



(12) 发明专利

(10) 授权公告号 CN 1802702 B

(45) 授权公告日 2010.07.07

(21) 申请号 200480008324.4
 (22) 申请日 2004.03.22
 (30) 优先权数据
 03007611.1 2003.04.02 EP
 (85) PCT申请进入国家阶段日
 2005.09.27
 (86) PCT申请的申请数据
 PCT/EP2004/002999 2004.03.22
 (87) PCT申请的公布数据
 W02004/088660 EN 2004.10.14
 (73) 专利权人 汤姆森许可贸易公司
 地址 法国布洛里
 (72) 发明人 迪尔克·阿多夫 哈拉尔德·席勒
 约布斯特·霍拉特鲁
 拉尔夫·奥斯特曼
 哈特穆特·彼得斯
 (74) 专利代理机构 中科专利商标代理有限责任
 公司 11021

代理人 罗松梅
 (51) Int. Cl.
G11B 20/10 (2006.01)
G11B 27/10 (2006.01)
G11B 27/30 (2006.01)
 (56) 对比文件
 WO 02/23896 A1, 2002.03.21, 全文.
 CN 1197575 A, 1998.10.28, 全文.
 US 5543861 A, 1996.08.06, 全文.
 审查员 李笑

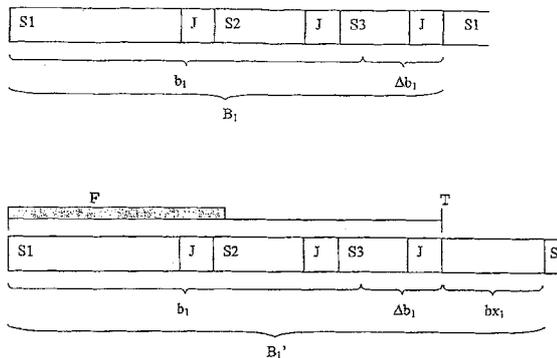
权利要求书 2 页 说明书 9 页 附图 2 页

(54) 发明名称

控制用于读取数据流的光学拾取器以便同时再现的方法

(57) 摘要

一种用于光学拾取器的最优化调度程序的方法,其减少了用于再现的拾取器跳跃频率和初始启动时间。拾取器从例如蓝光盘的光学存储介质上的不同文件中读取数据流,并将数据流缓冲在分离的缓冲区。当视频缓冲区 (B_1) 尽可能小,即,足以桥接用于其他流 (J, S2, S3) 的拾取器访问和读取次数,并扩展用于较低速率流的缓冲区 (B_1'),以便用于较低速率流的拾取器访问频率是用于通常是视频流的最高速率流的拾取器访问频率的整数倍。通过最初只将缓冲区填充到所需的最小级,能够使初始启动时间最小。



1. 一种控制用于从存储介质读取三个或更多数据流的光拾取器的方法,数据流被分布到所述存储介质上的多个文件中,被在读取之后被分离地缓冲,在缓冲之后,连续地从缓冲区读取数据流并将其用于同时再现,其中所述缓冲区具有各个不同的输出数据率和缓冲区尺寸,具有最高输出数据率的缓冲区是与视频数据流相关的第一缓冲区,所述方法包括:

对于缓冲区分别检测到缓冲量低于阈值;

根据所述检测,产生并调度请求拾取器从对应的数据流读取数据的请求,其中对于每一个所述缓冲区,在由其输出数据率、其尺寸和所述阈值产生的每一个时间间隔处产生各个平均请求数;

当服务于所述请求时,拾取器访问对应的数据流并从所访问的数据流中读取数据;

在对应的缓冲区中缓冲读取的数据;

其中,在第一缓冲区中缓冲的视频数据流的数据量足以用于在访问并读取任意一个其它数据流所需的时间中提供连续的缓冲区输出、利用从所述其它数据流读取的数据来填充相应的其它缓冲区、以及再次访问并读取所述视频数据流;以及

针对所述其它数据流中的每一个缓冲的数据量足以用于在访问视频数据流所需的时间中提供连续的缓冲区输出、填充并清空所述第一缓冲区多次、以及再次访问相应的其它数据流,其中在需要再次填充同一其它缓冲区之前要填充第一缓冲区的次数是一个整数,并且大于1。

2. 根据权利要求1所述的方法,其特征在于,对于所有所述其它数据流,所述整数是相同的。

3. 根据权利要求1或2所述的方法,其特征在于,在初始化期间,首先从存储介质读取所述其它数据流,然后读取所述视频数据流。

4. 根据权利要求1或2所述的方法,其特征在于,在初始化期间,完全地填充用于视频数据流的缓冲区,并且只部分地填充用于其它数据流的缓冲区。

5. 根据权利要求1或2所述的方法,其特征在于,在初始化期间,从存储介质上读取除所述视频数据流以外的数据流的顺序,与所述其它数据流处于存储介质上的顺序相同或相反。

6. 根据权利要求1或2所述的方法,其特征在于,所述存储介质是光盘,所述数据流至少包括视频流、音频流和字幕流,每个时间间隔中对于视频流的平均请求数目等于每一个时间间隔中对于音频流或字幕流的平均请求数目乘以所述整数。

7. 一种用于控制从存储介质读取三个或更多数据流光拾取器的设备,数据流被分布到所述存储介质上的多个文件中,被在读取之后被分离地缓冲,在缓冲之后,连续地从缓冲区读取数据流并将其用于同时再现,其中所述缓冲区具有各个不同的输出数据率和缓冲区尺寸,具有最高输出数据率的缓冲区是与视频数据流(S1)相关的第一缓冲区,所述设备包括:

拾取器,用于从所述存储介质读取数据流;

检测装置,用于对于缓冲区分别检测到缓冲填充低于阈值;

产生装置,用于根据所述检测,产生请求拾取器从对应数据流读取数据的请求,其中对于每一个所述缓冲区,产生每一个时间间隔的独立平均请求数,由缓冲区的输出数据率、其尺寸和所述阈值来产生每一个时间间隔;以及

调度装置,用于调度所产生的所述请求;

其中当服务于所述请求时,拾取器访问对应的数据流并从所访问的数据流中读取数据,

保存来自视频数据流的第一缓冲区的大小足以在访问并读取任意一个其它数据流所需的时间中提供连续的缓冲区输出、利用从所述其它数据流读取的数据来填充相应的其它缓冲区、以及再次访问并读取所述视频数据流;以及

保存来自所述其它数据流中每一个的数据的每一个其它缓冲区的大小足以用于在访问视频数据流所需的时间中提供连续的缓冲区输出、填充并清空所述第一缓冲区多次,以及再次访问相应的其它数据流,

其中在需要再次填充同一其它缓冲区之前要填充第一缓冲区的次数是一个整数,并且大于1。

8. 根据权利要求7所述的设备,其特征在于,对于所有所述其它数据流,所述整数是相同的。

9. 根据权利要求7或8所述的设备,其特征在于,在初始化期间,从存储介质中首先读取所述其它数据流,然后读取所述视频数据流。

10. 根据权利要求7或8所述的设备,其特征在于,所述存储介质是光盘,所述数据流至少包括视频流、音频流和字幕流,每一个时间间隔中对于视频流的平均请求数目等于每一个时间间隔中对于音频流或字幕流的平均请求数目乘以所述整数。

控制用于读取数据流的光学拾取器以便同时再现的方法

技术领域

[0001] 本发明涉及一种用于操作针对光学拾取器 (optical pick-up) 的调度程序 (scheduler) 的方法。拾取器从光存储介质中读取属于不同的数据类型数据流, 例如, 音频、视频、字幕或者其它数据, 并且数据流分布在存储介质上的多个文件中。

背景技术

[0002] 预先刻录和自刻录的盘可以支持“Out Of Multiplex” (OOM) 格式。Out of multiplex 是在盘的不同位置存储例如视频、音频和字幕的不同流分量的格式, 即, 不同的文件。这可以利用各种标准化介质来实现, 例如, 蓝光 (Blu-ray) 盘或者 DVD。此外, 可以实现公知为多角度的视频技术。多角度意味着对于特定场景, 视频电影 (video film) 可以包含在平行时间轴上的针对特定场景的各种可选视角, 其是用户可选择的, 并且可以被无缝地集成到视频中。为了回放 OOM 源, 在回放设备通过其针对同步表示的具体解码器对流进行解码之前, 拾取器从不同的位置读取所有需要的流。这意味着拾取器必须从一个流跳转到另一个流, 以便同时服务于所有解码器, 而没有可察觉的呈现中断。通常, 拾取器包含承载光学传感器的传动器 (actuator), 并且通过针对粗调节的机械驱动, 拾取器是可移动的, 而针对微调节, 传动器是可独立移动的, 不需机械驱动。

[0003] 用于提供具有光驱动的 OOM 技术的直接解决方案是缓冲区技术: 附加的流缓冲区用于桥接 (bridge) 跳转到另一个所请求的流并且将其读取所需的时间。典型示例包括三种流: 视频、音频和字幕。例如, 设计视频缓冲区的尺寸, 以便可以执行跳转到音频流、加载音频缓冲区、跳转到字幕流、加载字幕缓冲区以及跳回视频流, 而视频缓冲区不需空转。类似地设计例如针对音频和字幕的其它缓冲区的尺寸。

发明内容

[0004] 必须同时读取多个文件这一事实所引起的问题是: 由于机械驱动引起噪声和损耗的较高拾取器跳跃频率。此外的问题是: 在无缝视频角度切换期间出现的延迟。该延迟是从请求视频角度改变直到看到其它视频角度所需要的时间。其确定主要根据视频缓冲区的尺寸, 或者根据直到视频缓冲区变空并且新内容到达视频解码器的时间量。同样也应用于启动 OOM 解码。由于所有缓冲区必须从头开始填充, 因此, 从按下启动按钮直到有效地开始显示所经过的时间相当长。

[0005] 本发明所要解决的问题是提供一种用于控制拾取器的调度方法, 以便实现拾取器跳跃频率的减小。通过本发明所公开的方法和设备解决该问题。

[0006] 本发明解决的另一个问题在于减小呈现启动所需的延迟时间。优选实施例中公开的方法和设备解决了这一问题。

[0007] 优选实施例中公开了一种缓冲区填充方法, 提供了初始化期间的最短可能延迟。

[0008] 有利地, 包括一种用于减小角度切换所需延迟时间的方法。

[0009] 本发明的方法实现了较低的拾取器跳跃频率, 因此实现了噪声减小和改进的拾取

持续性。这通过非对称地设计缓冲区尺寸来实现。这里公开的另一个思想提供了一种用于 OOM 解码的技术,减小了无缝角度改变的切换时间。对于作为蓝光盘典型应用的 HDTV 流,缓冲区非常大。对于等待所请求的角度变化可见的用户来说,较大的缓冲区引起较长的延迟,其延迟可以通过本发明的方法来缩减。针对 OOM 的解码的启动出现相同的问题,并且可以通过所公开的方法而改善。

[0010] 本发明的一个目的是非对称地扩大较低速率流的缓冲区,同时保持最高速率流缓冲区尽可能的小。典型应用是高分辨率 (HD) 视频、音频和字幕的 OOM 解码。当读取用于视频、音频和字幕的三种流时,按照以下方式扩大较低速率流音频和字幕的缓冲区:在必须重新填充音频和字幕缓冲区之前,能够多次填充视频缓冲区。扩大用于音频和字幕的缓冲区尺寸引起拾取器跳跃之间的时间差别的扩大,并因此减小了拾取器跳跃频率。由于只需要传动器移动而无需拾取器跳跃,同时多次填充视频缓冲区不会增大拾取器跳跃频率。因此,降低了结果的拾取器跳跃频率。

[0011] 本发明的另一个目的是使用于无缝视频角度切换的切换延迟时间最小。这通过在视频缓冲区中引入角度切换标签来实现。角度切换标签用于确定视频缓冲区的以下部分:在角度切换情况下被废弃的部分,以及可以将其去除或重写而不给正在运行的缓冲区带来危险的部分。当已经从视频缓冲区去除废弃内容时,利用请求的新内容来填充缓冲区。重写废弃数据同时执行两个步骤。因为解码器不需要处理废弃内容,视频缓冲区的废弃内容的这种受控替换减少了视频角度切换时间。

[0012] 具体地,公开了一种用于控制从通常是光盘的存储介质中读取数据流的拾取器的方法,其中数据流属于不同的数据类型,例如音频、视频或字幕,因此具有不同的恒定或可变数据率。此外,数据流分布在所述存储介质上的多个文件中,由于其用于同步再现,当读取之后必须分离地进行缓冲。由于只有一个拾取器,与任意所述数据流相关的缓冲数据量最小,以便在访问和读取其它数据类型的其它数据流所需的时间期间,能够将缓冲的数据提供给例如解码器的后续处理。该方法包括:当与所述数据流相关的缓冲数据的剩余量低于阈值时,请求拾取器从数据流中读取更多数据;缓冲第一数据类型的第一数据流,所述数据流具有所有所述数据流中的最高数据率(例如视频);以及缓冲第二数据类型的至少第二数据流(例如音频),其中设置缓冲的数据量,以便与所述第二数据类型相关的读取请求之间的时间是与所述第一数据类型相关的读取请求之间的时间的倍数。

[0013] 有利地,本发明还可以用于针对所描述的应用,最优化地设计给定数量的缓冲空间。

[0014] 在从属权利要求、以下描述以及附图中公开了本发明的有利实施例。

附图说明

[0015] 参考附图描述了本发明的典型实施例,其中

[0016] 图 1a) 以传统方式设计尺寸的用于连接三种 OOM 流的跳跃和加载时间的比特流缓冲区;

[0017] 图 1b) 根据本发明的包含另外扩展缓冲区空间的用于连接三种 OOM 流的跳跃和加载时间的比特流缓冲区;

[0018] 图 2 针对结果的拾取器跳跃频率的定性的示例,比较了相等共享扩展缓冲区和非

对称共享扩展缓冲区；

[0019] 图 3 是视频角度切换期间的传统视频缓冲区模型；

[0020] 图 4a) 根据本发明的在视频角度之前的视频缓冲区模型；以及

[0021] 图 4b) 根据本发明的在视频角度之后的视频缓冲区模型。

具体实施方式

[0022] 以下将给出本发明的详细描述,其包括问题的详细分析。

[0023] OOM 解码主要受以下互相依赖的参数所影响：

[0024] 拾取器最大值信道比特率 R ,

[0025] 拾取器访问时间 T_{acc} ,

[0026] 拾取器跳跃频率 f_{jump} ,

[0027] 结果总缓冲区尺寸 B_{Σ} ,

[0028] 分离的流数目 N ,

[0029] 以及,各个流的比特率 r_i 。

[0030] 针对 OOM 解码的一般方法是:缓冲所有跳跃以及请求用于服务所有 N 个已解码流的加载时间。图 1a) 示出了用于三种流的示例。例如,通过桥接缓冲区 Δb_1 扩大针对视频的基本缓冲区 b_1 ,所述桥接缓冲区 Δb_1 的尺寸按照以下方式设计:在读取视频缓冲区的同时,可以执行以下操作且视频缓冲区不变空:跳转 J 到音频流、加载 $S2$ 音频缓冲区、跳转 J 到字幕流、加载 $S3$ 字幕缓冲区以及跳回 J 到视频流。 $S1$ 是需要自身加载整个视频缓冲区 b_1 、 Δb_1 的时间。通过 Δb_2 和 Δb_3 以相同方式扩大音频和字幕的基本缓冲区。

[0031] 缓冲区尺寸可以计算为：

[0032] $B_i = b_i + \Delta b_i$ (eq. 1.0) 每个流的缓冲区尺寸

[0033] $B = \sum b_i$ (eq. 1.1) 总基本缓冲区尺寸

[0034] $\Delta B = \sum \Delta b_i$ (eq. 1.2) 总扩展缓冲区尺寸

[0035] $B_{\Sigma} = B + \Delta B = \sum b_i + \sum \Delta b_i$ (eq. 1.3) 总缓冲区尺寸

[0036] 针对单个流缓冲区的缓冲区填充时间由下式确定：

[0037] $T_{fill,i} = \frac{B_i}{(R - r_i)}$ (eq. 2)

[0038] 当累积针对三种比特率的所有跳跃和填充时间时,可以建立线性方程系统 (LES),并且可以确定产生的桥接缓冲器 Δb_i 。针对 N 个流产生的 LES 可以写为以下矩阵形式：

$$[0039] \begin{pmatrix} \Delta b_1 \\ \Delta b_2 \\ \cdot \\ \cdot \\ \Delta b_N \end{pmatrix} = \begin{pmatrix} \frac{1}{r_1} & \frac{1}{r_2 - R} & \cdot & \cdot & \frac{1}{r_N - R} \\ \frac{1}{r_1 - R} & \frac{1}{r_2} & & & \frac{1}{r_N - R} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{r_1 - R} & \frac{1}{r_2 - R} & & & \frac{1}{r_N} \end{pmatrix}^{-1} \begin{pmatrix} N \cdot T_{acc} - \sum_{i=1}^N \frac{b_i}{r_i - R} \\ N \cdot T_{acc} - \sum_{i=2}^N \frac{b_i}{r_i - R} \\ \cdot \\ \cdot \\ N \cdot T_{acc} - \sum_{i=N}^N \frac{b_i}{r_i - R} \end{pmatrix} \quad (\text{eq. 3})$$

[0040] 为了提供 N 个 OOM 流,产生的拾取器跳跃频率可以估计为：

$$[0041] \quad f_{jump} \approx \frac{N}{\sum_{i=1}^N (T_{acc} + T_{fill,i})} = \frac{1}{T_{acc} + \frac{1}{N} \sum_{i=1}^N \frac{B_i}{(R-r_i)}} \quad (\text{eq. 4})$$

[0042] 由于 r_i 是可变比特率 (VBR), eq. 4 是针对平均值的估计。此外,有效的跳跃频率可以较高,因为不同的缓冲区不始终完全空运行,即,有效的 B_i 较小。

[0043] Eq. 3 和 eq. 4 描述了简单的循环 (round-robin) 调度程序:在完全充满缓冲区之后,调度程序切换到下一个缓冲区。继续进行该处理,直到已经用于所有的流缓冲区,并且调度程序通过所有流开始新循环。当具有较高的读取数据速率时,为了避免最差情况的缓冲区下溢,可以独立于单独的缓冲区充满度 (fullness) 进行循环调度。这意味着比实际所需更高的恒定拾取器跳跃频率。

[0044] 根据本发明,当进一步扩大桥接的流缓冲区并使用另一种循环调度程序模型时,可以减小拾取器跳跃频率。根据本发明,这种其它调度程序模型可以是如下所述的自由运行的调度程序:

[0045] 一每一个 OOM 流具有其自身的缓冲区,

[0046] 一当达到阈值时,例如“几乎空闲”,根据请求填充每一个 OOM 缓冲区,

[0047] 一使用排队机制处理并发请求,即,对每一个用于填充 OOM 缓冲区的请求排队一次。

[0048] 针对根据本发明的自由运行的调度程序,通过累积单个拾取器跳跃频率来确定产生的拾取器跳跃频率,并且通过以下计算:

$$[0049] \quad f_{jump} = \sum_{i=1}^N f_i \quad (\text{eq. 5})$$

[0050] 图 1b) 示出了进一步扩大的流缓冲区的方案。缓冲区 b_i 、 Δb_i 通过扩展缓冲区 b_{x_i} 进行扩展。在读取扩展缓冲区的同时,不使用拾取器。这产生单个流拾取器跳跃频率的降低,并且,因此影响根据 eq. 5 产生的拾取器跳跃频率。使用扩展缓冲区改变 eq. 1.0,其现在是

$$[0051] \quad B_i = b_i + \Delta b_i + b_{x_i} \quad (\text{eq. 6})$$

[0052] 但是,缓冲区扩展 b_{x_i} 具有两个缺点,第一是其要求更多的存储器,并且第二是由于 b_{x_i} 所需的额外填充时间,其影响其它流缓冲区需要的桥接缓冲区 Δb_2 、 Δb_3 。通过增加扩展缓冲区 b_{x_2} 、 b_{x_3} 来扩大其它桥接缓冲区 Δb_2 、 Δb_3 ,确保了针对任一流缓冲区填充请求的自由运行调度程序的及时响应。根据本发明,针对桥接缓冲区的尺寸的最好折衷方法如下。

[0053] 首先,确定单个流拾取器跳跃频率 f_i 。在自由运行调度程序已完全填充流缓冲区 $\Delta b_i + b_i + b_{x_i}$ 之后,继续读取缓冲区并且缓冲区空运行。当达到阈值 T 时,例如, $\Delta b_i + b_i$, 缓冲区可以向调度程序发送重新填充请求,并且调度程序对请求进行排队。此外,流缓冲区为空,直到调度程序对该请求进行服务。该示例假设通常剩余缓冲区的一半 $(\Delta b_i + b_i)/2$ 在调度程序作用之前空运行。当服务于请求时,剩余缓冲区的填充由图 1b) 中的灰色区域 F 所示。将以下计算应用于单个流拾取器跳跃频率:

$$[0054] \quad f_i = (T_{fill,i} + T_{empty,i} + T_{acc})^{-1} \quad (\text{eq. 7.1})$$

$$[0055] \quad f_i = \left(\frac{\frac{b_i + \Delta b_i}{2} + b_{x_i}}{R - r_i} + \frac{\frac{b_i + \Delta b_i}{2} + b_{x_i}}{r_i} + T_{acc} \right)^{-1} \quad (\text{eq. 7.2})$$

$$[0056] \quad f_i = \frac{2 \cdot r_i \cdot (R - r_i)}{b_i \cdot R + \Delta b_i R + 2 \cdot b_{x_i} \cdot R + 2 \cdot T_{acc} \cdot r_i \cdot R - 2 \cdot T_{acc} \cdot r_i^2} \quad (\text{eq. 7.3})$$

[0057] $T_{\text{empty}, i}$ 是读取针对流 i 的缓冲区的时间, 而不需被填充。扩大 eq. 5 和 eq. 7 所描述的扩展缓冲区 b_{x_i} 可以减小产生的拾取器跳跃频率。

[0058] 根据本发明, 非对称地增加扩展缓冲区 b_{x_i} 是尤其有利的, 即, 单独选择每一个扩展缓冲区 b_{x_i} , 以使最高的拾取器跳跃频率 $f_{i, r = \max}$ 是产生的单个拾取器跳跃频率 $f_{i, r \neq \max}$ 的倍数。这由以下等式表示:

$$[0059] \quad f_{i, r = \max} = \lambda \cdot f_1 = \lambda \cdot f_2 = \dots = \lambda \cdot f_{n, n \neq i} \dots = \lambda \cdot f_N \quad (\text{eq. 8.1})$$

$$[0060] \quad f_{\text{Video}} = \lambda \cdot f_{\text{Audio}} = \lambda \cdot f_{\text{Subtitle}} \quad (\text{eq. 8.2})$$

[0061] 在 eq. 8.1 中, f_1 是 b_{x_1} 等的函数, 并且 λ 是非对称因子。当按照以下方式选择 eq. 8.1 的关系: 具有最高比特率的流, 通常是视频流, 具有最高的跳跃频率 $f_{i, r = \max}$, 则这导致产生的拾取器跳跃频率降低, 同时使用相同总量的扩展缓冲区 B_{Σ} 作为以前的调度处理系统。这使得额外缓冲区的利用率最优, 因此针对额外缓冲区节省了最多字节。单独扩大针对较低比特率流的流缓冲区, 从而满足非对称因子 λ 。

[0062] eq. 8.2 是针对多媒体内容的 OOM 解码的典型蓝光盘应用的示例, 例如所述多媒体内容包括: HD 视频、音频和字幕。当读取针对视频、音频和字幕的三种流并根据本发明扩大较低速率流的缓冲区时, 则在音频和字幕缓冲区需要被重新填充之前, 可以填充视频缓冲区 λ 次。通过这种针对音频和字幕流的缓冲区尺寸的扩大, 拾取器跳跃之间的时间可以较长, 并且因此单个流拾取器跳跃频率较低。同时, 填充视频缓冲区 λ 次不会增加拾取器跳跃频率, 因为其仅要求拾取器的传动器运动, 没有跳跃。因此, 结果是降低了拾取器跳跃频率。

[0063] 可以采用以下来计算改善的结果跳跃频率:

$$[0064] \quad f_{\text{jump}}^* \leq f_{\text{jump}} = \sum_{i=1}^N f_i \quad (\text{eq. 9.1})$$

$$[0065] \quad f_{\text{jump}}^* = \min\left(\frac{N-1}{\lambda}; 1\right) \cdot f_{r=\max} + \left(\frac{N-1}{\lambda}\right) \cdot f_{r \neq \max} \quad (\text{eq. 9.2})$$

[0066] 在典型的应用示例中, $f_{r = \max}$ 与视频流缓冲区相对应。通过针对三种流结合 eq. 3 和 eq. 8.2, 可以建立新的 LES, 所述 LES 还包含针对音频和字幕的扩展缓冲区 b_{x_2} 、 b_{x_3} 的值。LES 是

$$[0067] \quad (\Delta b_1 \quad \Delta b_2 \quad \Delta b_3 \quad b_{x_2} \quad b_{x_3}) = M^{-1}(\lambda) \times v(\lambda) \quad (\text{eq. 10.1})$$

$$[0068] \quad M(\lambda) = \begin{pmatrix} 1 & \frac{-r_1}{R-r_2} & \frac{-r_1}{R-r_3} & \frac{-r_1}{R-r_2} & \frac{-r_1}{R-r_3} \\ \frac{-r_2}{R-r_1} & 1 & \frac{-r_2}{R-r_3} & 0 & \frac{-r_2}{R-r_3} \\ \frac{-r_3}{R-r_1} & \frac{-r_3}{R-r_2} & 1 & \frac{-r_3}{R-r_2} & 0 \\ 0 & \frac{R}{r_2 \cdot (R-r_2)} & \frac{-R}{r_3 \cdot (R-r_3)} & \frac{2R}{r_2 \cdot (R-r_2)} & \frac{-2R}{r_3 \cdot (R-r_3)} \\ \frac{R}{r_1 \cdot (R-r_1)} & 0 & \frac{-R}{\lambda \cdot r_3 \cdot (R-r_3)} & 0 & \frac{-2R}{\lambda \cdot r_3 \cdot (R-r_3)} \end{pmatrix} \quad (\text{eq. 10.2})$$

$$[0069] \quad v(\lambda) = \left(\begin{array}{c} r_1 \cdot 3 \cdot T_{acc} + \frac{r_1 \cdot b_2}{R - r_2} + \frac{r_1 \cdot b_3}{R - r_3} \\ r_2 \cdot 3 \cdot T_{acc} + \frac{r_2 \cdot b_1}{R - r_1} + \frac{r_2 \cdot b_3}{R - r_3} \\ r_3 \cdot 3 \cdot T_{acc} + \frac{r_3 \cdot b_1}{R - r_1} + \frac{r_3 \cdot b_2}{R - r_2} \\ \frac{-R \cdot b_2}{r_2 \cdot (R - r_2)} + \frac{R \cdot b_3}{r_3 \cdot (R - r_3)} \\ 2 \cdot T_{acc} \cdot \left(\frac{1}{\lambda} - 1 \right) - \frac{R \cdot b_1}{r_1 \cdot (R - r_1)} + \frac{R \cdot b_3}{\lambda \cdot r_3 \cdot (R - r_3)} \end{array} \right) \quad (\text{eq. 10.3})$$

[0070] $M(\lambda)$ 和干扰 $v(\lambda)$ 是非对称 λ 的函数。但是不能随意选择 λ 。由于其修改了需要充满其它缓冲区的时间,由以下条件限制

$$[0071] \quad T_{empty,i} - 3 \cdot T_{acc} - \sum_{n=1, n \neq i}^N T_{fill,n} \geq 0 \quad (\text{eq. 11})$$

[0072] 这意味着可以通过缓冲区 $T_{empty,i}$ 进行桥连而无需重新填充的时间必须等于或高于需要访问、读取其它流并将其存储到各自缓冲区的时间。使用 LES eq. 10, 并且考虑到 eq. 11 的条件,可以确定桥接缓冲区 Δb_i 和扩展缓冲区 b_{x_i} 。

[0073] 针对例如来自蓝光盘的电影的典型应用,在图 2 中描述了通过非对称 λ 所实现的增益。其示出了使用相等共享扩展缓冲区的调度系统的结果拾取器跳跃频率 $f_{skip,i}$, 与根据本发明的使用非对称扩展缓冲区的系统产生的拾取器跳跃频率 $f_{skip,i}^*$ 相比较,二者都是变量 λ 的函数。对于所示出的 λ 的合理值,即, λ 大于等于 2,针对本发明的调度程序的拾取器跳跃频率低于针对传统调度程序的拾取器跳跃频率。图示基于参数的典型值,即, $T_{acc} = 0.8s$, $R = 54Mbps$, $r_{video} = 40Mbps$, $r_{audio} = 640kbps$, $r_{subtitle} = 2kbps$ 。

[0074] 本发明的另一个要点是降低了用于无缝视频角度切换的切换延迟时间。对于 OOM 解码,可以将无缝视频角度切换比作视频流文件的变化。这必须针对用户无缝地完成,即,在视频中没有任何画面瑕疵、空白或者暂停。仅在流中的具体字节位置可以进行无缝视频角度切换。通过关于流的导航信息表示这些位置,例如,针对 MPEG 情况的画面组 (GOP) 边界。

[0075] 图 3 示出了传统的视频流缓冲区,其被拾取器比特率 R 填充,并且同时随解码比特率 r_i 变空。平均地,填充率 R 必须大于或等于解码比特率 r_i ,从而防止缓冲区变空。当再现视频时,将关于第一视角 A1 的流加载到缓冲区并且将其再现。在再现一定时间之后,用户可以请求视频角度改变。此时,缓冲区具有剩余填充级 L1。当调度程序已接收该请求并且要重新填充视频缓冲区时,拾取器将不立即跳转到其它与第二视角 A2 相关的视频流。相反,拾取器继续读取第一视角 A1 的其它字节,直到在比特率中检测到无缝连接;此时,缓冲区具有新的填充级 L2,包含关于第一视角 A1 的视频流。实现无缝连接之后,拾取器可以通过跳转到关于第二视角 A2 的视频流,从而切换视频输入流。然后,将该流加载到在级别 L2 之上的视频缓冲区中。

[0076] 同时,解码器在读取位置 L0 处从视频比特缓冲区读取数据,并且因此减少了缓冲区的填充,即,级别 L1 和 L2 连续地下降。该场景中切换视频角度之前的有效延迟时间由以下因素确定:当视频角度改变请求到达调度程序时的缓冲区充满度 L1;与实现无缝连接之

前的旧视频角度 A1 相关的新加载序列的长度, 其将缓冲区填充到 L2 ; 以及 视频解码比特率 r_i 。针对用户的有效延迟时间由缓冲区充满度 L2-L0 的量和视频比特率来确定, 如下所述

$$[0077] \quad T_{delay} = \frac{B_{L2-L0}}{r_i} \quad (\text{eq.12})$$

[0078] 为了实现较小的延迟, 保持较小的视频比特缓冲区是有用的。但是, 如上所述, 这提高了拾取器跳跃频率。此外, 延迟时间可以显著地变化, 因为没有确定请求时的缓冲区充满度 L1。可以通过本发明实现视频角度切换的改善, 并且有利于每一个信道切换。

[0079] 最小化缓冲区尺寸 B_{L2-L0} 降低了有效延迟时间 T_{delay} 。根据本发明, 这可以通过在视频缓冲区内引入角度切换标签而实现。角度切换标签标记位于比特流之内可能的无缝连接点。当根据本发明的调度程序填充视频比特缓冲区时, 其估计通过相关导航信息所给出的可能的无缝连接进入点, 并且使用标签来标记相对应的比特缓冲区位置。这可以通过各种方式实现, 例如, 通过将标签加入缓冲区内容中。在用于解码而读取视频比特流的同时, 标签逻辑上向下移动, 始终与相同的无缝连接相关联。但是, 实际上, 可以使用环形缓冲区或者 FIFO, 其有效地不移动缓冲区中的任何字节并保留相同的标签物理位置或地址。因此, 还可以代替标签, 存储例如分离缓冲区中无缝连接位置的地址, 并且通过将表示标签的地址与数据读取地址相比较, 来定位无缝连接进入点。

[0080] 有利地, 角度切换标签可以用于容易地确定视频缓冲区的以下部分: 当请求角度变化时被废弃的部分, 以及可以将其移除而不给正在运行的缓冲区带来危险的部分。根据本发明, 可以使用以下所描述的流切换算法。

[0081] 图 4a) 示出了典型的视频缓冲区, 其正被拾取器比特率 R 填充, 并且被读取解码器比特率 r_i 。当用户请求从当前角度 A1 到另一角度 A2 的视频角度改变时, 调度程序估计其目前已存储的角度切换标签 SL。当处理该请求时, 缓冲区被填充并达到当前填充级 L2, 并且可以包含各种角度切换标签 SL。在接收到请求之后, 将拾取器移动到可以读取新的视频比特率的位置需要最差情况的最小时间 t_{min} 。在该时间 t_{min} 期间, 由于角度切换应为无缝的, 所以视频缓冲区被恒定地读取并不会变空。通常根据以下关系来计算将在时间 t_{min} 期间读取的数据量所需的缓冲区空间:

$$[0082] \quad \text{buffer} = r_i \cdot \text{time} \quad (\text{eq. 13})$$

[0083] 由于根据 VBR, r_i 通常是可变的, 可以使用可能的最高速率应对最差情况。根据本发明, 将根据 eq. 13 所计算的缓冲区尺寸加到当前缓冲区读取位置 L0 中, 从而计算最早可能的切换位置 L1。此外, 在该位置 L1 上所发现的第一角度切换标签 $L2^*$ 是最早可能的角度切换位置。这里, 将该具体角度切换位置 $L2^*$ 称作“接合标签”。当调度程序从接合标签 $L2^*$ 开始下一次视频缓冲区填充过程时, 可以加载所请求角度 A2 的新视频内容。因此, 删除旧缓冲区的关于角度 A1 的一部分内容, 即, 从接合标签 $L2^*$ 到当前缓冲区填充级 L2, 并且由关于角度 A2 的新内容所代替。图 4b) 中示出了该情况。针对用户的有效延迟时间通过与旧角度 A1 相关的剩余数据量以及视频比特率来确定, 并且计算为:

$$[0084] \quad T^*_{delay} = \frac{B_{L2^*-L0}}{r_i} \quad (\text{eq. 14})$$

[0085] 由于 eq. 14 中 B_{L2^*-L0} 的小于 eq. 12 中的 B_{L2-L0} , 并且 r_i 在两个等式中相同, 所以缩短

了延迟时间 T_{delay}^* 。

[0086] 可以使用哪一个切换标签作为接合标签取决于最小可能切换时间 t_{min} 。这是拾取器开始移到新视频流与缓冲新视频数据之间的时间。其包括以前所提到的拾取器访问时间 T_{acc} 以及与 T_{acc} 相比非常小的中间处理时间。因此,如 eq. 14 中所描述,可以通过减小最小可能切换时间 t_{min} ,进一步减少真正切换视频角度之前的延迟时间。根据本发明,如下所述,这可以通过修改自由运行的调度程序模型而实现。当本发明的调度程序接收到角度切换请求时,其可以分析在队列中注册了哪些流缓冲区填充请求。根据排队的填充请求,其可以确定最小时间 t_{min} ,并且选择最优的接合标签。

[0087] 如果缓冲区不包含切换标签,拾取器可以继续读取旧的数据流,直到检测到切换标签,然后切换到新的数据流。

[0088] 有利地,针对无缝切换所描述的机制不局限于视频角度切换,而是可以用于任意类型的视频数据切换,或者用户操作的视频数据的再现,例如,当可以由另一视频场景代替一个视频场景时。此外,其不仅可以用于视频数据流,而且还可以用于其它数据流。具体地,例如,在视频角度切换的情况下可以针对音频数据流使用相同的机制,以便适应音频再现。此外,可以存在各种类型的切换标签,仅涉及具体数据流,并且包含相应的指示。在这种情况下,可以类似地应用本发明的方法。

[0089] 在本发明的其它实施例中,即使当出现角度切换请求时其已经在队列中,也可以修改请求的处理。调度程序可以执行队列中的所有请求,而不会过早地停止其执行,即,在完全填充流缓冲区之前。这节省了更多时间,并且减少了视频角度切换延迟时间。但是,所中断的流缓冲区填充处理必须服从

$$[0090] \quad T_{\text{fill,min}} = \sum_{i=1}^{\text{in queue}} \left(T_{\text{acc}} + \frac{b_i + \Delta b_i}{r_i} \right) \quad (\text{eq. 15})$$

[0091] Eq. 15 意味着在过早中断填充请求的流缓冲区之前,至少必须已经填充了基本缓冲区 b_i 和桥接缓冲区 Δb_i 。

[0092] 根据本发明的调度程序的另一个优点是启动 N 个流的完全的 OOM 解码的具体策略。启动时的正常延迟是由于 N 次拾取器跳跃和 N 次流缓冲区填充的累积。通过使用 eq. 2 和 eq. 6,启动延迟时间是

$$[0093] \quad T_{\text{startup}} = \sum_{i=1}^N \left(T_{\text{acc}} + \frac{b_i + \Delta b_i + bx_i}{R} \right) \quad (\text{eq. 16})$$

[0094] 因为解码器还没有运行,所以在 eq. 2 中分母较大,因此较快地填充缓冲区。

[0095] 根据本发明的启动步骤的最优化包括彼此独立的两个改进,并且在以下进行描述。

[0096] 第一个改进是在启动期间考虑要填充的缓冲区的数量。当自由运行的调度程序接收到启动命令时,其不完全加载 OOM 流缓冲区。与上述角度切换算法相同,其仅填充 OOM 流缓冲区的一部分,并且最后服务于具有最高比特率的比特流。以以上应用示例为例,启动时,调度程序首先填充用于音频和字幕的 OOM 流缓冲区,并且然后填充 OOM 视频流缓冲区。此外,针对除最后 OOM 流以外的所有流,其仅需要加载基本缓冲区 b_i 和桥接缓冲区 Δb_i 。此外,对于除最后 OOM 流以外的所有流,设置 OOM 流缓冲区重新填充请求。然后,可以完全加

载最后的流缓冲区,但是,当仅填充了最后 OOM 流缓冲区的基本缓冲区部分 b_i 时,可以开始针对所有流的解码。因此,启动延迟是

$$[0097] \quad T_{startup}^* = N \cdot T_{acc} + \frac{b_{i=r_{i,max}}}{R} + \sum_{i=1, i \neq r_{i,max}}^{N-1} \frac{b_i + \Delta b_i}{R} \quad (\text{eq. 17})$$

[0098] 在该启动步骤之后,自由运行的调度程序可以如以上所述正常工作,并且可以通过桥缓冲区 Δb_i 来桥接所有排队的缓冲区填充请求。

[0099] 根据本发明,针对启动过程的第二个改进涉及 OOM 流缓冲区的序列,即,除了最后之外的所有 OOM 流缓冲区的加载顺序。eq. 17 的主要部分是拾取器访问时间 T_{acc} ,其是物理参数,即在最差情况下整个冲程跳跃所需要的时间,并且因此无法缩减。根据本发明,由于第一加数“ $N \cdot T_{acc}$ ”是最差情况值,所以可以减小 eq. 17 的第一乘积。可以根据以下方法将其最小化。

[0100] 当开始读取盘时,读取设备确定其包含哪些文件以及所述文件在盘上的物理位置。因此,调度程序可以具有该信息,并且选择 OOM 流缓冲区填充命令,其取决于 OOM 流文件在盘上的物理位置。根据哪一个距离当前拾取器位置较近,调度程序可以从最外部或者最内部的物理盘位置处的 OOM 流缓冲区开始。此后,调度程序将物理上最近的 OOM 流加载到其各自的 OOM 流缓冲区。重复进行该处理,直到仅剩最后具有最高比特率的流。独立于其在盘上的物理位置,最后加载具有最高比特率的 OOM 流。

[0101] 有利地,当使用该启动过程时,拾取器的移动将不超过横跨整个盘两次。这对应于两个完整的冲程跳跃。产生的启动延迟是

$$[0102] \quad T_{startup}^{**} = 2 \cdot T_{acc} + \frac{b_{i=r_{i,max}}}{R} + \sum_{i=1, i \neq r_{i,max}}^{N-1} \frac{b_i + \Delta b_i}{R} \quad (\text{eq. 18})$$

[0103] Eq. 18 示出了当 $N > 2$ 时,来自 Eq. 17 的主要加数“ $N \cdot T_{acc}$ ”减小,并且独立于所处理的流的数目,因此缩减了启动时间。

[0104] 本发明还可以应用于不是所有流都要求缓冲的系统中。例如,可以存在介质上所包括的另一数据流,其包含不重复或者不周期性读取的数据,例如,仅在呈现开始时一次,并且可以以拾取器数据率进行处理,因此,不需要缓冲。

[0105] 例如所有类型的介质再现设备可以使用本发明的方法,所述设备包含光学或非光学拾取器,具体是蓝光盘播放器。

[0106] 有利地,还可以使用这里所公开的方法,从而针对所描述的应用最优地划分给定数量的缓冲空间。

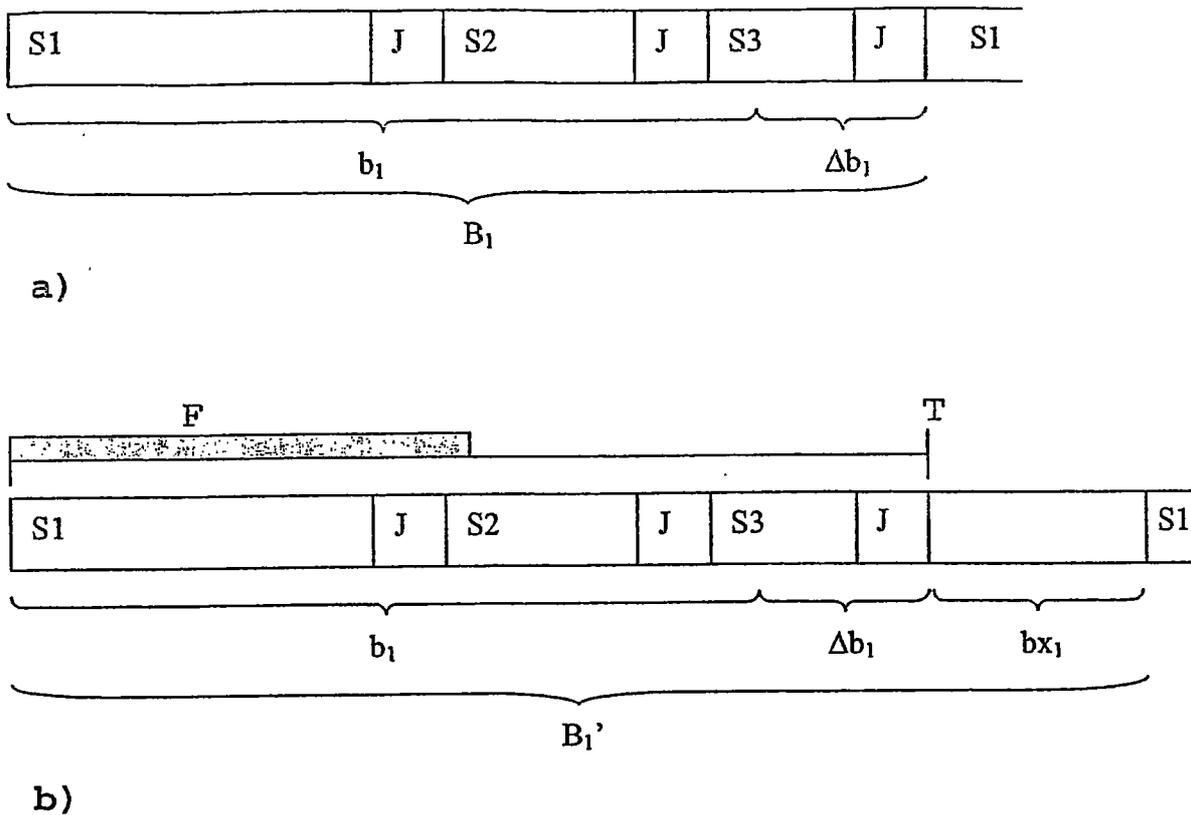
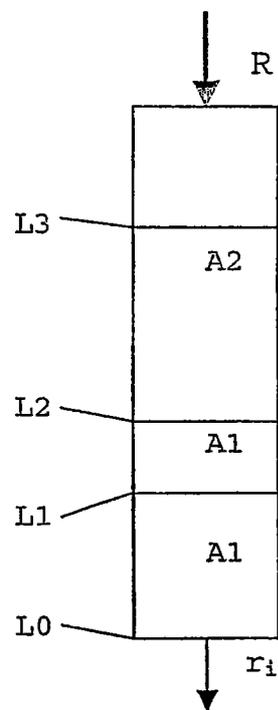
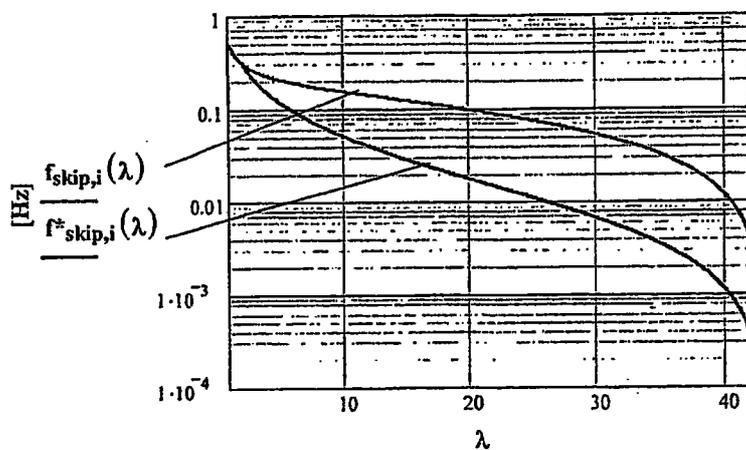


图 1



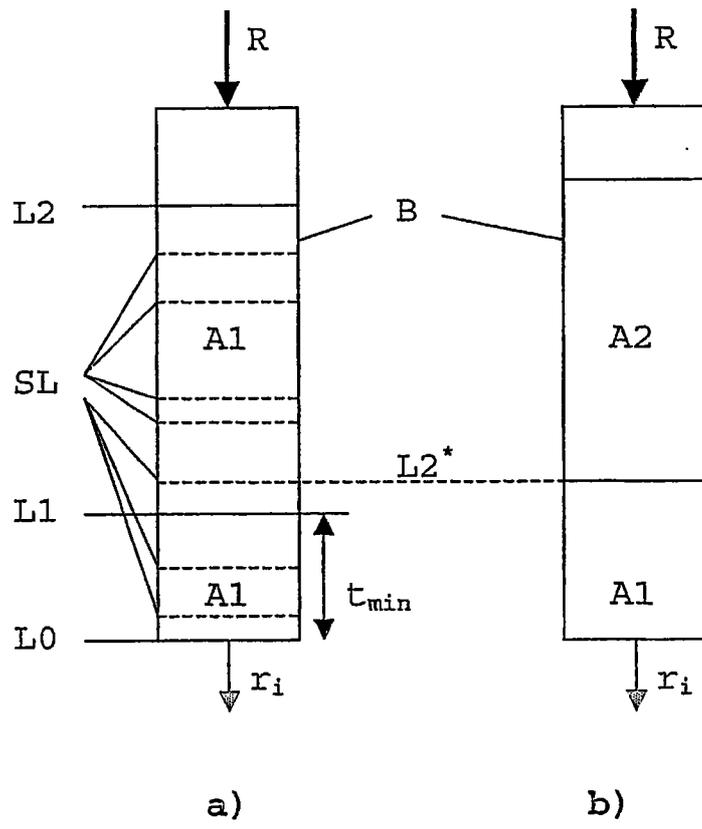


图 4