基于内容特征的图像数据库检索技术及实现①

Retrieval Algorithm and Its Realization for Content-Based Characteristics of the Database Images

闫 实[1] 付 佳[2] (1.牡丹江医学院 教育技术与信息中心 黑龙江 牡丹江 157011;
2.牡丹江医学院 图书馆 黑龙江 牡丹江 157011)

摘 要: 随着图像数据库容量的增大，迫切的需要提高对数据库内图像进行检索的准确率和效率，基于图像内容特征的检索技术发展是一个重要的提高检索效率的途径。本文在分析了基于内容特征的图像检索技术基础上，针对多维索引技术发展及图像数据库的特点，提出了一种新的改进 NB-Tree 的基于颜色特征的图像检索技术，通过引入新的信息特征矢量，实现了检索效率的提高，并给出了一个具体的实例验证了技术的正确性。

关键词: 图像检索 NB-Tree 图像数据库 多维索引技术

1 引言

基于内容特征从图像数据库中进行检索的技术及其实现在于如何恰当地提取反映图像内容的特征，以及基于这些特征的高效鲁棒的分类和检索方法。一直以来，人们都期望计算机在能够接受人类用自然语言对于其内容的描述(称之为高层特征)后，可自动检索出期望的图像。当前研究热点放在如何构建若干规范化、能反映基本图像内容的视觉特征。这些规范化了的图像特征称之为低层特征，图像低层特征主要有颜色、纹理、形状和空间特征等[1]。低层特征提取的有效性，直接影响到图像检索的精确度。

已有不少的研究者提出了很多涉及图像内容，如颜色空间、形状分布等信息特征提取的新方法及实现算法。如闫庆红等研究了一种新的基于视觉特征的数字图书馆图像检索算法等[2-4]。本文通过分析基于内容的图像特征提取方法基础上，重点从考虑颜色空间分布信息内容为特征的图像检索及提取方面进行研究，并分析其算法。

2 基于内容特征的检索技术分析

图像的形状特征由图像的几何特征参数来表征，单个特征和高层语义之间存在的关联性较弱，某些语义上完全不相关的图像的特征矢量在特征空间上的位置相当接近。因此，当单个特征不足以提供足够的鉴别信息时，检索系统往往会给出一些错误的检索结果。因此可引入利用多特征组合检索，语义上不相关的图像虽然在某一特征描述上有很好的相似度，但是同时在其他特征描述上也取得好的相似度的几率是非常小的。

3 图像数据库下的多维检索技术

3.1 多维索引技术发展及图像数据库

多维索引技术是随着应用需求而逐渐提出和发展的。最初，数据量不是很大，顺序扫描已经能够满足多数应用需求。随着计算机辅助设计(CAD)和地理信息系统(GIS)等应用的发展，迫切需要一种高效的索引机制来支持对空间数据的有效检索。于是在上世纪末期出现了大量多维索引结构。这些索引结构将索引数据的维数从单维扩展到了多维而且支持的查询种类也很多，不仅支持传统数据库中的精确查询，也支持范围查询、最近邻查询和空间连接等。进入新世纪以来，数字医疗、数字图书馆等应用领域出现了大量图像数据库，众多研究已经从不同方面改进了图像检索系统的性能。本文对多维检索技术的研究是基于 NB-tree 的多维检索技术[5]。

① 收稿时间:2008-09-01

3.2 基于 NB-tree 的多维检索技术

NB-Tree 是 Manuel J 等人在 2003 年提出的一种新的索引结构[2]。算法基本思想：先对每一个多维适量(P1,P2,…，Pn)计算其欧氏距离：

$$\| P \| = \sqrt{p_1^2 + p_2^2 + ... + p_n^2}$$

计算完每个多维矢量的欧氏距离后，通过欧氏距离将 N 维空间映射到 1 维空间上，用一维空间中最有效的 B+树来构造索引。该算法避开了复杂的结构，利用 B+-Tree 来存储对象的欧氏距离，该算法有较好的适应性。然而 NB-Tree 的缺点是只存储了对象的欧氏距离，忽略了具有相同欧氏距离的对象的位置信息。不可避免会有欧氏距离相近而多维矢量；不相关的对象被作为检索对象读入内存，进行二次过滤，增大了不必要的 I/O 操作和距离计算。已经证明，磁盘进行一次 I/O 操作的时间里，一台典型的机器能执行百万条的指令，因此 I/O 所用的时间是算法所用时间的近似值，是影响索引算法效率的主要因素，通过仔细研究原来的算法，我们引入多维矢量的空间位置信息：偏移角，提出了一种新的索引结构：New-NB-Tree，通过加强过滤能力，进一步减少访问对象的数目。进而减少 I/O 操作，实验表明，New-NB-Tree 能大大提高检索效率。

3.3 New-NB-Tree 的设计思想

针对 NB-Tree 的不足之处，我们在 NB-Tree 的叶子节点上加入和维数无关的偏移角信息，设计了一种新的索引结构 New-NB-Tree。通过少量的计算，进一步加强过滤，最大程度的减少候选数据里的“脏”数据。通过改进的索引结构，保留了 NB-Tree 的优点，有效改进了原来算法的不足，提高了检索效率。设计思想如下：

在 N 维空间中，选取向量[1,1,…,1n](n 是矢量维数)作为基准向量，计算每个对象的特征向量与该向量的夹角，并把该夹角信息写入索引的叶子节点中。该夹角信息说明了对象的特征向量在空间中的位置。

改进后的节点结构如图 1 所示。

NB-Tree 检索过程中点查询：首先计算查询对象 Q 的欧氏距离 R，利用 NB-Tree 初步确定欧氏距离为 R 的查询结果，然后计算查询结果中的对象和 Q 的距离，若有结果为零，则说明 Q 在 NB-Tree 中，返回结果。该算法简单有效，很容易实现；避免了维数危机，随着维数的上升，性能变化缓慢；B+-Tree 的优良性能，使得该算法也有很好的性能表现。

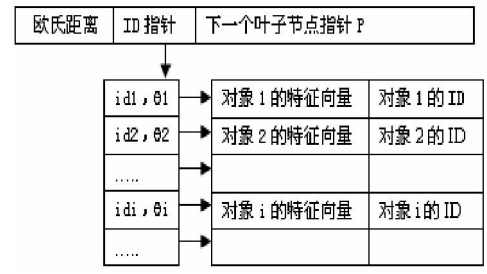

图 1　改进后的节点结构

4　New-NB-Tree 的范围查询算法

该索引方法的范围查询需分三个步骤：首先是是否满足欧氏距离条件，如果满足；进行第二步：角度过滤，生成候选集；第三步将候选集中的对象读入内存进行精确匹配，计算每个对象与查询对象之间的欧氏距离，满足条件的对象作为查询结果输出。查询示意图如图 2 所示。

查询的具体步骤如下：

输入查询向量 x 和查询半径 r;

(1)计算 x 到原点的欧氏距离 Dist 和 x 与基准向量的夹角 β ,x 与切线的夹角 θ ;

(2)结点指针 P 指向索引 B+树的根;

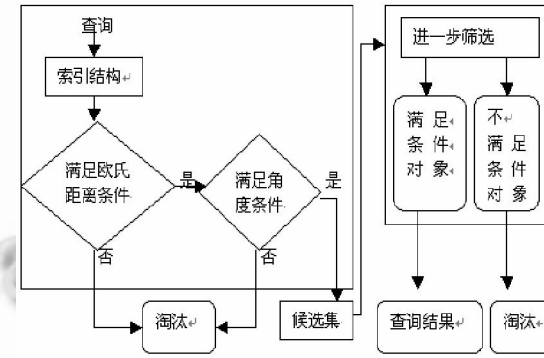

图 2　New-NB-Tree 的查询示意图

(3)如果 Dist-r > P.dist[P.nodecount]则无查询结果输出,查询过程结束;

(4)如果 P.dist[i-1]< Dist-r< P.dist[i]

当 P.type=0 则 P=P.child[i] 程序转至步骤 4;

当 P.type=1 则 FPI=P.P[i];

(5)从 LEAF_FILE 的 FPI 位置读入 m 个记录到内存 Buffer,令 j=0;

(6)如果 Buffer[j]. dist>Dist+r 查询过程结束;

(7)如果 max(Dist-r,0)≤Buffer[j]. dist≤Dist+r

当 max(0, β - θ)≤Buffer[j].rad≤ β + θ 同时成立，

则从 DATA_FILE 中 Buffer[j].fp 位置读入一条记录，根据相似度公式，计算其于查询对象之间的精确距离，对于查询范围内的对象，作为查询结果输出；

(8)j=j+1；如果 j!=m 程序转至步骤(7)，否则 FPI=FPI+m*RL 程序转至步骤(6)；

5 测试结果及分析

5.1 测试环境

本文中所做的所有测试使用的软硬件环境如下：

(1)硬件环境：P4 联想兼容机，3.0GHz 双 CPU，1G 内存，160GB 硬盘。

(2)操作系统平台：Microsoft Windows XP

(3)编程环境：Microsoft Visual C++6.0 编译器。

5.2 测试方案

我们选用专用数据库提供的大容量图像特征数据集。该数据集包括从 68040 副各种各样的图像中提取的 16 维的纹理特征数据集、32 维的颜色特征集和 9 维的颜色特征集。该组数据集曾被用作 Hybrid-Tree 的测试数据。

对每个数据集，查询时随即在数据集里选取 10 个查询矢量，每个查询矢量对不同的查询半径分别查询 1 次，并记录下角度信息过滤前后的数据量，表 1 是查询结果。由于所测试的索引结构都是基于外存的索引结构，因此内存和外存之间的 I/O 操作是影响查询性能的最主要的因素。而每次查询执行之后，查找路径上的节点都被读进了内存中，这样下次查询时，就减少了外存到内存的读操作，这样得到的查询响应时间就是所谓的热结果，但是这种热结果反应不出基于外存的索引结构的性能特点，因此我们只使用冷结果作为评价算法性能的依据：每次查询进行之后，都有相应的清理内存的动作。

5.3 结果分析

在磁盘进行一次 I/O 的时间里，一台典型的机器能执行百万条的指令，因此 I/O 操作所用的时间是算法所用时间的近似值，是影响索引算法效率的主要因素。在实验中以角度过滤前后的对象数量作为 I/O 次数比较的量度。这里采用的检索效率提高程度的计算公式为：

$\sum_{i=1}^{N}\frac{oldnum_i-newnum_i}{oldnum_i}$ 其中，oldnumi 表示第 i 次旧算法过滤后的返回的数据量，newnumi 表示第 i 次新算法过滤后的返回的数据量，N 表示检索的次数。按着这种计算方法得到 4 种情况的检索效率的平均提高率分别为：98.46%，97.87%，97.43%，96.86%。从实验结果可以看出，当查询半径较小时，索引结构的效率提高较大。

表 1　实验结果分析表　　单位：次

查询半径 0.04		查询半径 0.06		查询半径 0.08		查询半径 0.1	
旧算法	新算法	旧算法	新算法	旧算法	新算法	旧算法	新算法
40	2	69	4	95	5	116	7
232	1	367	4	481	13	587	19
1216	5	1826	28	2429	51	3048	82
2195	34	3231	77	4290	132	5404	203
2063	30	3057	79	4019	136	4960	206
349	3	525	4	732	7	893	11
476	5	694	5	950	9	1180	12

6 结论

我们选择该类算法中的一个典型算法 NB-Tree 进行了深入研究。针对 NB-Tree 只存储对象的欧氏距离而忽略位置信息的缺点，在索引的叶子节点中加入了能体现多维矢量空间位置关系的角度信息。通过对原算法的改进设计了一种新的索引结构。新的索引结构通过较小的存储空间和代价不大的计算，进一步加强了过滤功能。减少了范围查询的 I/O 次数，提高了算法的效率。通过实验验证了该索引的性能。新的索引结构不仅增强了过滤能力，也很好的保留了原算法与维数无关的特性。因为计算角度时的基准向量采用单位矢量，所以与维数无关。因此，该索引方法可以使用于大容量、不定维的高维矢量数据库。

参考文献

1 Kankanhalli MS, Mehtre BM, Wu JK. Cluster-Based Color Matching for Image Retrieval. Pattern Recognition, 1996,29(4):701－709.

2 Manuel JF, Joaquim J. Indexing High-Dimensional Data for Content-Based Retrieval in Large Databases. Proceedings of the 8th International Conference on Database Systems for Advanced Applications (DASFAA '03). Mar 2003.

3 赵英,刘佳佳.基于贝叶斯定理的遥感图像检索.现代图书情报技术,2006, 136(5):36－39.

4 原福永,王海霞,杨治秋.基于内容图像检索中纹理分析的研究.现代图书情报技术,2006,132(1):59－61.

5 闫庆红,彭宇行.一种新的数字图书馆图像检索算法.现代图书情报技术,2005,131(12):30－33.