



(12)发明专利

(10)授权公告号 CN 103297771 B

(45)授权公告日 2017.03.01

(21)申请号 201310110516.7

H04N 19/467(2014.01)

(22)申请日 2009.04.23

(56)对比文件

(65)同一申请的已公布的文献号

US 2007291850 A1, 2007.12.20,

申请公布号 CN 103297771 A

US 2007030356 A1, 2007.02.08,

(43)申请公布日 2013.09.11

JP H09261653 A, 1997.10.03,

(30)优先权数据

ramanathan p et al. MULTI-HYPOTHESIS

61/125,520 2008.04.25 US

PREDICTION FOR DISPARITY-COMPENSATED

(62)分案原申请数据

LIGHT FIELD COMPRESSION.《Image

200980114390.2 2009.04.23

Processing, 2001. Proceedings. 2001

(73)专利权人 汤姆森许可贸易公司

International Conference》.2001,第2卷第

地址 法国布洛涅-比朗库尔

101-104页.

(续)

(72)发明人 田冬 普尔温·比贝哈斯·潘迪特

审查员 董翠翠

尹鹏

(74)专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 杨静

(51)Int.Cl.

H04N 19/597(2014.01)

H04N 19/50(2014.01)

H04N 19/61(2014.01)

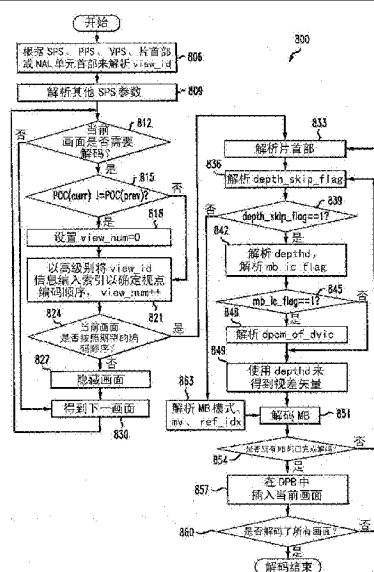
权利要求书5页 说明书23页 附图13页

(54)发明名称

使用基于深度信息的视差估计的多视点视频编码

(57)摘要

描述了多种实现方式。若干实现方式涉及使用深度信息来确定对应参考块的一个或多个编码模式。根据总体方面,对来自多视点系统中第一视点的第一视点图像的一部分进行编码。所述部分是第一视点图像的全部或部分。确定来自多视点系统中第二视点的第二视点图像的一部分与第一视点图像的所述部分相对应。所述确定基于第二视点深度值,所述第二视点深度值提供了第二视点图像的所述部分的深度信息。使用通过对第一视点图像的所述部分进行编码而得到的信息来对第二视点图像的所述部分进行编码。



[转续页]

[接上页]

(56)对比文件

WIEGAND T ET AL. Overview of the H.264/AVC video coding standard.《IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY》.2003,第13卷(第7期),560-576.

ZHU G ET AL. MVC inter-view skip mode with depth information.《 26. JVT MEETING (JOINT VIDEO TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), no. JVT-Z029》.2008,(第JVT-Z029期),全文.

1. 一种编码方法,包括:

对来自多视点系统中第一视点的第一视点图像的一部分进行编码,所述部分是所述第一视点图像的全部或部分;

确定来自多视点系统中第二视点的第二视点图像的一部分与第一视点图像的所述部分相对应,其中,所述确定基于第二视点深度值,所述第二视点深度值提供了第二视点图像的所述部分的深度或视差信息;

使用通过对第一视点图像的所述部分进行编码而得到的信息来对第二视点图像的所述部分进行编码;

针对第二视点图像的所述部分产生语法,其中,所产生的语法包括:(i)对于第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的指示,以及(ii)对于第二视点深度值的指示;以及

传输对于第二视点深度值的指示,

确定所述第二视点图像的一部分与第一视点图像的所述部分相对应包括使用卷曲处理,所述卷曲处理包括以下操作:

基于点在第二视点图像的所述部分中的位置,来确定所述点在真实空间中的位置,以及

基于所述点在真实空间中的位置,来确定所述点在第一视点图像的所述部分中的位置。

2. 根据权利要求1所述的方法,其中,第一视点图像的已编码部分、第二视点图像的已编码部分以及编码的第二视点深度值是根据支持多视点编码而不支持深度图编码的标准而编码的。

3. 根据权利要求1所述的方法,还包括:提供(i)第一视点图像的已编码部分、(ii)第二视点图像的已编码部分以及(iii)第二视点深度值,用于存储或传输。

4. 根据权利要求3所述的方法,其中,所述提供包括:将第一视点图像的已编码部分、第二视点图像的已编码部分以及第二视点深度值组装成结构化格式。

5. 根据权利要求1所述的方法,还包括:对第二视点深度值进行编码。

6. 根据权利要求1所述的方法,其中,第二视点深度值是针对给定的宏块或宏块的一部分的单个值。

7. 根据权利要求1所述的方法,还包括:确定第二视点图像的所述部分的视点差矢量,所述视点差矢量指示第二视点图像的所述部分的位置与第一视点图像的所述部分的位置之间的差异,所述视点差矢量是基于第二视点深度值来确定的。

8. 根据权利要求7所述的方法,其中:

确定视点差矢量还基于细化以使视点差矢量更适合于编码而不是合成。

9. 根据权利要求7所述的方法,还包括:

形成基于视点差矢量预测值的视点差矢量;以及

基于视点差矢量预测值来对视点差矢量进行编码。

10. 根据权利要求7所述的方法,还包括:

基于根据深度而得到的、针对第一视点图像的所述部分的视点差矢量,来形成视点差矢量预测值;以及

基于视点差矢量预测值来对视点差矢量进行编码。

11. 根据权利要求1所述的方法, 其中:

对第二视点图像的所述部分进行编码包括设置指示以下情况的标志: 将使用第一视点图像的所述部分的重构, 来重构第二视点图像的所述部分。

12. 根据权利要求1所述的方法, 其中:

对第二视点图像的所述部分进行编码包括: 使用通过对第一视点图像的所述部分进行编码而得到的运动矢量, 以及确定第二视点图像的所述部分与所述运动矢量从第二视点图像所指向的参考图像之间的残差。

13. 根据权利要求1所述的方法, 其中:

第一视点图像包括来自视频序列的画面, 以及

对第一视点图像的所述部分进行编码包括: 对画面的宏块进行编码。

14. 根据权利要求1所述的方法, 其中, 至少所述第二视点图像被包含在视频序列中, 所述视频序列由多个图像形成, 并且针对所述多个图像中的每个图像, 具有至少一个视点差矢量。

15. 根据权利要求1所述的方法, 其中, 至少所述第二视点图像被包含在视频序列中, 所述视频序列由多个图像形成, 并且针对所述多个图像中的每个图像, 具有多个视点差矢量。

16. 根据权利要求1所述的方法, 其中, 所述方法是在编码器中实现的。

17. 根据权利要求1所述的方法, 还包括:

将第一视点图像的所述部分的照度与第二视点图像的所述部分的照度相比较; 以及

基于比较结果来确定针对第二视点图像的所述部分的照度偏移。

18. 根据权利要求15所述的方法, 其中, 对于第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的指示包括标志, 所述标志指示了将使用第一视点图像的所述部分的重构, 来重构第二视点图像的所述部分, 作为第二视点图像的所述部分的重构。

19. 根据权利要求1所述的方法, 其中, 对于第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的指示包括标志, 所述标志指示了: 第二视点图像的所述部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来编码的。

20. 一种编码设备, 包括:

用于对来自多视点系统中第一视点的第一视点图像的一部分进行编码的装置, 所述部分是所述第一视点图像的全部或部分;

用于确定来自多视点系统中第二视点的第二视点图像的一部分与第一视点图像的所述部分相对应的装置, 其中, 所述确定基于第二视点深度值, 所述第二视点深度值提供了第二视点图像的所述部分的深度或视差信息;

用于使用通过对第一视点图像的所述部分进行编码而得到的信息来对第二视点图像的所述部分进行编码的装置;

用于针对第二视点图像的所述部分产生信息的装置, 其中, 所产生的信息包括: (i) 对于第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的指示, 以及(ii) 对于第二视点深度值的指示; 以及

用于传输对于第二视点深度值的指示的装置，

确定所述第二视点图像的一部分与第一视点图像的所述部分相对应包括使用卷曲处理，所述卷曲处理包括以下操作：

基于点在第二视点图像的所述部分中的位置，来确定所述点在真实空间中的位置，以及

基于所述点在真实空间中的位置，来确定所述点在第一视点图像的所述部分中的位置。

21. 一种编码设备，包括：

视差转换器，用于确定来自多视点系统中第二视点的第二视点图像的一部分与来自多视点系统中第一视点的第一视点图像的一部分相对应，第一视点图像的所述部分是第一视点图像的全部或部分，其中，所述确定基于第二视点深度值，所述第二视点深度值提供了第二视点图像的所述部分的深度信息；以及

编码单元，用于对第一视点图像的所述部分进行编码，用于使用通过对第一视点图像的所述部分进行编码而得到的信息来对第二视点图像的所述部分进行编码，用于针对第二视点图像的所述部分产生信息，以及用于传输对于第二视点深度值的指示，其中，所产生的信息包括：(i)对于第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的指示，以及(ii)对于第二视点深度值的指示；

其中确定所述第二视点图像的一部分与第一视点图像的所述部分相对应包括使用卷曲处理，所述卷曲处理包括以下操作：

基于点在第二视点图像的所述部分中的位置，来确定所述点在真实空间中的位置，以及

基于所述点在真实空间中的位置，来确定所述点在第一视点图像的所述部分中的位置。

22. 根据权利要求21所述的设备，还包括：输出单元，用于提供第一视点图像的已编码部分、第二视点图像的已编码部分、编码的第二视点深度值以及所产生的信息，用于存储或传输。

23. 一种解码方法，包括：

对来自多视点系统中第一视点的第一视点图像的一部分的编码进行解码，所述部分是第一视点图像的全部或部分；

存取语法，所述语法包括：(i)对于来自多视点系统中第二视点的第二视点图像的一部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来进行编码的指示，以及(ii)对于第二视点深度值的指示，所述第二视点深度值提供了第二视点图像的所述部分的深度或视差信息，其中所述深度或视差信息可用于确定第一视点图像的所述相应部分；

确定第二视点图像的所述部分与第一视点图像的所述部分相对应，并且确定第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的，其中，所述确定基于：对于第二视点深度值的指示，以及对于第二视点图像的所述部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来进行编码的指示；以及

基于所述确定，使用来自第一视点图像的解码部分的信息来对第二视点图像的所述部分的编码进行解码；

对第二视点图像的所述部分进行解码包括读取指示以下情况的标志：第二视点图像的所述部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来编码的。

24. 根据权利要求23所述的方法，还包括：

存取照度偏移，所述照度偏移指示第一视点图像的所述部分的照度与第二视点图像的所述部分的照度之差；以及

基于照度偏移来调整第二视点图像的所述部分的重构的照度。

25. 根据权利要求23所述的方法，其中：

对第二视点图像的所述部分进行解码包括：使用来自第一视点图像的解码部分的运动矢量，以及确定第二视点图像的所述部分与所述运动矢量从第二视点图像所指向的参考图像之间的残差。

26. 根据权利要求23所述的方法，还包括：处理第二视点深度值。

27. 根据权利要求26所述的方法，其中，处理第二视点深度值包括以下操作中的一个或多个：根据接收到的信号来解析第二视点深度值，或对第二视点深度值进行解码。

28. 根据权利要求23所述的方法，其中，以根据支持多视点编码而不支持深度图编码的标准而结构化的格式，来接收第一视点图像的编码部分以及第二视点图像的编码部分。

29. 根据权利要求23所述的方法，其中，第二视点深度值是针对给定的宏块或宏块的一部分的单个值。

30. 根据权利要求23所述的方法，还包括：确定第二视点图像的所述部分的视点差矢量，所述视点差矢量指示第二视点图像的所述部分与第一视点图像的所述部分的位置之差，所述视点差矢量是基于第二视点深度值来确定的。

31. 根据权利要求23所述的方法，还包括：

形成第二视点图像的所述部分的视点差矢量预测值，所述视点差矢量预测值指示第二视点图像的所述部分与第一视点图像的所述部分的位置之差，所述视点差矢量预测值是基于第二视点深度值来确定的；以及

基于视点差矢量预测值来对视点差矢量进行解码。

32. 根据权利要求23所述的方法，包括：

基于根据深度而得到的、针对第一视点图像的所述部分的视点差矢量，来形成视点差矢量预测值；以及

基于视点差矢量预测值来对视点差矢量进行解码。

33. 根据权利要求23所述的方法，其中，所述方法是在解码器中实现的。

34. 根据权利要求23所述的方法，其中，所述方法是在编码器中作为编码过程的一部分而实现的。

35. 根据权利要求23所述的方法，其中，确定所述第二视点图像的一部分与第一视点图像的所述部分相对应包括使用卷曲处理，所述卷曲处理包括以下操作：

基于点在第二视点图像的所述部分中的位置，来确定所述点在真实空间中的位置，以及

基于所述点在真实空间中的位置，来确定所述点在第一视点图像的所述部分中的位置。

36. 一种解码设备，包括：

用于对来自多视点系统中第一视点的第一视点图像的一部分的编码进行解码的装置，所述部分是第一视点图像的全部或部分；

用于存取语法的装置，所述语法包括：(i)对于来自多视点系统中第二视点的第二视点图像的一部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来进行编码的指示，以及(ii)对于第二视点深度值的指示，所述第二视点深度值提供了第二视点图像的所述部分的深度或视差信息，其中所述深度或视差信息可用于确定第一视点图像的所述相应部分；

用于确定第二视点图像的所述部分与第一视点图像的所述部分相对应，并且确定第二视点图像的所述部分是使用通过对第一视点图像的所述部分进行编码而得到的信息来进行编码的装置，其中，所述确定基于：对于第二视点深度值的指示，以及对于第二视点图像的所述部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来进行编码的指示；以及

用于基于所述确定使用来自第一视点图像的解码部分的信息来对第二视点图像的所述部分的编码进行解码的装置；

其中，对第二视点图像的所述部分进行解码包括读取指示以下情况的标志：第二视点图像的所述部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来编码的。

37.一种解码设备，包括：

解码单元，用于对语法进行解码，所述语法包括(i)编码指示和(ii)深度指示，所述编码指示指示了来自多视点系统中第二视点的第二视点图像的一部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来进行编码的，所述深度指示指示了第二视点深度值，所述第二视点深度值提供了第二视点图像的所述部分的深度或视差信息，其中所述深度或视差信息可用于确定第一视点图像的所述相应部分的位置；以及

视差转换器，用于确定第一视点图像的所述相应部分的位置，第一视点图像的所述部分是所述第一视点图像的全部或部分，其中，所述确定基于编码指示和深度指示；

其中，解码单元还用于对第一视点图像的所述相应部分的编码进行解码，以及用于基于所述确定使用来自第一视点图像的所述相应部分的解码部分的信息来对第二视点图像的所述部分的编码进行解码；

其中，对第二视点图像的所述部分进行解码包括读取指示以下情况的标志：第二视点图像的所述部分是使用通过对第一视点图像的相应部分进行编码而得到的信息来编码的。

使用基于深度信息的视差估计的多视点视频编码

[0001] 本申请是申请日为2009年4月23日的中国专利申请200980114390.2(“使用基于深度信息的视差估计的多视点视频编码”)的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请要求在2008年4月25日提交的题为“Inter-view Skip Mode with Depth”的序列号为61/125,520的美国临时专利申请的申请日的权益,其内容一并在此作为参考。

技术领域

[0004] 描述了涉及编码系统的实现方式。多种具体实现方式涉及使用深度的视点间跳过模式。

背景技术

[0005] 存在使用来自另一宏块的信息对当前宏块进行解码的一种或多种编码工具。一种编码工具是使用深度信息对当前宏块进行解码的视点间跳过模式。本申请中指出了视点间跳过编码工具的一个或多个缺点。

发明内容

[0006] 根据总体方面,对来自多视点系统中第一视点的第二视点图像的一部分进行编码。该部分是第二视点图像的全部或一部分。确定来自多视点系统中第二视点的第二视点图像的一部分与第二视点图像的所述部分相对应。所述确定基于为第二视点图像的所述部分提供深度信息的第二视点深度值。使用通过对第二视点图像的所述部分的编码而得到的信息来对第二视点图像的所述部分进行编码。

[0007] 根据另一总体方面,视频信号或视频信号结构包括以下区段。第二视点图像区段包括来自多视点系统中第二视点的第二视点图像的一部分的编码信息。所述第二视点图像的所述部分包括第二视点图像的全部或一部分。第二视点图像区段包括:与第二视点图像的所述部分相对应的、来自多视点系统中第二视点的第二视点图像的一部分。第二视点深度区段包括为第二视点图像的所述部分提供深度信息的深度值。第二视点图像区段、第二视点图像区段和第二视点图像深度区段是根据支持多视点编码而不支持深度图编码的标准来格式化的。

[0008] 根据另一总体方面,对来自多视点系统中第二视点的第二视点图像的一部分的编码进行解码。该部分是第二视点图像的全部或一部分。确定来自多视点系统中第二视点的第二视点图像的一部分与第二视点图像的所述部分相对应。所述确定基于为第二视点图像的所述部分提供深度信息的第二视点深度值。使用来自第二视点图像的解码部分的信息来对第二视点图像的所述部分的编码进行解码。

[0009] 在附图和以下描述中产生了一种或多种实现方式的细节。尽管以具体方式描述,然而应当清楚可以以多种方式来配置或体现这些实现方式。例如,实现方式可以以方法的形式来执行,体现为设备,例如被配置为执行一组操作的设备或者用于执行一组操作的设

备存储结构,或者体现为信号。通过以下结合附图和权利要求考虑的详细描述,其他方面和特征将变得显而易见。

附图说明

- [0010] 图1是编码器的实现方式的图。
- [0011] 图2是解码器的实现方式的图。
- [0012] 图3是视频发送系统的实现方式的图。
- [0013] 图4是视频接收系统的实现方式的图。
- [0014] 图5是视频处理设备的实现方式的图。
- [0015] 图6是对于一种实现方式标识了当前宏块的相邻宏块的图。
- [0016] 图7是第一编码过程的实现方式的图。
- [0017] 图8是第一解码过程的实现方式的图。
- [0018] 图9是运动匹配的实现方式的图。
- [0019] 图10是第二编码过程的实现方式的图。
- [0020] 图11是第二解码过程的实现方式的图。
- [0021] 图12是第三编码过程的实现方式的图。
- [0022] 图13是第三解码过程的实现方式的图。

具体实施方式

[0023] 视点间跳过是一种使用深度信息来对当前宏块进行解码的编码工具。发明人确定视点间跳过的缺点是需要或假定深度图在解码器处可用。这在需要深度图来实现视点合成的3DV架构下是可能的。在多视点视频编码(MVC)的情况下,没有指定深度图的标准传输。因此,不能使用需要深度图的模式。

[0024] 在至少一个实现方式中,提出了一种使用利用深度的视点间跳过模式的架构。在至少一个实现方式中,提出了使用视点间跳过作为用于多视点编码的编码模式。在本文所描述的至少一个实现方式中,提出了通过发送对应宏块的深度而不是例如单独地发送深度图来实现这种方法的修改。从原始视频序列和对应摄像机参数产生的深度信息不仅可以有益于解码器端的视点合成,还可以有益于编码器处的编码效率。借助于深度信息,对两个视点间画面之间对应关系的估计可以比传统方法更精确。在至少一个实现方式中,引入了新的“跳过模式”,称作“深度跳过模式”。

[0025] 因此,至少一些实现方式解决的至少一个问题是,使用利用深度的视点间跳过模式来高效地编码多视点视频序列。多视点视频序列是由从不同视点捕获某场景的两个或多个视频序列组成的集合。

[0026] 此外,如上所述,在多视点视频编码(MVC)的情况下,没有指定深度图的标准传输。因此,不可以使用需要深度图的模式。本文描述的至少一个实现方式通过在需要时明确地发送深度信息(通常与深度图不同)来解决该问题。注意,“深度信息”是与深度有关的各种信息的统称。一种类型的深度信息是“深度图”,所述深度图通常是指逐像素深度图像。其他类型的深度信息可以包括例如针对每个编码块而不是每个编码像素使用单个深度值。

[0027] 图1示出了示例性编码器100,根据本原理的实施例,可以对所述编码器100应用本

发明。编码器100包括组合器105,组合器105具有以信号通信的方式与变换器110的输入相连的输出。变换器110的输出以信号通信的方式与量化器115的输入相连。量化器115的输出以信号通信的方式与熵编码器120的输入和逆量化器125的输入相连。逆量化器125的输出以信号通信的方式与逆变换器130的输入相邻。逆变换器130的输出以信号通信的方式与组合器135的第一非反向输入相连。组合器135的输出以信号通信的方式与帧内预测器145的输入和解块滤波器150的输入相连。解块滤波器150去除例如沿着宏块边缘的伪像。解块滤波器150的第一输出以信号通信的方式与参考画面存储装置155的输入(用于时间预测)和参考画面存储装置160的第一输入(用于视点间预测)相连。参考画面存储装置155的输出以信号通信的方式与运动补偿器175的第一输入和运动估计器180的第一输入相连。运动估计器180的输出以信号通信的方式与运动补偿器175的第二输入相连。参考画面存储装置160的第一输出以信号通信的方式与视差估计器170的第一输入相连。参考画面存储装置160的第二输出以信号通信的方式与视差补偿器165的第一输入相连。视差估计器170的输出以信号通信的方式与开关134的第一输入相连。(基于深度的)视差转换器132的输出以信号通信的方式与开关134的第二输入相连。视差选择器131的第一输出以信号通信的方式与开关134的控制输入相连,以选择开关的第一或第二输入。开关134的输出以信号通信的方式与视差补偿器165的第二输入相连。

[0028] 熵解码器120的输出、模式判定模块115的第一输出以及视差选择器131的输出各自均可用作编码器100的相应输出,以输出比特流。视差选择器131还可以用于输出多种形式的深度信息。可以存储或发送这样的深度信息,例如这样的深度信息可以用于允许解码操作确定编码视点的全部或一部分的视差矢量。

[0029] 组合器105的第一输入、运动补偿器175的第三输入、运动估计器180的第二输入、视差估计器170的第二输入以及视差转换器132的第一输入各自都可用作编码器的相应输入,以接收视点i的画面。视差转换器132的第二输入可用作到编码器100的输入,用于接收带外深度(out-of-band-depth)。

[0030] 运动补偿器175的输出以信号通信的方式与开关185的第一输入相连。视差补偿器165的输出以信号通信的方式与开关185的第二输入相连。帧内预测器145的输出以信号通信的方式与开关185的第三输入相连。开关185的输出以信号通信的方式与组合器105的反向输入和组合器135的第二非反向输入相连。模式判定模块115的输出确定将哪个输入提供至开关185。

[0031] 图1的一部分也可以分别地或集体地称作编码器、编码单元或存取单元(例如,块110、115和120)。类似地,例如块125、130、135和150可以分别地或集体地称作解码器或解码单元。

[0032] 图2示出了示例性解码器200,根据本原理的实施例,本原理可以应用于所述解码器200。解码器200包括熵解码器205,熵解码器205具有以信号通信的方式与逆量化器210的输入相连的输出。逆量化器210的输出以信号通信的方式与逆变换器215相连。逆变换器215的输出以信号通信的方式与组合器220的第一非反向输入相连。组合器220的输出以信号通信的方式与解块滤波器225的输入和帧内预测器230的输入相连。解块滤波器225的第一输出以信号通信的方式与参考画面存储装置240的输入(用于时间预测)和参考画面存储装置245的第一输入(用于视点间预测)相连。参考画面存储装置240的输出以信号通信的方式与运动补偿器235的第一输入相连。参考画面存储装置245的输出以信号通信的方式与视差补偿器250的第一输入相连。

[0033] 比特流接收机201的输出以信号通信的方式与比特流解析器202的输入相连。比特流解析器202的第一输出(用于提供残差比特流)以信号通信的方式与熵解码器205的输入相连。比特流解析器202的第二输出(用于提供控制语法以控制开关255选择哪个输入)以信号通信的方式与模式选择器222的输入相连。比特流解析器202的第三输出(用于提供运动矢量)以信号通信的方式与运动补偿器235的第二输入相连。比特流解析器202的第四输出(用于提供照度偏移)以信号通信的方式与视差补偿器250的第二输入相连。比特流解析器202的第五输出(用于提供视差控制语法)以信号通信的方式与视差选择器204的输入相连。比特流解析器202的第六输出(用于提供视差矢量)以信号通信的方式与开关211的第一输入相连。比特流解析器202的第七输出(用于提供带内深度)以信号通信的方式与视差转换器203的第一输入相连。视差转换器203的输出以信号通信的方式与开关211的第二输入相连。视差选择器204的输出确定将哪个输入提供至开关211。开关的输出以信号通信的方式与视差补偿器250的第三输入相连。将意识到,照度偏移是可选的输入,根据实现方式可以使用或不使用该照度偏移。

[0034] 开关255的输出以信号通信的方式与组合器220的第二非反向输入相连。开关255的第一输入以信号通信的方式与视差补偿器250的输出相连。开关255的第二输入以信号通信的方式与运动补偿器235的输出相连。开关255的第三输入以信号通信的方式与帧内预测器230的输出相连。模式模块222的输出以信号通信的方式与开关255相连,以控制开关255选择哪个输入。解块滤波器225的第二输出可用作解码器200的输出。视差转换器203的第二输入可用作解码器200的输入,以接收带外深度。

[0035] 图2的一部分也可以分别地或集体地称作存取单元,例如,比特流解析器202和使得可以存取特定数据或信息片的任何其他块。类似地,例如块205、210、215、220和225可以分别地或集体地称作解码器或解码单元。

[0036] 图3示出了示例性视频发送系统300,根据本原理的实现方式,本原理可以应用于所述视频发送系统300。视频发送系统300可以是例如用于使用诸如卫星、线缆、电话线路、或地面广播等多种介质中的任何介质来发送信号的头端或发送系统。可以在因特网或其他网络上提供发送。

[0037] 视频发送系统300可以产生和传递使用利用深度的视点间跳过模式来编码的视频内容。这是通过产生编码信号来实现的,所述编码信号包括深度信息或能够用于在可以例如具有解码器的接收机端合成深度信息的信息。

[0038] 视频发送系统300包括编码器310以及能够发送编码信号的发射机320。编码器310接收视频信息并使用利用深度的视点间跳过模式从视频信息中产生编码信号。编码器310可以是例如以上详细描述编码器300。编码器310可以包括子模块,所述子模块包括例如组装单元,所述组装单元用于接收多个信息片并将所述多个信息片组装成用于存储或发送的结构格式。不同的信息片可以包括例如编码或非编码的视频、编码或非编码的深度信息、以及编码或非编码的元素(例如,运动矢量、编码模式指示符、和语法元素)。

[0039] 发射机320例如可以适于发送节目信号,所述节目信号具有表示编码画面和/或其相关信息的一个或多个比特流。典型的发射机执行例如以下功能中的一个或多个功能:提供纠错编码、对信号中的数据进行交织、使信号中的能量随机化、以及将信号调制到一个或多个载波上。发射机可以包括天线(未示出)或与天线接口连接。相应地,发射机320的实现方式可以包括但不限于调制器。

[0040] 图4示出了示例性视频接收系统400,根据本原理的实施例,本原理可以应用于所述视频接收系统400。视频接收系统400可以被配置为通过例如卫星、线缆、电话线路或地面广播等多种介质来接收信号。可以通过因特网或其他网络来接收信号。

[0041] 视频接收系统400可以是例如蜂窝电话、计算机、机顶盒、电视或接收编码视频并提供例如解码视频以向用户显示或用于存储的其他设备。因此,视频接收系统400可以将其输出提供至例如电视屏幕、计算机监视器、计算机(以供存储、处理或显示)或其他的存储、处理或显示设备。

[0042] 视频接收系统400可以接收和处理包括视频信息的视频内容。视频接收系统400包括:例如本申请的实现方式中描述的能够接收编码信号的接收机410,以及能够对接收到的信号进行解码的解码器420。

[0043] 接收机410可以例如适于接收节目信号,所述节目信号具有表示编码画面的多个比特流。典型的接收机执行例如以下功能中的一个或多个功能:接收调制和编码的数据信号、对来自一个或多个载波的数据信号进行解调、使信号中的能量去随机化、将信号中的数据去交织、以及对信号进行纠错解码。接收机410可以包括天线(未示出)或与天线接口连接。接收机410的实现方式可以包括或限于解调器。

[0044] 解码器420输出包括视频信息和深度信息的视频信号。解码器420例如可以是以上描述的解码器400。

[0045] 图5示出了示例性视频处理设备500,根据本原理的实施例,本原理可以应用于所述视频处理设备500。视频处理设备500可以例如是机顶盒或接收编码视频并提供例如解码视频以向用户显示或存储的其他设备。因此,视频处理设备500可以将其输出提供至电视、计算机监视器或计算机或其他处理设备。

[0046] 视频处理设备500包括前端(FE)装置505和解码器510。前端装置505可以是例如接收机,所述接收机适于接收具有表示编码画面的多个比特流的节目信号,以及适于选择一个或多个比特流以根据所述多个比特流进行解码。典型的接收机执行例如以下功能中的一个或多个功能:接收调制和编码数据信号、解调数据信号、对数据信号的一个或多个编码(例如,信道编码和/或源编码)进行解码、和/或对数据信号进行纠错。前端装置505可以接收来自例如天线(未示出)的节目信号。前端装置505将接收到的数据信号提供至解码器510。

[0047] 解码器510接收数据信号520。数据信号520可以包括例如一个或多个与高级视频编码(AVC)、可缩放视频编码(SVC)或多视点视频编码(MVC)兼容的流。

[0048] AVC具体是指国际标准化组织/国际电子技术委员会(ISO/IEC)运动图像专家组-4(MPEG-4)部分10高级视频编码(AVC)标准/国际电信联盟-电信部分(ITU-T)H.264推荐(下文中,“H.264/MPEG-4AVC标准”或其变体,如“AVC标准”,或简称“AVC”)。

[0049] MVC具体是指AVC标准的多视点视频编码(“MVC”)扩展(附录H),称作H.264/MPEG-4AVC,MVC扩展(“MVC”扩展或简称“MVC”)。

[0050] SVC具体是指AVC标准的可缩放视频编码(“SVC”)扩展(附录G),称作H.264/MPEG-4AVC,SVC扩展(“SVC”扩展或简称“SVC”)。

[0051] 解码器510对接收信号520的全部或部分进行解码,并提供解码视频信号530作为输出。将解码视频530提供至选择器550。设备500还包括接收用户输入570的用户解码560。用户界面560基于用户输入570来向选择器550提供画面选择信号580。画面选择信号580和

用户输入570指示用户希望显示以下项目中的哪个：多个画面、序列、可缩放版本、视图或可用解码数据的其他选择。选择器550提供所选的画面作为输出590。选择器550使用画面选择信息580来选择提供解码视频530的哪个画面作为输出590。

[0052] 在不同实现方式中，选择器550包括用户界面560，在其他实现方式中，由于选择器550直接接收用户输入570而不执行单独的接口功能，所以选择器550可以例如以软件形式实现或实现为集成电路。在一个实现方式中，选择器550与解码器510合并，在另一个实现方式中，解码器510、选择器550和用户界面560合并在一起。

[0053] 在一个应用中，前端505接收多种电视节目的广播并选择一个电视节目来处理。一个节目的选择基于要观看的期望频道的用户输入。尽管图5并未示出向前端装置505的用户输入，然而前端装置505接收用户输入570。前端505接收广播并通过对广播谱的相关部分进行解调以及对解调节目的任何外部编码进行解码，来处理期望的节目。前端505将解码节目提供至解码器510。解码器510是包括装置560和550的集成单元。因此解码器510接收用户输入，所述用户输入是由用户提供的对要在节目中观看的期望视点的指示。解码器510对所选的视点以及来自其他视点的任何所需参考画面进行解码，并提供解码视点590以显示在电视（未示出）上。

[0054] 继续上述应用，用户可能希望切换所显示的视点，然后可以向解码器510提供新的输入。在接收到来自用户的“视点改变”之后，解码器510对旧视点和新视点以及在旧视点与新视点之间的任何视点进行解码。即，解码器对从物理上位于捕获旧视点的摄像机与捕获新视点的摄像机之间的摄像机得到的任何视点进行解码。前端装置505还接收标识旧视点、新视点以及两者之间视点的信息。该信息可以例如由具有与视点位置有关的信息的控制器（图5中未示出）或者解码器510来提供。其他实现方式可以使用具有控制器的前端装置，所述控制器与前端装置相集成。

[0055] 解码器510提供所有这些解码视点作为输出590。后处理器（图5中未示出）在视点之间进行内插，以提供从旧视点到新视点的平滑过渡，并向用户显示这种过渡。在过渡到新视点之后，后处理器通知（通过一个或多个通信链路，未示出）解码器510和前端装置505只需要新视点。此后，解码器510仅提供新视点作为输入590。

[0056] 系统500可以用于接收图像序列的多个视点，提供用于显示的单个视点，以及以平滑的方式在多个视点之间切换。平滑方式可以包括在视点之间进行内插以移到另一视点。此外，系统500可以允许用户旋转对象或场景，或允许用户看到对象或场景的三维表示。对象的旋转例如可以对应于：从一个视点移到另一视点，并且在视点之间进行内插，以得到视点之间的平滑过渡或仅得到三维表示。即，用户可以“选择”内插的视点作为要显示的“视点”。

[0057] 使用深度图来寻找视点间对应关系

[0058] 假定点 $M(x, y, z)$ （ z 是得自于深度图）以坐标 (u_1, v_1) 投影在摄像机平面1上，以坐标 (u_2, v_2) 投影在摄像机平面2上。点 (x, y, z) 来自与摄像机设置无关的真实坐标系，并且标识了3D点的位置。点 (u_1, v_1) 是3D点 (x, y, z) 投影到视点1的摄像机坐标系中的投影坐标。类似地，点 (u_2, v_2) 是3D点 (x, y, z) 投影到视点2的摄像机坐标系中的投影坐标。

[0059] 如下从摄像机参数 K 、 R 和 T 得出点矩阵 P ：

[0060] $P = K \cdot [R|T]$ (1)

[0061] 其中 K 是本征矩阵， R 是旋转矩阵， T 是变换矢量，等式1指示矩阵 R 被矩阵 T 右合并并且合并结果乘以矩阵 K 。

[0062] 对于视点1和视点2,根据等式(2)和等式(3),视点2中的像素(u2,v2)将在视点1中找到其对应像素(u1,v1)。这里P1和P2如下表示视点1和视点2的投影矩阵:

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = P_1 \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = P_2 \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3)$$

[0065] 每个摄像机具有唯一的投影矩阵P,如以上3×4矩阵所示。通过将投影矩阵左乘以全局坐标,可以得到摄像机坐标系中的坐标。每个摄像机产生其自己的等式集合。

[0066] 视差计算和深度跳过模式

[0067] 尽管以上计算是基于像素的,然而至少一种实际实现方式仅需要整个宏块的左上方像素的对应像素作为视差矢量。对于左上方像素是(u1,v1)的当前宏块,如果该像素的视点间对应关系是像素(u2,v2),则深度跳过模式的视差矢量(DV)是:DV=(u2-u1,v2-v1)。该模式的参考画面被设置为参考列表0中的第一视点间画面。不需要发送残差。

[0068] 在编码器或解码器处确定DV的过程可以典型地包括:将(u1,v1)用于画面中的点,使用与(u1,v1)相关联的深度值来确定画面中点的位置(x,y,z)。使用等式(2)来实现这一操作。然后在等式3中使用(x,y,z)来确定该点的(u2,v2)。该过程称作“卷曲(warping)”并且基于深度值,所述深度值可以被编码或在外部独立发送或者作为深度图的一部分来发送。使用卷曲过程,对于视点1中的每个像素(u1,v1),可以典型地识别视点2中的对应像素(u2,v2)。然后将DV设置为(u2-u1,v2-v1)。DV可以指向参考画面中的对应块。如果对应块与当前块足够类似,则如下所述,可以使用“深度跳过”模式通过拷贝来自对应块的像素来编码当前块。

[0069] 如本文所示,深度信息可以用于产生视差矢量。

[0070] 深度跳过模式下的照度补偿

[0071] 对于传统跳过模式,参考画面来自相同的视点,因此没有照度变化。然而,当参考画面来自其他视点时,可以考虑照度补偿(IC)。因此,如果启用照度补偿工具,则应当发送深度跳过模式下照度补偿的补偿偏移。如现有技术已知的,可以以多种方式来计算该偏移。

[0072] 在带外发送深度数据的实施例0

[0073] 表1A示出了对来自MVC标准的现有语法结构的修改。表1A的语法可以用于示意在宏块级对深度跳过模式的使用。还以照度补偿偏移值的形式示意了照度补偿。修改部分一

般由斜体字示出。注意,令macroblock_layer_mvc_extension()替换从slice_data()中调用的macroblock_layer()。

[0074] 表1A

macroblock_layer_mvc_extension () {	C	Descriptor
if (anchor_pic_flag && slice_type == P) {		
depth_skip_flag	2	u(1) ae(v)
} else {		
depth_skip_flag = 0		
}		
if (depth_skip_flag) {		
if (ic_enable) {		
mb_ic_flag	2	u(1) ae(v)
if (mb_ic_flag)		
dpcm_of_dvic	2	ue(v) ae(v)
}		
} else {		
mb_type	2	ue(v) ae(v)
if (mb_type == I_PCM) {		
while (!byte_aligned ())		
pcm_alignment_zero_bit	3	l(1)
for (i = 0; i < 256; i++)		
pcm_sample_luma[i]	3	U(v)
for (i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i]	3	u(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 1		
if (mb_type != I_NxN && MbPartPredMode(mb_type, 0) != Intra_16x16 && NumMbPart(mb_type) == 4) {		
sub_mb_pred(mb_type)	2	
for (mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if (sub_mb_type[mbPartIdx] != B_Direct_8x8) {		
if (NumSubMbPart(sub_mb_type[mbPartIdx]) > 1)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else if (!direct_8x8_inference_flag)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else {		
if (transform_8x8_mode_flag && mb_type == I_NxN)		
transform_size_8x8_flag	2	u(1) ae(v)
mb_pred(mb_type)	2	
}		
if (MbPartPredMode(mb_type, 0) != Intra_16x16) {		
coded_block_pattern	2	me(v) ae(v)
if (CodedBlockPatternLuma > 0 && transform_8x8_mode_flag && mb_type != I_NxN && noSubMbPartSizeLessThan8x8Flag && (mb_type != B_Direct_16x16 direct_8x8_inference_flag))		
transform_size_8x8_flag	2	u(1) ae(v)
}		
if (CodedBlockPatternLuma > 0 CodedBlockPatternChroma > 0 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
mb_qp_delta	2	se(v) ae(v)

[0075]

[0076]	<code>residual(0, 15)</code>	<code>3 4</code>	
	<code>}</code>		
	<code>}</code>		
	<code>}</code>		

[0077] depth_skip_flag等于1指示编码宏块处于深度跳过模式。depth_skip_flag等于0指示编码宏块没有处于深度跳过模式。

[0078] ic_enable等于1指示对于当前片启用照度补偿。ic_enable等于0指示不针对当前片使用照度补偿。

[0079] mb_ic_flag等于1指示针对当前宏块使用照度补偿。mb_ic_flag等于0指示不针对当前宏块使用照度补偿。mb_ic_flag的默认值是零。

[0080] dpcm_of_dvdc指定了要用于当前宏块的照度补偿偏移量。

[0081] 在至少一种实现方式中,提出改进深度跳过模式的性能。由于该模式假定深度数据在解码器处可用,所以该方法所需的总比特率是比特率(视频)+比特率(深度)。如果深度数据仅被编码为支持深度跳过模式,则在编码优化期间计数深度数据的附加比特率。此外,多视点视频编码标准并不支持将深度信号与视频信号一起发送。

[0082] 为了至少部分解决这两个问题,提出了一种将深度信息与视频信号一起发送的解决方案。

[0083] 在带内发送深度数据的实施例1

[0084] 假定深度数据针对每个视点在编码器处可用。如以上过程中提到的,对于一种实现方式,深度跳过模式需要被编码为深度跳过模式的宏块的左上方像素的深度值。其他实现方式可以使用例如右下方像素或块中所有像素的平均。在深度跳过模式下,该深度值是从与当前视点相对应的深度信号的对应像素得到的。

[0085] 可以基于速率失真优化来判定选择宏块。因此,不是所有宏块都将被选择为跳过模式。除此之外,还将针对整个画面或所有画面来发送深度数据。这在不需要深度数据用于任何其他用途(例如,呈现用途等等)时,导致对整个多视点系统的高比特率需求。

[0086] 因此,提出了一种显著降低这种高比特率需求的方法。由于仅针对特定比例的宏块选择深度跳过模式,所以不需要发送所有像素或所有画面的深度值。因此,提出仅发送使用深度跳过模式的宏块的深度值。

[0087] 表1B

[0088]

	C	Descriptor
macroblock_layer_mvc_extension () {		
if (anchor_pic_flag && slice_type == P) {		
depth_skip_flag	2	u(1) ae(v)
} else {		
depth_skip_flag = 0		
}		
if (depth_skip_flag) {		
depthd	2	se(v) ae(v)
if (ic_enable) {		
mb_ic_flag	2	u(1) ae(v)
if (mb_ic_flag)		
dpcm_of_dvlc	2	ue(v) ae(v)
}		
} else {		
mb_type	2	Ue(v) ae(v)
if (mb_type == I_PCM) {		
While (!byte_aligned ())		
pcm_alignment_zero_bit	3	f(1)
for (i = 0; i < 256; i++)		
pcm_sample_luma[i]	3	u(v)
for (i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i]	3	u(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 1		

[0089]	if(mb_type != I_NxN && MbPartPredMode(mb_type, 0) != Intra_16x16 && NumMbPart(mb_type) == 4) {		
	sub_mb_pred(mb_type)	2	
	for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
	if(sub_mb_type[mbPartIdx] != B_Direct_8x8) {		
	if(NumSubMbPart(sub_mb_type[mbPartIdx]) > 1)		
	noSubMbPartSizeLessThan8x8Flag = 0		
	} else if(!direct_8x8_inference_flag)		
	noSubMbPartSizeLessThan8x8Flag = 0		
	} else {		
	if(transform_8x8_mode_flag && mb_type == I_NxN)		
	transform_size_8x8_flag	2	u(1) ae(v)
	mb_pred(mb_type)	2	
	}		
	if(MbPartPredMode(mb_type, 0) != Intra_16x16) {		
	coded_block_pattern	2	me(v) ae(v)
	if(CodedBlockPatternLuma > 0 && transform_8x8_mode_flag && mb_type != I_NxN && noSubMbPartSizeLessThan8x8Flag && (mb_type != B_Direct_16x16 direct_8x8_inference_flag))		
	transform_size_8x8_flag	2	u(1) ae(v)
	}		
	if(CodedBlockPatternLuma > 0 CodedBlockPatternChroma > 0 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
	mb_qp_delta	2	se(v) ae(v)
	residual(0, 15)	3 4	
	}		
	}		
	}		
	}		

[0090] 在表1B中示出了这一点,其中macroblock_layer_mvc_extension()结构被修改为发送depth_skip_flag和该深度的编码值。当depth_skip_flag为真时,发送该宏块的附加深度值。这由表1B中示出的语法元素depthd来示意。可以如下写出该语法元素的语义:

[0091] depthd指定了当前宏块使用的深度值与其预测值之差。

[0092] depth_skip_flag等于1指示编码宏块处于深度跳过模式。depth_skip_flag等于0指示编码宏块不处于深度跳过模式。如果slice_type不是P也不是B,则推导出该值为0。

[0093] 如下得出当前宏块的最终(即,重构的)深度:

[0094] predDepth=Min(depthA,depthB),

[0095] 当mbA不存在时,

[0096] predDepth=depthB

[0097] 当mbB不存在时,

[0098] predDepth=depthA

[0099] 当mbA和mbB都不存在或没有深度与mbA和mbB相关联时,

[0100] predDepth=128

[0101] depth=predDepth+depthd

[0102] 这里,depthA是左侧相邻宏块的重构深度信号,depthB是顶部相邻宏块的重构深度信号。depthA和depthB是单个值。图6示出了根据本原理的实施例的标记为“Curr”的当前

宏块的相邻宏块,标记为“A”、“B”和“C”。

[0103] 深度跳过模式并不发送宏块的残差,即,像素值的残差。这是有局限性的,因此提出扩展的深度跳过模式,使得允许针对宏块发送残差。将这称作深度导向模式。可以利用或不利用深度标志和深度信息来实现对残差的发送。

[0104] 图7是示出了根据第一说明性实施例的使用利用深度的深度跳过模式来进行编码的示例性方法700的图。在步骤706,读取编码器配置文件。在步骤709,设置SPS扩展中的锚和非锚画面参考。在步骤712,将视点的数目设置为等于N,将变量i和j初始化为零。在步骤715,确定是否 $i < N$ 。如果是,则将控制传递至步骤718。否则,将控制传递至步骤760。

[0105] 在步骤718,确定是否 $j < \text{视点} i \text{ 中的画面数目}(\text{num})$ 。如果是,则将控制传递至步骤721。否则,将控制传递至步骤754。

[0106] 在步骤721,开始对当前宏块的编码。在步骤724,检验宏块模式。在步骤727,检验视点间跳过宏块模式。在步骤730,确定视点间跳过是否是最佳的模式。如果是,则将控制传递至步骤733。否则将控制传递至步骤757。

[0107] 在步骤733,将depth_skip_flag设置为等于1,将depthd设置为左上方像素的深度减去深度预测值。在步骤736,确定是否启用照度补偿。如果是,则将控制传递至步骤739。否则,将控制传递至步骤742。

[0108] 在步骤739,将mb_ic_flag设置为等于1,并设置dpcm_of_dvic。在步骤742,对当前宏块进行编码。在步骤745,确定是否所有宏块都已被编码。如果是,则将控制传递至步骤748。否则,将控制传递至步骤721。

[0109] 在步骤748,递增变量j。在步骤751,递增frame_num和图像序号(POC)。

[0110] 在步骤754,递增变量i并重置frame_num和POC。

[0111] 在步骤757,将depth_skip_flag设置为等于0。

[0112] 在步骤760,确定是否在带内以信号通知SPS、PPS和/或VPS。如果是,则将控制传递至步骤763。否则,将控制传递至步骤766。

[0113] 在步骤763,在带内发送SPS、PPS和/或VPS。

[0114] 在步骤766,在带外发送SPS、PPS和/或VPS。

[0115] 在步骤769,将比特流写入文件或使比特流在网络上流传输。

[0116] 图8是示出了根据第一说明性实施例的使用利用深度的视点间跳过模式来进行解码的示例性方法800的流程图。

[0117] 在步骤806,根据SPS、PPS、VPS、片首部、或网络抽象层(NAL)单元首部来解析view_id。在步骤809,解析其他SPS参数。在步骤812,确定当前画面是否需要解码。如果是,则将控制传递至步骤815。否则,将控制传递至步骤830。

[0118] 在步骤815,确定是否 $\text{POC}(\text{curr}) \neq \text{POC}(\text{prev})$ 。如果是,则将控制传递至步骤818。否则,将控制传递至步骤821。

[0119] 在步骤818,将view_num设置为等于0。在步骤821,以高级别将view_id信息编入索引以确定视点编码顺序,并递增view_num。

[0120] 在步骤824,确定当前画面是否按照期望的编码顺序。如果是,则将控制传递至步骤833。否则,将控制传递至步骤827。

[0121] 在步骤833,解析片首部。在步骤836,解析depth_skip_flag。在步骤839,确定

depth_skip_flag是否等于1。如果是,则将控制传递至步骤842。否则,将控制传递至步骤863。

[0122] 在步骤842,解析depthd和mb_ic_flag。在步骤845,确定mb_ic_flag是否等于1。如果是,则将控制传递至步骤848。否则,将控制传递至步骤849。

[0123] 在步骤848,解析dpcm_of_dvic。在步骤849,使用depthd以获得视差矢量。在步骤851,对当前宏块进行解码。在步骤854,确定是否完成(解码)所有宏块。如果是,则将控制传递至步骤857。否则,将该控制返回步骤836。

[0124] 在步骤857,在解码图像缓冲器(DPB)中插入当前画面。在步骤860,确定是否所有画面都被解码。如果是,则进行解码。否则,将控制返回步骤833。

[0125] 在步骤827,隐藏当前画面。

[0126] 在步骤830,得到下一画面。

[0127] 在步骤863,解析MB模式、mv和ref_idx。

[0128] 实施例2

[0129] 在该实施例中,提出修改运动跳过模式以使用深度信息而不是全局视差矢量(GDV)。

[0130] 在运动跳过模式的至少一种实现方式中,运动跳过模式使用全局视差矢量从相邻视点中推断模式和运动信息。该视差矢量是在编码器处计算的,并且在每个锚画面上发送。然后可以从之前的连续锚画面的GDV得到非锚画面的视差矢量。GDV用于确定另一视点中的对应宏块或子宏块。提出并不使用该全局视差矢量来选择运动跳过模式,而是由针对宏块或子宏块而发送的深度值来选择运动跳过模式。如下所述,深度值可以用于确定视差矢量。表2中示出了子宏块的该深度信号以及矢量的细化的情况。表2示出了语法的修改,修改例如包括以斜体字示出的“depthd”。

[0131] 表2

[0132]

macroblock_layer() {	C	Descriptor
if (! anchor_pic_flag) {		
Motion_skip_flag	2	U(1) ae(v)
}		
if(motion_skip_flag) {		
depthd	2	se(v) ae(v)
for(compidx = 0; compidx < 2; compidx++)		
motion_info_offset_blk[compidx]	2	ue(v) ae(v)
if(num_non_anchor_refs_l0[view_id] > 0 && num_non_anchor_refs_l1[view_id] > 0)		
motion_ref_view_dir	2	U(1) ae(v)
}		
Else {		
mb_type	2	ue(v) ae(v)
...		
}		
if(MbPartPredMode(mb_type, 0) != Intra_16x16) {		
coded_block_pattern	2	me(v) ae(v)
...		
}		
}		

[0133] 如上所述,在编码器处,模式判定将使用宏块或子宏块的深度并卷曲该深度。这等同于视差矢量。

[0134] 该卷曲将指向参考视点中的位置。由于该位置不必须在宏块或子宏块边界处对齐(如运动跳过模式所需的),因此该位置可以被舍入成最近的宏块或子宏块边界。注意,在深度跳过模式下,这样的对齐并不是不可或缺的,因为即使对应像素没有在宏块或子宏块边界上也可以拷贝对应像素。

[0135] 此外,可以在该点周围进行另外的搜索,以细化视差矢量。图9中示出了这一点。即,图9示出了视差匹配900的示例。细化的视差匹配涉及具有要在其中编码的当前宏块910的编码画面920、视点间参考画面950、以及在视点间参考画面950中从当前宏块910到对应宏块980的初始视差矢量970。例如还可以通过以下操作来执行另外的搜索:对910和980执行基于像素的补偿,然后将对应宏块980移至搜索区域990(示为阴影)中的不同位置并选择具有最佳基于像素补偿的块。如图9所示,对应块980占据阴影搜索区域990的中央部分,对应块980以较粗的线框出。

[0136] 在解码器处,执行类似的过程,其中,解码深度值用于将当前宏块或子宏块卷曲到相邻视点的位置。在发送细化的情况下,将细化添加到该位置,然后舍入成最接近的宏块或子宏块位置。

[0137] 编码器使用来自另一视点中的对应宏块或子宏块的模式和运动对当前宏块或子宏块进行编码。模式和运动信息可以例如指向时间上不同但是来自与当前宏块或子宏块的

视点相同的视点的画面中的具体参考块/子块。类似地,解码器相对于得到的运动信息所指向的在时间上不同的(相同视点的)宏块或子宏块,来解码当前宏块或子宏块。

[0138] 图10是示出了根据本发明第二说明性实施例的使用利用深度的运动跳过模式来进行编码的示例性方法1000的流程图。在步骤1006,读取编码器配置文件。在步骤1009,设置SPS扩展中的锚和非锚画面参考。在步骤1012,将视点的数目设置为等于N,将变量i和j初始化为零。在步骤1015,确定是否 $i < N$ 。如果是,则将控制传递至步骤1018。否则,将控制传递至步骤1060。

[0139] 在步骤1018,确定是否 $j < \text{视点} i \text{ 中的画面数目}(\text{num})$ 。如果是,则将控制传递至步骤1021。否则,将控制传递至步骤1057。

[0140] 在步骤1021,开始对当前宏块的编码。在步骤924,检验宏块模式。在步骤1028,使用宏块(MB)的深度值得到视差矢量。在步骤1031,视差矢量被舍入成最接近宏块或子宏块位置。在步骤1032,针对最佳的视差矢量在该(舍入的)位置周围执行搜索。在步骤1034,确定运动跳过是否是最佳模式。如果是,则将控制传递至步骤1035。否则,将控制传递至步骤1058。

[0141] 在步骤1035,将motion_skip_flag设置为等于1,将depthd设置为等于左上方像素的深度减去深度预测值。在步骤1037,设置偏移值。在步骤1042,对当前宏块进行编码。在步骤1045,确定是否所有宏块都已被编码。如果是,则将控制传递至步骤1048。否则,将控制传递至步骤1021。

[0142] 在步骤1048,递增变量j。在步骤1051,递增frame_num和图像顺序号(POC)。

[0143] 在步骤1057,递增变量i并重置frame_num和POC。

[0144] 在步骤1058,将motion_skip_flag设置为等于0。

[0145] 在步骤1060,确定是否在带内以信号通知SPS、PPS和/或VPS。如果是,则将控制传递至步骤1063。否则,将控制传递至步骤1066。

[0146] 在步骤1063,在带内发送SPS、PPS和/或VPS。

[0147] 在步骤1066,在带外发送SPS、PPS和/或VPS。

[0148] 在步骤1069,将比特流写入文件或使比特流在网络上流传输。

[0149] 图11是示出了根据第二说明性实施例的使用利用深度的视点间跳过模式来进行解码的示例性方法1100的流程图。

[0150] 在步骤1106,根据SPS、PPS、VPS、片首部、或网络抽象层(NAL)单元首部来解析view_id。在步骤1109,解析其他SPS参数。在步骤1112,确定当前画面是否需要解码。如果是,则将控制传递至步骤1115。否则,将控制传递至步骤1130。

[0151] 在步骤1115,确定是否 $\text{POC}(\text{curr}) \neq \text{POC}(\text{prev})$ 。如果是,则将控制传递至步骤1118。否则,将控制传递至步骤1121。

[0152] 在步骤1118,将view_hum设置为等于0。在步骤1121,以高级别将view_id信息编入索引以确定视点编码顺序,并递增view_num。

[0153] 在步骤1124,确定当前画面是否按照期望的编码顺序。如果是,则将控制传递至步骤1133。否则,将控制传递至步骤1127。

[0154] 在步骤1133,解析片首部。在步骤1135,解析motion_skip_flag。在步骤1138,确定motion_skip_flag是否等于1。如果是,则将控制传递至步骤1141。否则,将控制传递至步骤

1163。

[0155] 在步骤1141,解析depthd。在步骤1144,解析视差矢量偏移。在步骤1145,通过使用深度并添加视差偏移,来计算视差。在步骤1147,对当前宏块进行解码。在步骤1154,确定是否所有宏块都已完成(被解码)。如果是,则将控制传递至步骤1157。否则,将控制传递至步骤1135。

[0156] 在步骤1157,在解码图像缓冲器(DPB)中插入当前画面。在步骤1160,确定是否所有画面都被解码。如果是,则进行解码。否则,将控制返回步骤1133。

[0157] 在步骤1127,隐藏当前画面。

[0158] 在步骤1130,得到下一画面。

[0159] 在步骤1163,解析MB模式和ref_idx。

[0160] 实施例3

[0161] 在运动矢量预测中,指定了确定将哪个矢量用作运动矢量预测值的过程。类似地,对于视点间运动补偿,使用视差矢量。通常,用于运动矢量预测的相同过程可以应用于视差矢量预测。此外,提出了将从深度值转换而来的视差矢量用作是视差矢量预测值的附加候选。

[0162] 可以将卷曲值用作视差矢量预测值,或者将传统中间预测用作视差矢量预测值。针对宏块级语法的表3以及针对子宏块级语法的表4示出了这一点。

[0163] 表3

[0164]

mb_pred(mb_type) {	C	Descriptor
if(MbPartPredMode(mb_type, 0) == Intra_4x4		
MbPartPredMode(mb_type, 0) == Intra_8x8		
MbPartPredMode(mb_type, 0) == Intra_16x16) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BkIdx=0; luma4x4BkIdx<16; luma4x4BkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BkIdx])		
rem_intra4x4_pred_mode[luma4x4BkIdx]	2	u(3) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BkIdx=0; luma8x8BkIdx<4; luma8x8BkIdx++) {		
prev_intra8x8_pred_mode_flag[luma8x8BkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag[luma8x8BkIdx])		
rem_intra8x8_pred_mode[luma8x8BkIdx]	2	u(3) ae(v)
}		
if(ChromaArrayType != 0)		
intra_chroma_pred_mode	2	ue(v) ae(v)
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
disparity_vec_pred_type	2	u(1) ae(v)
For(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0		
mb_field_decoding_flag) &&		
MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
ref_idx_l0[mbPartIdx]	2	te(v) ae(v)
For(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0		
mb_field_decoding_flag) &&		
MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
ref_idx_l1[mbPartIdx]	2	te(v) ae(v)
For(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx][0][compIdx]	2	se(v) ae(v)
For(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx][0][compIdx]	2	se(v) ae(v)
}		
}		

[0165] 表4

[0166]

sub_mb_pred(mb_type) {	C	Descriptor
disparity_vec_pred_type	2	u(1) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
sub_mb_type[mbPartIdx]	2	ue(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 mb_field_decoding_flag) && mb_type != P_8x8ref0 && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
ref_idx_l0[mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 mb_field_decoding_flag) && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
ref_idx_l1[mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx][subMbPartIdx][compIdx]	2	se(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx][subMbPartIdx][compIdx]	2	se(v) ae(v)
}		

[0167] 表3和表4包括新语法元素“disparity_vec_pred_type”。对于工作在宏块级或子宏块级的实施例,还可以发送深度信息。

[0168] 语法disparity_vec_pred_type具有以下语义:

[0169] disparity_vec_pred_type指示如何得到当前宏块或子宏块的视差矢量预测值。disparity_vec_pred_type等于1指示将从深度值转换而来的视差矢量用作视差矢量预测值。disparity_vec_pred_type等于0指示将使用中间预测值得到的视差矢量用作视差矢量预测值。

[0170] 以下要点可以至少部分地详细产生前述构思并提供多种实现方式的细节。以下这样的实现方式可以对应于之前的实现方式、本发明的变体和/或新实现方式。

[0171] 关于第一点,在深度跳过模式下,可以不需要残差编码,或者可以在带外发送深度。可以在另一视点中确定对应块,这些像素可以用于重构当前块(通常称作宏块或子宏块)。可以发送标志,所述标志指示该编码模式并命令解码器使用来自该重构的参考视点中的对应宏块的像素。

[0172] 然而,在许多实现方式中,需要当前块的深度信息以便找到对应块。如已知的,深度信息可以用于产生视差矢量。相应地,在多种实现方式中,提出了以信号通知模式(可以称作“深度跳过模式”)的语法,并且还提出了也用于发送深度信息的修改语法。

[0173] 注意,许多实现方式使用深度预测值,因此仅需要发送深度差。

[0174] 关于第二点,另一编码模式是运动跳过模式(“MSM”)。该模式查看视点间块以获得

该视点间块的运动数据,以描述从时间参考的运动。即,当前块借用来自另一视点中的块的时间运动信息。然而,在运动跳过模式下,通常使用全局视差矢量来寻找另一视点中的对应块。通常将全局视差矢量与该画面组(GOP)的另一锚画面一起发送,并且该全局视差矢量可以并不是对于画面组中任何给定的块而言都是最佳视差矢量。一种或多种实现方式使用当前块的深度信息来获得指向来自另一视点中对应块的指针。使用深度信息来产生视差矢量,视差矢量基于当前宏块的深度信息,而不使用全局视差矢量。这种实现方式典型地在比特流中或在带外发送深度数据,使得解码器可以执行类似的解码操作。

[0175] 从深度信息计算出的视差矢量指向另一视点中的对应块。然后将来自该对应块的运动信息(例如,运动矢量、参考画面索引等等)用于当前块。当前块使用运动信息来指向时间参考,所述时间参考将被用于对当前块进行编码。

[0176] 关于第三点,在视点间预测中,使用视点运动矢量来寻找另一视点中的对应块。视点运动矢量描述在相同时刻来自不同视点的画面(或画面部分)之间的差异(称作视点运动或简称运动)。视点运动矢量指向另一视点中的参考块,所述另一视点中的参考帧用于对当前宏块进行预测编码。

[0177] 在给定了视点运动与时间运动之间在构思方面的相似性的情况下,通常使用例如传统运动估计技术来产生视点运动矢量,所述传统运动估计技术用于确定在不同时刻在相同视点的画面之间运动。视点运动矢量(“V-MV”)常常简称为“视差矢量”,但是典型地并不是使用深度数据来产生的。也使用V-MV预测值(“V-MVP”),并发送预测值与V-MV之差。该过程与产生MV(在时间预测中)的过程类似,但是该过程应用在视点间预测场景中。然而,在任一场景下,目标涉及使用预测(时间预测或视点间预测)以及将解码所需的信息通知给解码器。提出了修改V-MVP。V-MVP通常是基于来自空间上相邻的块的V-MV而产生的。例如,中间值可以用作预测值。

[0178] 然而,在不同实现方式中,基于当前块的深度信息来计算视差矢量(“DV”)。DV典型地是可以与当前块最紧密对应的另一视点中的块的良好估计,DV可以是比V-MV更优的估计。DV所作指向的块被称作对应块。相应地,多种实现方式还(或备选地)在确定当前块的V-MVP的过程中使用DV。作为三个示例,当前块的V-MVP(1)可以被设置为等于DV,(2)可以是来自与当前块空间上相邻的块的DV和一个或多个V-MV的函数,或者(3)可以是来自DV所指向的对应块的V-MV的函数。注意,如果例如摄像机规则地分隔开或者对象的运动是恒定的,则在计算中,来自对应块的V-MV可以是相关的且有用的。

[0179] 在至少一种实现方式中,解码器执行以下操作:(1)接收所发送的用于当前块的深度数据,(2)计算DV,(3)寻找DV所指向的对应块,(4)可以获得来自该对应块的V-MV,(5)可以获得来自当前块的空间上相邻的一个或多个块,(6)基于一个或多个V-MV和/或DV来计算当前块的V-MVP,(7)接收由编码器发送的“差值”,所述“差值”是编码器确定的V-MVP与V-MV_p之间的差值,(8)通过将接收到的“差值”与计算出的V-MVP相加来计算当前块的V-MV,以及(9)基于计算出的当前块的V-MV所指向的块来对当前块进行解码。实现方式不需要按照所列顺序来执行所列的操作。

[0180] 另一种实现方式仅适用DV来寻找另一视点中的对应块。然后将该对应块看作是编码当前块的参考块。这种实现方式例如在DV所指向的当前块是比V-MV所指向的对应块更好的参考的情况下运作良好。这种实现方式使得无需计算V-MVP和V-MV,还避免了发送“差值”

(V-MVP-V-MV)。

[0181] 图12是示出了根据第三说明性实施例的使用利用深度的深度跳过模式来进行编码的示例性方法1200的流程图。在步骤1206,读取编码器配置文件。在步骤1209,设置SPS扩展中的锚和非锚画面参考。在步骤1212,将视点的数目设置为等于N,将变量i和j初始化为零。在步骤1215,确定是否 $i < N$ 。如果是,则将控制传递至步骤1218。否则,将控制传递至步骤1260。

[0182] 在步骤1218,确定是否 $j < \text{视点} i \text{ 中的画面数目}(\text{num})$ 。如果是,则将控制传递至步骤1221。否则,将控制传递至步骤1254。

[0183] 在步骤1221,开始对当前宏块的编码。在步骤1224,检验宏块模式。在步骤1227,检验视点间跳过宏块模式。在步骤1235,如果使用深度来计算视差矢量(VD)预测值,则将disparity_vec_pred_type设置为等于1;否则,将display_vec_pred_type设置为0。在步骤1242,对当前宏块进行编码。在步骤1245,确定是否所有宏块都已被编码。如果是,则将控制传递至步骤1248。否则,将控制传递至步骤1221。

[0184] 在步骤1248,递增变量j。在步骤1251,递增frame_num和图像顺序号(POC)。

[0185] 在步骤1254,递增变量i并重置frame_num和POC。

[0186] 在步骤1260,确定是否在带内以信号通知SPS、PPS和/或VPS。如果是,则将控制传递至步骤1263。否则,将控制传递至步骤1266。

[0187] 在步骤1263,在带内发送SPS、PPS和/或VPS。

[0188] 在步骤1266,在带外发送SPS、PPS和/或VPS。

[0189] 在步骤1269,将比特流写入文件或使比特流在网络上流传输。

[0190] 在步骤1282,使用视差矢量,将对应宏块的深度用作视差矢量(DV)预测值。

[0191] 在步骤1284,将中间运动矢量用作运动矢量(MV)预测值。

[0192] 图13是示出了根据第三说明性实施例的使用利用深度的视点间跳过模式来进行解码的示例性方法1300的流程图。

[0193] 在步骤1306,根据SPS、PPS、VPS、片首部、或网络抽象层(NAL)单元首部来解析view_id。在步骤1309,解析其他SPS参数。在步骤1312,确定当前画面是否需要解码。如果是,则将控制传递至步骤1315。否则,将控制传递至步骤1330。

[0194] 在步骤1315,确定是否 $\text{POC}(\text{curr}) \neq \text{POC}(\text{prev})$ 。如果是,则将控制传递至步骤1318。否则,将控制传递至步骤1321。

[0195] 在步骤1318,将view_num设置为等于0。在步骤1321,以高级别将view_id信息编入索引以确定视点编码顺序,并递增view_num。

[0196] 在步骤1324,确定当前画面是否按照期望的编码顺序。如果是,则将控制传递至步骤1333。否则,将控制传递至步骤1327。

[0197] 在步骤1333,解析片首部。在步骤1335,解析宏块级数据和深度。在步骤1338,解析disparity_vec_pred_type。在步骤1341,确定disparity_vec_pred_type是否等于1。如果是,则将控制传递至步骤1344。否则,将控制传递至步骤1347。

[0198] 在步骤1344,使用depthd来得到视差矢量预测值。

[0199] 在步骤1347,使用中间运动矢量来得到运动矢量预测值。

[0200] 在步骤1350,对宏块进行解码。在步骤1354,确定是否所有宏块都已完成(被解

码)。如果是,则将控制传递至步骤1357。否则,将控制传递至步骤1335。

[0201] 在步骤1357,在解码图像缓冲器(DPB)中插入当前画面。在步骤1360,确定是否所有画面都已被解码。如果是,则进行解码。否则,将控制返回步骤1333。

[0202] 在步骤1327,隐藏当前画面。

[0203] 在步骤1330,得到下一画面。

[0204] 因此,根据实现方式,使用深度图来寻找另一视点中的对应块,并在没有任何编码的情况下使用对应块来执行深度跳过模式。在其他实现方式中,使用像素比较来估计视差矢量,例如需要可得到运动矢量。然而,在某些情况下,例如,在与像素比较视差矢量相比,深度图产生更精确的视差矢量的情况下,根据本原理对深度图的使用可以比像素比较方法更为优选。实际上,在典型方法中,使用比简单的像素比较更复杂的方法来获得深度图,例如,运动估计风格(ME-style)运动匹配。因此,在这种情况下,深度图可以比简单的ME-style运动匹配提供对视差的更精确的指示。此外,在这种情况下,在给定视差的精确指示时,深度图的使用可以提高视频编码效率。

[0205] 根据另一种实现方式,使用深度图来找到另一视点中的对应块,并执行运动跳过模式,在运动跳过模式下,基于该运动矢量(因此产生新的残差)从对应块和码中得到运动矢量。在具体实现方式中,使用与另一时间相对应的来自相同视点的参考。这至少在深度图的使用上与其他方法不同。此外,其他方法所执行的运动跳过解析通常针对整个图像组(GOP)仅适用单个视差矢量(全局视差矢量)。

[0206] 根据另一实施例,对基于深度图的视差矢量进行优化/细化以得到更好的编码结果。一个这样的细化可以包括在搜索窗口内移动视差矢量并确定速率失真是否得到改善。然后,(至少部分地)使用基于实际深度图的视差矢量来预测基于深度图的视差矢量。

[0207] 上述若干实施例可以对包括深度信号的视频数据进行编码。深度信号不需要被编码,但是可以使用例如差分编码和/或熵编码对深度信号进行编码。类似地,上述若干实施例可以解码包括深度信号的视频数据,并且可以以多种方式处理深度信号或深度值。这样的处理可以包括(根据实现方式),例如解析接收到的深度值并对深度值进行解码(假定深度值已经被编码)。注意,用于处理深度值的处理单元可以包括:例如,比特流解析器202,(2)基于深度的视差转换器203,以及(3)在特定实施例中可以用于对熵编码的深度值进行解码的熵解码器205。

[0208] 本申请的多种实现方式和特征可以用在H.264/MPEG-4AVC(AVC)标准、或者采用MVC扩展的AVC标准、或者采样SVC扩展的AVC标准的上下文中。然而,类似的实现方式和特征可以用在其他标准(已有的或未来的)上下文中,或者用在并不涉及标准的上下文中。因此,提供了具有具体特征和方面的一个或多个实现方式。然而,所描述的实现方式的特征和方面还可以适于其他实现方式。

[0209] 此外,许多实现方式可以被实现在以下装置中的一个或多个装置中:编码器、解码器、对来自解码器的输出进行处理的后处理器、或者向编码器提供输入的预处理器。此外,通过本公开可以想到其他实现方式。

[0210] 因此,提供了一个或多个具有具体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于其他实现方式。实现方式可以使用多种技术来以信号传递信息,所述多种技术包括但不限于SEI消息、片首部、其他高级语法、非高级语法、带外信息、数

据流数据、以及隐式信号传递。相应地,尽管本文所描述的实现方式可以是在具体上下文中描述的,然而这样的描述不应以任何方式将特征和构思限于这样的实现方式或上下文。

[0211] 在说明书中,对本原理的“一个实施例”或“实施例”或“一种实现方式”或“实现方式”及其其他变形的引用,意思是指,结合实施例而描述的特定的特征、结构、特性等等包含在本原理的至少一个实施例中。因此,在说明书全文中多处出现的短语“在一个实施例中”或“在实施例中”或“在一种实现方式”中或“在实现方式中”及其其他变形并不一定全都指的是同一个实施例。

[0212] 应意识到,对于以下“/”、“和/或”、以及“…中的至少一个”(例如,“A/B”、“A和/或B”以及“A和B中的至少一个”)的使用旨在包含:仅对第一所列选项(A)的选择,或者仅对第二所列选项(B)的选择,或者对两个选项(A和B)的选择。作为另一示例,在“A、B和/或C”以及“A、B和C中的至少一个”的情况下,这样的措辞旨在包括:仅对第一所列选项(A)的选择,或者仅对第二所列选项(B)的选择,或者仅第三对所列选项(C)的选择,或者仅对第一和第二所列选项(A和B)的选择,或者仅对第二和第三所列选项(B和C)的选择,或者仅对第一和第三所列选项(A和C)的选择,或者对所有三个选项(A、B和C)的选择。对于本领域和相关领域的技术人员来说,这显然可以扩展到多个所列项目的情况。

[0213] 例如可以以方法或过程、设备、软件程序、数据流或信号的形式来实现本文所描述的实现方式。即使仅在单个形式的实现方式(例如,仅作为方法而讨论的)的上下文中讨论,所讨论的特征的实现方式也可以以其他形式来实现(例如,设备或程序)。例如可以以适当硬件、软件和固件的形式来实现设备。例如可以以设备的形式来实现方法,所述设备例如是处理器,处理器通常是指处理设备,包括例如计算机、微处理器、检测电路或可编程逻辑设备。处理器还包括通信设备,例如,计算机、蜂窝电话、便携式/个人数字助理(“PDA”)、以及便于在最终用户之间传递信息的其他设备。

[0214] 本文所描述的各种过程和特征的实现方式可以以多种不同设备或设施(具体地,例如与数据编码和解码相关联的设备或设施)的形式来体现。这样的设备的示例包括编码器、解码器、对来自解码器的输出进行处理的后处理器、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、web服务器、机顶盒、膝上型计算机、个人计算机、蜂窝电话、PDA以及其他通信设备。应当清楚,设备可以是移动的以及甚至安装在移动交通工具中的。

[0215] 此外,可以通过有处理器来执行的指令来实现方法,这样的指令(和/或由实现方式产生的数据值)可以存储在处理器刻度介质上,例如,集成电路、软件载体或例如硬盘、紧致盘、随机存取存储器(“RAM”)或制度存储器(“RAM”)等其他存储设备。指令可以形成实际体现在计算机可读介质上的应用程序。指令可以是例如硬件、固件、软件或其组合形式的。指令可以存在于例如操作系统、单独的应用程序、或二者组合中。因此,例如,处理器可以表征为被配置为执行处理的设备以及包括具有用于执行处理的指令的处理器可读介质的设备(如,存储设备)。此外,处理器可读介质可以存储由实现方式产生的数据值而不存储指令,或者存储由实现方式产生的数据值以及指令。

[0216] 本领域技术人员将清楚,实现方式可以产生被格式化为承载信息的多种信号,所述信息例如可以被存储或发送。所述信息可以包括例如用于执行方法的指令、或者由所述实现方式之一产生的数据。例如,信号可以被格式化为以数据的形式承载用于写入或读取

所述实施例的语法的规则,或者以数据的形式承载所述实施例所写入的实际语法值。这样的信号例如可以被格式化为电磁波(例如,使用谱的射频部分),或被格式化为基带信号。格式化可以包括例如对数据流进行编码并利用编码数据流对载波进行调制。信号所承载的信息可以是例如模拟信息或数字信息。如已知的,可以在多种不同的有线或无线链路上发送信号。可以将信号存储在处理器可读介质中。

[0217] 已经描述了多种实现方式。然而,将理解,可以进行多种修改。例如,可以对不同的实现方式的元素进行组合、替换、修改或删除,以产生其他实现方式。此外,本领域技术人员将理解,可以用其他结构和过程来替换所公开的这些结构和过程,得到的实现方式将与所公开的实现方式以至少实质上相同的方式之行至少实质上相同的功能,从而达到至少实质上相同的结果。此外,通过本申请可以想到在权利要求范围之内的这些和其他实现方式。

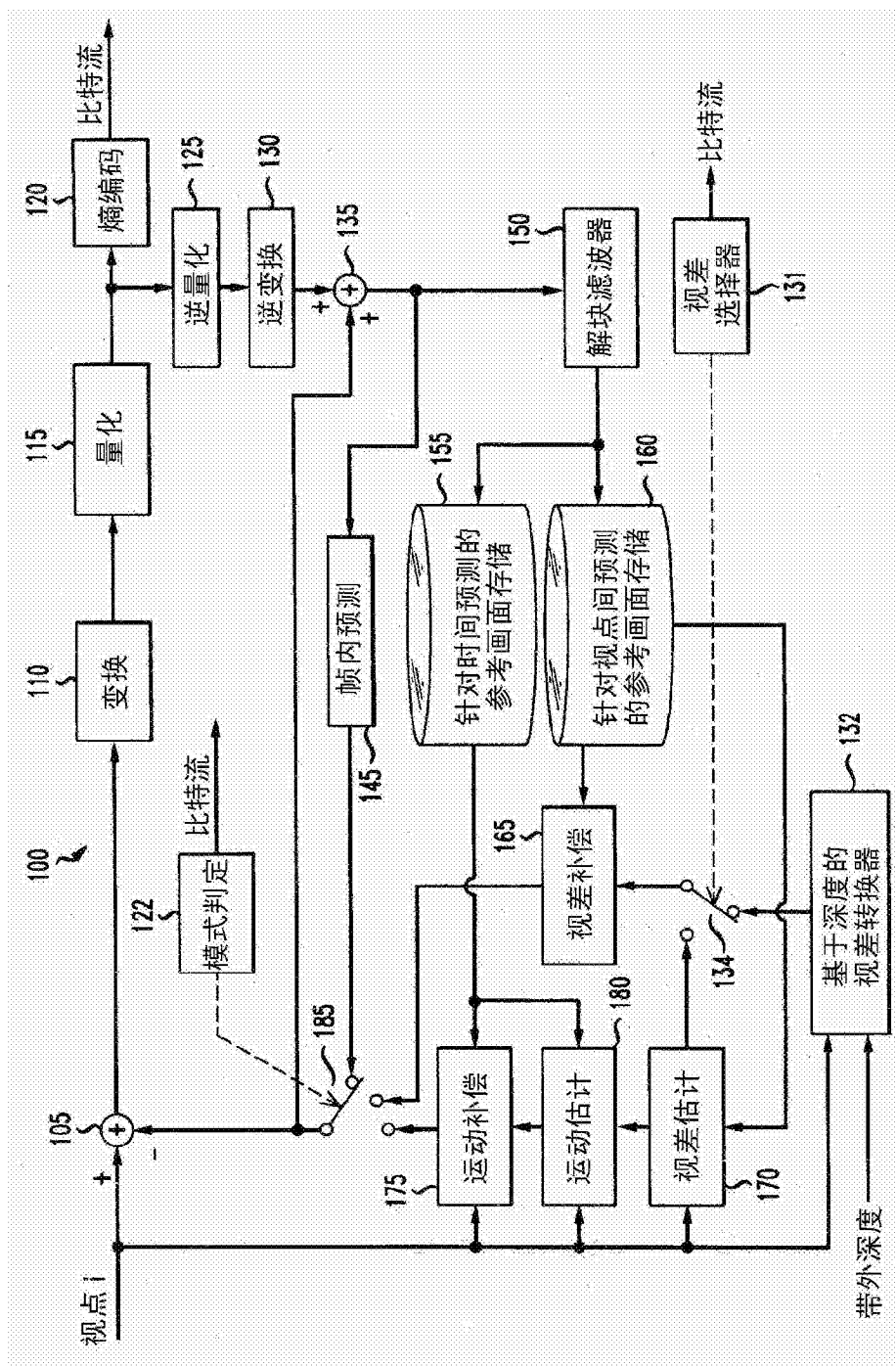


图1

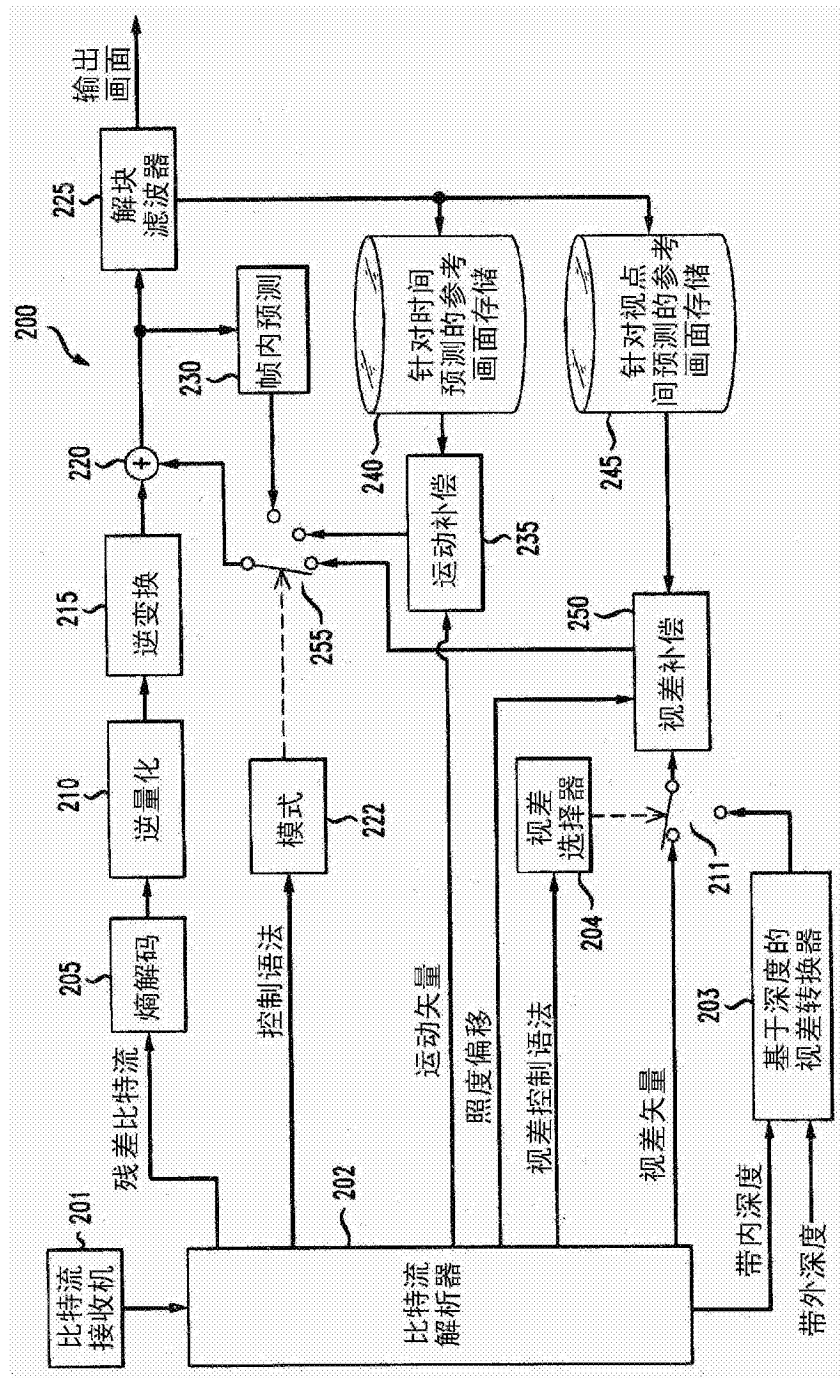


图2

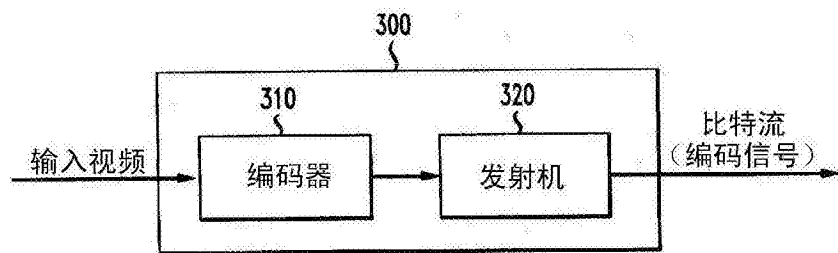


图3

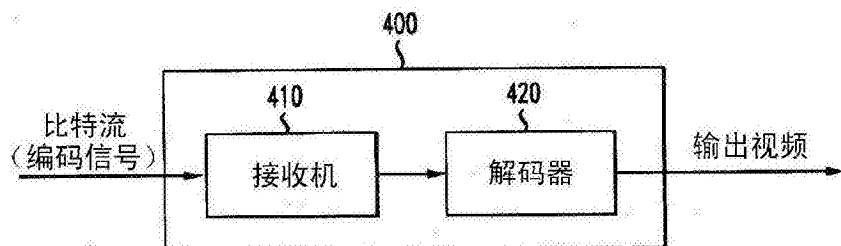


图4

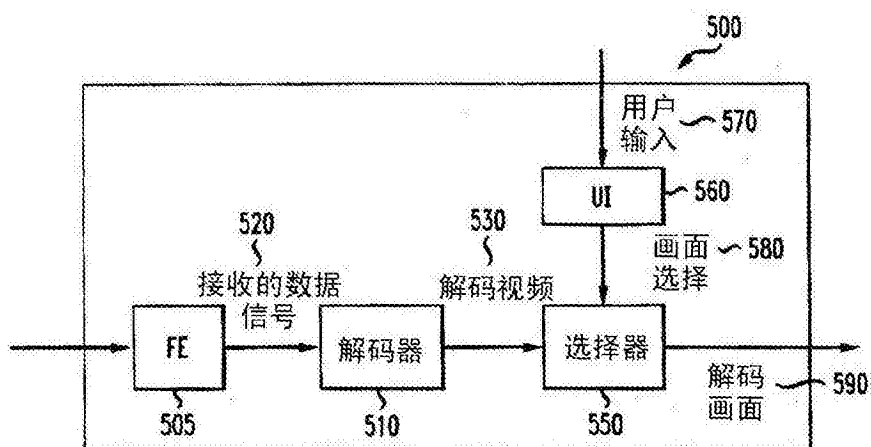


图5

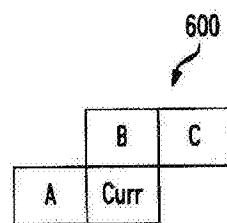


图6

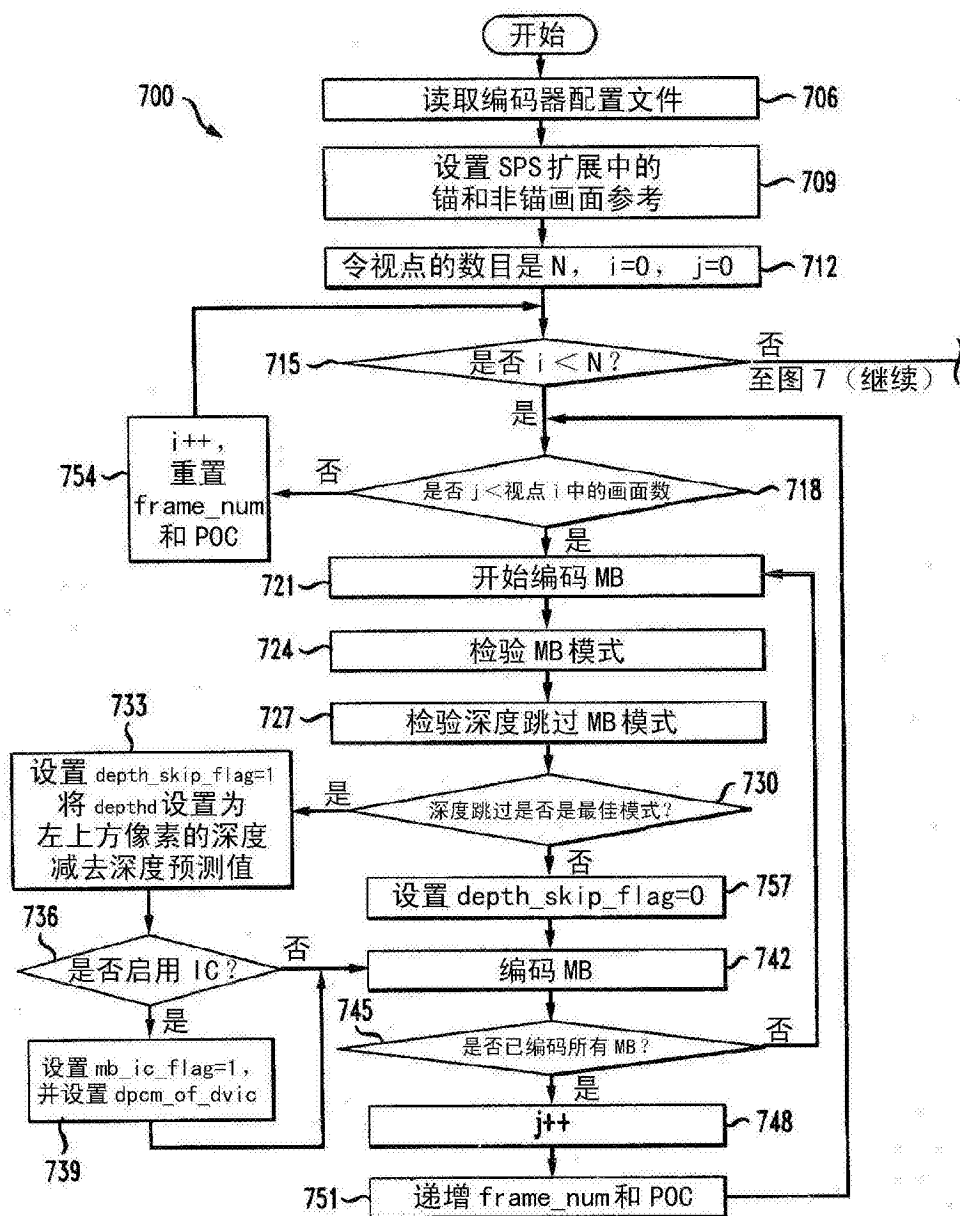


图7

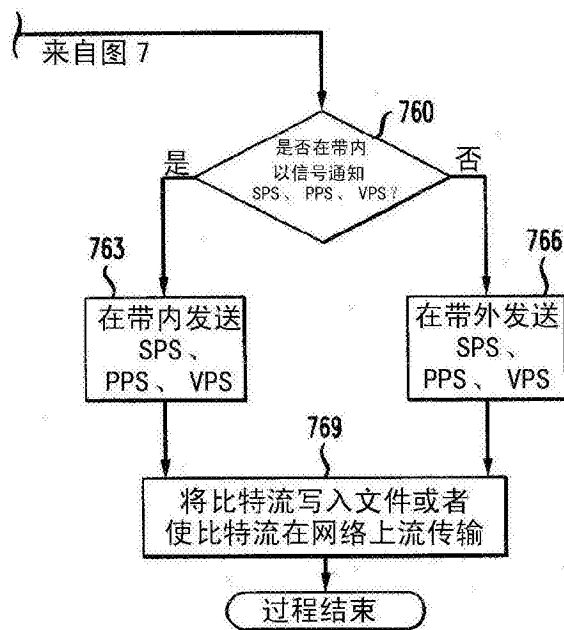


图7(继续)

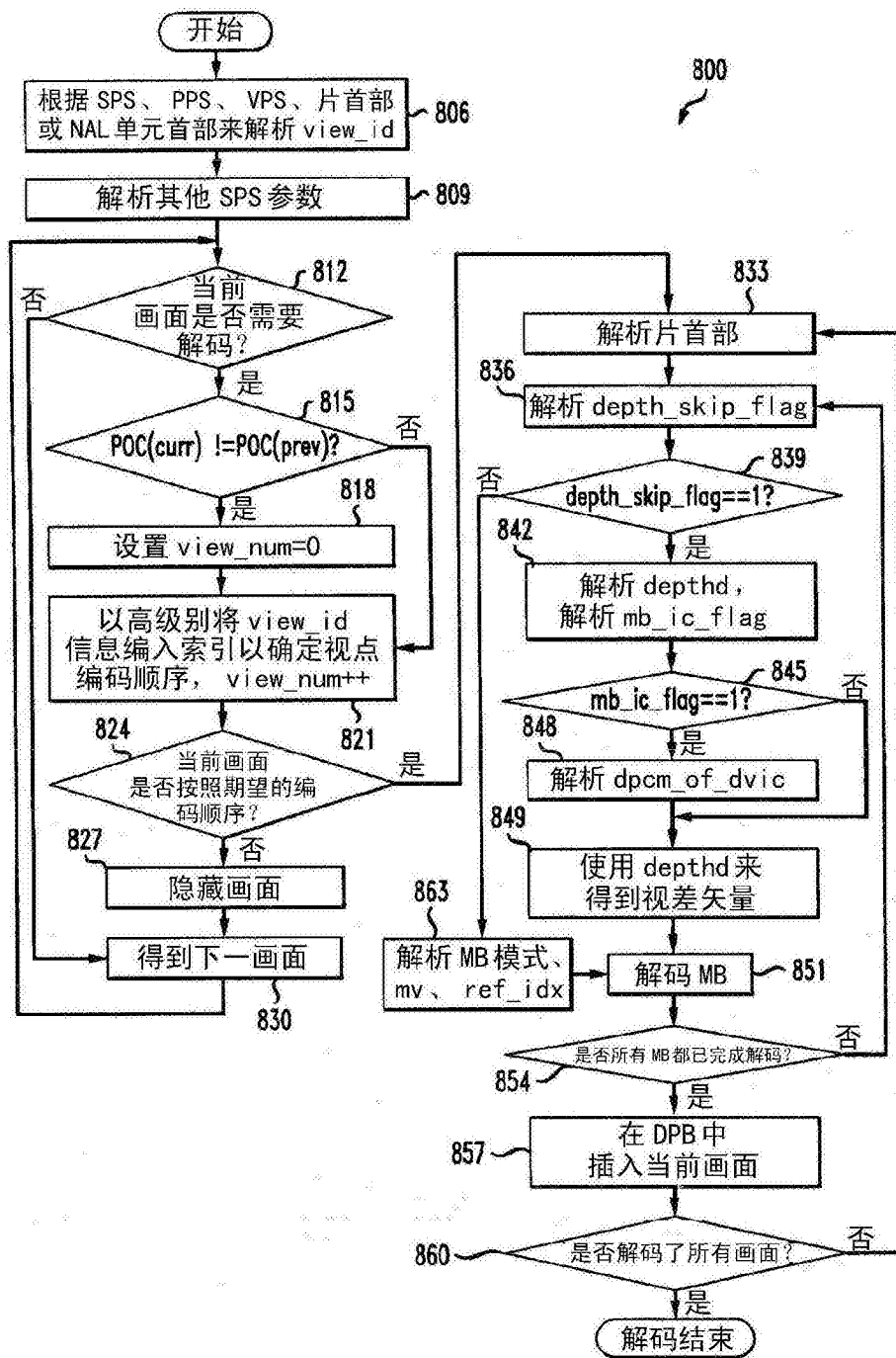


图8

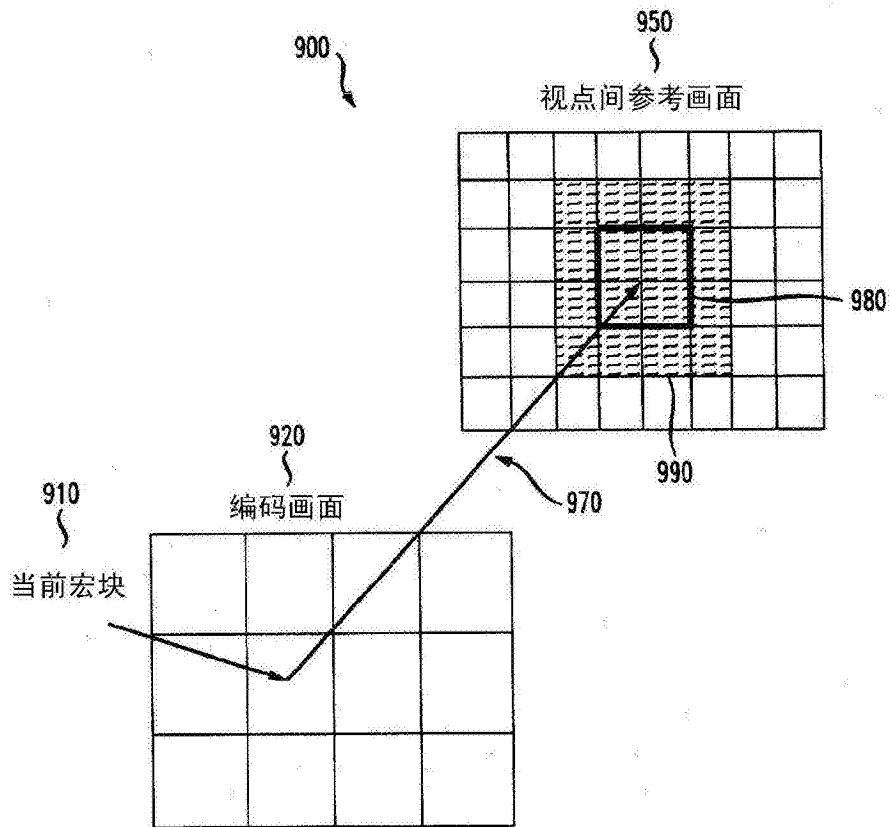


图9

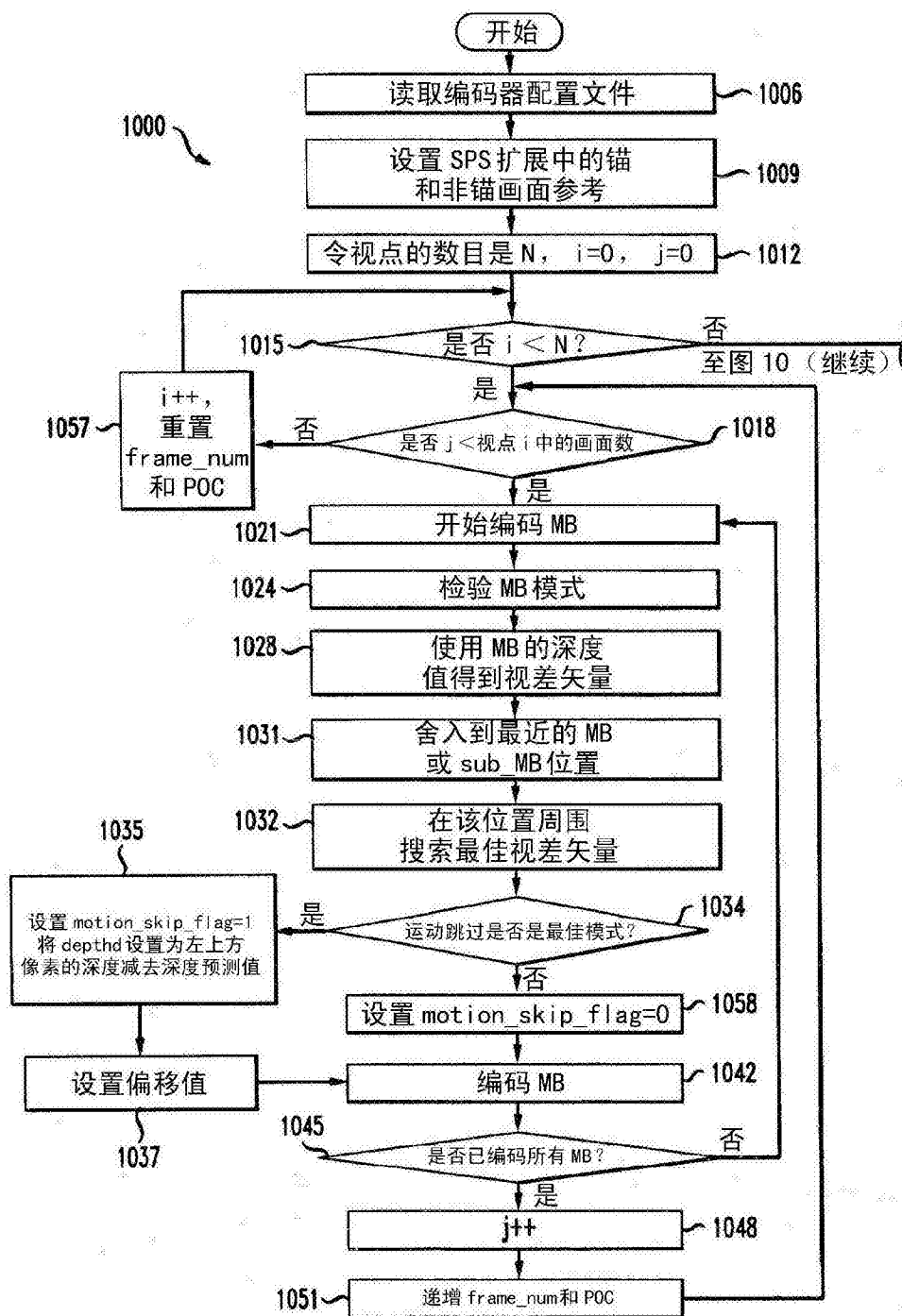


图10

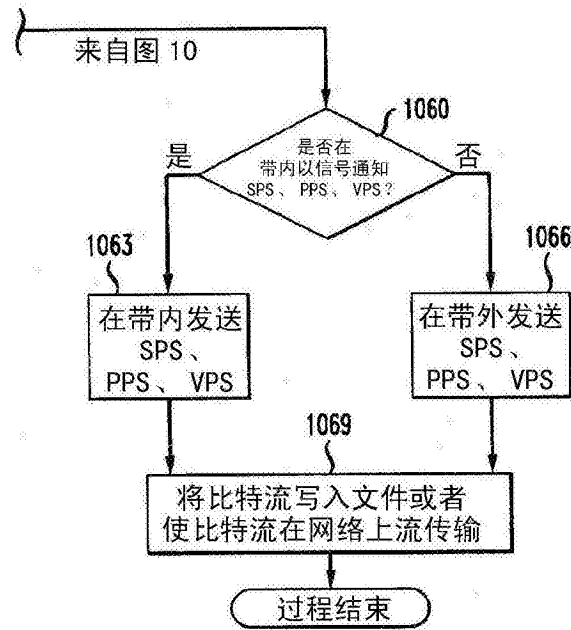


图10(继续)

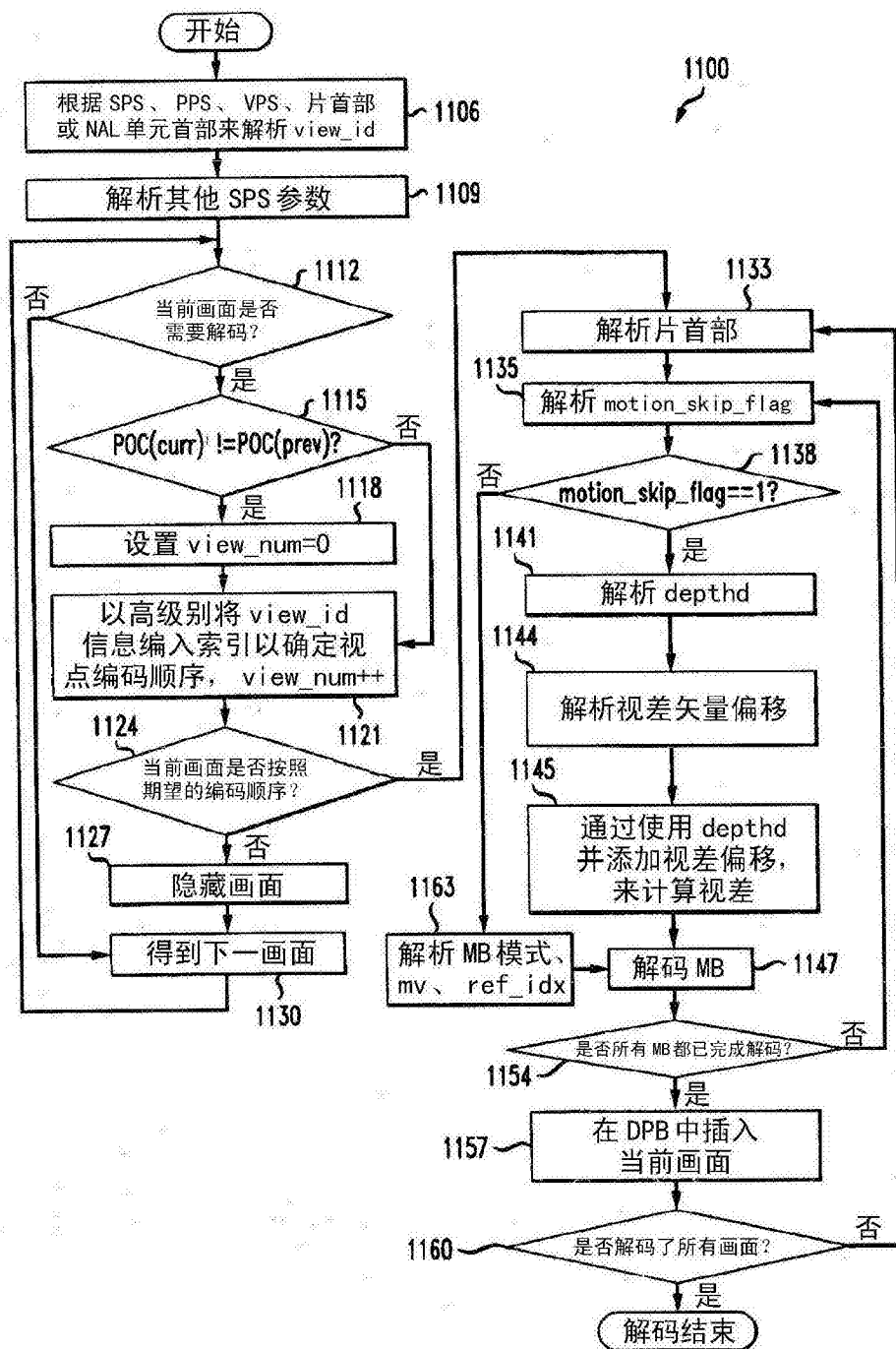


图11

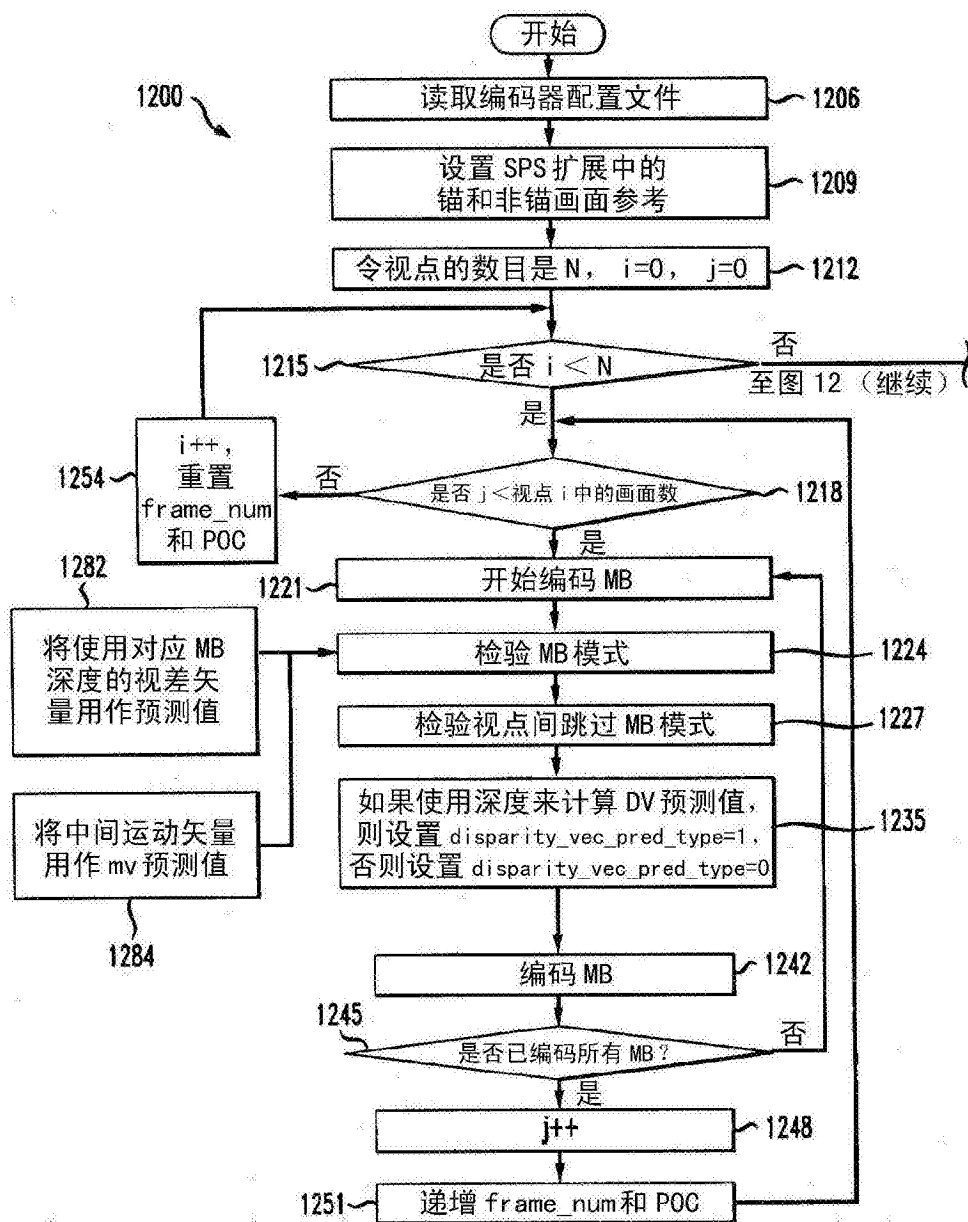


图12

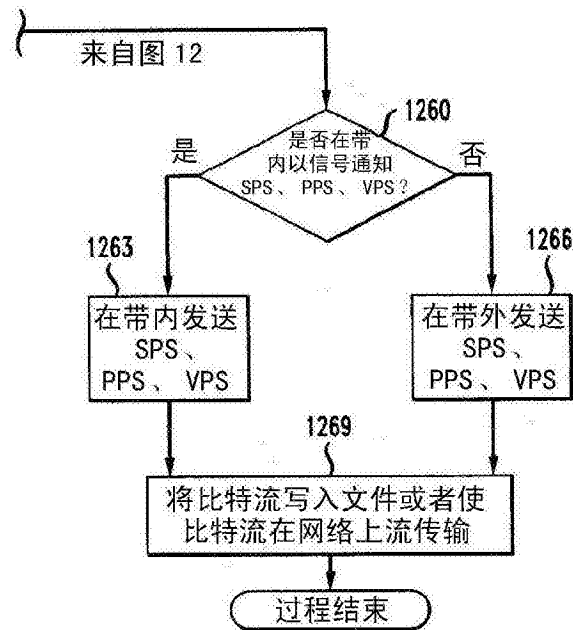


图12(继续)

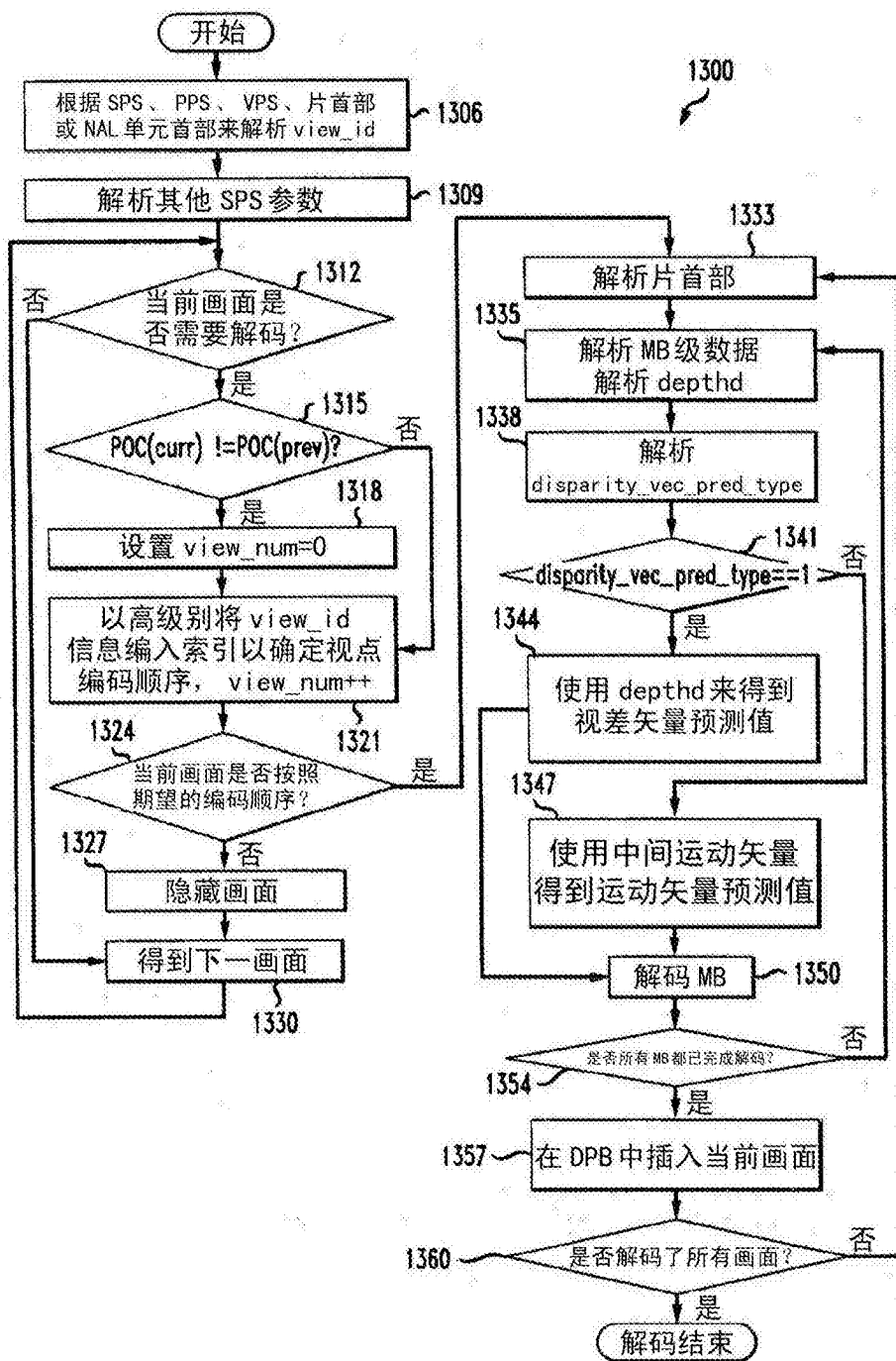


图13