

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 200410061909.4

[45] 授权公告日 2009 年 7 月 22 日

[11] 授权公告号 CN 100517306C

[22] 申请日 2004.6.24

[21] 申请号 200410061909.4

[30] 优先权

[32] 2003.6.25 [33] US [31] 10/603,328

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 G·T·顿巴尔 E·鲁多尔弗

S·B·莫哈米德

[56] 参考文献

EP1065589A1 2001.1.3

US5625845A 1997.4.29

Microsoft DirectShow: A New Media Architecture. Amit Chatterjee, Andrew Maltz. SMPTE Journal, Vol. 106 No. 12. 1997

审查员 赵晓敏

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 钱慰民

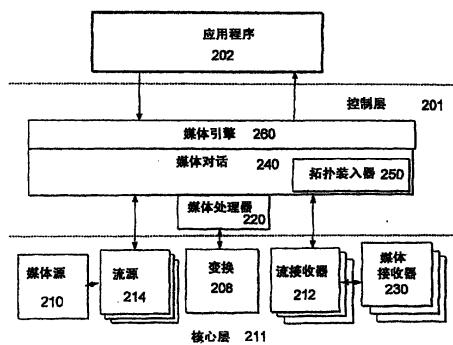
权利要求书 3 页 说明书 10 页 附图 6 页

[54] 发明名称

媒体基础媒体处理器

[57] 摘要

一种媒体处理器把拓扑结构形成和维护功能与通过拓扑结构处理数据的功能相分离的系统与方法。所述系统包括控制层和媒体处理器，前者含有生成拓扑结构的拓扑发生单元，所述拓扑结构描绘成组输入多媒体流、一个或多个输入多媒体流的源、对多媒体数据执行的操作次序和一组输出多媒体流；后者管理拓扑结构描绘的多媒体数据的传送，并按制对多媒体数据的多媒体操作次序的执行以形成一组输出多媒体流。核心层包括输入媒体流，输入多媒体流的源、处理多媒体数据的一种或多种变换，流接收器和提供一组输出多媒体流的媒体接收器。



1、一种处理多媒体数据的方法，其特征在于，所述方法包括：

在拓扑发生单元的一个或多个多媒体元件之间建立一种连接拓扑结构，所述拓扑结构描述：一组输入多媒体流、输入多媒体流的一个或多个源，对多媒体数据所执行的操作次序、以及一组输出多媒体流；

向媒体处理器发送所述拓扑结构；和

按拓扑结构传送数据，传送受制于媒体处理器，其中该媒体处理器通过：

从用户处接收给定数据速率，

询问拓扑结构中的源和多媒体元件，

如果拓扑结构中的源和多媒体元件均支持给定数据速率，则将给定数据速率设置给源和多媒体元件，

来控制数据传送的数据速率。

2、如权利要求1的所述方法，其特征在于，还包括对多媒体数据执行多媒体操作次序，以产生一组输出多媒体流。

3、如权利要求1的所述方法，其特征在于，所述多媒体元件是软件对象。

4、如权利要求1的所述方法，其特征在于，所述拓扑发生单元是拓扑装入器。

5、如权利要求1所述的方法，其特征在于，所述拓扑发生单元是应用程序。

6、如权利要求1所述的方法，其特征在于，所述媒体处理器把多媒体数据显现给应用程序。

7、如权利要求1所述的方法，其特征在于，所述媒体处理器通过被配置成媒体接收器而接收多媒体数据。

8、一种处理多媒体数据的系统，其特征在于，所述系统包括：

配置成从应用程序接收指令的控制层，所述控制层包括：

配置成生成拓扑结构的拓扑发生单元，所述拓扑结构描述：一组输入多媒体流、输入多媒体流的一个或多个源、对多媒体数据执行的操作次序、以及一组输出多媒体流；

耦接到控制层的核心层，所述核心层配置成包括：输入多媒体流，输入多媒体流的源，配置成处理多媒体源数据的一个或多个变换，耦接到控制层的一个或多个流接收器，和配置成提供一组输出多媒体流的一个或多个媒体接收器；和

媒体处理器，配置成管理由拓扑结构描述的多媒体数据的传送，并调整对多媒体数据的多媒体操作次序的执行，以形成一组输出多媒体流，其中该媒体处理器通过：

从用户处接收给定数据速率，

询问拓扑结构中的源和多媒体元件，

如果拓扑结构中的源和多媒体元件均支持给定数据速率，则将给定数据速率设置给源和多媒体元件，

来控制多媒体数据传送的数据速率。

9、如权利要求 8 所述的系统，其特征在于，所述多媒体元件是软件对象。

10、如权利要求 8 所述的系统，其特征在于，所述拓扑发生单元是拓扑装入器。

11、如权利要求 8 所述的系统，其特征在于，所述拓扑发生单元是应用程序。

12、如权利要求 8 所述的系统，其特征在于，所述媒体处理器把多媒体数据显露给应用程序。

13、如权利要求 8 所述的系统，其特征在于，媒体处理器通过被配置成媒体接收器而接收多媒体数据。

14、一种用于处理多媒体数据的设备，其特征在于，所述设备包括：

用于形成拓扑发生单元中一个或多个多媒体元件之间的连接拓扑结构的装置，所述拓扑结构描述：一组输入多媒体流、输入多媒体流的一个或多个源、对多媒体数据执行的操作次序、以及一组输出多媒体流；

用于向媒体处理器发送所述拓扑结构的装置；和

用于按所述拓扑结构传送数据的装置，该传送受制于媒体处理器，其中该媒体处理器通过：

从用户处接收给定数据速率，

询问拓扑结构中的源和多媒体元件，

如果拓扑结构中的源和多媒体元件均支持给定数据速率，则将给定数据速率设置给源和多媒体元件，

来控制数据传送的数据速率。

15、如权利要求 14 所述的设备，其特征在于，还包括用于对多媒体数据执行多媒体操作次序以形成一组输出多媒体流的装置。

16、如权利要求 14 所述的设备，其特征在于，所述多媒体元件是软件对象。

17、如权利要求 14 所述的设备，其特征在于，所述拓朴发生单元是拓朴装入器。

18、如权利要求 14 所述的设备，其特征在于，所述拓朴发生单元是应用程序。

19、如权利要求 14 所述的设备，其特征在于，所述媒体处理器对应用程序显露多媒体数据。

20、如权利要求 14 所述的设备，其特征在于，所述媒体处理器通过被配置成媒体接收器而接收多媒体数据。

媒体基础媒体处理器

技术领域

本发明一般涉及计算，尤其涉及处理计算环境中的多媒体数据。

发明背景

由于计算机能力扩展成一度要求分立电子元件的娱乐形式，所以希望提高效率与用户友善性。一种方法就是 Microsoft® 的 Direfshow®, 它能回放来自本机文件或因特网服务器的多媒体流，捕获诸设备的多媒体流，并对多媒体流作格式转换。它能回放文件类型的视频与音频内容，诸如 Windows Media Audio、Windows Media Video、MPEG、Apple® Quick Time®、Audio Video Interleaved (AVI) 与 WAV。Direct-Show® 包括可插过滤元件的系统。过滤器 (filter) 是支持 Direct Show® 接口并能通过对文件读、复制、修改和写数据而处理数据流的对象。基本的过滤器类型包括源过滤器，它从某个源如盘上文件、外围馈源 (satellite feed)、因特网服务器或 VCR 获取数据，并把它送入过滤器网络 (filter graph) (一种过滤器连接)。过滤器网络提供转换数据格式的变换过滤器、接收并发送数据的同步与源过滤器，以及呈递数据诸如把数据呈递给显示设备的呈递过滤器。也可将数据呈递给任一接收媒体的地点。包括在 Direct Show® 的其它类型过滤器有效果过滤器和语法分析程序过滤器，前者增设效果而不改变数据类型，后者弄清源数据的格式而且知道如何读正确的字节、形成记时印记和执行搜寻。

因此，所有数据与大量控制信息在过滤器之间传送。在用引线连接诸过滤器时，就形成了过滤器网络。为了管理数据和过滤器网络的连接，Direct Show® 包括一过滤器网络管理器，它帮助过滤器肯定以正确顺序连接，但数据和大多数控制不通过过滤器网络管理器。诸过滤器必须正确连接，例如该管理器必须搜索某种呈递配置，确定现有的过滤器类型，对给出的数据类型以正确的顺序连接诸过滤器，并提供一合适的呈递过滤器。

虽然过滤器允许大量重复使用程序，但是应用过滤器也造成某些预计不到的问题。过滤器造成的问题之一是对过滤器形成了大量 API。各过滤器基本上都有一个独立的 API，因而给定的过滤器必须能与同其连接的每一过滤器的 API 接口。而且，过滤器的应用还会造成让一有问题的给定过滤器停止工作的问题。当网络内某给定过滤器停止工作时，与其接口的任一过滤器就要求一不同的关联的接口。由于失去接口时

不知道过滤器的状态,故通常难以对该过滤器作编程而妥善处理该接口的失去。因此,失去接口易导致不可预测的过滤器行为,最终导致操作异常的程序。再者,Direct Show®里的整个控制分布在两个程序块之间,过滤器间的接口控制数据流,而过滤器管理器控制着过滤器的例示与除去。以这种方式分配控制,因显然存在某些跨越程序块之间边界的控制功能,故软件设计有麻烦。Direct Show®的另一问题是过滤器要负起媒体格式交涉与缓冲器管理功能的责任,为此,一些过滤器要与另一些过滤器联系。对过滤器的依赖性使在Direct Show上建立的应用程序容易受到编程至过滤器的错误与无效性的影响,因而编写不良的过滤器很容易降低过滤器网络和与该过滤器网络关联的应用程序的质量。

需要解决Direct Show®结构的这些问题,尤其要改进对多媒体数据处理的控制,并且对多媒体元件之间的通信解决对过滤器的依赖性。

发明内容

因此,处理多媒体数据的系统与方法把控制功能与数据处理功能分开,以有效地处理多媒体流。一种方法在拓扑发生单元的一个或多个多媒体元件之间形成连接拓扑结构,该拓扑结构描绘一组输入多媒体流、输入多媒体流的一个或多个源,对多媒体数据执行的操作次序和一组输出多媒体流。该方法还把拓扑结构发送到媒体处理器,并按拓扑结构传送数据,而传送受媒体处理器控制。拓扑发生单元可以是拓扑装入器或应用程序,在媒体处理器控制的范围之外工作。媒体处理器控制对多媒体数据执行的多媒体操作次序,形成一组输出多媒体流。在一实施例中,多媒体元件是软件对象。

另一实施例方法,该方法在媒体处理器工作时,由该媒体处理器改变在使用的第一拓扑结构。按照该方法,媒体处理器保留其目前状态,接收把第一拓扑结构转换成第二拓扑结构的指令,并根据指令把第一拓扑结构更新为第二拓扑结构。该指令包括第一与第二拓扑结构的差异。在把第一拓扑结构更新为第二拓扑结构之后,媒体处理器恢复接口活动,把消息发送给应用程序。在拓扑结构改变之前,媒体处理器被配置成允许消息呼叫。转换第一拓扑结构的指令可通过来自外部源的消息而接收,以启动改变第一拓扑结构的过程。

另一实施例提出的方法用来决定如何应用一组多媒体元件对一个或多个多媒体数据流执行多媒体操作次序。该方法为递归法,因为多媒体元件的应用是通过询问现有样本数据先前的元件确定的。该方法包括寻找一个或多个输出端接至某接收设备输入端的多媒体元件,询问多媒体元件以判断有无样本,若无样本,则询问可包括检查对多媒体元件的输入。若输入无现成样本,则检查对多媒体元件馈送样本的一媒体源。若该媒体源没有现成的样本,该方法就结束文件功能,或宣布错误状态。若有样本,该方法就将该样本移到多媒体元件中的下一个多媒体元件。

另一实施例提出的方法用于检索一段媒体流，称为“清洗”。该法包括高速缓存这段媒体流，而高速缓存的这段媒体流包含该媒体流的呈现点。然后该方法接收外部源对媒体处理器提出的请求，以检索高速缓存的该段媒体流，并搜索识别是否高速缓存了该段媒体流。若该段媒体流被高速缓存，该方法就转移该请求的高速缓存段。该高速缓存可根据用户在应用程序中的设定，该设定包括是否高速缓存，准备包含在高速缓存段内的帧数与样本数。在一实施例中，缓存段是连续的视频数据。

附图简介

虽然所附如权利要求具体列出了本发明诸特征，但通过下面结合附图所作的详述，能更好地理解本发明及其目的与优点，其特征在于，：

图 1 是概括地说明能实施本发明的示例分布式计算系统的框图。

图 2 是本发明诸实施例的媒体基础系统的框图。

图 3 是本发明诸实施例要求播放 DVD 的媒体引擎中一例数据的流程图。

图 4 是本发明诸实施例在媒体处理器中如何实现数据流的框图。

图 5 是本发明诸实施例中媒体处理器动态拓扑变化的流程图。

图 6 是本发明诸实施例示例说明清洗过程的流程图。

发明的详细描述

参照图 1，图示为可实施本发明的一示例计算设备 100 它只是一例合适的计算设备，并不用来限制本发明的使用或功能的范围，比如它并不等于图 1 所示的任一计算设备 10-17。该示例的计算设备 100 能实现一个或多个计算设备 10-17，如通过存储器划分、虚拟机器或类似的编程技术，使一个物理上的计算结构执行下述属于多个结构的动作。

本发明在一般计算机可执行指令如计算机执行的程序模块的范围内描述。通常，程序模块包括执行具体任务或构成具体抽象数据类型的例行子程序、程序、对象、元件、数据结构等。在分布计算环境中，任务由通过通信网联接的远程处理设备完成。在分布计算环境中，程序模块可以设置在包含存储器存贮设备的本机和远程计算机存贮媒体内。

计算机设备 100 的元件包括但不限于处理单元 120、系统存储器 130 和系统总线 121，后者把各系统元件包括系统存储器耦合至处理单元 120。系统总线 121 可以是若干种总线结构中的任一种，包括存储器总线或存储器控制器，外设总线和应用任一种总线体系结构的局域总线。作为举例，但不限制于，这种结构包括工业标准结构（ISA）总线、微通道结构（MCA）总线、增强的 ISA（EISA）线、视频电子学标准协会（VESA 局域总线）和称为夹层（Mezzanine）总线的外设元件互连（PCI）总线。

计算设备 100 一般包括各种计算机可读媒体。计算机可读媒体可以是任一种能被计算设备 100 访问的市售媒体, 包括易失与非易失媒体、可拆装与不可拆装媒体。作为举例, 但不限制于, 计算机可读媒体可以包括计算机存贮媒体和通信媒体。计算机存贮媒体包括易失与非易失、可拆装与不可拆装两类以任何方法或技术构制的媒体, 用于存贮信息, 如计算机可读指令、数据结构、程序模块或其它数据。计算机存贮媒体包括但不限于: RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字化通用光盘 (DVD) 或其它光盘存储器、盒式磁带、磁带、磁盘存储器或其它磁性存储设备, 或任何其它可存贮所需信息并被计算设备 100 存取的媒体。通信媒体一般包含计算机可读指令、数据结构、程序模块或其它调制的数据信号如载波或其它传输机构里的数据, 而且包括任何信息传输媒体。术语“调制的数据信号”指具有其一个或多个特性组的信号或关于在信号中编码信息的方式改变的一种信号。作为举例, 但不限制于, 通信媒体包括有线网或直接有线连接等有线媒体和声、射频、红外等无线媒体与其它无线媒体。计算机可读媒体的范围还包括上述的各种组合。

系统存储器 130 包括易失和/或非易失存储器形式的计算机存贮媒体, 如只读存储器 (ROM) 131 与随机存取存储器 (RAM) 132。基本输入/输出系统 133 (BIOS) 一般存入 ROM131, 包含如在启动期间有助于计算机 110 内诸单元之间传递信息的基本例程。RAM132 一般包含处理单元 120 可立即存取和/或当前操作的数据和/或程序模块。作为举例, 但不限制于, 图 1 示出了操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137。

计算设备 100 还可包括其它可拆装/不可拆装、易失/非易失的计算机存贮媒体。作为举例, 图 1 示出了对不可拆装的非易失磁媒体读或写的硬盘驱动器 141、对可拆装的非易失磁盘 152 读或写的磁盘驱动器 151, 以及对可拆装的非易失光盘 156 如 CD-ROM 或其它光媒体读或写的光盘驱动器 155。能用于示例操作环境的其它可拆装/不可拆装、易失/非易计算机存贮媒体, 包括但不限于: 盒式磁带、快速存储卡、数字通用盘、数字像带、固态 RAM、固态 ROM 等。硬盘驱动器 141 一般通过不可拆装存储器接口如接口 140 连接系统总线 121, 磁盘驱动器 151 和光盘驱动器 155 一般通过可拆装存储器接口如接口 150 连接系统总线 121。

上述和图 1 所示的驱动器及其有关的计算机存贮媒体, 为计算设备 100 存贮计算机可读指令、数据结构、程序模块与其它数据。如在图 1 中, 硬盘驱动器 141 示成存贮操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147。注意, 这些元件可同操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同或不同。操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 在这里给出不同的编号最低限度的说明, 它们是不同的副本。用户可通过输入设备如键盘 162 和点击设备 161 (通称为鼠标、轨迹球或接触垫) 等把命令与信息输入计算设备 100。其它输

入设备（未示出）包括话筒、操纵杆、游戏垫、碟形卫星天线、扫描仪等。这类输入设备往往通过耦接系统总线的用户输入接口 160 连接处理单元 120，也可用其它接口和总线结构如平行端口、游戏端口或通用串行总线（USB）等连接。监视器 191 或其它显示设备也可经接口如视频接口 190 连接系统总线 121。除了监视器，计算机还包括其它可经输出外围接口 195 连接的其它外围输出设备，如扬声器 147 与打印机 196 等。

计算设备 100 利用与一台或多台远程计算机的逻辑连接，工作于如图 1 的联网环境中。图 1 示出与远程计算设备 180 的一般网络连接 171，它可以是任一种不同类型的网络连接，包括局域网（LAN）、广域网（WAW），符合以太网协议、令牌环网（Token-Ring）协议的网或者如因特网或万维网等其它逻辑的或物理的网。

用于网络环境时，计算设备 100 通过网络接口或适配器 170（可以是网络接口卡、调制解调器或类似的上网设备）连接一般的网络连接 171。在联网环境中，描绘相关计算设备 100 或其部分的程序模块被存储在远程存储器存贮设备中。本领域的技术人员应明白，图示的网络连接为示例情况，可以应用在计算机之间建立通信链路的其它方法。

在以下描述中，将参照一台或多台计算设备的执行操作的动作与符号表示来描述本发明，除非另行指明。因此应理解，常说的正被计算机执行的此类动作与操作，包括计算设备的处理单元对代表构成形式的数据的电信号的处理。该处理对数据作变换，或把它保持在计算设备存储器系统里的存贮单元里，以本领域技术人员理解的方式重新配置或改变计算设备的操作。保持数据的数据结构是存储器中具有数据格式限定的具体特性的物理位置。然而，本发明虽以前述范围描述，但并不表示限制，本领域的技术人员应明白，下述的若干动作和操作也可在硬件中实现。

参照图 2，框图示出一媒体基础系统 200。核心层 211 包括媒体源 210、变换 208 和媒体接收器 230。基础系统 200 图示成耦接应用程序 202，接收和发送媒体流。控制层 201 包括媒体引擎 260、媒体对话 240、媒体处理器 220 和拓朴装入器 250。数据通过媒体基础 200 从媒体源 210 开始流入媒体对话 240，再流入媒体处理器 220。数据将从媒体处理器 220 流入媒体对话 240，再流入媒体处理器 220。数据将从媒体处理器 220 流入变换 208 并回到媒体处理器 220 一次或多次。然后，数据从媒体处理器 220 流入媒体对话 240，再流入流接收器 212。媒体引擎 260 控制与应用程序 202 的接口并对控制层 201 作总控制，而拓朴装入器 250 保证出现以拓朴结构规定的事件。媒体基础系统 200 提供连接诸流动媒体对象的接口与布设。

核心层包括媒体源元件 210 和媒体接收元件 230，还包括在媒体源 210 控制下操作的流源 214 和在媒体接收器 230 控制下操作的接收器 212。流源 214 把多媒体数据从存贮或捕获设备传递到控制层 201，流接收器 212 把多媒体数据从媒体引擎 260 传递到呈递或存贮设备（未示出）。媒体源元件 210 构成控制流源 214 的状态机，媒

体接收器元件 230 构成控制流接收器 212 的状态机。在每一种场合，状态处理与数据移动相分离。

媒体源 210、媒体接收器 230 和变换 208 同流源 214 与流接收器 212 一同包括构成核心层 211 部分的诸对象，这些元件都是实施预定功能的有计划性的对象。媒体源 210 和流源 214 或是捕获或是检索多媒体数据，并把该数据供给媒体对话 240。数据源包括但不限于：诸如硬驱器、CD 或 DVD 等盘片、因特网、随机存取存储器(RAM)、视频 RAM、电视摄像机、扫描仪、静像摄像机与话筒。媒体接收器 230 包括控制流接收器 212 内数据传递的对象。流接收器 212 包括接收来自控制层 201 用于存贮或呈递的数据的对象。数据接收器包括但不限于：诸如硬驱器、可写 CD 或可写 DVD 等盘片、因特网等计算机网上的广播、打印机、监视器等显示设备和扬声器。媒体源 210 和媒体接收器 230 的数据都可通过众多媒体传输，这类媒体包括但不限于：以太网、无线网、数字化前的模拟电缆、USB、IEEE1384、平行端口、串行端口和磁盘接口。

变换 208 包括处理数据的对象，这些变换包括编码器、译码器、分离器、复用器、添加混响效果的高低音控制等音频处理、调节色障板等视频处理、图像锐化和反差与亮度控制。编码器和译码器处理音频、视频与图像数据。视频数据类型包括 MPEG、Apple Quick time[®]、AVI 和 H.263 与 Windows Media Video(WMV)。注意，许多视频标准是真正的多媒体标准，因为这些标准具有传递音频与视频的条款。图像数据格式包括 JPEG、GIF、Fax 与 Tiff。音频标准包括 MP3、PCM、ADPCM 以及 CD 播放与 Windows Media Audio (WMA) 的标准。变换 208 用于转换数据格式，如把 JPEG 图像转换成适于在监视器上显示的格式。

参照图 3，图示为典型多媒体处理的流程图。假定用户想观看 DVD。将发出应用程序让用户观看 DVD，而该应用程序的图形用户接口 (GUI) 让用户执行播放、停止、暂停、快进与倒片等功能。在方框 302，用户选择播放钮、应用程序向媒体基础的媒体引擎元件发消息。该信息包含应用程序想观看 DVD 的信息。在方框 304，媒体引擎向媒体对话发消息，拓朴装入器命令这些程序块开始播放 DVD。在方框 306，拓朴装入器设置拓朴结构，该结构提供一条数据流通过媒体与流源、变换和媒体与流接收器所经过的路径。在方框 308，拓朴装入器把该拓朴结构传给媒体处理器，后者设置和执行该拓朴结构。在方框 310，媒体处理器向核心层元件发送消息，例示为拓朴装入器调出的对象。除了以正确的次序调用核心层诸对象并在对象间传送数据外，还要控制数据速率，使音频视频同步并按期望的速率呈递。数据速率可在媒体对话中确定。在方框 312，媒体对话将询问各对象以确定能支持的期望速率，并向媒体处理器传送带速率信息。在方框 314，媒体处理器确定核心层中确定对象速率的时钟速率，通常为媒体接收器是时钟速率，并设置该时钟速率。在方框 316，媒体处理器调用核心层，按拓朴结构所需在对象之间传送数据。该数据最后由媒体接收器呈递给扬声器与监视器。

媒体处理器是实现拓扑结构所描绘的数据流的对象，用描绘数据流的拓扑结构初始化，并经媒体源接口暴露于用户。因此，一经配置，媒体处理器实际上对用户像一个媒体源。媒体处理器显露若干基于该拓扑结构的媒体流，拓扑结构的各输出节点有一媒体流。

图4示出一种拓扑结构。用户调用驱动数据流，以对媒体处理器的多个媒体流之一取一样本。研究图4的上部路径。该数据流从拓扑结构的输出对象（接收器对象412）开始工作，通过产生数据的对象清单递归地移动。图4上部路径的处理从接收器对象412开始。连接输入416的唯一输出节点是连接变换对象408的输出418，若变换对象408具有现成的样本，则媒体处理器220读该样本并把它写到接收器对象412。若变换对象408没有现成的样本，那么媒体处理器220就察看变换对象408的输入端420，它接至变换对象404的输出422，于是对变换对象404询问样本。若有样本，媒体处理器220应读该样本并把它写到变换对象408；若没有样本，则媒体处理器220再左移一个节点，对媒体流对象403询问现成的样本。媒体流对象403经源对象402装入。若样本从媒体流对象403得到，就把它从媒体流对象403检索到变换对象404；若未发现样本，媒体处理器将请求该媒体流对象403读源。在找到有效样本并传给下一方框时，处理再次开始。数据一旦找出，通过诸对象操作而完成该处理。变换对象404处理该数据并把变换的数据传给变换对象408，而后者处理该数据并把变换的数据传给接收器对象412，完成在输出端生成样本的处理。在一实施例中，媒体处理器保持具有其它对象输入的对象清单。利用该清单，媒体处理器能察看输出对象，并确定该媒体处理器需要从那个对象来检索媒体数据。

在一实施例中，该清单跟踪持有诸对象的节点的类型和其它对象信息。各类对象按媒体处理器请求生成数据的方式不同。变换节点包含变换对象，拥有一组用来提供输入媒体样本并生成输出媒体样本的调用。T形节点提供表示媒体处理器需要复制样本。在开始对它们说明之前，应对T形节点进行描述。源节点有一媒体流，可提供媒体处理器用于向该流要求数据的异步调用，然后该流在有数据时提供数据。

诸实施例针对应用媒体处理器220的数据流。在一实施例中，数据流为异步，即用户要求为某一媒体流生成数据，于是媒体处理器220生成该数据，并在有数据时通知用户。在一实施例中，拓扑结构引用的诸元件相互不调用，而由媒体处理器220负责所有的通信。利用中央定位的通信方法，数据流相协调，元件之间有更大的互操作性。

在一实施例中，媒体处理器220不寻址数据流的每一方面，如在一实施例中，媒体处理器220不控制媒体样本。

就整个媒体基础体系结构而言，媒体处理器常用于媒体引擎的范围。在应用图2的媒体基础结构时，媒体引擎260是应用程序202直接使用的元件。媒体引擎260可用应用程序202配置以规定数据源（文件名称，URL，装置或通常为规定多个简单源的复杂配置）与数据目的地（如输出多媒体文件或呈递设备如视频卡）。然后，在

运行时通过启、停等命令控制媒体引擎 260。这样，媒体引擎 260 用其它媒体基础元件实现这一功能，而且是用应用程序 202 执行多媒体操作的主要目标。媒体处理器 220 运用媒体处理对话直接控制，并与拓扑装入器 250 一起使用。媒体处理对话是媒体引擎 260 用来操作媒体源 210、媒体处理器 220、媒体接收器 230 与拓扑装入器 250 的对象。

在一实施例中，媒体处理器 220 支持 T 形对象和具有一个以上输入或输出的变换对象，例如转换通常实施为变换对象，而后者有两个输入和一个输出。

在节点有一个以上输入时，媒体处理器 220 应用下述方法：试图生成变换的输入数据时，媒体处理器 220 根据前面媒体样本的记时印记只选择一个输入，并生成该输入的数据。媒体处理器 220 每次向变换对象提供输入样本，就试图生成该变换的输出数据。若变换不生成任何数据，媒体处理器 220 就向变换对象提供输入样本，可能是已用过的同一个输入。

在节点有一个以上输出时，媒体处理器 220 就要求更多的输出信息。拓扑装入器指明或是主要的或是可丢弃的输出的特性。

主要输出对传给输入节点的样本用作主存储器分配器。可丢弃输出不保证所有的样本都进入该节点；若用户在生成了输入样本而不请求该输出的样本，则可丢弃输出绝对不接收该样本。在宁可丢失数据而不愿减慢处理的情况下，如编码器应用中的预检显示，可丢弃的理念是有用的。

多媒体系统的一个主要功能是在运行时改变多媒体对话的特性，例如重放应用程序可将视窗模式的重放切换成全屏模式，或者编码应用程序可从一台摄像机的编码切换到另一台摄像机编码。揭示的媒体基础系统通过拓扑结构的变化来处理这些功能。

但在在一实施例中，媒体处理器 220 从不改变拓扑结构本身；拓扑结构总是由另一元件改变并通知该媒体处理器。若媒体处理器 220 正被媒体对话 240 控制，则后者用拓扑装入器对媒体处理器 220 建立完整的拓扑结构，然后把这些结构通知媒体处理器。但在在一实施例中，媒体处理器的任一直接用户也可对媒体处理器作出拓扑结构变化。

在一实施例中，拓扑结构变化可以是静态的或动态的。在媒体处理器 220 不处理数据时，可作静态的拓扑结构变化，用新的拓扑结构完全取代老的拓扑结构。在媒体处理器运行时，可作动态的拓扑结构变化，也可只改变部分拓扑结构，其它部分的拓扑结构保持不变。

有几种可产生拓扑结构动态变化的机理。第一种是媒体源生成的，此时媒体处理器 220 的内部媒体源之一检测其格式以某种方式发生了变化；某一媒体流改变了，某一媒体流结束了，或新的媒体流产生了。媒体源通知媒体处理器 220 所发生的变化；媒体处理器 220 把该通知传给用户以处理和停止处理数据。用户负责建立新的拓扑结构并把它送到媒体处理器。第二种是拓扑结构生成的，此时该拓扑结构本身包含其将

在某一点发生变化的信息：一个或多个拓扑对象有设定为特性的有效期，在媒体处理器 220 检出该有效期到了时，就通知用户并停止处理数据。最后一种用户生成的，此时在媒体处理器 220 运行时，用户仅对其设置新的拓扑结构。在所有情况下，媒体处理器 220 以同样的方式对拓扑变化作出反应。

参照图 5，流程图示出拓扑结构动态变化的步骤。在方框 502，媒体处理器要求拓扑结构变化。在方框 504，媒体处理器 220 首先把所需要的通知用户例如媒体源已改变格式。在方框 506，媒体处理器 220 停止处理并保持其状态，使它能在保留在新的拓扑结构物里的任何节点上恢复处理。媒体处理器 220 还可将其置于某一状态，使用户要等到完成了拓扑结构变化后才能继续运行。在方框 508，媒体处理器 220 接收新的拓扑结构，后者来自拓扑装入器或应用程序。在方框 510，媒体处理器 220 更新其内部的拓扑结构以与新的拓扑结构相配。在方框 512，媒体处理器 220 通知应用程序该新的拓扑结构已就位，并恢复处理。

媒体处理器 220 设计成支持以任意速率运行。媒体源接口是一种拉（pull）接口，即用户能尽快地拉出样本，因而从这一点上讲，媒体处理器 220 自动地支持任意的速率。但在许多场合中，媒体源或变换知道了速率后，还具有优化的或改进的特性。因此，当用户明确地告诉媒体处理器 220 以给定速率运行时，媒体处理器 220 将询问任一媒体源与拓扑结构里的变换，以确定其速率特性。若所有的媒体源与变换都支持给定的速率，媒体处理器 220 就设置所有媒体源对象与变换对象的速率，使这些对象适当改变它们的特性。反方向运行多媒体流，则是速率变化的特例。

清洗被定义为能在多媒体呈现内迅速地查看，并检索该位置的少量媒体样本。鉴于多媒体数据译码的复杂性，清洗比查到多媒体呈现里面更复杂，所花时间更长，这对在独立可译码的关键帧之间通常有长时延的压缩视频数据尤其如此。

为了优化清洗，媒体处理器 220 拥有逻辑，能高速缓存通常在当前呈现位置附近一定量的预译码样本，因而寻找这些预译码样本之一能让媒体处理器 220 及时地生成期望的样本。这种高速缓存特性可由应用程序配置，以在存储器应用与良好的清洗性能之间实现折衷。

图 6 是用于优化清洗的处理流程图。方框 602 示出媒体处理器 220 对一节点生成的每个样本应用有关样本数据高速缓存的逻辑。判断方框 604 表示，在判断是否高速缓存样本数据时，媒体处理器先要判断该样本数据是否已被高速缓存；若被高速缓存了，方框 606 就不做什么；若未被高速缓存，则方框 608 规定媒体处理器 220 确定用户设置的高速缓存设定值。高速缓存设定包括希望对那个节点高速缓存的节点标识符，高速缓存频率、高速缓存所用的最媒体处理器存贮量等。方框 610 运用该设定判断是否高速缓存样本，而且需要时作高速缓存。

方框 612 规定媒体处理器 220 接收应用程序 202 发出的数据“查找”。例如，可对下一帧数据请求数据查找，使视频流逐帧进动。收到查找后，媒体处理器对每一节点检查，在判断方框 614 中查看高速缓存器里有无请求的数据；若有，方框 616 变

把该数据送回应用程序 202 而不是再生样本数据；若无，则方框 618 利用有关的变换、源或其它有关元件生成样本数据。

有些视频译码支持用预译码的样本对译码器状态初始化的模式，此时能只高速缓存一小部分预译码样本而仍保持良好的清洗性能。例如，若高速缓存每第四个预译码样本而用户查找样本之一不在缓存器里的，则至多译码三个样本来生成所需的输出样本。

根据众多可应用本发明原理的实施例，应该明白，这里参照附图所描述的实施例仅作示例，不应视作限制本发明的范围。例如本领域的技术人员应明白，以软件示出的示例实施例的诸元件可用硬件实施，反之亦然，或者示例实施例可在结构和细节上作出修改而不违背本发明的精神。因此，本文描述的本发明试图让所有这类实施例都落在下列如权利要求及其等同物的范围内。

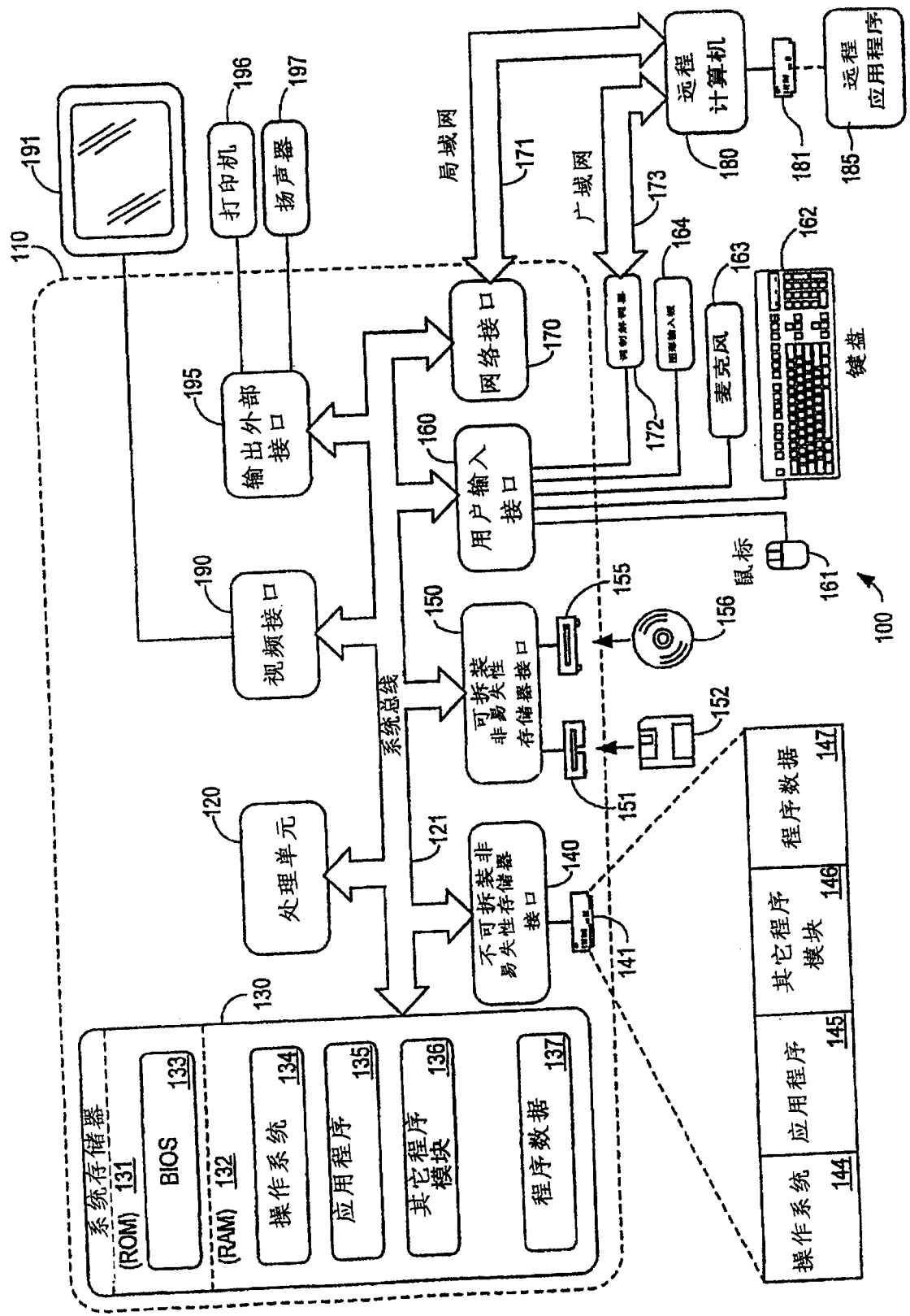


图 1

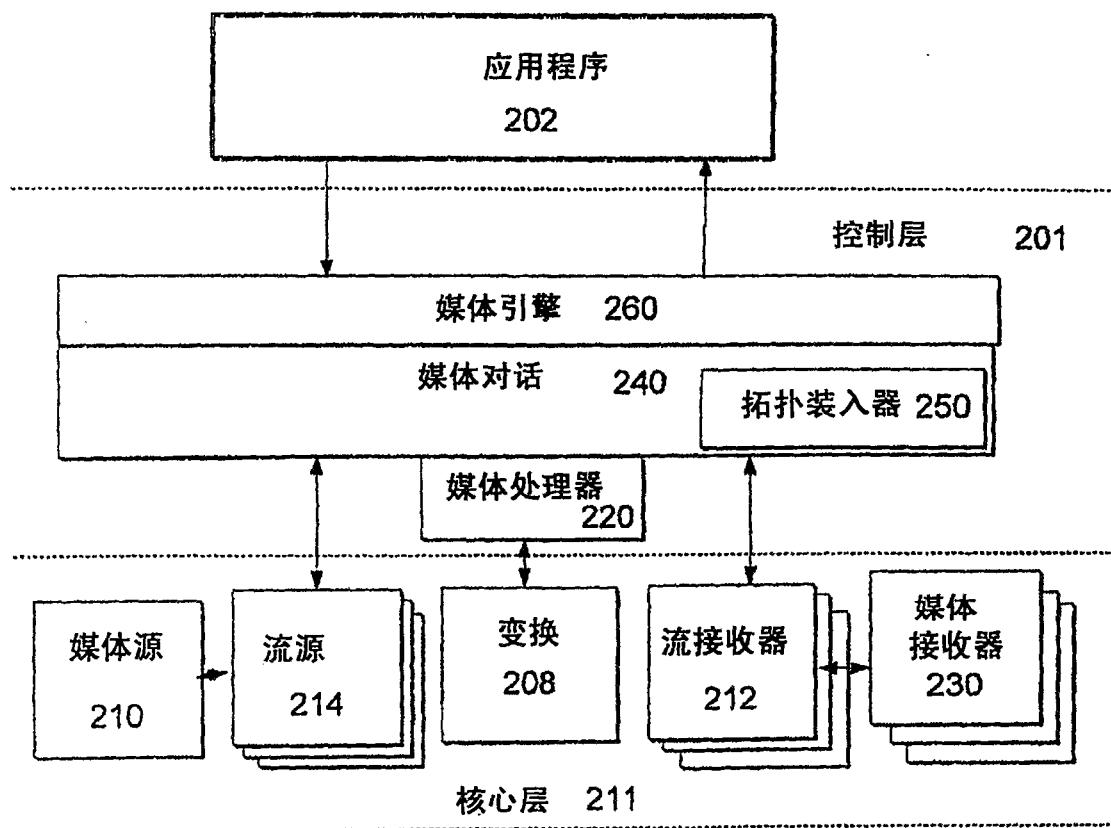


图 2

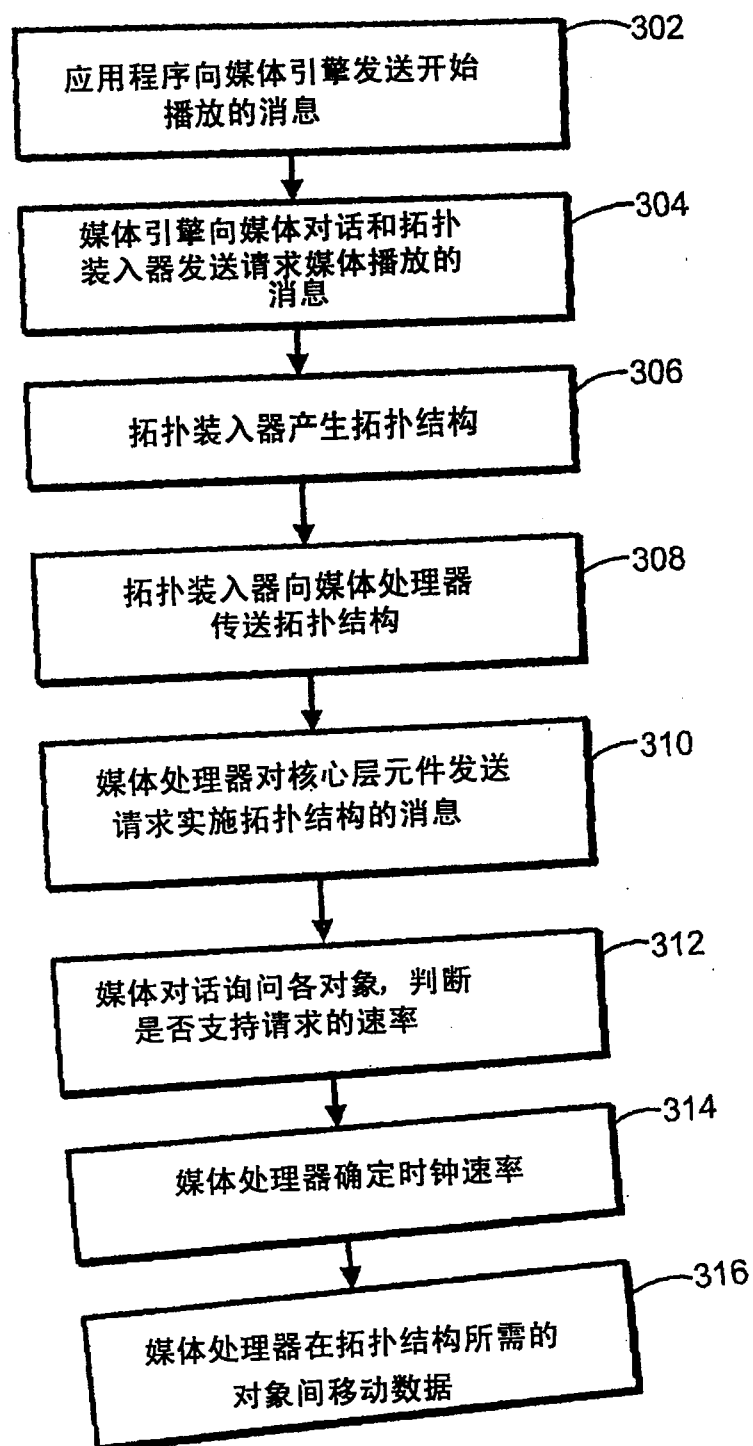


图 3

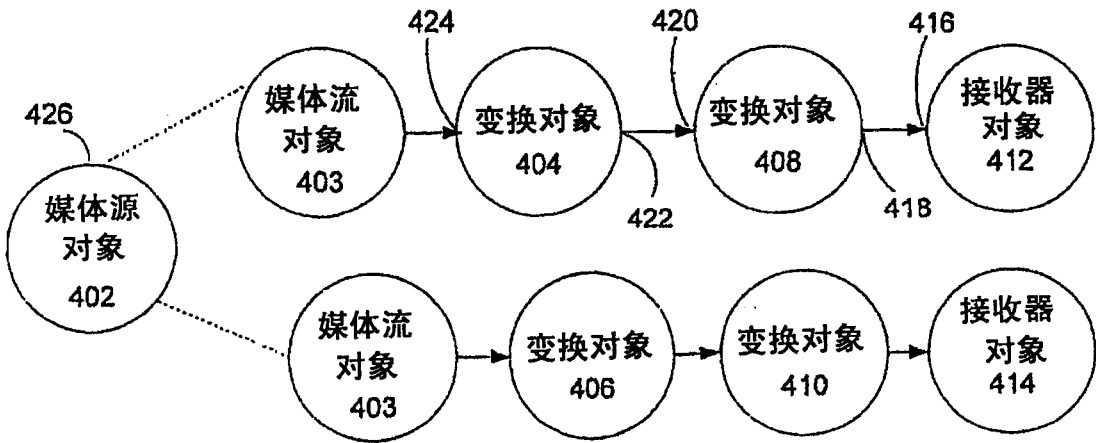


图 4

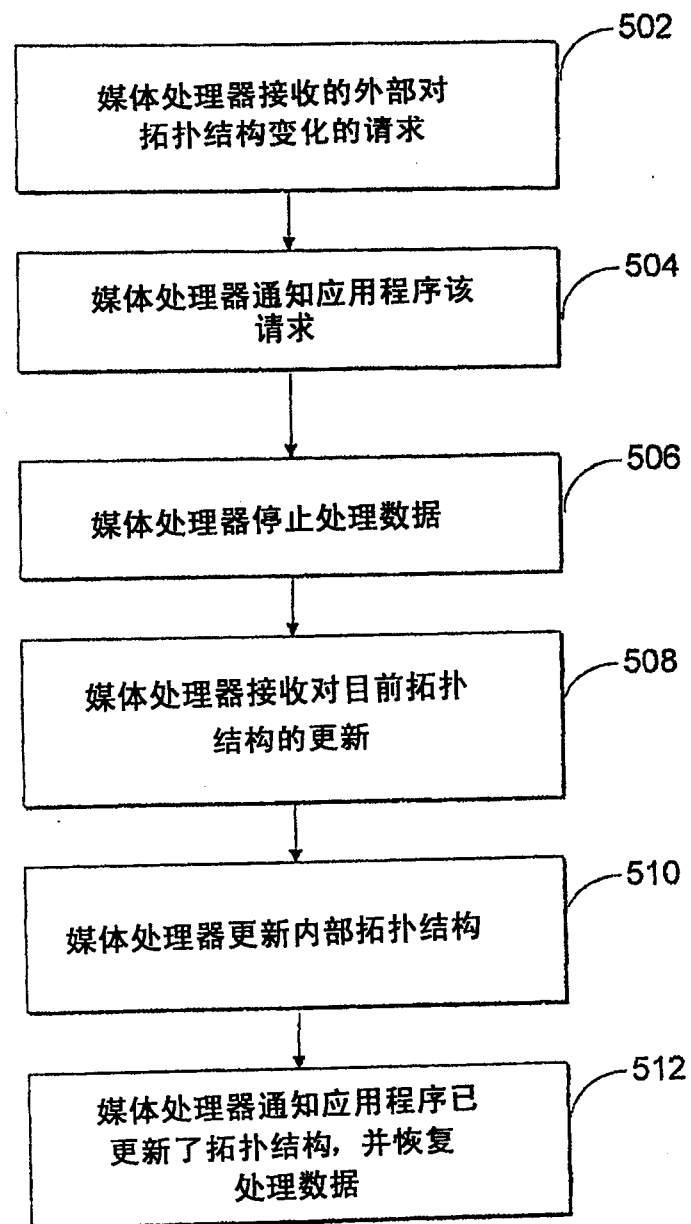


图 5

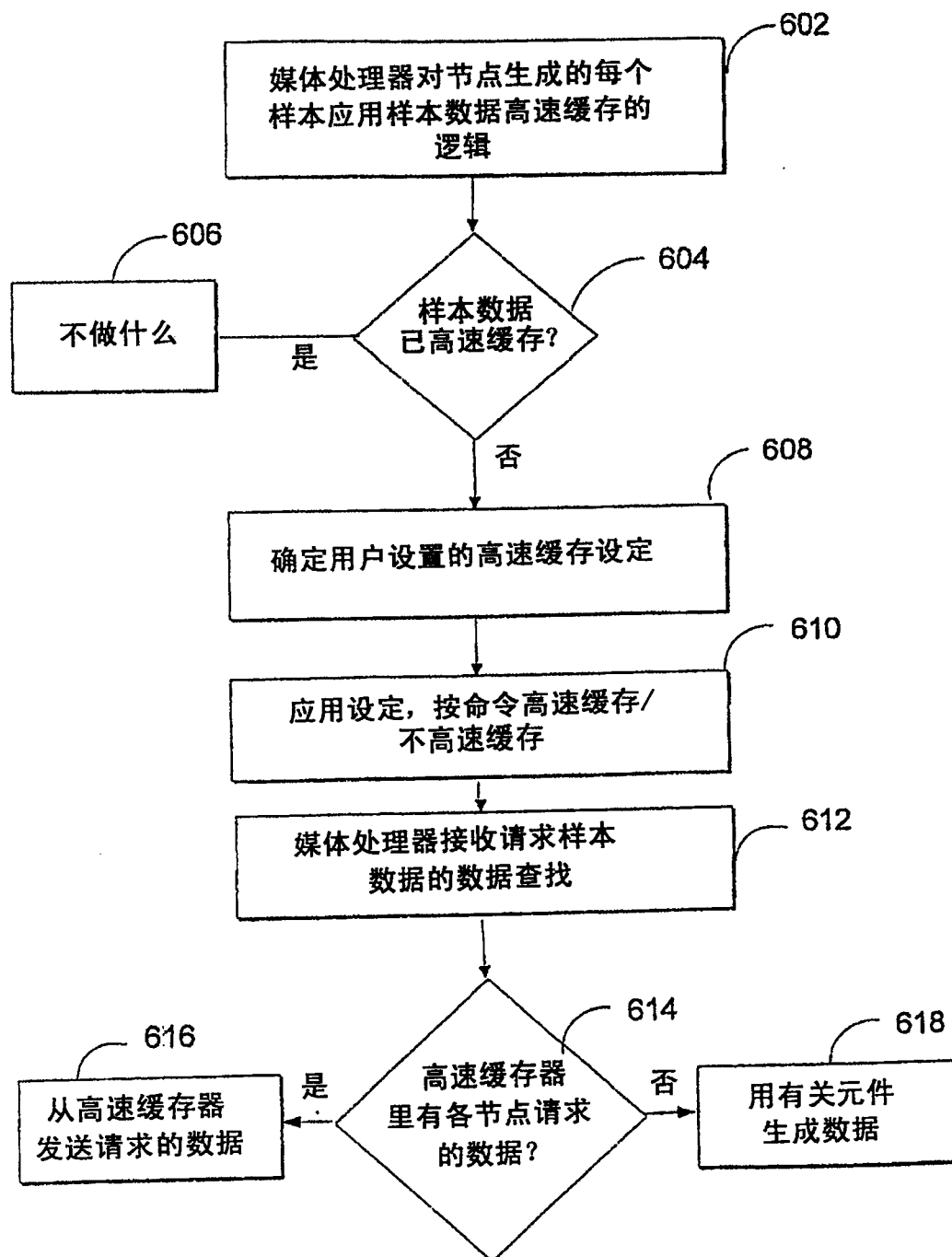


图 6