。另外, 应注意, 此处, 时间参考与在时间上与参考列表所分配给的对应画面不同的画面的参考对应。例如, 参照图 11, 时间参考可以与用于编码 / 解码视图 1112 的视图 1104 的参考对应。继而, 视图间参考可以与用于编码 / 解码视图 1106 的视图 1104 的参考对应。通过将时间参考画面和视图间参考画面插入到参考画面列表中, 支持了现有的时间和视图间预测技术 (例如, 根据 AVC 和 / 或 MVC)。如已知的, AVC 系统可能在参考画面列表中包括时间参考画面, MVC 系统可能在参考画面列表中进一步包括视图间参考画面。

[0195] 该处理中的第二阶段可以包括添加可以对每个层独立地定义的层间参考画面。图 15 中提供了用于实施例 4 的一个层间预测结构 1500。结构 1500 中的箭头指示预测方向。例如, 特定视图的 2D 视频 (视图) 层 1502 (箭头离开的) 被用于编码视图的深度层 1504 (箭头抵达的) 的参考。相应地, 层间预测结构可以用于确定哪个 (哪些) 画面可以用于参考并且因此在参考画面列表中应包括哪个 (哪些) 画面。在结构 1500 中, 2D 视频层还用于遮

挡视频层 1506 和透明层 1510 二者的参考。另外,深度层 1504 用于遮挡深度层 1508 的参考。

[0196] 如图 15 所描绘的,对于层间预测结构 1500,每个 3DV 层具有至多一个层间参考。为编码给定的层,具有类似特性的层被用于参考。例如,再次参照图 2,遮挡视频层 206 包括 2D 视频层 202 的背景,而遮挡深度层 208 包括深度层 204 的背景。由此,为更好地利用跨越各层的冗余性,实现方式可以使用视图的 2D 视频层作为用于视图的遮挡层的参考并且可以使用视图的深度层作为用于视图的遮挡深度层的参考。其它实现方式可能允许用于给定的 3DV 层的多个层间参考。

[0197] 对于 2D 视频层画面,在第二阶段不需要做任何事,这是由于在 2D 视频层画面的实现方式中不需要使用层间参考。其它实施例可能真正对 2D 视频层提供层间参考。例如,给定视图的遮挡层可以用于参考的 2D 视频层的参考。避免使用 2D 视图层的层间参考的优点是可以由常规的 MVC 解码器来解码所有的 2D 视图层。应注意,在其它实现方式中,弯曲的(warp)的画面,诸如例如,合成的虚拟参考画面可以附加到参考列表。关于参考列表中的弯曲的画面参考位置,弯曲的画面参考可以被插入在具有高合成质量的初始参考列表的开始处或者具有中等合成质量的参考列表的结尾。以此方式使用弯曲的画面可以改进编码效率。

[0198] 返回到图 15,对于深度层画面 1504,2D 视频层画面 1502(被示为图 15 中的深度层的参考)可以在第二阶段中被附加到 RefPicListX 的结尾。在各种实现方式中,2D 视频画面参考被附加到参考列表的结尾处,而不是参考列表的开始处,这是因为其期望具有最小的冗余性(与第一阶段的时间和视图间参考相比较)并且期望有最小可能性被用作参考。由此,此处,在参考画面列表中的任何时间和视图间参考之后提供层间参考。

[0199] 对于遮挡视频层画面 1506,2D 视频层画面 1502 可以在第二阶段被附加到 RefPicListX 的开始处。2D 视频画面可以被附加到开始处(预先计划的(prepend)),在参考画面列表中任何时间和视图间参考之前,而不是在结尾处或者在中间,这是因为 2D 视频画面被期望具有可用的参考画面的最大冗余性并且有最大可能性被用作参考。

[0200] 对于遮挡深度层画面 1508,深度画面 1504 可以在第二阶段被附加到 RefPicListX 的开始处,在参考画面列表中的任何时间和视图间参考之前,这是由于期望的高级冗余性(与第一阶段的时间和视图间参考的任一个相比较)在遮挡层和深度层之间。

[0201] 对于透明层画面 1510,2D 视频层画面 1502 可以在第二阶段被附加到 RefPicListX 的结尾处,在参考画面列表中的任何时间和视图间参考之后,这是由于期望的低级冗余性(与第一阶段的时间和视图间参考的任一个相比较)在透明层和 2D 视频层之间。

[0202] 更一般地,可以在通过参考被使用的频度所确定的位置处将画面的层间参考插入到该画面的参考画面列表中。对于其中向每个参考指定优先级的实现方式,可以基于该参考被使用的频度来指定优先级。作为示例,一个实现方式通过宏块编码画面,并且每个宏块可以使用或者可以不使用来自参考画面列表的给定参考。对于该实现方式的每个宏块,在包括不同的编码模式和不同参考各种编码选项中,执行率失真优化。由此,给定的层间参考可能仅仅用于编码画面的宏块的子集。与有多少宏块使用在参考画面列表中可用的其它参考相比较,可以基于有多少宏块使用层间参考而确定向给定的层间参考指定的优先级。

[0203] 现在参照图 16 和 17,分别图示了用于构建编码和解码处理的参考画面列表的方

法 1600 和 1700。可以通过图 3 的编码器 300 来执行用于构建依据实施例 4 的一个实现方式的编码处理的参考画面列表的方法 1600。例如,可以将 3DV 参考缓冲器 316 配置为实现方法 1600。

[0204] 方法 1600 可以开始于步骤 1602,其中编码器 300 可以初始化参考画面列表 RefPicListX。如以上提到的,RefPicListX 可以依据 AVC 草案来初始化,X 是 0 或 1。例如,如上指示的,可以将时间和 / 或视图间参考画面插入到初始参考画面列表中。

[0205] 在步骤 1604,编码器 300 可以确定参考画面列表是否用于 2D 视频层画面。如果参考画面列表是用于 2D 视频层画面,则该方法可以前进到步骤 1622,其中编码器 300 可以继续编码当前正在被处理的码片。此后,该方法可以结束或者该方法可以重复来构建另一 3DV 层画面的参考画面列表。可替换地,如果 3DV 层画面是 B 画面,则该方法可以重复用于相同的 3DV 层画面以构建 RefPicList1。

[0206] 如果在步骤 1604,编码器 300 确定参考画面列表不是用于 2D 视频层画面,则该方法可以前进到步骤 1606,其中编码器 300 可以确定参考画面列表是否是用于深度层画面。如果参考画面列表是用于深度层画面,则该方法可以前进到步骤 1608,其中来自与深度层画面相同的 3D 视图的 2D 视频层画面被附加到参考画面列表的结尾。此后,该方法可以前进到步骤 1622,其中编码器 300 可以继续编码当前正在被处理的码片。然后该方法可以结束或者该方法可以重复来构建另一 3DV 层画面的参考画面列表。可替换地,如果 3DV 层画面是 B 画面,则该方法可以重复用于相同的 3DV 层画面以构建 RefPicList1。

[0207] 如果在步骤 1606,编码器 300 确定参考画面列表不是用于深度层画面,则该方法可以前进到步骤 1610,其中编码器 300 可以确定参考画面列表是否是用于遮挡视频层画面。如果参考画面列表是用于遮挡视频层画面,则该方法可以前进到步骤 1612,其中来自与遮挡视频层画面相同的 3D 视图的 2D 视频层画面被附加到参考画面列表的开始。此后,该方法可以前进到步骤 1622,其中编码器 300 可以继续编码当前正在被处理的码片。然后该方法可以结束或者可以重复来构建另一 3DV 层画面的参考画面列表。可替换地,如果 3DV 层画面是 B 画面,则该方法可以重复用于相同的 3DV 层画面以构建 RefPicList1。

[0208] 如果在步骤 1610,编码器 300 确定参考画面列表不是用于遮挡视频层画面,则该方法可以前进到步骤 1614,其中编码器 300 可以确定参考画面列表是否是用于遮挡深度层画面。如果参考画面列表是用于遮挡深度层画面,则该方法可以前进到步骤 1616,其中来自与遮挡深度层画面相同的 3D 视图的深度层画面被附加到参考画面列表的开始。此后,该方法可以前进到步骤 1622,其中编码器 300 可以继续编码当前正在被处理的码片。然后该方法可以结束或者可以重复来构建另一 3DV 层画面的参考画面列表。可替换地,如果 3DV 层画面是 B 画面,则该方法可以重复用于相同的 3DV 层画面以构建 RefPicList1。

[0209] 如果在步骤 1614,编码器 300 确定参考画面列表不是用于遮挡深度层画面,则该方法可以前进到步骤 1618,其中编码器 300 可以确定参考画面列表是否是用于透明层画面。如果参考画面列表是用于透明层画面,则该方法可以前进到步骤 1620,其中来自与透明层画面相同的 3D 视图的 2D 视频层画面被附加到参考画面列表的结尾。此后,该方法可以前进到步骤 1622,其中编码器 300 可以继续编码当前正在被处理的码片。然后该方法可以结束或者可以重复来构建另一 3DV 层画面的参考画面列表。可替换地,如果 3DV 层画面是 B 画面,则该方法可以重复用于相同的 3DV 层画面以构建 RefPicList1。类似地,如果在步

骤 1618, 编码器 300 确定该层不是透明层画面, 则该方法可以前进到步骤 1622, 其中编码器 300 可以继续编码当前正在被处理的码片。然后该方法可以结束或者可以重复来构建另一 3DV 层画面的参考画面列表。可替换地, 如果 3DV 层画面是 B 画面, 则该方法可以重复用于相同的 3DV 层画面以构建 RefPicList1。

[0210] 现在转到图 17 的方法 1700, 可以通过图 4 的解码器 400 来执行用于构建依据实施例 4 的一个实现方式的解码处理的参考画面列表的方法 1700。例如, 可以将 3DV 参考 / 输出缓冲器 414 配置为执行方法 1700。

[0211] 方法 1700 可以开始于步骤 1702, 其中解码器 400 可以解析接收的 NAL 单元和码片首标以提取 3DV 层标识符。例如, NAL 单元可以是如上关于实施例 1 讨论的 3DV 前缀单元 16、实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20、和 / 或实施例 3 的 NAL 单元 21。此外, 如上指示的, 可以由解码器 400 从解码器 400 接收的包括 3DV 内容的比特流中提取的其它信息可以包括来自 NAL 单元首标的 inter_view_flag 和从序列参数集解码的视图依赖性信息。此后, 可以初始参考画面列表 RefPicListX。如上提到的, RefPicListX 可以依据 AVC 草案来初始化, X 是 0 或 1。例如, 如上所指示的, 可以采用来自 NAL 单元首标的 inter_view_flag 和从序列参数集解码的视图依赖性信息来初始 RefPicListX。继而, 时间和 / 或视图间参考画面可以被插入到初始的参考画面列表中。

[0212] 可以以如上关于方法 1600 讨论的相同的方式由解码器 400 来执行方法 1700 的剩余步骤, 除了由步骤 1722 来取代步骤 1622 之外。例如, 可以与由编码器 300 执行步骤 1604-1620 的相同的方式由解码器 400 来执行步骤 1704-1720。然而, 在步骤 1722, 与编码当前处理的码片相反, 解码器继续解码当前处理的码片。

[0213] 应理解, 使用以上关于实施例 4 提供的教导, 本领域普通技术人员可以容易想到具有与以上关于图 15 描述的不同的层间依赖性的层间预测结构。

[0214] 相应地, 实施例 4 可以支持不同类型的层间预测。此外, 实施例 4 将参考画面列表适配于诸如例如以上关于图 15 描述的结构之类的层间预测结构。结果, 实施例 4 提供了基于系统的层间预测结构的参考画面列表, 而同时允许常规的 MVC 解码器提取 3DV 内容并且格式化该内容用于显示。

[0215] 应注意, 可以组织参考画面使得它们与 AVC 系统兼容。例如, 层间和视图间参考画面可以被多路复用为在时间上不同的画面。

[0216] 实施例 5: 用于子集 SPS 3DV 的新颖的 NAL 单元类型

[0217] 如上所指示的, 在至少一个实施例中, 可以扩展 SPS 使得可以发信号通知 3DV 格式的新的序列参数。扩展的 3DV 的 SPS 在以下被称为“子集 SPS3DV”。在实施例 5 中, 可以采用用于子集 SPS 3DV 的新颖的 NAL 单元类型。在实施例 6 和 7 中, 如下讨论的, 描述了子集 SPS 3DV 可以是怎样组成的。应理解, 所提出的参数不限于 SPS 内, 而是可以出现在 NAL 单元首标、画面参数集 (PPS)、补充增强信息 (SEI)、码片首标, 以及任何其它高级语法元素中。实施例还可以使用低级语法或者带外信息 (out-of-band information)。

[0218] 此处, 在实施例 5 中, 新的 NAL 单元类型可以用于指示子集 SPS 3DV。该实施例中的 NAL 单元类型编号可以是以上的表 3 中未被分配的任一个值, 如上述的, 已经从 AVC 草案转录。此外, 还应以与以上关于实施例 1 和 3 描述的新颖的 NAL 单元类型不同的方式来选择分配给 3DV 层的 VCL NAL 单元的新颖的 NAL 单元类型编号。结果, 17 被选择为用于子集 SPS

3DV 的 NAL 单元类型编号,其在以下的表 11 中被表示为 subset_seq_parameter_set_3dv_rbsp()。当然,可以选择其它 NAL 单元类型编号。如果不组合实施例,则 NAL 单元类型 16 或者 21 也可以被用于替代 17。关于以上的表 2,新添加了用于 nal_unit_type 17 和 nal_unit_type 18...20 的行。

[0219] 表 11-NAL 单元类型代码、语法元素类别,以及 NAL 单元类型分类

[0220]

nal_unit_type	NAL 单元的内容和 RBSP 语法结构	C	附录 A NAL 单 元类型 分类	附录 G 和附录 H NAL 单 元类型 分类
0...16	如表 2 中定义的			
17	subset_seq_parameter_set_3dv_rbsp()		非 VCL	非 VCL
18...20	保留的			
21	编码的 3DV 码片扩展 3dv_slice_layer_extension_rbsp()	2,3,4	非 VCL	VCL
22..23	保留的		非 VCL	非 VCL
24..31	如表 2 中定义的		非 VCL	非 VCL

[0221] 新颖的 NAL 单元类型可以允许 MVC 解码器或者 3DV 解码器确定是丢弃还是解析子集 SPS 3DV 内的内容。因为在 MVC 下,类型 17 被保留, MVC 解码器可以选取忽略或者丢弃该 NAL 单元中的数据。然而,3DV 解码器可以解析该单元中的数据,这允许 3DV 解码器解码 3DV 补充层。

[0222] 对于智能网络设备,例如,可以辨识新颖的 NAL 单元类型的路由器,如果网络提供商确定在特定环境下不应传输 3DV 补充层,该网络设备可以选择选择丢弃子集 SPS 3DV。可替换地或者另外地,可以解析并利用子集 SPS 3DV 中的内容以将流传输适配为可用的网络带宽。例如,利用 3DV 层预测结构的知识,当网络遭受突发通信量时,网络设备(例如,流传输服务器或者路由器)可以丢弃不用作参考的 3DV 层。

[0223] 比特流提取器,也被称为流服务器,也可以用于提取 3DV 流的各个部分。以上的路由器解析比特流并且做出关于是否转发(传输)各种 3DV 层的决定。比特流提取器还可以解析比特流,并且基于优先级做出转发决定,但是也可以将提取的比特流(也被叫做子比特流)调整到下游的设备。例如,比特流提取器可以仅提取 2D 视频和深度层,这是因为下游的接收器不使用遮挡或者透明层。此外,比特流提取器还可以仅提取与比特流中前两个视图对应的层,这是因为下游的接收器不使用超过两个视图。然而,另外,比特流提取器可以能够分析 3DV SPS,以及任何 MVC SPS,或者其它依赖性信息,来确定 2D 视频或者深度层

是否使用遮挡层或者透明层中的任一个作为层间参考,并且确定前两个视图是否使用其它视图的任一个作为视图间参考。如果需要其它层或者视图来合适地解码期望的 3DV 层,其可以是前两个视图的 2D 视频和深度层,则比特流提取器也将提取这些层和 / 或视图。

[0224] 注意,可以通过路由器或者比特流提取器来确定 3DV 层和 3DV 视图的优先级信息。然而,这样的优先级信息也可以例如通过被放置在 NAL 单元首标中在比特流中提供。这种优先级信息可以包括,例如,时间级别 ID、优先级 ID、视图 ID,以及与 3DV 信息有关的优先级 ID。

[0225] 现在参照图 18 和 19,分别图示了用于编码和解码依据实施例 5 的实现方式的子集 SPS 3DV 信息的 NAL 单元的方法 1800 和 1900。可以分别通过图编码器 300 的 3DV 参考缓冲器 316 和通过解码器 400 的 3DV 参考缓冲器 414 来执行方法 1800 和 1900。

[0226] 方法 1800 可以开始于例如步骤 1802,其中编码器 300 可以将 NAL 单元的 NAL 单元类型设置为 17。在步骤 1804,编码器 300 可以写入 NAL 单元首标。此后,在步骤 1806,编码器 300 可以组成并写入 SPS。例如,该 SPS 可以对应于 `subset_sequence_parameter_set_3dv_rbsp()` 并且可以如以下关于实施例 6 和 7 所讨论的组成和写入。

[0227] 方法 1900 可以开始于例如步骤 1902,其中解码器 400 可以接收 NAL 单元并且读取 NAL 单元首标。该 NAL 单元可以对应于在方法 1800 编码的 NAL 单元。在步骤 1904,解码器 400 可以提取 NAL 单元类型。如果 NAL 单元类型被设置为 17,则编码器可以读取并且解析 SPS。例如,该 SPS 可以对应于 `subset_sequence_parameter_set_3dv_rbsp()` 并且可以如以下关于实施例 6 和 7 所讨论的被读取和解析。

[0228] 实施例 6 :发信号通知 3DV 应用的参数的 SPS 的扩展

[0229] 如以上关于实施例 1-4 所讨论的,可以采用 3DV 补充层来支持增强的 3DV 再现能力,并且因此可以在 SPS 中发信号通知 3DV 层识别编号 (`3dv_layer_id`)。此外,如上讨论的,为了移除层间冗余性,可以利用层间编码并且可以将层间画面添加到参考画面列表以便利于层间编码。因此,为了允许解码器确定怎样利用层间参考来解码画面,编码器可以在 SPS 中指定层间预测结构。这种层间预测结构可以例如对应于如上关于图 15 讨论的结构 1500。

[0230] 在详细讨论 SPS 构造之前,应注意,依据各种实现方式,可以采用支持 3DV 内容的比特流的新颖的简档。2009 年 3 月,ITU-T “Advanced Video Coding for Generic audiovisual Services-Recommendation ITU-T H. 264”,下文中称为“更新的 AVC 草案”,提供了对简档的讨论并且通过引用在此并入。在一个或者多个实现方式中,可以将 `profile_idc` 设置为 218。更新的 AVC 草案描述了 AVC/MVC 中的其它现有的简档。

[0231] 以下提供的表 12 详述了以上关于实施例 5 提到的函数 `subset_sequence_parameter_set_3dv_rbsp()` 经历的处理。具体地,表 12,在语句 `else if(profile_idc == 218) {...}`,说明了依据实施例 6 的子集 SPS 3DV 的一个高级实现方式。可以用函数 `seq_parameter_set_3dv_extension()` 来实现详细的信令,例如如以下表 12 所示的。为 218 的 `profile_idc` 表示用于 MVC 标准的新的简档,并且是 3DV 简档。

[0232] 表 12-subset_seq_parameter_set_3dv_rbsp()

[0233]

subset_seq_parameter_set_3dv_rbsp() {	C	描述符
seq_parameter_set_data()	0	
if(profile_idc == 83 profile_idc == 86) {		
seq_parameter_set_svc_extension() /* 在更新的 AVC 草案的附录G中指定 */	0	
svc_vui_parameters_present_flag	0	u(1)
if(svc_vui_parameters_present_flag == 1)		
svc_vui_parameters_extension() /* 在更新的 AVC 草案的附录G中指定 */	0	
} else if(profile_idc == 118) {		
bit_equal_to_one /* 等于 1 */	0	f(1)
seq_parameter_set_mvc_extension() /* 在更新的 AVC 草案的附录H中指定 */	0	
mvc_vui_parameters_present_flag	0	u(1)
if(mvc_vui_parameters_present_flag == 1)		
mvc_vui_parameters_extension() /* 在更新的 AVC 草案的附录H中指定 */	0	
} else if(profile_idc == 218) {		
bit_equal_to_one /* 等于 1 */	0	f(1)
seq_parameter_set_3dv_extension() /* 在表13或14 中指定 */	0	
}		
Additional_extension2_flag	0	u(1)
if(additional_extension2_flag == 1)		
while(more_rbsp_data())		
additional_extension2_data_flag	0	u(1)
rbp_trailing_bits()	0	
}		

[0234] 图 20 和 21 分别图示了编码 2000 和解码 2100 依据实施例 6 的各种实现方式的 SPS 的方法的高级流程图。方法 2000 和 2100 分别编码和解码以例如表 12 给出的形式的 SPS。表 12 可以用于例如 NAL 单元类型 17。应理解,图 3 的编码器 300 可以被配置为执行方法 2000 而图 4 的解码器可以被配置为执行方法 2100。

[0235] 方法 200 可以开始于步骤 2002,其中编码器 300 可以设置 profile_idc。如上指示的,profile_idc 例如可以被设置为用于子集 SPS 3DV 的 218。

[0236] 在步骤 2004,编码器 300 可以写入序列参数集数据。例如,这样的数据可以对应于关于 seq_parameter_set_data() 语法结构的更新的 AVC 草案中描述的任何 SPS 数据。

[0237] 在步骤 2006,编码器 300 可以确定 profile_idc 是否被设置为 83 或 86。如果 profile_idc 被设置为 83 或 86,则该方法可以前进到步骤 2008,其中编码器 300 可以写入 seq_parameter_set_svc_extension(),设置并写入 svc_vui_parameters_present_flag,如在更新的 AVC 草案中讨论的。另外,在步骤 2008,如果 svc_vui_parameters_present_flag 被设置为 1,则编码器 300 可以写入 svc_vui_parameter_extension(),如在更新的 AVC 草案中讨论的。此后,该方法可以前进到步骤 2010,其在以下更详细地讨论。

[0238] 返回到步骤 2006,如果 profile_idc 未被设置为 83 或 86,则该方法可以前进

到步骤 2014, 其中编码器 300 可以确定 `profile_idc` 是否被设置为 118。如果 `profile_idc` 被设置为 118, 则该方法可以前进到步骤 2016, 其中编码器 300 可以设置 `bit_equal_to_one` 等于 1, 写入 `bit_equal_to_one`, 写入 `seq_parameter_set_mvc_extension()`, 设置并写入 `mvc_vui_parameters_present_flag`, 如在更新的 AVC 草案中描述的。如果 `mvc_vui_parameters_present_flag` 等于 1, 则编码器 300 可以写入 `mvc_vui_parameters_extension()`, 如在更新的 AVC 草案中描述的。此后, 该方法可以前进到步骤 2010, 其在以下更详细地讨论。

[0239] 如果在步骤 2014, 编码器 300 确定 `profile_idc` 未被设置为 118, 则该方法可以前进到步骤 2018, 其中编码器 300 可以确定 `profile_idc` 是否被设置为 218。如果 `profile_idc` 未被设置为 218, 则该方法可以前进到步骤 2022, 其中编码器 300 可以确定 `profile_idc` 是未知的并且可以输出错误消息。

[0240] 然而, 如果 `profile_idc` 被设置为 218, 则编码器 300 可以执行步骤 2020, 其中编码器 300 可以设置 `bit_equal_to_one` 为 1 并且写入 `bit_equal_to_one`。如上注意的, 在更新的 AVC 草案中描述了 `bit_equal_to_one`。在步骤 2020, 编码器 300 可以进一步写入 `seq_parameter_set_3dv_extension()`, 其在以下关于表 13 和 14 以及图 22-25 更详细地描述。如以下在此讨论的, `seq_parameter_set_3dv_extension()` 可以向解码器指示或者传达层间依赖性以允许解码器在画面的解码期间确定用于画面的合适的预测参考。此后, 该方法可以前进到步骤 2010。

[0241] 在步骤 2010, 编码器 300 可以设置 `additional_extension2_flag`, 并且如果 `additional_extension2_flag` 被设置为 1, 则编码器 300 可以写入所有的 `additional_extension2_data_flag`, 如在更新的 AVC 草案中讨论的。在步骤 2012, 编码器 300 可以写入 `rbsp_trailing_bits()`, 如在更新的 AVC 草案中描述的, 并且此后该方法可以结束。

[0242] 现在转到图 21, 图 21 图示了用于解码例如依据方法 2000 可能已经生成的 SPS 的方法 2100, 方法 2100 可以开始于步骤 2102, 其中解码器 400 可以从接收的比特流中解码序列参数集数据 `seq_parameter_set_data()` 并且可以设置 `profile_idc`, 如在更新的 AVC 草案中讨论的。

[0243] 在步骤 2104, 解码器 400 可以确定 `profile_idc` 是否被设置为 83 或 86。如果 `profile_idc` 被设置为 83 或 86, 则该方法可以前进到步骤 2106, 其中解码器 400 可以解码 `seq_parameter_set_svc_extension()` 并且解码 `svc_vui_parameters_present_flag`, 如在更新的 AVC 草案中讨论的。另外, 在步骤 2106, 如果 `svc_vui_parameters_present_flag` 被设置为 1, 则解码器 400 可以解码 `vui_vui_parameter_extension()`, 如在更新的 AVC 草案中讨论的。此后, 该方法可以前进到步骤 2108, 其在以下更详细地讨论。

[0244] 返回到步骤 2104, 如果 `profile_idc` 未被设置为 83 或 86, 则该方法可以前进到步骤 2112, 其中解码器 400 可以确定 `profile_idc` 是否被设置为 118。如果 `profile_idc` 被设置为 118, 则该方法可以前进到步骤 2114, 其中解码器 400 可以解码 `bit_equal_to_one`, 解码 `seq_parameter_set_mvc_extension()` 以及解码 `mvc_vui_parameters_present_flag`, 如在更新的 AVC 草案中描述的。另外, 如果 `mvc_vui_parameters_present_flag` 被设置为 1, 则解码器 400 可以解码 `mvc_vui_parameters_extension()`, 如在更新的 AVC 草案中描述的。此后, 该方法可以前进到步骤 2108, 其在以下更详细地讨论。

[0245] 如果在步骤 2112, 解码器 400 确定 `profile_idc` 未被设置为 118, 则该方法可以前

进到步骤 2116,其中解码器 400 可以确定 profile_idc 是否被设置为 218。如果 profile_idc 未被设置为 218,则该方法可以前进到步骤 2120,其中解码器 400 可以确定已经读取到未知的 profile_idc 并且可以输出错误消息。

[0246] 然而,如果 profile_idc 被设置为 218,则解码器 400 可以执行步骤 2118,其中解码器 400 可以解码 bit_equal_to_one 并且可以进一步解码 bseq_parameter_set_3dv_extension(),其在以下关于表 13 和 14 以及图 22-25 更详细地描述。此后,该方法可以前进到步骤 2108。

[0247] 在步骤 2108,解码器 400 可以解码 additional_extension2_flag,并且如果 additional_extension2_flag 被设置为 1,则解码器 400 可以解码所有的 additional_extension2_data_flag,如在更新的 AVC 草案中讨论的。在步骤 2110,解码器 400 可以解码 rbsp_trailing_bits(),如在更新的 AVC 草案中描述的,并且此后该方法可以结束。

[0248] 如上述的,表 13 示出了 seq_parameter_set_3dv_extension() 的一个实现方式,其中显式地发信号通知 3dv_layer_id 和层间预测结构。这种实现方式提供了大量的灵活性,这是因为可以指定不同顺序的 3DV 层和不同的层间预测结构。

[0249] 表 13-seq_parameter_set_3dv_extension() 的一个实现方式

[0250]

seq_parameter_set_3dv_extension() {	C	描述符
seq_parameter_set_mvc_extension()		
num_3dv_layer_minus1		ue(v)
for(i = 0 ; i <= num_3dv_layer_minus1 ; i++)		
3dv_layer_id[i]		ue(v)
for(i = 1 ; i <= num_3dv_layer_minus1 ; i++) {		
num_3dv_layer_refs_I0[i]		ue(v)
for(j = 0 ; j < num_3dv_layer_refs_I0[i] ; j++)		
3dv_layer_ref_I0[i][j]		ue(v)
num_3dv_layer_refs_I1[i]		ue(v)
for(j = 0 ; j < num_3dv_layer_refs_I1[i] ; j++)		
3dv_layer_ref_I1[i][j]		ue(v)
}		
}		

[0251] 如下给出表 13 的语义：

- [0252] num_3dv_layer_minus1 加 1 指示 3DV 层的数量。
- [0253] 3dv_layer_id[i] 指定第 i 个 3DV 层标识编号。
- [0254] 3dv_num_layer_refs_I0[i] 指定在用于具有 3DV 层标识编号为 3dv_layer_id[i] 的 3DV 层的参考画面列表 0 中的层间参考的编号。
- [0255] 3dv_layer_refs_I0[i][j] 指定在用于具有 3DV 层标识编号为 3dv_layer_id[i] 的 3DV 层的参考画面列表 0 中用作第 j 个层间参考的 3DV 层标识编号。
- [0256] num_3dv_layer_refs_I1[i] 指定在用于具有 3DV 层标识编号为 3dv_layer_id[i] 的 3DV 层的参考画面列表 1 中的层间参考的编号。
- [0257] 3dv_layer_refs_I1[i][j] 指定在用于具有 3DV 层标识编号为 3dv_layer_id[i] 的 3DV 层的参考画面列表 1 中用作第 j 个层间参考的 3DV 层标识编号。
- [0258] 为了更好地说明在实施例 6 中可以怎样采用表 13 的 seq_parameter_set_3dv_extension(), 参照图 22 和 23, 分别图示编码 2200 和解码 2300 子集 SPS 3DV 扩展的方法。应理解, 可以通过编码器 300 来实现方法 2200 而通过解码器 400 来实现方法 2300。
- [0259] 方法 2200 可以开始于步骤 2202, 其中编码器 300 可以编码 seq_parameter_set_mvc_extension(), 其在更新的 AVC 草案中描述。
- [0260] 在步骤 2204, 编码器 300 可以设置并编码 num_3dv_layer_minus1。如上提供的, num_3dv_layer_minus1 指示在要编码的 3DV 内容的 3DV 视图中采用的 3DV 层的总数。为了便于编码和解码, num_3dv_layer_minus1 的数字值比 3DV 层的实际数量小 1。
- [0261] 如上注意的, “i” 表示 3DV 层 id 编号。例如, 3DV 层 id 可以与以上在表 1 中定义的 3DV 层 id 对应。此处, 在步骤 2208, 编码器 300 可以为要编码的 3DV 内容中采用的每种类型的 3DV 层设置并编码 3DV 层 ID。由此, 编码器 300 在循环 2206 中迭代地处理每个 3DV 层 id, 直至达到 3DV 内容的 3DV 视图中采用的 3DV 层的总数。
- [0262] 在循环 2210 中, 如在循环 2210 的第一行注意的, 编码器 300 在循环 2210 中连续地处理每个 3DV 层 id 以设置并编码每个参考画面列表类型 0 以及潜在的 1 的每个 3DV 层的 3DV 层间参考。例如, 在步骤 2212, 编码器 300 可以设置并编码被分配了参考画面列表的 3DV 层 (由 “i” 表示的) 的参考画面列表 0 中的层间参考的总数 (num_3dv_layer_refs_I0[i])。应注意, 任何参考画面列表中的层间参考的数量依赖于采用的层间依赖性结构。例如, 在图 15 的结构 1500 中, 每个 3DV 层在被分配给 3DV 层的参考画面列表中至多具有一个层间参考。然而, 可以采用其它层间依赖性或者预测结构, 诸如以下关于实施例 7 在此讨论的结构。
- [0263] 在设置了参考画面列表 ‘0’ 中的 3DV 层 ‘i’ 的层间参考的总数之后, 编码器 300 可以在步骤 2216, 设置并编码 3DV 层 ‘i’ 的参考画面列表 ‘0’ 的层间参考。具体地, 编码器 300 可以在 3DV 层 ‘i’ 的参考画面列表 ‘0’ 中指定层间参考的 3DV 层 id (3dv_layer_ref_I0[i][j])。在图 22 中, 以及表 13 中, 3DV 层 ‘i’ 的参考画面列表 ‘0’ 中的层间参考可以由 ‘j’ 来表示, 使得可以在循环 2214 中迭代步骤 2216, 直至已经达到了参考画面列表 ‘0’ 的 3DV 层 ‘i’ 的层间参考的总数为止。
- [0264] 编码器 300 可以进一步被配置为对 3DV 层 ‘i’ 的任何参考画面列表 ‘1’ 提供层间参考。然而, 应理解, 如果特定的 3DV 层 ‘i’ 不具有参考画面列表 ‘1’, 可以跳过方法 2200 的以下步骤。如果 3DV 层 ‘i’ 具有参考画面列表 ‘1’, 方法可以前进到步骤 2218, 其中编码

器 300 可以设置并编码被分配了参考画面列表‘1’的 3DV 层 i 的参考画面列表 1 中的层间参考的总数 (num_3dv_layer_refs_I1[i])。

[0265] 在设置了参考画面列表‘1’中的 3DV 层‘i’的层间参考的总数之后,编码器 300 可以在步骤 2222,设置并编码 3DV 层‘i’的参考画面列表‘1’的层间参考。具体地,编码器 300 可以在 3DV 层‘i’的参考画面列表‘1’中指定层间参考的 3DV 层 id(3dv_layer_ref_I1[i][j])。类似于以上关于 3DV 层‘i’的参考画面列表‘0’提供的讨论,3DV 层‘i’的参考画面列表‘1’中的层间参考可以由‘j’来表示,使得可以在循环 2220 中迭代步骤 2222,直至已经达到了参考画面列表‘1’的 3DV 层‘i’的层间参考的总数为止。

[0266] 另外,如上指示的,在循环 2210,可以为要编码的 3DV 内容的 3DV 视图中采用的 3DV 层的每个层迭代步骤 2212 和 2218 以及循环 2214 和 2220,直至已经处理的所有这样的层。

[0267] 现在转到图 23,描述了使用 seq_parameter_set_3dv_extension() 解码比特流中接收的 SPS 3DV 扩展的方法 2300。方法 2300 可以开始于步骤 2302,其中解码器 400 可以解码 seq_parameter_set_mvc_extension(),其在更新的 C 草案中描述。

[0268] 在步骤 2304,解码器 400 可以解码并获得 num_3dv_layer_minus1。如上述的, num_3dv_layer_minus1 指示在 3DV 内容的 3DV 视图中采用的 3DV 层的总数。如上述的, num_3dv_layer_minus1 的数字值比 3DV 层的实际数量小 1。

[0269] 如上注意的,“i”表示 3DV 层 id 编号。例如,3DV 层 id 可以与以上在表 1 中定义的 3DV 层 id 对应。此处,在步骤 2308,解码器 400 可以解码并获得 3DV 内容中采用的每种类型的 3DV 层的 3DV 层 ID。由此,解码器 400 在循环 2306 中迭代地处理每个 3DV 层 id,直至达到 3DV 内容的 3DV 视图中采用的 3DV 层的总数并且获得每个 3DV 层 id。

[0270] 在循环 2310 中,如在循环 2310 的第一行注意的,解码器 400 在循环 2310 中连续地处理每个 3DV 层 id 以解码并获得每个参考画面列表类型 0 以及潜在的 1 的每个 3DV 层的 3DV 层间参考。例如,在步骤 2312,解码器 400 可以解码并获得被分配了参考画面列表的 3DV 层(由“i”表示的)的参考画面列表 0 中的层间参考的总数 (num_3dv_layer_refs_I0[i])。应注意,任何参考画面列表中的层间参考的数量依赖于采用的层间依赖性结构。例如,在图 15 的结构 1500 中,每个 3DV 层在被分配给 3DV 层的参考画面列表中至多具有一个层间参考。然而,可以采用其它层间依赖性或者预测结构,诸如以下关于实施例 7 在此讨论的结构。

[0271] 在获得了参考画面列表‘0’中的 3DV 层‘i’的层间参考的总数之后,解码器 400 可以在步骤 2316,解码并获得 3DV 层‘i’的参考画面列表‘0’的层间参考。具体地,解码器 400 可以在 3DV 层‘i’的参考画面列表‘0’中获得层间参考的 3DV 层 id(3dv_layer_ref_I0[i][j])。在图 23 中,以及表 13 中,3DV 层‘i’的参考画面列表‘0’中的层间参考可以由‘j’来表示,使得可以在循环 2314 中迭代步骤 2216,直至已经达到了参考画面列表‘0’的 3DV 层‘i’的层间参考的总数为止。

[0272] 解码器 400 可以进一步被配置获得 3DV 层‘i’的任何参考画面列表‘1’的层间参考。然而,应理解,如果特定的 3DV 层‘i’不具有参考画面列表‘1’,可以跳过方法 2300 的以下步骤。如果 3DV 层‘i’具有参考画面列表‘1’,方法可以前进到步骤 2318,其中解码器 400 可以解码并获得被分配了参考画面列表‘1’的 3DV 层 i 的参考画面列表 1 中的层间参

考的总数 (num_3dv_layer_refs_I1[i])。

[0273] 在获得了参考画面列表‘1’中的 3DV 层‘i’的层间参考的总数之后,解码器 400 可以在步骤 2322,解码并获得 3DV 层‘i’的参考画面列表‘1’的层间参考。具体地,解码器 400 可以在 3DV 层‘i’的参考画面列表‘1’中指定层间参考的 3DV 层 id(3dv_layer_ref_I1[i][j])。类似于以上关于 3DV 层‘i’的参考画面列表‘0’提供的讨论,3DV 层‘i’的参考画面列表‘1’中的层间参考可以由‘j’来表示,使得可以在循环 2320 中迭代步骤 2322,直至已经达到了参考画面列表‘1’的 3DV 层‘i’的层间参考的总数为止。

[0274] 另外,如上指示的,在循环 2310,可以为 3DV 内容的 3DV 视图中采用的 3DV 层的每个层迭代步骤 2312 和 2318 以及循环 2314 和 2320,直至已经处理的所有这样的层为止。由此,解码器 400 可以重构每个 3DV 层的(多个)参考画面列表以由此允许解码器 400 在解码画面期间确定接收的每个 3DV 层画面的层间参考。

[0275] 应注意,在网络装置解析关于 3DV 层和预测结构的信息时,它可以在传输来自不同 3DV 层的 NAL 单元期间分配不同的优先级。因此,当发生拥塞时,可以丢弃来自“较高”3DV 补充层(例如表 1 中较高的 3DV 层 id)的一些 NAL 单元以缓解业务量。

[0276] 实施例 7:发信号通知 3DV 应用的参数的 SPS 的可替换的扩展

[0277] 在某些实现方式中,因为使用的 3DV 层的潜在数量可能受到限制,并且继而,因为 3DV 层中的内容可能具有特定且一致的特征,用于编码和解码 3DV 的预测结构可以被预先配置并且对编码器和解码器二者是已知的。由此,不需要以显式方式,如例如在实施例 6 的表 13,来发信号通知或者传递特定的预测或层间依赖性结构。相反,在实施例 7 中,层间预测结构可能对编码器和解码器二者是已知的,由此简化向解码器传递 3DV 的扩展的 SPS。

[0278] 为了提供简单的示例,采用如上定义的以下 3DV 层:2D 视频层、深度层、遮挡视频层、遮挡深度层、以及透明层。

[0279] 以下提供了根据各种实现方式的可以采用的预定义的层间预测结构的示例。然而,应理解,可以在其它实现方式中利用其它预定义的层间预测结构。在该结构中,对于 2D 视频层,不使用 3DV 补充层作为层间预测参考。对于深度层,2D 视频层被用作层间预测参考。对于遮挡视频层,2D 视频层和深度层被用作层间参考。对于遮挡深度层,2D 视频层和深度层被用作层间参考。对于透明层,2D 视频层和深度层被用作层间参考。

[0280] 此处,在实施例 7 中,因为可以预定义层间预测结构,所以 3DV 的扩展的 SPS 可以简单地传递对于每个 3DV 视图是否存在每个层,如表 14 中所示。相应地,seq_parameter_set_3dv_extension() 可以简单地采用用于每个可能的 3DV 层的标志来指示是否在 3DV 内容中的每个 3DV 视图中采用它们。因此,3DV 的扩展的 SPS 不需要以任何显式方式来发信号通知或者传递层间预测结构。在一个实现方式中,层间预测结构是恒定的并且不能被改变。在另一实现方式中,层间预测结构使用表 13 来设置(例如,表 12 的初始出现,或者周期的出现),并且否则表 14 用于传送扩展信息。应理解,表 12-14 可以依据设计选择每当需要时被重传到解码器,而在一个实现方式中,仅仅在存在对该信息的改变时被重传。

[0281] 表 14-seq_parmeter_set_3dv_extension() 的第二实现方式

[0282]

seq_parameter_set_3dv_extension() {	C	描述符
seq_parameter_set_mvc_extension()		
for(I = 0 ; i <= num_views_minus1 ; i++) {		
video_layer_flag[i]		u(1)
depth_layer_flag[i]		u(1)
occlusion_layer_video_flag[i]		u(1)
occlusion_layer_depth_flag[i]		u(1)
transparency_layer_flag[i]		u(1)
}		
}		

[0283] 为了更好地说明在实施例 7 中可以怎样利用表 14 的 seq_parmeter_set_3dv_extension(), 参照分别图示编码 2400 和解码 2500 子集 SPS 3DV 的方法的图 24 和 25。应理解, 可以由编码器 300 来实现方法 2400 而可以由解码器 400 来实现方法 2500。

[0284] 方法 2400 可以开始于 2402, 其中编码器 300 可以编码 seq_parmeter_set_mvc_extension(), 其在更新的 AVC 草案中描述。编码器 300 然后可以执行循环 2404, 其中编码器 300 可以设置 3DV 层标志以指示对于特定的 3DV 视图 ‘i’ 是否存在相应的 3DV 层。例如, num_views_minus1 指示在 3DV 内容中采用的 3DV 视图的总数。例如, 在图 10-12 提供的示例中, 采用了 3 个 3DV 视图 (3DV 视图 0-3DV 视图 2)。为了便于编码和解码, num_views_minus1 的数字值比 3DV 视图的实际数量小 1。编码器 300 可以对于每个 3DV 视图 ‘i’ 迭代步骤 2406-2414, 直至到达了 3DV 内容中采用的 3DV 视图的总数。

[0285] 具体地, 在循环 2404 中, 编码器 300 可以在步骤 2406 设置并编码 2D 视频层标志以指示在 3DV 视图 ‘i’ 中是否存在 2D 视频层, 可以在步骤 2408 设置并编码 (2D) 深度层标志以指示在 3DV 视图 ‘i’ 中是否存在深度层, 可以在步骤 2410 设置并编码遮视频层标志以指示在 3DV 视图 ‘i’ 中是否存在遮挡视频层, 可以在步骤 2412 设置并编码遮挡深度层标志以指示在 3DV 视图 ‘i’ 中是否存在遮挡深度层, 并且可以在步骤 2414 设置并编码透明层标志以指示在 3DV 视图 ‘i’ 中是否存在透明层。

[0286] 现在转到使用表 14 解码子集 SPS 3DV 的方法 2500, 方法 2500 可以开始于 2502, 其中解码器 400 可以解码 seq_parmeter_set_mvc_extension(), 其在更新的 AVC 草案中描述。应注意, 方法 2500 中的解码器 400, 可以接收依据方法 2400 由编码器 300 编码的比特流。解码器 400 还可以执行循环 2504, 其中解码器 400 可以解码 3DV 层标志以确定对于特定的 3DV 视图 ‘i’ 是否存在相应的 3DV 层。例如, 如上关于方法 2400 讨论的, num_views_minus1 指示在接收的 3DV 内容中采用的 3DV 视图的总数。解码器 400 可以对于每个 3DV 视

图 ‘i’ 迭代步骤 2506-2514,直至到达了 3DV 内容中采用的 3DV 视图的总数。

[0287] 具体地,在循环 2504 中,解码器 400 可以在步骤 2506 解码并获得 2D 视频层标志以确定在 3DV 视图 ‘i’ 中是否存在 2D 视频层,可以在步骤 2508 解码并获得 (2D) 深度层标志以确定在 3DV 视图 ‘i’ 中是否存在深度层,可以在步骤 2510 解码并获得遮视频层标志以确定在 3DV 视图 ‘i’ 中是否存在遮挡视频层,可以在步骤 2512 解码并获得遮挡深度层标志以确定在 3DV 视图 ‘i’ 中是否存在遮挡深度层,并且可以在步骤 2514 解码并获得透明层标志以确定在 3DV 视图 ‘i’ 中是否存在透明层。

[0288] 如上讨论的,解码器 400 可以重构每个 3DV 视图中每个 3DV 层的 (多个) 参考画面列表由此允许解码器 400 在解码画面期间为接收的每个 3DV 层画面确定层间参考。

[0289] 附加实施例

[0290] 现在参照图 26 和 27,图示了编码和解码 3DV 内容的方法 2600 和 2700。应理解,可以以方法 2600 和 2700 来实现或者利用它们来实现关于实施例的在此讨论的任一个或者多个方面及其组合。例如,如以下在此进一步讨论的,可以以方法 2600 和 2700 并且通过它们来实现采取单个或者任何组合形式的实施例 1-3。此外,还应注意,图 3 的编码器 300 和图 4 的解码器 400 可以分别用于实现方法 2600 和 2700。

[0291] 方法 2600 可以开始于步骤 2602,其中编码器 300 可以编码多个画面,其中该多个画面描述了在给定时间处的给定视图的不同的 3D 信息。例如,以上关于编码器 300 讨论的任一个或者多个层编码器可以用于实现依据实施例 1,2 和 / 或 3 的任一个或更多来实现编码多个画面。该多个画面可以是例如 2D 视频层画面和深度层画面。由 2D 视频层画面描述的 3D 信息可以是例如 2D 视频。类似地,由深度层画面描述的 3D 信息可以是例如深度信息。2D 视频信息和深度信息均是给定时间处给定视图的 3D 信息的示例。

[0292] 为了描述附加实施例的方法的目的,“画面”可以等效于如上关于各种实施例讨论的“帧”。此外,画面可以对应于如上讨论的任一个或更多 3DV 层。例如,2D 视图 1010 和深度视图 1008 每一个可以构成单独的画面。另外,如上关于图 11 和 / 或图 12 讨论的任何 2D 视图层 1118、1122、1136、1218、1222、1236 和 / 或任何深度层 1120、1124、1220、1224 每一个可以构成单独的画面。另外,如上讨论的,未在图 11 和图 12 中显式图示的其它 3DV 补充层每一个也可以构成单独的画面。此外,如上讨论的任一个或者多个 3DV 视图可以构成给定时间处的给定视图,诸如如上关于图 11 和图 12 讨论的,在时间 T0 和 T1 处的 3DV 视图 0、1 和 2。

[0293] 在步骤 2604,编码器 300 可以生成语法元素,所述语法元素对于编码的多个画面指示怎样将编码画面装配到支持 3D 处理的结构中,所述结构定义该多个画面的内容类型。例如,3DV 参考缓冲器 316 可以依据任一个或更多实施例 1、2 和 / 或 3 生成语法元素。该语法元素例如可以是如上关于实施例 1 讨论的 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21。如上讨论的,根据实施例 1、2 和 3 的新颖的 NAL 单元可以为编码的 3DV 层指示怎样将每个层装配到支持 3D 处理的结构中,诸如图 10 的结构 1000 中。此外,新颖的 NAL 单元,诸如 NAL 单元 16 和 21 的使用,可以指示已经在比特流中使用了诸如图 10 中图示的 3DV 结构。如上注意的,结构 1000 可以定义不同内容类型,诸如不同类型的 3DV 层。应理解,结构可以对应于诸如图 10 所指示的一组 3DV 视图,和 / 或可以对应于 3DV 视图内的一组层。还应理解,编码器 300 可以使用不同编码的

画面作为参考来编码一画面,由此提供不同内容类型的画面之间的层间编码。例如,使用图 10 作为示例,视图 1004 的深度视图可以依赖于并且参考不同层,诸如视图 1004 的 2D 视图,由此提供层间编码。另外,图 10 的编码结构可以被配置为使得视图 1004 的 2D 视图可以依赖于并且参考不同的层,诸如视图 1006 的深度层。层间编码的其它类型是可能的,如上指示的,并且可以由本领域普通技术人员在考虑此处提供的教导的情况下实现。

[0294] 在步骤 2606,编码器 300 可以生成包括编码的多个画面和语法元素的比特流,其中包括的语法元素在编码的比特流级别上提供了对结构中编码的多个画面之间的关系指示。例如,3DV 参考缓冲器 316 可以生成比特流 318,其可以包括依据如上讨论的实施例 1、2 和 / 或 3 生成的任何编码的比特流。由此,该比特流可以包括多个编码的画面,诸如以上关于图 10-12 讨论的任何一个或更多层帧,并且还可以包括实施例 1 的任何一个或更多 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21,如上讨论的,这可以在编码比特流级别提供对结构中的编码的多个画面之间的关系指示。例如,语法元素可以指示在图 10 的结构或者支持 3DV 内容的其它结构中的画面或层之间的依赖性和关系。例如,语法元素可以提供应怎样组合画面来生成 3DV 内容的指示。

[0295] 应理解,依据各种实施例,编码器 300 的层编码器 304-314 的组可以被配置为执行步骤 2602。此外,3DV 参考缓冲器 316 和 / 或层编码器 304-314 可以被配置为执行步骤 2604-2606 中的一个或者多个步骤。编码器 300 可以替代地或者另外地包括被配置为至少执行方法 2600 的处理器。另外,实施例可以包括视频信号和 / 或视频信号结构,其被格式化为包括在步骤 2602 生成的多个编码画面,在步骤 2604 生成的语法元素,和 / 或在 2606 生成的比特流中包括的任何一个或更多元素,包括比特流自身。此外,实施例可以包括在其上存储了视频信号结构的处理器可读介质。另外,如上指示的,图 7 的调制器 722 可以被配置为调制视频信号。此外,实施例可以包括在其上存储了用于使得处理器至少执行方法 2600 的指令的处理器可读介质。

[0296] 再次参照用于解码 3DV 内容的图 27 的方法 2700,方法 2700 可以开始于步骤 2702,其中解码器 400 可以从比特流访问编码的多个画面。该多个画面描述了在给定时间处的给定视图的不同的 3D 信息。例如,比特流可以对应于依据方法 2600 生成的比特流。如上关于方法 2600 讨论的,以上关于图 11 和 / 或图 12 讨论的任何 2D 视图层和 / 或任何深度层每一个可以构成单独的画面。另外,如上讨论的,未在图 11 和图 12 中显式图示的其它 3DV 补充层每一个也可以构成单独的画面。此外,如上讨论的任一个或者多个 3DV 视图可以构成给定时间处的给定视图,诸如如上关于图 11 和图 12 讨论的,在时间 T0 和 T1 处的 3DV 视图 0、1 和 2。

[0297] 在步骤 2704,解码器 400 可以从比特流访问语法元素。所述语法元素对于编码的多个画面指示怎样将编码画面装配到支持 3DV 处理的结构中。所述结构提供了该多个画面之间的定义的关系。例如,3DV 参考缓冲器 414 可以依据任一个或更多实施例 1、2 和 / 或 3 访问语法元素。该语法元素例如可以是如上关于实施例 1 讨论的 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21。如上讨论的,根据实施例 1、2 和 3 的新颖的 NAL 单元可以为编码的 3DV 层指示怎样将每个层装配到支持 3D 处理的结构中,诸如图 10 的结构 1000 中。此外,新颖的 NAL 单元,诸如 NAL 单元 16 和 21

的使用,可以指示已经在比特流中使用了诸如图 10 中图示的 3DV 结构。如上注意的,结构 1000 可以定义不同内容类型,诸如不同类型的 3DV 层。应理解,结构可以对应于如图 10 所指示的一组 3DV 视图,和 / 或可以对应于 3DV 视图内的一组层。还应理解,解码器 400 可以使用不同编码的画面作为参考来解码一画面,由此允许不同内容类型的画面之间的层间解码。例如,使用图 10 作为示例,视图 1004 的深度视图可以依赖于并且参考不同层,诸如视图 1004 的 2D 视图,由此允许层间解码。另外,图 10 的编码结构可以被配置为使得视图 1004 的 2D 视图可以依赖于并且参考不同的层,诸如视图 1006 的深度层。层间编码的其它类型是可能的,如上指示的,并且可以由本领域普通技术人员在考虑此处提供的教导的情况下实现。此外,如上关于实施例 1-3 讨论的,实施例 1 的任何一个或更多 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21 可以通过使用 3DV 视图 ID 和 3DV 层 ID 来提供比特流的画面之间的定义的关系,如上讨论的。例如,解码器 400 可以被预先配置为依据诸如图 10 的结构 1000 之类的 3DV 结构来组合画面,并且可以使用 3DV 视图 ID 和 3DV 层 ID 来识别接收的那些画面对应于预定义结构中的不同层。

[0298] 在步骤 2706,解码器 400 可以被配置为解码所编码的多个画面。例如,解码器 400 可以使用如上例如关于图 4 和图 6 讨论的层解码器 402-412 来解码接收的画面。例如,解码器 400 可以使用由语法元素指示并且提供的定义的关系来呈现附加画面,所述附加画面参考一个或更多二维 (2D) 视频层画面、深度层画面、遮挡层画面,或者透明画面。例如,如上讨论的,图 10 的视图 1004 的深度视图可以依赖于并且参考不同层,诸如视图 1004 的 2D 视图,由此提供层间编码。由此,解码器 400 可以从一个或者多个各种不同层画面中呈现附加画面,诸如视图 1004 的深度视图。

[0299] 在步骤 2708,解码器可以以指示多个画面之间的定义关系的输出格式来提供解码的画面。例如,解码器 400 的 3DV 参考 / 输出缓冲器 414 可以输出依据 3DV 结构格式化的 3DV 内容。由此,该输出可以依据该结构向显示设备指示多个画面之间的关系以允许在显示设备上对 3DV 内容进行合适的显示并且使得用户能够观看 3DV 内容。具体地,该输出格式可以包括用于指定怎样将解码的画面装配到结构中的语法元素。这样的语法元素的示例可以包括实施例 1 的任何一个或更多 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21。

[0300] 在执行了步骤 2708 之后,可以在解码器处执行可选的步骤 2710-2714。实现方式可以执行作为步骤 2708 的一部分和 / 或作为步骤 2706 的解码的一部分的一个或更多步骤 2710-2714。在各种实现方式中,可以在显示器处执行一个或更多步骤 2710-2714,特别是步骤 2714。

[0301] 可选地,在步骤 2710,解码器 400 可以使用语法元素从多个画面中识别 2D 视频画面。例如,通过解析用于编码 3DV 层所实现的实施例 1 的任何一个或更多 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21,解码器 400 可以识别 2D 视频画面或者层。解码器 400 可以进一步确定编码的画面中的哪些具有在以上被表示为 '0' 的 2D 视图层 ID,并且使用 3DV 视图 ID 来确定对应的 3DV 视图。

[0302] 可选地,在步骤 2712,解码器 400 可以使用语法元素从多个画面中识别深度画面。例如,通过解析用于编码 3DV 层所实现的实施例 1 的任何一个或更多 3DV 前缀单元 16,实施例 2 的 NAL 前缀单元 14 和 / 或 NAL 单元 20,和 / 或实施例 3 的 NAL 单元 21,解码器 400 可

以识别深度画面或者层。另外,解码器 400 可以确定编码的画面中的哪些具有在以上被表示为‘1’的深度层 ID,并且使用 3DV 视图 ID 来确定对应的 3DV 视图。应注意,依据各种实施例,可以使用语法元素来识别其它 3DV 补充层。

[0303] 可选地,在步骤 2714,解码器 400 可以基于 2D 视频画面和深度画面呈现附加视图的新的画面。例如,识别的画面或者视图可以对应于图 10 的 2D 视图 1010 和深度视图 1008。另外,依据以上关于图 10 提供的描述,3D 视图 1004 和 1006 例如可以通过使用 3DV 基本视图 1002 的 2D 视图 1010 和深度视图 1008 作为参考来呈现。类似地,依据以上关于图 10 提供的描述,3DV 视图 1006 的 2D 视频层和深度层可以用作呈现 3DV 视图 1004 的参考。

[0304] 应理解,依据各种实施例,解码器 400 的层解码器 402-412 的组可以被配置为执行步骤 2702 和 2706。此外,3DV 参考缓冲器 414 和 / 或层解码器 402-412 可以被配置为执行步骤 2704 和 2708 中的一个或者多个步骤。解码器 400 可以替代地或者另外地包括被配置为至少执行方法 2700 的处理器。另外,如上指示的,图 8 的解调器 822 可以被配置为解调视频信号,所述视频信号包括在步骤 2702 中从其中访问多个编码画面的比特流。此外,实施例可以包括在其上存储了用于使得处理器至少执行方法 2700 的指令的处理器可读介质。

[0305] 现在参照图 28,图示了构建参考画面列表的方法 2800。应理解,可以在方法 2800 中实现或者以方法 2800 来实现关于实施例的在此讨论的任何一个或者多个方面以及其组合。例如,如以下在此进一步讨论的,可以在方法 2800 中或者以方法 2800 来实现实施例 4。另外,实施例 1-3 以及 5-7 中的任何一个或者多个可以与实施例 4 进行组合并且在方法 2800 中或者以方法 2800 来实现。此外,还应注意,图 3 的编码器 300 和 / 或图 4 的解码器 400 可以用于实现方法 2800。另外,尽管方法 2800 描述了构建用于画面的参考画面列表,但这种参考列表可以被构建用于画面的序列,用于在多个视图上的一组画面或者用于画面的子集,如以上关于实施例 4 讨论的。

[0306] 方法 2800 可以开始于可选步骤 2802,其中编码器 300 或者解码器 400 可以基于用于画面的依赖性信息确定用于画面的层间参考。例如,如以上讨论的,解码器 400 可以从接收的用于传递序列参数集 (SPS) 的语法元素中提取和解码该依赖性信息。继而,对于编码器 300,该依赖性信息可以与在 SPS 中包括的编码器 300 的依赖性信息相同,如上例如关于实施例 5-8 讨论的。例如,编码器 300 可以从在编码器上存储的配置文件中获得依赖性信息。应理解,该依赖性信息可以包括指示怎样预测性地编码不同画面和画面类型的时间依赖性、视图间依赖性和层间依赖性中的任何一个或更多。由此,基于该依赖性信息,编码器 300 或解码器 400 可以确定对其正在构建参考画面列表的画面的层间参考。另外,该层间参考可以符合以上关于实施例 4 讨论的层间参考。例如,该层间参考可以对应于以上关于图 15 讨论的结构中的任何一个或更多。

[0307] 在步骤 2804,编码器 300 或解码器 400 可以确定层间参考相对于画面的一个或者多个其它参考的优先级。例如,编码器 300 或解码器 400 可以被配置为应用优先级方案而排列参考画面列表中的画面的优先级。例如,如以上关于实施例 4 讨论的,参考列表中的画面可以依据对其构建参考画面列表的画面关于在其参考画面列表中列出的画面具有的冗余性程度进行排序 / 排列优先级。例如,如以上关于深度画面讨论的,期望层间参考与参考列表中的时间参考和视图间参考相比具有最低冗余性。因此,层间参考具有比时间和视图间参考更低的冗余。应注意以上关于实施例 4 中的不同 3DV 层类型提供的任何优先级此处可

以在步骤 2804 中应用。然而,还应理解,依据在此描述的各个方面还可以采用不同的优先级。例如,优先级可以依据画面参考和与用于 3DV 内容的参考画面列表相关联的画面之间的实际冗余性而变化。例如,冗余性可以基于组成 3DV 内容的画面或层的测量来确定并且优先级方案可以被调整为反映测量的冗余性等级使得具有与参考列表相关联的画面有较高冗余性的参考画面比具有较低冗余性的参考画面被给予较高的优先级。此外,在其它方面或者实施例中,可以将这种优先级方案设计为针对每个画面或者参考画面列表而不同。

[0308] 在步骤 2806,编码器或者解码器 400 可以在基于优先级的排序的画面的参考列表中包括层间参考。例如,可以在具有较高优先级的其它参考画面之后或者在列表的结尾处包括具有较低或者最低优先级的层间参考画面。反过来,可以在具有较低优先级的其它参考画面之前或者在列表的开始处包括具有较高或者最高优先级的层间参考画面。这种参考可以包括如上讨论的时间和 / 或视图间参考。如上指示的,可以依据用于编码器实现方式的方法 1600 或者用于解码器实现方式的方法 1700 在参考的列表中包括层间参考。此外,如以上关于步骤 2804 讨论的,可以依据其它优先级方案在参考列表中包括层间参考。应注意,可以基于期望的用途来对列表进行排序以及排列优先级使得较小的索引可以用于更常用的参考并且可以在传输时节约比特。

[0309] 在可选步骤 2808,编码器或者解码器 400 可以在涉及画面的编解码操作中使用层间参考。例如,编码器 300 可以执行预测编解码操作以使用层间参考作为参考画面而编码对其构建参考列表的画面。反过来,解码器 400 可以执行预测解码操作以使用层间参考作为参考画面来解码对其构建了参考列表的画面。因此,可以至少部分地基于层间参考来编码或解码画面。

[0310] 可选地,在步骤 2810,编码器 300 或解码器 400 可以生成包括编解码画面的比特流。例如,编码器 300 可以依据以上关于图 3 和图 5 提供的讨论在比特流 318 中包括编码的画面。另外,解码器 400 可以依据以上关于图 4 和图 6 提供的讨论在比特流 416 中包括解码的画面。

[0311] 此后,该方法可以结束或者可以重复使得编码器 300 或者解码器 400 可以为另一画面生成参考画面列表或者可以对相同的画面生成第二参考画面列表,如果该画面是 B 画面。

[0312] 一个实现方式仅仅执行步骤 2804 和 2806。可以提供层间参考,例如,该实现方式确定该层间参考的优先级。该实现方式然后基于所确定的优先级在排序的列表值包括该层间参考。

[0313] 转到步骤 2802,可选地,步骤 2802 可以包括在图 29 中提供的用于处理 2D 视频层画面的方法 2900 性能。例如,方法 2900 可以开始于步骤 2902,其中,编码器 300 或解码器 400 可以确定对其构建参考画面列表的画面是否是 2D 视频层画面。如果该参考不是 2D 视频层画面,则该方法可以前进到方法 2800 的步骤 2804。否则,该方法可以前进到步骤 2904,其中编码器 300 或解码器 400 可以从参考画面列表中排除任何层间参考。例如,如以上关于实施例 4 讨论的,限制使用 2D 视频层的层间参考可以使得常规的 MVC 提取 3DV 内容并且格式化该内容用于显示。其后,该方法可以前进到方法 2800 的步骤 2804。

[0314] 还可以修改步骤 2904 以从正在用作 2D 视频层的参考中仅仅排除深度层。这种实现方式可以例如依靠作为 2D 视频层的层间参考的遮挡视频层。

[0315] 应理解,依据各种实施例,可以配置一组层间编解码器,诸如解码器 400 的层间解码器 402-412 或者编码器 300 的层间编码器 304-314,来执行步骤 2808 和步骤 2810。此外,可以配置 3DV 参考缓冲器 414、3DV 参考缓冲器 316,和 / 或层间编解码器来执行步骤 2802-2806 以及 2810 中的任一个或者多个。编码器 300 或者解码器 400 可以替代地或者另外地包括被配置为至少执行方法 2800 的处理器。此外,实施例可以包括其上存储了用于促使处理器至少执行方法 2800 的指令的处理器可读介质。

[0316] 现在参照图 30 和 31,图示了用于编码和解码 3DV 内容使得传递 3DV 层间依赖性结构的方法 3000 和 3100。应理解,可以在方法 3000 和 3100 中实现或者以方法 3000 和 3100 来实现关于各种实施例的在此讨论的任何一个或者多个方面以及其组合。例如,如以下在此进一步讨论的,可以在方法 2600 和 2700 中或者以方法 2600 和 2700 来实现实施例 5-7。另外,还应注意,图 3 的编码器 300 和图 4 的解码器 400 可以用于分别实现方法 3000 和 3100。

[0317] 方法 3000 可以开始于步骤 3002,其中编码器 300 生成指示 3DV 层之间的层间依赖性结构的语法元素。例如,可以如上关于实施例 5-7 中的任何一个或更多讨论的来生成语法元素。例如,可以采用 NAL 单元 17 作为语法元素来传递互相依赖性结构,如以上关于实施例 5 讨论的。此外,可以如以上关于实施例 6 和 7 以及关于表 13 和 14 所讨论的来传递互相依赖性结构。例如,可以采用方法 2000、2200 和 2400 中的任何一个或更多来传递互相依赖性结构。例如,语法元素可以显式地传递层间依赖性结构,如上关于实施例 6 讨论的,或者语法元素可以使用 3DV 层 id 通过传递对每个 3DV 视图是否存在特定的 3DV 层来指示层间依赖性结构,其中层间依赖性预先定义的,如以上关于实施例 7 所讨论的。另外,层间依赖性结构可以对应于许多不同层间依赖性结构中的一个。例如,层间依赖性结构可以对应于以上关于图 15 描述的以及以上关于实施例 7 讨论的那些。此外,如上述的,可以在 NAL 单元首标、SPS、PPS、SEI 或者码片首标中的任何一个或更多中提供层间依赖性结构。此外,编码器 300 可以通过构建并采用参考画面列表来生成语法元素,例如以上关于实施例 4 所讨论的。

[0318] 在步骤 3004,编码器 300 可以基于层间依赖性结构从 3D 层的层中识别层间参考。例如,如果层间依赖性结构对应于以上关于图 15 描述的结构,为编码深度层画面,编码器 300 可以采用可以在步骤 3002 中构建的参考画面列表来确定用于深度层画面的层间参考是与该深度层画面相同的视图或者 3DV 视图中的 2D 视频层画面。如上注意的,互相依赖性结构可以变化并且可以包括许多不同类型的层,诸如 2D 视频层,深度层,遮挡视频层,遮挡深度层和透明层,以及其它具有不同互相依赖性的层,包括例如,不同 3DV 视图之间的互相依赖性。

[0319] 在步骤 3006,编码器 300 可以至少部分地基于层间参考来编码画面。例如,编码器 300 可以使用编码器 304-314 如以上关于图 3 和图 5 讨论的来编码画面。此处,再次使用结构 1500 和深度层作为示例,深度层可以至少部分地基于 2D 视频层来编码,如上所讨论的。

[0320] 在可选步骤 3008,编码器 300 可以生成包括编码的画面的比特流。例如,编码的比特流可以如以上关于图 3 和图 5 讨论的而生成并且可以对应于例如比特流 318。

[0321] 在可选步骤 3010,编码器 300 可以提供编码的画面和用于在解码所编码的画面中使用的语法元素。例如,语法元素和编码的画面可以经由比特流 318 传输给解码器 400。可

替换地,语法元素可以在与用于传输 3DV 数据内容的比特流分开的比特流中进行传输。由此,图 3 中的比特流 318 可以表示两个分离的对应比特流。可替换地,可以单独地传输不同的比特流。例如,一个比特流可以在经由电缆网络被传输给解码器 400,而另一个比特流可以无线地被传输给解码器 400。另外,语法元素可以用于解码所编码的画面,如以下关于方法 3100 在此讨论的。

[0322] 应理解,依据各种实施例,编码器 300 的层编码器 304-314 的集合可以被配置为执行步骤 3006。此外,3DV 参考缓冲器 316 和 / 或层编码器 304-314 可以被配置为执行步骤 3002、3004、3008 和 3010 中的一个或更多。编码器 300 可以替换地或另外包括被配置为执行至少方法 3000 的处理器。另外,实施例可以包括被格式化为包括依据方法 3000 生成的编码画面、语法元素和 / 或比特流的视频信号和 / 或视频信号结构。此外,实施例可以包括其上存储了视频信号结构的处理器可读介质。另外,如上指示的,图 7 的调制器 722 可以被配置为调制视频信号。此外,实施例可以包括在其上存储了用于使得处理器至少执行方法 3000 的指令的处理器可读介质。

[0323] 一个实现方式仅仅执行步骤 3002-3006。该实现方式生成用于识别画面的层间参考的语法元素并且然后至少部分地基于所识别的层间参考来编码画面。在该情形下,该实现方式不需要生成包括编码的画面的比特流或者提供编码画面和用于解码的语法元素。

[0324] 再次参照用于解码 3DV 内容的图 31 的方法 3100,方法 3100 可以开始于步骤 3102。解码器 400 可以从比特流中访问编码的画面,其中该画面描述在给定时间处来自给定视图的特定的 3DV 层的 3DV 信息。例如,编码画面可以对应于以上讨论的任何一个或更多 3DV 层。例如,2D 视图 1010 和深度视图 1008 每一个可以构成单独的画面。另外,如上关于图 11 和 / 或图 12 讨论的任何 2D 视图层 1118、1122、1136、1218、1222、1236 和 / 或任何深度层 1120、1124、1220、1224 每一个可以构成单独的画面。另外,如上讨论的,未在图 11 和图 12 中显式图示的其它 3DV 补充层每一个也可以构成单独的画面。此外,如上讨论的任一个或者多个 3DV 视图可以构成给定时间处的给定视图,诸如如上关于图 11 和图 12 讨论的,在时间 T0 和 T1 处的 3DV 视图 0、1 和 2。此外,编码画面可以是由方法 3000 生成的编码画面。

[0325] 在步骤 3104,解码器 400 可以访问为一组包括特定 3DV 层的 3DV 层指示层间依赖性结构的语法元素。例如,NAL 单元 17 可以是指示互相依赖性结构的语法元素,如上关于实施例 5 所讨论的。此外,可以如以上关于实施例 6 和 7 以及关于表 13 和 14 所讨论的来指示或传递互相依赖性结构。例如,可以采用方法 2000、2200 和 2400 中的任何一个或更多来传递或指示互相依赖性结构。

[0326] 例如,语法元素可以显式地传递层间依赖性结构,如上关于实施例 6 讨论的。或者语法元素可以使用 3DV 层 id 通过传递对每个 3DV 视图是否存在特定的 3DV 层来指示层间依赖性结构,其中层间依赖性预先定义的,如以上关于实施例 7 所讨论的。另外,层间依赖性结构可以对应于许多不同层间依赖性结构中的一个。例如,层间依赖性结构可以对应于以上关于图 15 描述的以及以上关于实施例 7 讨论的那些。此外,如上述的,可以从 NAL 单元首标、SPS、PPS、SEI 或者码片首标中的任何一个或更多中获得层间依赖性结构和语法元素。此外,解码器可以访问语法元素,例如如以上关于方法 2100、2300 和 2500 中的任何一个或者多个所讨论的。

[0327] 在步骤 3106,解码器 400 可以至少部分地基于层间依赖性结构来解码所编码的画

面。例如,解码器 400 可以如以上关于图 4 和图 6 所讨论的来解码所编码的画面。此外,解码器 400 可以使用语法元素来构建并采用一个或者参考画面列表以便解码所编码的画面,例如如以上关于实施例 4 所讨论的。由此,解码器 400 可以为预测编码目的确定编码的画面的参考并且可以至少部分地基于其参考来解码画面。

[0328] 在可选步骤 3108,解码器 400 可以以指示层间依赖性结构的输出格式来提供解码的画面。例如,解码器 400 的 3DV 参考 / 输出缓冲器 414 可以输出依据层间依赖性结构格式化的 3DV 内容。由此,该输出可以依据该结构向显示装置指示多个画面之间的关系以允许在显示装置上对 3DV 内容进行合适的显示并且使得用户能够观看 3DV 内容。具体地,该输出格式可以包括用于指定怎样将解码的画面装入结构中的语法元素。这样的语法元素的示例可以包括如上讨论的 NAL 单元 17。

[0329] 应理解,依据各种实施例,解码器 400 的层解码器 402-412 的集合可以被配置为执行步骤 3106。此外,3DV 参考缓冲器 414 和 / 或层解码器 402-412 可以被配置为执行步骤 3102、3104、3108 中的一个或更多。解码器 400 可以替换地或另外包括被配置为执行至少方法 3100 的处理器。另外,如上所指示的,图 8 的解调器 822 可以被配置为解调在步骤 3102 中从中访问多个编码画面的比特流的视频信号。此外,实施例可以包括在其上存储了用于使得处理器至少执行方法 3100 的指令的处理器可读介质。

[0330] 应理解,以上讨论的实施例可以通过本领域技术人员在考虑到此处提供的教导的情况下以各种方式进行组合。例如,现在参照图 32,图示了合并来自以上讨论的若干实施例的特征的 NAL 单元流 3200。此处,流 3200 可以包括用于 MVC 的子集序列参数集的 NAL 单元 15(3202),如以上在表 3 中提供并且在 AVC 草案中定义的。另外,流 3200 可以进一步包括指示至少一个层间依赖性结构的、用于 3DV 的扩展的 SPS 的 NAL 单元 17,如以上关于实施例 5-7 讨论的。此处,为了简明的目的,在流 3200 中采用图 10 中示出的层间依赖性结构。

[0331] 类似于图 11 和图 12,图 32 提供了分别与时间 T0 和时间 T1 对应的 3DV 视图的组。以上讨论的图 11 和图 12 的截断也应用于图 32 并且图 32 的箭头指示 NAL 单元的传输顺序,类似于图 11 和 12 的箭头。当然,图 32 是流 3200 的小的节选。在实际应用中,流 3200 可能包括用于众多不同时间实例的许多 NAL 单元。另外,三个 3DV 视图的使用是一示例并且在解码器处可以采用和 / 或呈现许多视图,如熟悉 MVC 的本领域技术人员所理解的。此外,每个视图两个 3DV 层的使用也是一示例并且应理解可以采用若干另外的 3DV 层,如以上详细讨论的。

[0332] 在流 3200 的节选中,三个 3DV 视图 3206、3208 和 3210 对应于时间 T0,而三个 3DV 视图 3212、3214 和 3216 对应于时间 T1。类似于图 11 和图 12,3DV 视图 0(3206,3212) 可以对应于图 10 中的基本视图 1002,而 3DV 视图 2(3208,3214) 以及 3DV 视图 1(3210,3216) 可以分别对应于图 10 的 P 视图 1006 和 B 视图 1004。3DV 视图 3206 可以包括组成 2D 视频层 3218 的 NAL 单元 16(3220)、14(3222) 和 5(3224)。如上讨论的,NAL 单元 5 包括瞬时解码刷新 (IDR) 画面的编解码码片的视频数据并且仅由帧内码片或 SI 码片组成,如 AVC 草案中定义的。另外,NAL 单元 14 可以包括用于表示作为依据 MVC 的其它视图的基本层的 2D 视频层 3218 的参考作为 MVC 前缀。在另一实现方式中,其中使用立体简档,可以省略 NAL 单元 14 和 17。

[0333] NAL 单元 16 例如可以包括 3DV 视图 ID 和 3DV 层 ID,如上关于实施例 1 讨论的。此

处,解码器 400 例如可以使用 3DV 视图 ID 和 3DV 层 ID 来识别用于其它深度层或者用于其它 3DV 层的层间参考的 2D 视频层 3218。如在图 32 中示出的,3DV 视图 3206 还包括由 NAL 单元 21 (3228) 组成的深度层 3226,如以上关于实施例 3 描述的。如以上关于实施例 3 讨论的,除了在 MVC NAL 单元首标扩展中提供的其它信息以外, NAL 单元 21 还可以包括 3DV 视图 ID 和 3DV 层 ID。

[0334] 如以上关于实施例 4-7 所讨论的,解码器 400 可以使用在 SPS 中提供的信息,诸如由 NAL 单元 17 提供的层间依赖性结构,来重构参考画面列表,并且使用该参考画面列表来合适地解码 3DV 内容。例如,基于 3DV 视图 ID 和 3DV 层 ID,解码器 400 可以在 SPS 中传递的层间依赖性结构中确定对应的层(在该情形下,深度层 3226)的作用。此处,3DV 视图 ID 和 3DV 层 ID 可以指示 2D 视频层 3218 可以被用作解码深度层 3226 的参考。

[0335] 也如在图 32 中示出的,时间 T0 的其它每个 3DV 视图由分别对应于 3DV 视图中的 2D 视频层和深度层的 NAL 单元 20 和 21 组成。视图 3208 和 3210 内的 NAL 单元可以具有与视图 1206 和 1208 中的 NAL 单元的功能相同的功能,如以上关于图 12 所讨论的。类似地,时间 T1 的 3DV 视图的集合在本质上是与时间 T0 的 3DV 视图集合相同的方式构建的,除了利用 3DV 视图 3212 中的 NAL 单元 1 取代了 3DV 视图 3206 中的 NAL 单元 5。如以上关于实施例 3 讨论的, NAL 单元 1 包括非 IDR 画面的编解码码片的视频数据。

[0336] 现在参照图 33,图示了提供采用层间依赖性结构管理网络通信量的系统 3300。系统 3300 可以包括以上关于图 7 和图 8 描述的传输系统/装置 700 和接收系统/装置 800。具体地,传输系统/装置 700 的编码器 710 可以由以上关于此处描述的各种实现方式讨论的编码器 300 来实现。类似地,传输系统/装置 800 的编码器/解码器 820 可以由以上关于此处描述的各种实现方式讨论的解码器 400 来实现。在图 33 中,系统 3300 的输入和输出被列为“(多个)输入视频”和“输出视频”。应清楚,至少在该实现方式中,这些指代包括多个层的 3D 视频。

[0337] 系统 3300 还可以包括在传输系统/装置 700 和接收系统/装置 800 之间的网络 3350 中提供的网络设备/系统 3301。网络 3350 可以是例如有线的网络,诸如因特网、广域网或者局域网 (LAN),或者无线网络,诸如无线蜂窝网络或者无线 LAN。继而,网络设备 3301 可以被实现为有线网络中的路由器或者无线网络中的基站。如图 33 中图示的,网络设备 3301 可以包括解析器 3302、控制器 3304、网络通信量监视器 3306 和转发模块 3308。另外,网络设备 3301 的每个元件可以被实现为硬件元件或者软件和硬件的组合。网络设备 3301 和其元件的功能在以下关于可以由网络设备 3301 实现的图 34 的方法 3400 而更详细地描述。

[0338] 继续参照图 33,参照图 34,提供了管理网络资源的方法 3400。方法 3400 可以开始于步骤 3402,其中解析器 3302 可以解析接收的用于指示 3DV 层的层间依赖性结构的语法元素,以基于该结构对于 3DV 层的至少一子集确定转发优先级。例如,语法元素可以是在从传输系统/装置 700 接收的 NAL 单元 17 中,其可以指示依据例如表 13 或 14,如以上关于实施例 5-7 讨论的层间依赖性结构。

[0339] 此处,解析器 3302 可以依据如在层间依赖性结构中指示的 3DV 层的重要性来确定转发优先级。例如,3DV 层 id 可以被配置为使得最小的编号对应于最高优先级,而最大的编号对应于最低优先级。如果采用图 15 的互相依赖性结构和表 1 的 3DV 层标识符,解析器

3302 可以基于 3DV 层标识符确定 2D 视频层具有最高优先级,深度层具有下一最高优先级,等等。具体地,可以依据其提供 3DV 内容的贡献的重要性来排序 3DV 层标识符。例如,参照图 2,2D 视频的深度层 204 可以被考虑得比遮挡视频层 206 或者遮挡深度层 208 更重要,这是因为其对视图中的主要对象提供三维效果,而遮挡视频层 206 或遮挡深度层 208 不能这样。可以应用确定不同 3DV 层之间的重要性的变型。

[0340] 例如,一个变型可以是依据层在层间依赖性结构中具有的参考的数目或者参考画面列表中可以包括的特定层的数目以层的优先级为基础。例如,响应于确定层间依赖性结构和 / 或对应的层间参考,解析器 3302 可以向最多数层参考的层分配最高优先级,而向最少数层参考的层分配最低优先级。例如,在图 15 的层间依赖性结构中,2D 视频层可能具有最高优先级,这是由于它被三个层参考,而被一个层参考的深度层将具有下一最高优先级,等等。

[0341] 其它变型可以针对依据用于合适地编码 / 解码层所采用的参考的数量而对 3DV 层进行排序。例如,在以上关于实施例 7 所描述的互相依赖性结构中,可以向遮挡视频层、遮挡深度层,以及透明层给予最低优先级,这是因为它们每一个采用两个(层间)参考,可以向深度层给予下一更高优先级,这是因为它采用一个(层间)参考,并且可以向 2D 视频层给予最高优先级,这是因为它不采用任何(层间)参考。

[0342] 此外,可以应用不同的组合来确定优先级。例如,可以使用考虑到在呈现 3D 视图时给定的层的重要性和参考该给定层的层的数量二者的加权函数来确定转发优先级。另外,应理解,可以考虑除了层间参考之外的其它类型的参考。例如,上述的优先级确定可以进一步考虑特定层依赖的时间参考和视图间参考。由此,可以将上述的推理应用到任何类型的参考和 / 或参考的组合,诸如时间参考和视图间参考,和 / 或层间参考。

[0343] 在步骤 3404,解析器 3302 可以接收用于构建 3DV 层的数据单元。例如,再次参照图 11,解析器 3302 可以接收所采用的用于构建 2D 视频层 1118 的 NAL 单元 16、14 和 5。解析器 3302 还可以接收用于构建深度层 1120 的 NAL 单元 16 和 20,等等。

[0344] 在步骤 3406,网络通信量监视器 3306 可以测量网络 3350 上的通信量 / 网络拥塞。如本领域技术人员理解的,此处可以采用各种已知的网络通信量监视器。

[0345] 在步骤 3408,基于从网络通信量监视器 3306 接收的拥塞测量,控制器 3304 可以确定在步骤 3406 测量的网络通信量是否满足第一拥塞阈值。应注意,此处,可选地,可以依据所确定的转发优先级,采用多种不同的拥塞阈值并且将其与 3DV 层相关联,如上述的,所述转发优先级可以基于层间依赖性结构。例如,可以对呈现 3DV 内容所采用的每一个 3DV 层或者对每个可丢弃的 3DV 层使用一个拥塞阈值。例如,再次参照表 1,如果如以上关于步骤 3402 所讨论的,依据 3DV 层 ID 编号来确定转发优先级,则第一阈值可以与透明层相关联,对应于比第一阈值高级别的网络拥塞的第二阈值可以与遮挡深度层相关联,对应于比第一和第二阈值更高级别的网络拥塞的第三阈值可以与遮挡视频层相关联,等等。

[0346] 由此,如果满足第一拥塞阈值,则在步骤 3412,控制器 3304 可以丢弃在步骤 3404 接收的用于具有最低优先级的 3DV 层的单元或者 NAL 单元,以及可以引导转发模块 3308 将用于剩余 3DV 层的单元(如果不满足下一阈值)转发到接收系统装置 800。如果不满足第一拥塞阈值,则转发模块 3308 在控制器 3304 的引导下,可以在步骤 3410 转发用于所有 3DV 层的单元。应理解,可以对 N 个 3DV 层的每一个重复阈值确定。例如 N 个层可以对应于一

个或更多视图内用于呈现 3DV 内容采用的层的数量。同样,可以对每个阈值重复阈值确定并且可以依赖于结果进行单元丢弃和转发决定。

[0347] 例如,如果在步骤 3412 之后,不满足第二阈值,则可以通过转发模块 3308 在步骤 3412 将用于 N-1 个 3DV 层的单元转发给接收单元 800。可替换地,如果在步骤 3412,控制器 3304 确定满足第一 N-2 个阈值,则该方法可以前进到步骤 3414,其中控制器 3304 可以确定是否满足第 (N-1) 个拥塞阈值。如果不满足第 (N-1) 个拥塞阈值,则转发模块 3308 在控制器 3304 的引导下,可以在步骤 3416 转发用于具有最高两个优先级的 3DV 层的单元。另外,在步骤 3416,控制器 3304 可以丢弃 N-2 最低优先级 3DV 层,这是由于 N-2 最低优先级 3DV 层已经满足了阈值。如果满足第 (N-1) 个拥塞阈值,则转发模块 3308 在控制器 3304 的引导下可以在步骤 3418 转发具有最高优先级的 3DV 层的单元。另外,控制器 3304 可以丢弃 (N-1) 最低优先级 3DV 层的单元。相应地,方法 3400 可以前进通过阈值确定,使得当满足第 M 个阈值并且不满足第 (M+1) 个阈值时,例如用于 M 最低优先级 3DV 层的 NAL 单元被丢弃并且剩余的较高优先级层的单元被转发。应注意,在该示例,仅仅考虑 N-1 个阈值以确保至少最高优先级层不被丢弃来确保接收装置 / 系统 800 可以至少解码一些内容。然而,可以采用方法 3400 的变型。还应注意,可以周期性地重复方法 3400 的一个或更多步骤以考虑网络拥塞的任何改变。

[0348] 应清楚,除了方法 3400 以外,其它实现方式是可行的。一种这样的实现方式是更通用的并且包括访问指示三维视频 (3DV) 层之间的层间依赖性结构的语法元素。如在步骤 3402 中所示的,可以例如通过解析接收的语法元素来执行该访问。

[0349] 实现方式还基于结构来确定 3DV 层的特定 3DV 层的传输优先级。传输优先级可以例如是与转发来自流的画面 (或者画面的一部分) 或者丢弃画面 (或者画面的一部分) 有关的优先级。可以通过确定有多少层使用该特定的 3DV 层作为参考 (层间参考、视图间参考,和 / 或时间参考) 来确定传输优先级。

[0350] 该实现方式还确定是否传输属于该特定 3DV 层的编码数据。确定是否进行传输是基于所确定的特定 3DV 层的传输优先级以及基于网络拥塞的指示。例如,如在步骤 3406 中确定网络拥塞。网络拥塞的指示可以包括例如,指示是否已经满足一个或更多拥塞阈值的标志 (或者一组标志),如在步骤 3408 和 3414。其它指示符可以包括例如网络活动的量度 (吞吐量率、误码率、重传请求的数量或者速率、应答的数量或者速率,等等)。

[0351] 另一实现方式访问这种传输优先级,并且基于所访问的用于特定 3DV 层的传输优先级和基于网络拥塞的指示来确定是否传输属于特定 3DV 层的编码数据。然而,该实现方式不需要访问指示 3DV 层的层间依赖性结构的语法。该实现方式也不需要基于层间依赖性结构来确定传输优先级。

[0352] 还应清楚,传输优先级可以整个地或者部分地基于其它信息。这种信息可以包括例如与 AVC、MVC 或者 SVC 系统有关的时间级 ID、优先级 ID,或者视图 ID。

[0353] 因此通过一个或更多具有特定特征和方面的实现方式。然而,所描述的实现方式的特征和方面还可以适配于其它实现方式。

[0354] 在本申请中描述的若干实现方式和特征可以用于 H. 264/MPEG-4AVC (AVC) 标准,或者具有 MVC 扩展的 AVC 标准,或者具有 SVC 扩展的 AVC 标准的背景下。另外,实现方式可以用于来自以下的编码标准或者编码建议的背景中:(a)MPEG 和 ITU-T 的视频编码联合

协作组 (JCT-VC), (b) 来自 MPEG 的高性能视频编码组, (c) 来自 ITU-T 的视频编码专家组 (VCEG) 的下一代视频编码组, (d) 来自 MPEG 的 3D 视频编码组, (e) 与 MPEG 或 ITU-T 中的一个或更多相关联的任何其它组, 或者 (f) 由公司开发的标准 (专有的或者公用的)。然而, 这些实现方式和特征可以用于另一标准 (现有的或者将来的) 的背景下, 或者不涉及标准的背景下。

[0355] 此外, 实现方式可以使用各种技术来发信号通知信息, 所述各种技术包括但不限于, SEI 消息、码片首标、其它高级语法、非高级语法、带外信息、数据流数据, 和隐式的信令。相应地, 尽管可以在特定的背景下描述此处所描述的实现方式, 但这种描述绝不应被作为将所述特征和构想限制于这样的实现方式或者背景。

[0356] 另外, 可以在以下中的一个或者多个中实现许多实现方式: 编码器、解码器、处理来自解码器的输出的后处理器, 或者向编码器提供输入的预处理器。此外, 通过本公开可设想其它实现方式。

[0357] 说明书中对于“一个实施例”或者“一实施例”或“一个实现方式”或“一实现方式”以及其它变型的引用意味着连同实施例描述的特定的特性、结构、特征, 等等被包括在至少一个实施例中。由此, 贯穿本说明书的各个位置中出现的在“一个实施例中”或在“一实施例”或者在“一实现方式”或在“实现方式”以及任何其它变型中出现的短语不是必须都指代相同的实施例。

[0358] 要理解的是以下“/”, “和 / 或”, 以及“至少一个”中任一个的使用, 例如, 在“A/B”, “A 和 / 或 B”以及“A 和 B 中的至少一个”的情形下, 旨在包括仅选择列出的第一选项 (A)、仅选择列出的第二选项 (B), 或者选择两个选项 (A 和 B)。作为另一个例子, 在“A、B、和 / 或 C”和“A、B 和 C 中的至少一”以及“A、B 或 C 中的至少一个”的情形下, 这样的短语旨在包括仅选择列出的第一选项 (A)、仅选择列出的第二选项 (B)、或者仅选择列出的第三选项 (C)、或仅选择列出的第一选项和第二选项 (A 和 B)、或仅选择列出的第一选项和第三选项 (A 和 C)、或仅列出的第二选项和第三选项 (B 和 C), 或者选择所有的三个选项 (A 和 B 和 C)。如本领域和相关领域的普通技术人员易于清楚的, 这可以被扩展用于所列出的许多条目。

[0359] 另外, 如在此使用的, 单词“画面”和“图像”被可互换地使用, 并且指代例如来自静态图像的全部或者一部分 (局部) 或者来自视频序列的画面的全部或者一部分 (局部)。更一般地, 画面例如指代画面或者视频数据的任何集合。画面可以是例如像素、宏块、码片、帧、场、全画面、界定画面中的对象的区域、画面的前景、画面的背景, 或者画面中特定的 (x, y) 坐标组。类似地, 画面的“局部”可以是例如像素、宏块、码片、帧、场、界定画面中的对象的区域、画面的前景、画面的背景, 或者画面中特定的 (x, y) 坐标组。作为另一示例, 深度画面 (深度图) 可以是例如完整的深度图或者仅包括例如用于对应的视频帧的单个宏块的深度信息的局部深度图。

[0360] 另外, 本领域技术人员将理解, 层 (或“视频”或“图像”或“画面”) 可以指代各种视频分量的任一种或者它们的组合。这种分量或者它们的组合包括例如, 亮度、色度、(YUV 或 YCbCr 或 YPbPr 或 YPcPr 的) Y、(YUV 的) U、(YUV 的) V、(YCbCr 的) Cb、(YCbCr 的) Cr、(YPbPr 的) Pb、(YPbPr 或 YPcPr 的) Pr、(YPcPr 的) Pc、(RGB 的) 红色, (RGB 的) 绿色, (RGB 的) 蓝色, S 视频、以及这些分量的任一种的负值或正值。此外, 这些不同类型的分量可以用于描述的实现方式。例如, YUV 组分量可以用于所描述的一个或更多实现方式, 并且在典

型的实现方式中在宏块级上进行组合。另外,其它画面类型可以用于此处描述的实现方式和特征。这种其它画面类型可以包括例如包含除 2D 视频、深度、遮挡或者背景、透明或者边缘不连续性之外的信息。

[0361] 另外,本申请或者其权利要求中可以引用“确定”各种信息。确定信息可以包括例如以下中的一个或更多:估计信息、计算信息、预测信息、从数据列表或者其它组中识别信息,或者从存储器中检索信息。

[0362] 类似地,“访问”意图是宽泛的用语。访问信息可以包括任何操作,例如,使用、存储、发送、传送、接收、检索、修改、解析,或者提供信息。

[0363] 许多实现方式引用“参考”。“参考”可以是例如来自参考的基于像素的差分用于预测源的传统的参考。参考还可以或者可替换地用于以不同方式来预测源。例如,在一个实现方式中,边缘位置或者边缘不连续性的量度被用于预测源。通常,可以从参考中借用任何信息来辅助预测源。已经给出诸如像素值、边缘位置,和边缘不连续的量度之类的信息的示例,但是其它类型的信息也是可行的。

[0364] 在此描述的实现方式可以在例如方法或处理,装置、软件程序、数据流,或者信号中实现。即使仅仅在实现方式的单一形式的背景下进行讨论(例如,仅仅作为方法讨论),但所讨论的特征的实现方式还可以在其它形式中实现(例如,装置或者程序)。装置可以在例如合适的硬件、软件和固件中实现。方法可以在例如诸如例如处理器的装置中实现,所述处理器通常指代处理装置,包括例如计算机、微处理器、集成电路,或者可编程逻辑器件。处理器还可以包括通信装置,诸如,例如,计算机、蜂窝电话、便携式/个人数字助理(“PDA”),以及其它便于在终端用户之间交互信息的装置。

[0365] 在此描述的各种处理和特征的实现方式可以在各种不同的设备或者应用中体现,具体地,例如与数据编码和解码相关联的设备或者应用。这种设备的示例包括:编码器、解码器、处理来自解码器的输出的后处理器,或者向编码器提供输入的预处理器、视频编码器、视频解码器、视频编解码器、网络服务器、机顶盒、膝上型计算机、个人计算机、蜂窝电话、PDA,以及其它通信装置。如应清楚地,该设备可以是移动的并且甚至被安装在移动车辆上。

[0366] 另外,可以通过由处理器执行的指令来实现所述方法,并且这种指令(和/或由实现方式产生的数据值)可以被存储在处理器可读介质上,诸如,例如,集成电路、软件载体或者其它存储装置上,诸如例如,硬盘、致密盘、随机存取存储器(“RAM”),或者只读存储器(“ROM”)。所述指令可以形成在处理器可读介质上有形体现的应用程序。指令可以是例如处于硬件、固件、软件,或者其组合中。指令可以在例如操作系统、单独的应用、或者二者的组合中发现。因此处理器可以被表征为例如被配置为执行处理的装置和包括具有用于执行处理的指令的处理器可读介质(诸如存储装置)这二者。此外,除了指令或者代替指令,处理器可读介质可以存储由实现方式产生的数据值。

[0367] 对于本领域技术人员明显的,实现方式可以产生被格式化为承载例如可以被存储或传输的信息的各种信号。该信息可以包括例如用于执行方法的指令,或者由上述的实现方式之一产生的数据。例如,信号可以被格式化为承载作为数据的用于写入或者读取描述的实施例的语法的规则,或者承载由描述的实施例写的实际语法值。这种信号可以被格式化为例如电磁波(例如,使用频谱的无线电频率部分)或者基带信号。所述格式可以包括

例如编码数据流并且利用编码的数据流调制载波。信号承载的信息可以是例如模拟或者数字信息。如已知的,该信号可以通过各种不同的有线或者无线链路而传输。信号可以被存储在处理器可读的介质上。

[0368] 应理解,在实现方式的以上描述中,为了使本公开简化并且辅助对一个或更多各个方面的理解,在单个实现方式中、附图中,或者说明书中,有时将各种特征分组在一起。然而,本公开的该方法不应被解释为反映这种意图:所要求保护的发明比在每个权利要求中明确陈述的需要更多的特征。相反,如以下权利要求反映的,发明方面可以存在比在公开实施例中的单个上述的所有特征要少。因此,要理解,每个权利要求还提供单独的实现方式。

[0369] 已经描述了多个实现方式。然而,还应理解,可以进行各种修改。例如,不同实现方式中的要素可以被组合、补充、修改,或者移除以产生其它实现方式。此外,可以在功能块之间互换操作。另外,本领域技术人员将理解,可以对所公开实现方式替换其它结构和处理并且作为结果的实现方式将如所公开的实现方式以至少基本上相同的(多个)方式执行至少基本上相同的(多个)功能,以实现至少基本上相同的(多个)效果。相应地,这些和其它实现方式是本申请可设想到的并且在以下权利要求的范围内。

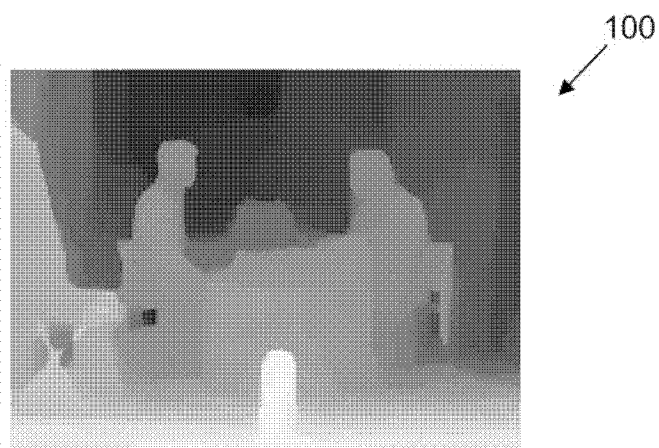


图 1

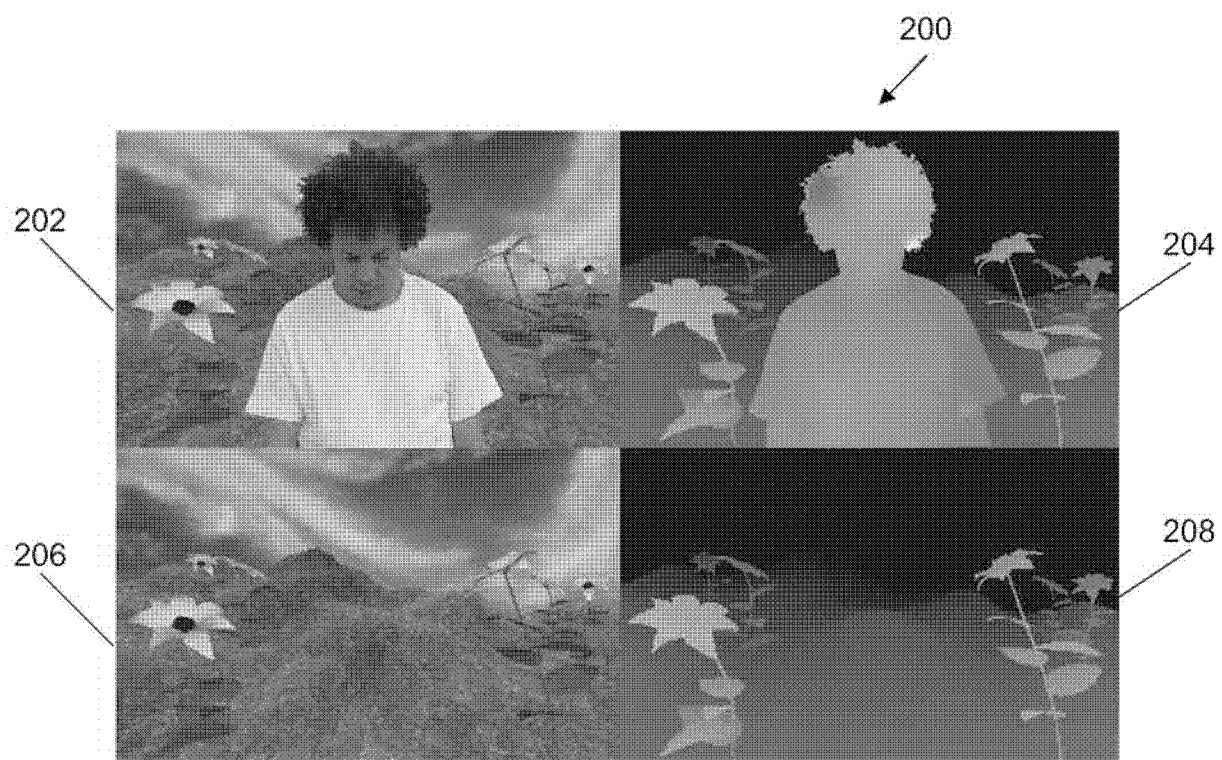


图 2

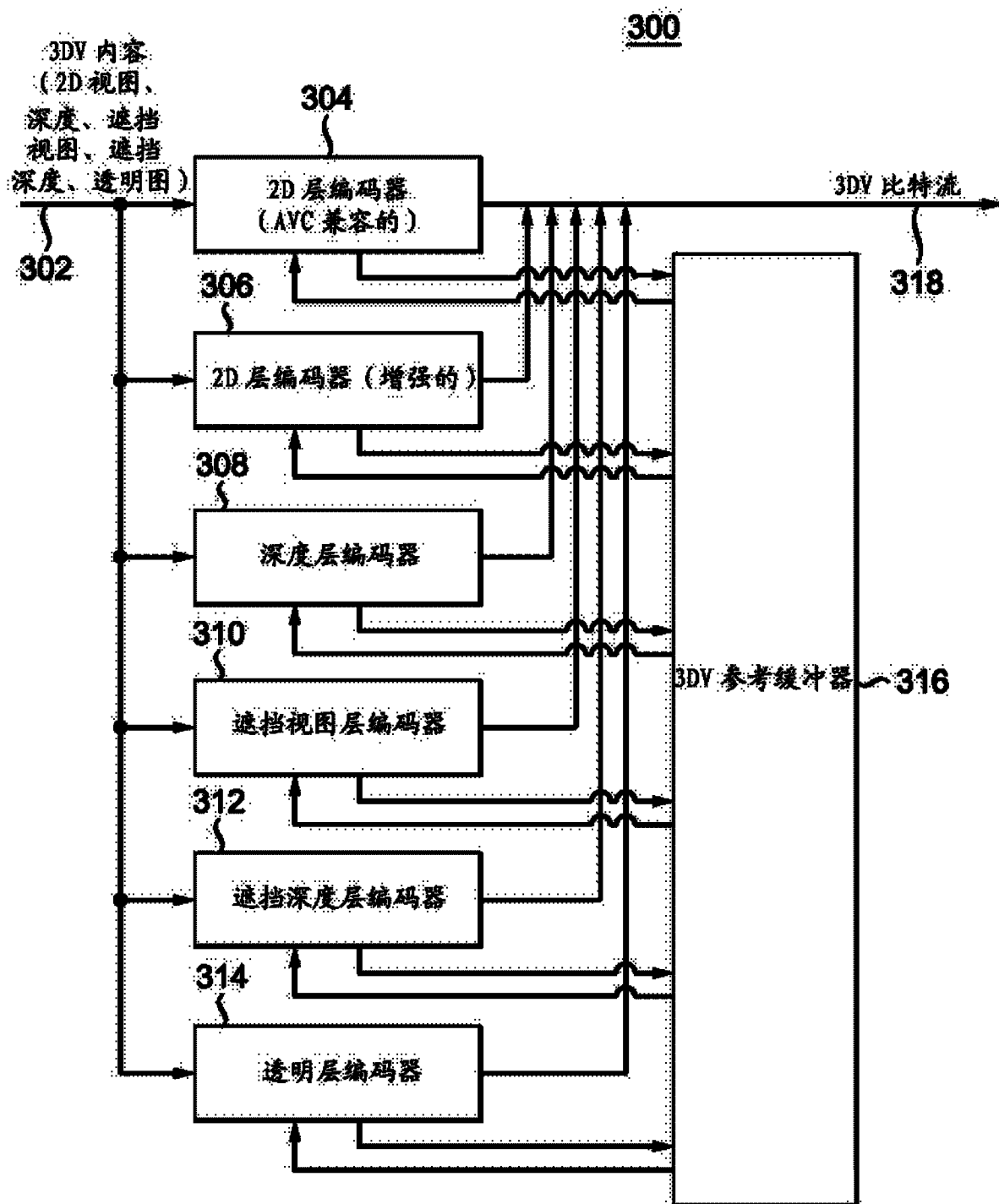


图 3

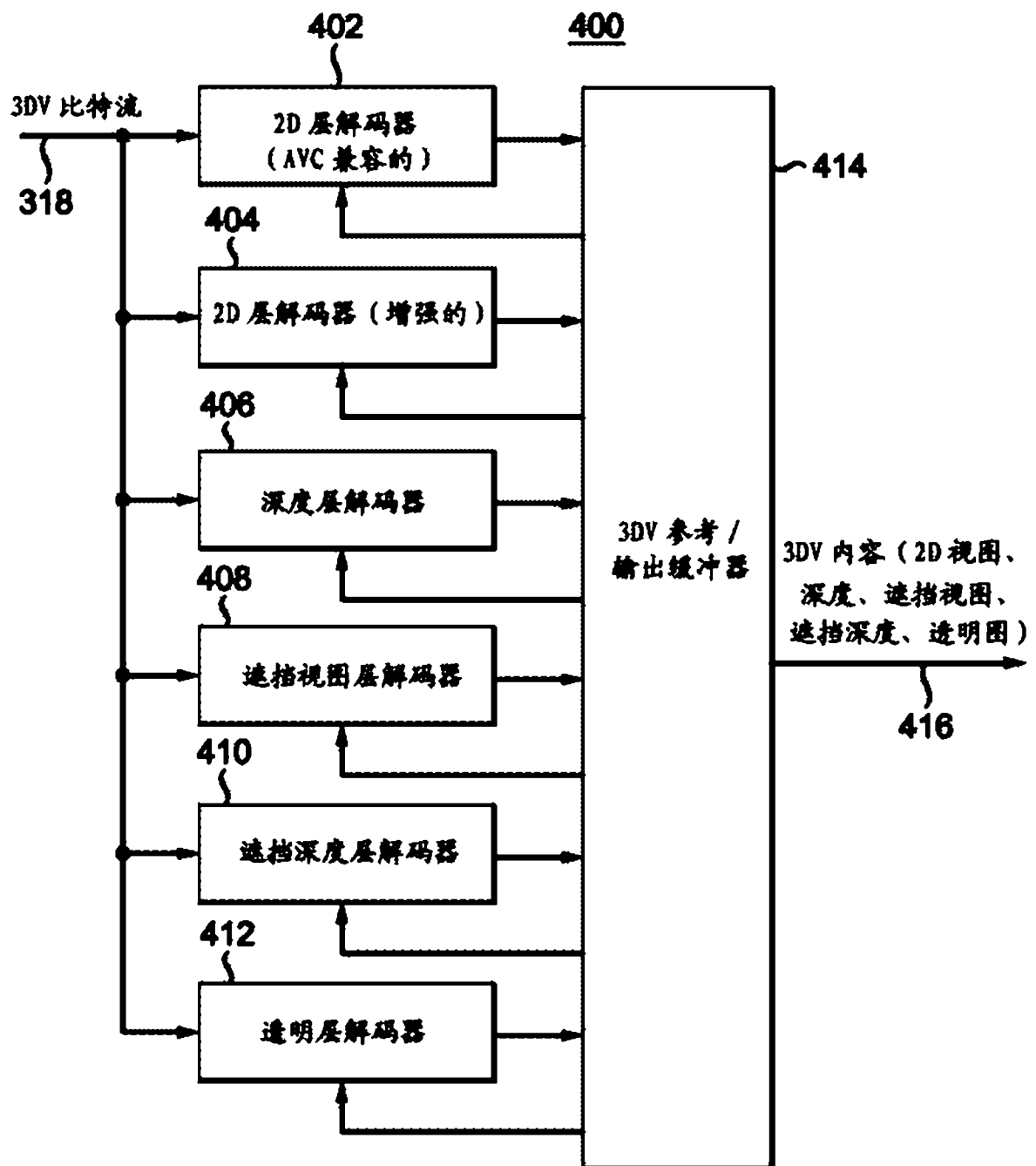


图 4

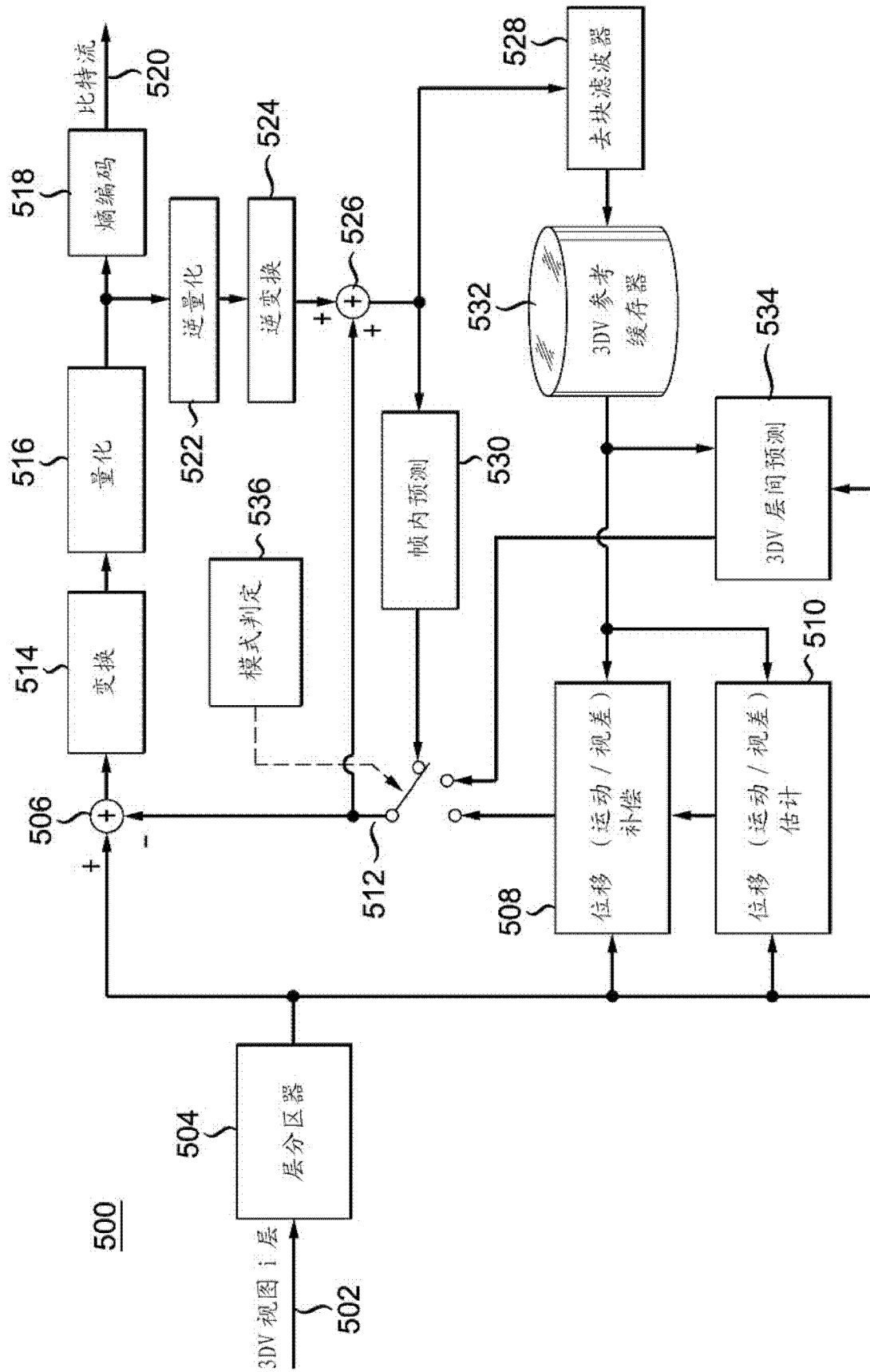


图 5

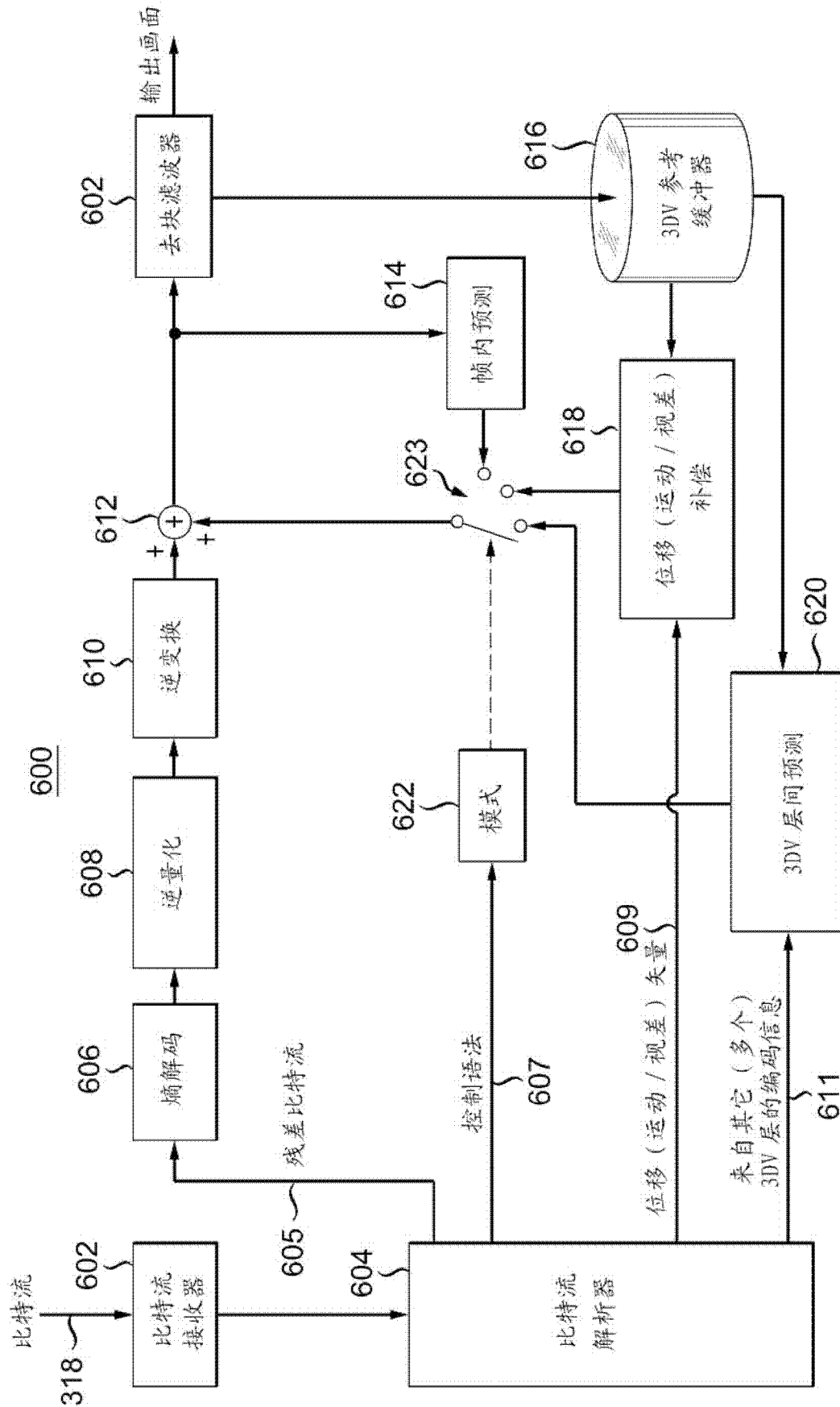


图 6

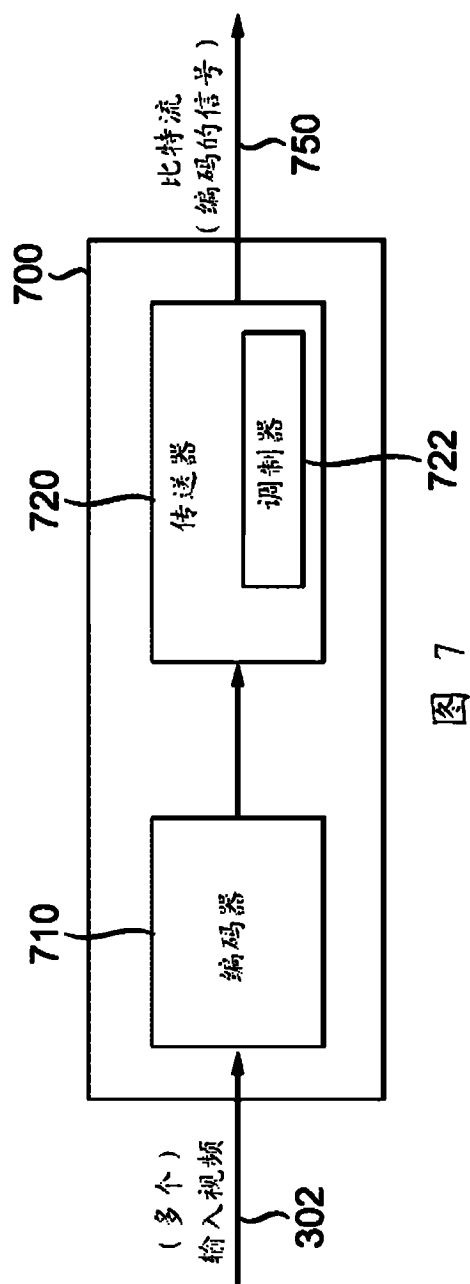


图 7

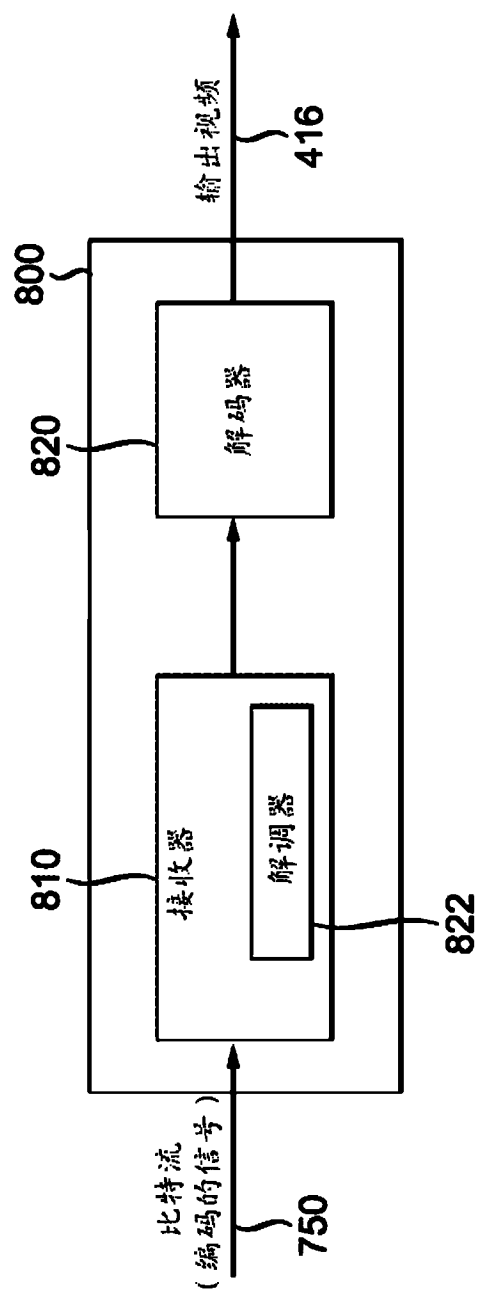


图 8

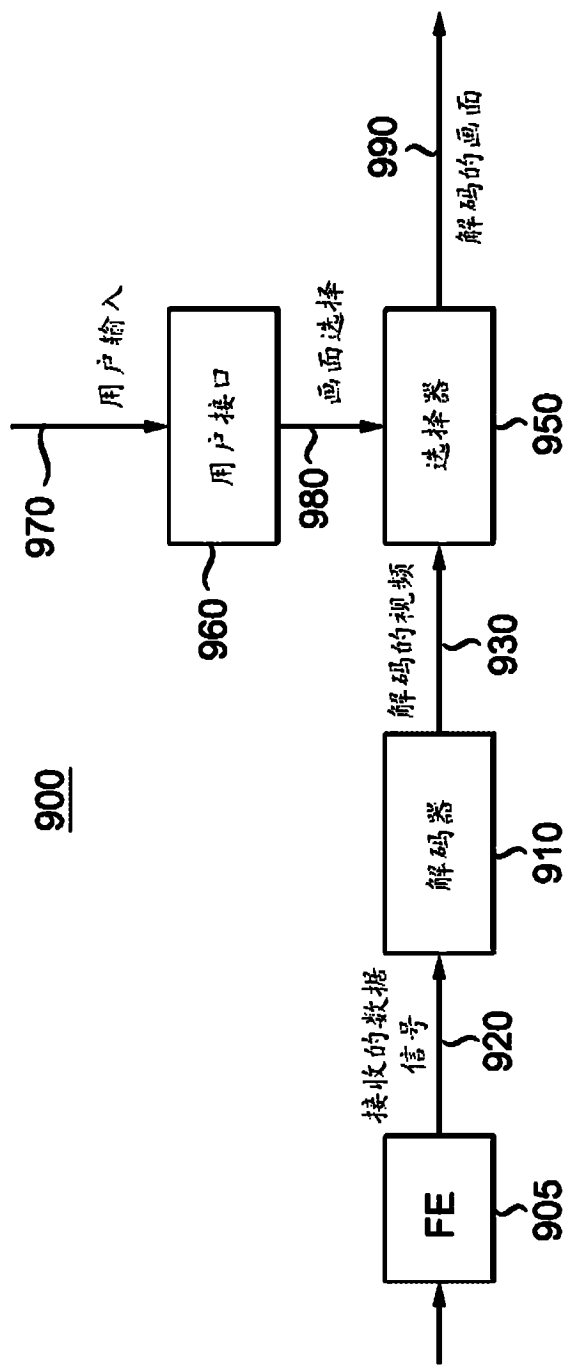


图 9

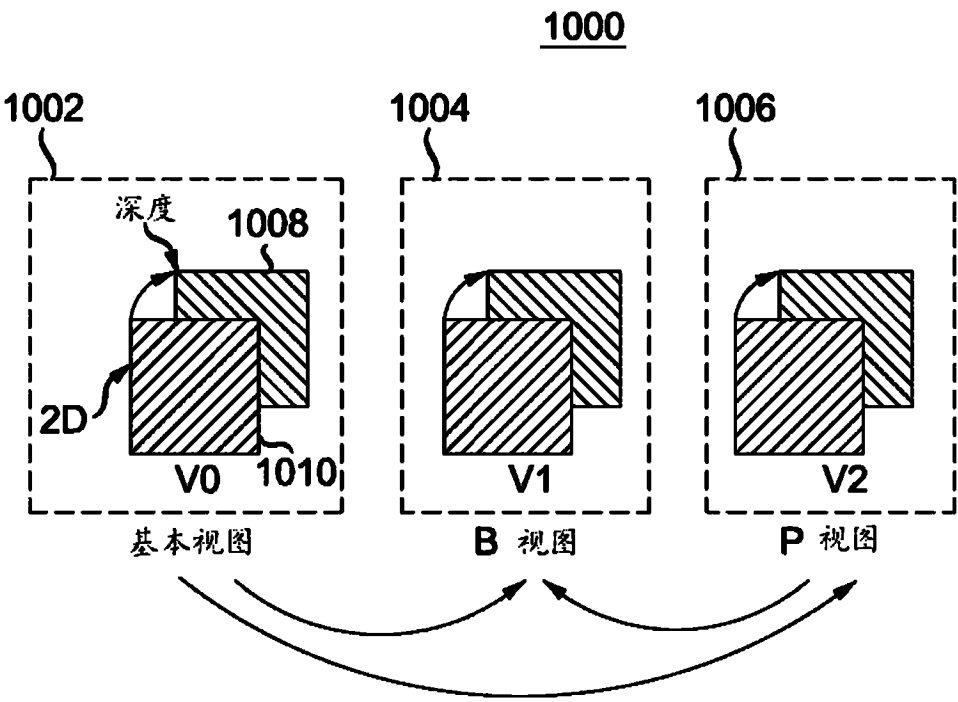


图 10

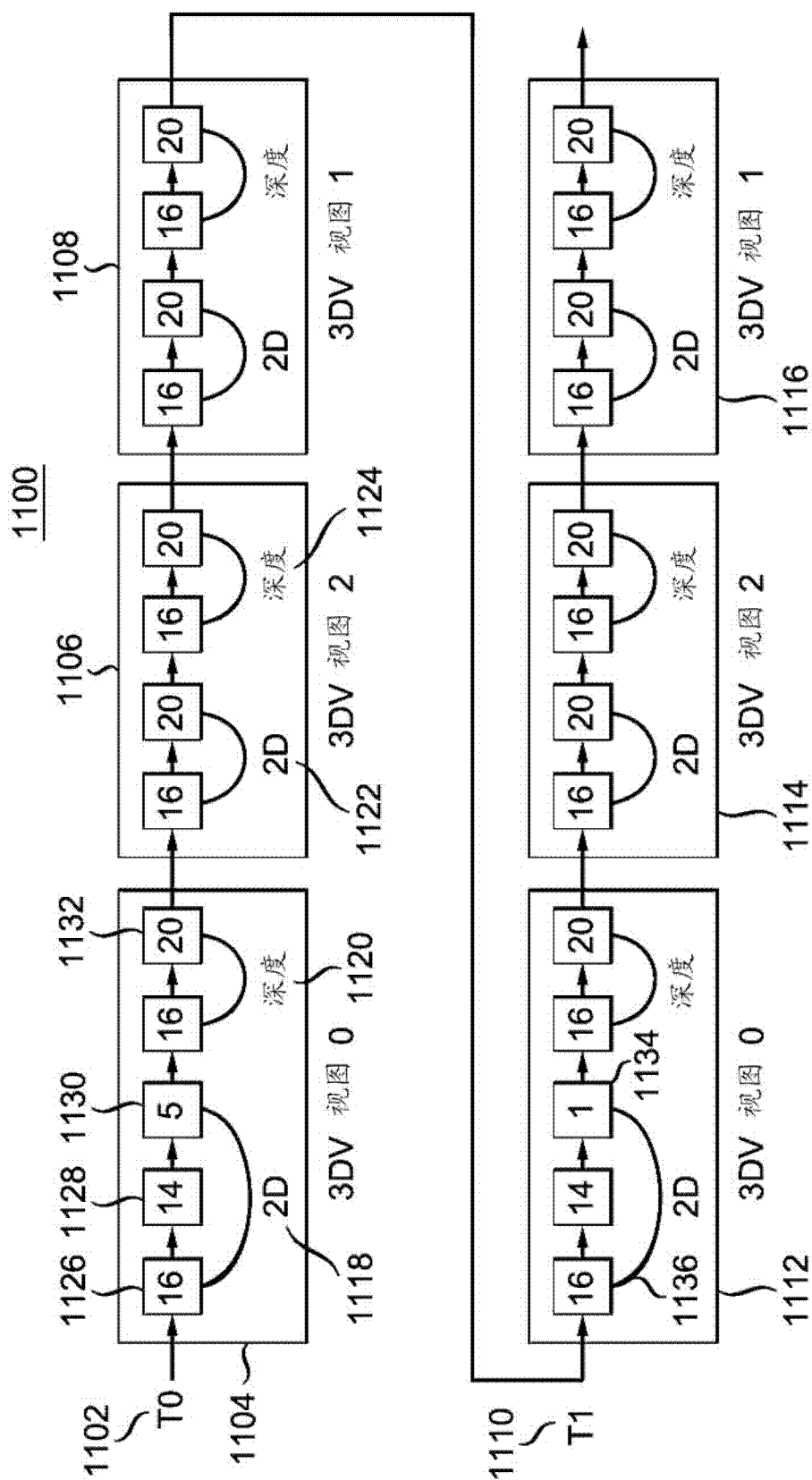


图 11

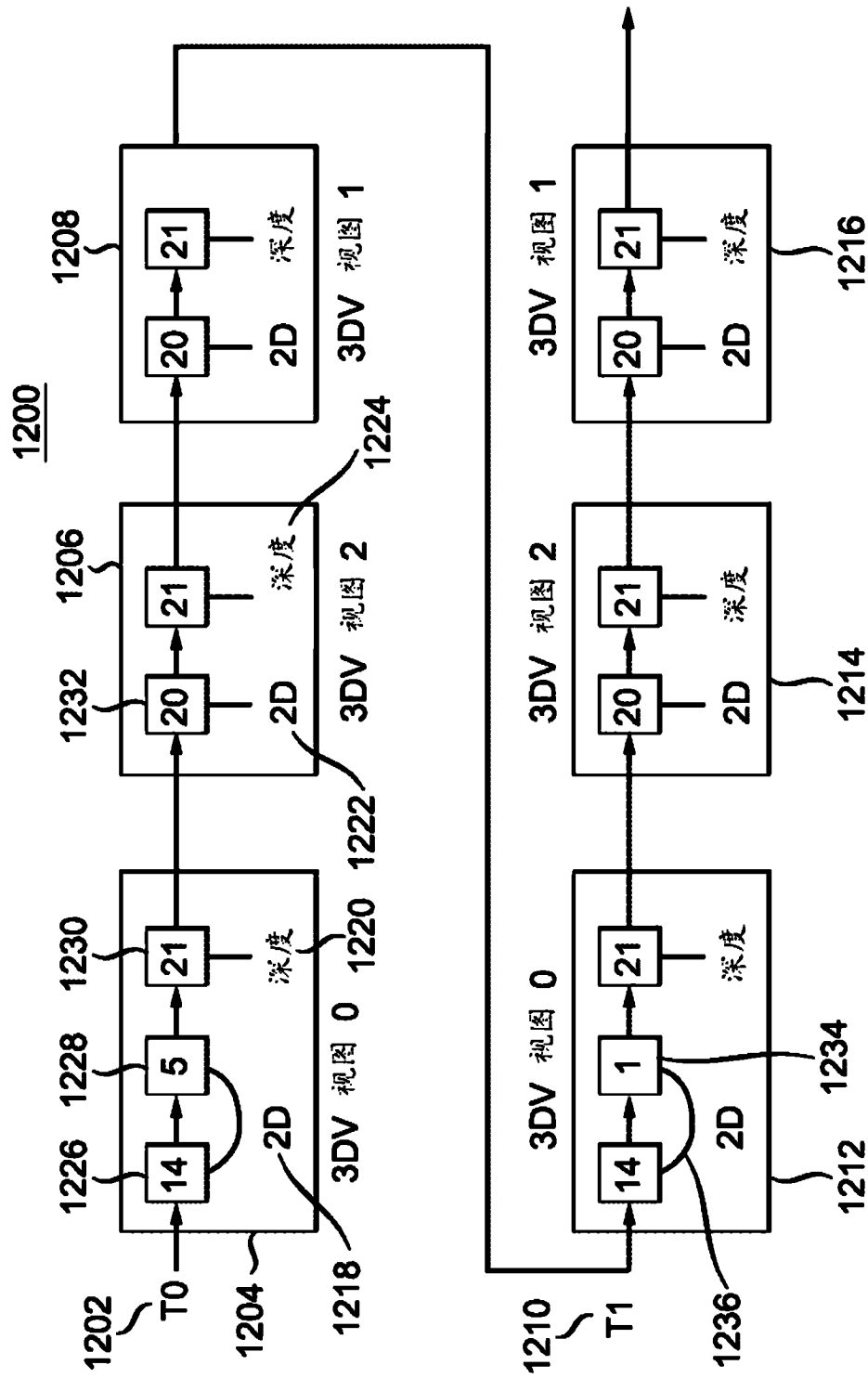


图 12

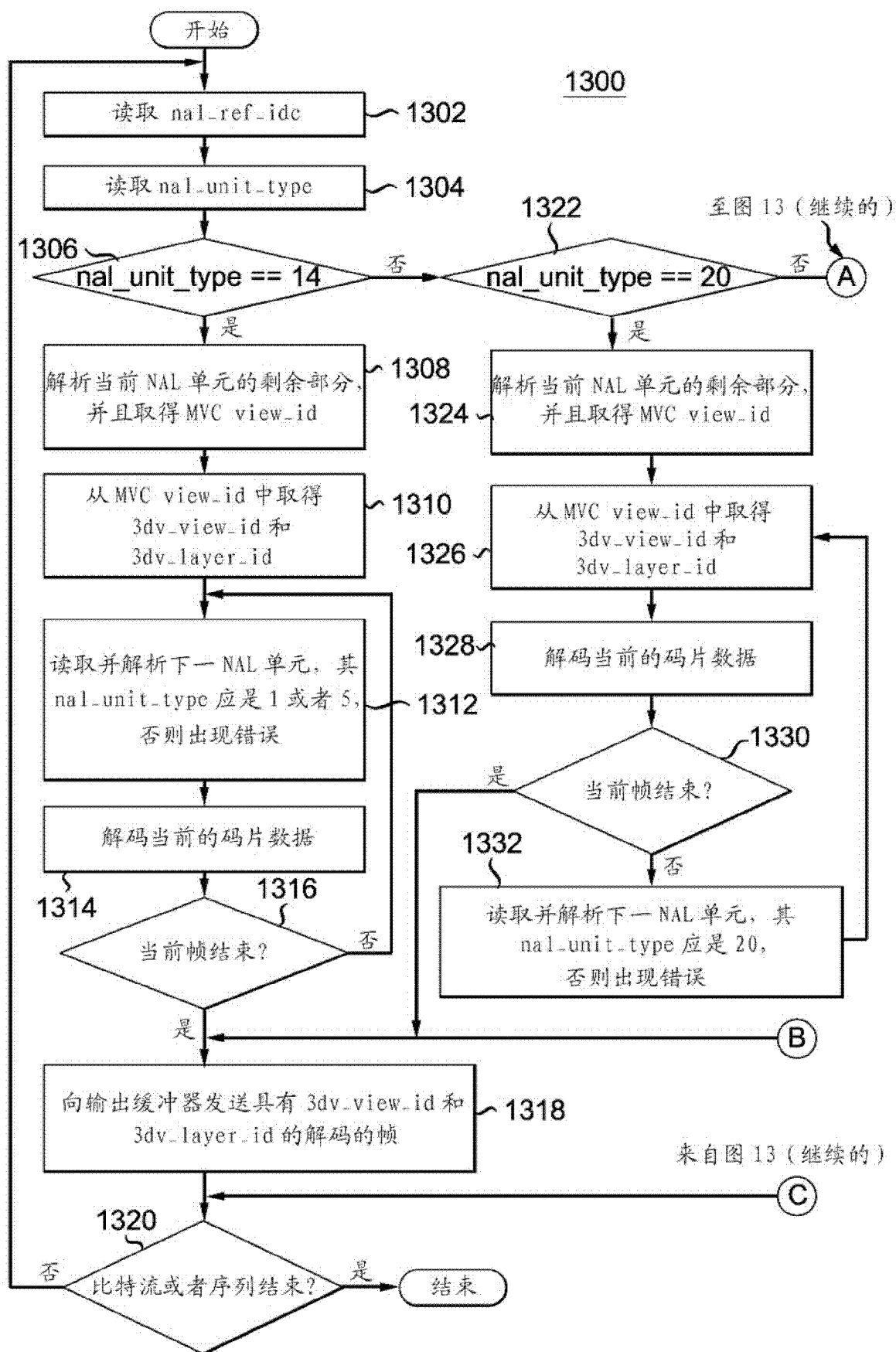


图 13

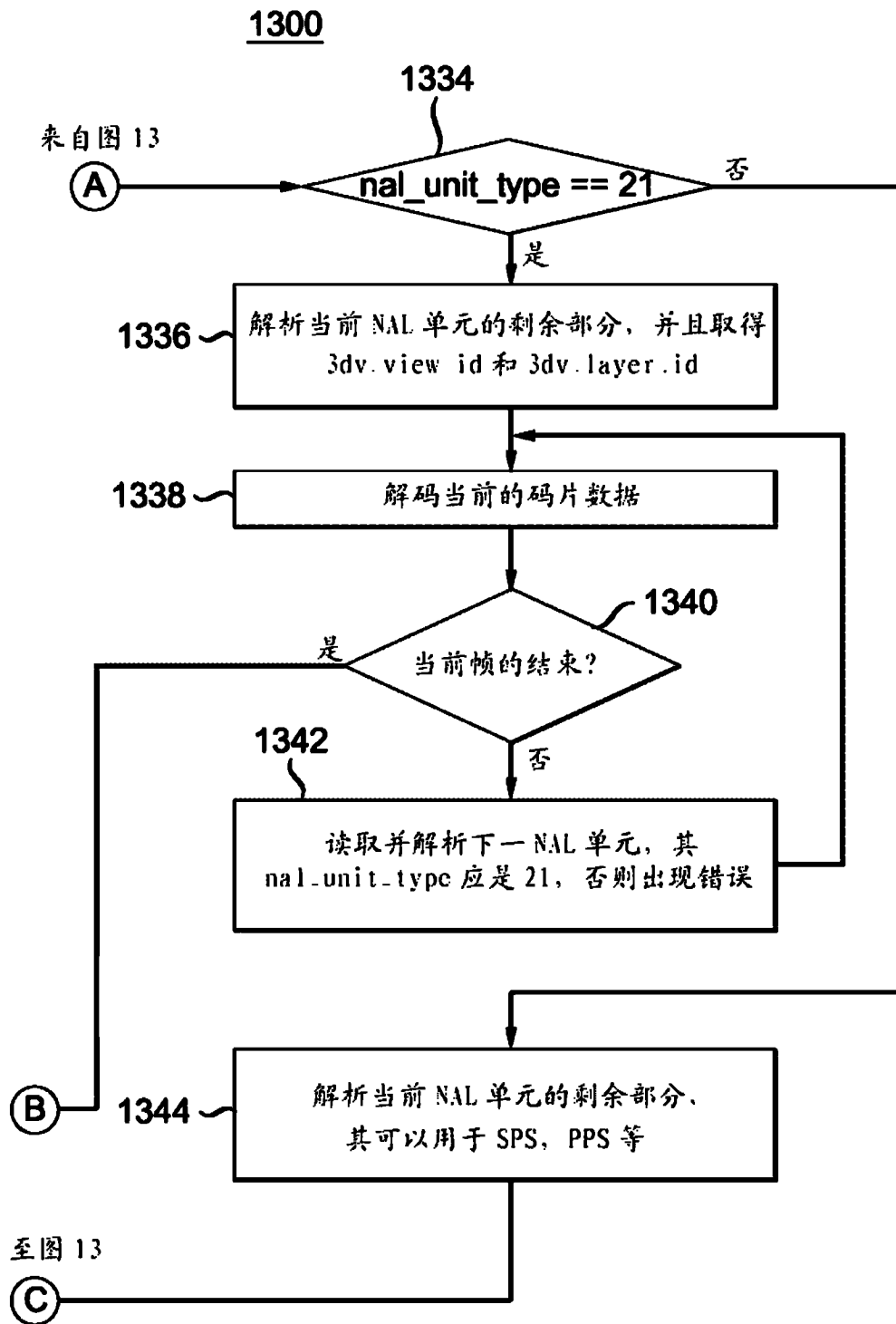


图 13(继续的)

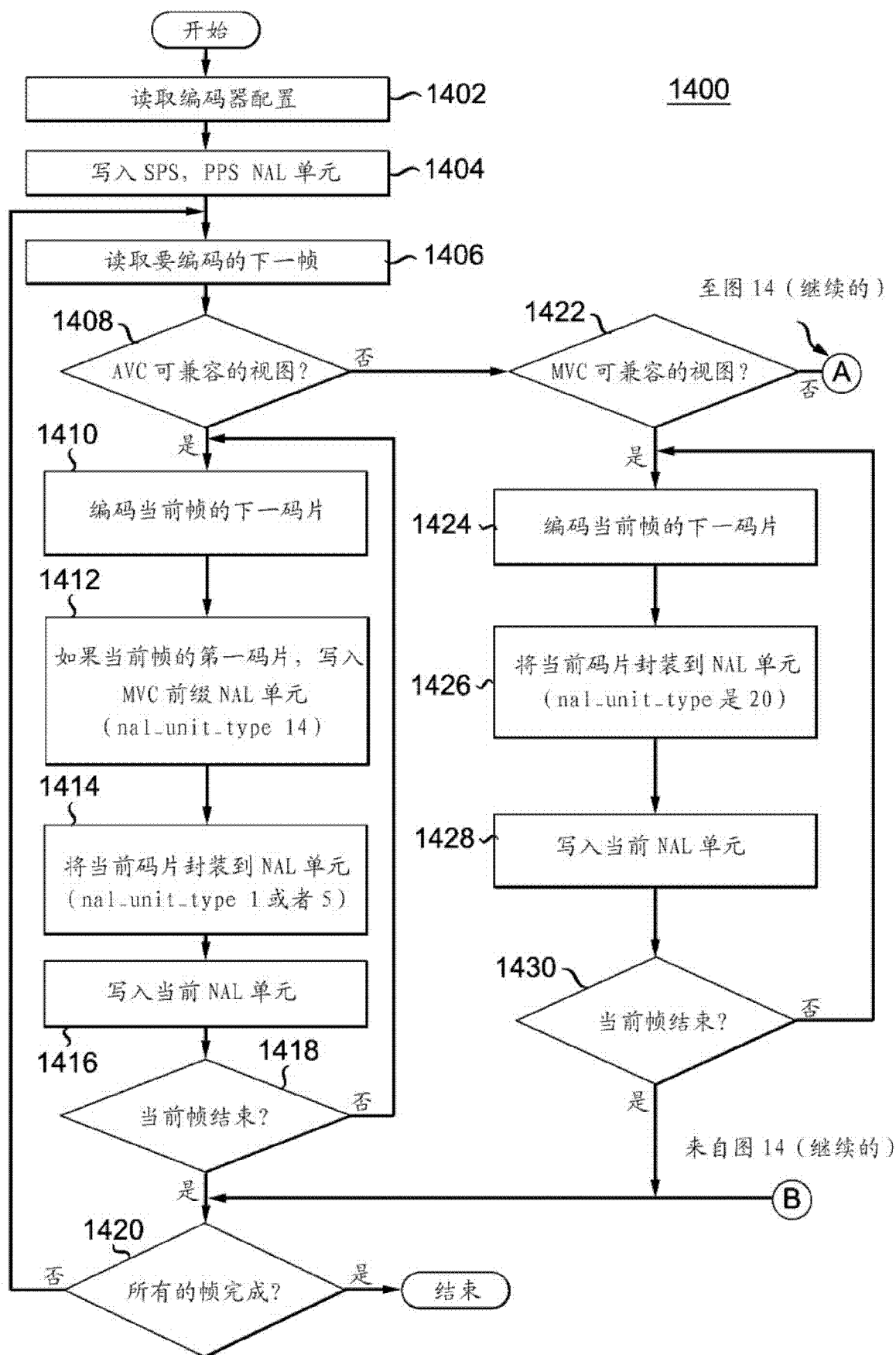


图 14

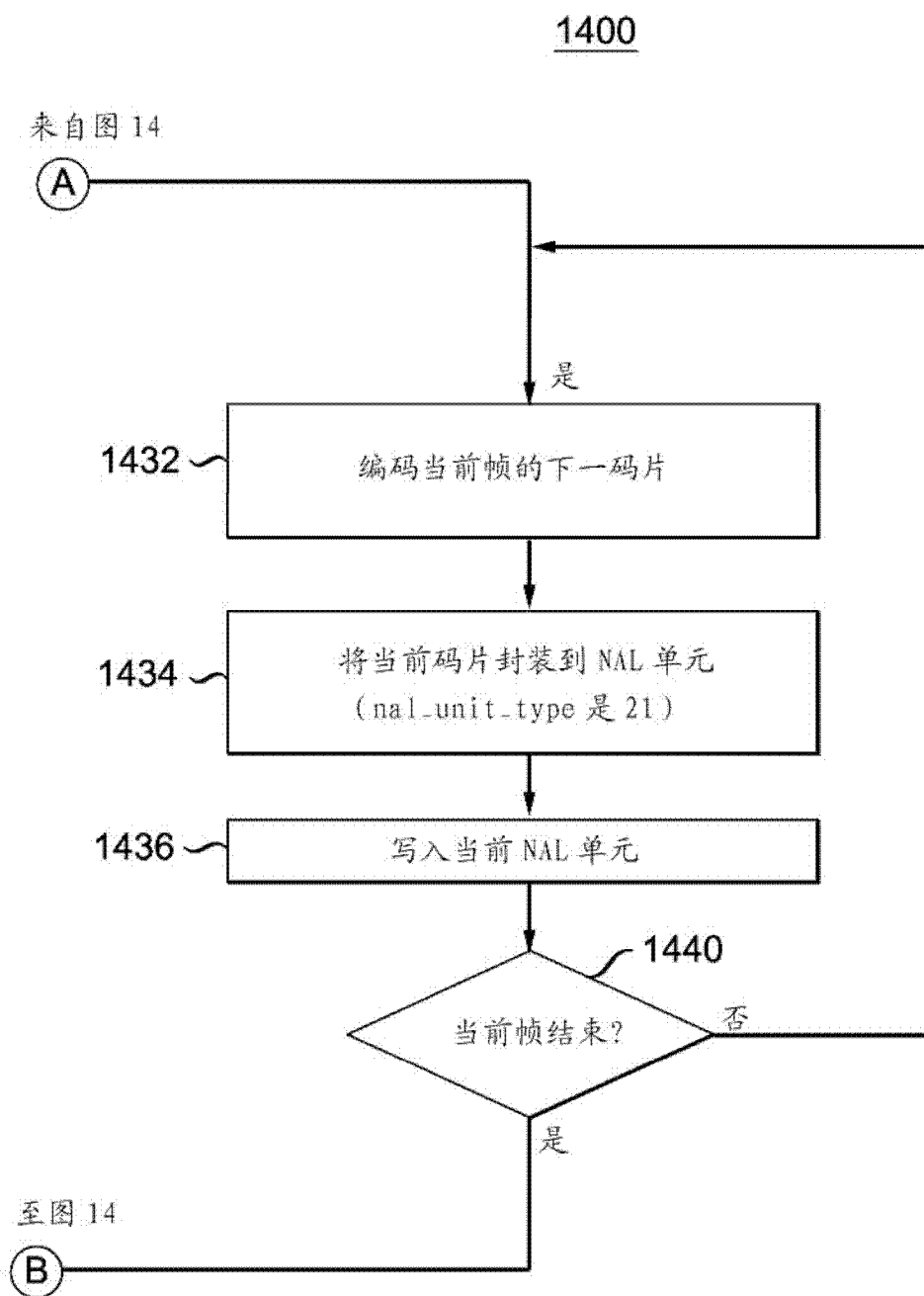


图 14(继续的)

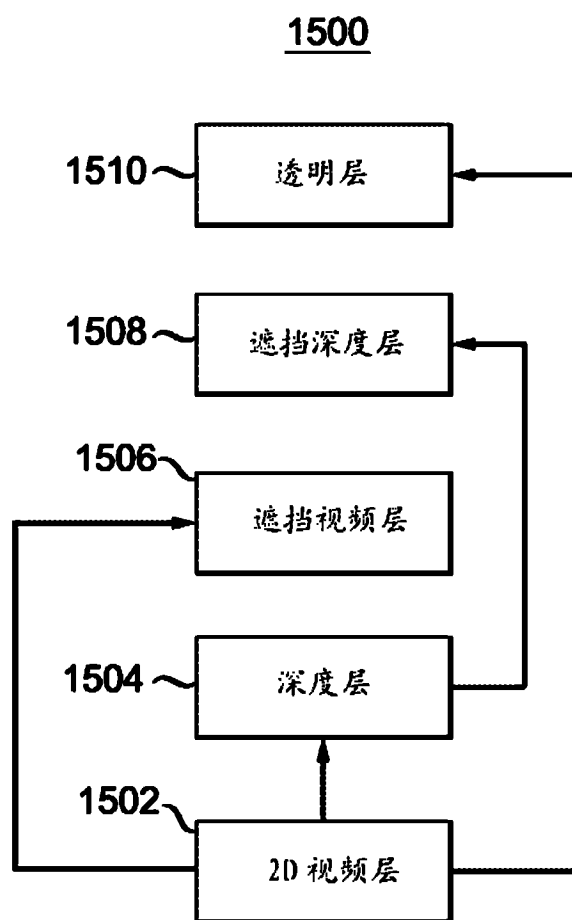


图 15

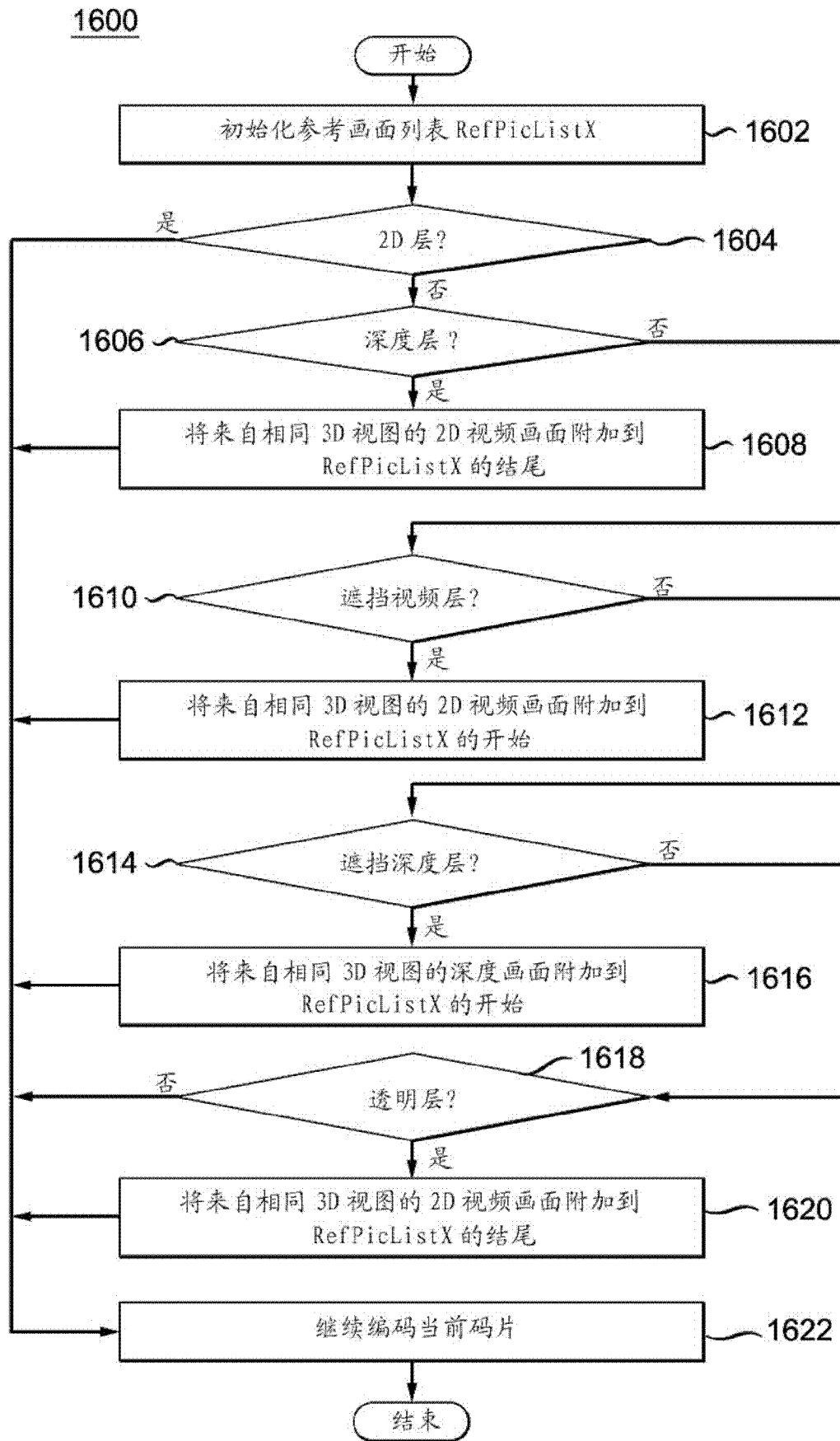


图 16

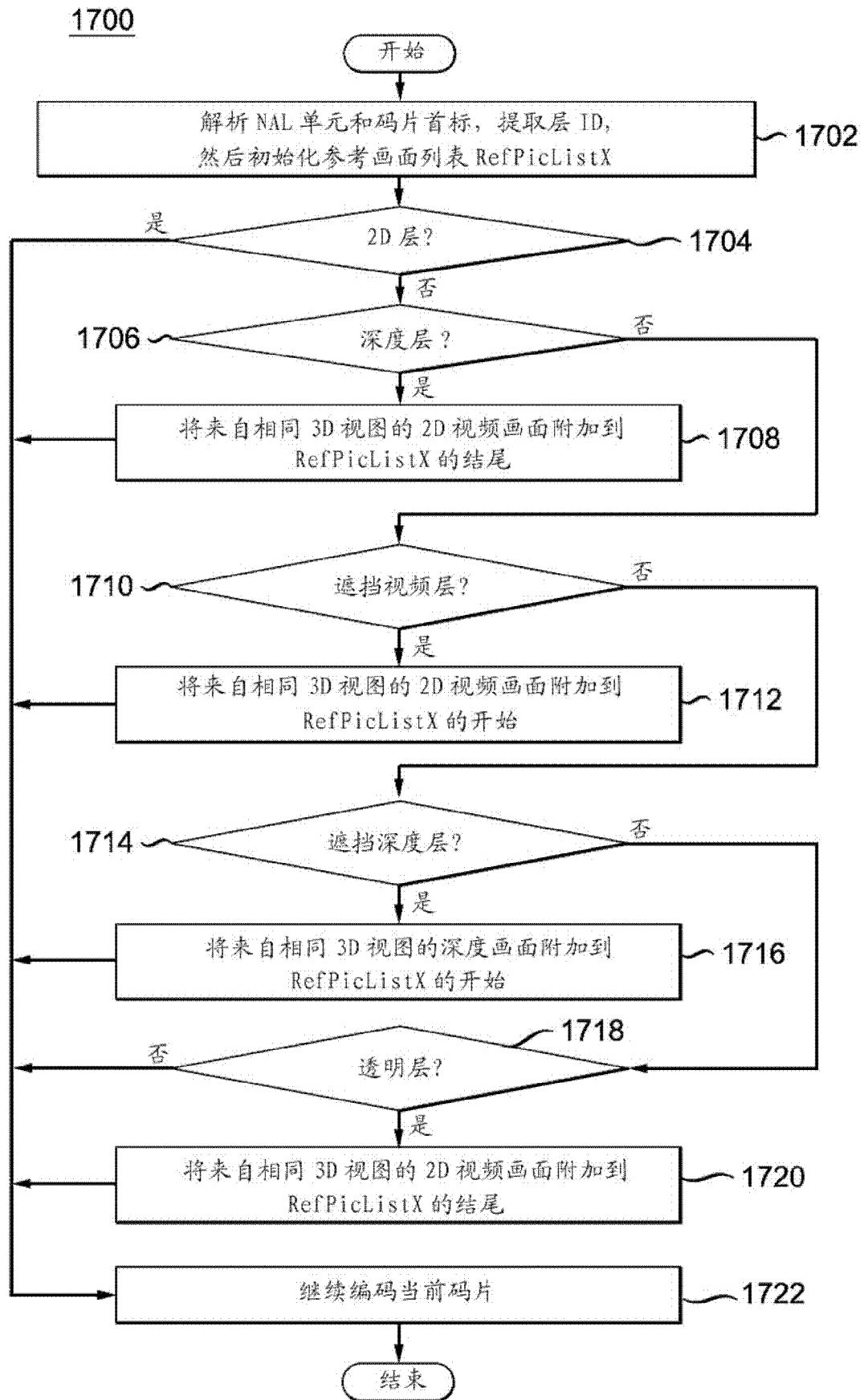


图 17

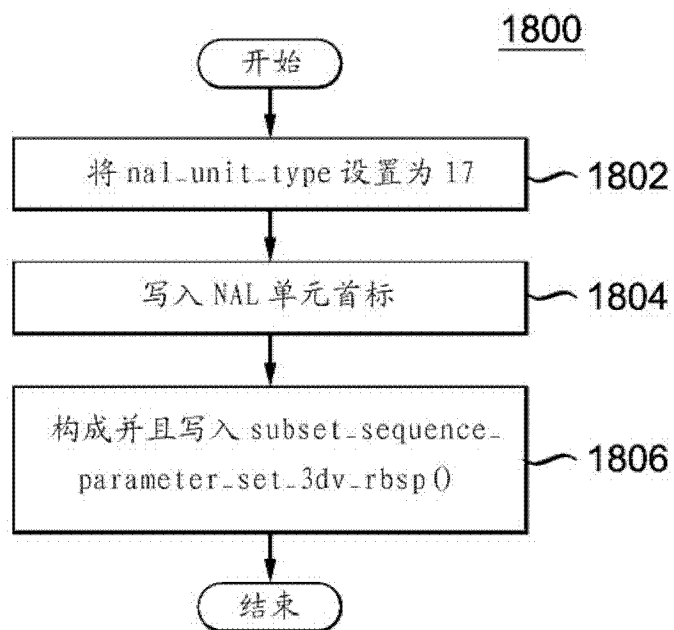


图 18

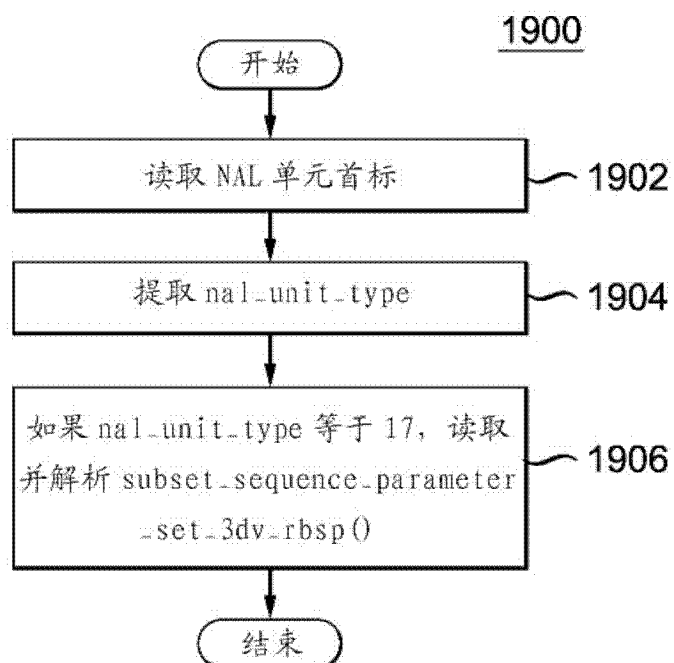


图 19

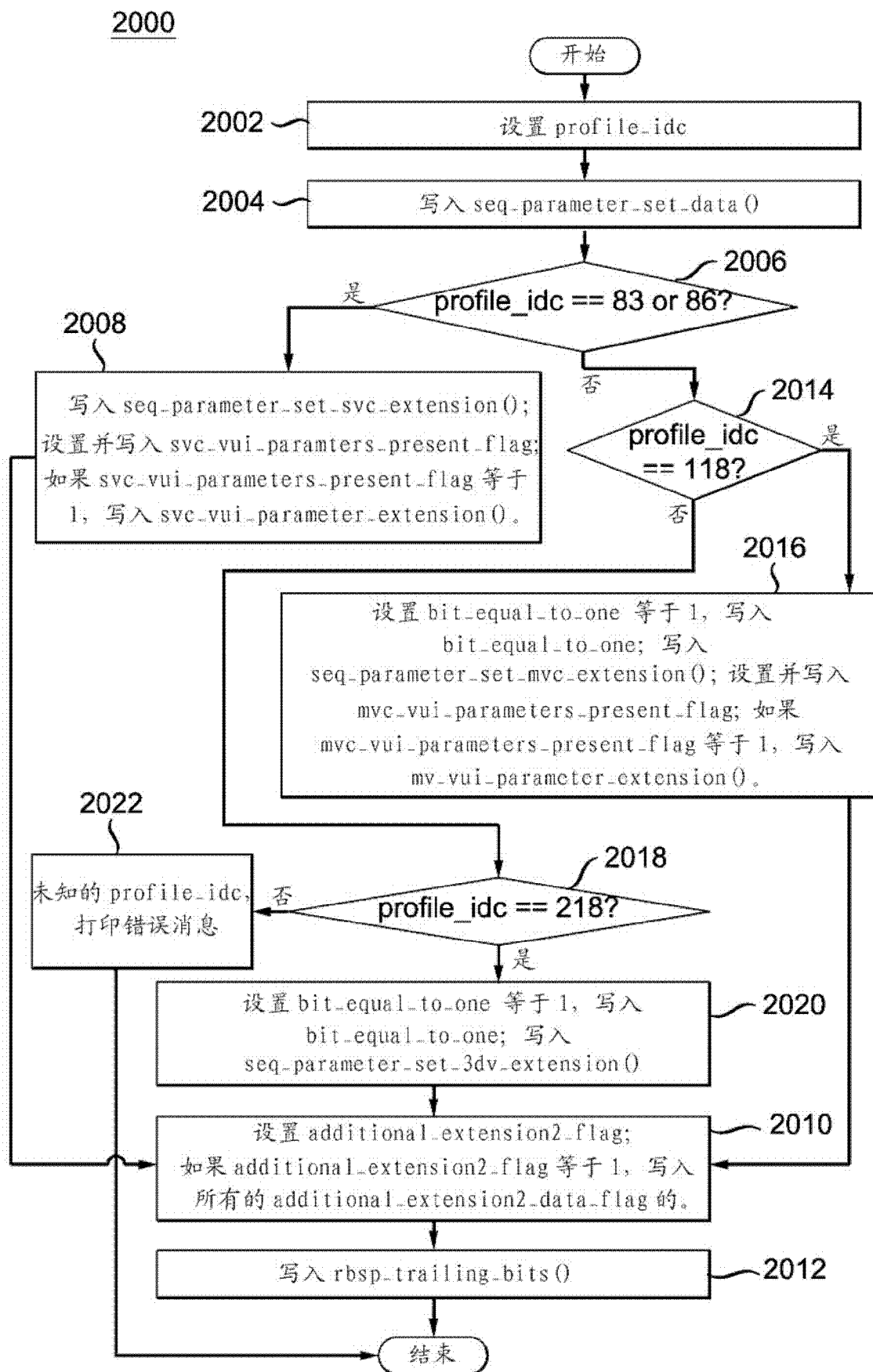


图 20

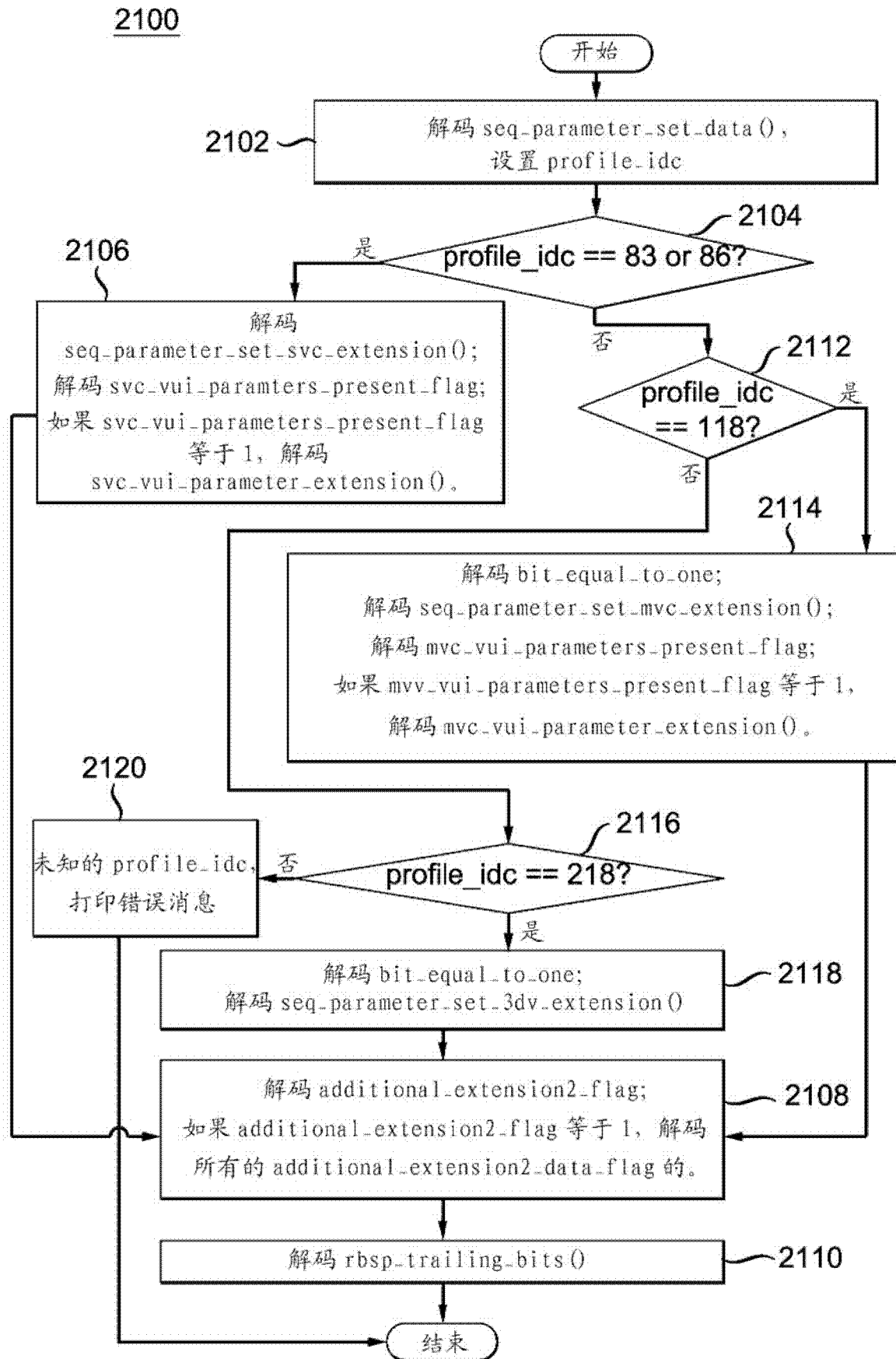


图 21

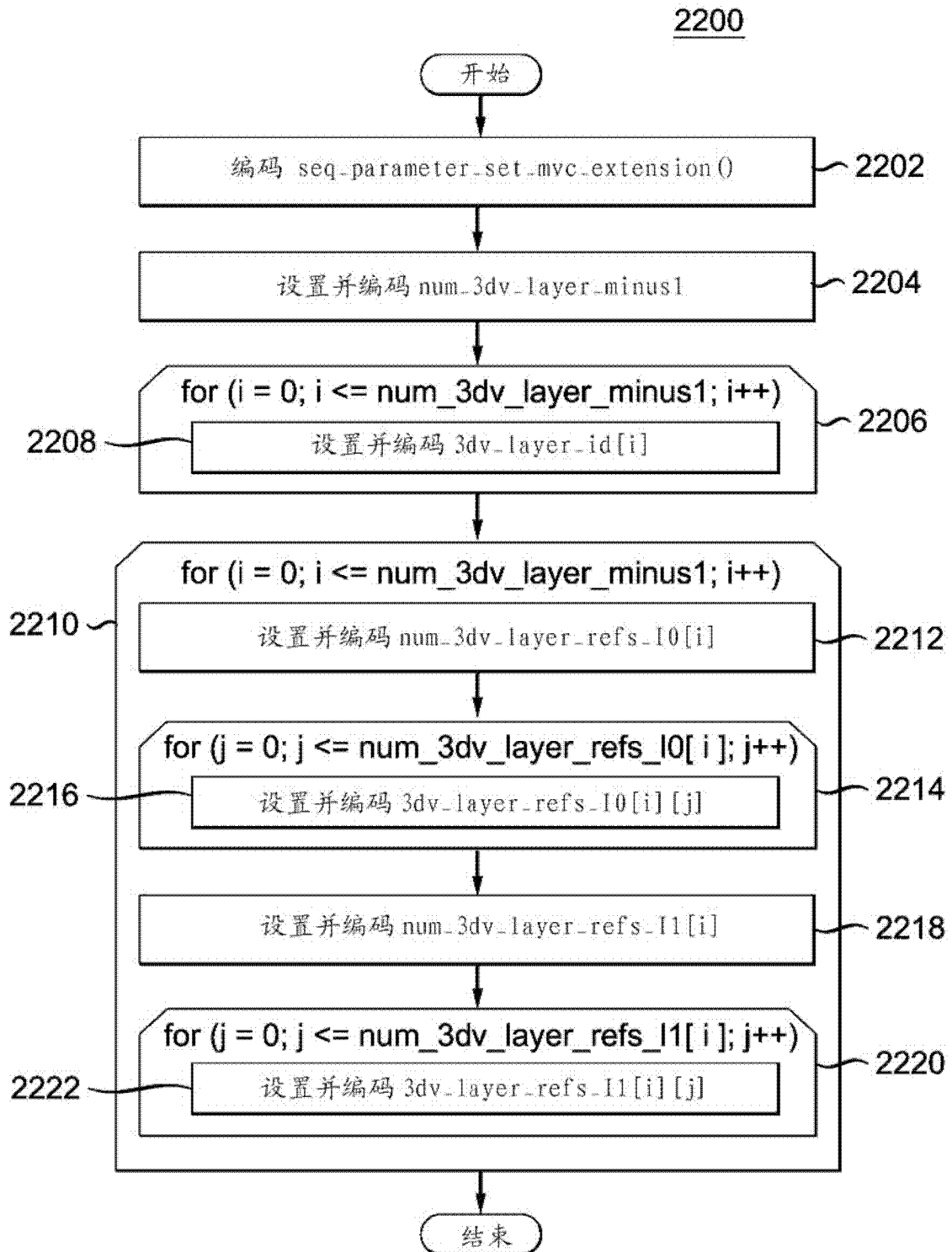


图 22

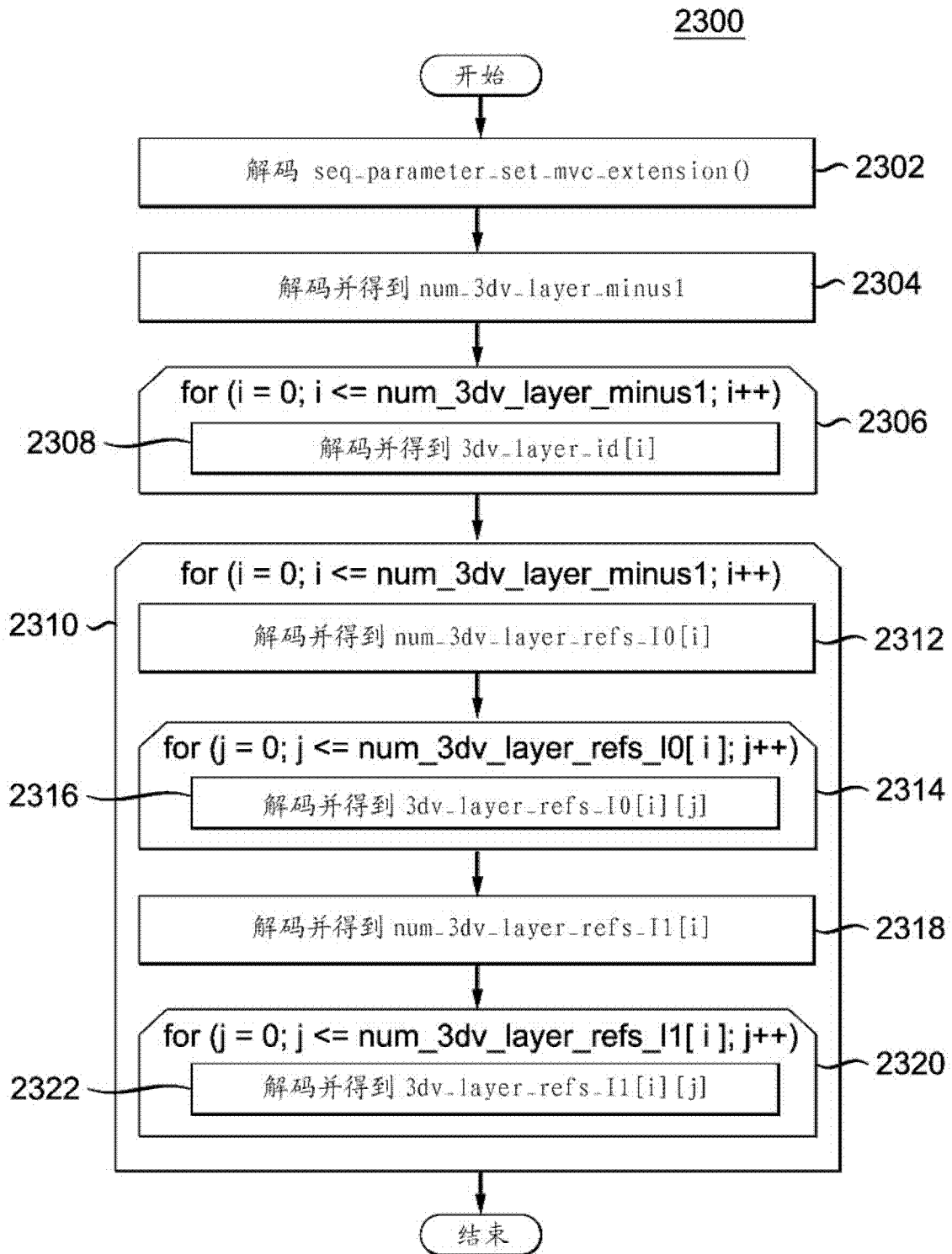


图 23

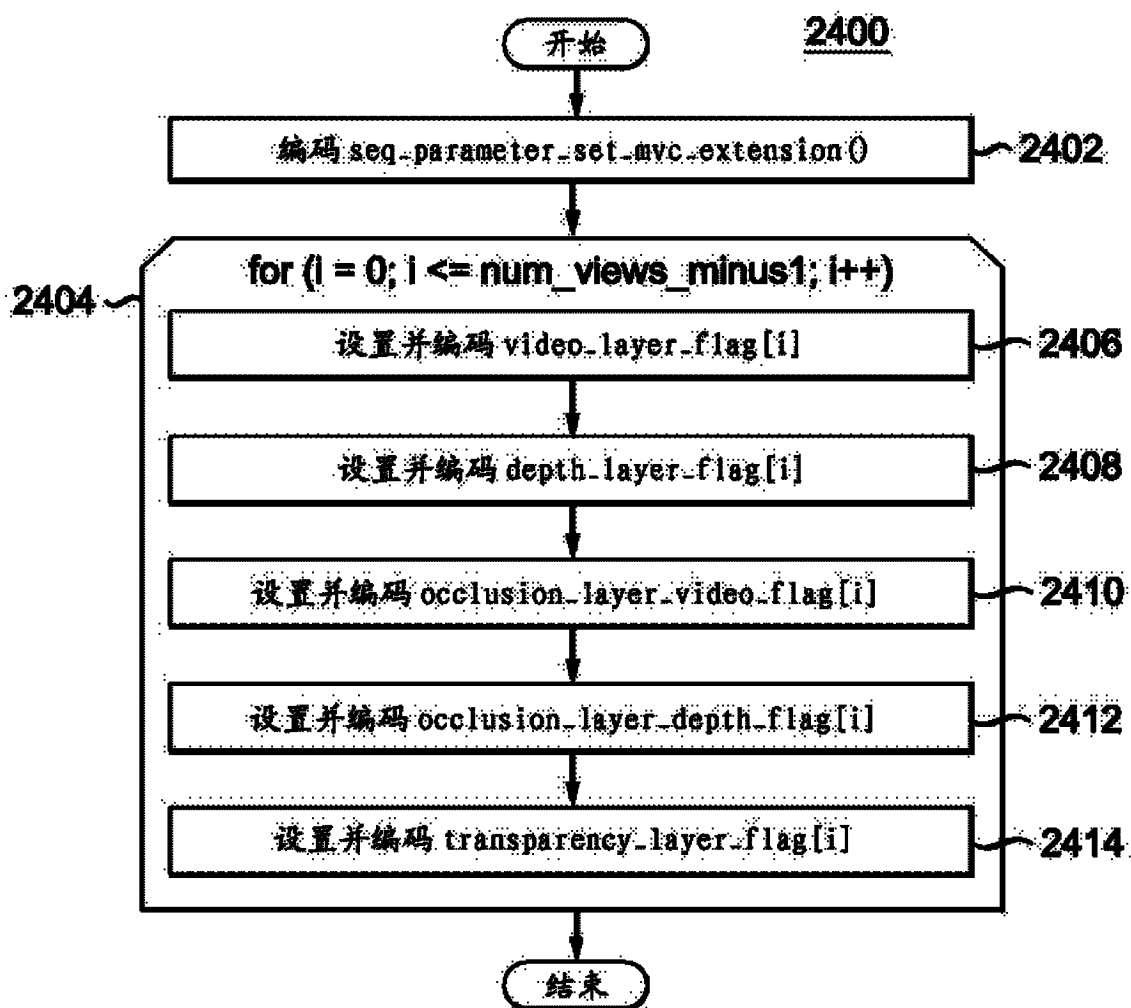


图 24

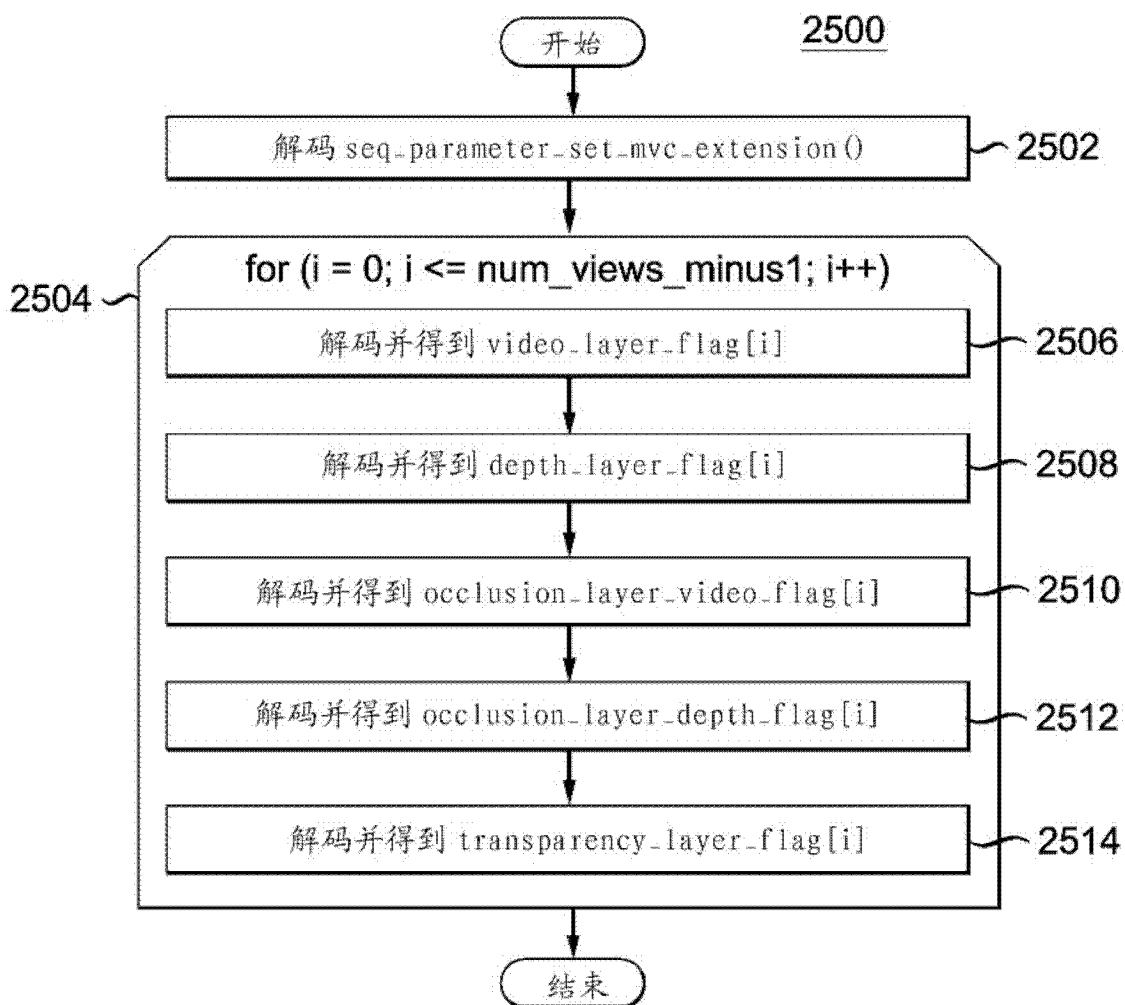


图 25

2600

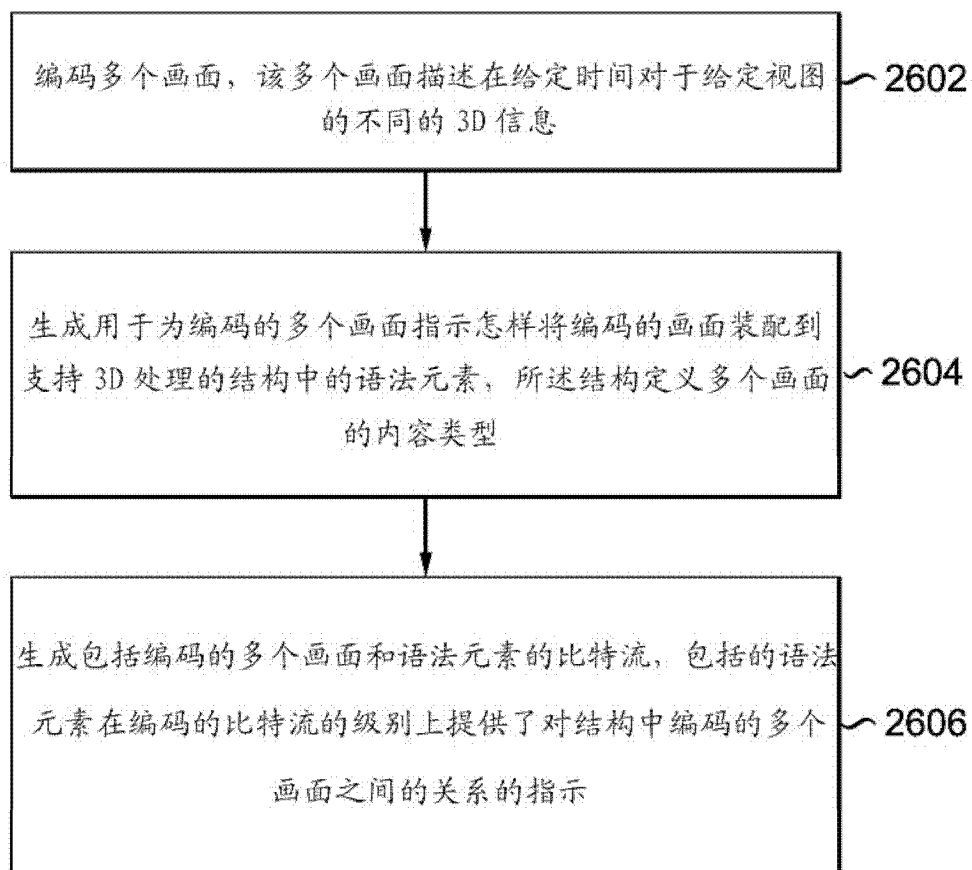


图 26

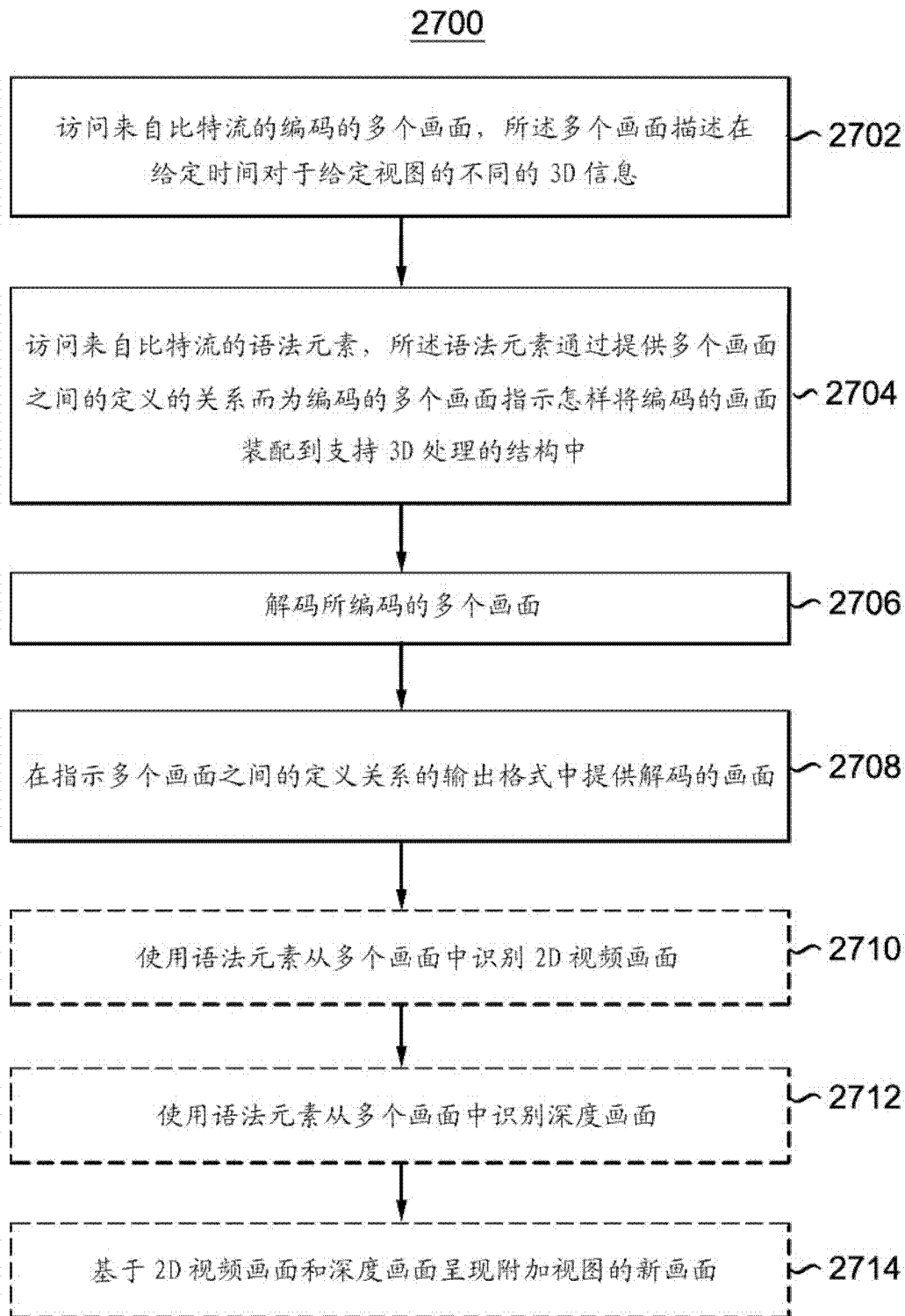


图 27

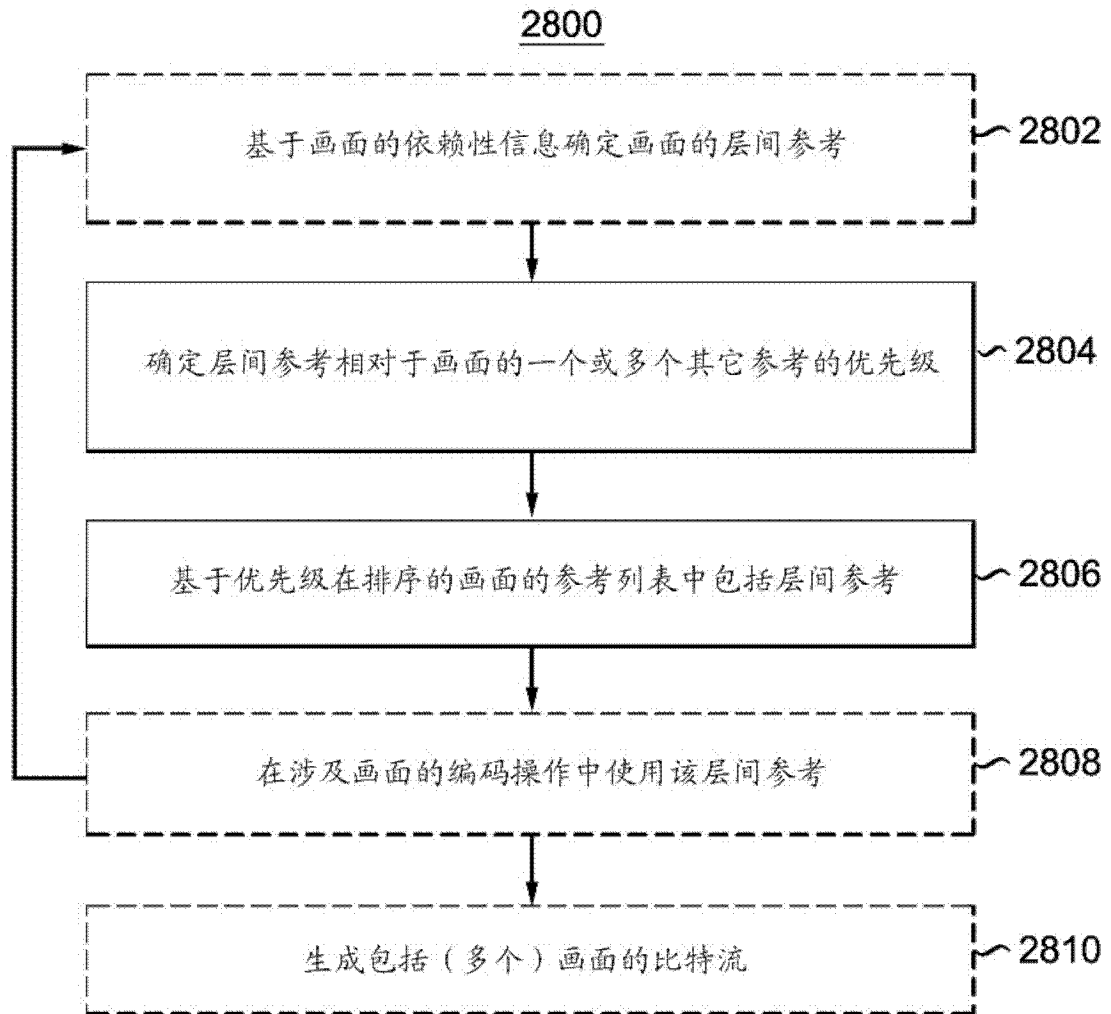


图 28

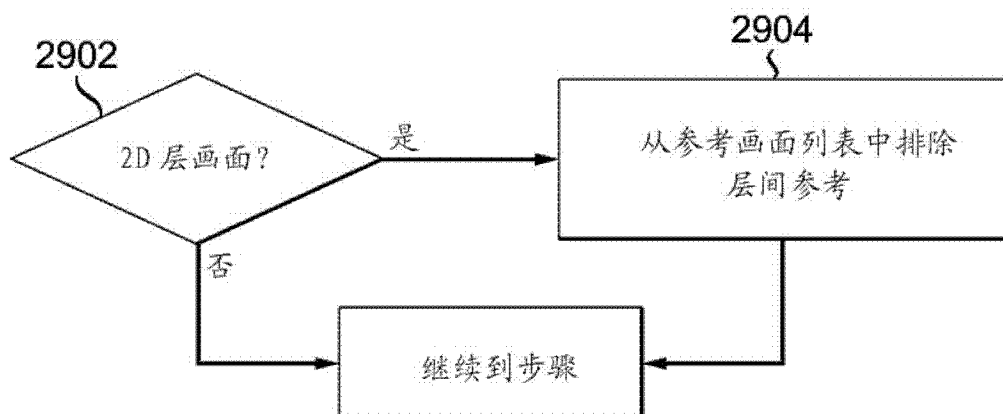


图 29

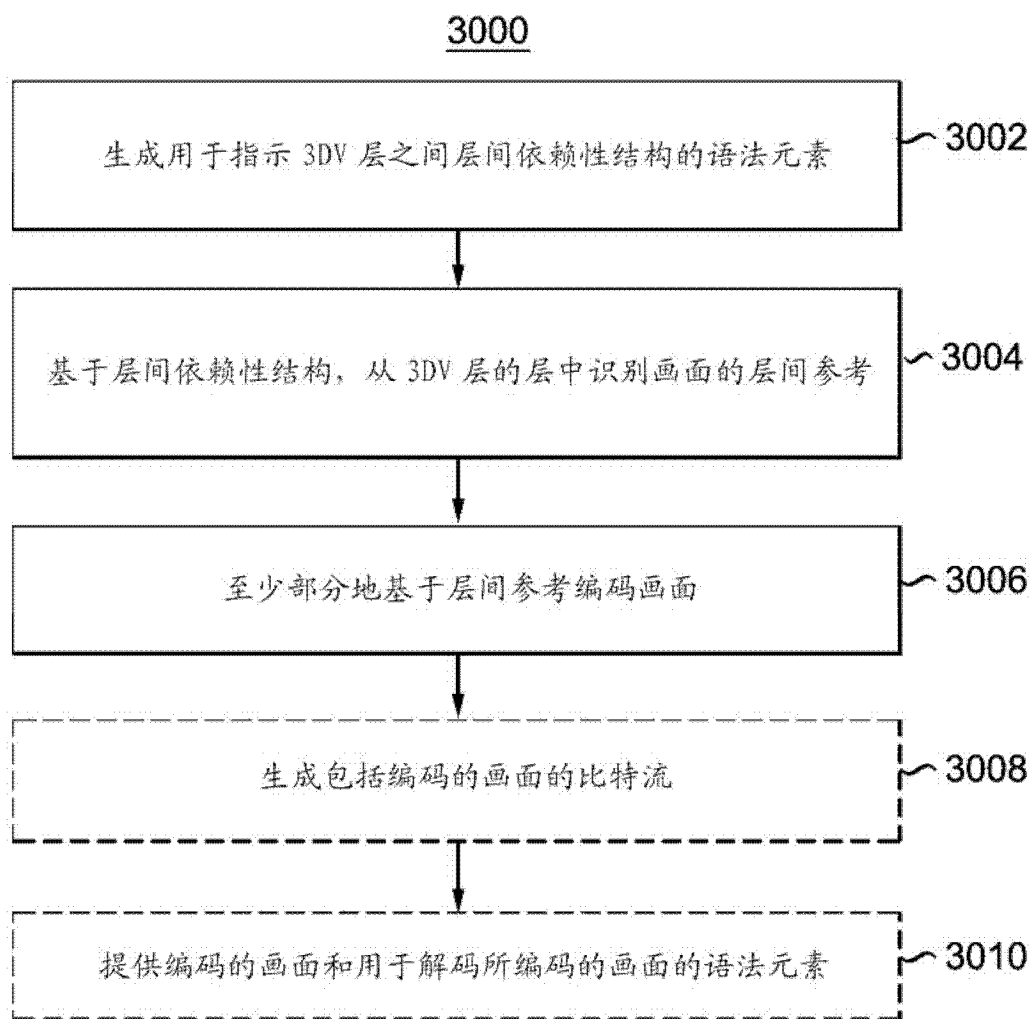


图 30

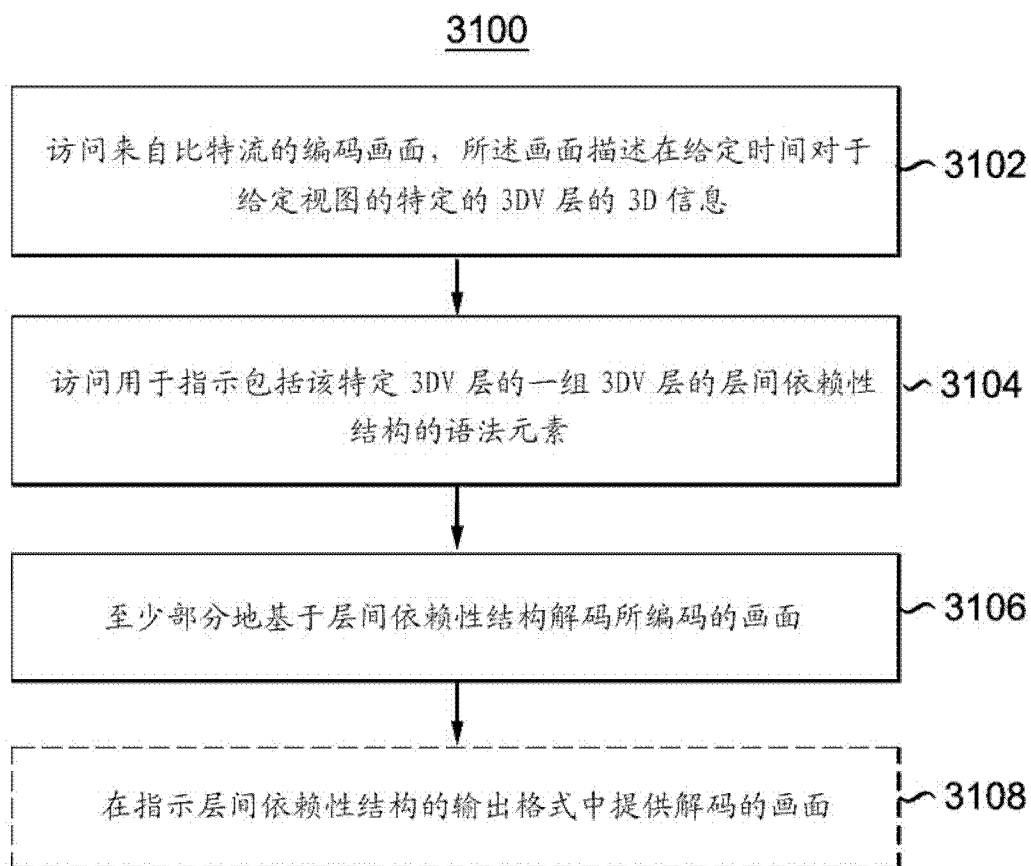


图 31

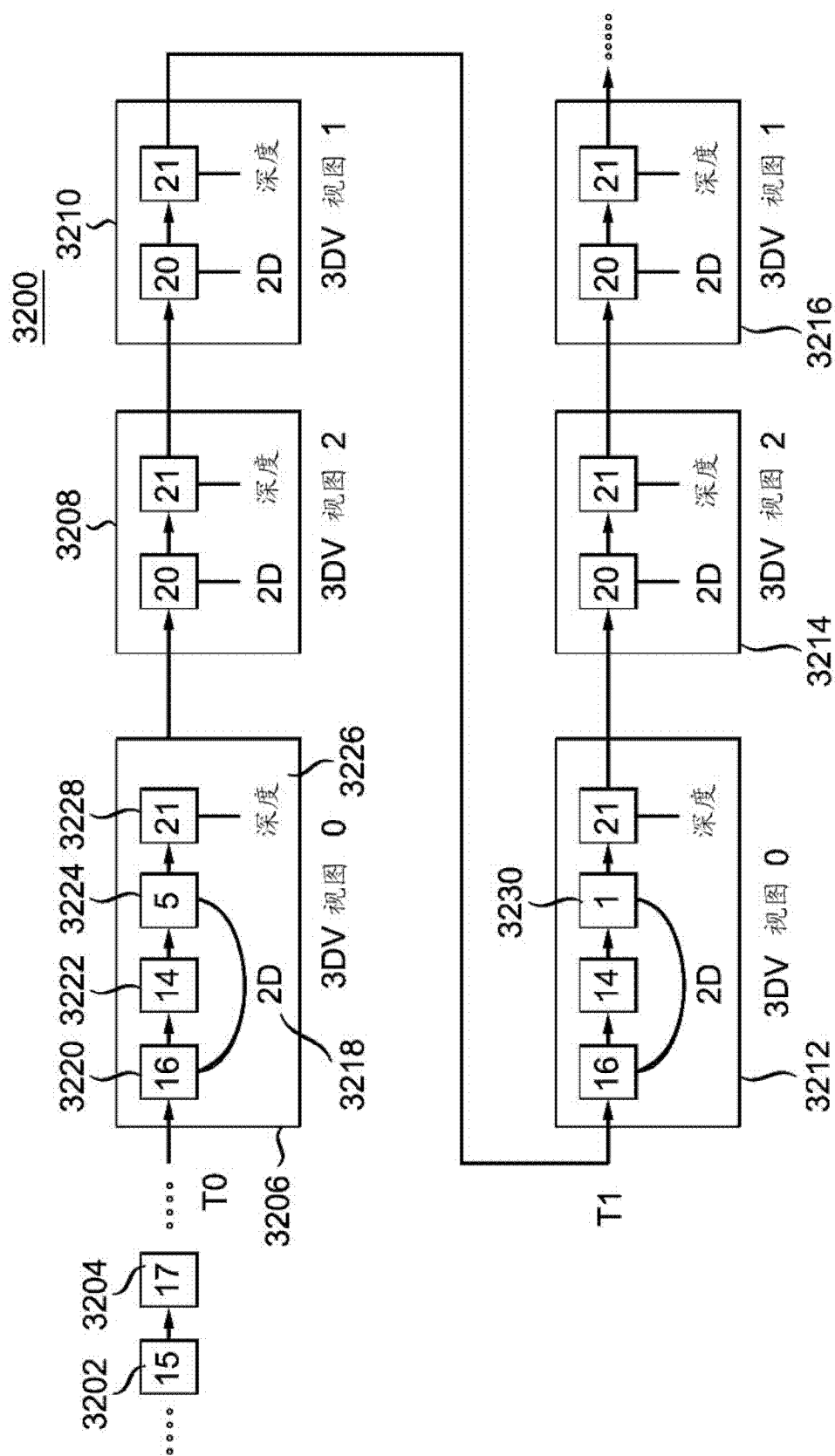


图 32

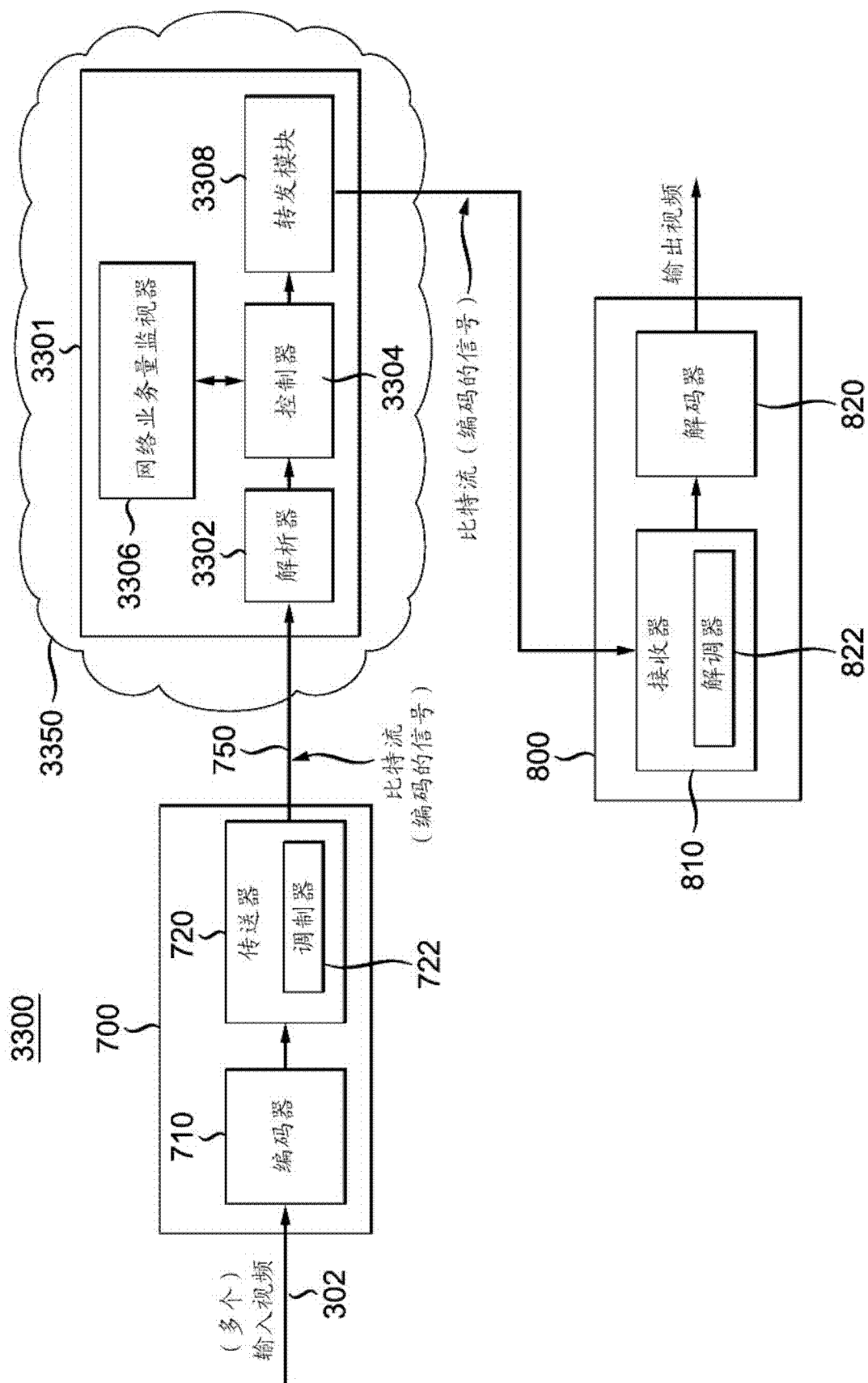


图 33

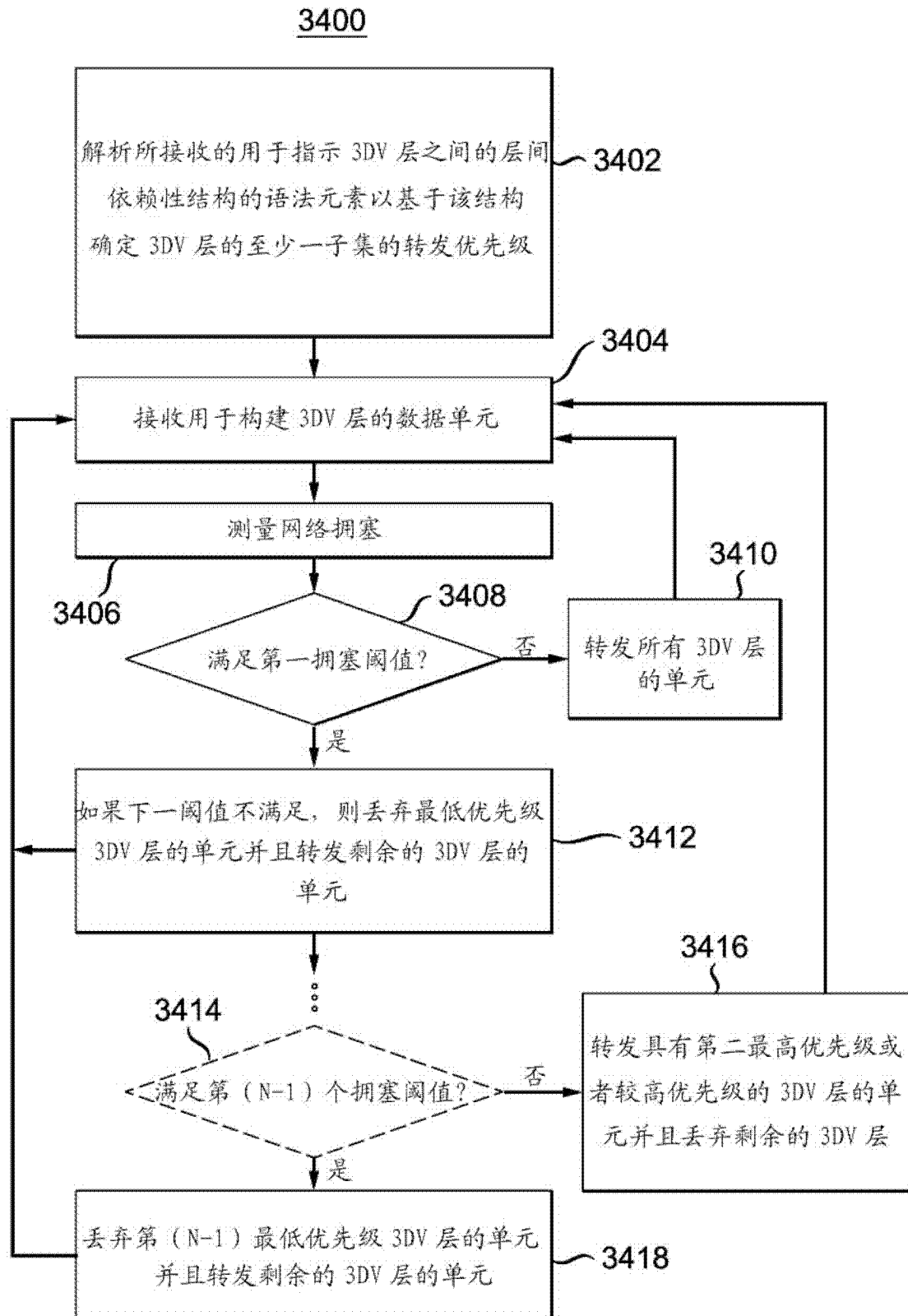


图 34