

多媒体对象的组织与结构化检索

李 宁

(北京信息工程学院软件工程研究中心, 北京 100101)

摘 要 多媒体的结构化检索具有广泛的应用前景,但多媒体的检索技术尚不成熟。多媒体数据模型的复杂性,连续媒体基于内容检索的低效,以及缺乏适用的查询语言,都使得多媒体检索困难重重。英国肯特大学新近研制的多媒体检索系统在多媒体检索的相关领域取得了一定的突破。这个系统把用户视图中的多媒体对象组织成具有层次结构的虚拟数据库,使用属性来标识数据库中多媒体对象索引的特征;查询代理机允许用户直观地构造查询过程——包括一个结构框架与嵌入式的查询语句。查询代理机以结构框架作导航,利用虚拟数据库的属性及调用外部查询过程来进行特定多媒体对象的检索。这个系统的研制成功是多媒体检索领域一个很有意义的尝试。

关键词 多媒体检索,虚拟数据库,查询代理机,结构化检索

1 简介

多媒体对象的检索依赖于多媒体对象的组织形式。多媒体对象组织的优劣一定程度上决定了其检索的效率。而将多媒体对象组织成多媒体数据库的目的正是为了将来对之进行检索。一方面,对于对象的组织,我们已有许多成熟的数据模型,如层次模型、关系模型、网状模型等等。另一方面,常见的多媒体对象其本身是构造型的复合对象,多媒体对象本身也可用多种数据模型来描述,典型的有:超文本模型(网状模型)、文献模型(层次模型)和信息元模型(层次模型)等等。这种多媒体对象的构造特点决定了多媒体的检索必须是基于结构的,即多媒体对象的检索不但要依据像存取路径这样的对象间的关系,而且要依据组成该对象的各子对象间的时空关系。例如“在传记类书目中寻找一本书,其第一章有一幅张三的照片,第二章的某一节有关于李四的介绍”,多媒体的检索机制应该可以处理这些关系。然而,以往的多媒体检索技术,如过滤式信息浏览查询、基于内容的查询,基于时延敏感性媒体的检索等等,大多只适用于特定的简单媒体,而描述多媒体的数据模型大多数并不提供基于各自模型的结构化检

索方法,这是多媒体基于结构的检索所面临的困难。

除了基于结构的检索之外,多媒体的检索还必须是基于内容的。现有的多媒体信息系统大多只能按文字描述对多媒体对象进行查询,亦即在多媒体对象之上加进文字标识(关键字或属性等),然后使用关键字等方式进行检索或查询^[1]。这种做法虽然容易实现,但有很大的局限性。首先,多媒体对象要增加附加标注,而这种标注一般无法反映多媒体的全貌;其次,基于文字描述的查询语言也难以表达用户的查询要求。因而在实际应用中,用户更需要的一种自动语义分析的功能,即多媒体信息系统能对图象、声音等连续媒体的内容进行分析,以达到较好的检索效果。基于内容的检索可以以 QBE(Query By Example)为代表,这一类基于媒体内容的检索直接依赖于图象理解、语音识别等的模式识别技术,先进行特征抽取再计算相似性^[2]。由于计算机模式识别技术尚未完全成熟,并不存在通用的高效的算法,这是多媒体基于内容检索的主要障碍。

另外,传统的数据库查询语言 SQL 无法适应多媒体的检索,尽管 ISO 对 SQL 作了多次扩充,特别是 SQL3 增加了面向对象的概念和功能,并对过程加以扩充^[3],但形式化地表达和实现用于多媒体检索的 SQL 仍是非常困难的。近年来,国外一些研究

机构在这方面做了许多工作,出现了一些侧重于对象管理的查询语言,如 HyQ、ITASCA、O2、POSTQUEL 等等。其中 HyQ^[4]处理构造型多媒体对象的功能较强,但是作为 HyTime^[5]标准的一部分,尚未见到具体的实现版本。总之,多媒体的检索还缺乏强有力的查询语言支持。

能否找到一种普遍适用的模型,既适合多媒体对象的组织,又符合多媒体对象的构造,并在此基础上建立一种高层的查询机制用来对多媒体及其各成分进行统一的检索,正是本文所要解决的问题。作者于 1993 年至 1995 年间在英国肯特大学(UKC)作为主要参加者进行了一项多媒体检索系统的研究。这个项目的成果是实现了虚拟数据库和查询代理机¹⁾。虚拟数据库是多媒体对象的结构化组织形式,而查询代理机用于构造一套面向过程的、带有嵌入式查询语句的查询描述,对虚拟数据库中的多媒体对象进行自动查询。我们以互联网络的 HTML 超文本对象为测试蓝本,取得了很好的效果。在这套系统的研制过程中,对上述的多媒体检索的主要难点各有所突破。

2 UKC 的多媒体检索系统

UKC 多媒体检索系统的虚拟数据库采用一种常见的树状层次结构来组织用户关心的多媒体对象,这些多媒体对象的集合组成一幅用户视图。这个树状结构是由用户来创建和维护的。虽然这个结构存贮在本地,但是它所包含的多媒体对象本身并不一定需要存放在本地,因为树形结构的叶节点一般只是一个索引指针,指向所表示的多媒体对象。只有在对这个多媒体对象进行处理时,才有可能从远地点取回该多媒体对象。这使得虚拟数据库不像一般数据库那样占用大量空间。每个多媒体对象的索引都带有一系列用户定义的属性。这些属性将在查询代理机中用来构造查询条件。虚拟数据库所表示的每个多媒体对象本身可能也带有结构(如超文本等),虚拟数据库的树状层次结构把各种多媒体结构统一在一起,成为一种抽象的数据模型(图 1)。

在这个抽象的数据模型之上,查询代理机代表用户对虚拟数据库含有的多媒体对象进行查询和浏

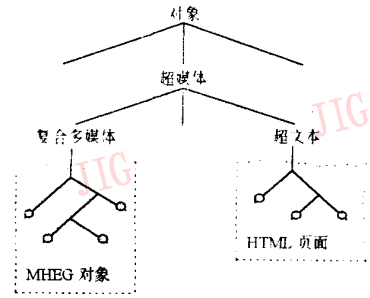


图 1 虚拟数据库中的对象模型

Fig. 1 Object model in the virtual database

览。用户按查询要求构造一种直观的查询过程描述,来告诉查询代理机如何找到、如何识别所需的多媒体对象。然后查询代理机开始自动查询,并将结果返回给用户。用户可以控制整个查询过程。查询亦可能在远地存放的多媒体对象中进行,查询的结果也可以加进虚拟数据库之中。查询代理机所使用的查询语言,不是普通的面向逻辑描述的 SQL,而是一种面向过程的查询描述。图 2 代表一个查询过程描述,用来查找“所有与多媒体相关的图书的封面和书名”。

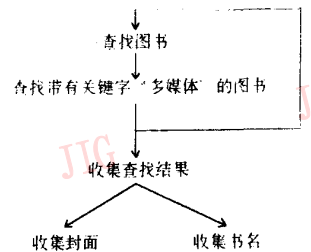


图 2 查询过程举例

Fig. 2 An example of query evaluation

在基于内容的检索上,该项目研制了一个 QBE 形式的图象匹配算法。这个算法的原理是让用户使用系统提供的绘画工具,画出一个草图,其中包括大概的色彩及其位置,不确定的部分使用灰色填充,然后把比较和被比较图象在频域上进行量化压缩(减到 8 种基本颜色),并在空间上进行规范化(缩放成 200×150 个像素),然后进行点对点比较,简单地解决了基于内容的图象检索。至于音频和视频的基于内容的检索尚待进一步实现。

下面分别介绍 UKC 多媒体检索系统的虚拟数据库和查询代理机。

2.1 虚拟数据库

虚拟数据库存储多媒体对象的索引信息。索引信息包括对象存放的地点及存取协议等,这些都是

1) Fred Cole and Ning Li et al, Virtual Databases: Private Views of Distributed Information, & Automated Browsing of Distributed Information, Submitted to Electronic Publishing, 1995.

对用户透明的。虚拟数据库的路径名用来标识每一多媒体对象(对象标识)。作为本地结构的扩充,索引还可以指向远程的多媒体信息库。虚拟数据库是层次形的,亦即其中的路径名可以是目录节点,也可以是多媒体对象的索引,随所处的位置而不同。UKC的多媒体检索系统提供了一套图形界面用于创建和维护虚拟数据库。提供的操作除了增加、移动、修改、删除多媒体索引以外,还提供了创建、修改属性的界面,特别是提供了两个自动机制(transducer),用来从多媒体对象中自动提取关键信息。比如,从文字类对象中提取摘要;从图象类对象中取得用于图象匹配的规范化检索图象等等。虚拟数据库的每一个多媒体索引都可附加一套属性,作为用户对该多媒体对象的标注,如名称、所有者、时间、版本号等等。另外属性也可以是多媒体类型,如一段关键的声音或影像等等,这些属性值都是用户指定的。用户借助属性在不取得对象数据的情况下也可了解多媒体对象的大致内容,更重要的是这些属性可在查询代理机中构造查询条件。

2.2 查询代理机

查询代理机可有多个,各自用来存放特定的查询过程描述。查询代理机也作为一个对象存放在虚拟数据库中,且具有相应的属性。这是因为查询代理机可以返回一系列查询结果,这些结果可以作为查询机节点的子节点而包括在虚拟数据库的层次目录中。

查询代理机提供给用户用于建立、修改和执行查询程序的图形界面。当建立查询过程描述时,用户首先指出所要寻找的对象的属性与类型,然后描述寻找的途径(查询网络),比如包含那几个步骤,每个步骤所要查询的中间结果以及步骤间的数据流向等等。查询从查询网络的某一个或某几个节点开始,遍历整个查询网络,逐个取得并检验查询所涉及的各个多媒体对象,选取符合条件的对象,废弃不合条件的结果,最后将结果储存在查询网络的终端节点中。查询网络允许有递归和循环,查询结果的获得往往要经过多个步骤并在中间步骤中循环多次。另外,因为网络速度和远程服务器繁忙程度的关系,远地对象的获取往往需要一定时间。因而查询代理机对于大容量虚拟数据库的查询可能运行很长的时间才能找到结果。查询代理机为此提供了丰富的控制功能,比如可以进行局部的查询,控制查询数目的上限,避免重复查询及保留中间结果等等。

2.3 查询过程描述

由查询过程描述而得到的查询程序可以分为两部分:查询框架与一组查询语句。查询框架为查询过程作导航,它由节点(查询节点)和节点间的连接(结果数据流向)组成,它决定查询源自何处,结果流向何方。而每一查询节点都含有如下信息:(1)所处理的媒体对象的种类;(2)一组初始对象;(3)嵌入的查询语句和查询过程;(4)查询过的对象;(5)未查询过的对象;(6)查询数目的上限;(7)查询状态。

一个节点的初始对象是该节点的所有前导节点的输出,初始对象又称输入对象。上述信息中的“查询过的对象”与“未查询过的对象”用于记录查询的中间状态,并避免重复查询。“查询数目的上限”用来防止返回过多的查询结果,比如,用户希望某个节点查找出5个结果足矣,查询代理机当找到5个结果后便停止这个节点的查询,从而提高检索效率。“查询状态”表示查询程序的执行状况,比如:“超时中止”、“完成”、“准备好”等等。

在查询节点中具体对象的查询是由嵌入的、与查询对象类型相匹配的查询语句来完成的。查询框架并不理解各个底层的查询,它从顶层节点开始,把各节点的查询语句连同初始对象标识甩给与节点查询类型相对应的外部查询过程,并将外部过程产生的结果(可能为空)分送下层节点,直至到达终端节点。查询框架是完全开放的,它允许嵌入各种查询语句,处理这些查询语句的外部过程也是用户定义的。查询代理机知道对于何种媒体对象调用何种外部查询过程。

例如,查询节点A专用于查询一个数据库内存贮的数据对象,则节点A内可嵌入该数据库所支持的SQL语句,而处理该嵌入查询条件的外部过程即为该数据库的查询引擎(SQL服务器),该查询引擎的指针可作为URL地址记载在节点A中(查询指定的数据库);该指针也可在运行时由上层节点产生并作为初始对象传给节点A(动态查询数个数据库)。当求值节点A时,查询代理机按指定的格式调用数据库查询引擎来处理本节点中嵌入的SQL条件,所得到的结果依据各自的类型分送下层节点。又如,查询节点B专用于文件系统中的关键字检索,则节点B中嵌入的查询条件为布尔运算符连接起来的关键字串,节点B的查询过程即可为一个全文检索命令。再如,节点C专用于查询文件系统中的某一类图象文件,C节点的查询过程可能是专为这

一类图象识别用途所写的模式匹配程序,C节点的嵌入查询条件即为该图象识别程序所需的参数,如QBE草图或颜色分布等等。值得一提的是,UKC的查询代理机已在内部实现了后两类节点的查询引擎,因而对于一般用途而言不必使用外部的查询过程。

查询网络有若干终端节点没有嵌入的查询语句,这些节点不受理查询,它们只是搜集上层节点的查询结果,查询的最终结果就分布在这些终端节点中。

由于查询代理机使用嵌入式的查询机制,它本身是抽象的和通用的。任何新的媒体出现在虚拟数据库中时,只需写好针对该媒体类型的外部查询过程,就可在虚拟数据库中对这种媒体加以检索。另外,只要是具有层次结构的复合多媒体对象,也可以映射到虚拟数据库的层次结构中,使用虚拟数据库本身的查询机制进行检索,而不必另写外部查询过程。

以上的查询过程按一定步骤找到用户所希望找到的对象,另有一类重要的查询过程用来判定某一复合多媒体对象是否符合用户指定的逻辑条件。这一类查询过程常作为查询网络的中间节点用来过滤中间步骤中的查询结果。这类查询过程与第一类查询过程相似,也是由节点和节点间的数据流向构成的查询网络,节点除了前述的内容以外,还带有判定条件。与第一类查询过程不同的是,它仅接收一个初

始对象,并且仅返回真假值。它还有两类新的节点——逻辑‘与’、‘或’节点。这类查询过程要解决的问题类似于“判断一本书,其中是否至少有两个章节配有彩色插图,或有两个章节配有黑白插图。”这个查询过程描述如图3所示。

这一类查询网络目前只限于带有局部简单循环的树形结构,它的解析过程是第一类查询自顶而下的求值与自底而上的真值回溯的结合。

3 结论

UKC的多媒体检索系统采用虚拟数据库的形式来组织多媒体对象,它构造了一个抽象的复合媒体的数据模型。查询代理机通过结构化的查询框架与嵌入式的查询语句,实现了多媒体的通用结构化检索。该系统曾应用在互联网络上,对HTML对象进行了成功的检索,取得了很好的效果,它可以弥补其它的资源发现机制如Web-crawlers与Fish search^[6]等的不足。这个系统还有许多可以进一步开发的地方,例如,完善与远程虚拟数据库的通信机制,使检索可以利用分布式环境并发进行;形式化表达查询网络,配之以语法检查,使用户可以准确地定义、描述查询过程;另外,检索过程尚可加以优化,例如通过记忆查询路径的快慢使查询代理机尽快地找到目标。总之,这个系统研制是多媒体检索领域一个很有意义的尝试。

参考文献

- 1 John F. Koegel Buford Eds. Multimedia Systems. Addison-Wesley Pub. Company, 1994;280.
- 2 Wayne Niblack Eds. In Storage and Retrieval for Image and Video Database. SPIE, 1993.
- 3 Kulkarni K G. Object-orientation and the SQL Standard. Computer Standards and Interfaces, Vol 15, Elsevier, 1993;287~300.
- 4 Kimber E W. HyTime and SGML: Understanding the HyTime HyQ Query Language. IBM, 1993.
- 5 ISO. Information Technology-Hypermedia/Time-based Structuring Language (HyTime), 1992.
- 6 Post R D J, Bra. P M E D. Information Retrieval in the World-Wide Web: Making Client-based Searching Feasible, Proceedings WWW Conf, 1994.

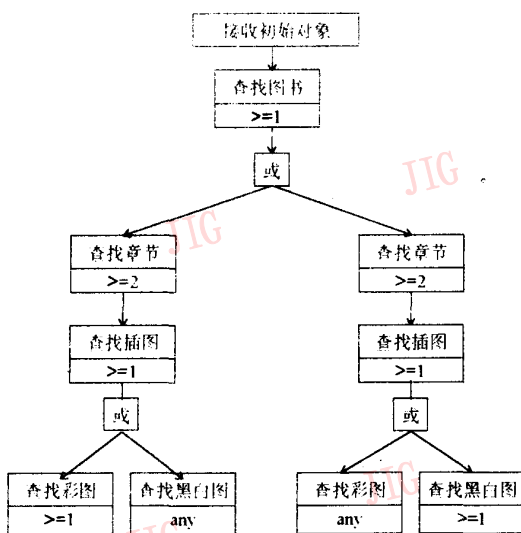


图3 返回真值的查询结构

Fig. 3 A query structure returning truth value



李宁,1985年毕业于北京信息工程学院计算机科学与工程系软件专业。毕业后留校任教,并在本院软件工程研究中心从事科研。1989年赴英国进修。曾担任英国肯特大学研究员,先后参加过多媒体分布式系统的研制和开发,以及多媒体检索技术的研究课题。并在此期间,开创性地进行了文字学专家系统的研究,提出了计算文字的理论,以此于1994年获计算机科学博士学位。1996年初回国,在北京信息工程学院任助理研究员,继续从事科研与教学工作。目前主要研究方向为超文本与软件工程文档支持环境,以及中文信息处理技术。

Structural Indexing of Organized Multimedia Objects

Li Ning

(*Software Engineering Research Center, Beijing Information Technology Institute, Beijing 100101*)

Abstract The development of information technology provides the accessibility to enormous numbers of multimedia objects, simple and complex. Indexing of multimedia objects has become a crucial issue. The variety of multimedia data model, the low efficiency of content-based searching, and the lack of proper query language all have made multimedia indexing a difficult subject. The multimedia indexing project recently carried by the University of Kent at Canterbury (UK) has made a breakthrough in the related areas. The virtual database developed in the project provides users a tool to create and maintain a private view of interesting multimedia objects. The virtual database is organized in a hierarchical structure. The multimedia object references it stored are associated with a set of attributes called 'cues' which are used in the browsing agent for constructing query conditions. The browsing agent performs the searching process on the user's behalf. Users first tell the agent how to find and recognize interesting objects, then set it going and control the query evaluation. The query specified by the user includes a framework, which plays the role of navigator in the evaluation process, and a collection of embedded queries. The browsing agent evaluates the query by checking the object cues stored in the virtual database and by calling individual external evaluators.

Keywords Multimedia indexing, Virtual database, Browsing agent, Structural indexing