

# 基于多媒体图像数据的查询检索<sup>\*</sup>

彭 敏 何炎祥 刘茂福

(武汉大学计算机学院/软件工程国家重点实验室 武汉430072)

**摘 要** 本文对多媒体图像数据提出一套较灵活的存储、索引查询方式。针对多媒体数据库信息查询通用性的问题,构建了一个较为集成的特征库。针对高维数据查询效率问题,将基于内容、颜色、距离和形状的匹配算法与 FNN 算法及查找聚类重心的加速检索算法集成于查询检索引擎 IRE,使查询更高效、应用更广泛。

**关键词** 内容特征,空间特征,聚类,维度,R 树

## Retrieve for Multimedia-Based Image Data

PENG Min HE Yan-Xiang LIU Mao-Fu

(School of Computer/State Key Lab for Software Engineering, Wuhan University, Wuhan 430072)

**Abstract** In this paper, a series of storage and retrieval strategies are put forward for multimedia image data. To the problem of general capability of information retrieval for multimedia database, a more integrated feature-base is constructed; to the problem of retrieval efficiency of high-dimensional data, some algorithms, such as the match algorithms based on content, color, distance and shape, are integrated with the algorithm FNN and the algorithm of quickening retrieval by finding cluster centers, all these are combined into IRE and make the application of our query method more efficient and broad.

**Keywords** Content-based feature, Spatial-based feature, Cluster, Dimension, R-tree

多媒体数据库通常处理极大量的数据,包括图形、图像、文本、视频和音频数据等,数据类型复杂,多为半结构或非结构化数据。本文的研究对象是图像,图像既包含了属于内容特征的数据,如颜色,也有空间特征的数据,如距离、相对位置、空间关系等,这些数据多数是多维的。因此提供高效数据查询的索引结构非常重要。

目前许多研究机构对多媒体数据库的信息检索技术进行研究,如 M. Fjicker 等人开发的 QBIC 系统和 A. Gupta 等人开发的 VIM SYS 系统能够对图像的形状、纹理、颜色等内容进行查询。而对于基于空间的数据,其检索查询系统也有大量的研究成果并有力支持了 GIS 和 GPS 技术的发展。但是,以往提出的各种媒体数据的查询机制,都专门针对某种应用,不能兼顾图像数据的各种特征,使用范围较窄,并且对于高维度的多媒体数据,查询检索效率不高。

本文针对这类数据提出一套较灵活的存储、索引查询方式。

针对多媒体数据库信息查询通用性的问题,构建了一个较为集成的特征库。综合与图像媒体有关的空间特征、形状特征、颜色特征、纹理特征等,详细设计了特征库中各种特征的计算方法和存储机制。

针对高维数据查询效率问题,在基于 R 树和数据立方体构建的高效集成索引结构的基础上,将基于内容、颜色、距离、形状和语义的匹配算法融入 FNN 算法,并通过查找聚类重心的加速检索算法提高检索效率。基于这些工作设计了一个查询检索引擎 IRE (Index And Retrieval Engine),有力支持了进一步的数据挖掘工作。

## 1 图像数据特征库

特征库的建立,直接关系到多媒体数据检索的效率和准确性,是进行多媒体数据挖掘的基本前提。本章主要针对图像数据建立特征库<sup>[4]</sup>。

### 1.1 空间特征

图像中出现多个对象时,对象间的距离、方向关系和拓扑关系构成了对象间的空间关系<sup>[3]</sup>。拓扑关系包括相离、相切、重合、相交、内含、内含于(与内含相对)、覆盖、被覆盖(与覆盖相对)。两个相离或相切的对象具有方向关系。其方向关系由它的拓扑关系确定。方向关系包括:左、右、上、下、前、后、东、南、西、北、西北、东北、西南、东南,而且前、后也可与其它方向关系组合,如:左前等。

对于空间关系的计算,首先构造对象的最小矩形框:以图像的左上角为原点,把图像放在 X-Y 坐标中,画一个与 X 轴平行的最小矩形框,称为该图像的 Mbb,它由该矩形的左上角(Xmin, Ymin)和右下角(Xmax, Ymax)定义。则两对象的距离可由两个对象的质心距离来计算,一个对象的质心定义为:

$$x_{avg} = \frac{\sum_{i=1}^n x_i}{n} \quad y_{avg} = \frac{\sum_{i=1}^n y_i}{n} \quad (1)$$

其中,  $n$  为组成对象的像素数。

两对象间的方向可由角度数或8个方向测量。

两对象间的拓扑关系可以用 X-间段和 Y-间段来表示,且可以用空间算子表示: *before()*, *equal()*, *finish()*, *meet()*, *overlap()*, *during()*, *start()*等。

<sup>\*</sup>基金项目:国家自然科学基金重大研究计划(项目编号:90104005)。彭 敏 博士生,讲师,现从事分布并行处理和数据挖掘的研究。

## 1.2 形状特征

形状可分为基本形状,如圆、正方形、长方形等,以及任意形状(非基本形状)。一个几何形状是原子的,或是复合的。

原子形状可分为点、1维、2维三类。原子形状图像放置在以左上角为原点的  $X$ - $Y$  坐标中, $X$  轴从左往右增加, $Y$  轴从上往下增加,以像素为单位。各种点、线、圆、椭圆、长方形、三角形、多边形等都是原子形状。

一个复合形状包含多个原子形状。组成复合形状的这些原子形状的边界可以是相交或相离的。

**1.2.1 形状特征的计算** 在讨论形状特征的计算时<sup>[4]</sup>,以最常使用的二维形状为主要研究对象,讨论两种矩形覆盖:累加矩形覆盖是该矩形由几个矩形合并构成;通用矩形覆盖是对矩形的加和减。

通用矩形覆盖的好处是可以更细致地描述图像,缺点是描述过程更复杂。令  $C_i$  是经过  $i$  个矩形(预期)覆盖后的当前(局部)覆盖( $i=1,2,\dots,n$ )。 $C_0$  是空集(指像素或点)。对于一个累加矩形覆盖, $C_{j+1}=C_j \cup R_{j+1}$  ( $j=1,2,\dots,n$ )。 $R_j$  是第  $j$  个被加入的矩形。在通用矩形覆盖中,根据新的矩形是加入还是减去而选择  $C_{j+1}=C_j \cup R_{j+1}$  或  $C_{j+1}=C_j - R_{j+1}$ 。

设某形状被覆盖了  $S$  个累加矩形,对于每个有限条边的形状都存在一个正整数  $K$  使得  $C_k=S$ ,并且不需要有矩形加入  $C_k$ ,则可以定义  $C_j=C_k, j \geq k$ 。

对于一个给定的形状,累加矩形覆盖和通用矩形覆盖都不唯一。当一个矩形  $R$  加入累加矩形覆盖  $C_i$  时,必须保证  $R$  中不存在一个矩形  $R'$  ( $R' \subseteq R$ ) 使得  $R \subset R' \cup C_i$  (即不存在  $R' \cup C_i = R \cup C_i$ ),这保证了矩形不会大于必要大小。同理,在通用矩形覆盖中,一矩形  $R$  加入目前的通用矩形覆盖  $C_i$ ,必须是  $R$  中不存在矩形  $R'$  ( $R' \subseteq R$ ) 使得  $R' \subset R \cup S \cup C_i$ 。仅当  $R$  中不存在一个  $R'$  ( $R' \subseteq R$ ) 使得  $(R-R') \cap C_i = \emptyset$  时,这个  $R$  才可从  $C_i$  中减去。

**1.2.2 存储结构** 对每一个矩形定义一个左边最底部和右边顶部的角,分别称为  $L$  角和  $U$  角,每个角可由一对  $X, Y$  坐标表示,代表在数码照片或像素图像中的位置,这样就可以用一个  $4k$  坐标的集合(两个角被  $K$  次矩形,每个角有2个坐标值, $k$  为正整数)来表示  $K$  个矩形的集合。

为了便于读取,我们不直接存储这些坐标值,而是进行一些转化。

首先,并不直接存储每一个矩形的  $L$  角和  $U$  角的点,而是获得特有的位置和尺寸值,矩形的位置值由  $L$  和  $U$  角所在点给出,如,点  $(x_L+x_U/2, y_L+y_U/2)$ ,  $x_L$  是  $L$  角所在点的  $X$  坐标,其余同理。对应  $L$  和  $U$  拐角点的不同所获得的矩形的尺寸边不同,如  $(x_U-x_L, y_U-y_L)$ ,这样我们还是用四个数值(或两对数字)来存储每一个矩形,然而,经过这种转化后代表了矩形的位置和尺寸,而不是角点所在位置。

第二,第一个矩形的位置规范其它矩形的位置。即,第一个矩形的中心被初始化,其它相应取值,如同经过一个移动,每个矩形相应所有的  $X$  和  $Y$  坐标都进行相应的加减,但矩形的大小不受影响,因为移动后第一个矩形的中心坐标为0,0,不用存储矩形,而是存储它的移动量,即移动前中心点的坐标。

第三,第一个矩形的大小用来规范其它矩形的位置和大小。因此,第一个读写的  $X$  和  $Y$  参数用来划分其它矩形相应

的  $X, Y$ , 进而,取规范化尺寸值的对数,使它们可以象位置值一样进行加减(不需记载位置值)。规范化以后,第一个矩形的大小为(1,1)(它的对数是(0,0)),存储经过第一个规范化后这个矩形的初始化尺寸参数而不是存储这个(1,1)的值,作为第三次转化后的值。这个值是其它矩形  $X$  和  $Y$  坐标值的衡量标准。最后再附加一个更改,不是求两个度量元素本身各自代表的维度,而是将它们的作用值作为坐标值(这是一个线性值的平方,代表面积刻度),和一个比值( $Y$  刻度值被  $X$  刻度值除),称为“变形值”。

这样,由  $K$  个矩形描述的形状,就可以存储为一对  $X, Y$  的移动值,一个坐标值和一个变形值。这四个值代表了第一个矩形,中心点的一对  $X, Y$  坐标和剩下的  $K-1$  个矩形中每一个的一对  $X, Y$  尺寸值。

描述一个形状的矩形的个数  $K$  对一些形状来说可能非常大,如此多的维度来建立空间特征的复杂结构可能不太实际,但大多数形状信息都可由少数几个矩形组成。同时,在数据库检索中,首先要求它提供足够的区别以防止获取数据库中的一大部分,而不仅仅是为了实现精确匹配,由于这个原因,进行相似性匹配比精确匹配更好,所以检索一个很小的矩形数  $K$  就足够了。根据我们的经验,对于不同的复杂形状,即使在一些大型数据库中, $K$  值从2到5就足够。

当图形描述转化为一个  $4K$  维度空间的点的坐标集时,我们可使用所希望的任何多维点检索方法,如 grid-files,  $K$ -D-Btrees, buddy trees, holey-brick trees,  $Z$ -curves, ..., 唯一可能出现的问题是  $K$  值很大时,这些技术使用困难,但本文提出的综合检索机制可以针对多维检索。

## 1.3 颜色特征

由于图像的索引结构是建立在基于图像颜色相似度的基础上,因此必须从图像中选取有效的颜色特征及图像相似度算法来构造索引结构。目前选取颜色特征的方法可以分为两大类:

(1) 每个对象具有的颜色标识作为主码,从一个查看表中取出相应的颜色值。其特点是节省存储空间。

(2) 颜色的表示模式有 HSV 模式、RGB 模式等,直接存储每一对象颜色的  $H, S, V$  值。特点是执行速度快。

下面使用 HSV 模式的特征抽取方法来抽取颜色特征。

对象和图像都可与颜色相关,由于图像的颜色非常多,尤其对真彩色图像,因此直方图矢量的维度数很多。为了减少计算量,对 HSV 空间进行量化,将  $H, S, V$  3个分量按照人的颜色感知进行非等间隔的量化:

(1) 按照人的视觉分辨能力把色调  $H$  空间分成8份,饱和度和亮度  $V$  空间分别分成3份。

(2) 根据色彩的不同范围进行量化。

$$H = \begin{cases} 0 & \text{if } h \in [316, 20] \\ 1 & \text{if } h \in [21, 40] \\ 2 & \text{if } h \in [41, 75] \\ 3 & \text{if } h \in [76, 155] \\ 4 & \text{if } h \in [156, 190] \\ 5 & \text{if } h \in [191, 270] \\ 6 & \text{if } h \in [271, 295] \\ 7 & \text{if } h \in [296, 315] \end{cases} \quad S = \begin{cases} 0 & \text{if } s \in [0, 0.2] \\ 1 & \text{if } s \in [0.2, 0.7] \\ 2 & \text{if } s \in [0.7, 1] \end{cases} \quad (2)$$

$$V = \begin{cases} 0 & \text{if } v \in [0, 0.2] \\ 1 & \text{if } v \in [0.2, 0.7] \\ 2 & \text{if } v \in [0.7, 1] \end{cases}$$

(3) 构造一维特征矢量。按照以上的量化级,把3个颜色分

量合成为一维特征矢量:

$$I = HQ_S Q_V + SQ_V + V \quad (3)$$

其中,  $Q_S Q_V$  分别是分量  $S$  和  $V$  的量化级数, 这里令  $Q_S = 3$ ,  $Q_V = 3$ . 因此上式可表示为:

$$I = 9H + 3S + V \quad (4)$$

由此式,  $I$  的取值范围为  $[0, 1, \dots, 71]$ , 计算  $I$  即可获得 72bin 的一维直方图。

图像的颜色特征可以是 256 维、72 维、64 维等。维度的减少会影响查询的效果和精确度, 但可以大大提高查询速度, 所以颜色特征值的确定需要根据用户的查询需求, 必要时甚至可以减少到 3 维。

#### 1.4 纹理特征

纹理是图像中一个重要而又难以描述的特性。很多图像在局部区域内可能呈现了不规则性, 而在整体上表现出某种规则性, 习惯上把图像中这种局部不规则的, 而宏观有规律的特性称之为纹理, 它适合描述诸如山脉、水纹、树、砖瓦、纤维等图像。

一个对象的纹理可用不同的维定义<sup>[3]</sup>, 例如: 纹理按水平、垂直或一定角度方向测量的方向性、光滑度、粗糙度、对比度和周期性等。使用的维越多, 结果越精确, 计算时间也越长。

## 2 基于内容的相似性匹配

### 2.1 相似性函数

在本文的基于内容的相似性计算中, 每一个用户查询请求可概念化为一个 E-R 图。一个数据库中图像查询系统是由图像组成的 E-R 图的集成。目前使用 E-R 图来表达图像内容的相关数据靠人工构造形成, 但我们希望计算机视频和模式识别技术能帮助自动识别某些对象以及它们之间的关系。围绕在数据库中寻找 E-R 图的查询过程与对应用户查询的 E-R 图很相似。用户查询 E-R 图也需要确定一个数字  $u$  作为查询测试用的图像数, 与用户描述最接近的近似值被获取即为  $u$  幅图像。

对于一个查询  $Q$ , 我们在数据库中计算每一幅图  $P$  的相似值, 来反映与查询  $Q$  匹配, 这幅图有多接近。假设相似值为一函数  $f$ ,  $f$  由  $Q$  和  $P$  给出;  $f(Q, P)$  越大, 图像与查询要求越接近。为了决定  $P$  与  $Q$  有多匹配, 首先将  $Q$  中的对象与  $P$  中的对象进行匹配, 然后决定满足  $Q$  中的对象的哪个模糊关系与  $P$  中相应的对象匹配。

通常,  $P$  中可有不止一个对象可与  $Q$  中的对象匹配。如果  $Q$  中有多个对象, 那么将会有这类匹配的很多合成。因此, 我们定义  $f(Q, P) = \max_{\rho} \{g(Q, P, \rho)\}$ ,  $\max_{\rho}$  是所有  $\rho$  中最大的一个;  $\rho$  是一对一函数中任一, 即匹配, 假若可以将  $Q$  中每一个对象在  $P$  中指出,  $g$  是一个相似性函数, 用一个实数表示  $Q$  与  $P$  在给定匹配  $\rho$  之下对象匹配的近似程度。

$g(Q, P, \rho)$  值是三个实数值的综合, 第一个数纯粹基于查询  $Q$  中对象与  $P$  中对象的匹配; 这个数本身是  $Q$  中所有对象  $A$  的相似值的总和, 代表对象  $A$  与  $P$  中对象  $\rho(A)$  的相似程度 (如果  $\rho(A)$  无定义则  $A$  对象的相似度为 0); 第二个数值基于无空间关系匹配, 是  $Q$  中所有无空间关系  $r$  的相似度的总和, 代表在  $P$  和  $\rho$  下如何满足  $r$ , 第三个数值基于空间关系。

### 2.2 形状特征相似匹配

1. 形状特征匹配概述 数据库中对于不同的查询要求, 有不同的匹配方式, 主要有: 完全匹配、移动匹配、相同比例匹配、不同比例匹配等。这些方式所描述的形状轮廓即使不同, 也有相近的位置和尺寸。这种匹配方式的要求并不适合大多数查询要求。

本文主要采用相似匹配的方式。通过对每一个特征矢量规范一个范围, 来对每一个轮廓的位置和尺寸的确切值给予相应的灵活度。这种“模糊”的限度与具体的每个特征矢量无关, 而是由相似性参数决定。模糊性越大, 搜索的标准越低, 查询到的与指定形状相似的对象集合越大。在大多数应用中, 并不是在模糊允许的位置和尺寸范围内规范独立的参数, 而是定义若干球状参数。大型的位置和尺寸值将允许更多的边界错误。

形状的描述是有序而非任意的。通过控制描述形状的矩形框的数目可以选择一个形状相似于给定形状的程度。在查询中, 矩形框的数目是一个必须谨慎选择的参数。假设形状数据库中的索引结构是  $K$  个矩形框,  $K$  是矩形框数目参数的最高极限。为了使相似度更具有普遍性, 总是希望检索基于较少的矩形框, 但使用的矩形框越少, 查询后返回完全不相似形状的可能性越大。

2. 多样表示性 相似查询的一个问题是一些非常吻合或相近的形状却具有完全不相似的描述。解决这类问题的一种方式是采取基本相同的两个或多个顺序描述。还有一种更好的解决办法是获得给定查询形状的多种“好的”顺序描述方式, 然后用每一种方法实现一个查询, 获得一个结合的结果。这种方式会多花费一些查询时间, 但数据库和检索结构不需扩充, 而且需要尝试的不同顺序描述的数目对应于查询描述的长度, 这个长度一般小于查询形状或数据库对象顺序描述的长度。

通常, 一个对象的顺序描述的长度不需要与索引的维度数目相匹配。如果数据库中的一个对象有一个较长的顺序描述, 则通常情况下只有前面的一部分用于索引结构。

### 2.3 纹理特征匹配

维和距离函数的选择依据具体应用而定, 已知一组  $n$  维规范化的纹理特征值  $t_1$  到  $t_n$ , 最简单的距离函数是线性平均值, 其中  $D_i$  为数据库中的图像  $t_i$  和目标图像之间的绝对差 (式 5); 或者将距离函数描述为加权的线性和 (式 6),  $w_i$  为第  $i$  维的权, 且满足公式 (式 7):

$$sim = \frac{\sum_{i=1}^n D_i}{n} \quad (5)$$

$$sim = \sum_{i=1}^n W_i D_i \quad (6)$$

$$\sum_{i=1}^n W_i = 1 \quad (7)$$

也可用欧几里得距离:

$$sim = \sqrt{\sum_{i=1}^n W_i (D_i)^2} \quad (8)$$

维和距离函数的选择依据具体应用而定。

### 2.4 空间关系匹配

采用质心法计算空间关系, 相似度可以根据角度和距离来计算。角度  $\theta$  是目标图像和图像库中的图像相对于某一个固定点角度之差, 即二者相对于固定点的夹角。相似度描述

为:

$$Sim = \omega_d |d_{db} - d_{qr}| + \omega_\theta |\cos \theta| \quad (9)$$

其中  $\omega_d, \omega_\theta$  是权值, 且  $\omega_d + \omega_\theta = 1$ ,  $d_{db}$  为数据库中图像的对象间距离,  $d_{qr}$  为目标图像中对象间的距离。

## 2.5 颜色匹配

得到图像特征的统计直方图后, 不同图像间的特征匹配可借助计算直方图间的距离来进行。

如果两图像中含有不同的颜色, 那么, 或者这些不相同的颜色不是组成检索目标的颜色, 或者数据库图像中不含有检索目标。另外, 相同的目标对象在不同的图像中出现, 其组成目标对象的颜色应该是基本成比例的, 如果两图像归一化直方图中具有比例相差很大的颜色, 则必定为杂色, 或者说其中有一幅图像中的该种颜色的大部分为杂色。根据这两点事实, 建立两图像的相交颜色直方图。

设示例图像的主颜色表直方图为  $Q(r)$ , 数据库图像根据  $Q(r)$  所建立的颜色直方图为  $I(r)$ , 则建立相交颜色直方图算法为:

(1) 过滤掉  $I(r)$  中  $Q(r_i)$  与  $I(r_i)$  比例相差若干倍的颜色, 设为  $t$ , 其大小根据经验和具体检索要求而定, 即:

若  $\min(Q(r_i), I(r_i)) / \max(Q(r_i), I(r_i)) > 0$ , 则  $I(r_i) = 0$ , 其中,  $Q(r_i) \in Q(r), I(r_i) \in I(r)$ ,  $t$  取某一常量 (一般小于 1), 处理后的  $I(r_i)$  设为  $I'(r_i)$ 。

(2) 对  $Q(r)$  过滤掉没有在  $I'(r_i)$  中出现的颜色, 得到  $Q'(r)$ 。

(3) 对  $Q'(r)$  与  $I'(r)$  进行归一化, 并计算其相交距离, 得到相似度  $Sim$ 。如果两幅图像的颜色分布相同, 那么相似性测度为 1, 否则为 0 到 1 之间的一个数。最不相似为 0, 如黑和白图像的直方图交集。经过交叉算法后, 可以比较明显地区分出 not 相似的图像。

该算法可以区别比较明显的不相似图像, 但要检索匹配图像, 在该步骤之后, 还需要根据相似性函数, 使用本文的检索机制进行二次匹配。

## 3 快速查询检索

### 3.1 查询原理

多媒体图像数据多是高维数据, 例如由 HSV 空间得来的颜色特征用 72bin 来表示, 纹理特征涉及光滑度。周期性等多种维度才可以描述精确。如此高维的特征向量用我们基于  $R$  树的最近邻快速查询 FNN 算法<sup>[3]</sup>实现还是效率较低, 有一定的困难, 需要进行降维处理之后, 再引用 FNN 算法, 这里我们将聚类方法引入多维数据查询中。

目前使用聚类思想进行的降维方式主要有两类: (1) 主要通过聚类的方式加速检索的过程, 将图像库中的图像分成  $K$  类。具体每个类中图像的个数和图像的颜色特征分布有关。但此方法需要事先规定聚类个数。(2) 根据数据间相似程度的定义, 以一维数据的聚类方法为基础, 通过对各映射子空间聚类相交, 得到高维聚类的近似解。

使用第一种降维思路与 FNN 算法相结合, 可以比较高效准确地进行基于颜色特征的索引查询。查询开始前, 将图像聚成  $R$  类, 每个图像聚类选择一个代表样本。查询开始时, 将示例图像分别与图像库中的聚类代表样本 (又称为聚类重心) 按上述的算法进行匹配, 找到与示例图像距离最近的聚类重

心, 即与示例图像最相似的聚类重心。然后在聚类中, 将示例图像与类中每一幅图像使用 FNN 算法进行匹配, 根据用户的要求, 返回结果。

### 3.2 聚类内进行特征快速查询

根据上述聚类算法得到的线性索引链表, 使用基于  $R$  树结构设计的最近邻快速查询算法 FNN, 完成颜色特征查询。FNN 算法性能的提高主要体现在检索时对维度的限制更小, 同时查询对象和数据对象的范围可以扩展到更宽的领域, 包括区域颜色、空间位置等基于内容和空间的数据类型。

输入 FNN 的是一个查询节点  $q$  和包含一个数据对象集合的  $R$  树。查询对象和数据对象的范围可扩展到任意维度和数据库中属于内容特征与空间特征的任何类型, 包括区域颜色、纹理、尺寸、宽、高、距离、空间位置、空间方向、相对关系、尺寸、距离等数据类型。这些类型都可作为计算从  $q$  到数据对象和最小矩形框的相容距离函数。

FNN 算法, 将数据对象存储在  $R$  树, 将聚类重心作为根节点, 每一个树节点代表一片空间区域, 数据对象只能存储在叶子节点, 数据对象与叶节点所占区域相交。一个节点所占区域必须完全包含在父节点的区域, 且每一节点仅有一父节点。叶子节点的上一层节点包含对象的最小矩形框。最小矩形框可以直接修改删减, 以此减少对象各个代表项 I/O 的代价。

在 FNN 算法的主循环中, 当遍历  $R$  树的一个叶子时, 并不真地计算对象之间的真实距离, 而是计算最小环绕矩阵之间的距离并插入队列。只有已从队列里查询过对象的最小矩形框后才计算真实的距离。在优先级队列中, 如果数据对象离查询对象比其它元素都近, 则这个数据对象就可以报告成最近邻接对象, 否则这个对象以及它的真实距离就插入到队列中。

### 3.3 算法分析

对于查找聚类重心算法, 作者在得到聚类重心之后的示例图像匹配查询中, 采用查询聚类内部线性链表匹配的方式, 则从在  $R$  个聚类中寻找匹配的聚类重心到找到匹配图像, 共需要匹配  $R + N / R$  次, 其中  $N$  为图像总数。一般情况下  $R < N$ 。将该算法与 FNN 算法结合起来使用, 找到聚类重心后在该聚类内部, 使用基于  $R$  树的检索机制, 若在聚类内部共查询了  $k$  个最近邻接节点后找到目标, 经分析平均的时间复杂度为  $O(k \log k)$ 。因为搜索的区域半径  $r$  的值等于聚类重心到最近邻居节点  $I$  的距离  $Distance(I)$ , 根据聚类重心算法得:

$$Distance(I) \leq 1 - (Sim(I) / 100) \quad (10)$$

其中  $Sim(I)$  为图像某个特征与重心相对应的特征的相似度, 取值范围在  $[0, 1]$ ,

FNN 检索区域面积:

$$\pi(Distance(I))^2 = kR/N \quad (11)$$

当我们确定的  $Distance(I)$  值越小, 即对相似性要求越高, 则  $k$  的取值就越小。因为我们首先将图像聚类并计算了聚类重心, 所以相对原先的整个图像库来讲,  $Distance(I)$  已经是在一个较小的取值范围内。相应  $k$  取值越小,  $k \log k$  值相对于  $N / R$  的值来讲就越小, 利用我们的 FNN 算法, 需要扫描到的图像也就较少。所以我们的算法收敛性更优, 查询效率也就较线性链表查询高, 检索的错误率也会相应较小。

另一方面, 原有的查找聚类重心算法, 聚类个数  $R$  的选

择要靠经验决定并且影响着检索效率。在本文的算法中,由于聚类内部不再使用线性查找的方法,而是使用 FNN,检索效率与本聚类内部的总节点数无关,只和扫描到的节点数有关,因此很大程度上减少了  $R$  值的选择对检索效率的影响。

#### 4 查询检索引擎 IRE 的工作原理

多媒体数据库通常要处理极大量的数据,因此一个可以提供高效数据查找的检索结构非常重要。本模型中的研究对象既有属于内容特征的数据,如颜色、纹理,也有空间特征的数据,如距离、相对位置、空间关系等。本章介绍作者针对这类数据提出的一个较灵活的检索引擎 IRE。我们的查询检索技术是建立在基于内容检索技术 CBR、基于空间检索技术 SBR 和基于模糊语义检索技术 SFBR<sup>[2]</sup> 的基础之上的较为灵活集成的一个新的检索机制 IRE (Index And Retrieval Engine), 是一个逐步求精的过程。主要流程如图1所示。

用户通过查询接口提交查询要求,主要通过示例或简单说明的方式。检索引擎 IRE 对示例进行特征提取,将特征与特征库中的特征进行匹配,返回一组候选结果。用户对候选的检索结果进行浏览,选择满意结果,或从候选结果中选择最接近的示例,将它的特征进行调整后形成一个新的查询。按照这种循环方式逐步缩小查询范围,直到用户对查询结果满意和放弃为止。

在检索过程中,给用户提阶段化和数据整理工具以全自动和半自动的方式对图像媒体进行分割、过滤、集成等预处理,并附以相关的知识辅助,以便更好地满足用户的查询要求。

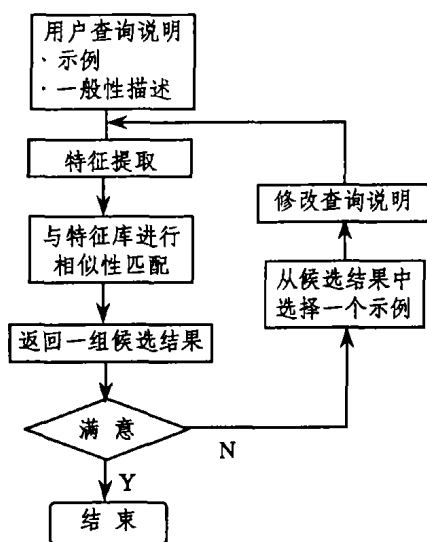


图1 查询检索流程图

特征数据模型主要根据实际运用采用的特征匹配方法来确定。图像库特征匹配方法,使用颜色、纹理、形状、空间特征来进行匹配,图像  $P$  的特征数据模型可以描述如下:

```

Feature Model P
{color feature:
parameter 1: ...;
parameter 2: ...;
.....
veins feature:
parameter 1: ...;
parameter 2: ...;
}
  
```

```

.....
figure feature:
parameter 1: ...;
parameter 2: ...;
.....
}
  
```

#### 5 与同类工作的比较

目前许多研究机构对多媒体数据库的信息检索技术进行研究,如 QBIC 系统和 VIM SYS 系统等能够对图像的形状、纹理、颜色等内容进行查询,对于这些高维度数据,借助聚类思想进行降维处理是一种比较创新的方法,通过聚类的方式加速检索的过程,将图像库中的图像分成  $K$  类,以便减少不必要的匹配和扫描。此方法在 Cluster Indexing Scheme<sup>[5]</sup> 等的研究中处理颜色特征数据,但缺点是需要事先规定聚类个数,它决定着查询效率和准确率。而对基于空间的数据的检索查询系统也有大量的研究成果,并有力支持了 GIS 和 GPS 技术的发展。其中基于  $R$  树结构的查询检索机制因其灵活高效等特点常用于存储和检索以距离函数为主的空间特征数据,但对于高维度的数据查询效率和准确率较低。

本文针对多媒体数据库信息查询通用性的问题,综合与图像媒体有关的空间特征、形状特征、颜色特征、纹理特征等,构建了一个较为集成的特征库。特征库集成并不是简单的对各种特征的合并,而是将特征函数的计算方法和存储机制,尽可能归一化到近邻方法的匹配技术上来。这是本文的主要贡献之一。此外,在集成的特征库基础上进行查询检索,针对高维数据查询效率问题,通过查找聚类重心的加速检索算法提高检索效率,同时利用基于  $R$  树的 FNN 算法,提高聚类内部的检索效率,并使得聚类个数  $R$  的选择对查询效率和准确率不再起到至关重要的作用,降低了因为  $R$  的错选带来的风险。在第3节的算法分析中,对这部分已进行了详细阐述。基于这些工作设计的查询检索引擎 IRE,是一个高效集成的多媒体信息管理模型,具有很好的研究和研究前景。

**结束语** 本文针对多媒体数据的特征研究了相关的存储结构和相似性检索技术,以及基于内容、颜色、距离、形状和语义等的匹配算法。在此基础上构造的查询检索引擎 IRE 可应用于任何属于内容和空间特征的多维数据库中,并且具有较高的查询效率。下一步工作需要将 IRE 系统化,在保证检索准确性的同时,进一步提高算法的查询检索效率。

#### 参考文献

- 1 Ichikawa Y T. A Survey on Content-based Retrieval for Multimedia Databases. IEEE Trans. On Knowledge & Data Engineering, 1999, 11(1): 81~93
- 2 Cao Jia-heng, Peng Min, et al. A Semantic Retrieval Method Based on Fuzzy Reasoning. Wuhan University Journal of Nature Science, 2002, 7(2)
- 3 彭敏,等. 多媒体数据库检索查询的新机制. 武汉大学学报, 2001, 47(3)
- 4 曹加恒, 彭敏, 等. 多媒体数据挖掘的相关媒体特征库方法. 武汉大学学报, 2000, 46(5)
- 5 樊玮红, 等. 一种用于图象检索的聚类方法. 计算机工程, 2000, 26(2)