



(12) 发明专利

(10) 授权公告号 CN 101491097 B

(45) 授权公告日 2011. 12. 14

(21) 申请号 200780025915. 6

(22) 申请日 2007. 07. 12

(30) 优先权数据

60/830, 891 2006. 07. 13 US

60/866, 999 2006. 11. 22 US

60/917, 542 2007. 05. 11 US

(85) PCT申请进入国家阶段日

2009. 01. 08

(86) PCT申请的申请数据

PCT/US2007/073354 2007. 07. 12

(87) PCT申请的公布数据

W02008/008888 EN 2008. 01. 17

(73) 专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72) 发明人 鲍易亮 纳伦德拉纳特·马拉亚什

沙拉什·曼朱纳什 叶琰

(74) 专利代理机构 北京律盟知识产权代理有限
责任公司 11287

代理人 刘国伟

(51) Int. Cl.

H04N 7/26 (2006. 01)

(56) 对比文件

WO 03/039159 A2, 2003. 05. 08, 全文.

WO 03/051060 A1, 2003. 06. 19, 全文.

CN 1726487 A, 2006. 01. 25, 全文.

CN 1698383 A, 2005. 11. 16, 全文.

张方, 吴成柯, 程培星, 肖嵩. 一种改进的可
分级视频编码方法及其网络传输研究. 《电子与信息
学报》. 2005, 第 27 卷 (第 1 期), 第 108-111
页.

审查员 刘江

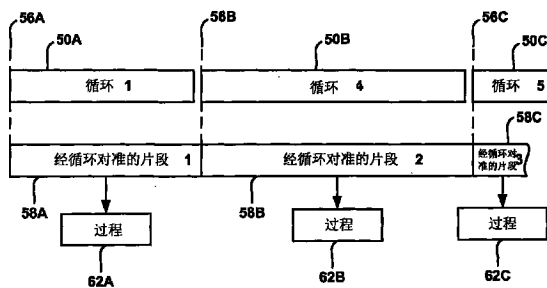
权利要求书 3 页 说明书 29 页 附图 12 页

(54) 发明名称

使用经循环对准的片段的具有细粒度可缩放
性的视频编码

(57) 摘要

本发明说明使用经循环对准的片段 (CAF) 的
FGS 视频编码技术。所述技术可执行对 FGS 视频
数据区块系数及语法元素的基于循环的编码, 并
将循环封装于片段中以供传输。所述片段可经循
环对准, 以使得所述片段中的每一者的有效负载
的开始与所述循环中的一者的开始大致重合。以
此方式, 可经由个别片段容易地存取循环。可使
用向量模式来控制某些循环, 以在移动到另一区
块之前扫描到区块内的预定位置。以此方式, 可
减少循环的数量, 从而减少片段的数量及相关联
的开销。可彼此独立地对 CAF 进行熵编码, 使得
可在无需等待其它片段的解码的情况下容易地存
取及解码每一片段。独立的熵编码可准许对片
段进行并行解码及同时处理。



1. 一种视频编码方法,其包含:

在循环中对细粒度可缩放性 FGS 视频数据区块进行编码,其中每一个所述循环表示对所述视频数据区块中的一者或一者以上的 Z 字形次序扫描的至少一部分;及

将所述循环包封于多个片段中,以使得所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

2. 如权利要求 1 所述的方法,其中包封包含将所述循环包封于多个片段中,以使得所述片段中的每一者的有效负载的开始与所述循环中的一者的开始大致重合。

3. 如权利要求 1 所述的方法,其中所述 FGS 视频数据区块包含变换系数区块,且对所述 FGS 视频数据区块进行编码包括在所述循环中对与所述 FGS 视频数据区块相关联的系数进行编码。

4. 如权利要求 1 所述的方法,其中所述 FGS 视频数据区块包含变换系数区块,且其中所述循环中的至少一些循环跨越与多个所述 FGS 视频数据区块相关联的变换系数。

5. 如权利要求 1 所述的方法,其中所述 FGS 视频数据区块对应于视频数据的渐进细化 (PR) 片。

6. 如权利要求 1 所述的方法,其进一步包含:

对所述片段应用熵编码;及

重设与所述片段中的每一者的所述熵编码相关联的编码上下文。

7. 如权利要求 6 所述的方法,其中重设所述熵编码包含:

在对所述片段中的每一者进行熵编码之后,刷新用于应用所述熵编码的熵编码器;及

在对所述片段中的每一者进行熵编码之后,重设所述熵编码器所使用的所述编码上下文。

8. 如权利要求 1 所述的方法,其进一步包含使用向量模式来控制所述循环中的一者或一者以上,以在移动到所述视频数据区块中的另一区块之前扫描到所述视频数据区块中的每一者内的预定扫描位置。

9. 如权利要求 8 所述的方法,其进一步包含对信息进行编码以发信号通知所述向量模式。

10. 一种视频编码器,其包含:

基于循环的编码模块,其在循环中对细粒度可缩放性 FGS 视频数据区块进行编码,其中每一个所述循环表示对所述视频数据区块中的一者或一者以上的 Z 字形次序扫描的至少一部分;及

分段模块,其将所述循环包封于多个片段中以使得所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

11. 如权利要求 10 所述的编码器,其中所述分段模块将所述循环包封于多个片段中,以使得所述片段中的每一者的有效负载的开始与所述循环中的一者的开始大致重合。

12. 如权利要求 10 所述的视频编码器,其中所述 FGS 视频数据区块包含变换系数区块,且所述基于循环的编码模块在所述循环中对与所述 FGS 视频数据区块相关联的系数进行编码。

13. 如权利要求 10 所述的视频编码器,其中所述 FGS 视频数据区块包含变换系数区块,其中所述循环中的至少一些循环跨越与多个所述 FGS 视频数据区块相关联的变换系数。

14. 如权利要求 10 所述的视频编码器,其中所述 FGS 视频数据区块对应于视频数据的渐进细化 (PR) 片。

15. 如权利要求 10 所述的视频编码器,其进一步包含对所述片段应用熵编码并重设与所述片段中的每一者的所述熵编码相关联的编码上下文的熵编码器模块。

16. 如权利要求 15 所述的视频编码器,其中为重设所述熵编码,所述熵编码器模块在所述片段中的每一者经熵编码之后刷新用于应用所述熵编码的熵编码器,且在所述片段中的每一者经熵编码之后重设所述熵编码器所使用的所述编码上下文。

17. 如权利要求 10 所述的视频编码器,其中所述基于循环的编码模块使用向量模式来控制每一个所述循环,以在移动到所述视频数据区块中的另一区块之前扫描到所述视频数据区块中的每一者内的预定扫描位置。

18. 如权利要求 17 所述的视频编码器,其中所述编码模块对信息进行编码以发信号通知所述向量模式。

19. 一种视频解码方法,其包含:

接收包括对细粒度可缩放性 FGS 视频数据区块进行编码的编码循环的片段,其中每一个所述循环表示对所述视频数据区块中的一者或一者以上的 Z 字形次序扫描的至少一部分;及

对所述片段中的至少一些片段的至少若干部分同时进行解码,其中所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

20. 如权利要求 19 所述的方法,其中所述片段中的每一者的有效负载的开始与所述循环中的一者的开始大致重合。

21. 如权利要求 19 所述的方法,其中所述循环对与所述 FGS 视频数据区块相关联的变换系数进行编码。

22. 如权利要求 19 所述的方法,其中所述循环中的至少一些循环跨越与多个所述 FGS 视频数据区块相关联的变换系数。

23. 如权利要求 19 所述的方法,其中所述 FGS 视频数据区块对应于视频数据的渐进细化 (PR) 片。

24. 如权利要求 19 所述的方法,其中在所述片段中的每一者经熵编码之后重设与所述片段中的每一者的熵编码相关联的编码上下文。

25. 一种视频解码器,其包含:

接收装置,其用于接收包括对细粒度可缩放性 FGS 视频数据区块进行编码的编码循环的片段,其中每一个所述循环表示对所述视频数据区块中的一者或一者以上的 Z 字形次序扫描的至少一部分;及

解码装置,其用于对所述片段中的至少一些片段的至少若干部分同时进行解码,其中所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

26. 如权利要求 25 所述的视频解码器,其中所述片段中的每一者的有效负载的开始与所述循环中的一者的开始大致重合。

27. 如权利要求 25 所述的视频解码器,其中所述循环对与所述 FGS 视频数据区块相关联的变换系数进行编码。

28. 如权利要求 25 所述的视频解码器,其中所述循环中的至少一些循环跨越与多个所

述 FGS 视频数据区块相关联的变换系数。

29. 如权利要求 25 所述的视频解码器,其中所述 FGS 视频数据区块对应于视频数据的渐进细化 (PR) 片。

30. 如权利要求 25 所述的视频解码器,其中在所述片段中的每一者经熵编码之后重设与所述片段中的每一者的熵编码相关联的编码上下文。

使用经循环对准的片段的具有细粒度可缩放性的视频编码

[0001] 相关申请交叉参考案

[0002] 本申请案请求对 2006 年 7 月 13 日提出申请的第 60/830,891 号美国临时专利申请案、2006 年 11 月 22 日提出申请的第 60/866,999 号美国临时申请案、2007 年 5 月 11 日提出申请的第 60/917,542 号美国临时申请案的权益,所述申请案中的每一者的整体内容以引用的方式并入本文中。

技术领域

[0003] 本发明涉及视频编码,更特定来说,涉及数字视频的可缩放编码。

背景技术

[0004] 可将数字视频能力并入各种各样的装置中,所述装置包括数字电视机、数字直播系统、无线通信装置、个人数字助理(PDA)、膝上型计算机、桌上型计算机、数字相机、数字记录装置、蜂窝式或卫星无线电电话及类似装置。在有效创建、修改、传输、存储、记录及播放运动视频序列方面,数字视频装置可提供胜过常规模拟视频系统的显著改善。

[0005] 细粒度可缩放性(FGS)通常是指可任意截断视频位流以在既定位速率范围内实现视频质量的适度降格的能力。经 FGS 编码的视频位流包括具有规定质量的基础层及一个或一个以上链接到所述基础层的增强层。所述增强层包括用以改善所述基础层质量的额外数据。随着经由一个或一个以上增强层接收及解码更多的 FGS 位流,所述经解码视频的质量会改善。

发明内容

[0006] 一般来说,本发明涉及利用经循环对准的片段(CAF)的 FGS 视频编码技术。所述 FGS 视频编码技术可执行对 FGS 视频数据块系数及语法元素的基于循环的编码,并将所得编码循环封装于片段中以便经由网络传输单元(例如,网络抽象层(NAL)单元)进行传输。根据本发明,可循环对准所述片段以使所述片段中的每一者的开始与所述循环中的一者的开始大致重合。以此方式,可经由个别片段迅速地存取编码循环。

[0007] 所述循环中的每一者可表示对一个或一个以上变换系数区块的 Z 字形次序扫描的至少一部分。所述循环中的至少一些循环可跨越与多个区块相关联的编码系数。所述 FGS 视频数据块可形成视频数据的 FGS 片的部分。在一些方面中,所述 FGS 片可以是渐进细化(PR)片,但本发明并非局限于此。可通过向量模式命令来控制所述循环中的至少一些循环,以在移动到所述区块中的另一个区块之前扫描到所述区块中的每一者内的预定扫描位置。以此方式,可减少遍历所述区块所必需的循环数量,从而减少片段的数量并减小开销。

[0008] 可使用单独的编码上下文彼此独立地对所述片段进行熵编码,使得可对多个片段同时进行解码。在对下一片段进行熵编码之前,可重设用于所述片段中的每一者的熵编码,使得可独立于其它片段地对所述片段中的每一者进行熵编码。举例来说,在对所述片段中的每一者进行熵编码之后,可刷新熵编码器并可重设所述熵编码器所使用的编码上下文。

所述片段的独立熵编码可准许对不同片段及所述片段内的循环进行并行解码及独立处理，使得可对多个片段同时进行解码（如果必要）。

[0009] 在一个方面中，本发明提供一种视频编码方法，其包含：在循环中对细粒度可缩放性（FGS）视频数据区块进行编码，并将所述循环封装于多个片段中，以使所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

[0010] 在另一方面中，本发明提供一种视频编码器，其包含：基于循环的编码模块，其在循环中对细粒度可缩放性（FGS）视频数据区块进行编码；及分段模块，其将所述循环封装于多个片段中，以使所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

[0011] 在额外方面中，本发明提供一种视频解码方法，其包含：接收包括对细粒度可缩放性（FGS）视频数据区块进行编码的编码循环的片段；及对所述片段中的至少一些片段的至少若干部分同时进行解码，其中所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

[0012] 在再一方面中，本发明提供一种包含基于循环的解码模块的视频解码器，所述基于循环的解码模块接收包括对细粒度可缩放性（FGS）视频数据区块进行编码的编码循环的片段，并对所述片段中的至少一些片段的至少若干部分同时进行解码，其中所述片段中的每一者的开始与所述循环中的一者的开始大致重合。

[0013] 本发明中所说明的技术可实施为硬件、软件、固件或其任何组合。如果实施为软件，那么可在一个或一个以上处理器（例如，微处理器或数字信号处理器（DSP））中执行所述软件。起初，可将执行所述技术的软件存储在计算机可读媒体中并将其载入处理器进行执行。因此，本发明还涵盖一种包含用以执行本发明中所说明技术的指令的计算机可读媒体。

[0014] 在附图及下文说明中，论述本发明一个或一个以上方面的细节。根据本说明及图式且根据权利要求书，将易知本发明的其它特征、目的及优点。

附图说明

[0015] 图 1 是图解说明视频编码与解码系统的框图。

[0016] 图 2 是图解说明视频位流的多层 FGS 编码的图表。

[0017] 图 3 是图解说明针对视频帧的 FGS 编码的区块系数的 Z 字形扫描的图表。

[0018] 图 4 是图表，其图解说明对以 Z 字形扫描次序布置的视频区块的 FGS 片中的系数进行基于循环的编码的实例。

[0019] 图 5 是图表，其图解说明对以传输次序布置的视频区块的 FGS 片中的系数进行基于循环的编码的实例。

[0020] 图 6 是图表，其图解说明将编码循环封装于正常片段中以便经由网络抽象层（NAL）单元进行传输。

[0021] 图 7 是图表，其图解说明将编码循环封装于经循环对准的片段（CAF）中以便经由网络抽象层（NAL）单元进行传输。

[0022] 图 8 是图解说明将编码循环封装于 CAF 中的 FGS 片编码器的框图。

[0023] 图 9 是图解说明对封装于 CAF 中的编码循环进行解码的 FGS 片解码器的框图。

[0024] 图 10 是图解说明将编码循环封装于 CAF 中的流程图。

- [0025] 图 11 是更加详细地图解说明将编码循环包封于 CAF 中的流程图。
- [0026] 图 12 是图解说明对 CAF 中的循环进行解码的流程图。
- [0027] 图 13 是更加详细地图解说明 CAF 中的循环的解码的流程图。
- [0028] 图 14 是图解说明正常片段及 CAF 的包错误的影响的图表。

具体实施方式

[0029] 一般来说,本发明涉及利用经循环对准的片段(CAF)的FGS视频编码技术。所述FGS视频编码技术可执行对FGS视频数据区块系数及语法元素的基于循环的编码,并将所得编码循环包封于片段中以便经由网络传输单元(例如,网络抽象层(NAL)单元)进行传输。根据本发明,可循环对准所述片段以使所述片段中的每一者的开始与所述循环中的一者的开始大致重合。以此方式,可经由个别片段迅速地存取编码循环。

[0030] 所述循环中的每一者可表示对一个或一个以上变换系数区块的Z字形次序扫描的至少一部分。所述循环中的至少一些循环可跨越与多个区块相关联的编码系数。所述FGS视频数据区块可形成视频数据的FGS片的部分。在一些方面中,所述FGS片可以是渐进细化(PR)片,但本发明中所说明的技术不一定局限于PR片。可通过向量模式命令来控制所述循环中的至少一些循环,以在移动到所述区块中的另一个区块之前扫描到所述区块中的每一者内的预定扫描位置。以此方式,可减少遍历所述区块所必需的循环数量,从而减少片段的数量并减小开销。

[0031] 可以此种方式对所述片段进行熵编码,使得可在其它片段的解码未完成的情况下对片段进行解码。在一些情况下,如果必要可对多个片段同时进行解码。作为实例,可在对下一片段进行熵编码之前重设用于所述片段中的每一者的熵编码,以使所述片段中的每一者具有独立于其它片段的熵编码上下文的熵编码上下文。举例来说,在对所述片段中的每一者进行熵编码之后,可刷新熵编码器并可重设所述熵编码器所使用的编码上下文。所述片段的独立熵编码可准许对不同片段及所述片段内的循环进行并行解码及处理。因此,可对至少一些片段的至少若干部分同时进行解码,例如,同时期或无需等待其它片段的解码的完成。

[0032] 一些片段可能并非彼此完全独立。举例来说,如果区块A已在片段0中完全解码,那么在其它片段中,区块A不存在任何需解码的信息。在此情况下,对稍后片段1中区块A的信息的解码可取决于片段0针对同一区块的解码结果。然而,其它的编码上下文可保持独立。举例来说,片段1的解码不会改变在解码片段1之前片段0经完全解码,也不会改变片段0与片段1是同时解码。因此,至少一些片段可与其它片段的至少若干部分同时解码,因此不必等待其它片段的解码的完成。而是,如上所述,至少一些片段可与其它片段的至少一部分的解码并行同时期地解码,例如,无需等待其它片段的解码的完成。

[0033] 图 1 是图解说明视频编码与解码系统 10 的框图。如图 1 中所示,系统 10 包括源装置 12,所述源装置经由传输信道 16 将经编码的视频传输到目的装置 14。源装置 12 可包括视频源装置 18 及视频编码器 20。目的装置 14 可包括视频解码器 22 及视频显示器装置 24。传输信道 16 可以是有线或无线通信媒体或有线或无线媒体的任何组合。源装置 12 产生供传输到目的装置 14 的视频。然而,在一些情况下,装置 12、14 可以大致对称的方式来操作。举例来说,装置 12、14 中的每一者可包括视频编码及解码组件。因此,系统 10 可支

持视频装置 12、14 之间的单向或双向视频传输以（例如）用于视频流式传输、视频广播或视频电话。

[0034] 视频源 18 可包括视频捕获装置，例如，一个或一个以上摄像机、包含先前所捕获视频的视频档案存储器或来自视频内容提供商的实时视频馈送。作为另一替代方案，视频源 18 可产生基于计算机图形的数据作为所述源视频，或可产生实时视频与计算机所产生视频的组合。在每一情况下，经捕获、预先捕获或计算机产生的视频可由视频编码器 20 来编码以供经由传输信道 16 从视频源装置 12 传输到视频目的装置 14。显示器装置 24 可包括各种显示器装置中的任一种，例如，液晶显示器 (LCD)、等离子显示器或有机发光 (OLED) 二极管显示器。

[0035] 视频编码器 20 及视频解码器 22 可经配置以支持细粒度信噪比 (SNR) 可缩放性 (FGS) 视频编码。举例来说，编码器 20 及解码器 26 可通过支持以 FGS 格式对可缩放增强层进行编码、传输及解码来支持各种程度的可缩放性。对于可缩放编码，基础层以最小的质量等级来携带视频数据。一个或一个以上增强层携带用以支持更高质量等级的额外位流。因此，可提供增强层信息来支持信噪比 (SNR) 可缩放性。另外，在一些情况下，一个或一个以上增强层所携带的额外位流可支持提高的空间可缩放性及 / 或提高的时间可缩放性。

[0036] 视频编码器 20 及视频解码器 26 可根据视频压缩标准来操作，例如，MPEG-2、MPEG-4、ITU-T H. 263 或 ITU-T H. 264/MPEG-4 先进视频编码 (AVC)。更特定来说，视频编码器 20 及视频解码器 26 可经配置以实施所述可缩放视频编码 (SVC) 标准的扩展。尽管图 1 中未显示，但在一些方面中，视频编码器 20 及视频解码器 22 可分别与音频编码器及解码器集成在一起，且包括适当的 MUX-DEMUX 单元或其它硬件及软件，以处置共用数据流或单独数据流中的音频及视频两者的编码。如果适合，MUX-DEMUX 单元可符合 ITU H. 223 多路复用器协议或其它协议，例如用户数据报协议 (UDP)。

[0037] 所述 H. 264/MPEG-4 (AVC) 标准是由 ITU-T 视频编码专家组 (VCEG) 与 ISO/IEC 运动图像专家组 (MPEG) 一起制定的，作为称为联合视频组 (JVT) 的联合合作组织的产物。在一些方面中，可应用本发明中所说明技术以为另外符合 H. 264 标准的装置实施低复杂性的视频可缩放性扩展。举例来说，视频可缩放性扩展可表示对 H. 264 标准或其它标准的未来版本或扩展的可能修改。所述 H. 264 标准说明于 ITU-T 研究组于 2005 年 3 月作出的用于一般视听服务的 ITU-T 推荐 H. 264 先进视频编码中，其在本文中可称为 H. 264 标准或 H. 264 规范，或 H. 264/AVC 标准或标准或规范。

[0038] 联合视频组 (JVT) 继续致力于 H. 264/MPEG-4AVC 的可缩放视频编码 (SVC) 扩展。H. 264/MPEG-4AVC 及正演变的 SVC 扩展两者的规范采用联合草案 (JD) 的形式。由 JVT 产生的联合可缩放视频模型 (JSVM) 实施可缩放视频中所使用的工具，其可在系统 10 中使用以用于本发明中所说明的各种编码任务。关于细粒度 SNR 可缩放性 (FGS) 编码系统的一个实例的详细信息可在联合草案文件中找到，且特定来说可在联合草案 6 (JD6) (托马斯·韦根、加里·沙利文、朱莉·里奇、海科·施瓦兹及马赛厄斯·威恩 (Thomas Wiegand, Gary Sullivan, Julien Reichel, Heiko Schwarz, and Mathias Wien), “联合草案 6: 可缩放视频编码”, JVT-S 201, ISO/IEC MPEG 与 ITU-T VCEG 的联合视频组 (JVT), 2006 年 4 月, 日内瓦) 及联合草案 9 (SVC JD9) (托马斯·韦根、加里·沙利文、朱莉·里奇、海科·施瓦兹及马赛厄斯·威恩 (Thomas Wiegand, Gary Sullivan, Julien Reichel, Heiko Schwarz, and

Mathias Wien),“SVC 修正案的联合草案 9”,JVT-V 201,ISO/IEC MPEG 与 ITU-T VCEG 的联合视频组 (JVT),2007 年 1 月,摩洛哥,马拉喀什)中找到。

[0039] 视频编码器 20 可经配置以(例如)根据 H.264/MPEG-4AVC 标准的 SVC 扩展的 JD 文件来产生 FGS 片。在一些方面中,所述 FGS 片可以是渐进细化 (PR) 片。可几乎任意地截断 FGS 片。随着视频解码器 22 从 FGS 片接收并解码更多的位,其可产生更好的视频质量。JSVM 中的 FGS 编码以以下方式设计:视频质量的改善在整个视频帧上均匀地展开,而并非仅集中在所述帧某一空间区域内。编码及解码 FGS 片的复杂性在计算及存储器要求方面比较高。另外,如所述 JD 文件中所规定,FGS 片的解码过程可能比较复杂。

[0040] 在一些方面中,对于视频广播,本发明涵盖,应用增强的 H.264 视频编码以在使用唯正向链路 (FLO) 空中接口规范的地面移动多媒体多播 (TM3) 系统中传递实时视频服务,“地面移动多媒体多播的唯正向空中接口规范”将出版为技术标准 TIA-1099 (“FLO 规范”)。所述 FLO 规范包括定义位流语法及语义及解码适于 FLO 空中接口的过程的若干实例。另一选择为,视频可根据其它标准进行广播,例如,DVB-H(手持式数字视频广播)、ISDB-T(地面综合服务数字广播)或 DMB(数字媒体广播)。因此,源装置 12 可以是移动无线终端,例如,移动无线无线电电话、视频流式传输服务器或视频广播服务器。然而,本发明中所说明的技术并不局限于任何特定类型的广播、多播或点-对-点系统。

[0041] 视频编码器 20 及视频解码器 22 各自可实施为一个或一个以上微处理器、数字信号处理器、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA)、离散逻辑、软件、硬件、固件或其任何组合。视频编码器 20 及视频解码器 22 中的每一者可包括在一个或一个以上编码器或解码器中,其任一者可集成为相应订户装置、广播装置、服务器或类似装置中的组合式编码器/解码器 (CODEC) 的部分。另外,视频源装置 12 及视频目的装置 14 各自可包括适当的调制、解调制、变频、滤波及放大器组件以传输及接收经编码视频,如果适合,其中包括射频 (RF) 无线组件及天线。然而,为便于图解说明,所述组件未显示于图 1 中。

[0042] 视频序列包括一系列视频帧。视频编码器 20 处理个别视频帧内的像素区块以对视频数据进行编码。根据规定的编码标准,所述视频区块可具有固定或变化的大小,且大小可不同。作为实例,所述 ITU-T H.264 标准支持关于各种区块大小的内部预测(例如,针对亮度分量的 16x16、8x8、4x4,及针对色度分量的 8x8)以及关于各种区块大小的中间预测(例如,针对亮度分量的 16x16、16x8、8x16、8x8、8x4、4x8 及 4x4,及色度分量的对应的缩放大小)。较小的视频区块可提供更好的分辨率,且可用于视频帧中包括较高等级的细节的位置。在所述预测之后,如果使用所述内部_16x16 预测模式,那么可对 8x8 剩余区块或 4x4 剩余区块执行变换,且可向色度分量或亮度分量的 4x4 区块的 DC 系数应用额外变换。

[0043] 图 2 是图解说明视频位流的多层 FGS 编码的图表。如图 2 中所示,视频位流的 FGS 编码产生基础层 30 及一个或一个以上增强层 32A-32N(统称为层 32)。基础层 30 包括一系列帧 34A-34N,所述帧在既定位速率下以最小的质量等级来对所述视频位流进行编码。增强层 32 中的每一者提供额外视频位流信息,解码器 26 可使用所述额外视频位流信息渐进地提高基础层 30 所提供的初始质量。视频解码器 22 所处理的增强层 32 的数量取决于(例如)在既定信道条件或其它限制下,源装置 12 所传输及目的装置 14 所接收的增强层数量。

[0044] 一般来说,根据本发明,由解码器 22 处理的每一相应增强层 32A-32N 或其部分可使从基础层 30 获得的视频的质量(在 SNR 方面)逐渐提高。增强层 32A-32N 还可用于空

间及 / 或时间可缩放性。替代将每一增强层 32A-32N 整体作为离散的增强层进行处理, 系统 10 可准许使用通用 FGS 方法来对可缩放性增强层进行编码、传输及处理。以此方式, 可以细粒度来缩放 SNR, 从而在部分地解码增强层时提供连续的质量改善。在增强层 32A-32N 随着其被解码而提供渐进提高的质量的意义上, 所述增强层可分级。举例来说, 解码所有的增强层将产生最高的质量, 而仅解码第一增强层将产生逐渐提高的质量 (相对于仅解码基础层)。

[0045] 图 3 是图表, 其图解说明针对视频帧的 FGS 编码的区块 46A-46C 中的区块系数的 Z 字形扫描。可使用变换系数的基于循环的编码而非基于区块的编码来产生 SVC 中的 FGS 片 (例如, PR 片)。跨越区块的基于循环的编码可减少视频帧内 SNR 增强的空间集中。在每一循环中, 经 FGS 配置的视频编码器 20 对变换系数区块中最多一个非零系数进行编码且移动到下一区块。如图 3 及 4 中所示, 在循环 0 中, 视频编码器 20 首先对第一区块 46A (即, 区块 0) 中的经编码取块旗标 (CBF) 进行编码。如果区块 0 包含非零有效系数, 那么视频编码器 20 以图 3 中所图解说明的 Z 字形扫描次序对来自同一区块的一个或一个以上系数进行编码, 直到其完成第一有效系数的编码。此过程会继续直到已扫描 (例如) 视频帧或视频帧的部分内的所有区块。然后, 视频编码器 20 进入下一编码循环, 循环 1。如果循环的扫描位置处的系数是细化系数, 那么视频编码器 20 将仅对所述细化系数进行编码并移动到下一区块。

[0046] 在图 3 的实例中, 假设区块 46A、46B 及 46C 是视频帧的增强层 FGS 片中的前三个区块。在此实例中, 每一区块 46A、46B、46C 由 4x4 的变换系数区块来表示。图中描绘区块 46A、46B 及 46C 处于变换域中。因此, 区块 46A、46B、46C 中的每一数或变量是将由无损耗熵编码过程进行编码的经量化系数。标号以字母“S”开头的系数 (例如, 如区块 46A 中的参考编号 41 所指示) 是非零有效系数。由于区块 46A 位于 FGS SNR 可缩放性增强层中, 因此对于区块中的每一系数, 基础层中存在对应的系数。

[0047] 对于非零有效系数, 其值为非零而其对应基础层系数是零。举例来说, 系数“S0,1”对应于区块“0”中 Z 字形扫描索引为“1”的非零有效系数。加标签为“0”的系数是零系数且其对应基础层系数同样为零。标号以字母“R”开头的系数 (例如, 如区块 46A 中的参考编号 43 所指示) 是对应基础层系数为非零 (即, 有效) 的细化系数。所述细化系数的值可以是零或非零。举例来说, 系数“R2,2” (如参考编号 43 所指示) 是区块“2”中 Z 字形扫描索引为“2”的细化系数。在分类增强层 FGS 片系数时, 经常使用有效性地图。此地图指示在基础层中已变得有效的系数的位置。在增强层中, 这些位置处的系数是细化系数。

[0048] 可将本文所说明的视频编码器 20 看作为编码基础层时所使用的熵编码器 (例如, ITU-T H. 264 标准所涵盖的熵编码器) 提供扩展。特定来说, 在增强层中, 视频编码器 20 所使用的区块分割、变换及量化类似于基础层中所使用的区块分割、变换及量化。为更加高效地编码所述零, 可使用语法元素, 例如经编码区块旗标 (CBF) 及区块结束 (EOB)。在基础层编码中还使用类似的语法元素。针对每一区块发送一次 CBF, 且所述 CBF 指示所述区块中非零有效系数的存在。如果所述 CBF 为 0, 那么不存在非零有效系数; 否则, 存在至少一个有效系数。所述 EOB 旗标用来指示刚刚经编码的非零系数是否是扫描次序中的最后非零有效系数。

[0049] 视频编码器 20 进行的基础层编码与增强层编码之间的一个差别是: 在 FGS 层中,

细化系数与其它系数分开;及编码次序。在一些方面中,视频编码器 20 可包括分开的基础层编码及增强层编码模块。在基础层中,在编码下一区块之前,区块经完全编码,从而提供基于区块的编码。然而,对于增强层编码,在循环中使来自不同区块的系数彼此交错,从而提供基于循环的编码。如上所述,在每一循环中,仅对来自既定区块的一些系数进行编码。此外,以此方式,视频编码器 20 确保一旦截断位流,视频质量的改善在整个视频帧上更加空间均匀。

[0050] 图 4 是图表,其图解说明对以 Z 字形扫描次序布置的视频区块 46A-46C 的 FGS 片中的系数及语法元素进行基于循环的编码的实例。在本发明的一些方面中,所述 FGS 片可以是 PR 片,但本发明中所说明的技术并不局限于 PR 片的应用。图 5 是图表,其图解说明对以传输次序布置的视频区块的 FGS 片中的系数及语法元素进行基于循环的编码。图 5 表示图 4 的重新布置,以便将在循环中编码的语法元素及系数布置在同一条线上。在图 4 及 5 中,每一系数由方框表示。在图 4 及 5 的实例中,在单个循环中,针对既定区块最多编码一个非零系数,且循环索引恰好与 Z 字形扫描索引相同。在 FGS 的不同实施方案中,可不要求必须在循环中编码非零系数。另外,可以与有效系数相同的方式处理细化系数。

[0051] 如图 4 中所示,对于循环 0,针对每一区块 46A、46B、46C 发送一次 CBF (CBF0、CBF1、CBF2),且所述 CBF 指示所述区块中存在非零有效系数。举例来说,CBF0/1 对应于第一区块 0,且指示所述区块中存在非零有效系数。另外,在循环 0 中,发送零系数“0”及来自区块 0 的非零系数“S0,1”(其对应于区块“0”中 Z 字形扫描索引为“1”的非零有效系数)。然而,CBF1/0 对应于第二区块(区块 1)且指示所述区块中不存在非零有效系数。因此,在后续循环中,不会针对区块 1 发送系数。循环 0 进一步包括 CBF2/1(其指示区块 2 包括非零有效系数 S2,0)及系数 S2,0 自身。

[0052] 循环 1 包括针对区块 2 的 EOB 旗标 (EOB2/0),所述 EOB 旗标指示循环 0 中刚刚经编码的非零有效系数 S2,0 并不是扫描次序中的最后非零有效系数。在剩余的循环期间,区块的编码继续遵循扫描 Z 字形次序,且如果适合可包括语法元素,例如 CBF 及 EOB。所述处理针对循环 2-15 以所述 Z 字形扫描次序继续进行,其中每一循环对应于图 4 中的一个箭头转换。出于图解说明的目的,图 3、4 及 5 的实例涉及视频帧的增强层中的前三个 4x4 区块。然而,图 4-6 中所图解说明的一般方案可适用于大小较大的区块以及较大数量的跨越视频帧或视频帧的部分的区块。

[0053] 编码及解码 FGS 片的复杂性在计算及存储器要求方面比较高。另外,如 JD 文件中所规定,FGS 片的解码过程可能比较复杂。举例来说,FGS 片的 FGS 编码可能需要大量的存储器来存储中间编码状态信息,例如,先前编码循环中每一区块中所编码的最后系数的位置。另外,FGS 编码可能需要大量的存储器存取,如在基于循环的编码过程期间,视频编码器 20 会频繁地在区块之间跳转。以子带次序存储所述系数可是高效的。在此情况下,可相邻地存储同一频带的系数。然而,以子带次序存储系数可与区块变换不兼容,其可存取不同频率的系数。

[0054] 如果可直接存取来自不同循环的位流,那么可完全解码区块且随后以与基础层中基于区块的处理类似的方式来对所述区块进行处理。因此,迅速地存取来自不同循环的位流的能力可有助于降低 FGS 片编码过程的复杂性。根据本发明的实例性方面,视频编码器 20 可经配置以执行对 FGS 视频数据区块系数的基于循环的编码并将所得编码循环封装于

片段中以供经由网络传输单元（例如，网络抽象层（NAL）单元）进行传输。为准许迅速地存取来自不同循环的位流，可循环对准所述片段以使所述片段中的每一者的开始与所述循环中的一者的开始大致重合。以此方式，可经由所述片段迅速地存取与个别编码循环相关联的位流。

[0055] H. 264/MPEG-4AVC 标准的 SVC 扩展的联合草案定义包含部分 FGS 层的称为“片段”的数据单元。片段可用作包封可废弃 FGS 流的方法。举例来说，在编码侧处，视频编码器 20 可调用正常的 FGS 编码过程来产生一个 FGS 层的位流。在产生所述 FGS 层位流之后，视频编码器 20 可应用位流分段。在解码侧处，解码器 22 将邻近的片段连接并对所述合并的位流进行解码，如同所述位流不曾被分段一样。因此，一个片段的处理通常会需要先前片段的处理的完成。换句话说，无法对普通片段同时进行解码。而是，必须在解码之前将其连接。

[0056] 根据本发明，可对片段进行修改并将其用于提供对与特定 FGS 编码循环相关联的位流的部分的直接存取。特定来说，替代任意地将经编码的位流包封于一系列片段中，视频编码器 20 对分段进行控制以使片段与 FGS 编码循环对准。对准通常是指片段的形成，以使所述片段的开始与所述循环中的一者的开始大致重合。以此方式，可通过识别所述片段的开始来存取与循环相关联的位流的开始。片段可包括位于所述片段的有效负载之前的标头。所述标头可携带管理信息，例如识别片段的信息。片段的有效负载可通常是指所述片段的内容部分，所述内容部分可携带编码循环数据且可位于标头信息之后。根据本发明的各个方面，可将循环包封于多个片段中以使所述片段中的每一者的开始与所述循环中的一者的开始大致重合。因此，大致重合可指使所述片段中的每一者的有效负载（例如，在标头信息之后）的开始与所述循环中的一者的开始大致重合的包封。一般来说，无论是否存在标头信息，每一片段的开始可与循环的开始大致对准，使得可经由相应片段迅速地存取编码循环，从而准许对至少一些片段的至少若干部分同时进行解码。

[0057] 经循环对准的片段（CAF）不同于普通的片段，在普通的片段中，片段的开始可对应于位流中任何未知的点。而是，所述 CAF 与所述循环对准，使得可大致在所述片段的开始存取所述位流的开始。另外，在一些方面中，视频编码器 20 可对分段进行控制，以使每一循环（可能与其它循环一同）包含在单个片段中，而并非散布在连续片段的部分上。产生 CAF 可降低处理复杂性，且准许连续片段的并行处理来作为连续片段的依序处理的替代。

[0058] 为使在无需等待完成先前片段的解码的情况下解码经循环对准的片段（CAF）成为可能，视频编码器 20 可在不使用仅在解码先前片段之后方可获得的信息的情况下对每一 CAF 进行编码。举例来说，在完成第一片段之后，视频解码器 20 可刷新用于对所述片段进行熵编码的熵编码器。另外，在对下一片段进行编码之前，视频编码器 20 可重设与熵编码相关联的编码上下文。通过重设编码上下文并刷新所述熵编码器，视频编码器 20 产生可在无需等待其它片段的解码的情况下迅速地存取及解码的 CAF，从而实现 CAF 的并行、同时处理。举例来说，在一个 CAF 的解码可与另一 CAF 的至少一部分的解码可同时期执行的意义上，所述解码可同时进行。以此方式，可在无需等待完成先前片段的解码的情况下对 CAF 进行解码。此与正常片段形成对比，正常片段需在解码之前进行重新汇编。

[0059] 视频编码器 20 可将熵编码实施为上下文自适应可变长度编码（CAVLC）或上下文自适应二进制自适应编码（CABAC）。与普通片段相比，由于与熵编码相关联的刷新操作及编码上下文的重设，CAF 可呈现一些开销。CAVLC 仅受刷新操作的影响，所述刷新操作可导致

平均约为一个半字节的额外开销。对于 CABAC, 由于编码上下文的重设, CAF 呈现额外的不利结果。额外 NAL 标头及片段标头所需要的开销通常与正常片段及 CAF 相同。

[0060] 如果将每一循环编码为 CAF, 那么由于每一循环的长度倾向于减小, 因此片段的大小可随着循环索引的增大而快速减小。出于此原因, 可期望将一些循环分组在一起以减小总体开销。特定来说, 如果将一些循环分组在一起, 那么可在所述循环之间共享与所述片段相关联的开销。如果不将循环分组在一起, 那么许多循环将被携载于其自己的片段内, 因此产生每一循环的片段开销。实际上, 可使用 (例如) 向量模式 (例如, 如 JVT 最近在联合草案 6 (JD6) 中引入) 来实现对共用片段内的循环的分组。

[0061] 将向量模式引入 JD6 中以便降低 FGS 编码器的复杂性。在所述向量模式中, FGS 编码器将在既定循环中处理区块, 直到其到达预设的扫描位置, 之后才移动到下一区块。换句话说, 替代在编码第一有效系数之后移动到下一区块, 循环将在区块内继续进行直到到达所述预设扫描位置的点。通过所述向量模式, 可增大区块内既定循环的扫描深度。因此, 可减少 FGS 编码器扫描区块的平均次数。所述向量模式可定义确定将在既定循环中扫描的系数数量的向量长度。如果所述向量长度为 1, 那么所述向量模式没有效果, 且所述 FGS 编码器正常地操作。如果所述向量长度等于区块中系数的数量, 那么所述 FGS 编码器会退化为正常的基于区块的编码器。

[0062] 所述向量模式不改变 FGS 编码器多次访问区块的事实。然而, 其可通过减少所述 FGS 编码器访问区块的次数及减小编码循环的数量将基于循环的编码的复杂性降低到某一程度。当与经循环对准的片段一同使用时, 可使用经修改的向量模式来产生类似大小的片段, 从而减小分段的开销。替代具有非常小的循环大小的多个片段, 可使用所述向量模式来减少循环的数量并增加每一循环的深度, 使得减少容纳所述循环所需要的片段数量。由于片段数量减少, 因此减少了片段开销的实例数量, 从而提高效率。

[0063] 在 JD6 中所说明的向量模式中, 在序列参数组中定义向量长度。在经修改的向量模式中, 如本发明中所说明, 可在片标头中而并非在所述序列参数组中规定向量长度。使用所述向量长度, 视频编码器 22 增加个别循环的扫描深度以在既定循环中编码更多的系数, 从而减少片段的数量。作为实例, 所述特定向量长度可以是固定的或可基于内容、信道条件、处理负载或其它因素动态地调节。所述向量长度可对于所有的循环均相同, 对于选定的循环不同, 或从第一个循环到最后一个循环渐进地改变。

[0064] 图 6 是图表, 其图解说明将循环封装于普通片段中以供经由网络抽象层 (NAL) 单元进行传输。如图 6 中所示, 不考虑循环边界, 在连续片段 52A-52C 之间对连续循环 50A-50C 所携载的位流进行划分。结果, 每一既定片段 52 包含与一个或一个以上循环 50 相关联的位流的一部分, 且不提供任何关于既定循环的位流开始的指示。而是, 既定循环 50 的不同部分可跨越连续片段 52 的边界 56A-56C 而分布。另外, 每一片段 52 可在循环 50 的位流内的未知、通常任意点处开始, 因此无法经由片段直接对循环进行存取。

[0065] 在循环 50 的位流跨越连续片段 52 的情况下, 需以独立的方式来处理连续片段。特定来说, 视频解码器 22 将连续片段 52 所携载的位流的任意部分连接以复制所述循环的位流。如果第一及第二片段 52A 及 52B 携载循环 1 (50A) 的部分, 那么 (例如) 会处理所述两个片段并将其连接以复制循环 1 的位流。为支持并置, 在可对下一片段 52B (FRAG 2) 进行解码及处理之前, 必须先对片段 52A (FRAG 1) 进行解码及处理 (54A)。

[0066] 因此,无法对普通片段 52B 同时进行解码及处理。另外,稍后的片段 52B 通常会依赖于早先片段 52A 的熵编码。因此,片段 52B 必须等待完成先前片段 52A 的解码及处理。因此,图 6 中处理操作 54A、54B、54C 的布置表示稍后片段 52 在完成先前片段的处理的基础上的依序独立的解码及处理。在计算及存储器要求方面,图 6 的实例中所示的片段 52 的依序处理可呈现相当大的计算复杂性。

[0067] 图 7 是图解说明根据本发明的一方面将循环包封于经循环对准的片段 (CAF) 中以供经由网络抽象层 (NAL) 单元进行传输。在图 7 的实例中,视频编码器 20 对分段进行控制以使每一片段是以循环 50A-50E 的开始为开始的经循环对准的片段 58A-58C。另外,作为一选择,可使用向量模式来制定图 7 实例中的循环 50,以便扩展所述循环中的至少一些循环的长度,从而减少片段的总体数量及携载所述循环所需要的相关联开销。

[0068] 一般来说,每一 CAF 58 携载大致在所述 CAF 开始时开始的循环 50。举例来说,视频编码器 20 可将所述循环包封于多个 CAF 58 中,以使所述片段中的每一者的开始(例如,所述片段中的每一者的有效负载的开始)与所述循环中的一者的开始大致重合。在一些情况下,所述有效负载的开始可在标头(其消耗片段 58 内的空间)之后。通过使循环 50 的开始与片段 58 的开始大致对准,便可迅速地确定与循环 50 相关联的位流的开始,从而准许存取个别循环。由于每一 CAF 58 携载可直接存取的循环 50,因此不需要将连续片段连接来复制与循环 50 相关联的位流。而是,视频编码器 22 可经配置以便如处理操作 62A-62C 所表示(例如,并行)对片段 58 同时进行解码及处理。此外,片段 58 的解码及处理可与一个或一个以上其它片段的至少一部分解码及处理同时期地并行地进行。然而,一些片段 58 仍可依序地解码及处理。

[0069] 为使 CAF 可在无需等待完成先前片段的情况下解码,视频编码器 20 可在不使用仅在先前片段经解码后方可获得的信息的情况下对所述 CAF 进行编码。因此,除了使所述循环开始与所述片段开始对准以外,视频编码器 20 还可重设与连续编码的片段之间的熵编码相关联的编码上下文。特定来说,在完成第一片段的熵编码之后,视频编码器 20 刷新所述 CAVLC 或 CABAC 熵编码器并重设编码上下文。以此方式,经循环对准的片段并不依赖于来自先前编码的片段的任何编码信息,且可与其它片段并行地解码及处理。

[0070] 图 8 是图解说明将循环包封于经循环对准的片段中的 FGS 片编码器 64 的框图。FGS 片编码器 64 可形成数字视频编码器(例如,图 1 的视频编码器 20)的部分。如图 8 中所示,FGS 片编码器 64 可包括变换模块 66、量化模块 68、基于循环的编码模块 70、经循环对准的分段模块 72 及熵编码模块 74。变换模块 66 对视频编码器 22 所产生的 FGS 片的源视频残余应用空间变换以产生变换系数区块。量化模块 68 使所得变换系数量化。基于循环的编码模块 70 扫描所述变换系数区块以便(例如)以类似于图 3-5 中所图解说明的方式来产生编码循环。

[0071] 在本发明全文中,将不同的特征描绘成模块、区块或组件旨在突出视频编码器 20 或视频解码器 22 的不同功能方面,而未必暗示必须通过单独的硬件及/或软件组件来实现所述模块。而是,可将与一个或一个以上模块(例如,变换模块 66、量化模块 68、基于循环的编码模块 70、经循环对准的分段模块 72 及熵编码模块 74)相关联的功能性集成在共用或单独的硬件及/或软件组件中。在一些情况下,可通过共用或单独的软件或软件模块来实现所述特征,所述共用或单独的软件或软件模块包括经配置以致使计算机执行归因于所述

特征的功能的码。

[0072] 经循环对准的分段模块 72 将与编码循环相关联的位流截成若干片段。每一片段包封循环,以使所述循环位流的开始与所述片段所携带的有效负载的开始大致重合。经循环对准的分段模块 72 可经配置以分析所述位流并检测循环边界以触发经循环对准的分段。熵编码模块 74 对经循环对准的分段模块 72 所产生的片段中的每一者应用熵编码(例如,CAVLC 或 CABAC 编码)。显著地,为消除连续片段之间的相互依赖性,熵编码模块 74 可刷新其先前的统计数据并在每一片段经熵编码之后重设其编码上下文。以此方式,可对连续片段中的至少一些连续片段同时进行解码及处理。可将所述经熵编码的片段放置在网络传输单元(例如,网络抽象层(NAL)单元)中,以便将其从源装置 12 传输到目的装置 14。

[0073] 图 9 是图解说明对包封于经循环对准的片段(CAF)中的循环进行解码的 FGS 片解码器 75 的框图。FGS 片解码器 75 可形成视频解码器(例如,图 1 的视频解码器 22)的部分。如图 9 中所示,FGS 片解码器 75 可包括熵解码模块 76、经循环对准的解分段模块 80、基于循环的解码模块 80、逆量化模块 82 及逆变换模块 84。熵解码模块 76 对经编码的视频片段应用熵解码。显著地,可并行地对所述经编码的视频 CAF 进行熵解码,因为所述 CAF 并非使用与其它 CAF 中的编码上下文相同的编码上下文来编码,且每一 CAF 提供对特定循环的位流的直接存取。

[0074] 经循环对准的解分段模块 78 处理所述经熵解码的片段以产生经编码的视频位流。基于循环的解码模块 80 扫描所述视频位流以产生经量化的变换系数。逆量化模块 82 解量化所述变换系数。逆变换模块 84 对所述经解量化的变换系数应用逆变换以复制所述残余。通过 CAF, FGS 片的解码过程可得到简化而与离散层的解码过程非常相似。作为实例,在一些方面中,所述 FGS 片可以是 PR 片。如本发明中所说明,CAF 还可显著地简化所述 FGS 片的语法的规范。

[0075] 图 10 是图解说明如本发明中所说明将循环包封于经循环对准的片段(CAF)中的流程图。可在与视频编码器 20 相关联的 FGS 片编码器(例如,图 8 的 FGS 片编码器 64)内执行 CAF 的产生。在一些方面中,FGS 片编码器 64 可以是 PR 片编码器。如图 10 中所示,FGS 片编码器 64 在循环中对 FGS 视频数据区块进行编码(86),将所述循环包封于经循环对准的片段中(88),并对所述片段进行熵编码(90)。如本发明中所说明,在与循环相关联的位流的开始与用以包封所述循环的相关片段的有效负载的开始大致重合的意义上,所述片段是经循环对准的。

[0076] 图 11 是更加详细地图解说明将循环包封于经循环对准的片段中的流程图。此外,可在与视频编码器 20 相关联的 FGS 片编码器(例如,图 8 的 FGS 片编码器 64)内执行 CAF 的产生。如图 11 中所示,FGS 片编码器 64 获得剩余视频区块的变换系数(92),且在跨越所述区块的循环中 Z 字形扫描所述变换系数(94)。FGS 片编码器 64 产生片段,使所述片段有效负载的开始与所述循环的开始对准(96),且将所述循环放置在所述片段中(98)。FGS 片编码器 64 对所述片段进行熵编码(100),且然后刷新所述熵编码器统计数据,并在所述片段经熵编码之后重设与熵编码相关联的编码上下文(102)。以此方式,将要熵编码的下一片段不需要仅在先前片段经完全解码后方可获得的任何信息。因此,FGS 片解码器 75 可同时并行地解码多个片段,以使一些片段的至少若干部分的解码至少部分地与一个或一个以上其它片段的至少若干部分的解码同时期进行。

[0077] 如果存在更多将要编码的循环 (104), 那么 FGS 片编码器 64 继续跨越剩余区块扫描系数 (94) 并重复操作 96、98、100、102。如果基于循环的编码完成, 那么 FGS 片编码器 64 从 (例如) 视频帧的另一部分或从后续帧获得更多剩余区块的变换系数, 并重复操作 96、98、100、102。在整个过程中, FGS 片编码器 64 产生 CAF, 使得可同时处理所述片段且可直接存取与所述片段中的循环相关联的位流。产生 CAF 可导致简化的处理及减小的计算及存储器开销。

[0078] 图 12 是图解说明经循环对准的片段 (CAF) 中的循环的解码的流程图。可在与视频解码器 22 (例如, 图 9 的 FGS 片编码器 75) 相关联的 FGS 片解码器内执行 CAF 内循环的解码。在图 12 的实例中, FGS 片解码器 75 接收具有经由信道 16 传输的传入视频的 CAF (106)。FGS 片解码器 106 对所述 CAF 进行熵解码 (108) 并对所述 CAF 中的至少一些 CAF 的至少若干部分同时进行解码 (110), 即, 没有仅在其它 CAF 经完全解码之后方可获得的信息。以此方式, FGS 片解码器 75 并行地对所述 CAF 进行解码及处理。因此, 可对多个 CAF 同时并行地 (替代依序地) 进行处理及解码。一些 CAF 可彼此完全独立。在其它情况下, 解码稍后 CAF 中区块的信息可取决于同一区块的先前 CAF 的解码。然而, 在此情况下, 所述 CAF 的其它编码上下文可保持独立, 使得无论先前 CAF 是否已经解码或所述先前 CAF 与稍后 CAF 是否同时解码, 所述稍后 CAF 的解码均相同。

[0079] 图 13 是更加详细地图解说明 CAF 中的循环的解码的流程图。如图 13 中所示, FGS 片解码器 106 可获得 CAF (112), 并对所述 CAF 进行熵解码 (114)。FGS 片解码器 75 可直接存取与所述 CAF 所携带的循环相关联的位流 (116), 并对所述循环位流进行解码 (118) 以产生对应的剩余视频数据。如果更多的 CAF 可用 (120), 那么 FGS 片解码器 75 获得下一 CAF (112) 并重复图 13 的操作 (即, 操作 114、116、118)。

[0080] 图 14 是图解说明正常片段及经循环对准的片段 (CAF) 的包错误的影响的图表。根据本发明, 除了可简化处理及降低计算机及存储器要求以外, 使用 CAF 还可显著地改善 FGS 层的错误弹性。图 14 显示一系列正常片段 (FRAG 0、FRAG 1、FRAG 2) 及一系列 CAF (CAF 0、CAF 1、CAF 2)。当第一正常片段 FRAG 0 的部分如 X 所指示被破坏时, 所述破坏致使后续片段 FRAG 1 及 FRAG 2 完全不可用。换句话说, 先前编码的 FRAG 0 中任何程度的破坏可导致稍后编码的 FRAG 1 及 FRAG 2 的总体破坏。对于正常片段, 先前 FRAG 0 及稍后片段 FRAG 1 及 FRAG 2 实际上是从同一编码过程中产生。因此, 正常 FRAG 0 末尾处的错误会破坏 FRAG 1 及 FRAG 2 中的所有信息, 因为出于熵编码的目的, FRAG 1 及 FRAG 2 依赖于 FRAG 0, 且可包含共用循环位流的不同部分。

[0081] 相反, 对于 CAF, 一个片段的破坏不会总体地破坏其它片段。举例来说, 如果 CAF 0 末尾处存在错误, 那么所述错误将仅影响 CAF 1 及 CAF 2 中的对应部分。所述 CAF 可同时进行编码且可携带单独编码循环的位流。因此, 没有任何 CAF 需要来自另一 CAF 的用于熵解码的信息。另外, 编码循环 (例如, 针对特定变换系数区块) 末尾处的信息损失将仅影响所述区块的后续扫描信息。因此, 与后续一组区块 (被破坏的区块除外) 的扫描相关联的位流将保持完整无缺并可直接地存取及正确地解码。因此, 使用本发明中所说明的 CAF 可提高经编码视频中的错误弹性。

[0082] 为支持 CAF, 可期望或需要上文所解释的特定 FGS 编码方案进行额外的修改。举例来说, 根据 JD 6, 使用变量“chromaStartCycle (色度开始循环)”来相对于亮度系数的传

输延迟色度系数的传输。在 JSVM 中的当前实施方案中,基于从第一编码循环收集的统计数据来计算 chromaStartCycle 的值。然后,在第一编码循环末尾处,FGS 片编码器发送所得 chromaStartCycle 值。如果对多个 CAF 同时进行解码,那么可期望在第一片段的片标头中发送所述 chromaStartCycle 值。因此,视频编码器 22 可进一步经配置以在所述第一片段的片标头中发送所述 chromaStartCycle 值。

[0083] 另一可能的修改还涉及色度系数的传输。在原始 JSVM 中,最后的色度 AC 编码循环可能不与最后的亮度编码循环重合。由于此原因,视频编码器 22 可经配置以强制执行约束,以使色度系数的传输不晚于亮度系数。以此方式,视频编码器 22 可确保最后的 AC 编码循环与最后的亮度编码循环重合。

[0084] 作为进一步细化,可在视频编码器 22 中对熵编码进行调节。可针对 FGS 位流的产生,将基于上下文的 VLC 编码方案用于熵编码。为尽可能多地解耦所述 CAF,可使用当前的扫描索引来检索所述 VLC 参数。所述第一反复可以是例外,因为其它编码反复中不存在对应于 EOB 的符号。如果细化系数与有效系数分开编码且在第一扫描或前几个扫描位置处存在一些细化系数,那么第一零游程可以非零扫描位置开始。

[0085] 可使用若干种不同的方法来处置此情况。根据第一方法,可在视频编码器 22 中分配单独频段以总是处置区块的第一反复。根据第二方法,视频编码器 22 可使用扫描索引来检索 VLC 参数,但利用以下知识:所述第一反复并不具有 EOB,以使符号组不具有插入的 EOB。根据第三方法,视频编码器 22 可将第一反复中的经编码区块旗标 (CBF) 当作 EOB。如果所述 CBF 为 1,那么针对所述第一反复发送值为 0 的 EOB。否则,针对所述第一反复发送值为 1 的 EOB。使用此第三方法,所述第一反复与其它反复之间可几乎不存在差别。

[0086] 可提供用以改善 FGS 编码的性能的额外技术。如上所解释,可期望在编码任何有效系数之前编码所述 CBF,因为所述 CBF 指示相关区块中是否存在任何有效系数。在一些情况下,所述区块可具有一些首项细化系数。举例来说,如果所述第一系数正好是细化系数且所述细化系数与有效系数分开编码,那么可在发送第一细化系数之后发送所述区块的 CBF。结果,所述 CBF 可能并非总是在第一编码循环中进行编码。如果所述 CBF 未经编码,那么使用 CABAC 熵编码器的 CAF 实施方案可产生一些问题。在 CABAC 熵编码中,基于相邻区块的 CBF 的值在上下文中对所述 CBF 进行编码。另外,对于编码不同区块类型的 CBF (例如,亮度 4x4 区块、色度 AC 区块、色度 DC 区块等等),使用不同的上下文。如本发明中所说明,在低复杂性 FGS 编码系统中,如果经循环对准的片段中的所有经编码区块旗标在同一区块的一些细化系数之后编码,那么视频编码器 22 可在单个额外上下文中对其进行编码。

[0087] 作为实例,假设区块中存在一个细化系数且所述细化系数是在第一扫描位置处。另外,假设所述区块中还存在一些额外的有效系数。在此情况下,视频编码器 22 可经配置以首先对所述细化系数进行编码,继而对所述区块的 CBF 进行编码。可在新定义的上下文中对所述 CBF 进行编码。然后,对所述区块中的有效系数进行编码。

[0088] 基于 CABAC 的 CAF 实施方案中可能出现的另一问题还涉及上下文定义。CABAC 熵编码使用基于扫描位置所定义的上下文对所述有效性旗标及最后的有效性旗标进行编码。在简化的 CAF 实施方案的情况下,可期望单独地为视频编码器 22 内的每一片段维持类似的上下文组。维持单独的上下文组可引起额外的实施方案成本。然而,如果在不存在经循环对准的片段时使用更多的编码上下文来编码相同量的系数,那么可能存在导致较低编码性

能的上下文稀释问题。在视频编码器 22 的 FGS 片编码器中,可将多个有效性旗标的编码上下文分组。

[0089] 举例来说,如果开始循环扫描位置是 3 且片段的向量长度是 3,那么使用原始的有效性编码上下文来扫描位置 3、4 及 5。除扫描位置 5 以外,可以各种方式来将有效性旗标的编码上下文分组。一个实例是设定阈值 $C \geq 6$ 。如果扫描位置在 C 之前或在 C 处,那么可使用单独的上下文来对所述有效性旗标进行编码。然后,以单个上下文对所有对应于 C 之后的扫描位置的有效性旗标进行编码。可使用类似方法来定义用于对最后的有效性系数旗标进行编码的额外编码上下文。在另一实例中,所述扫描可限制在由向量长度所规定的范围内,而并非下一非零系数的位置。

[0090] 以下表 1 中论述的是语法修改的实例,可实施所述语法修改以支持根据本发明一些方面的 CAF。可相对于 H. 264/MPEG-4AVC 标准的 SVC 扩展的联合草案 6 (JD6) 中所论述的适用语法或相对于 JVSM 中所另外提出的语法作出所述语法修改。对于所述特定 FGS 编码方案,所述 FGS 片还可称作 PR(渐进细化)片。

[0091] 在本发明的各个表格中,所有语法元素可具有(例如)体现于 JVSM 或 JD6 中的 ITU-T H. 264 标准或 SVC 扩展中所指示的相关语法及语义,如同所述语法元素说明于 H. 264 标准中一般,除非另有规定。大体来说,H. 264 标准或 JD6 中未说明的语法元素及语义在本发明中予以说明。

[0092] 在本发明的各个表格中,标记为“C”的列列举了 NAL 单元中可存在的语法元素的类别,其可符合 H. 264 标准中的类别。另外,可存在具有语法类别“A11(所有)”的语法元素,其由 FGS 片 NAL 单元中所包含的原始位序列有效负载(RBSP)数据结构的语法及语义确定。

[0093] 存在或不存在任何特定列举类别的语法元素是根据相关联 RBSP 数据结构的语法及语义确定的。描述符列规定描述符,例如, $f(n)$ 、 $u(n)$ 、 $b(n)$ 、 $ue(v)$ 、 $se(v)$ 、 $me(v)$ 、 $ce(v)$, 所述描述符通常可符合 H. 264 标准或 JD6 中所规定的描述符,除非本发明中另有规定。

[0094] 表 1

[0095] 用以支持经循环对准的片段的语法修改

[0096]

seq_parameter_set_rbsp() {	C	描述符
profile_idc	0	u(8)
.....		
if(profile_idc == 83) {		
seq_parameter_set_svc_extension() /* specified in Annex G */		
}		
.....		
seq_parameter_set_svc_extension() {	C	描述符
.....		
pr_info_present_flag [Notes : may be qualified by profile_idc once profiles are defined]	2	u(1)
if(pr_info_present_flag) {		
pr_cycle_aligned_fragment_flag	2	u(1)
num_pr_vector_modes_minus1	2	ue(v)
for(i = 0; i <= num_pr_vector_modes_minus1; i ++) {		
pr_coding_mode[i]	2	u(1)
if(pr_coding_mode[i] == 0) {		
grouping_size_minus1[i]	2	ue(v)
} else {		
NumPosVector[i] = 0		
remVectorLen = 16		
do {		
reverse_pr_vector_len_minus1[i][NumPosVector[i]] = 0		
if(remVectorLen > 1)		
reverse_pr_vector_len_minus1[i][NumPosVector[i]]	2	u(v)
remVectorLen - =		
reverse_pr_vector_len_minus1[i][NumPosVector[i]] + 1		
NumPosVector[i] ++		
} while (RemVectorLen > 0)		
}		
}		
}		
.....		
}		

[0097] 表 1(续)

[0098]

slice header in scalable extension() {	C	描述符
first mb in slice	2	ue(v)
slice_type	2	ue(v)
.....		
if(slice_type == PR && fragment_order == 0)		
luma_chroma_sep_flag		u(1)
if(NumPrVectorModes > 1)		
pr_vector_mode_idx	2	te(v)
if(luma_chroma_sep_flag == 0) {		
<i>chromaStartCycle = 1</i>		
cycle_start_symbol	2	u(1)
if(cycle_start_symbol == 1) {		
cycle_start_symbol_ext	2	u(1)
<i>chromaStartCycle = 2 + cycle_start_symbol_ext</i>		
}		
}		
.....		

[0099] 现在将说明表 1 中所提出的新的或经修改语法元素的实例性语义。较不频繁发送的序列参数组中的语法元素说明如下：

[0100] 语法元素 `pr_info_present_flag` 规定在 FGS 片解码（例如，PR 片解码）中使用的语法元素的存在。当 `pr_info_present_flag` 等于 0 时，所述序列参数组中不存在用于 FGS 片解码的任何其它语法元素。当 `pr_info_present_flag` 等于 1 时，所述序列参数组中存在规定使用 CAF 及 PR 向量模式参数的语法元素。当 `pr_info_present_flag` 不存在时，可推断其等于 0。

[0101] 语法元素 `pr_cycle_aligned_fragment_flag` 规定针对 FGS 片解码是否应调用使用 CAF 的基于区块的解码。当 `pr_cycle_aligned_fragment_flag` 的值等于 1 时，应调用使用 CAF 的基于区块的解码。当 `pr_cycle_aligned_fragment_flag` 不存在时，那么应推断其等于 1。

[0102] 语法元素 `num_pr_vector_modes_minus1` 规定所述序列参数组中所存在的向量模式参数阵列的大小。向量模式参数从后续语法元素导出并存储在二维阵列 `ScanPosVectSet` 中，其中每一维阵列 `ScanPosVectSet[i]` 存储关于第 i 向量模式的参数，其中 $i = 0, 1, \dots, \text{num_pr_vector_modes_minus1}$ 。

[0103] 变量 `NumPrVectorModes` 可如下导出。如果语法元素 `num_pr_vector_modes_minus1` 存在，那么将 `NumPrVectorModes` 设定为等于 $(\text{num_pr_vector_modes_minus1} + 1)$ 。否则，`NumPrVectorModes` 等于 1。

[0104] 语法元素 `pr_coding_mode[i]` 规定使用哪一个语法元素来导出阵列 `ScanPosVectSet[i]`。当 `pr_coding_mode[i]` 等于 0 时，使用语法元素 `grouping_size_minus1[i]` 来导出阵列 `ScanPosVectSet[i]` 中的向量模式参数。当 `pr_coding_mode[i]` 等

于1时,使用阵列reverse_pr_vector_len_minus1[i][k] ($k = 0, \dots, \text{NumPosVector}[i]-1$) 来导出阵列 ScanPosVectSet[i] 中的向量模式参数。如果不存在 num_pr_vector_modes_minus1,那么将 pr_coding_mode[0] 设定为等于0。

[0105] 语法元素 grouping_size_minus1[i] 规定分组于每一向量内的扫描位置的数量减去1。如果 num_pr_vector_modes_minus1 不存在,那么将 grouping_size_minus1[0] 设定为等于15。

[0106] 语法元素 reverse_pr_vector_len_minus1[i][k] 规定 ScanPosVectSet[i][NumPosVector[i]-1-k] 与 ScanPosVectSet[i][NumPosVector[i]-k] 之间的向量长度减去1。应使用 $\text{ceil}(\log(\text{remVectorLen}-1))$ 位将语法元素 reverse_pr_vector_len_minus1[i][k] 解码为无符号值。

[0107] 阵列 ScanPosVectSet 可如下导出:

[0108]

```

for(i = 0; i < NumPrVectorModes; i++) {
    if(pr_coding_mode[i] == 0) {
        posVectLen = grouping_size_minus1[i] + 1
        NumPosVector[i] = 1 + 15 / posVectLen
        for(j = 0; j < NumPosVector[i]; j++)
            ScanPosVectSet[i][j] = j * posVectLen
        ScanPosVectSet[i][NumPosVector[i]] = 16
    }
    else {
        ScanPosVectSet[i][NumPosVector[i]] = 16
        for(j = NumPosVector[i] - 1; j >= 0; j++)
            ScanPosVectSet[i][j] = ScanPosVectSet[i][j + 1] -
                (reverse_pr_vector_len_minus1[i][NumPosVector[i] - 1 - j] + 1)
    }
}

```

[0109] 片标头中的对应语法元素可提供如下。

[0110] 语法元素 pr_vector_mode_idx 规定进入存储向量模式参数的阵列 ScanPosVectSet 的索引。pr_vector_mode_idx 的值应处于0与 (NumPrVectorModes-1) 的范围中,0及 (NumPrVectorModes-1) 也包括在内。ScanPosVectSet[pr_vector_mode_idx] 及 NumPosVector[pr_vector_mode_idx] 用来导出以下用于对当前渐进细化片进行解码的参数:

[0111]

NumPrCycles,

[0112]

```

ScanPosVectLuma[ cycleIdx ], cycleIdx = 0, ..., NumPrCycles,
ScanPosVectLuma8x8[ cycleIdx ], cycleIdx = 0, ..., NumPrCycles,
ScanPosVectChromaDC[ cycleIdx ], cycleIdx = 0, ..., NumPrCycles,
ScanPosVectChromaAC[ cycleIdx ], cycleIdx = 0, ..., NumPrCycles.

```

[0113] 这些参数可如下导出：

[0114]

```

ScanPosVectLuma [ 0 ] = 0
ScanPosVectLuma8x8 [ 0 ] = 0
ScanPosVectChromaAC[ 0 ] = 1
ScanPosVectChromaDC[ 0 ] = 0
for( cycleIdx=1; cycleIdx <= NumPosVector[ pr_vector_mode_idx ]; cycleIdx ++ )
{
ScanPosVectLuma[cycleIdx]=ScanPosVectSet[pr_vector_mode_idx][cycleIdx]
ScanPosVectLuma8x8 [ cycleIdx ]=ScanPosVectLuma[ cycleIdx ] * 4
ScanPosVectChromaDC[ cycleIdx ]=ScanPosVectChromaDC[ cycleIdx - 1 ]
ScanPosVectChromaAC[ cycleIdx ] = ScanPosVectChromaAC[ cycleIdx - 1 ]
// find the start scanning position for chroma DC and chroma AC
if( luma_chroma_sep_flag == 0 ) {
for( sIdx = ScanPosVectLuma[ cycleIdx - 1 ]; sIdx <
ScanPosVectLuma[ cycleIdx ]; sIdx ++ ) {
if( ( ScanPosVectChromaDC[ cycleIdx ] < 4 ) && ( ( sIdx == 0 ) ||
( ( sIdx >= ChromaStartCycle ) && ( ( scanIdx - ChromaStartCycle)
% 2 == 0 ) ) ) )
ScanPosVectChromaDC[ cycleIdx ] ++
if( ( ScanPosVectChromaAC[ cycleIdx ] < 16 ) && ( ( sIdx > 0 ) &&
( ( sIdx == ChromaStartCycle ) ||
( sIdx >= ChromaStartCycle && ( ( sIdx - ChromaStartCycle) % 3 =
= 1 ) ) ) ) )
ScanPosVectChromaAC[ cycleIdx ] ++
}
}
}
NumPrCycles = NumPosVector[ pr_vector_mode_idx ] +
( ( luma_chroma_sep_flag == 1 ) ? 1 : 0 );
ScanPosVectLuma [ NumPrCycles ] = 16;
ScanPosVectLuma8x8 [ NumPrCycles ] = 64;
ScanPosVectChromaDC [ NumPrCycles ] = 4

```

[0115] ScanPosVectChromaAC[NumPrCycles] = 16 ;

[0116] 语法元素 ScanPosVectLuma[i] 给出 4x4 亮度块在索引为 i 的编码循环中的开始扫描位置。ScanPosVectLuma8x8[i] 给出 8x8 亮度块在索引为 i 的编码循环中的开始扫描位置。ScanPosVectChromaDC[i] 给出色度 DC 块在索引为 i 的编码循环中的开始扫描位置。ScanPosVectChromaAC[i] 给出色度 AC 块在索引为 i 的编码循环中的开始扫描位置。

[0117] 如以下表 2 中所图解说明,还可在片标头中进行不同向量模式配置的信令。因此,在此经修改的向量模式中,可在片标头中而并非在所述序列参数组中规定向量长度。此方法可涉及使用超驰旗标来产生一组新的扫描向量。还可使用其它在复杂性与开销量之间具有不同折衷的方法来发信号通知所述向量长度。因此,本发明涵盖各种用于发信号通知向量模式的技术中的任一种,其中信息经编码以发信号通知向量模式,所述技术包括如表 2 中所图解说明的在片标头中进行信号传送的实例性技术。

[0118] 表 2

[0119] 不同向量模式配置的信令

[0120]

slice header in scalable extension() {	C	描述符
first mb in slice	2	ue(v)
slice type	2	ue(v)
.....		
if(slice_type == PR && fragment order == 0) {		
num mbs in slice minus1	2	ue(v)
luma chroma sep flag	2	u(1)
if(luma chroma sep flag == 0) {	2	u(1)
chromaStartCycle = 1		
cycle start symbol	2	u(1)
if(cycle start symbol == 1) {		
cycle start symbol ext	2	u(1)
chromaStartCycle = 2 + cycle start symbol ext		
}		
}		
cycle aligned fragment flag	2	u(1)
vector mode override	2	u(1)
if(vector mode override) {		
numPosVector = 0		
RemVectorLen = 16		
do {		

[0121]

<code>rev_fgs_vector[len - 1][numPosVector] = 0</code>		
<code>if(RemVectorLen > 1)</code>		
<code>reverse_fgs_vector[len - 1][numPosVector]</code>	2	u(v)
<code>RemVectorLen -=</code>		
<code>rev_fgs_vector[len - 1][numPosVector] + 1</code>		
<code>numPosVector ++</code>		
<code>} while (RemVectorLen > 0)</code>		
<code>}</code>		
<code>.....</code>		
<code>}</code>		

[0122] 现在将说明 FGS 片中的宏块标头的定义。在 JSVM 中,渐进片的位流结构以尽可能晚地发送不直接导致经重构视频的质量改善的语法元素的方式设计。如果部分地截断所述位流,那么所述经重构视频可具有最好的质量。以下表 3 提供伪码段,所述伪码段提供 fgs 片的基本位流结构。以下表 4 图解说明用于定义 fgs 片中的宏块标头的实例性语法元素。

[0123] 表 3

[0124] FGS 片位流结构

[0125]

<code>for(cycles = 0 ;cycle < 16 ;cycle++){</code>
<code>for(mb_idx = first_mb_in_slice ;mb_idx <= last_mb_in_slice ;mb_idx++){</code>
<code>if(need_to_send_motion_refinement)</code>
<code>send_mb_fgs_motion_refinement()</code>
<code>for(blk8x8 = 0 ;blk8x8 < 4 ;blk8x8++){</code>
<code>if(luma_sub_mb_cbp_is_not_sent){</code>
<code>send_luma_sub_mb_cbp()</code>
<code>if(is_first_nonzero_sub_mb_cbp){</code>
<code>if(delta_qp_is_not_sent&&base_mb_cbp == 0)</code>
<code>delta_qp()</code>
<code>if(! transform_8x8_specified_in_base_layer)</code>
<code>transform_8x8_flag()</code>

}
}
luma_coefficients_in_the_cycle()
if(allow_chroma_dc){
if(chroma_dc_cbp_is_not_sent){
chroma_dc_cbp()
if(delta_qp_is_not_sent&&chroma_dc_cbp != 0){
if(base_mb_dbp == 0)
delta_qp()
}
chroma_dc_coefficients_in_the_cycle()
}
if(allow_chroma_dc){
if(chroma_ac_cbp_is_not_sent){
chroma_ac_cbp()
if(delta_qp_is_not_sent&&chroma_ac_cbp != 0){
if(base_mb_cbp == 0)
delta_qp()
}
chroma_ac_coefficients_in_the_cycle()
}
}
}

[0126] 作为实例,可紧在色度系数之前但在同一宏块中第一编码循环的亮度系数之后发送色度 CBP,然而在 AVC 基础层中,可在宏块标头中发送色度 CBP。另一实例是 ΔOP 的传

输。如果增强层中的宏块不具有任何非零的亮度系数及非零的色度 DC 系数,但其具有一些非零的色度 AC 系数,且所述基础层中的宏块 CBP 是零,那么可紧在编码色度 AC 系数之前发送 ΔQP 。此方法与其中通常在 AVC 基础层中发送这些语法元素的方式比较不同。

[0127] 以上表 3 中所提出的伪码区段提供 FGS 片的基本位流结构。然而,通过对实际编码次序执行简单的分析,延迟传输这些语法元素实际上可能并不提供益处。

[0128] 在视频编码器 20 中可使用两个与色度相关的旗标 allow_chroma_dc 及 allow_chroma_ac,以相对于传输亮度系数的速率来控制传输色度系数的速率。在编码循环 0 中,这两个旗标可总是设定为 1,以使上文提及的所有语法元素将在第一循环中传输。在发送所述语法元素时,所述位将被延迟,但仍可在宏块的第一编码循环内传输。在截断位流的情况下,所述位的小的混洗不应影响编码性能产生大的影响。

[0129] FGS 片的 MB 标头可如以下表 4 中所指示来定义。显著地,MB 标头的此特定实例可具有与粗粒度 SNR 可缩放性 (CGS) 层的结构类似的相同结构。

[0130] 表 4

[0131] FGS 片中的宏块标头的定义

[0132]

for(cycles = 0 ;cycle < 16 ;cycle++){
for(mb_idx = first_mb_in_slice ;mb_idx <= last_mb_in_slice ;mb_idx++){
if(cycle = 0) {
//send MB header separately from the coefficients
if(need_to_send_motion_refinement)
send_mb_fgs_motion_refinement()
mb_luma_cbp()
chrom_cbp()
if(mb_luma_cbp != 0 && ! transform_8x8_specified_in_base_layer)
transform_8x8_flag()
if(mb_luma_cbp != 0 && base_mb_cbp != 0)
mb_delta_qp()
}
luma_coefficients_in_the_cycle()
if(allow_chroma_dc) {

chroma_dc_coefficients_in_the_cycle()
}
if(allow_chroma_dc){
chroma_ac_coefficients_in_the_cycle()
}
}
}

[0133] 如本发明中所说明,通过 CAF 及相关联的编码技术,FGS 片的解码过程可得到显著地简化,且可类似于离散增强层的解码过程。所述 CAF 及相关联技术还可显著地简化 FGS 片的语法规则。以下表 5 中论述的是当使用如本发明中所说明的 CAF 及相关联宏块标头时(例如)在视频解码器 22 内对串流进行解码的实例。除可使用本发明上文解释的改善或修改以外,一些未列举的函数可类似于 SVC JC 中所使用的函数。CAF 还可用不同的 FGS 编码方案来工作。

[0134] 可在 residual_block 解码函数内处置输入数据(即,片段内的数据)的切换。切换输入缓冲器的成本比与基于帧的解码过程相关联的成本小得多。

[0135] 表 5

[0136] 可缩放扩展语法中的 FGS 片数据

[0137]

	C	描述符
<code>progressive_refinement_slice_data_in_block_order() {</code>		
<code> progressive_refinement_data_supplemental_info_in_scalable_extension()</code>		
<code> while(CurrMbAddr <= lastMbAddr) {</code>		
<code> mb_header_in_progressive_refinement_slices()</code>		
<code> for(i8x8 = 0; i8x8 < 4; i8x8++) {</code>		
<code> if(!transform_size_8x8_flag !entropy_coding_mode_flag) {</code>		
<code> for(i4x4 = 0; i4x4 < 4; i4x4++) {</code>		
<code> for(i = 0; i < 16; i++) {</code>		
<code> baseCoeffs[i] = (! TransformSize8x8Flag) ?</code>		
<code> base_luma_level(CurrMbAddr, i8x8, i4x4, i) :</code>		
<code> base_luma_level8x8 (CurrMbAddr, i8x8, i * 4 + i4x4)</code>		
<code> }</code>		
<code> blk4x4 = i8x8 * 4 + i4x4</code>		
<code> coded_block_flag_luma[blk4x4]</code>	3 4	ge(v) ae(v)
<code> pr_slice_residual_block(LumaLevel[blk4x4], baseCoeffs,</code>	3 4	
<code> 0, 16, ScanVectPosLuma, coded_block_flag_luma[blk4x4], 0)</code>		
<code> if(!entropy_coding_mode_flag && transform_size_8x8_flag)</code>		
<code> for(i = 0; i < 16; i++)</code>		
<code> LumaLevel8x8[i8x8][4 * i + i4x4] =</code>		
<code> LumaLevel[blk4x4][i]</code>		
<code> }</code>		
<code> }</code>		
<code> }</code>		
<code> else {</code>		
<code> for(i = 0; i < 64; i++) {</code>		
<code> baseCoeffs[i] = base_luma_level8x8 (CurrMbAddr, i8x8, i)</code>		
<code> pr_residual_block(LumaLevel8x8[i8x8], baseCoeffs,</code>	3 4	
<code> 0, 64, ScanVectPosLuma8x8, 1, 0)</code>		
<code> }</code>		
<code>}</code>		

[0138] 表 5 (续)

[0139]

if(chroma_format_idc != 0) {		
NumC8x8 = 4 / (SubWidthC * SubHeightC)		
for(iCbCr = 0; iCbCr < 2; iCbCr++) {		
for(i = 0; i < 4 * NumC8x8; i++)		
baseCoeffs[i] = base_chromaDC_level(CurrMbAddr, iCbCr, i)		
pr_slice_residual_block(ChromaDCLevel[iCbCr], baseCoeffs, 0, 4 * NumC8x8, ScanVectPosChromaDC, 0, 1)	3 4	
}		
for(iCbCr = 0; iCbCr < 2; iCbCr++) {		
for(i8x8 = 0; i8x8 < NumC8x8; i8x8++) {		
for(i4x4 = 0; i4x4 < 4; i4x4++) {		
b4x4 = i8x8 * 4 + i4x4		
for(i = 0; i < 4 * NumC8x8; i++) {		
baseCoeffs[i] = base_chromaAC_level (CurrMbAddr, iCbCr, i8x8, i4x4, i)		
pr_slice_residual_block(ChromaACLevel[iCbCr][b8x8], baseCoeffs, 1, 16, ScanVectPosChromaAC, 0, 1)	3 4	
}		
}		
}		
}		

[0140] 以下表 6 图解说明可缩放扩展中的 FGS 片剩余区块数据的实例性语法。

[0141] 表 6

[0142] 可缩放扩展语法中的 FGS 片剩余区块数据

[0143]

pr_slice_residual_block(coeffLevel, baseCoeffs, startIdx, stopIdx, scanVectPos, codedBlockFlag, isChroma) {	C	描述符
if(!entropy_coding_mode_flag)		
pr_slice_sig_coeff_and_run = pr_slice_sig_coeff_and_run_cavlc		
else		
pr_slice_sig_coeff_and_run = pr_slice_sig_coeff_and_run_cabac		
endOfBlock = codedBlockFlag == 0		
codedBlockFlagNotCoded = (isChroma && codedBlockFlag == 1) ? 1 : 0		
runLen = 0		
firstCoeff = 1		
for(fragIdx = 0, scanIdx = startIdx; scanIdx < stopIdx; scanIdx ++) {		

[0144] 表 6 (续)

[0145]

while(scanIdx >= scanVectPos[fragIdx + 1]) {		
fragIdx ++		
Switch CAF()		
}		
if(baseSigCoeffMap[scanIdx]) {		
bLSign = (baseCoeffs[scanIdx] < 0 ? 1 : 0)		
coeff_refinement(bLSign, coeffLevel[scanIdx])		
}		
if(codedBlockFlagNotCoded) {		
codedBlockFlagNotCoded = 0		
if(startIdx == 1) { /* chroma AC */		
coded_block_flag_chromaAC	3 4	u(v) ae(v)
endOfBlock = (coded_block_flag_chromaAC == 0) ? 1 : 0		
}		
else {		
coded_block_flag_chromaDC	3 4	u(v) ae(v)
endOfBlock = (coded_block_flag_chromaDC == 0) ? 1 : 0		
}		
}		
if(runLen > 0)		
runLen --		
else if(! endOfBlock) {		
pr_slice_sig_coeff_and_run(coeffLevel, baseCoeffs, firstCoeff,		
scanIdx, startIdx, lastIdx, trmCoeff, numCoeffs, isChroma)		
runLen = (numCoeffs > 0) ? numCoeffs - 1 : 0;		
endOfBlock = (numCoeffs == 0) > 1 : 0		
firstCoeff = 0		
}		
if(runLen == 0 && ! endOfBlock)		
coeffLevel[scanIdx] = trmCoeff		
}		
}		

[0146] 以下表 7 图解说明有效系数的实例性语法及 FGS 片数据 CABAC 语法中的游程。

[0147] 表 7

[0148] 有效系数及 FGS 片数据 CABAC 语法中的游程

[0149]

<code>pr_slice_sig_coeff_and_run_cabac(coeffLevel, baseCoeffs, firstCoeff, scanIdx, startIdx, lastIdx, trmCoeff, numCoeffs, isChroma) {</code>	C	描述符
<code> significant_coefficient_cabac(firstCoeff, numCoeffs, trmCoeff)</code>		
<code>}</code>		

[0150] 以下表 8 图解说明有效系数的实例性语法及 FGS 片数据 CAVLC 语法中的游程。

[0151] 表 8

[0152] 有效系数及 FGS 片数据 CAVLC 语法中的游程

[0153]

<code>pr_slice_sig_coeff_and_run_cavlc(coeffLevel, baseCoeffs, firstCoeff, scanIdx, startIdx, lastIdx, trmCoef, numCoeffs, isChroma) {</code>	C	描述符
<code> numRemain = 1</code>		
<code> for(i = scanIdx + 1; i < lastIdx; i ++)</code>		
<code> numRemain += (baseCoeffs[i] != 0) 0 : 1</code>		
<code> sigVlcSelector = sig_vlc_selector[scanIdx]</code>		
<code> eobShift = isChroma ?</code> <code> eob_shift_chroma[scanIdx] : eob_shift_luma[scanIdx]</code>		
<code> significant_coefficient_cavlc (firstCoeff, numRemain, sigVLCSelector, eobShift, numCoeffs, trmCoef)</code>		
<code> if(numCoeffs == 0) {</code>		
<code> residual_mag = (trmCoef <= 0) ? 1 : 0</code>		
<code> else {</code>		
<code> eob = 1</code>		
<code> for(i = scanIdx + numCoeffs + 1; i < stopIdx; i ++)</code>		
<code> eob &= baseCoeffs[i] != 0</code>		
<code> if(eob) {</code>		
<code> residual_mag = 1</code>		
<code> coeff_sig_vlc_symbol</code>	3 4	ge(v)
<code> trmCoef = coeff_sig_vlc_symbol</code>		
<code> }</code>		
<code> if(residual_mag) { /* magnitudes greater than 1 in CAVLC */</code>		
<code> for(i = startIdx; i < stopIdx; i ++)</code>		
<code> sigCoefMap[i] =</code> <code> (coeffLevel[i] != 0 && baseCoeffs[i] == 0) ? 1 : 0</code>		

[0154]

<pre> significant_coefficient_magnitude_cavlc(coeffLevel, SigCoefMap, startIdx, stopIdx, trmCoef) } } if(residual_mag) { /* magnitudes greater than 1 in CAVLC */ for(i = 0; i < 4; i++) sigCoefMap[i] = sig_chromaDC_coef(CurrMbAddr, iCbCr, i) ? 1 : 0 significant_coefficient_magnitude_cavlc(ChromaDCLevel[iCbCr], SigCoefMap, 0, 4, trmCoef) } } } </pre>		
---	--	--

[0155] 本发明中所说明的任何装置可表示各种类型的装置，例如无线电话、蜂窝式电话、膝上型计算机、无线多媒体装置、无线通信个人计算机 (PC) 卡、个人数字助理 (PDA)、外部或内部调制解调器、游戏装置或任何经由无线或有线信道进行通信的多媒体装置。此种装置可具有各种名称，例如存取终端 (AT)、存取单元、订户单元、移动台、移动装置、移动单元、移动电话、移动远程站、远程终端、远程单元、用户装置、用户设备、手持式装置或类似装置。

[0156] 本文中所说明的技术可实施为硬件、软件、固件或其任何组合。如果实施为软件，那么所述技术可至少部分地通过计算机程序产品的计算机可读媒体上的一个或一个以上所存储或所传输指令或码来实现，所述指令或码致使处理器执行所述技术。计算机可读媒体可包括计算机存储媒体、通信媒体或两者，且可包括促进计算机程序从一个位置传送到另一位置的任何媒体。存储媒体可以是可由计算机存取的任何可用媒体。

[0157] 通过实例而并非限制的方式，所述计算机可读媒体可包含数据存储媒体，例如 RAM、同步动态随机存取存储器 (SDRAM)、只读存储器 (ROM)、非易失性随机存取存储器 (NVRAM)、ROM、电可擦除可编程只读存储器 (EEPROM)、EEPROM、快闪存储器、CD-ROM 或其它光盘存储装置、磁盘存储装置或其它磁性存储装置，或任何其它可用来携载或存储呈指令或数据结构形式且可由计算机存取的所需程序码的计算机可读数据存储媒体。

[0158] 此外，可适当地将任何连接称为计算机可读媒体。举例来说，如果使用共轴电缆、光纤电缆、双绞线、数字订户线路 (DSL) 或例如红外线、无线电及微波等无线技术从网站、服务器或其它远程源传输所述软件，那么所述共轴电缆、光纤电缆、双绞线、DSL 或例如红外线、无线电及微波等无线技术包括在媒体的定义内。本文所使用的磁盘 (disk) 及磁盘 (disc) 包括：光盘 (CD)、激光盘、光学盘、数字通用光盘 (DVD)、软盘及蓝光光盘，其中磁盘通常以磁性方式来复制数据，而光盘以光学方式（例如，通过激光）来复制数据。以上的组合也应包括在计算机可读媒体的范围内。

[0159] 与计算机程序产品的计算机可读媒体相关联的码可由计算机来执行，例如，由一个或一个以上处理器来执行，例如一个或一个以上数字信号处理器 (DSP)、通用微处理器、专用集成电路 (ASIC)、现场可编程逻辑阵列 (FPGA) 或其它等效集成或离散逻辑电路。在一些方面中，本文中所说明的功能性可提供于经配置以进行编码及解码的专用软件模块或硬件模块内，或并入到组合式视频编码器 - 解码器 (CODEC) 中。

[0160] 上文已说明本发明的各个方面。这些及其它方面归属于以上权利要求书的范围内。

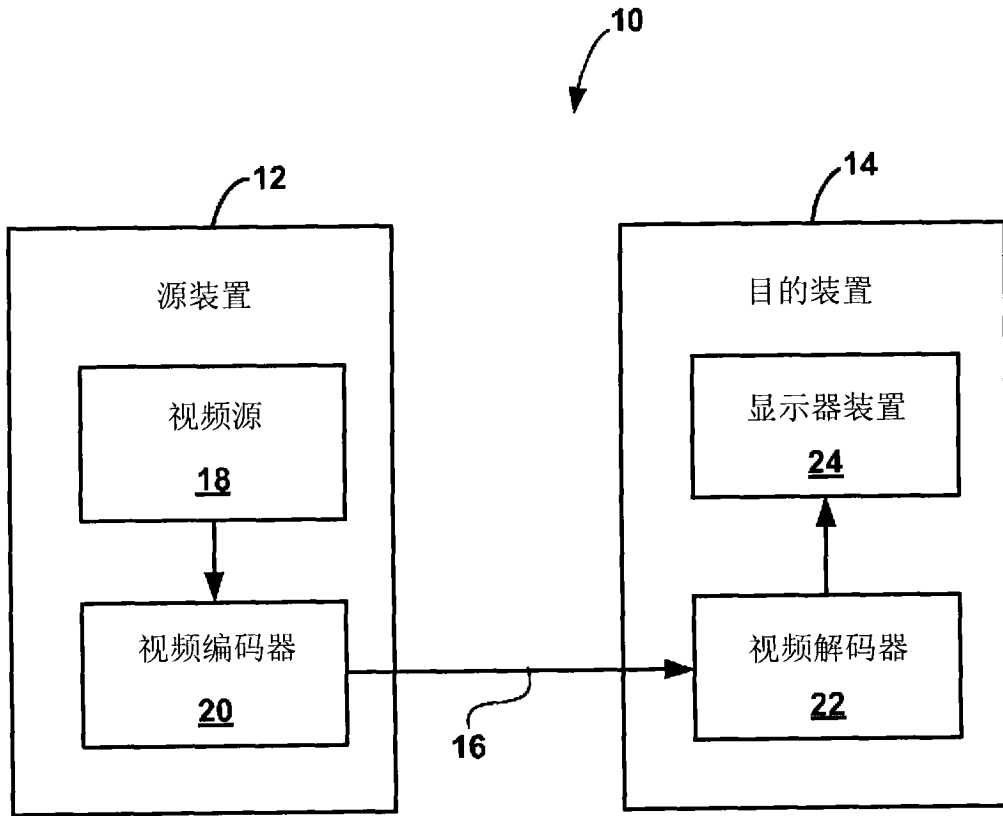


图 1

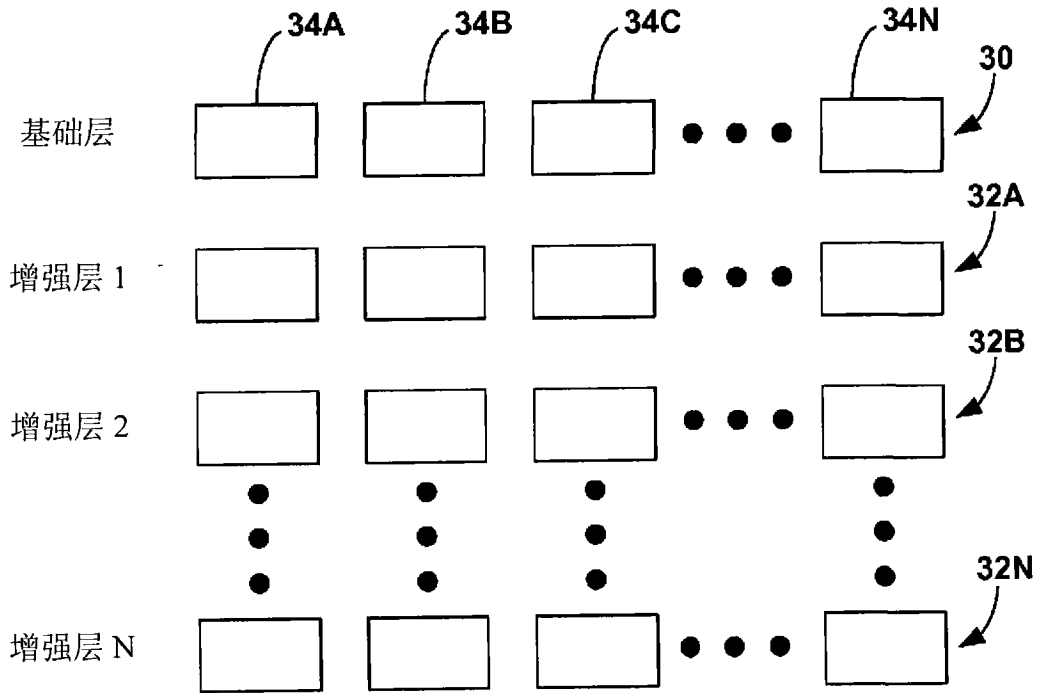


图 2

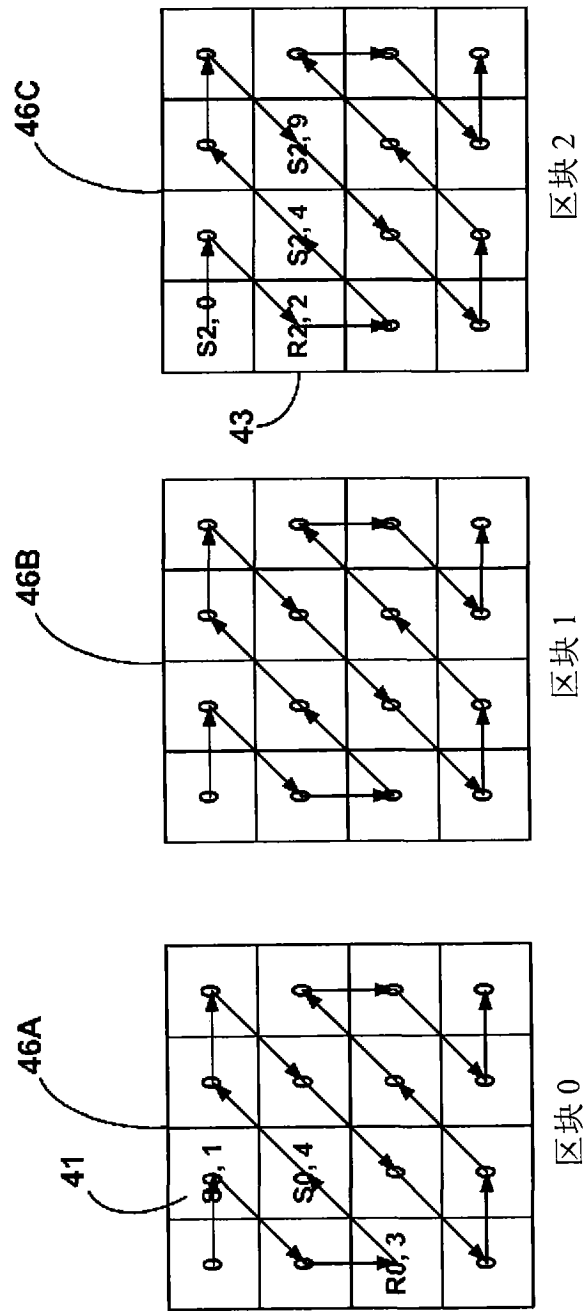


图 3

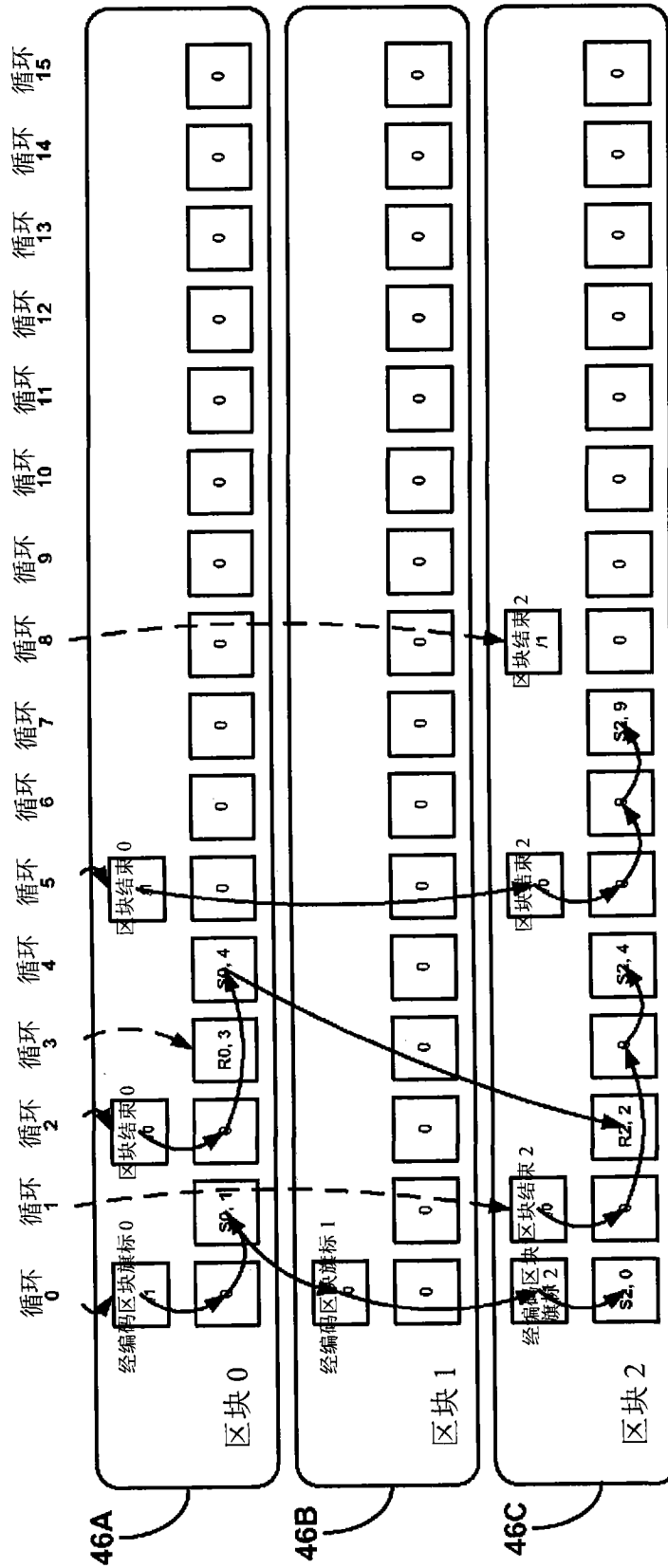


图 4

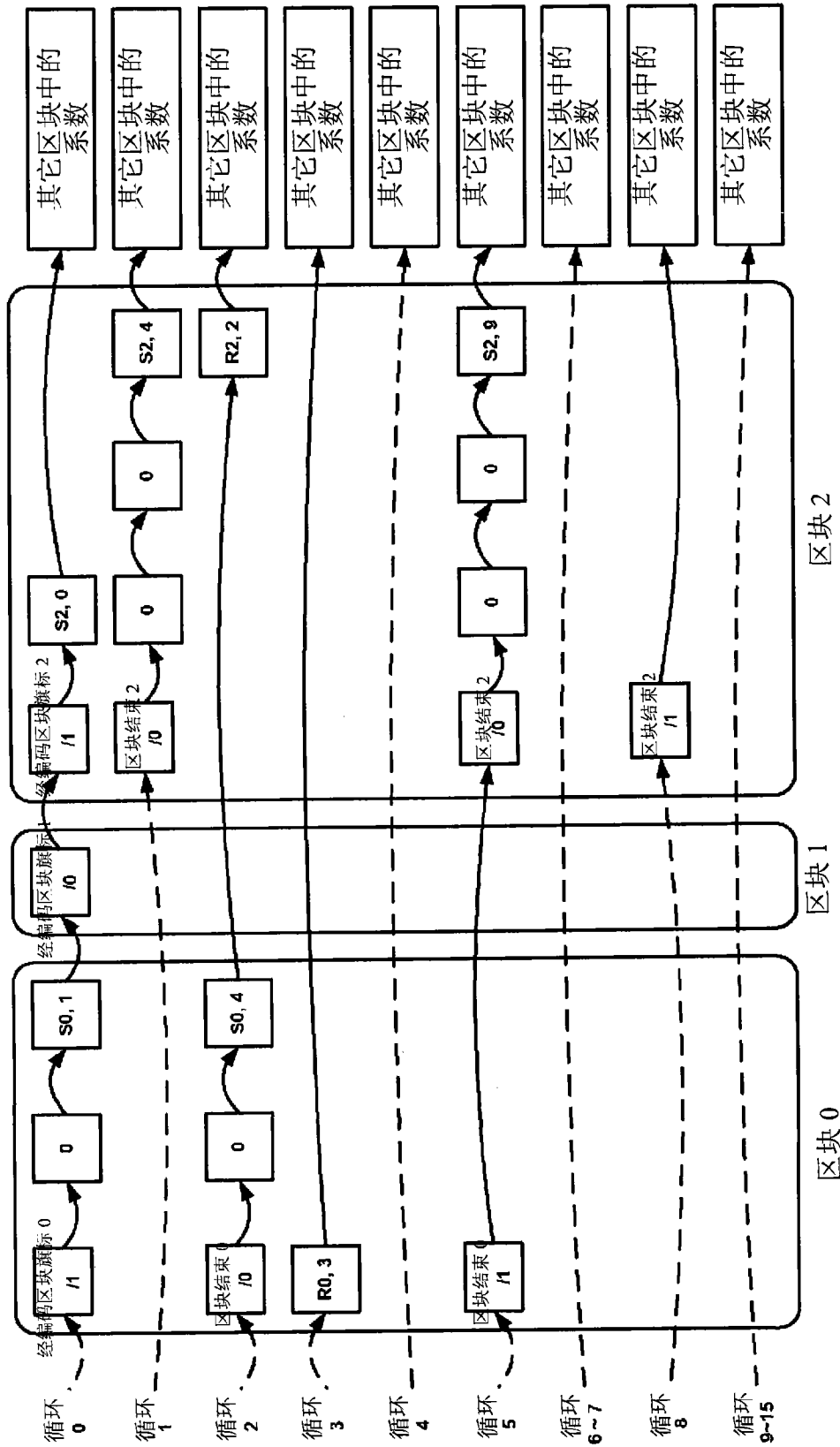


图 5

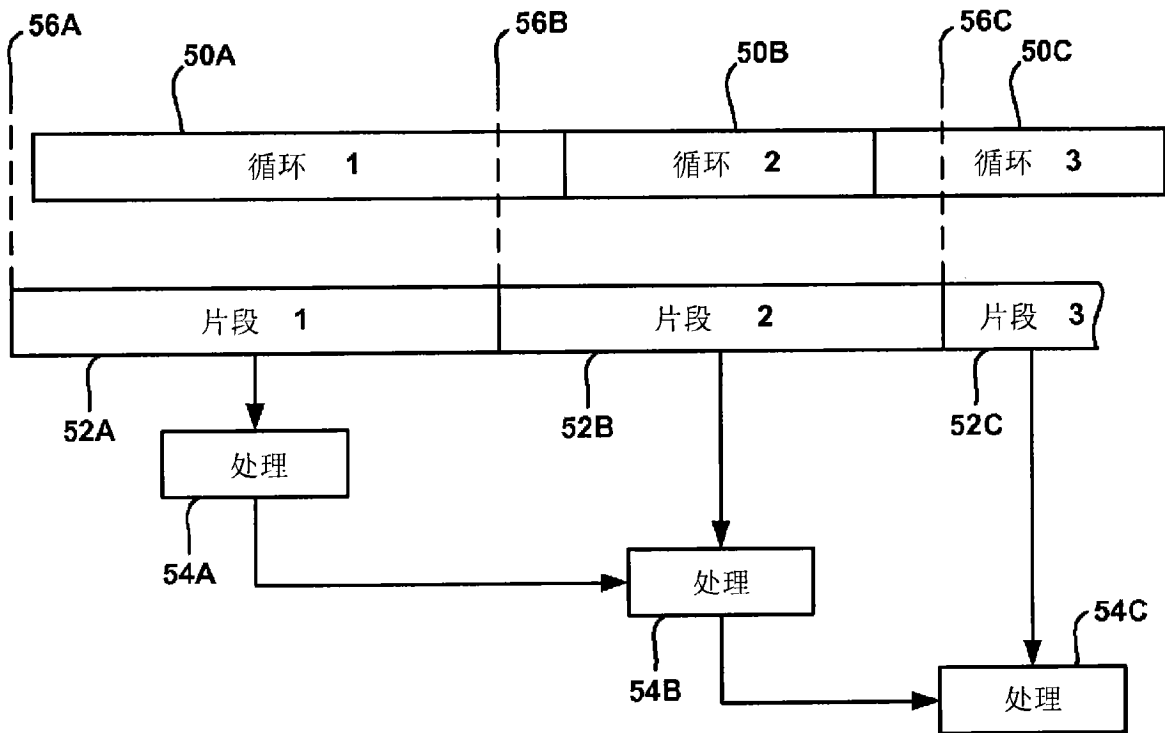


图 6

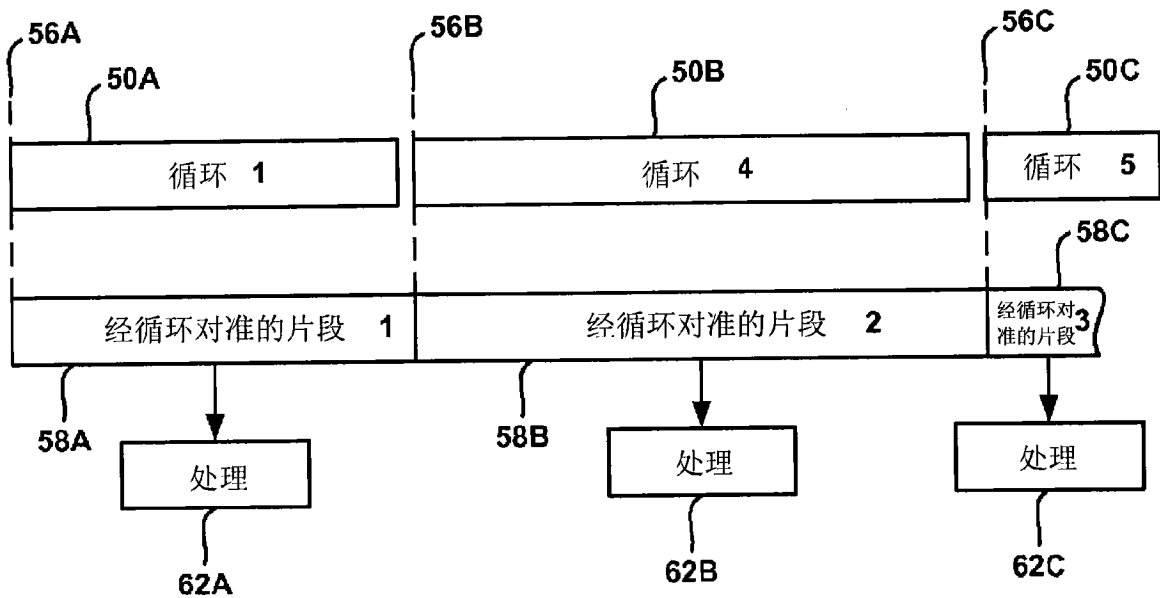
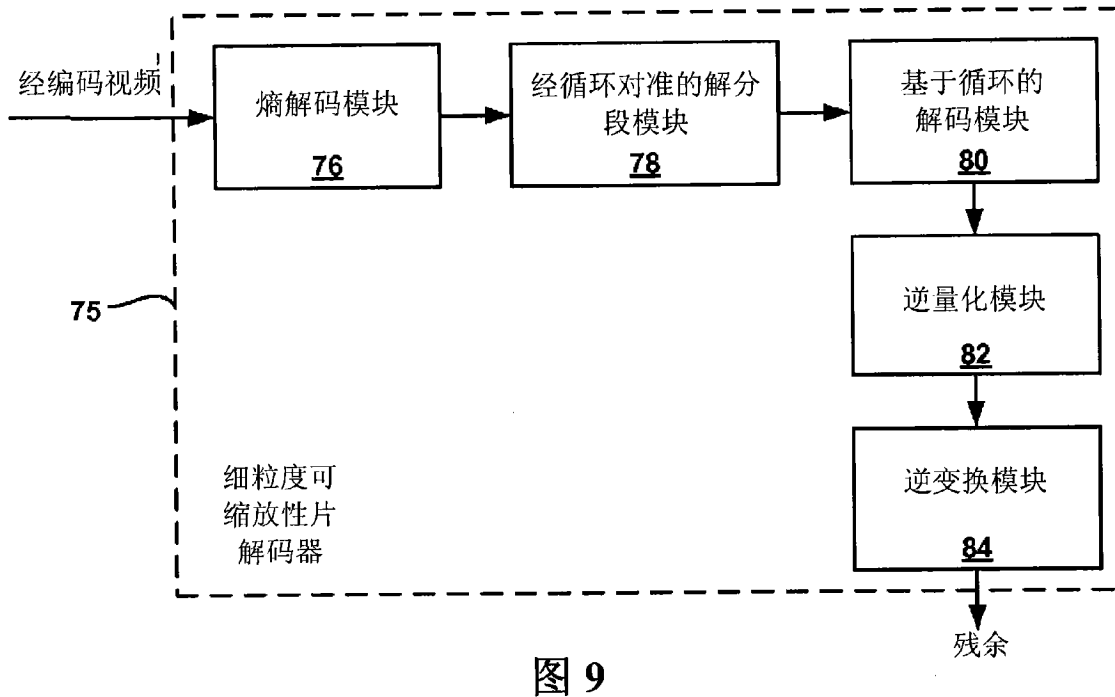
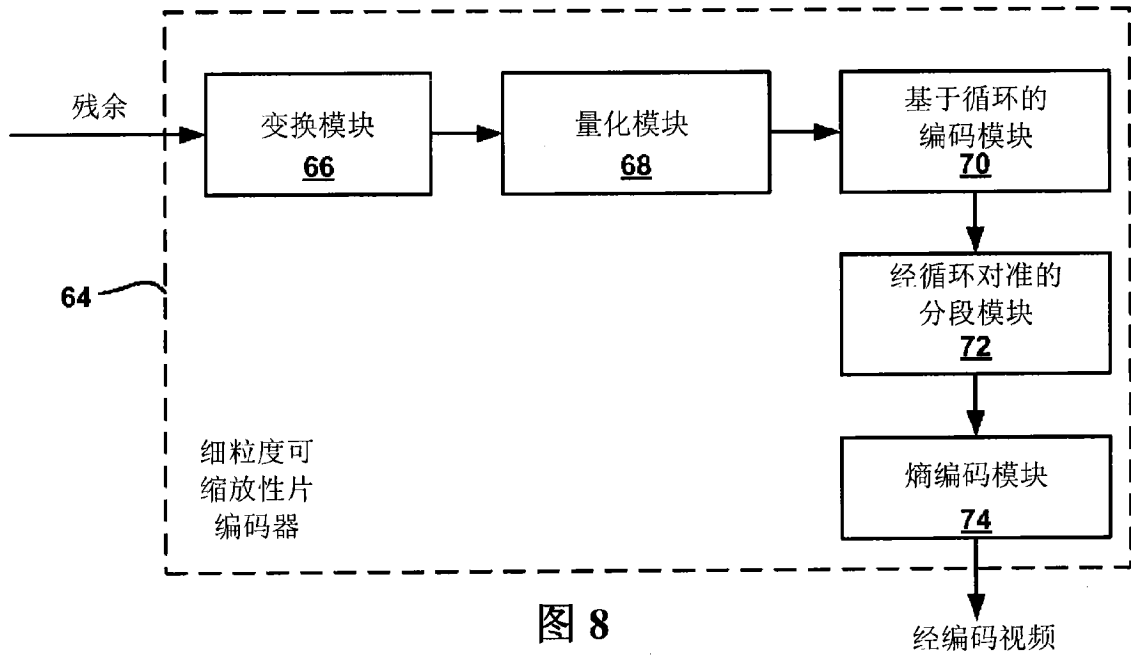


图 7



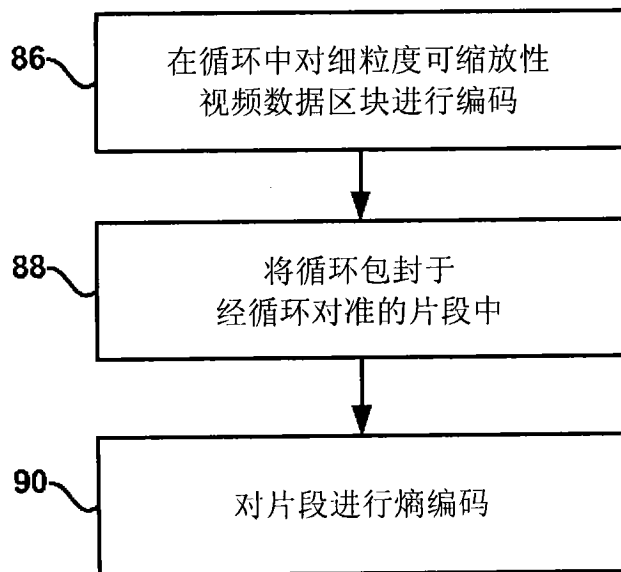


图 10

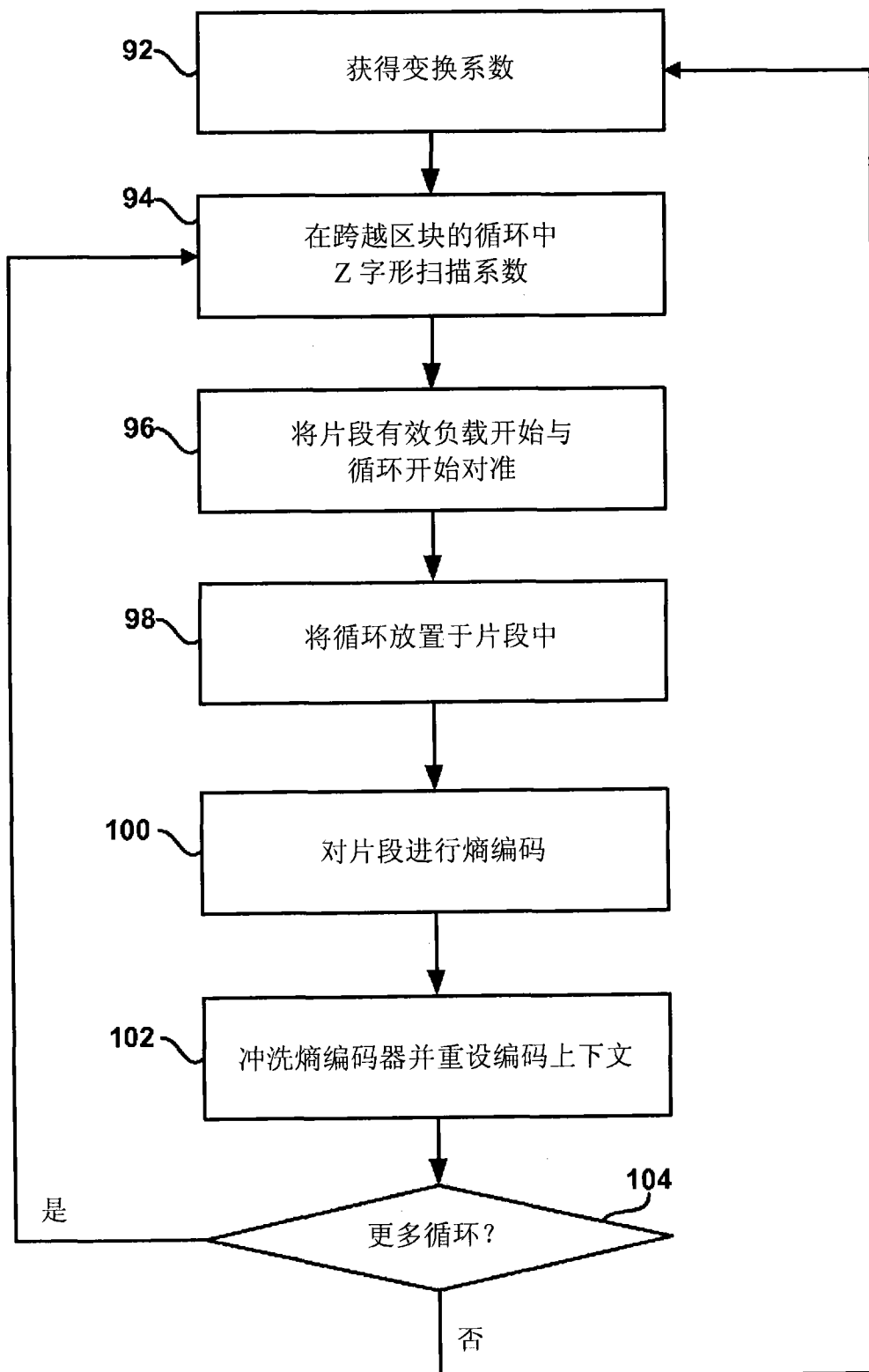


图 11

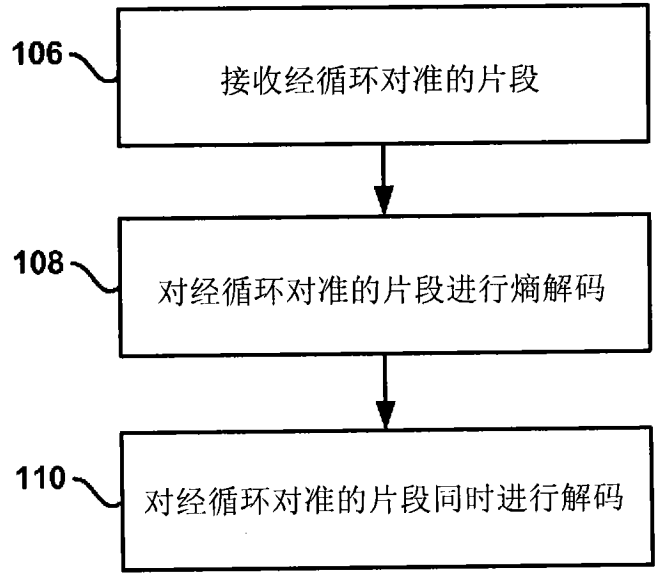


图 12

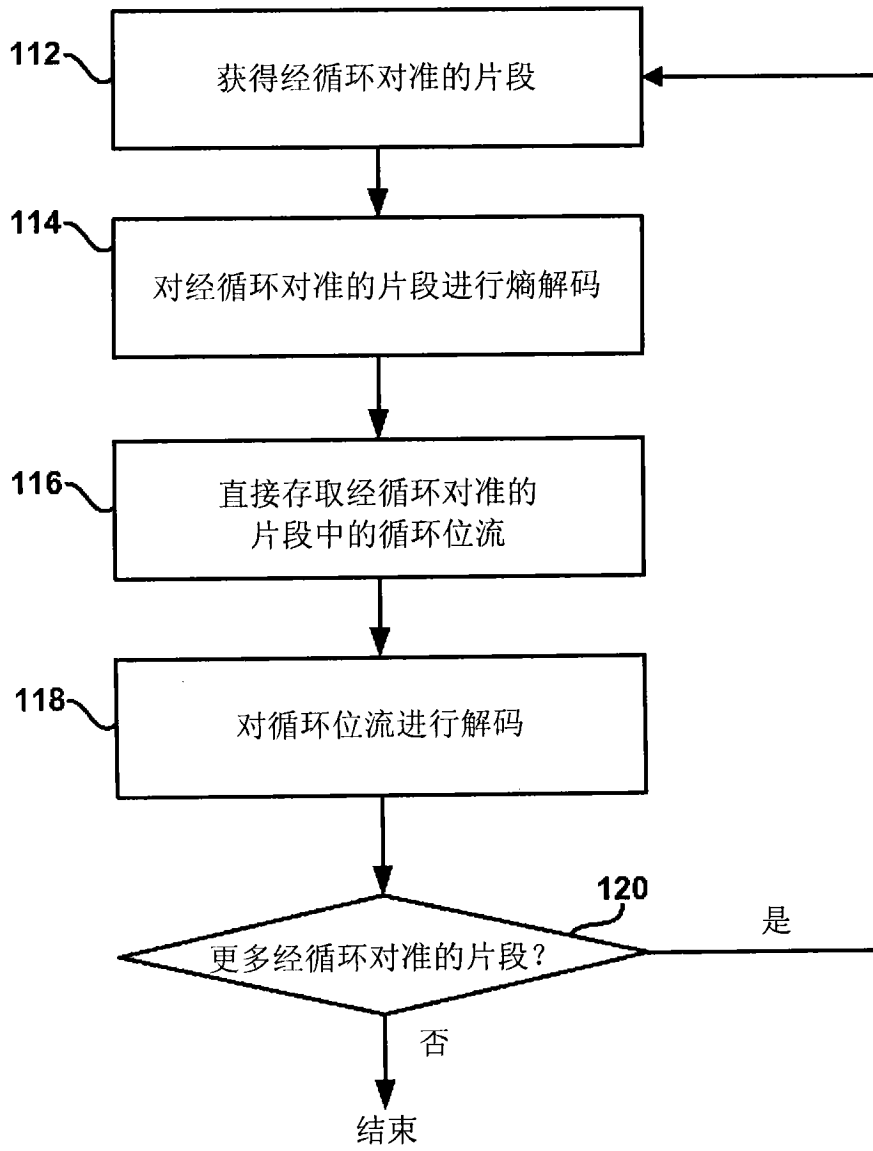


图 13

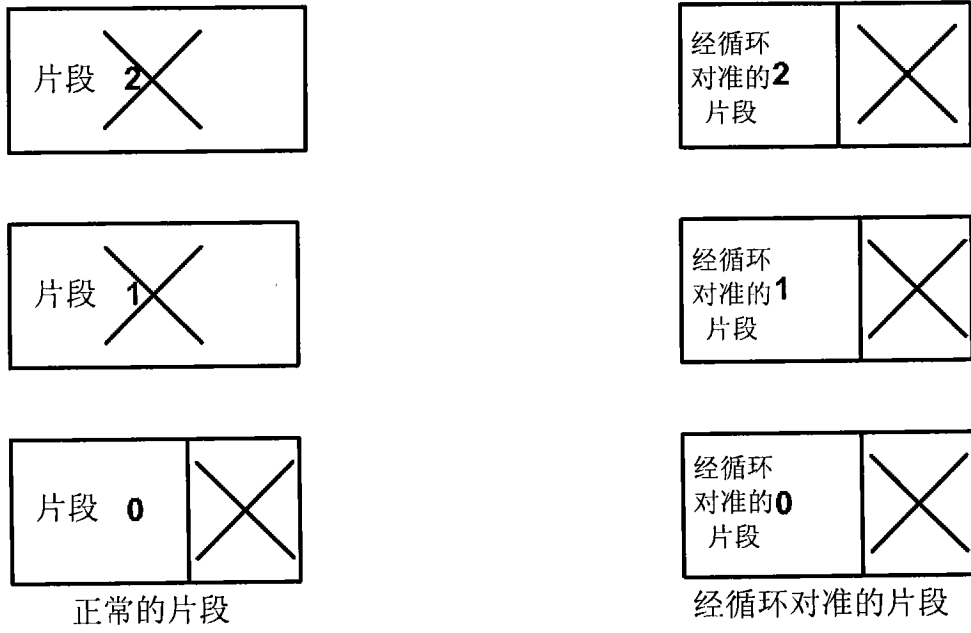


图 14