



(12) 发明专利申请

(10) 申请公布号 CN 101911700 A

(43) 申请公布日 2010. 12. 08

(21) 申请号 200880124580. 8

(51) Int. Cl.

(22) 申请日 2008. 12. 18

H04N 7/26 (2006. 01)

(30) 优先权数据

61/010, 823 2008. 01. 11 US

(85) PCT申请进入国家阶段日

2010. 07. 12

(86) PCT申请的申请数据

PCT/US2008/013822 2008. 12. 18

(87) PCT申请的公布数据

W02009/091383 EN 2009. 07. 23

(71) 申请人 汤姆逊许可证公司

地址 法国布洛尼-比扬古市

(72) 发明人 帕文·拜哈斯·潘迪特 尹鹏 田东

(74) 专利代理机构 北京东方亿思知识产权代理

有限责任公司 11258

代理人 宋鹤 南霆

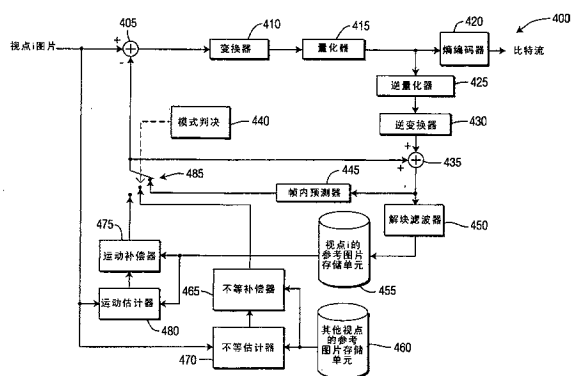
权利要求书 5 页 说明书 17 页 附图 13 页

(54) 发明名称

视频和深度编码

(57) 摘要

描述了多种实现方式。若干实现方式涉及视频和深度编码。一种方法包括选择一图片的视频信息的分量。针对所选视频信息或者针对所述图片的深度信息确定运动向量 (1010, 1040)。基于所确定的运动向量对所选视频信息编码 (1015)。基于所确定的运动向量对所述深度信息编码 (1035)。生成指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符 (1030, 1050)。生成一个或多个数据结构, 所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符 (1065, 1070)。



1. 一种方法,包括:

选择 (1410) 一图片的视频信息的分量;

针对所选视频信息或者针对所述图片的深度信息确定 (1420, 1010, 1040) 运动向量;

基于所确定的运动向量对所选视频信息编码 (1430, 1015);

基于所确定的运动向量对所述深度信息编码 (1440, 1035);

生成 (1450, 1030, 1050) 指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符;以及

生成 (1460, 1065, 1070) 一个或多个数据结构,所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符。

2. 如权利要求 1 所述的方法,其中:

基于所确定的运动向量对所选视频信息编码包括确定 (1015) 所选视频信息和一参考视频图片中的视频信息之间的残余,其中所述参考视频图片中的视频信息由所确定的运动向量指向;并且

基于所确定的运动向量对所述深度信息编码包括确定 (1035) 所述深度信息和一参考深度图片中的深度信息之间的残余,其中所述参考深度图片中的深度信息由所确定的运动向量指向。

3. 如权利要求 1 所述的方法,其中:

确定运动向量包括针对所选视频信息确定 (1010, 1110) 所述运动向量,

基于所确定的运动向量对所选视频信息编码包括确定 (1015, 1115) 所选视频信息和一参考视频图片中的视频信息之间的残余,其中所述参考视频图片中的视频信息由所确定的运动向量指向,并且

基于所确定的运动向量对所述深度信息编码包括:

对所确定的运动向量求精 (1120) 来产生经求精运动向量;以及

确定 (1135) 所述深度信息和一参考深度图片中的深度信息之间的残余,其中所述参考深度图片中的深度信息由所述经求精运动向量指向。

4. 如权利要求 3 所述的方法,还包括:

生成 (1230, 1255) 指示所确定的运动向量和所述经求精运动向量之间的差值的求精指示符;以及

将所述求精指示符包括 (1140, 1250) 到所生成的数据结构中。

5. 如权利要求 1 所述的方法,其中所述图片是帧中的宏块。

6. 如权利要求 1 所述的方法,还包括生成 (1030, 1050) 对所述图片中的一特定切片属于所选视频信息或所选深度信息的指示,并且其中所述数据结构还包括对于所述特定切片的所生成的所述指示。

7. 如权利要求 6 所述的方法,其中所述指示是利用至少一个高层语法元素提供的。

8. 如权利要求 1 所述的方法,其中所述图片对应于多视点视频内容,并且所述数据结构是通过如下操作生成的:对所述图片的给定视点的所述深度信息和所选视频信息进行交织,以使得所述图片的所述给定视点的所述深度信息跟随在所述图片的所述给定视点的所选视频信息之后 (1065)。

9. 如权利要求 1 所述的方法,其中所述图片对应于多视点视频内容,并且所述数据结

构是通过如下操作生成的：对在给定时间实例处所述图片的给定视点的所述深度信息和所选视频信息进行交织，以使得在所述给定时间实例处所述图片的给定视点的经交织后的深度信息和所选视频信息在所述给定时间实例处所述图片的另一视点的经交织后的深度信息和所选视频信息之前 (1065)。

10. 如权利要求 1 所述的方法，其中所述图片对应于多视点视频内容，并且所述数据结构是通过如下操作生成的：对所述深度信息和所选视频信息进行交织，以使得所述深度信息和所选视频信息针对每个时间实例按照视点被交织 (1070)。

11. 如权利要求 1 所述的方法，其中所述图片对应于多视点视频内容，并且所述数据结构是通过如下操作生成的：对所述深度信息和所选视频信息进行交织，以使得多个视点的深度信息和多个视点的所选视频信息针对每个时间实例被交织。

12. 如权利要求 1 所述的方法，其中所述数据结构是通过将所述深度信息布置为所选视频信息的附加分量而生成的，所选视频信息还包括至少一个亮度分量和至少一个色度分量 (1065)。

13. 如权利要求 1 所述的方法，其中相同采样被用于所述深度信息和视频信息的所选分量。

14. 如权利要求 13 所述的方法，其中视频信息的所选分量是亮度分量或色度分量 (1005)。

15. 如权利要求 1 所述的方法，其中所述方法是由编码器执行的。

16. 一种设备，包括：

用于选择一图片的视频信息的分量的装置；

用于针对所选视频信息或者针对所述图片的深度信息确定运动向量的装置；

用于基于所确定的运动向量对所选视频信息编码的装置；

用于基于所确定的运动向量对所述深度信息编码的装置；

用于生成指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符的装置；以及

用于生成一个或多个数据结构的装置，所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符。

17. 一种处理器可读介质，其上存储的指令用于使处理器执行至少以下步骤：

选择一图片的视频信息的分量；

针对所选视频信息或者针对所述图片的深度信息确定 (1010, 1040) 运动向量；

基于所确定的运动向量对所选视频信息编码 (1015)；

基于所确定的运动向量对所述深度信息编码 (1035)；

生成 (1030, 1050) 指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符；以及

生成 (1065, 1070) 一个或多个数据结构，所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符。

18. 一种设备，包括处理器，所述处理器配置来执行至少以下步骤：

选择一图片的视频信息的分量；

针对所选视频信息或者针对所述图片的深度信息确定 (1010, 1040) 运动向量；

基于所确定的运动向量对所选视频信息编码 (1015) ;

基于所确定的运动向量对所述深度信息编码 (1035) ;

生成 (1030, 1050) 指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符 ; 以及

生成 (1065, 1070) 一个或多个数据结构, 所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符。

19. 一种设备 (1500), 包括 :

选择器 (1510), 用于选择一图片的视频信息的分量 ;

运动向量生成器 (1530), 用于针对所选视频信息或者针对所述图片的深度信息确定运动向量 ;

编码器 (1540), 用于基于所确定的运动向量对所选视频信息编码, 并且用于基于所确定的运动向量对所述深度信息编码 ; 以及

生成器 (1580), 用于生成指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符, 并且用于生成一个或多个数据结构, 所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符。

20. 如权利要求 19 所述的设备, 其中所述设备包括一编码器, 该编码器包括所述选择器、所述运动向量生成器、所述编码器、指示符生成器和流生成器。

21. 一种信号, 该信号被格式化以包括数据结构, 所述数据结构包括一图片的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符。

22. 一种处理器可读介质, 其上存储有数据结构, 所述数据结构包括一图片的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符。

23. 一种方法, 包括 :

接收 (1610, 1302) 数据, 所述数据包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符 ;

生成 (1620, 1325, 1340) 在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量 ;

基于所生成的运动向量对所述经编码的视频信息进行解码 (1630, 1330) 来产生所述图片的经解码的视频信息 ; 以及

基于所生成的运动向量对所述经编码的深度信息进行解码 (1640, 1335) 来产生所述图片的经解码的深度信息。

24. 如权利要求 23 所述的方法, 还包括 :

生成 (1345) 包括所述经解码的视频信息和所述经解码的深度信息的数据结构 ;

存储所述数据结构以用在至少一个解码中 ; 以及

显示所述图片的至少一部分。

25. 如权利要求 23 所述的方法,还包括接收 (1302) 在所接收到的数据结构中的、对所述图片中的一特定切片属于所述经编码的视频信息或所述经编码的深度信息的指示。

26. 如权利要求 25 所述的方法,其中所述指示是利用至少一个高层语法元素提供的。

27. 如权利要求 23 所述的方法,其中所接收到的数据是与被布置为所述图片的额外视频分量的所述经编码的深度信息一起接收到的。

28. 如权利要求 23 所述的方法,其中所述方法是由解码器 (500,720) 执行的。

29. 一种设备,包括:

用于接收数据的装置,所述数据包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符;

用于生成在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量的装置;

用于基于所生成的运动向量对所述经编码的视频信息进行解码来产生所述图片的经解码的视频信息的装置;以及

用于基于所生成的运动向量对所述经编码的深度信息进行解码来产生所述图片的经解码的深度信息的装置。

30. 一种处理器可读介质,其上存储的指令用于使处理器执行至少以下步骤:

接收 (1302) 数据,所述数据包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符;

生成 (1325,1340) 在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量;

基于所生成的运动向量对所述经编码的视频信息进行解码 (1330) 来产生所述图片的经解码的视频信息;以及

基于所生成的运动向量对所述经编码的深度信息进行解码 (1335) 来产生所述图片的经解码的深度信息。

31. 一种设备,包括处理器,所述处理器配置来执行至少以下步骤:

接收 (1302) 数据,所述数据包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符;

生成 (1325,1340) 在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量;

基于所生成的运动向量对所述经编码的视频信息进行解码 (1330) 来产生所述图片的经解码的视频信息;以及

基于所生成的运动向量对所述经编码的深度信息进行解码 (1335) 来产生所述图片的经解码的深度信息。

32. 一种设备 (1770),包括:

缓冲器 (1710),用于接收数据,所述数据包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息

基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符；

运动向量生成器 (1760)，用于生成在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量；以及

解码器 (1780)，用于基于所生成的运动向量对所述经编码的视频信息进行解码来产生所述图片的经解码的视频信息，并且用于基于所生成的运动向量对所述经编码的深度信息进行解码来产生所述图片的经解码的深度信息。

33. 如权利要求 32 所述的设备，还包括组装器 (525)，用于生成包括所述经解码的视频信息和所述经解码的深度信息的数据结构。

34. 如权利要求 32 所述的设备，其中所述设备包括一解码器，所述解码器包括所述缓冲器、所述运动向量生成器和解码器。

35. 一种设备，包括：

解调器 (710)，其被配置来接收并解调信号，所述信号包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对所述视频信息或针对所述深度信息所确定的运动向量而被编码的指示符；以及

解码器 (720)，其被配置来执行至少以下步骤：

生成在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量，

基于所生成的运动向量对所述经编码的视频信息进行解码来产生所述图片的经解码的视频信息，并且

基于所生成的运动向量对所述经编码的深度信息进行解码来产生所述图片的经解码的深度信息。

36. 一种设备，包括：

编码器 (400, 610)，其被配置来执行以下步骤：

选择一图片的视频信息的分量，

针对所选视频信息或者针对所述图片的深度信息确定运动向量，

基于所确定的运动向量对所选视频信息编码，

基于所确定的运动向量对所述深度信息编码，

生成指示所选视频信息和所述深度信息基于所确定的运动向量被编码的指示符，以及

生成一个或多个数据结构，所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符；以及

调制器，其被配置来调制并发送所述数据结构。

视频和深度编码

[0001] 相关申请交叉引用

[0002] 本申请要求 2008 年 1 月 11 日提交的题为“Video and Depth Coding”的美国临时申请 No. 61/010,823 的优先权,该在先申请为了一切目的整体通过引用结合于此。

技术领域

[0003] 描述了涉及编码系统的实现方式。多种具体的实现方式涉及视频和深度编码。

背景技术

[0004] 已广泛认识到多视点视频编码 (MVC) 适用于多种应用的关键技术,这些应用包括例如自由视点和三维 (3D) 视频应用、家庭娱乐和监控。深度数据可与每个视点相关联。对于创建额外视点的视点合成来说深度数据是有用的。在多视点应用中,所涉及的视频和深度数据的量可能是庞大的。因此,需要一种框架来帮助提高对例如使用深度数据或者对独立视点执行同时联播的当前视频编码方案的编码效率。

发明内容

[0005] 根据一个总地方面,选择一图片的视频信息的分量。针对所选视频信息或者针对所述图片的深度信息确定运动向量。基于所确定的运动向量对所选视频信息编码。基于所确定的运动向量对所述深度信息编码。生成指示所选视频信息和所述深度信息各自基于所确定的运动向量被编码的指示符。生成一个或多个数据结构,所述数据结构总地包括经编码的视频信息、经编码的深度信息和所生成的指示符。

[0006] 根据另一个总地方面,一种信号被格式化来包括数据结构。该数据结构包括一图片的经编码的视频信息、所述图片的经编码的深度信息、以及一指示符。该指示符指示所述经编码的视频信息和所述经编码的深度信息是基于针对视频信息或针对深度信息所确定的运动向量而被编码的。

[0007] 根据另一个总地方面,接收数据,所述数据包括一图片的视频分量的经编码的视频信息、所述图片的经编码的深度信息、以及指示所述经编码的视频信息和所述经编码的深度信息基于针对视频信息或针对深度信息所确定的运动向量而被编码的指示符。生成在对所述经编码的视频信息和所述经编码的深度信息解码时使用的运动向量。基于所生成的运动向量对所述经编码的视频信息进行解码来产生所述图片的经解码的视频信息。基于所生成的运动向量对所述经编码的深度信息进行解码来产生所述图片的经解码的深度信息。

[0008] 在附图和下面的描述中阐述了一种或多种实现方式的细节。即使以一种特定方式进行了描述,也应当清楚可以以多种方式配置或实现这些实现方式。例如,一种实现方式可以作为方法执行,或者实现为设备,例如,配置来执行一组操作的设备或者存储用于执行一组操作的指令的设备,或者实现在信号中。结合附图和权利要求书,从下面的详细描述中将清楚其他方面和特征。

附图说明

- [0009] 图 1 是具有八个视点 (view) 的多视点视频编码系统的编码结构的实现方式的示意图。
- [0010] 图 2 是具有三个视点的多视点视频加深度编码系统的编码结构的实现方式的示意图。
- [0011] 图 3 是对视点 i 的深度数据的预测的实现方式的框图。
- [0012] 图 4 是用于编码多视点视频内容和深度的编码器的实现方式的框图。
- [0013] 图 5 是用于解码多视点视频内容和深度的解码器的实现方式的框图。
- [0014] 图 6 是视频发送器的实现方式的框图。
- [0015] 图 7 是视频接收器的实现方式的框图。
- [0016] 图 8 是视点和深度数据的排序的实现方式的示意图。
- [0017] 图 9 是视点和深度数据的排序的另一种实现方式的示意图。
- [0018] 图 10 是编码过程的实现方式的流程图。
- [0019] 图 11 是编码过程的另一种实现方式的流程图。
- [0020] 图 12 是编码过程的又一种实现方式的流程图。
- [0021] 图 13 是解码过程的实现方式的流程图。
- [0022] 图 14 是解码过程的另一种实现方式的流程图。
- [0023] 图 15 是编码器的另一种实现方式的框图。
- [0024] 图 16 是解码过程的另一种实现方式的流程图。
- [0025] 图 17 是解码器的另一种实现方式的框图。

具体实施方式

[0026] 在至少一种实现方式中,给出了一种框架用于对多视点视频加深度数据进行编码。另外,还提出了可以提高对视频和深度数据进行编码的编码效率的多种方式。此外,还描述了深度信号可以使用另一深度信号和视频信号来提高编码效率的方法。

[0027] 所解决的多个问题之一是对多视点视频序列进行高效编码。多视点视频序列是从不同视点捕捉同一场景的一组两个或更多个视频序列。尽管可将深度数据与多视点内容的每个视点相关联,但是在一些多视点视频编码应用中视频和深度数据的量可能是庞大的。因此,需要一种框架来帮助提高对例如使用深度数据或者对独立视点执行同时联播的当前视频编码方案的编码效率。

[0028] 由于多视点视频源包括同一场景的多个视点,所以一般在多个视点图像之间存在高度相关性。因此,除时间冗余外还可以利用视点冗余,并且这是通过在不同视点之间执行视点预测来实现的。

[0029] 在一个实际情景中,将使用不同种类的相机或者未完美校准的相机构建包括大量相机的多视点视频系统。对于如此多的相机,解码器的存储器需求可能增大到非常大的量,并且还可能增加了复杂度。另外,某些应用可能仅要求对一组视点中的一些视点进行解码。结果,可能不必完全重构不需要输出的视点。

[0030] 另外,一些视点可能仅承载了深度信息,并且随后利用相关联的深度数据在解码器中被合成。也可以使用深度数据来生成中间可视视点。

[0031] H. 264/AVC 的当前多视点视频编码扩展（下文也称作“MVC 规范”）规定了仅编码视频数据的构架。MVC 规范利用时间和视点间依赖关系来提高编码效率。图 1 中示出了具有 8 个视点的多视点视频编码系统的示例性编码结构 100，该编码结构 100 是 MVC 规范所支持的。图 1 中的箭头示出了依赖关系结构，其中箭头从参考图片指向基于该参考图片被编码的图片。在高层，通知语法来指示不同视点之间的预测结构。该语法在表 1 中示出。具体而言，根据一种实现方式，表 1 示出了符合 MVC 规范的序列参数集合。

[0032] 表 1

	C	Descriptor
seq_parameter_set_mvc_extension() {		
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

[0034] 为了进一步提高编码效率，提出了诸如照明补偿和运动跳过模式之类的多个工具。下面简要描述运动跳过工具。

[0035] 多视点视频编码的运动跳过模式

[0036] 提出运动跳过模式来提高多视点视频编码的编码效率。运动跳过模式至少基于如下概念：两个相邻视点之间存在运动相似性。

[0037] 运动跳过模式直接根据同一时刻相邻视点中的对应宏块来推断运动信息，例如，宏块类型、运动向量和参考索引。该方法可被分解为两个阶段，例如第一阶段中搜索相应宏块、以及第二阶段中导出运动信息。在该示例的第一阶段中，使用全局不等向量（global disparity vector, GDV）来指示相邻视点的图片中的对应位置。该方法利用该全局不等向量来定位向量视点中的对应宏块。全局不等向量是在当前图片和向量视点的图片之间以宏块大小为单位测量出的，从而 GDV 是指示宏块大小单位的位置的粗略向量。可以周期地估计和解码该全局不等向量，例如每个锚定图片（anchor picture）。在该情形中，可以利用来自锚定图片的最近全局不等向量来对非锚定图片的全局不等向量进行插值。例如，当前图片 c 的 GDV 为 $GDV_c = w1 * GDV1 + w2 * GDV2$ ，其中 w1 和 w2 是基于当前图片分别与锚定图片 1 和锚定图片 2 之间的距离的倒数的加权因子。在第二阶段中，从向量视点的图片中的对应宏块导出运动信息，并且拷贝该运动信息以应用到当前宏块。

[0038] 对于在当前宏块被定位于联合多视点视频模型（JMVM）中定义的基本视点的图片

中或者锚定图片中的情形,优选禁用运动跳过模式。这是因为来自邻居视点的图片被用来给出用于帧间预测处理的另一种方法。即,对于运动跳过模式,意在借用来自参考视点的编码模式/帧间预测信息。但是基本视点不具有参考视点,而锚定图片是帧内编码的,所以没有执行帧间预测。因此,对于这些情形优选禁用 MSM。

[0039] 注意,在 JMVM 中,GDV 被发送。

[0040] 为了向解码器通知使用运动跳过模式,例如在多视点视频编码的宏块层语法的头部中包括了一个新的标志, motion_skip_flag。如果 motion_skip_flag 被开启,则当前宏块从相邻视点的对应宏块中导出宏块类型、运动向量和参考索引。

[0041] 与视频数据分离的编码深度数据

[0042] 联合视频组 (JVT) 正工作的当前多视点视频编码规范规定了仅编码视频数据的框架。结果,不能完全支持要求利用深度来生成中间视点(例如,自由视点 TV (FTV)、浸入式媒体和 3D 电话会议)的应用。在该框架中,随后可将重构的视点用作视点的时间预测之外的视点间参考。图 1 根据本原理的一种实现方式示出了本原理可应用到的、具有八个视点的多视点视频编码系统的示例性编码结构 100。

[0043] 在至少一种实现方式中,提出了在多视点视频编码框架中添加深度。深度信号也可以使用与用于每个视点的视频信号的框架类似的框架。这可以通过将深度考虑为另一组视频数据并且利用用于视频数据的同一组工具来实现。图 2 根据本原理的一种实现方式示出了本原理可应用到的、具有三个视点的多视点视频加深度编码系统的另一个示例性编码结构 200(从上到下示出了:前 2 行图片中第一视点的视频和深度、中间两行图片中第二视点的视频和深度、后两行图片中第三视点的视频和深度)。

[0044] 在该示例的框架中,仅深度编码将使用来自运动跳过和视点间预测的深度数据的信息,而视频编码不使用。这种具体实现方式意在独立于视频信号对深度数据进行编码。然而,可以以运动跳过和视点间预测被应用到视频信号的类似方式来将它们应用到深度信号。为了提高编码深度数据的编码效率,提出了视点 i 的深度数据可以使用以下信息:来自时间 j 的其他深度数据的边信息(side information),例如视点间预测和运动信息(运动跳过模式)、视点合成信息等;以及来自与视点 i 相对应的关联视频数据的这种边信息。图 3 示出了视点 i 的深度数据的预测 300。T0、T1 和 T2 对应于不同的时间实例。尽管示出了在从视点 i 的视频数据和视点 j 的深度数据预测时仅可从同一时间实例预测视点 i 的深度,但是这仅是一个实施例。其他系统可以挑选使用任何时间实例。另外,其他系统和实现方式可以来自从多个视点和时间实例的深度数据和/或视频数据的信息的组合来预测视点 i 的深度数据。

[0045] 为了指示视点 i 的深度数据是否使用来自其相关联的视频数据视点 i 或来自另一视点 j 的深度数据的运动模式和其他预测信息,提出了利用语法元素来对此进行指示。语法元素可以是例如在宏块级通知的,或者是以属于深度数据的当前网络抽象层(NAL)单元为条件的。当然,这种通知也可以在另一级别发生,同时维持本原理的精神。

[0046] 表 2 根据一种实现方式示出了运动跳过模式的宏块层的语法元素。

[0047] 表 2

[0048]

macroblock layer() {	C	Descriptor
if (! anchor_pic_flag) {		
i = InverseViewID(view_id)		
if((num_non_anchor_ref_10[i] > 0) (num_non_anchor_ref_11[i] > 0) && motion_skip_enable_flag)		
motion_skip_flag	2	u(1) ae(v)
if(depth_flag)		
depth_data	2	u(1) ae(v)
}		
if (! motion_skip_flag) {		
Mb_type	2	ue(v) ae(v)
if(mb_type == I_PCM) {		
while(!byte_aligned())		
pcm_alignment_zero_bit	2	r(1)
for(i = 0; i < 256; i++)		
pcm_sample_luma[i]	2	u(v)
for(i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i]	2	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	ue(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()	3 4	
}		
}		
}		

[0049] 在一种实现方式中,例如与表 2 相对应的实现方式中,语法 depth_data 具有以下语义:

[0050] depth_data 等于 0 指示当前宏块应当使用与当前宏块的运动预测的当前深度数据相对应的视频数据。

[0051] depth_data 等于 1 指示当前宏块应当使用与在运动预测的依赖关系结构中指示的另一个视点的深度数据相对应的深度数据。

[0052] 另外,深度数据和视频数据可以具有不同的分辨率。一些视点可能具有亚采样的视频数据,而其他视点可能具有它们的亚采样的深度数据或者二者都有。如果如此,则对 depth_data 标志的解释依赖于参考图片的分辨率。在分辨率不同的情形中,可以使用与用于 H. 264/AVC 标准的可缩放视频编码 (SVC) 扩展的方法相同的方法来导出运动信息。在 SVC 中,如果在增强层中的分辨率是基本层的分辨率的整数倍,则编码器将通过首先上采样到相同的分辨率然后执行运动补偿来挑选执行运动和模式层间预测。

[0053] 如果参考图片（深度或视频）的分辨率低于当前正被编码的深度图片，则编码器可以选择不根据该参考图片执行运动和模式解释。

[0054] 有多种可将深度信息发送到解码器的方法。下面为了说明目的而描述这些方法中的若干种。但是，应当理解本原理不限于仅下面的方法，因此也可以使用其他方法来将深度信息发送给解码器，同时维持本原理的精神。

[0055] 图 4 根据本原理的一种实现方式示出了本原理可应用到的一种示例性多视点视频编码 (MVC) 编码器 400。编码器 400 包括组合器 405，组合器 405 的输出以信号通信方式与变换器 410 连接。变换器 410 的输出以信号通信方式与量化器 415 的输入连接。量化器 415 的输出以信号通信方式与熵解码器 420 的输入和逆量化器 425 的输入连接。逆量化器 425 的输出以信号通信方式与逆变换器 430 的输入连接。逆变换器 430 的输出以信号通信方式与组合器 435 的第一正相输入连接。组合器 435 的输出以信号通信方式与帧内预测器 445 的输入和解块滤波器 450 的输入连接。解块滤波器 450 的输出以信号通信方式与参考图片存储单元 455（用于视点 i）的输入连接。参考图片存储器 455 的输出以信号通信方式与运动补偿器 475 的第一输入和运动估计器 480 的第一输入连接。运动估计器 480 的输出以信号连接方式与运动补偿器 475 的第二输入连接。

[0056] 参考图片存储单元 460（用于其他视点）的输出以信号通信方式与不等 / 照明估计器 470 的第一输入和不等 / 照明补偿器 465 的第一输入连接。不等 / 照明估计器 470 的输出以信号连接方式与不等 / 照明补偿器 465 连接。

[0057] 熵解码器 420 的输出可用作编码器 400 的输出。组合器 405 的同相输入可用作编码器 400 的输入，并且以信号通信方式与不等 / 照明估计器 470 的第二输入以及运动估计器 480 的第二输入连接。开关 485 包括以信号通信方式与运动补偿器 475 的输出连接的第一输入、以信号通信方式与不等 / 照明补偿器 465 的输出连接的第三输入、以及以信号通信方式与帧内预测器 445 的输出连接的第三输入。

[0058] 模式判决模块 440 的输出连接到开关 485，用于控制开关 485 选择哪个输入。

[0059] 图 5 根据本原理的一种实现方式示出了本原理可应用到的示例性多视点视频编码 (MVC) 解码器。解码器 500 包括熵解码器 505，熵解码器 505 的输出以信号通信方式与逆量化器 510 的输入连接。该逆量化器的输出以信号通信方式与逆变换器 515 的输入连接。逆变换器 515 的输出以信号通信方式与组合器 520 的第一同相输入连接。组合器 520 的输出以信号通信方式与解块滤波器 525 的输入和帧内预测器 530 的输入连接。解块滤波器 525 的输出以信号通信方式与参考图片存储单元 540（用于视点 i）的输入连接。参考图片存储单元 540 的输出以信号通信方式与运动补偿器 535 的第一输入连接。

[0060] 参考图片存储单元 545（用于其他视点）的输出以信号通信方式与不等 / 照明补偿器 550 的第一输入连接。

[0061] 熵解码器 505 的输入可用作到解码器 500 的输入，用于接收残余比特流。此外，模式模块 560 的输入也可用作到解码器 500 的输入，用于接收控制语法，控制语法用来控制开关 555 选择哪个输入。此外，运动补偿器 535 的第二输入用作解码器 500 的输入，用于接收运动向量。另外，不等 / 照明补偿器 550 的第二输入可用作解码器 500 的输入，用于接收不等向量和照明补偿语法。

[0062] 开关 555 的输出以信号通信方式与组合器 520 的第二同相输入连接。开关 555 的

第一输入以信号通信方式与不等 / 照明补偿器 550 的输出连接。开关 555 的第二输入以信号通信方式与运动补偿器 535 的输出连接。开关 555 的第三输入以信号通信方式与帧内预测器 503 的输出连接。模式模块 560 的输出以信号通信方式与开关 555 连接,用于控制开关 555 选择哪个输入。解块滤波器 525 的输出可用作该解码器的输出。

[0063] 图 6 根据本原理的一种实现方式示出了本原理可应用到的视频发送系统 600。视频发送系统 600 例如可以是头端或者发送系统,用于利用诸如卫星、线缆、电话线或者陆地广播之类的多种媒介来发送信号。可以通过因特网或者一些其他网络来提供这种发送。

[0064] 视频发送系统 600 能够生成并递送包括视频和深度信息的视频内容。这是通过生成包括视频和深度信息的(一个或多个)经编码信息实现的。

[0065] 视频发送系统 600 包括编码器 610 和能够发送经编码信号的发送器 620。编码器 610 接收视频信息和深度信息,并且从其生成(一个或多个)经编码信号。编码器 610 可以是例如上述编码器 300。

[0066] 发送器 620 可以例如适于发送具有代表经编码图片和 / 或与所述图片相关的信息的一个或多个比特流的节目信号。典型的发送器执行例如以下功能中的一个或多个: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、对一个或多个载波上的信号进行调制。发送器可以包括天线(未示出)或者与天线接口。

[0067] 图 7 示出了视频接收系统 700 的一种实现方式的图示。视频接收系统 700 可配置来通过诸如卫星、线缆、电话线或者陆地广播之类的多种媒介接收信号。可通过因特网或者一些其他网络接收这些信号。

[0068] 视频接收系统 700 可以是例如蜂窝电话、计算机、机顶盒、电视机,或者接收经编码视频并且提供例如经解码视频以显示给用户或者存储的其他设备。因此,视频接收系统 700 可以将其输出提供给例如电视机的屏幕、计算机监视器、计算机(用于存储、处理或显示)、或者一些其他存储、处理或显示设备。

[0069] 视频接收系统 700 能接收和处理包括视频和深度信息的视频内容。这是通过接收包括视频和深度信息的(一个或多个)经编码信号实现的。

[0070] 视频接收系统 700 包括能接收经编码信号(例如在本申请的实现方式中描述的信号)的接收器 710 和能对所接收的信号解码的解码器 720。

[0071] 接收器 710 可以例如适于接收具有代表经编码图片的多个比特流的节目信号。典型的接收器执行例如以下功能中的一个或多个:接收经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解调、对信号中的能量去随机化、对信号中的数据去交织、以及对信号执行纠错编码。接收器 710 可以包括天线(未示出)或者与天线接口。

[0072] 解码器 720 输出包括视频信息和深度信息的视频信号。解码器 720 可以例如是上述解码器 400。

[0073] 实施例 1

[0074] 可以将深度与视频数据以下述方式交织,使得在视点 i 的视频数据之后,接着是其相关联的深度数据。图 8 示出了视点和深度数据的排序 800。在该情形中,一个访问单元可被考虑为包括一个给定时间实例的所有视点的视频和深度数据。为了区分一个网络抽象层单元的视频和深度数据,提出了例如在较高层添加语法元素,该语法元素指示该切片属于视频或深度数据。该高层语法可以网络抽象层单元头部、切变头部、序列参数集 (SPS)、图

片参数集 (PPS)、补充增强信息 (SEI) 消息等中给出。表 3 中示出了在网络抽象层单元头部中添加该语法的一个实施例。具体而言,表 3 根据一个实施例示出了 MVC 规范的网络抽象层单元头部。

[0075] 表 3

[0076]

nal unit header svc mvc extension() {	C	Descriptor
svc_mvc_flag	All	u(1)
if (!svc_mvc_flag) {		
idr_flag	All	u(1)
priority_id	All	u(6)
no_inter_layer_pred_flag	All	u(1)
dependency_id	All	u(3)
quality_id	All	u(4)
temporal_id	All	u(3)
use_base_prediction_flag	All	u(1)
discardable_flag	All	u(1)
output_flag	All	u(1)
reserved_three_2bits	All	u(2)
} else {		
priority_id	All	u(6)
temporal_id	All	u(3)
anchor_pic_flag	All	u(1)
view_id	All	u(10)
idr_flag	All	u(1)
inter_view_flag	All	u(1)
depth_flag	All	u(1)
}		
nalUnitHeaderBytes += 3		
}		

[0077] 在一个实施例中,例如在与表 2 相对应的实施例中,语法元素 depth_flag 可以具有以下语义:

[0078] depth_flag 等于 0 表示网络抽象层单元包括视频数据。

[0079] depth_flag 等于 1 表示 NAL 单元包括深度数据。

[0080] 可以针对其他编码标准或者不针对具体的编码标准裁剪其他实现方式。实现方式可以组织视频和深度数据,以使得对于给定的内容单元深度数据跟随视频数据,或者反之。内容单元可以是例如来自给定视点的图片序列、来自给定视点的单个图片、或者来自给定视点的图片的子图片部分(例如,切片、宏块或者子宏块部分)。内容单元可以或者是例如来自给定时间实例的所有可用视点的图片。

[0081] 实施例 2

[0082] 可以独立于视频信号发送深度。图 9 示出了视点和深度数据的另一个排序 900。所给出的表 2 中的高层语法改变仍可应用于该情形。应当注意,深度数据仍作为比特流的一部分与视频数据一起被发送(尽管其他实现方式分离地发送深度数据和视频数据)。可以进行交织使得针对每个时间实例视频和深度被交织。

[0083] 实施例 1 和 2 被认为涉及深度数据的带内发送,因为深度数据被作为比特流的一部分与视频数据一起被发送。实施例 2 产生了 2 个流(视频一个和深度一个),这 2 个流可在系统级或应用级被合并成一个。实施例 2 从而允许在合并的流中视频和深度数据的多种不同配置。此外,这 2 个分离的流也可以被不同地处理,在深度数据关键的应用中提供例如对深度数据的额外纠错(与对视频数据的纠错相比)。

[0084] 实施例 3

[0085] 对于不支持深度数据的使用的某些应用可以不要求深度数据。在这些情形中,可以在带外发送深度数据。这意味着视频和深度数据被去耦合,并且通过任意介质经由分离的信道发送。深度数据仅对于利用该深度数据来执行视点合成的应用是必须的。结果,即使深度数据未到达这些应用的接收器,这些应用仍可以正常工作。

[0086] 在使用深度数据的情形中,例如但不限于 FTV 和侵入式电话会议,可以保证对(带外发送的)深度数据的接收从而使得应用可以及时地使用深度数据。

[0087] 对作为视频数据分量的深度数据编码

[0088] 假定作为视频编码器的输入的视频信号包括亮度和色度数据。与第一方案不同,将深度图作为视频信号的附加分量对待。下面给出将 H. 264/AVC 修改为包括深度图,该深度图作为亮度和色度数据之外的输出。应当理解,该方法可应用于其他标准、视频编码器和/或视频解码器,同时维持本原理的精神。在具体实现方式中,视频和深度在同一 NAL 单元中。

[0089] 实施例 4

[0090] 与色度分量类似,可以在亮度分量之外的位置处对深度进行采样。在一个实施例中,可以按照 4:2:0, 4:2:2 和 4:4:4 对深度进行采样。与 H. 264/AVC 中的 4:4:4 简档类似,可以与亮度/色度分量独立地编码深度分量(独立模式),或者与亮度/色度分量组合地编码(组合模式)。为了实现该特征,给出了如表 4 所示对序列参数集的修改。具体而言,表 4 根据一种实现方式示出了能指示深度采样格式的修改后的序列参数集。

[0091] 表 4

[0092]

seq_parameter_set_rbsp() {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 profile_idc == 110 profile_idc == 122 profile_idc == 144){		
chroma_format_idc	0	ue(v)
if(chroma_format_idc == 3)		
residual_colour_transform_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)
qp_prime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if(seq_scaling_matrix_present_flag)		
for(i = 0; i < 8; i++){		
seq_scaling_list_present_flag[i]	0	u(1)
if(seq_scaling_list_present_flag[i])		
if(i < 6)		

[0093]	scaling_list(ScalingList4x4(i, 16, UseDefaultScalingMatrix4x4Flag(i))	0	
	Else		
	scaling_list(ScalingList8x8(i - 6), 64, UseDefaultScalingMatrix8x8Flag(i - 6))	0	
	}		
	depth_format_idc	0	ue(v)
	...		
	rbp_trailing_bits()	0	
	}		

[0094] depth_format_idc 语法元素的语义如下：

[0095] depth_format_idc 将相对于亮度采样的深度采样指定为色度采样位置。depth_format_idc 的值应在 0 到 3 包括 0 和 3 的范围内。当 depth_format_idc 未出现时,应当推断其等于 0(无深度图出现)。根据通过 depth_format_idc 指定的深度采样格式表 5 中指定了变量 SubWidthD 和 SubHeightD。

[0096]

depth_format_idc	Depth Format	SubWidth D	SubHeight D
0	2D	—	—
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

[0097] 表 5

[0098] 在该实施例中,depth_format_idc 和 chroma_format_idc 应当具有相同的值并且不等于 3,使得深度解码类似于色度分量的解码。包括预测模式的编码模式、以及参考列表索引、参考索引和运动向量都是从色度分量导出的。语法 coded_block_pattern 应被扩展来指示深度变换系数是如何被编码的。一个示例是使用以下公式。

[0099] CodedBlockPatternLuma = coded_block_pattern % 16

[0100] CodedBlockPatternChroma = (coded_block_pattern / 16) % 4

[0101] CodedBlockPatternDepth = (coded_block_pattern / 16) / 4

[0102] CodedBlockPatternDepth 的值 0 意味着所有深度变换级别都等于 0。CodedBlockPatternDepth 的值 1 意味着一个或多个深度 DC 变换系数级别应为非零值,并且所有深度 AC 变换系数级别都为 0。CodedBlockPatternDepth 的值 2 意味着零个或多个深度 DC 变换系数级别为非零值,并且一个或多个深度 AC 变换系数级别应为非零值。深度残余被如表 5 中所示编码。

[0103] 表 5

[0104]

residual() {	C	Descriptor
...		
if(chroma_format_idc != 0) {		
...		
}		
if(depth_format_idc != 0) {		
NumD8x8 = 4 / (SubWidthD * SubHeightD)		
if(CodedBlockPatternDepth & 3) /* depth DC residual present */		
residual_block(DepthDCLevel, 4 * NumD8x8)	3 4	
Else		
for(i = 0; i < 4 * NumD8x8; i++)		
DepthDCLevel[i] = 0		
for(i8x8 = 0; i8x8 < NumD8x8; i8x8++)		
for(i4x4 = 0; i4x4 < 4; i4x4++)		
if(CodedBlockPatternDepth & 2) /* depth AC residual present */		
residual_block(DepthACLevel[i8x8*4+i4x4], 15)	3 4	
Else		
for(i = 0; i < 15; i++)		
DepthACLevel[i8x8*4+i4x4][i] = 0		
}		
}		

[0105] 实施例 5

[0106] 在本实施例中, depth_format_idc 等于 3, 即, 深度是在与亮度相同的位置被采样的。包括预测模式的编码模式、以及参考列表索引、参考索引和运动向量都是从亮度分量导出的。语法 coded_block_pattern 可以与实施例 4 中相同的方式被扩展。

[0107] 实施例 6

[0108] 在实施例 4 和 5 中, 运动向量被设置为与亮度分量或色度分量相同。如果可以基于深度数据对运动向量求精 (refine), 则可以提高编码效率。如表 6 所示给出了运动求精向量。可以利用本领域中多种已知的或者已开发的技术中的任意技术来执行求精。

[0109] 表 6

[0110]

macroblock_layer() {	C	Descriptor
mb_type	2	ue(v) ae(v)
if(mb_type == I_PCM) {		
while(!byte_aligned())		
pcm_alignment_zero_bit	2	f(1)
for(i = 0; i < 256; i++)		
pcm_sample_luma[i]	2	u(v)
for(i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i]	2	u(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 1		
if(mb_type != I_NxN &&		
depth_format_idc != 0) {		
depth_motion_refine_flag	2	u(1) ae(v)
if(depth_motion_refine_flag) {		
motion_vector_refinement_list0_x	2	se(v)
motion_vector_refinement_list0_y	2	se(v)
if(slice_type == B) {		
motion_vector_refinement_list1_x	2	se(v)
motion_vector_refinement_list1_y	2	se(v)
}		
}		
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	
}		
...		
}		

[0111] 建议语法的语义如下：

[0112] depth_motion_refine_flag 指示对于当前宏块是否使能了运动求精。值 1 意味着从亮度分量拷贝来的运动向量将被求精。否则，对于当前向量将不执行求精。

[0113] motion_refinement_list0_x, motion_refinement_list0_y 出现时指示出，如果对于当前宏块设置了 depth_motion_refine，则 LIST0 运动向量将被与通知的求精向量相加。

[0114] motion_refinement_list1_x, motion_refinement_list1_y 出现时指示出，如果对于当前宏块设置了 depth_motion_refine，则 LIST1 运动向量将被与通知的求精向量相加。

[0115] 注意，表格中的上述部分在表格中一般用斜体标明。

[0116] 图 10 根据本原理的一种实现方式示出了用于编码视频和深度信息的一种方法 1000。在 S1005（注意，“S”表示步骤，也称作操作，所以“S1005”可读作“步骤 S1005”），选择相对于亮度和 / 或色度的深度采样。例如，所选择的深度采样可以处于与亮度采样位置

相同的位置或者不同的位置。在 S1010, 基于视频信息生成运动向量 MV_1 。在步骤 S1015, 利用运动向量 MV_1 编码视频信息。在 S1020, 计算出利用 MV_1 的深度编码速率失真成本 RD_1 。

[0117] 在 S1040, 基于深度信息生成运动向量 MV_2 。在 S1045, 计算出利用 MV_2 的深度编码速率失真成本 RD_2 。

[0118] 在 S1025, 判断 RD_1 是否小于 RD_2 。如果是, 则控制前进到 S1030。否则控制前进到 S1050。

[0119] 在 S1030, 将 depth_data 设置为 0, 并且将 MV 设置 MV_1 。

[0120] 在 S1050, 将 depth_data 设置为 1, 并且将 MV 设置 MV_2 。

[0121] “Depth_data”可被称作标志, 并且指示正在使用的运动向量。所以, depth_data 等于 0 意味着应当使用来自视频数据的运动向量。即, 与当前深度数据相对应的视频数据被用于当前宏块的运动预测。

[0122] 而 depth_data 等于 1 意味着应当使用来自深度数据的运动向量。即, 在运动预测的依赖关系结构中所指示的另一个视点的深度数据被用于当前宏块的运动预测。

[0123] 在 S1035, 利用 MV 对深度信息进行编码 (depth_data 被封装到比特流中)。在 S1055, 判断是否要在带内发送深度。如果是, 则控制前进到 S1060。否则, 控制前进到 S1075。

[0124] 在 S1060, 判断是否要将深度作为视频分量处理。如果是, 则控制前进到 S1065。否则, 控制前进到 S1070。

[0125] 在 S1065, 生成数据结构来包括视频和深度信息, 同时深度信息被作为 (例如, 第四) 视频分量处理 (例如, 通过交织视频和深度信息使得视点 i 的深度数据跟随在视点 i 的视频数据之后), 并且 depth_data 被包括在该数据结构中。视频和深度是在宏块级别被编码的。

[0126] 在 S1070, 生成数据结构来包括视频和深度信息, 同时不将深度信息作为视频分量处理 (例如, 通过交织视频和深度信息使得视频和深度信息针对每个时间实例被交织), 并且 depth_data 被包括在该数据结构中。

[0127] 在 S1075, 生成数据结构来包括视频信息但是不包括深度信息, 以便与该数据结构分离地发送深度信息。Depth_data 可被包括在该数据结构中, 或者作为分离的深度数据。注意, 可将视频信息包括在任何类型的格式化数据中, 而不管该格式化数据被称作数据结构还是其他。此外, 可以生成另一个数据结构来包括深度信息。可以在带外发送深度数据。注意, depth_data 可与视频数据 (例如, 在包括视频数据的数据结构中) 和 / 或深度数据 (例如, 在包括该深度数据的数据结构中) 一起被包括。

[0128] 图 11 根据本原理的一种实现方式示出了用于编码视频和深度信息的方法。在 S1110, 基于视频信息生成运动向量 MV_1 。在 S1115, 利用 MV_1 对视频信息进行编码 (例如, 通过确定视频信息和参考图片中的视频信息之间的残余)。在 S1120, 将 MV_1 求精为 MV_2 来最佳编码深度。对运动向量进行求精的一个示例包括在由运动向量指向的区域附近执行局部化搜索来判断是否找到更好的匹配。

[0129] 在 S1125, 生成求精指示符。在 S1130, 对经求精的运动向量 MV_2 进行编码。例如, 可以确定 MV_2 与 MV_1 之间的差并对其编码。

[0130] 在一种实现方式中, 求精指示符是在宏块层中设置的标志。表 6 可被修改来提供

可以如何发送这种标志的示例。表 6 是早期给出用在将深度作为第四维度处理的实现方式中的。但是,也可以在不同的、更宽广的上下文中使用表 6。在本上下文中,也可以使用表 6,并且可以使用下面的语法语义(而不是在表 6 中原始给出的语法的语义)。此外,在允许重新应用表 6 的语义中,如果 `depth_motion_refine_flag` 被设置为 1,则经编码的 MV 将被示为对从视频信号拷贝的运动向量的求精向量。

[0131] 对于对表 6 的重新应用,所提出的语法的语义如下:

[0132] `depth_motion_refine_flag` 指示对于当前宏块是否使能了运动求精。值 1 意味着将对从视频信号拷贝的运动向量求精。否则,不执行对运动向量的求精。

[0133] `motion_refinement_list0_x`, `motion_refinement_list0_y` 出现时指示出,如果对于当前宏块设置了 `depth_motion_refine`,则 LIST0 运动向量将被与通知的求精向量相加。

[0134] `motion_refinement_list1_x`, `motion_refinement_list1_y` 出现时指示出,如果对于当前宏块设置了 `depth_motion_refine`,则 LIST1 运动向量将被与通知的求精向量相加。

[0135] 注意,表格中的上述部分在表格中一般用斜体标明。

[0136] 在 S1135,利用 MV_2 对残余深度编码。这类似于 S1115 中对视频的编码。在 S1140,生成数据结构来包括求精指示符(以及视频信息和可选地包括深度信息)。

[0137] 图 12 根据本原理的一种实现方式示出了在运动向量求精和差值化的情况下对视频和深度信息编码的方法。在 S1210,基于视频信息生成运动向量 MV_1 。在 S1215,利用 MV_1 编码视频信息。在 S1220,将 MV_1 求精为 MV_2 来最佳编码深度。在 S1225,判断 MV_1 是否等于 MV_2 。如果是,则控制前进到 S1230。否则,控制前进到 S1255。

[0138] 在 S1230,将求精指示符设置为 0(假)。

[0139] 在 S1235,对该求精指示符编码。在 S1240,如果求精指示符被设置为真(根据 S1255),则对差值运动向量编码($MV_2 - MV_1$)。在 S1245,利用 MV_2 对残余深度编码。在 S1250,生成数据结构来包括该求精指示符(以及视频信息,可选地包括深度信息)。

[0140] 在 S1255,将求精指示符设置为 1(真)。

[0141] 图 13 根据本原理的一种实现方式示出了用于解码视频和深度信息的方法。在 S1302,接收到一个或多个比特流,所述比特流包括图片的视频分量的经编码视频信息、该图片的经编码深度信息、以及指示符 `depth_data`(其通知是否通过视频信息或深度信息确定了运动向量)。在 S1305,提取出该图片的视频分量的经编码视频信息。在 S1310,从比特流中提取出该图片的经编码深度信息。在 S1315,解析指示符 `depth_data`。在 S1320,判断 `depth_data` 是否等于 0。如果是,则控制前进到 S1325。否则,控制前进到 S1340。

[0142] 在 S1325,基于视频信息生成运动向量 MV。

[0143] 在 S1330,利用运动向量 MV 对视频信号解码。在 S1335,利用运动向量 MV 对深度信号解码。在 S1345,输出包括视频和深度信息的图片。

[0144] 在 S1340,基于深度信息生成运动向量 MV。

[0145] 注意,如果经求精运动向量曾被用于编码深度信息,则在 S1335 之前,可以提取出求精信息并且生成经求精的 MV。随后在 S1335 中,可以使用该经求精的 MV。

[0146] 参考图 14,示出了过程 1400。过程 1400 包括选择图片的视频信息的分量(1410)。

该分量可以是例如亮度、色度、红、绿或者蓝。

[0147] 过程 1400 包括为所选视频信息或者为图片的深度信息确定运动向量 (1420)。可以例如如在图 10 的操作 1010 和 1040 中所述执行操作 1420。

[0148] 过程 1400 包括基于所确定的运动向量对所选视频信息 (1430) 和深度信息 (1440) 进行编码。可以例如分别如在图 10 中的操作 1015 和 1035 中所述执行操作 1430 和 1440。

[0149] 过程 1400 包括生成指示所选视频信息和深度信息基于所确定的运动向量被编码的指示符 (1450)。可以例如如在图 10 中的操作 1030 和 1050 中所述执行操作 1450。

[0150] 过程 1400 包括生成一个或多个数据结构,所述数据结构总地包括经编码视频信息、经编码深度信息和所生成的指示符 (1460)。可以例如如在图 10 中的操作 1065 和 1070 中所述执行操作 1460。

[0151] 参考图 15,示出了设备 1500,例如 H.264 编码器。现在提供设备 1500 的结构和操作的示例。设备 1500 包括接收要被编码的视频的选择器 1510。选择器 1510 选择图片的视频信息的分量,并将所选视频信息 1520 提供给运动向量生成器 1530 和解码器 1540。选择器 1510 可以执行过程 1500 的操作 1410。

[0152] 运动向量生成器 1530 还接收图片的深度信息,并且为所选视频信息 1520 或该深度信息确定运动向量。运动向量生成器 1530 可以以例如类似于图 4 的运动估计块 480 的方式操作。运动向量生成器 1530 将运动向量 1550 提供给解码器 1540。

[0153] 解码器 1540 还接收图片的深度信息。解码器 1540 基于所确定的运动向量对所选视频信息进行编码,并且基于所确定的运动向量对深度信息进行编码。解码器 1540 将经编码的视频信息 1560 和经编码的深度信息 1570 提供给生成器 1580。解码器 1540 可以以例如类似于图 4 中的块 410-435、450、455 和 475 的方式操作。其他实现方式可以例如属于分离的编码器对视频和深度进行编码。解码器 1540 可以执行过程 1400 的操作 1430 和 1440。

[0154] 生成器 1580 生成指示所选视频信息和深度信息基于所确定的运动向量被编码的指示符。生成器 1580 还生成一个或多个数据结构 (示作输出 1590),所述数据结构总地包括经编码视频信息、经编码深度信息和所生成的指示符。生成器 1580 可以例如以与图 4 中的、产生编码器 400 的输出比特流的熵编码块 420 类似的方式操作。其他实现方式可以例如使用分离的生成器来生成指示符和 (一个或多个) 数据结构。此外,例如可由运动向量生成器 1530 或编码器 1504 生成指示符。生成器 1580 可以执行过程 1400 的操作 1450 和 1460。

[0155] 参考图 16,示出了过程 1600。过程 1600 包括接收数据 (1610)。该数据包括图片的视频分量的经编码视频信息、该图片的深度信息、以及指示所述经编码视频信息和经编码深度信息基于针对该视频信息或深度信息确定的运动向量被编码的指示符。该指示符可称作运动向量源指示符,其中该源例如是视频信息或深度信息。可以例如如上针对图 13 中的操作 1302 所述执行操作 1610。

[0156] 过程 1600 包括生成用于对经编码视频信息和经编码深度信息进行解码的运动向量 (1620)。可以例如如上针对图 13 中的操作 1340 所述执行操作 1620。

[0157] 过程 1600 包括基于所生成的运动向量对经编码视频信息进行解码 (1330) 来产生图片的经解码视频信息 (1630)。过程 1600 还包括基于所生成的运动向量对经编码深度信息进行解码 (1335) 来产生图片的经解码深度信息 (1640)。可以例如分别如上针对图 13 中

的操作 1330 和 1335 所述执行操作 1630 和 1640。

[0158] 参考图 17, 示出了设备 1700, 例如 H. 264 解码器。现在提供设备 1700 的结构和操作的示例。设备 1700 包括配置来接收数据的缓冲区 1710, 所述数据包括: (1) 图片的视频分量的经编码视频信息; (2) 该图片的经编码深度信息; 以及 (2) 指示所述经编码视频信息和经编码深度信息基于针对该视频信息或深度信息确定的运动向量被编码的指示符。缓冲区 1710 可以例如以与图 5 的、接收经编码信息的熵解码块 505 类似的方式操作。缓冲区 1710 可以执行过程 1600 的操作 1610。

[0159] 缓冲区 1710 将经编码视频信息 1730、经编码深度信息 1740 和指示符 1750 提供给包括在设备 1700 中的运动向量生成器 1760。运动向量生成器 1760 生成用于对经编码视频信息和经编码深度信息进行解码的运动向量 1770。注意, 运动向量生成器 1760 可以以多种方式生成运动向量 1770, 包括基于先前接收到的视频和 / 或深度数据来生成运动向量 1660、或者通过拷贝已针对先前接收到的视频和 / 或深度数据生成的运动向量来生成运动向量 1660。运动向量生成器 1760 可以执行过程 1600 的操作 1620。运动向量生成器 1760 将运动向量 1770 提供给解码器 1780。

[0160] 解码器 1780 还接收经编码视频信息 1730 和经编码深度信息 1740。解码器 1780 配置来基于所生成的运动向量 1770 对经编码视频信息 1730 进行解码来产生图片的经解码视频信息。解码器 1780 还配置来基于所生成的运动向量 1770 对经解码深度信息 1740 进行解码来产生图片的经解码深度信息。经解码视频和深度信息在图 17 中被示为输出 1790。输出 1790 可以以多种方式和数据结构被格式化。此外, 经解码视频和深度信息不需要作为输出被提供, 或者作为替换可以在输出之前被转换成另一种格式 (例如适于在屏幕上显示的格式)。解码器 1780 可以例如以与图 5 中的、对所接收的数据进行解码的块 510-525、535 和 540 类似的方式操作。解码器 1780 可以执行过程 1600 的操作 1630 和 1640。

[0161] 这里从而提供了多种实现方式。这些实现方式包括例如: (1) 使用来自对视频数据进行编码的信息来对深度数据进行编码; (2) 使用来自对深度数据进行编码的信息来对视频数据进行编码; (3) 将深度数据作为第四 (或者附加) 维度或分量来与视频的 Y、U 和 V 一起编码; 以及 / 或者 (4) 将深度数据作为与视频数据分离的信号进行编码。此外, 这些实现方式可用在多视点视频编码框架的上下文中、另一种标准的上下文中、或者在不涉及标准的上下文中 (例如推荐等)。

[0162] 从而提供了具有特定特征和方面的一种或多种实现方式。然而, 所述实现方式的特征和方面也可以适用于其他实现方式。这些实现方式可以利用多种技术来通知信息, 所述技术包括但不限于 SEI 消息、其他高层语法、非高层语法、带外信息、数据流数据和隐含的信号通知。另外, 尽管在特定上下文中描述了这里所述的实现方式, 但是这些描述绝不应被作为对这些实现方式或上下文的特征和概念的限制。

[0163] 另外, 许多实现方式可以被实现在编码器和解码器之一或二者中。

[0164] 在说明书中提到本原理的“一个实施例”、“实施例”、“一种实现方式”或者“实现方式”以及它们的其他变体意指结合该实施例描述的特定特征、结构、特性等被包括在本原理的至少一个实施例。因此, 在整个说明书中多个地方出现短语“在一个实施例中”、“在实施例中”、“在一种实现方式中”或者“在实现方式中”以及任意其他变体不一定指同一实施例。

[0165] 应当理解, 使用“/”、“和 / 或”和“xxx 中的至少一个” (例如, 在“A/B”、“A 和 / 或

B”和“A和B中至少一个”的情形中)意图包括仅选择第一列出选项(A)、仅选择第二列出选项(B)、或者选择两个选项(A和B)。作为另一个示例,在“A、B和/或C”和“A、B和C中至少一个”的情形中,这种短语用法意图包括仅选择第一列出选项(A)、仅选择第二列出选项(B)、仅选择第三列出选项(C)、仅选择第一和第二列出选项(A和B)、仅选择第一和第三列出选项(A和C)、仅选择第二和第三列出选项(B和C)、或者选择所有三个选项(A和B和C)。对于多个列出项目这可以相应扩展,本领域和相关领域技术人员能够容易地明白这点。

[0166] 这里所述的实现方式可以实现在例如方法或过程、设备、或者软件程序中。即使仅以一种实现方式的上下文中进行了讨论(例如,仅作为方法进行了讨论),也可以以其他形式(例如,设备或程序)实现这里所述的特征的实现方式。设备可以在例如合适的硬件、软件和固件中实现。方法可以在例如设备中,设备例如是一般称作处理设备的处理器,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设备,例如计算机、蜂窝电话、便携式/个人数字助理(“PDA”)和实现最终用户之间的信息传输的其他设备。

[0167] 这里所述的各个过程和特征的实现方式可以实现在多种不同的装备和应用中,具体而言,例如与数据编码和解码相关联的装备或应用中。装备的示例包括视频编码器、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和其他通信设备。应当清楚,装备可以是移动的甚至于安装在移动车辆中。

[0168] 另外,方法可通过由处理器执行的指令实现,并且这种指令(和/或由实现方式产生的数据值)可存储在处理器可读介质中,例如集成电路、软件载体或其他存储设备中,所述其他存储设备例如是硬盘、光盘、随机存取存储器(“RAM”)或只读存储器(“ROM”)。指令可形成有形包含在处理器可读介质中的应用程序。指令可以在例如硬件、固件、软件或者其组合中。可以在例如操作系统、分离的应用或者二者中找到指令。处理器可被表征为例如配置来执行过程的设备和包括具有用于执行过程的指令的处理器可读介质的设备二者。

[0169] 本领域技术人员将明白,这些实现方式可以产生格式化来承载例如可被存储或发送的信息的多种信号。信息可包括例如用于执行方法的指令、或者由上述实现方式之一产生的数据。例如,信号可被格式化来承载用于写或者读所述实施例的语法的规则作为数据,或者承载由所述实施例写入的实际语法值作为数据。这样的信号可被格式化为例如电磁波(例如,使用频谱的射频部分)或者基带信号。格式化可以包括例如对数据流进行编码和利用经编码数据流对载波进行调制。信号承载的信息可以例如是模拟信号或者数字信号。可以通过已知的多种不同有线或无线链路发送信号。

[0170] 已描述了多种实现方式。但是,将理解可以作出多种修改。例如,不同实现方式的元素可被组合、补充、修改或者移除来得到其他实施例。另外,本领域技术人员将理解,可以用其他结构和过程来替换这里公开的那些,从而得到的实现方式将以(一种或多种)至少基本相同的方式来执行(一种或多种)至少基本相同的功能,来实现(一个或多个)与这里公开的实现方式至少基本相同的结果。因此,这些和其他实现方式可由本申请所预期,并且在所附权利要求书的范围内。

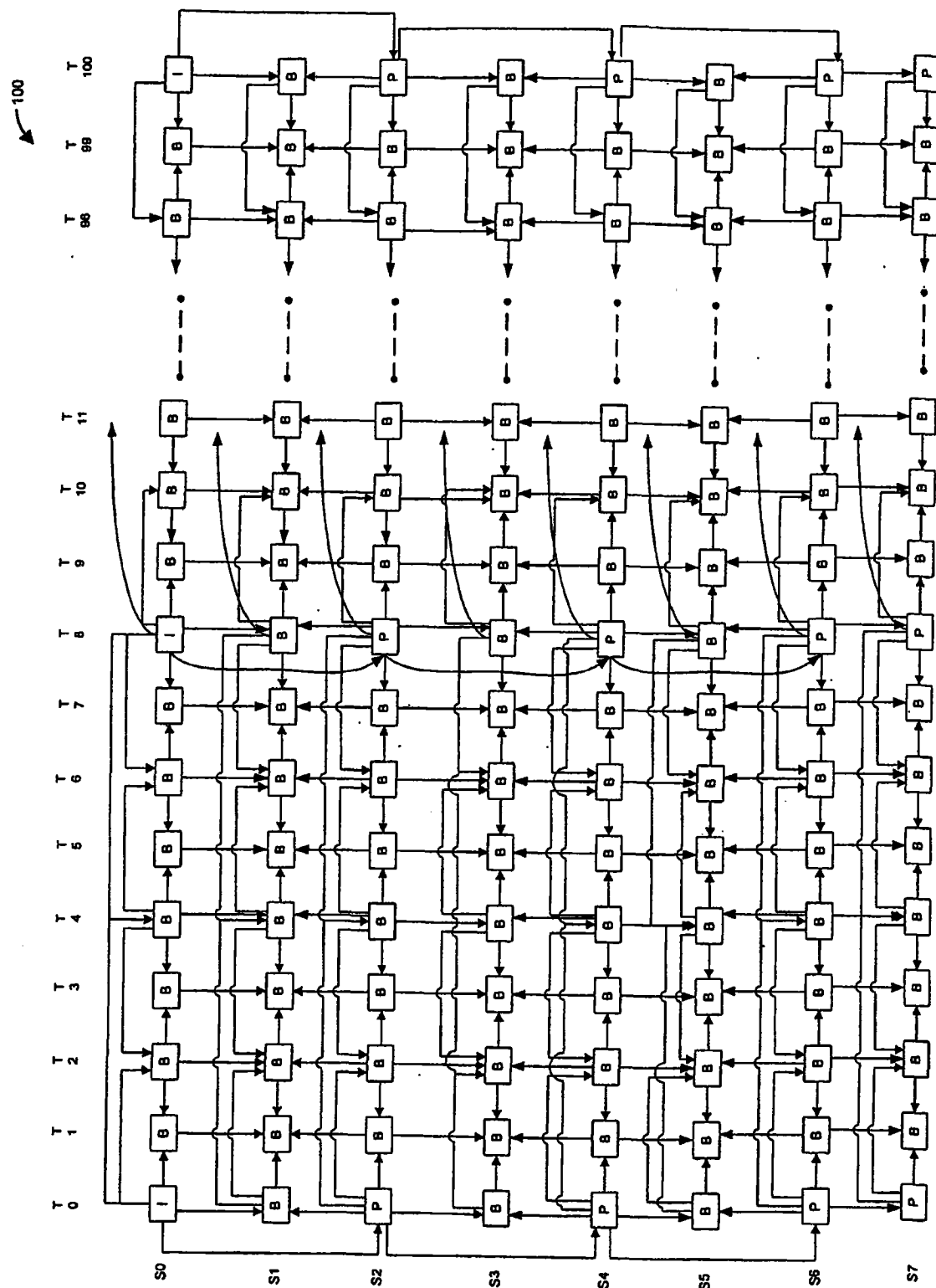


图 1

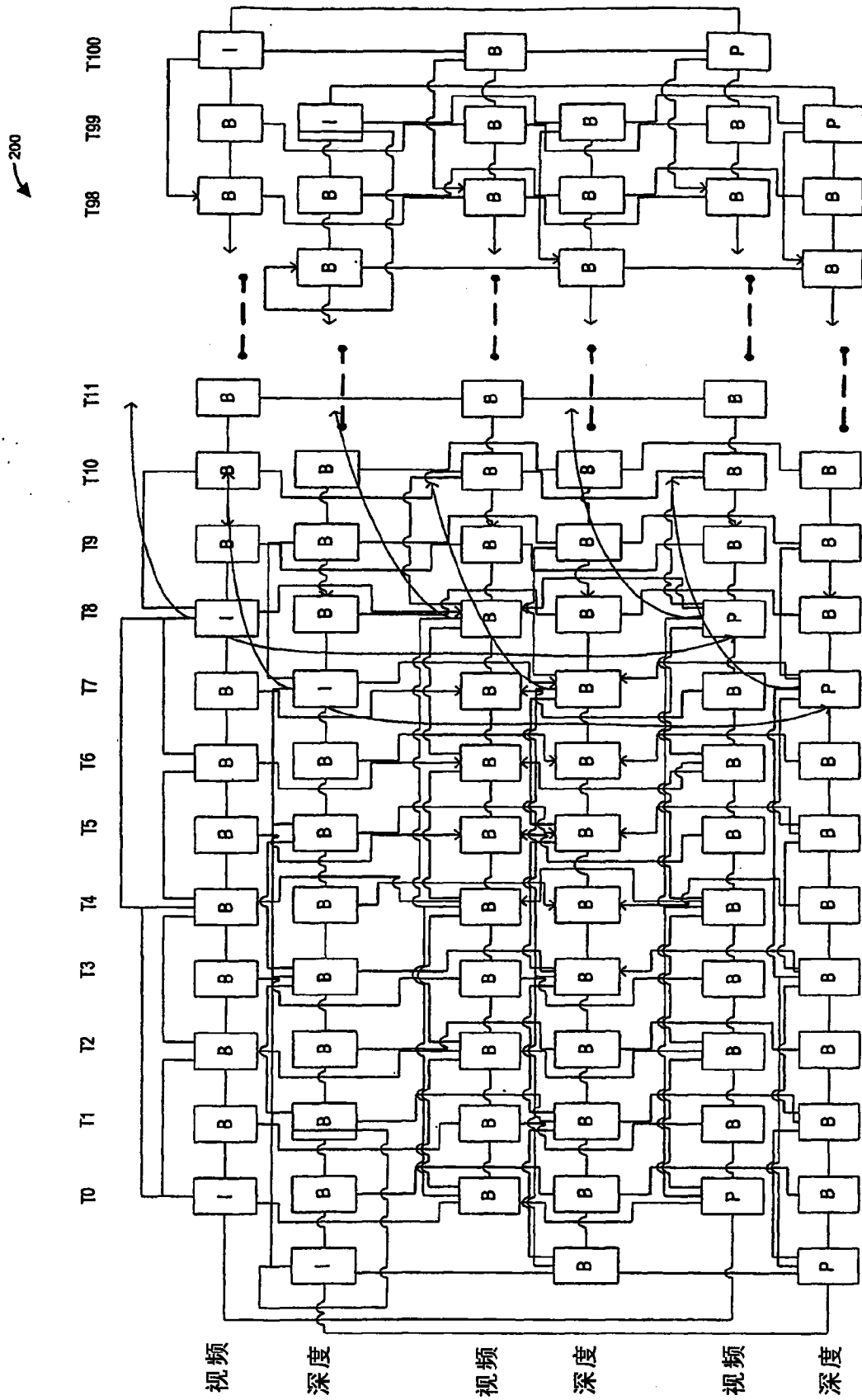


图 2

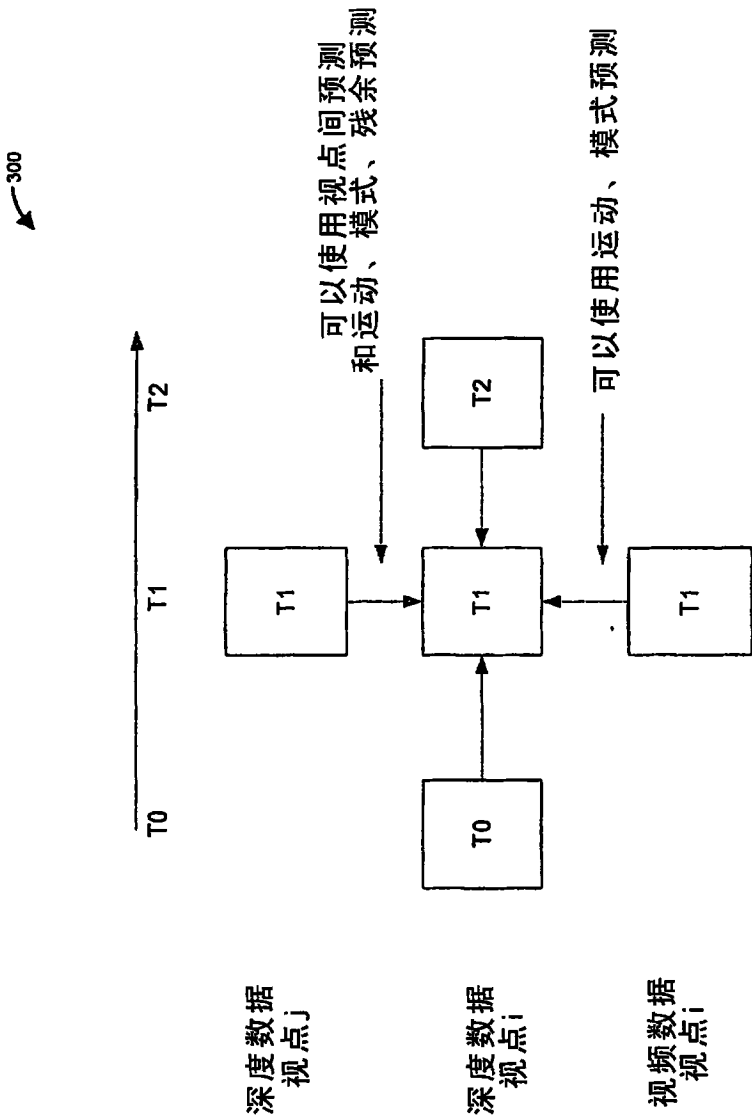


图 3

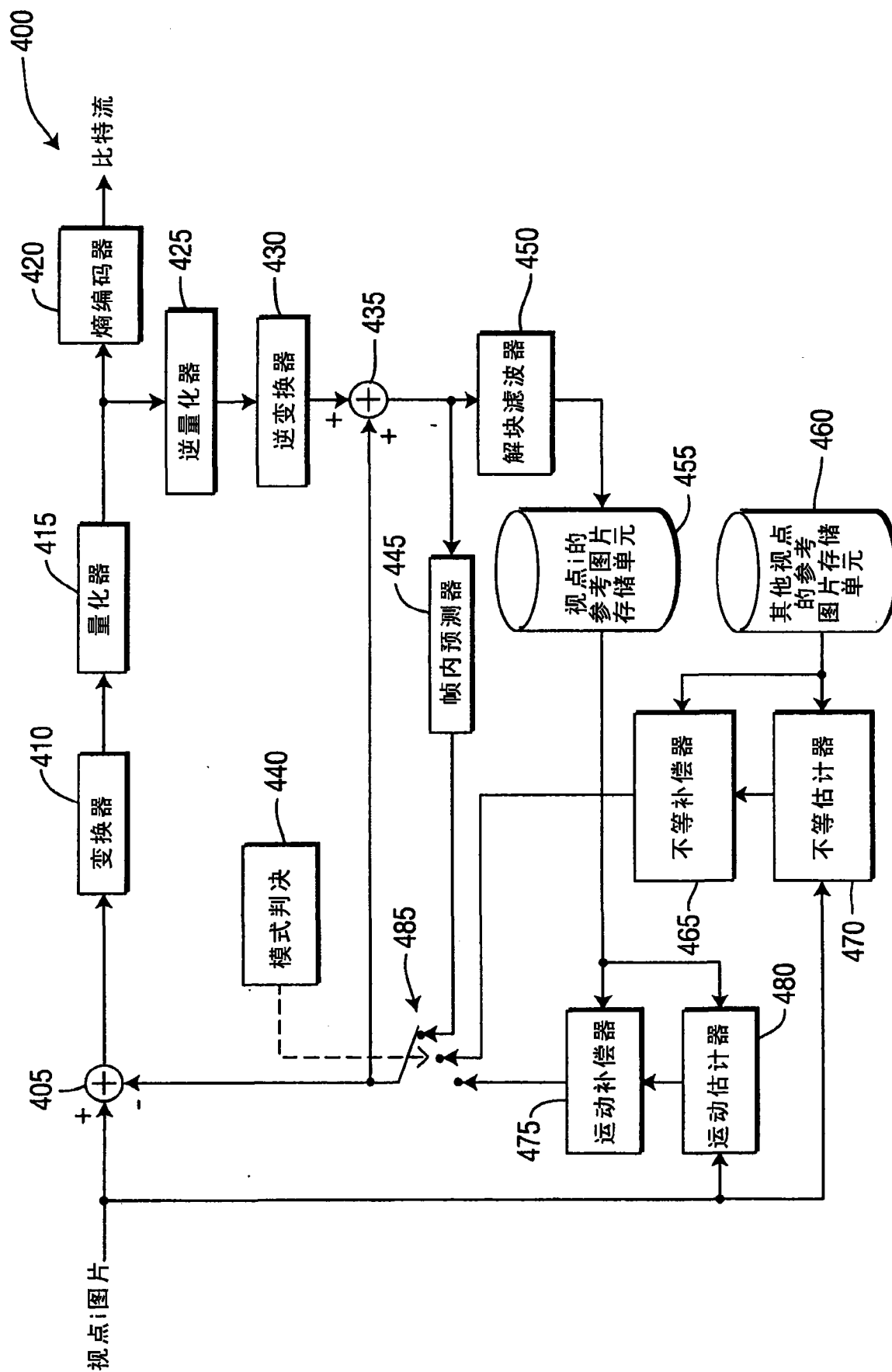


图 4

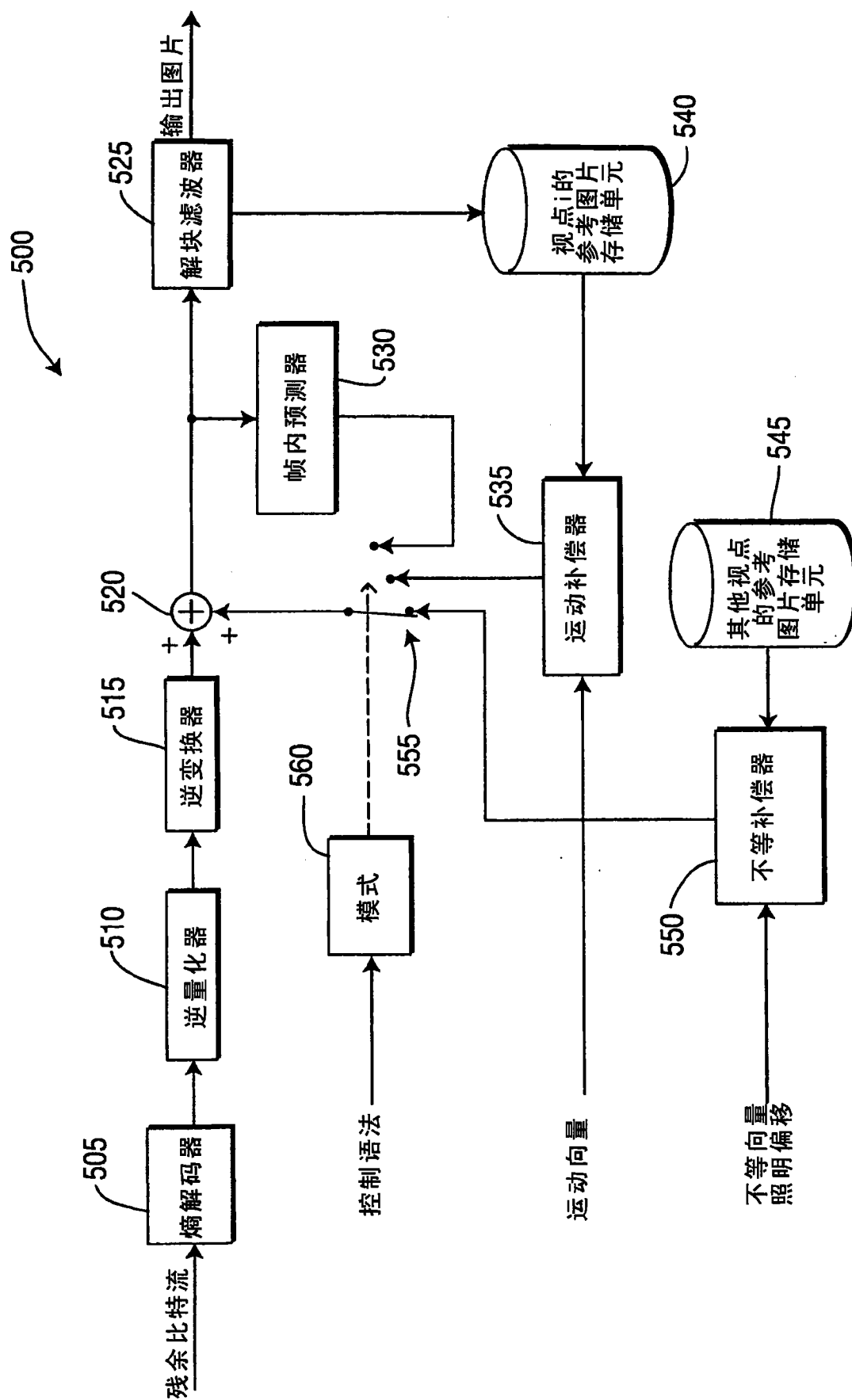


图 5

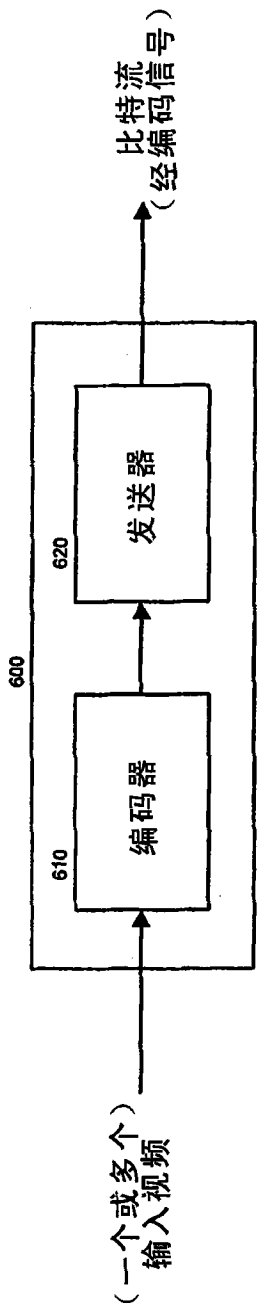


图 6

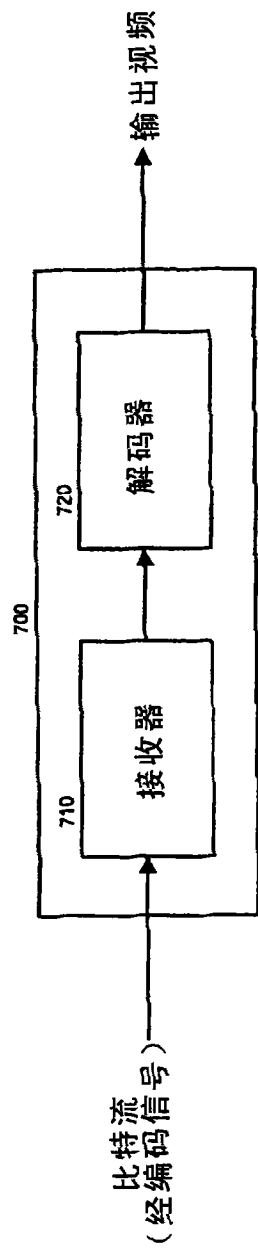


图 7

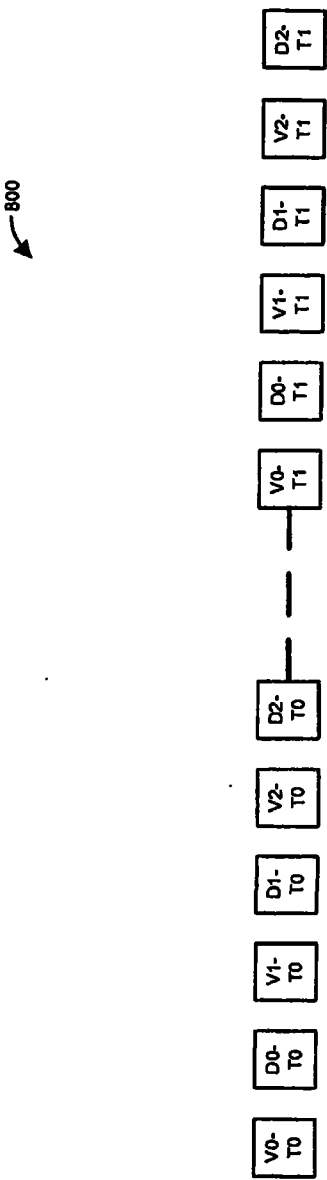


图 8

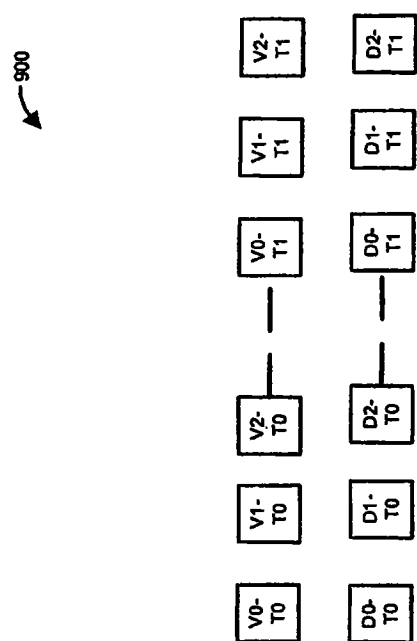


图 9

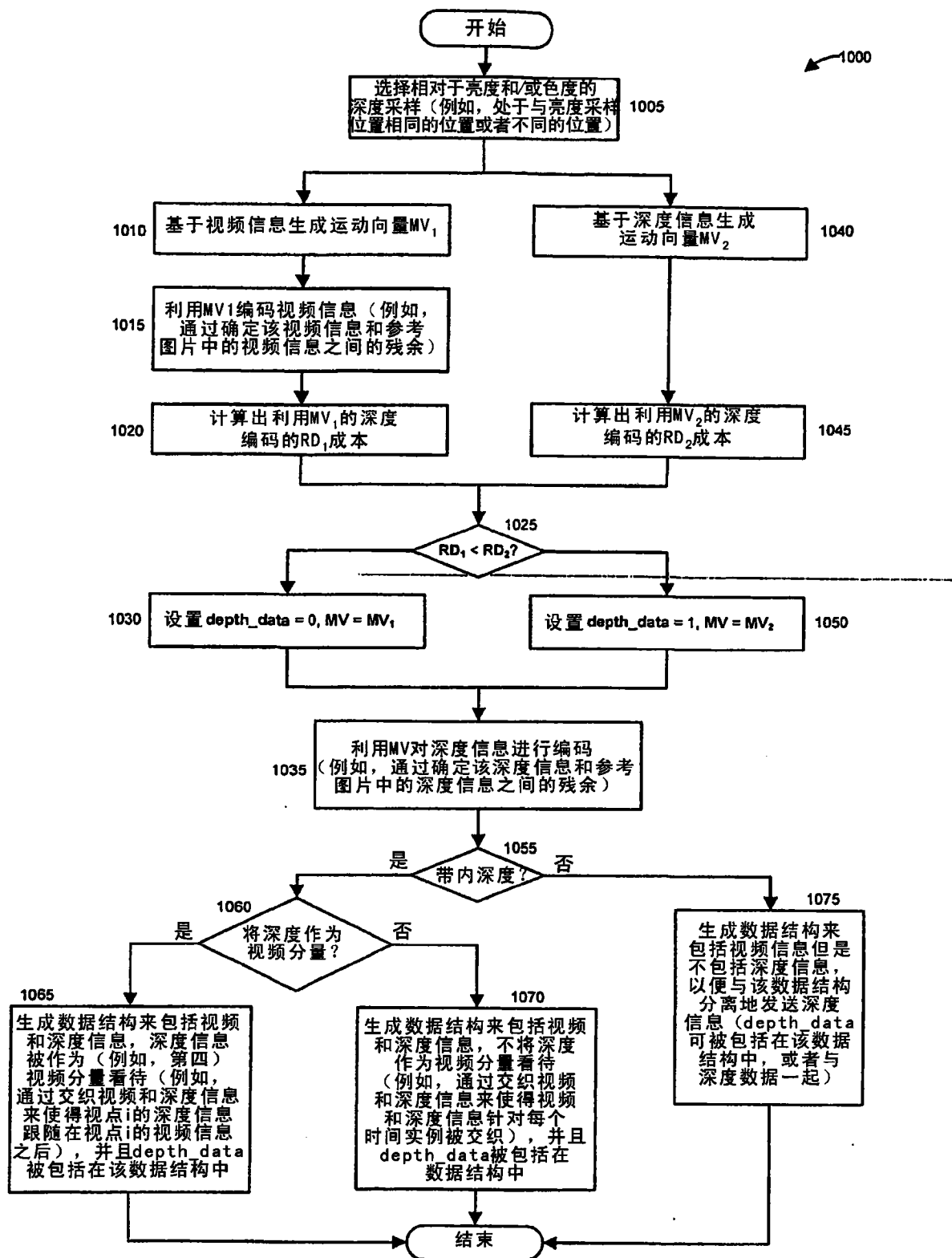


图 10

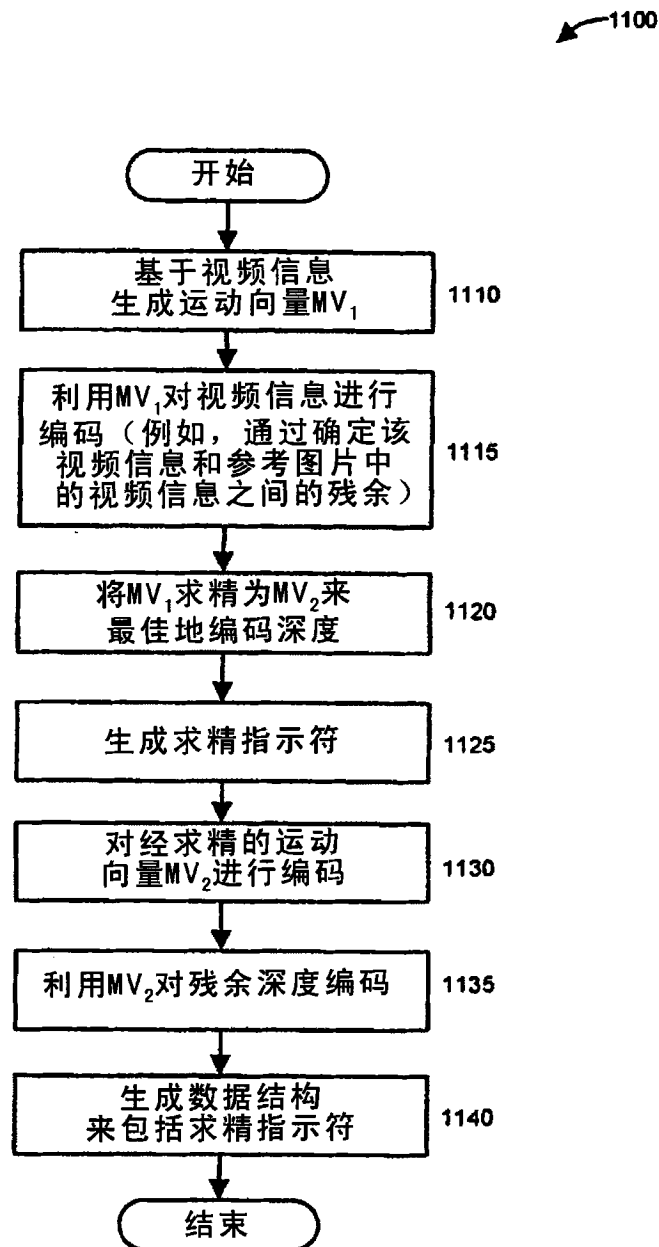


图 11

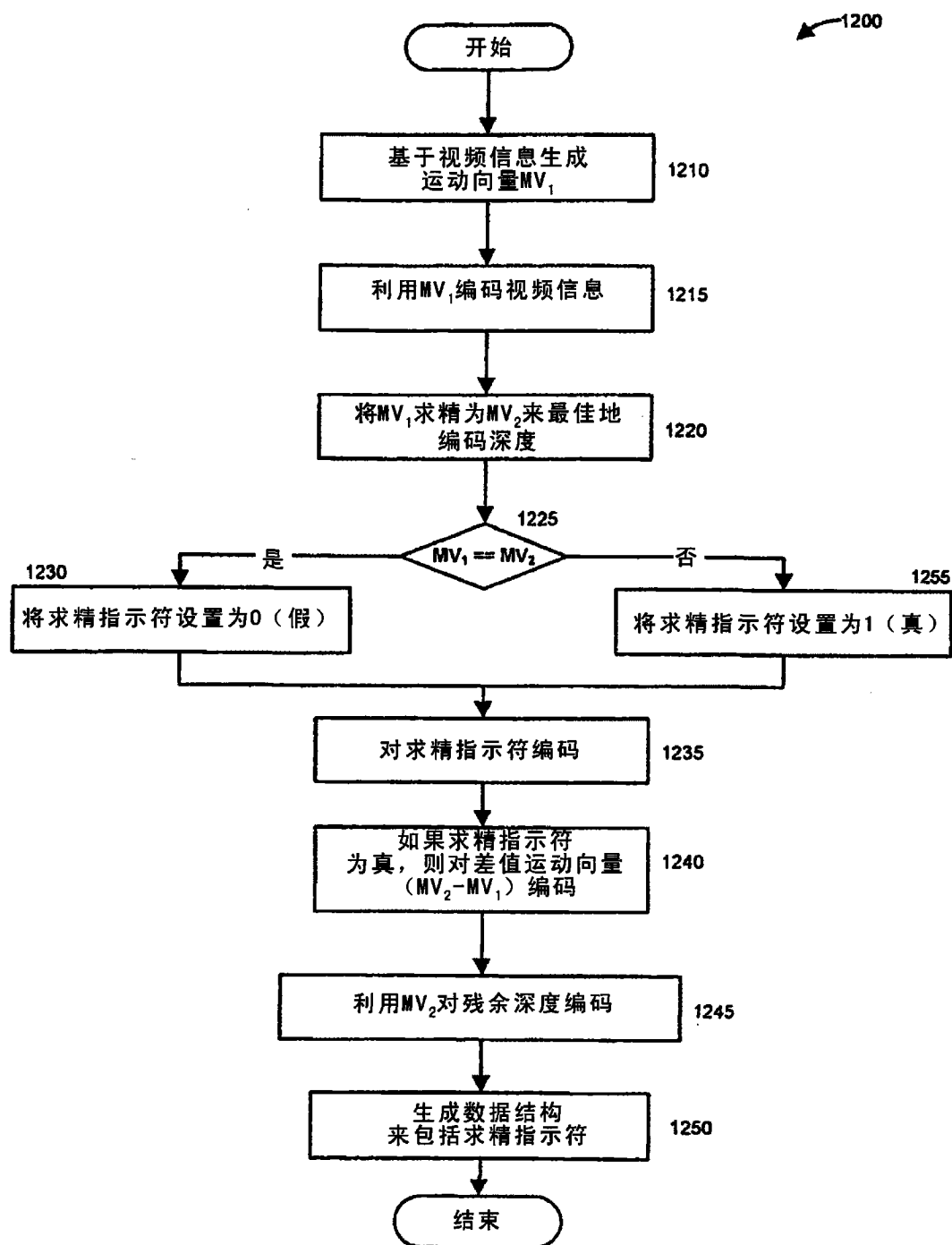


图 12

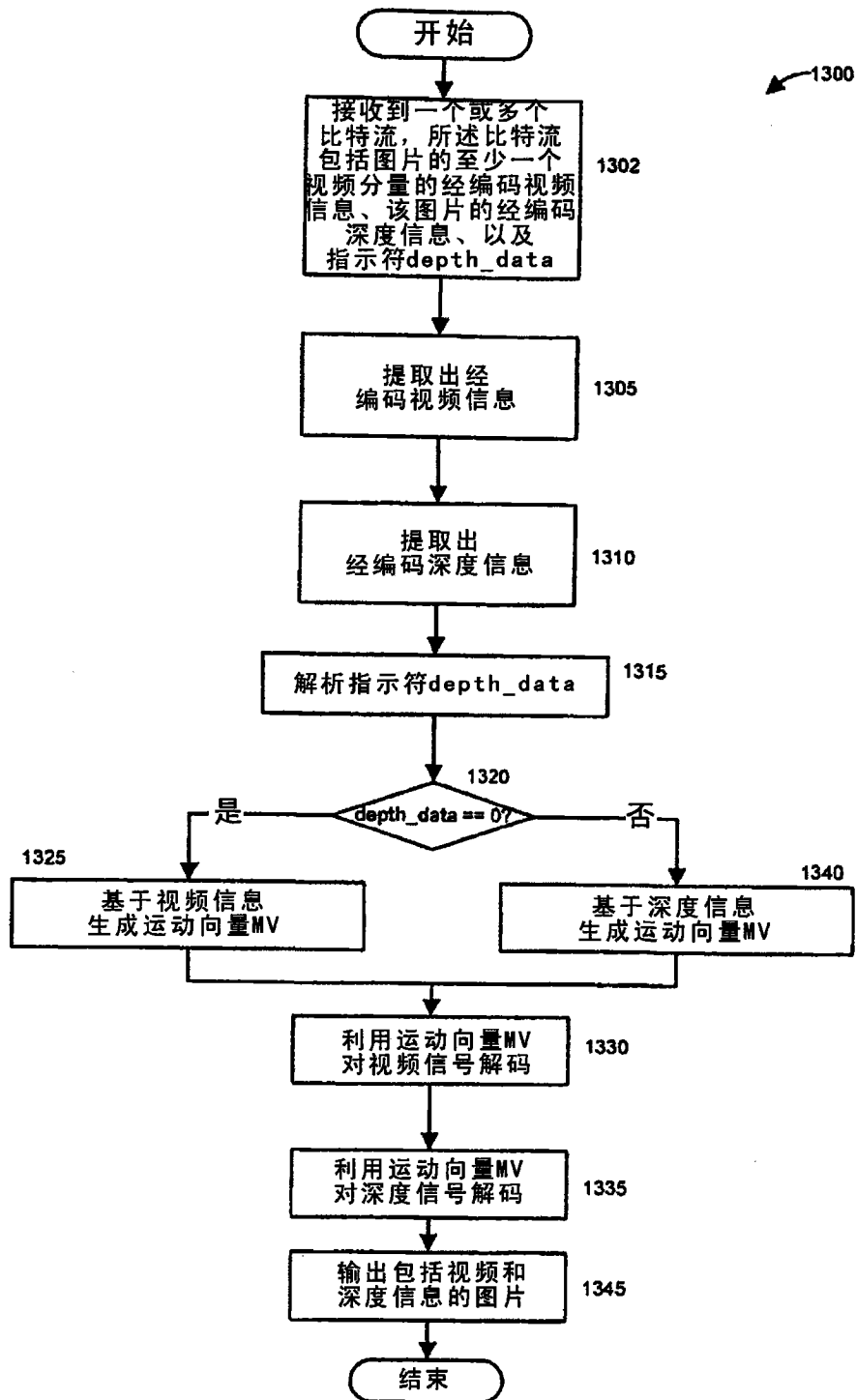


图 13

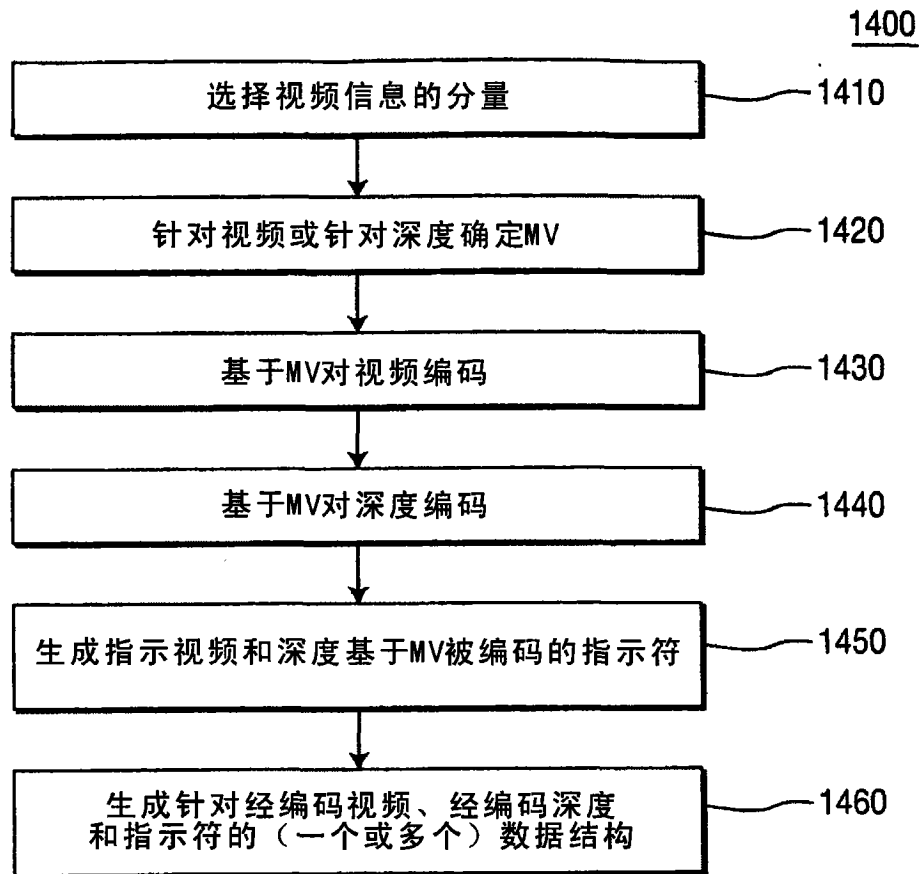


图 14

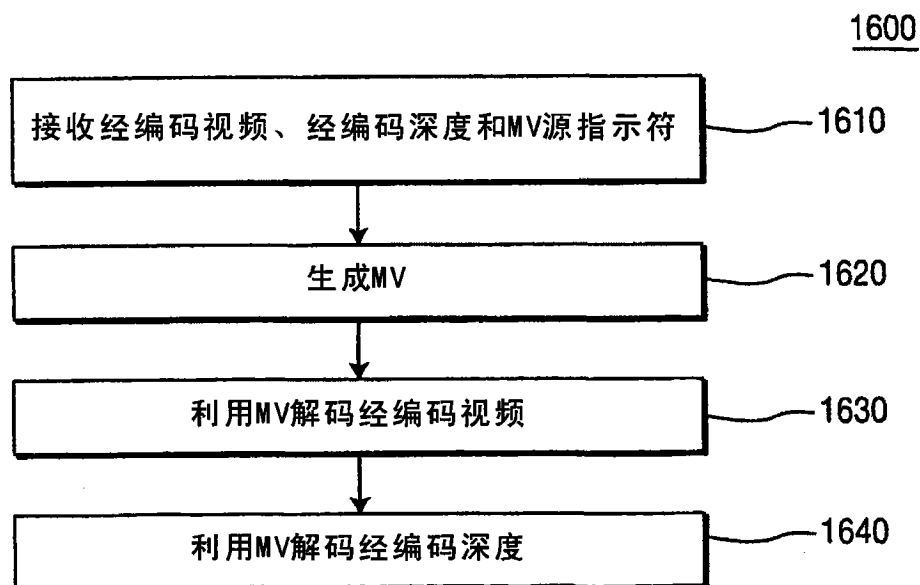


图 16

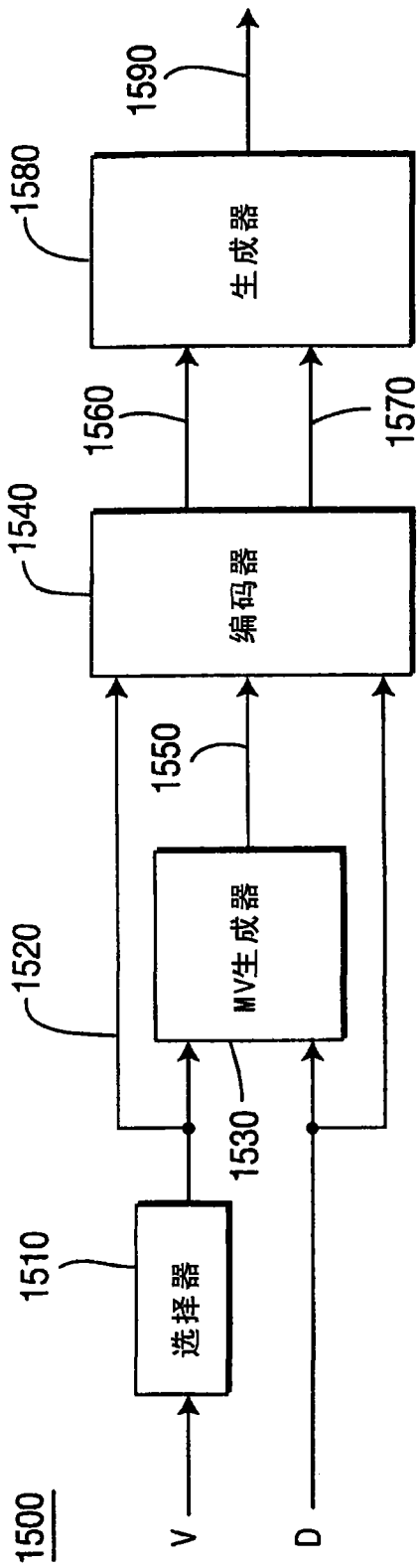


图 15

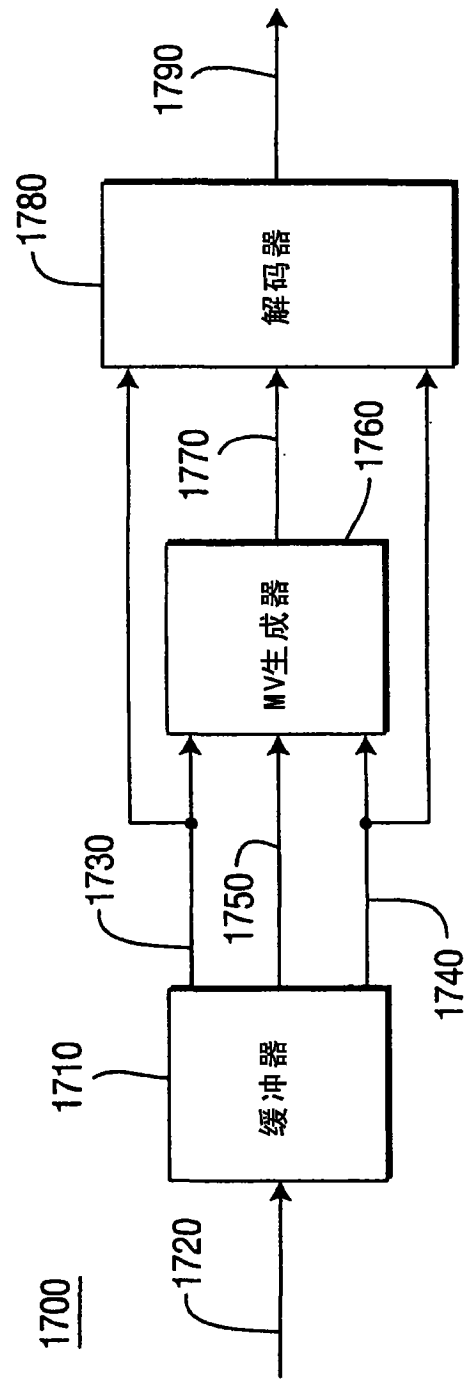


图 17