

# 多媒体数据库的时态索引技术

肖伟器 冯玉才 王元珍

(华中理工大学计算机科学与工程系, 数据库与多媒体技术研究所, 武汉 430074)

**摘要** 针对多媒体数据库中具有时态特性的媒体对象, 提出一种基于区间的索引机制, 目的是为了高时间检索的性能。这种时态索引可以快速查找任意时间段中的时态对象, 它能有效地支持多种时态关系操作符; 支持多种时态集函数; 支持时态 JOIN 操作等等。该索引支持媒体展示过程。

**关键词** 多媒体数据库, 时态索引, 时态检索

## 1 引言

多媒体应用系统中涉及到大量的具有时态特性的媒体对象, 如动态图象、动画、声音等。这些媒体往往以对象的形式存储在庞大的数据库中。在实际应用中, 针对数据库的时态检索是大量的<sup>[1,2]</sup>。因此, 如何从多媒体数据库中快速存取满足一定时态条件<sup>[3]</sup>的对象就形成了一个重要的研究课题。

为了提高时态检索的速度, 有效的途径就是研制高效的适合于时态特性的索引机制。传统的 B<sup>+</sup> 树索引结构<sup>[4]</sup>是针对单值数据的(线性的), 而时间是分区段的, 是非单值的, 因而 B<sup>+</sup> 树不能直接用于时间索引。本文提出基于区间分解的时态索引, 利用 B<sup>+</sup> 树的高效线性索引技术并将其加以扩充, 结合特定对象的时间特性进行时间分段, 从而大大提高了系统的时态检索效率。

## 2 时态数据模型

### 2.1 时间定义域

时态对象的时间特性可以用一个水平的时间轴上的区间来反映。设  $[0, T_{\max}]$  为一个多媒体应用的时间全域(GTD), 其中  $T_{\max}$  表示最大时刻。水

平时间轴上单位刻度所表示的时间由应用领域的度量粒度决定。“ $\leq$ ”是  $[0, T_{\max}]$  上的线性有序关系, 表示时间上“不后于”。

### 2.2 对象的时态区间

任何时态对象都具有一个连续的时间段, 称为时态区间。不妨设时态对象  $e$  的区间为  $[t_1, t_2]$ , 记为  $Time(e)$ , 其中  $t_1$  为起始时刻,  $t_2$  为终止时刻。 $t_2$  与  $t_1$  之差为时间跨度。

### 2.3 时态对象的结构

一个时态对象由常规属性和时态信息以及扩展属性三部分组成, 即:

时态对象元组 = 常规属性 + 时态信息 + 扩展属性

## 3 区间时态索引结构

区间时态索引机制基于对象记录存储系统 ODB, ODB 由对象的集合组成:  $ODB = \{e_1, e_2, \dots, e_n\}$ , 其中  $e_i$  为时态对象。图 1 描述了一组时态对象, 其中水平轴表示时间, 各线段表示相应的对象的时态区间。图中的虚线表示一个时态查询区间, 显然其结果应为  $\{e_3, e_5\}$ 。可以在时间轴上建立索引点, 索引点按如下方法建立: ①所有对象的时态区间起点



图1 时态对象在时间轴上的分布  
Fig. 1 Distribution of temporal objects

为索引点;②所有对象的时态区间的终点的后一个点为索引点;③ $T_{max}+1$ 为索引点。

图2是一个简例,其中 $T_{max}=13$ , $e_j$ 表示多媒体时态对象,它对应的线段表示时态区间的跨度,索引点在时间轴上用大圆点作了标记, $IP=\{0,2,4,6,8,10,11,12,14\}$ 。

为减少冗余,可采用增量累积方法,即在每个 $B^+$ 树叶结点块中的第一个桶中存放全部对象,而块内其它相继各桶中的对象要依据相对于前一桶的增量变化情况来看表示,限于篇幅,不再赘述。

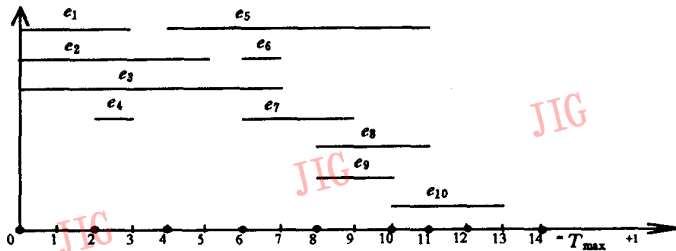


图2 在时间轴上的区间时态索引结构  
Fig. 2 The temporal index structure based on time intervals

## 4 索引结构的维护

### 4.1 时态区间检索算法

假设检索的时态区间为 $R=[t_a, t_b]$ ,满足条件的对象标识符集合为 $R\_SET$ 。(1)置集合 $R\_SET$ 初值为空;(2)依据 $t_a, t_b$ 搜索扩展的 $B^+$ 树,在叶结点上收集索引点;(3)将收集到的每一个索引结点相对应的存储桶中的对象标识插入集合 $R\_SET$ ;(4)集合 $R\_SET$ 中的元素,就是检索到的结果。

### 4.2 时态区间索引插入算法 $INS(e_k)$

- (1)  $t_a \leftarrow Time(e_k).t_1; t_b \leftarrow Time(e_k).t_2;$
- (2) 在 $B^+$ 树中搜索 $t_a$ ;  
若没找到 $t_a$ ,则将 $t_a$ 插入 $B^+$ 树中;

- (3) 在 $B^+$ 树中搜索 $t_b$ ;

若没找到 $t_b$ ,则将 $t_b$ 插入 $B^+$ 树中;

- (4) 将对象 $e_k$ 插入 $t_a$ 至 $t_b$ 所涉及到的存储桶内;

- (5) 若 $t_a, t_b$ 原本为索引点,则将索引计数器增1;若为新增的,则置1。除以上两种算法外,还有时态区间索引删除算法 $DEL(e_k)$ 。

## 5 时态操作

时态索引机制对时态操作给予了有力的支持,主要从四个方面讨论。

### (1)基本预定义函数

设 $V$ 是一个时态区间,则函数: $first(V)$ 返回 $V$ 的起始时刻; $last(V)$ 返回 $V$ 的终止时刻; $now()$ 返

回系统当前时刻;min\_time()返回系统所支持的最小时刻,可从索引结构的最左叶结点所对应的最左桶内,求最小起始时刻;max\_time()返回系统所支持的最大时刻,可从索引结构的最右叶结点所对应的最右桶内,求最大终止时刻。

### (2) 时态关系运算

<时态关系符> ::= before | after | meets | met by | overlaps | overlapped by | finishes | finished by | clustering | contains | starts | started by | equals

设  $V_1$  和  $V_2$  是两个时态区间,典型的时态关系符 before, after, during 的语义<sup>[3]</sup>如下:

$V_1$  before  $V_2$  iff last( $V_1$ ) < first( $V_2$ )

$V_1$  after  $V_2$  iff last( $V_2$ ) < first( $V_1$ )

$V_1$  during  $V_2$  iff first( $V_1$ ) > first( $V_2$ ) 且 last( $V_1$ ) < last( $V_2$ )

时态关系常用作检索条件,如:检索在时间  $t_1$  与  $t_2$  之间演播的音乐段:

```
SELECT object.name
```

```
FROM Music
```

```
WHERE object.time during [ $t_1, t_2$ ];
```

利用时态索引可以大大地缩小检索范围。

### (3) 时态集函数

在常规的数据库中,集函数如 COUNT、EXISTS、SUM、AVERAGE、MIN、MAX 主要是针对数据元组的集合或元组集合中的属性值。而在时态数据库中,集函数要适应于时态区间的特点。例如如图 2 中,不妨设  $e_i$  为乐器音响对象,试找出时态区域  $[4, 8]$  之间的乐器音响数目: Get COUNT Music IN  $[4, 8]$ , 这就有必要分小区间给出一系列结果,依据图 2, 其结果为 { 区间  $[4, 5]$  → 3 个, 区间  $[6, 7]$  → 4 个, 区间  $[7, 8]$  → 4 个 }。利用这种时态索引,很容易处理这种集函数,只需分桶统计即可,大大地缩小了搜索范围,从而减少了读入数据对象的个数。类似地,其它集函数的求值运算也是高效的。

### (4) 处理时态连接操作

这种时态索引也可用于提高某些时态连接操作的效率。许多对象可能被垂直划分在不同的关系中,例如在一个交响乐创作数据库应用中,大提琴与小号的数据分别存放在不同的关系中,它们可以通过时态区域作连接: JOIN 大提琴 AND 小号 WITH  $[t_1, t_2]$ 。这种操作可以利用索引分别求对应时态区域中的大提琴与小号,使得两个结果集合得到筛选,这就大大地优化了连接运算的速度。

## 6 时态媒体展示

该索引机制能高效支持时态媒体展示。展示过程有四种操作:(1)媒体展示起点定位;(2)自然展示;(3)快进;(4)快倒。利用索引机制,可以高效而自然地实现这四种操作的算法。其处理思想为:

### (1) 展示起点定位算法

依据输入的起始展示时刻值,检索  $B^+$  树,找出对应的存储桶,将桶中各对象读出并对准各对象的起始数据,准备展示。

### (2) 自然展示算法

确定展示起点之后,将桶内相关对象的数据送展示器展示,同时预读叶结点中下一桶内的对象数据并送入缓冲区。当当前桶内对象数据展示完时,就开始展示下一桶中的对象,重复前一步,查到最右桶内的所有对象数据展示完为止。

### (3) 快进算法

依媒体展示的当前时刻,暂停当前桶内媒体对象的展示。递增当前时刻,于当前时刻在叶结点上向右越过一桶时,放弃前桶中的媒体数据,并在新进入的桶中递增当前时刻,当停止快进过程时,依据当前时刻对准起点,读入当前桶内相关数据,开始展示。

### (4) 快倒算法

该算法与快进算法类似只是移动的方向相反,这受益于  $B^+$  树叶结点间的双向连接。

## 7 总结

本文提出一种时态索引机制。该索引机制可以支持多媒体时态对象的快速查找。它能有效地支持多种时态关系操作符,支持多种时态集函数,支持时态连接操作等等。该索引还支持媒体展示过程。关于这种时态索引机制的性能分析及基于其上的查询优化技术,我们将另文讨论。

### 参考文献

- 1 Elmasri R, Wu G T J, Kim Y J. The Time Index: An Access Structure for temporal data. In 16th VLDB, 1990. 8: 1~12.
- 2 Lu H, Ooi B C, Tan K L. On Spatially Partitioned Temporal Join, In 20th VLDB, 1994. 8: 546~557.
- 3 张毅, 张师超, 严小卫. 一个时态数据模型上的查询语言. 数据库研究与进展 '95. 哈尔滨: 哈尔滨工业大学出版社, 1995. 12.
- 4 冯玉才. 数据库系统基础. 第 2 版. 武汉: 华中理工大学出版社, 1993.



肖伟器,1991年于华中理工大学获计算机软件专业硕士学位。现任华中理工大学计算机系数据库与多媒体技术研究所副教授。主要从事分布式数据库管理系统、多媒体技术和地理信息系统研究,已有四项成果分获国家级和部委级科技进步奖。

## Temporal Index Mechanism for Multimedia Database

Xiao Weiqi, Feng Yucai, Wang Yuanzhen

(Department of Computer Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** In order to efficiently manage temporal objects in multimedia database, especially to improve the performance of time retrieval, this paper presents an index mechanism based on time intervals. This kind of temporal index can support quick search of temporal objects in an arbitrary time interval, support many kinds of temporal relation operators, and support many time aggregation functions and temporal JOIN operations.

**Keywords** Multimedia database, Temporal index, Temporal retrieval

新书推荐

### 《OpenGL 编程指南》

孙绍麟 费月娥 编译

全书共十二章,三个附录,16开,300页。全面、深入、详细地讨论了 OpenGL 编程中的实际问题,对 OpenGL 编程人员极具参考价值。

内容包括:绪论,绘几何对象,坐标变换,显示表,颜色,光照,混合、反走样和雾,绘象素、位图、字体和图象,纹理映射,帧缓存,鉴别器和 NURBS,选择和反馈,附录 1,2,3。定价:48元

邮购:《中国图象图形学报》读者服务部(100088,北京海淀区花园路6号,  
电话:62378784,联系人:李如珍)