



(12) 发明专利申请

(10) 申请公布号 CN 103098485 A

(43) 申请公布日 2013. 05. 08

(21) 申请号 201180039619. 8

H04N 21/235 (2011. 01)

(22) 申请日 2011. 06. 13

H04N 21/647 (2011. 01)

(30) 优先权数据

H04N 21/84 (2011. 01)

61/354, 422 2010. 06. 14 US

H04N 21/845 (2011. 01)

H04N 21/854 (2011. 01)

(85) PCT申请进入国家阶段日

2013. 02. 16

(86) PCT申请的申请数据

PCT/US2011/040164 2011. 06. 13

(87) PCT申请的公布数据

W02011/159602 EN 2011. 12. 22

(71) 申请人 汤姆森特许公司

地址 法国伊西莱穆利诺

(72) 发明人 吴振宇 朱立华

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 王娟

(51) Int. Cl.

H04N 21/435 (2011. 01)

H04N 21/2343 (2011. 01)

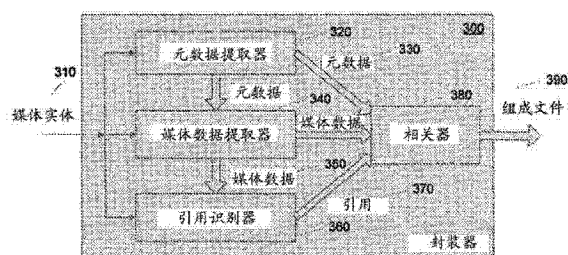
权利要求书2页 说明书7页 附图4页

(54) 发明名称

封装编码的多组成视频的方法和装置

(57) 摘要

描述了用于将包含不止一个层的媒体实体封装到多个组成文件的方法和设备以及用于组成文件读取的对应方法和设备,其中每个组成文件用于一个层。提出了对 SVC/MVC 文件格式的提取器数据结构的扩展。本发明的提取器扩展使得 NAL 单元能够跨越不同的组成文件进行引用。本发明使得能够进行媒体实体的自适应 HTTP 流传输。



1. 一种用于从包含不止一个层的媒体实体中创建组成文件的方法,所述方法包括以下步骤:

从所述媒体实体中提取每个层的元数据;

从所述媒体实体中提取与所述媒体实体的每个层所提取的元数据对应的媒体数据;以及

识别对与每个层所提取的媒体数据有关的附加媒体数据的引用;

将所述引用嵌入到每个层所提取的媒体数据中;以及

将每个层所提取的元数据和所提取的媒体数据相关联用于创建对应的组成文件。

2. 根据权利要求1所述的方法,其中,所述组成文件是电影盒、电影片段、分段和文件中的至少一个。

3. 根据权利要求2所述的方法,其中所述媒体数据和附加媒体数据包括数据样本。

4. 根据权利要求3所述的方法,其中数据样本包括网络抽象层单元。

5. 根据权利要求4所述的方法,其中与每个层所提取的媒体数据有关的所述附加媒体数据包括所提取的媒体数据中的网络抽象层单元所依赖的网络抽象层单元。

6. 根据权利要求5所述的方法,其中所述引用包含所述网络抽象层单元在所述附加媒体数据中的位置信息。

7. 根据权利要求6所述的方法,其中所述位置信息包括统一资源定位符和统一资源名称中的至少一个。

8. 根据权利要求7所述的方法,其中嵌入步骤还包括:

使用对所述附加媒体数据中的所述网络抽象层单元的引用来填充提取器;以及

将所述提取器嵌入到所提取的媒体数据的轨道中。

9. 一种用于从包含不止一个层的媒体实体中创建组成文件的文件封装器,该封装器包括:

提取器,其从媒体实体中提取每个层的元数据以及从所述媒体数据中提取与所述媒体实体的每个层所提取的元数据对应的媒体数据;

引用识别器,识别对与每个层所提取的媒体数据有关的、来自媒体实体的附加媒体数据的引用;

相关器,将该引用嵌入到每个层所提取的媒体数据中,并将所提取的媒体数据和所提取的元数据相关联使得能够为每个层创建包含所提取的元数据和所提取的媒体数据的组成文件。

10. 根据权利要求9所述的方法,其中,所述组成文件是电影盒、电影片段、分段和文件中的至少一个。

11. 根据权利要求9所述的文件封装器,其中所述媒体数据和附加媒体数据包括数据样本。

12. 根据权利要求11所述的文件封装器,其中数据样本包括网络抽象层单元。

13. 根据权利要求12所述的文件封装器,其中与每个层所提取的媒体数据有关的所述附加媒体数据包括所提取的媒体数据中的网络抽象层单元所依赖的网络抽象层单元。

14. 根据权利要求13所述的文件封装器,其中所述引用包含所述网络抽象层单元在所述附加媒体数据中的位置信息。

15. 根据权利要求 14 所述的文件封装器,其中所述位置信息包括统一资源定位符和统一资源名称中的至少一个。

16. 根据权利要求 15 所述的文件封装器,其中相关器还使用对所述附加媒体数据中的所述网络抽象层单元的引用来填充提取器;以及将所述提取器嵌入到所提取的媒体数据的轨道中。

17. 一种读取组成文件的方法,包括:

解析所述组成文件以获得其中的媒体数据和引用;以及

根据所述引用,如果所述组成文件的媒体数据与其它组成文件的媒体数据有关,则使用所述引用从所述其它组成文件中取得有关的媒体数据。

18. 根据权利要求 17 所述的方法,其中所述组成文件的媒体数据根据编码依赖性与其它组成文件的媒体数据有关。

19. 根据权利要求 17 的方法,其中所述组成文件是电影盒、电影片段、分段和文件中的至少一个。

20. 根据权利要求 17 所述的方法,其中所述媒体数据和所述有关的媒体数据包括数据样本。

21. 根据权利要求 20 所述的方法,其中数据样本包括网络抽象层单元。

22. 根据权利要求 21 所述的方法,其中所述引用包括提取器。

23. 一种文件读取器,包括:

解析器,解析组成文件以获得其中的媒体数据和引用;

取得器,根据所述引用从其它组成文件中取得与所述媒体数据有关的媒体数据;以及处理器,处理所述元数据、媒体数据以及从其它组成文件取得的媒体数据。

24. 根据权利要求 23 所述的文件读取器,其中所述组成文件的媒体数据根据编码依赖性与其它组成文件的媒体数据有关。

25. 根据权利要求 23 所述的文件读取器,其中所述组成文件是电影盒、电影片段、分段和文件中的至少一个。

26. 根据权利要求 23 所述的文件读取器,其中所述媒体数据和所述有关的媒体数据包括数据样本。

27. 根据权利要求 26 所述的文件读取器,其中数据样本包括网络抽象层单元。

28. 根据权利要求 27 所述的文件读取器,其中所述引用包括提取器。

29. 根据权利要求 23 所述的文件读取器,其中所述处理器包括视频解码器。

封装编码的多组成视频的方法和装置

[0001] 相关申请的交叉引用

[0002] 本专利申请要求 2010 年 6 月 14 日提交的、序列号 No. 61/354,422、标题“Extension to the Extractor data structure of SVC/MVC file formats”的美国临时专利申请的优先权权益。在此通过引用明确地并入上述临时专利申请的教导。

[0003] 本申请涉及与本申请同时提交的、以下待审的共同拥有的、序列号为 No. ____ 标题为“Method and Apparatus for Encapsulating Coded Multi-component Video”的美国专利申请(代理案号 NO. PU100141)。在此通过引用明确地并入上述非临时专利申请的教导。

技术领域

[0004] 本发明一般地涉及 HTTP 流传输。更具体地,本发明涉及对诸如可缩放视频编码(SVC)流和多视图编码(MVC)流之类的编码的多组成视频流的媒体实体进行封装,以用于 HTTP 流传输。

背景技术

[0005] 对于 HTTP 流传输应用,在服务器侧,经常将编码的视频进行封装并且存储为符合 BMFF 的文件,诸如 MP4 文件。此外,为了实现自适应 HTTP 流传输,文件通常被划分为多个电影片段并且这些片段被进一步分组为可由客户端 URL 请求寻址的分段。在实践上,将视频内容的不同编码的表示存储在这些分段中,使得客户端可以在会话期间动态地选取期望的表示来下载和播放。

[0006] 诸如 SVC 或 MVC 比特流之类的编码的分层视频,通过解码比特流的不同子集,在时间/空间分辨率、质量、视图等等方面启用不同的操作点,即表示,而提供了对这种比特率自适应的自然的支持。然而,现有的 ISO 基本媒体文件格式(BMFF)标准,诸如 MP4 文件格式,不能提供对每个层或者表示的单独访问,并且因此不能应用于 HTTP 流传输应用。如图 1 所示,在 MP4 文件格式中,一个媒体文件的所有层或者表示的元数据被存储在 moov 电影盒(movie box)中,而所有层或者表示的媒体内容数据被存储在 mdat 电影盒中。在 HTTP 流传输中,当客户端请求一个层时,由于所有层或者表示被混合在一起并且客户端不知道在哪里找到所需要的层或者表示,所以不得不发送整个文件。

[0007] 如稍后将看到的,在自适应 HTTP 流传输应用中,期望能够跨越电影片段或者组成文件的边界来引用媒体数据样本,诸如网络抽象层(NAL)单元。在 SVC/MVC 上下文中,可以通过使用如同“提取器”的机制来建立这种引用。提取器(extractor)是在对 BMFF 的 AVC 文件格式扩展的 SVC/MVC 修改中定义的内部文件数据结构:Information Technology-coding of audio-visual objects-Part15:Advanced Video Coding(AVC) file format,Amendment2:File format support for Scalable Video Coding,2008(第 15-17 页)。提取器被设计用来使得能够在不进行复制的情况下通过引用而从其它轨道(track)提取 NAL 单元。此处,轨道是 ISO 基本媒体文件中的有关样本的时间序列(timedsequence)。对于媒体数据而言,轨道对应于图像或者采样的音频的序列。以下示出

提取器的语法：

[0008]

```
class aligned(8) Extractor () {  
    NALUnitHeader( );  
    unsigned int(8)    track_ref_index;  
    signed int(8)  sample_offset;  
    unsigned int ((lengthSizeMinusOne + 1) * 8)  
        data_offset;  
    unsigned int ((lengthSizeMinusOne + 1) * 8)  
        data_length;  
}
```

[0009] 提取器数据结构的语义为：

[0010] NALUnitHeader: 该NAL单元结构如在ISO/IEC14496-10附录G中为类型20的NAL单元所指定的：

[0011] nal_unit_type 应该被设置为提取器NAL单元类型(类型31)。

[0012] forbidden_zeroz-bit, reserved_one_bit, 和 reserved_three_2bits 应该被设置为如在ISO/IEC14496-10附录G中所指定的。

[0013] 其它字段(nal_ref_idc, idr_flag, priority_id, no_inter_layer_pred_flag, dependency_id, quality_id, temporal_id, use_ref_base_pic_flag, discardable_flag, 和 output_flag) 应该被设置为如在Information Technology-Coding of audio-visual objects-part15:Advanced Video Coding(AVC) file format, Amendment2:File format support for Scalable Video Coding, ISO/IEC14496-15:2004/Amd. 2:2008(第17页)的B.4中所指定的。

[0014] track_ref_index 指定了用于找到从中提取数据的轨道的类型“scal”的轨道引用的索引。从中提取数据的轨道中的样本在时间上被对齐或者在媒体解码时间线上是最靠近前面的(closest preceding), 即, 仅使用时间-样本表格, 其利用具有包含提取器的样本(sample)的 sample_offset 指定的偏移来调整。第一轨道引用具有索引值1; 值0是保留的。

[0015] sample_offset 给出了样本在应该被用作信息源的链接的轨道中的相对索引。样本0(零)是具有与包含提取器的样本的解码时间相比较相同的或者最靠近前面的解码时间的样本; 值1(一)是下一个样本, 值-1(负1)是前一个样本, 等等。

[0016] data_offset: 引用样本内要复制的第一字节的偏移。如果提取以该样本中的第一字节数据开始, 则偏移值取值0。该偏移应该引用NAL单元长度字段的开始。

[0017] data_length: 要复制的字节数目。如果该字段取值0, 则复制完整的单个引用NAL单元(即, 从数据偏移引用的长度字段中取得要复制的长度, 在Aggregators的情况下增加additional_bytes 字段)。

[0018] 进一步的细节可以在 Information technology-Coding of audio-visual objects-part15:Advanced Video Coding(AVC)file format,Amendment2:File format support for Scalable Video Coding, ISO/IEC14496-15:2004/Amd. 2:2008 中找到。

[0019] 当前的提取器仅仅能够通过引用从其它轨道中,但是在同一电影盒 / 片段中,提取 NAL 单元。换言之,不可能使用提取器从不同的分段或者文件中提取 NAL 单元。这种约束限制了在上述使用情形下使用提取器。

[0020] 在现有技术中还没有充分地形成对上述问题的现有解决方案。将期望提供一种对层进行解析和封装而不牺牲速度和传输效率的能力。迄今在现有技术中还未实现这样的效果。

发明内容

[0021] 本发明针对用于从包含不止一个层的媒体实体中封装组成文件以及读取组成文件的方法和装置。

[0022] 根据本发明的一个方面,提供了一种用于从包含不止一个层的媒体实体中封装并创建组成文件的方法。该方法从媒体实体提取每个层的元数据以及与所提取的元数据对应的媒体数据,并且识别与每个层所提取的媒体数据有关的附加媒体数据的引用。将该引用嵌入到每个层所提取的媒体数据中。将所提取的媒体数据和元数据相关联使得能够为每个层创建包含所提取的元数据和所提取的媒体数据的组成文件。

[0023] 根据本发明的另一个方面,提供了一种文件封装器。该文件封装器包括:提取器,其从媒体实体提取每个层的元数据以及与所提取的元数据对应的媒体数据;引用识别器,识别对与每个层所提取的媒体数据有关的、来自媒体实体的附加媒体数据的引用;以及相关器,将该引用嵌入到每个层所提取的媒体数据中,并将所提取的媒体数据和所提取的元数据相关联使得能够为每个层创建组成文件。

附图说明

[0024] 参照附图,通过详细地描述本发明的示例性实施例,本发明的上述特征将变得更清楚,在附图中:

[0025] 图 1 示出了示例的 MP4 文件格式。

[0026] 图 2 示出了本发明的封装媒体实体的一个实施例。

[0027] 图 3 示出了用于从包含多个层 / 表示的媒体实体中封装或者创建组成文件的封装器的结构。

[0028] 图 4 示出了基于依赖关系将附加媒体数据与组成文件相关联的示例。

[0029] 图 5 示出了通过引用从与提取器所处的电影盒 / 片段不同的电影盒 / 片段中提取 NAL 单元的示例。

[0030] 图 6 示出了使用所发明的新的提取器数据结构所涉及的将 SVC/MVC 类型视频流封装到多个电影片段的封装操作。

[0031] 图 7 示出了用于读取组成文件的文件读取器的结构。

[0032] 图 8 示出了涉及本发明的视频解码器读取 SVC/MVC 类型视频流的过程。

具体实施方式

[0033] 在本发明中,诸如媒体文件或一组媒体文件或者流媒体之类的媒体实体被划分为或者封装到可以由客户端 URL 请求寻址的多个电影组成文件。在此,在更宽泛的含义中使用组成文件,表示片段、分段、文件以及与之等效的其它术语。

[0034] 在本发明的一个实施例中,解析包含多个表示或者组成的媒体实体以提取每个表示/组成的元数据和媒体数据。表示/组成的示例包括层,诸如在 SVC 中具有各种时间/空间分辨率和质量的层以及在 MVC 中的视图。在下文中,还使用层来指代表示/组成,并且这些术语可互换使用。例如,元数据描述对于每个表示在媒体实体中包含了什么以及如何使用其中包含的媒体数据。媒体数据包含了服务于媒体数据的目的(例如解码内容)所需的媒体数据样本,或者关于怎样获得所需的数据样本的任何必要信息。将为每个表示或者层提取的元数据和媒体数据进行相关联/相关并且将其存储在一起供用户访问。在物理上,可以在在硬盘驱动器或者其它存储介质上进行存储操作,或者可以通过关系管理机制虚拟地执行存储操作使得:当元数据和媒体数据实际上真正位于存储媒体的不同位置上时,当与其它应用或者模块连接时,它们表现为被存储在一起。图 2 图示了这个实施例的示例。在图 2 中,媒体实体包含三个层:基本层、增强层 1 和增强层 2。解析媒体实体以提取三个层的每一个的元数据和媒体数据,并且这些数据被单独地存储为组成文件,其中元数据和对应的媒体数据相关联在一起。

[0035] 图 3 示出了用于从包含多个层的媒体实体(诸如 SVC 编码视频)中封装并创建组成文件的优选的封装器 300 的结构。输入媒体实体 310 通过元数据提取器 320 和媒体数据提取器 340。元数据 320 提取每个层的元数据 330。媒体数据提取器 340 取得元数据 330 并且提取对应的媒体数据 350。注意,在不同的实施例中,元数据提取器 320 和媒体数据提取器 340 被实现为一个提取器。两种数据,元数据 330 和媒体数据 350 二者被馈入到相关器 380,相关器 380 将这两种类型的数据进行相关并且创建输出组成文件 390,每个层一个组成文件。

[0036] 分层的视频,诸如 SVC 或者 MVC 的 AVC 扩展所编码的视频,包含了多个媒体组成(可缩放的层或者视图)。通过解码比特流的不同子集,这种编码的比特流可以在时间/空间分辨率、质量、视图等等方面提供不同的操作点,即,表示或者层。此外,在比特流的层之间存在编码依赖性,即,一层的解码可能依赖于其它层。因此,为请求这种比特流表示之一可能需要从被封装的视频文件中取得并且解码一个或多个组成或者媒体数据。为了便于不同表示的提取过程,经常以使得每个层被单独地存储在不同的分段或者组成文件中的方式,将编码的分层视频封装为 MP4 文件。在这种情形下,由于上述的解码依赖性或者基于应用的其它依赖性,需要考虑多个分段或者组成文件所需要的或者与之有关的比特流的某些媒体数据样本,诸如 NAL 单元。

[0037] 在本发明的另一实施例中,提取分段或者组成文件所需要的附加媒体数据并且将其与该分段或者组成文件关联。图 4 示出了该实施例的示例。在图中,SVC 比特流具有三个空间层,HD1080p,SD 和 QVGA。形成对应于这三个操作点的三个电影片段或者组成文件,并且每一个可由不同的 URL 寻址。在每个电影片段或者组成文件内部,将解码所需要的所有媒体数据样本,该示例中的 NAL 单元,进行复制并且存储为“mdat”盒中包含的媒体样本。这样,当客户端通过使用合适的 URL 来请求特定的操作点或者表示时,服务器可以取得对应

的电影片段或者组成文件并且将其转发给客户端。在该示例中,图 3 中的媒体数据提取器 340 还为每个层从输入媒体实体 310 中提取为每个层所提取的媒体数据有关的附加媒体数据。相关器 380 还将每个层的所提取的附加媒体数据进行相关联来创建对应的组成文件。

[0038] 出于节省存储空间的目的,期望能够跨越电影片段或者组成文件边界来引用媒体数据样本,诸如 NAL 单元,而不用实际复制每个组成文件中的相同数据。然而,ISO 基本媒体文件格式(BMFF)及其扩展当前不支持该特征。为解决这个问题,在本发明的另一实施例中,对与电影片段或者组成文件的媒体数据有关的或者电影片段或者组成文件所需的那些附加媒体数据进行识别并建立引用。这种引用,而不是那些附加媒体数据,与元数据及其媒体数据一起与组成文件相关联。可以将这些引用嵌入到每个层所提取的媒体数据中,并且然后将每个层所提取的元数据和所提取的媒体数据相关联,用于创建对应的组成文件。

[0039] 在该实施例中,将引用识别器 360 加入到封装器 300 的结构中。引用识别器 360 从输入媒体实体 310 中识别对与每个层所提取的媒体数据 350 有关的那些附加媒体数据的引用 370。然后,例如通过将所述引用嵌入到所提取的媒体数据中,经由相关器 380 将引用 370 与每个层所提取的元数据 330 和所提取的媒体数据 350 相关联,用于创建对应的组成文件 390。

[0040] 如先前所讨论的,在 SVC/MVC 上下文中,可以通过使用类似“提取器”的机制来建立这种引用。当前的提取器仅仅能够通过引用从其它轨道中提取 NAL 单元,但却是在相同的电影盒/片段内。换句话说,不可能使用提取器从不同的分段或者文件中提取 NAL 单元。这种限制制约了提取器在其它情况下的使用。下文中,公开了一种对提取器数据结构的扩展,其中扩展的目的在于如上所述支持高效地将 SVC/MVC 类型分层的视频内容封装到多个组成文件中。

[0041] 将加入扩展以便向提取器数据结构提供用于引用驻留在与提取器所驻留的电影盒/片段或者组成文件不同的电影盒/片段或者组成文件中的 NAL 单元的额外能力。

[0042] 如下定义扩展的提取器:

[0043]

语法：

```

aligned (8) class DataEntryUrlBox (bit (24) flags)
    extends FullBox ('url', version = 0, flags) {
        string location;
    }

aligned (8) class DataEntryUrnBox (bit (24) flags)
    extends FullBox ('urn', version = 0, flags) {
        string name;
        string location;
    }

class aligned (8) Extractor () {
    NALUnitHeader ();
    DataEntryBox (entry_version, entry_flags) data_entry; // 增加的扩展
    unsigned int(8) track_ref_index;
    signed int(8) sample_offset;
    unsigned int ((lengthSizeMinusOne + 1) * 8)
        data_offset;
    unsigned int ((lengthSizeMinusOne + 1) * 8)
        data_length;
}

```

[0044] 语义：

[0045] data-entry 是统一资源定位符(URL)或者统一资源名称(URN)条目。Name 是 URN, 并且在 URN 条目中需要。Location 是 URL, 并且在 URL 条目中需要, 而在 URN 条目中是可选的, 其中它给出了寻找具有给定名称的资源的位置。每一个是使用 UTF-8 字符的空位结束的 (null-terminated) 字符串。如果设置了自包含的标志, 则使用 URL 形式并且不出现字符串; 框(box)以条目标志字段结束。URL 类型应该是递送文件的服务。相对 URL 是允许的并且与包含电影盒/片段的文件有关, 所述电影盒/片段包含提取器所属的轨道。

[0046] 其它字段具有与前述的原始提取器相同的语义。

[0047] 利用扩展的提取器, 现在可以通过引用从与提取器所处的电影盒/片段不同的电影盒/片段中提取 NAL 单元。图 5 示出了这种示例, 其与图 4 具有相同的 SVC 比特流, 但是使用新的扩展的提取器数据结构。如可以从图中看出的, 现在 SD 电影片段可以引用来自 QVGA 电影片段的 NAL 单元。同样, HD1080p 电影片段可以使用提取器引用来自 QVGA 和 SD 电影片段二者的 NAL 单元。与图 4 相比较, 没有跨越这些电影片段复制 NAL 单元, 因此节省了存储空间。

[0048] 图 6 示出了使用所发明的新的提取器数据结构将 SVC/MVC 类型的视频比特流封装

到多个电影片段或者组成文件时涉及的封装操作。该过程开始于步骤 601。在步骤 610 中逐一读取每个 NAL 单元。如果在步骤 620 中到达比特流的结尾,则该过程在 690 停止;否则,该过程前进到下一步骤 630。判定步骤 630 确定当前 NAL 单元是否依赖于来自其它轨道的 NAL 单元进行解码。如果确定当前 NAL 单元不依赖于来自其它轨道的 NAL 单元进行解码,则控制转移到步骤 640,其中使用当前 NAL 单元形成样本并且将其放置在当前轨道中。如果步骤 630 确定在当前 NAL 单元和来自其它轨道的 NAL 单元之间存在依赖性,则该过程继续转到步骤 650。判定步骤 650 进一步确定当前 NAL 单元所需要的 NAL 单元所源自的轨道是否驻留在相同的电影片段内。如果确定该轨道驻留在相同的电影片段中,则采用步骤 670 来填充扩展的提取器以便引用来自那些其它轨道的 NAL 单元。如果确定该轨道驻留在不同的电影片段中,在步骤 660 中识别这种电影片段的 URL 或者 URN,并且该过程前进到步骤 670,其中所识别的 URL 和 URN 被填充到扩展的提取器中。在填充了这种扩展的提取器之后,在步骤 680 将其嵌入到当前轨道中。然后,该过程在步骤 610 中利用下一个 NAL 单元重新开始。

[0049] 为了读取组成文件,采用图 7 所示的文件读取器 700。解析器 710 首先解析组成文件以得到元数据和媒体数据,以及引用(如果可得到)。根据解码的引用,如果诸如通过解码依赖性将媒体数据与其它组成文件的媒体数据相关,取得器(retriever) 720 从如引用中指示的其它组成文件中取得有关的媒体数据。处理器 730 还处理从组成文件中获得的元数据和媒体数据以及附加媒体数据(如果可得到)。解析器 710 的解析操作包括获得元数据、为处理器 730 准备好的媒体数据、为取得器 720 准备好的引用的各种必要的操作。其还将包括在必要时解析元数据和/或媒体数据。在一个实施例中,将引用嵌入到元数据中,并且因此通过解析元数据来获得引用。如果引用是可得到的,则解析步骤还包括分析引用的语法和解码引用。如果组成文件包含视频内容,处理器 730 可以包含视频解码器。在不同的实施例中,可以在处理器中并入解析器和取得器。

[0050] 图 8 示出了涉及本发明的、视频解码器读取 SVC/MVC 类型视频比特流的过程。步骤 801 访问组成视频文件,在步骤 805 中识别该组成视频文件的每一层的元数据和媒体数据。在步骤 810 解析所识别的元数据和媒体数据,并且在步骤 815 逐一地读取媒体数据的每个 NAL 单元。对于当前的 NAL 单元,在步骤 820 首先进行判定以便确定是否到达比特流的结尾,并且如果答案为“是”,该过程在步骤 825 结束。否则,该过程前进到判定步骤 830 以确定当前 NAL 单元是否是提取器。如果其不是提取器,这意味着其是包含解码数据的普通 NAL 单元,则在步骤 835 将 NAL 单元发送给解码器。如果当前 NAL 单元是提取器,则在步骤 840 确定当前 NAL 单元是否依赖于相同的组成文件外部的 NAL 单元。如果所需的 NAL 单元在相同的组成文件内,则在步骤 845 从当前文件中取得它,并且在步骤 835 将其发送给解码器。如果所需的 NAL 单元来自另一组成文件,则在步骤 850 使用提取器中的 Data_entry 信息对 NAL 单元进行定位,并且在步骤 855 中从远程文件中取得它,然后在步骤 835 中将其发送给解码器。

[0051] 尽管在此已经详细地描述了本发明的优选实施例,但应理解,本发明不限于这些实施例,并且本领域技术人员可以实现其它修改和变形而不脱离所附权利要求书定义的本发明的范围。

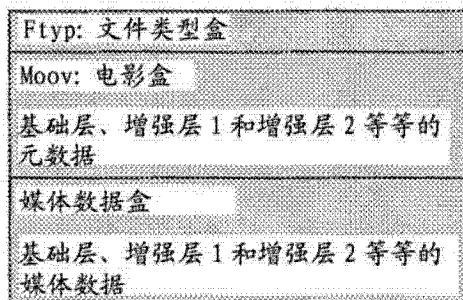


图 1

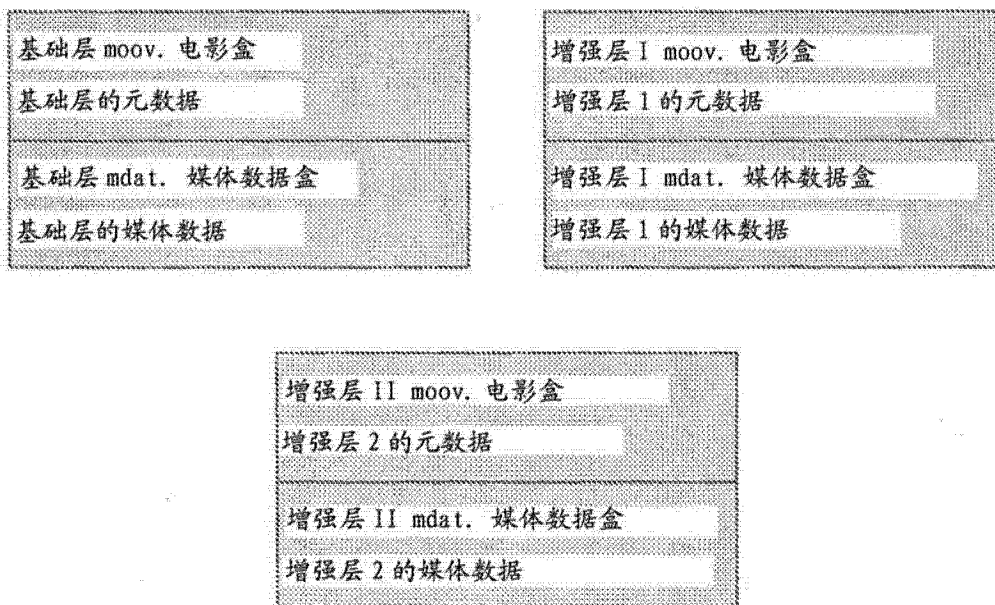


图 2

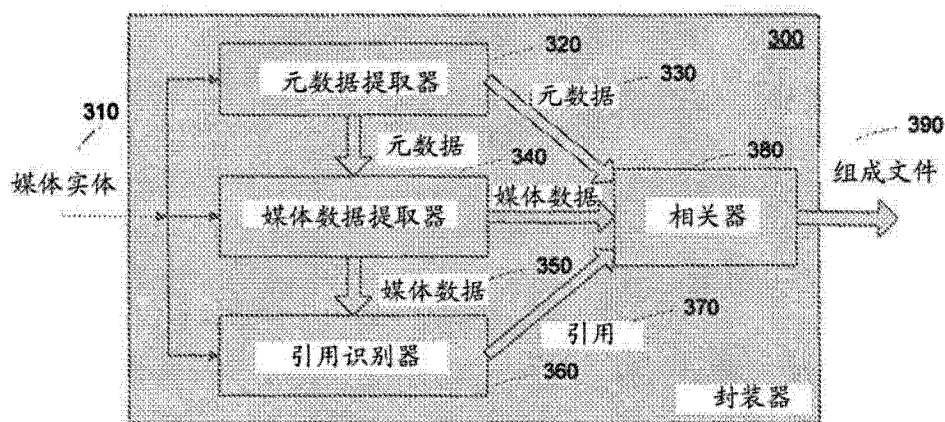


图 3

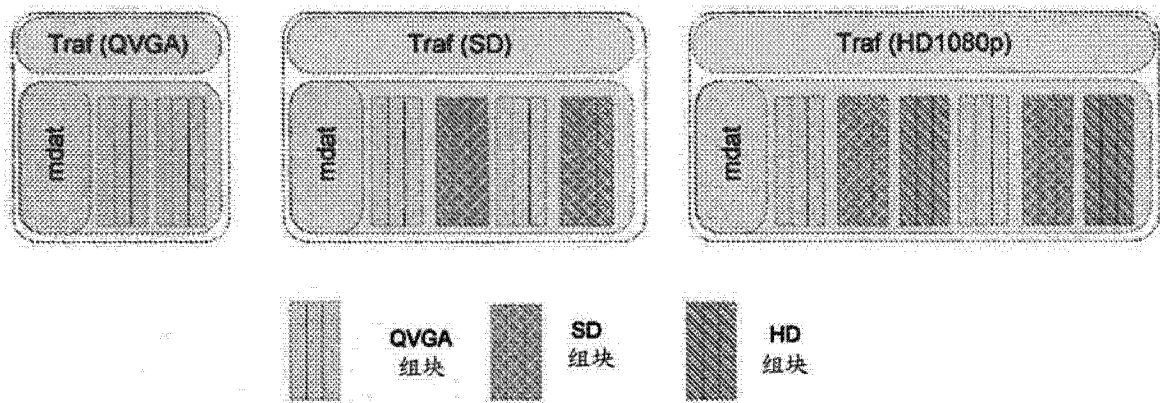


图 4

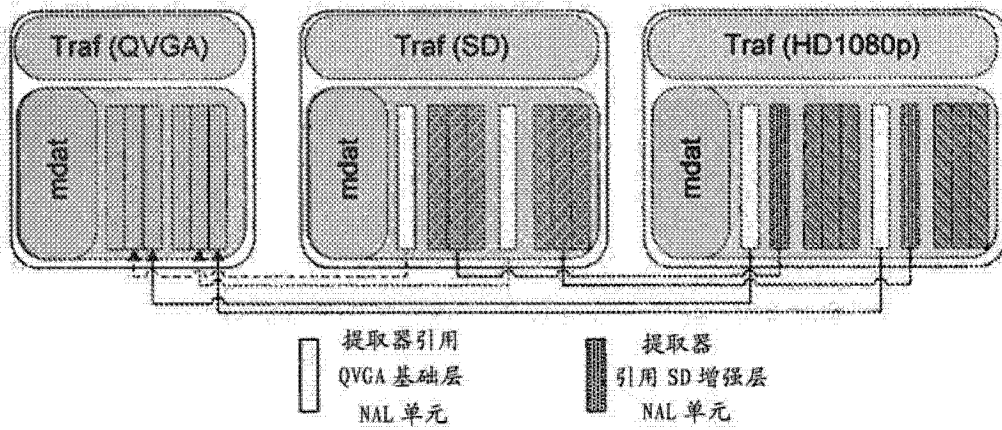


图 5

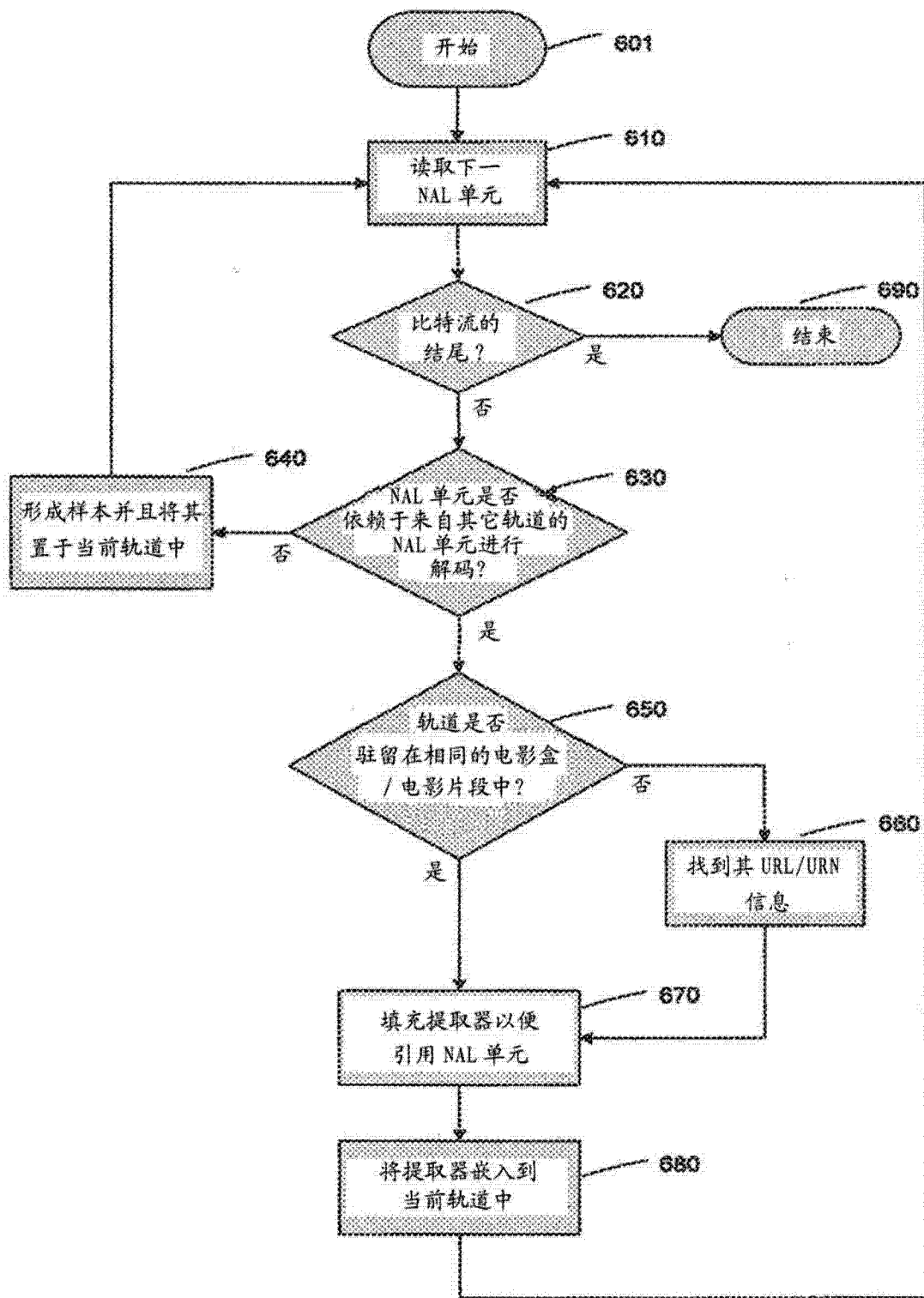


图6

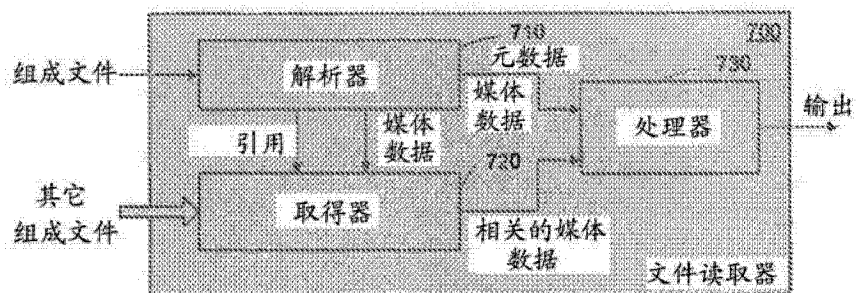


图 7

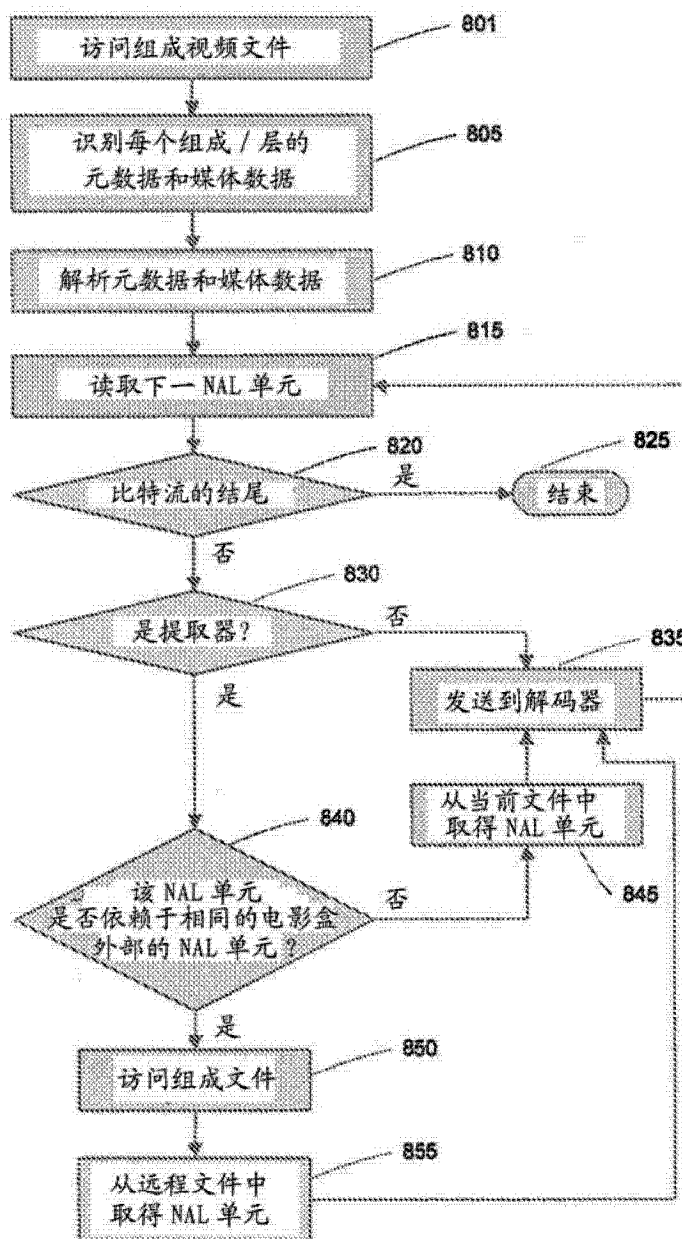


图 8