

多媒体数据库 数据库 建模 存储 查询 多媒体

70-71.75

多媒体数据库:建模、存储和查询

TP311.13

TP391

朱扬勇

(复旦大学计算机科学系 上海 200433)

摘要 In this paper, the features of multimedia data are analysed firstly. Then the main problems of modeling and storing multimedia data are explored. Furthermore, the query of multimedia databases and its process are discussed.

关键词 Multimedia, Database, Query.

目前,已经有一些厂商,根据某些特定的需要和自己的理解,开发出了一些多媒体数据处理系统,如,KMS,OMEGA,QBISM等^[1,2,3].但是在理论研究方面可以说才开始.关于多媒体数据库管理系统(以下称MDBMS)的功能和特征还没有达成共识,多媒体数据的合理存储和有效访问还在探讨之中.

查询是数据库系统的最基本操作之一,查询处理的好坏是衡量一个DBMS性能的主要内容.查询涉及到数据存储,数据存储涉及数据模型.因此,弄清多媒体数据的特点,探讨多媒体数据的建模和存储以及分析多媒体数据库的查询,这些都是设计开发多媒体数据库系统的先导工作.本文分析了多媒体数据的特点,讨论了多媒体数据的建模、存储和查询处理以及存在的难题.本文的结论是:对多媒体数据建模是一项困难的工作,并且这将导致存储多媒体数据的困难;多媒体数据库的查询涉及“模式匹配”条件,所以查询还有赖于AI领域的新成果.并指出广泛的应用背景和强大系统实验将促使多媒体数据库及相关的技术迅速发展.

为了下面讨论方便,现在给出两个例子.例1展示了多媒体数据库中数据查询的一些特征;例2给出了当前一些自称为MDBMS的系统框架.

例1 考虑我们每个人的档案卷宗.从广义的角度来看,一个卷宗的数据包括:数值(如:年龄)、字符串(如:姓名)、文本(如:

简历)、图形(如:指纹、签名)、图象(如:照片)、声音(如:讲话录音)、视觉(如:活动录像)等几乎所有类型的多媒体数据.假设有一个DBMS来管理每个人的档案卷宗,那么经常的查询工作将是:

- ① 查询某个(具体)人的档案.如给定一个人的姓名,查阅简历、照片等.这是通常方式的查询.
- ② 查询具有某种特征的人.如:给定一张照片,查找符合的人.一般来讲,当前的照片和数据库中的照片不是同一个时期的,因此,照片只能相似不可能

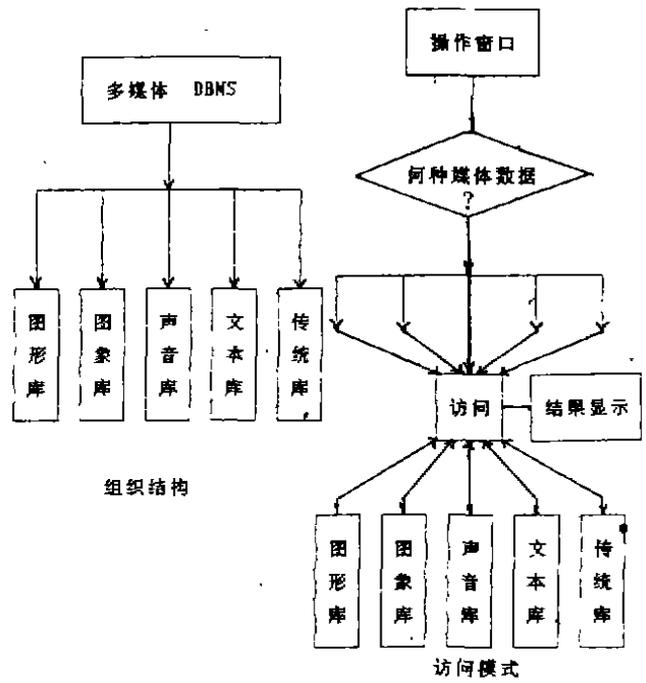


图1

朱扬勇 讲师,研究兴趣为数据库、知识库、多媒体.

相同。又如：查找脸上有刀疤的人，还有：查找在某张集体照上出现的人。这类查询条件模糊、过程递归，是难以处理的（要用到“模式识别”技术）。

③核对某人的特征。如：核对指纹、签名、声音。同样，这类查询也要用到“模式识别”技术。

例2 图1是一个常见的多媒体数据库的组织结构和访问数据的处理模式。

这类系统通常是关系数据库的扩充，用外部文件存放多媒体数据，在关系属性中存放指向这些外部文件的指针。在这种系统中，多媒体数据不但是分开存放的，事实上根本就没有存到数据库中。严格地讲，这只是个多媒体数据处理系统，而不是一个MDBMS。

1. 多媒体数据的特点、建模及存储

从结构上看多媒体数据可分为两大类：结构化数据（如：数值、字符串等）和非结构化数据（如：图象、声音等）。复杂、庞大、时序、冗余是多媒体数据的特点。复杂是指多媒体数据种类繁多并且大多是非结构化数据，多媒体数据来源于不同的媒体，具有完全不同的形式和格式；庞大是指多媒体数据容量大（如：2小时的电影画面需要GB级的存储容量）；时序（又称时态）是指多媒体数据的时间特性和版本概念，如某对象在一段时间内发生微小的变化或仅在其局部发生变化，最典型的是电影画面，一秒钟的电影胶片有24帧，这24帧影像不同之处却是极其微小的但具有时间上的先后；冗余是多媒体数据的一个重要特点，如前面的24帧电影画面，绝大部分都是冗余的。

多媒体数据来源于不同的媒体，具有完全不同的形式和格式（且不论声音和图象有多大的不同，就是电视录象也有多种制式，如：PAL、NTSC等）。因此，如何用一个数据库管理系统来有效地管理多媒体数据是一困难的问题。首先是多媒体数据的建模，即能否用一个合理的数据模型来表示多媒体数据；其次是如何在二级存储器中存放多媒体数据。

在数据模型方面，面向对象和超文本技术比较适用于媒体数据的建模。已经有一些系统用面向对象模型来表示多媒体数据^[2,4]，也有一些系统用超文本（超媒体）技术^[1]来表示多媒体数据^[1]。这些都是很好的探索。OODB以其丰富的建模能力而备受欢迎，但我们注意到，OO技术的强有力的数据抽象机制在多媒体环境下可能会削弱。回顾一下例1，一个人的档案卷宗可能包括了各种多媒体数据，并且，任

何两个对象（人）的信息量可能都不同（如：照片数目不同、有或无声音资料等等），即在多媒体空间中，任何两个人都可能不属于同一个类（OO观点）。另外，超文本技术是语义网，其结点是文本。如果将其结点扩充为可存放多媒体数据，则就是超媒体。这两种模型是通用的，仍在探讨当中。另外，对某种专门媒体数据的建模方面有较好的研究。典型的是影视数据的建模^[6,7]。而文[8]讨论了图象和文本的混合处理，主要用于报纸、期刊的处理，取得了较好的效果。从目前情况来看，不同媒体数据用不同模型建模能较快取得成功，但一个MDBMS支持多个数据模型不容易实现。

在数据存放方面，数据压缩和存储模式是两个重要内容，尤其是数据压缩。对于单个多媒体数据的压缩（如：一张照片的压缩存储）已经有很多优秀的算法了；然而，多媒体数据各版本的压缩存储还比较困难。如一秒钟的电影胶片有24帧，用一种版本技术来压缩这种多媒体数据是很有意义的。存储模式有二种：

①把多媒体数据的自然分割（一个文本、一张照片、一幅图、一部电影）作为不可再分割的基本单位存储在各自的库中（常常是外部文件），而在传统的数据库（如关系数据库）中只存指针及一些可结构化的特征。如例2的形式。这只要对传统数据库作简单扩充即可，实现比较容易，当然，性能和功能也有限。事实上这不是多媒体数据库系统。

②将多媒体数据包括特征信息（如：图象的色彩、灰度；声音的频率等）放入数据库，这种模式以数据模型为基础，对多媒体数据按提供的模型存储（如将一部电影分解成场景、帧等单元后再存储）。这是真正的MDBMS，提供例1所述的多媒体数据查询能力，并实现了许多查询优化方法。具体的数据模型还有待探讨。在实现方面，除了大容量主存、可高速访问的外存、数据压缩技术外，存在着DB（建模）和AI（模式匹配）两方面的技术难题。

最后要指出的就是：在目前的绝大部分多媒体系统中，多媒体数据是存放在外部文件中的，而在数据库中只存放指向该文件的指针。这样，数据操纵可以直接对文件进行，而不必通过DBMS。这就是说，多媒体数据还没有放入数据库中，即还不能称为多媒体数据库管理系统，只能称多媒体数据处理系统。从发展的观点来看，将来肯定会有多媒体数据库管理系统^[9]。

（下转第75页）

是一个具有零值的无限字节流,当/dev/zero被映射后,它建立一个未命名的对象返回给内存中的映射区域。

6. 源代码调试。P++的程序可以直接在系统提供的 debugger 中进行源代码调试,这主要是通过宏 `±line` 来实现的。

四、结论

ODBMS 支持 C++ 完全的类型系统,其核心是对象中各种指针的语义的维护,指针的语义的维护主要是通过 `swizzling` 和 `unswizzling` 操作完成的。充分利用操作系统的虚拟存储管理,可以极大地提高系统的效率,在相当的程度上减少了内外存数据对象之间的差异。

参考文献

- [1] Product Profile, Versant Object Technology Co., 1990
 [2] ONTOS DB 2.2 First Time User's Guide, ON-

TOS, INC. Feb. 1992

- [3] R. Agrawal et al., Ode (Object Database and Environment); The Language and the Data Model, Proc. ACM SIGMOD 1989 Int'l Conf. Management of Data, Portland, Oregon, May-June 1989
 [4] O. Deux et al., The O2 System, CACM Vol. 34, No. 10, 1991
 [5] C. Lamb, et al., The ObjectStore Database System, Same to [4]
 [6] Dunren Che, Zhongfan Mai, The Overall Design of MIDS/BUAA: a Multimedia Intelligent Database System, ICYCS'93, July 1993
 [7] 车敦仁, 麦中凡, MIDS/BUAA 语言的设计与实现, 第十一届全国数据库学术会议论文集, 1993. 9, 西安
 [8] UNIX 系统 V 第4版程序员指南: 系统服务和应用软件打包工具, 电子工业出版社, 1992年11月

(上接第71页)

2. 多媒体数据库的查询

查询处理是一个 DBMS 必备的特征,因此,多媒体 DBMS 同样也要能够处理查询,但是,超文本中的浏览不是严格意义上的查询,一个 DBMS 的查询包括查询条件、访问路径(显式或隐式)、结果模式,因而,处理一个查询须有条件识别、数据获取(含优化)、输出结果三个步骤。

下面,我们着重讨论查询条件和条件识别。多媒体数据库的查询条件可分为五类,①结构化数据的简单逻辑条件,是算术、字符的逻辑表达式,是传统 DBMS 所处理的查询条件,如:职称="教授". AND. 工资<600. ②结构化数据的项匹配条件,用于 DATALOG 查询,如: `:-?ancestor(李明, X)`. ③结构化数据的合一条件查询,用于带函数的 DATALOG 查询,如: `:-?ancestor(name(李, -), X)`. ④非结构化数据的模式匹配条件,是由多媒体数据引起的,如例1中列举的查询. ⑤模糊条件,出现在高智能化的系统中,如:“找出所有的胖人”。

结构化数据的条件容易识别,最复杂的合一条件由 AI 领域中的 A* 算法处理,也容易实现。然而、

模式匹配条件目前还不能很好处理,即使是采用神经网络技术,模式识别的准确率也还不高, DBMS 处理模式匹配条件的查询,恐怕还有待于模式识别领域取得更大的进展,至于模糊条件识别,可望在模糊 DBMS 研究领域取得进展。

对于多媒体 DBMS,上述五种条件的查询都是可能的,并且模式匹配条件的查询将是大量的(如例1),除此之外,多媒体数据库查询还具有递归性,为减少存储数据的冗余和数据版本的管理,一个多媒体数据对象往往分解成若干较小的对象,而小对象又可分成更小的对象,……这样,当查询一个对象时,可能会递归地引起许多对象的查询^[10]。递归查询处理,我们已经有了 DATALOG 的经验,但我们也知道,对于带函数的 DATALOG 程序,不动点的收敛性都不能保证,何况是多媒体数据对象,多媒体数据库的递归查询的收敛性是难以处理的,另外,文[11]中也证明了递归查询的固有低效率,这对于多媒体数据库的递归查询效率就更低了,虽然随着 GB 级的主存的使用,当前的许多 DB 可以全部装入主存,大大提高一般递归查询的效率,但是将多媒体数据库全部装入主存还需较长时间的努力。(参考文献共 11 篇略)