一种支持快速相似检索的多维索引结构*

冯玉才, 曹 奎, 曹忠升

(华中科学技术大学 计算机科学与技术学院,湖北 武汉 430074)
E-mail: ck2896@263.net
http://www.dm2.com.cn

摘要: 基于内容的图像检索是一种典型的相似检索问题,对于尺度空间上的图像相似匹配问题,一般认为距离计算费用很高.因此,需要建立有效的索引结构,以减少每个查询中的距离计算次数.为此,基于数据空间的“优化划分”,并且使用“代表点”,以层次结构方式划分数据,提出了一种新的基于距离的相似索引结构 opt-树及其变种 η-树.为了更有效地支持基于内容的图像检索,在η-树索引结构中采用了“η-最优化划分”和“η-对称冗余存储”策略,以提高相似检索的效率.详细讨论了这种索引结构的建立与检索等问题,并给出了相应的算法.实验结果显示了这种索引技术的有效性.

关 键 词: 高维索引结构;相似检索;尺度空间;距离函数;基于距离;基于内容的图像检索
中图法分类号: TP311,TP134 文献标识码: A

在图像数据库领域中,基于内容的图像检索(content-based image retrieval,简称 CBIR)技术日益受到人们的重视.CBIR 是一种信息检索技术,它关注的是以基于内容的方法快速发现信息.一般地,图像内容可由一组低层特征来描述,其中包括颜色、形状、纹理、空间位置以及图像对象间的相互关系等.CBIR 是一种相似匹配,它使用距离度量函数来计算两个图像间的相似度,评价的标准是预先定义的.相似查询是多媒体信息系统基于内容检索的本质需求,能否有效地支持这一特性是衡量系统查询功能强弱的重要标志.

多维索引结构是 CBIR 研究的一个基本问题.一般地,图像特征的表示均是高维向量,并且图像的距离度量也不仅限于 L_2 距离.因此,一个图像可使用尺度空间(metric space,尺度空间的定义见第 2 节)中的一个点来表示,而图像相似检索可视为在高维特征空间中寻找与指定点距离最近的一组点的问题.我们注意到,图像数据库一般都非常大,采用顺序比较方法显然不能满足实时检索的要求,因而需要建立一种有效的多维索引结构及算法以支持快速相似检索,即对于给定的例子图像(query image),利用这种索引结构可以在很小的范围内快速得到与例子图像距离最近(最相似)的一组图像.在传统计算机科学中,这个问题对应于 k 最近邻搜索(K-NNSearch)问题或相似索引问题(similarity indexing).常用的高维索引结构包括 k-d 树、R 树及其变种[1~5],这些索引结构大都是针对欧氏向量空间而设计的.对于一般的尺度空间,索引结构的建立与查询只能借助数据点之间的距离度量,并且利用“三角不等式”来减少相似检索中的距离计算次数,而不能使用任何其他的几何假设.Uhlmann[6]于 1991 年提出了一种支持相似检索的索引结构 VP 树,它是一种真正基于距离的尺度空间上的索引结构.除此之外,基于距离的索引结构还包括 FQ-tree[7],GNAT[8]和 M-tree[9]等.

对于尺度空间上的相似检索问题,一般认为距离计算费用很高.因此,建立索引结构的目的就是尽可能减少每个查询所需的距离计算次数,从而提高检索效率.考察这些基于距离的索引结构,它们的检索性能一般依赖于

* 收稿日期: 2001-03-23; 修改日期: 2001-07-18
基金项目: 国家 863 高科技发展计划资助项目(863-511-920-001);国家“九五”国防预研基金资助项目(15.4.1)
作者简介: 冯玉才(1946－),男,江苏扬州人,教授,博士生导师,主要研究领域为数据库技术,GIS;曹奎(1963－),男,河南驻马店人,博士生,副教授,主要研究领域为基于内容的图像检索,高维索引技术;曹忠升(1965－),男,湖北武汉人,博士,副教授,主要研究领域为数据库,多媒体技术.

具体的数据分布,并且大都包括一些需要由经验确定的设计参数(如 VP-树中的 vantage points),但却并非检索性能“最优”意义下的索引结构.为此,我们基于数据空间的“优化分割”,提出了一种新的基于距离的索引结构 opt-树及其变种η-树,讨论了它们的建立与检索等问题,并给出了相应的算法.

1 尺度空间与相似检索

尺度空间可以定义为一个二元组:$M=(D,d)$,其中 D 是对象的特征空间,d 是一个定义在 D 上的距离度量,并且满足下列条件①～④.

① $d(x,y)=d(y,x)$;

② $d(x,x)=0$;

③ $0<d(x,y)<\infty,\ x\neq y$;

④ $d(x,y)\le d(x,z)+d(z,y)$ (三角不等式).

由尺度空间的定义可知,当在尺度空间中建立基于距离的索引结构时,我们只能使用上述定义中的 4 个条件,而不能像在欧氏空间中使用任何其他的几何假设.

给定一个图像数据库 S,d 为图像特征空间上的一个距离度量,Q 为例子图像.一般地,图像检索可分为如下两种主要类型:

阈值型查询 Query(Q,t):对于给定的阈值 t,在图像库 S 中查询满足 $d(Q,I)\le t$ 的所有目标图像 I.

最佳匹配查询 Query(Q,n):在图像库中查询与例子图像 Q 距离最近的 n 个目标图像.

对于相似检索问题,我们可以利用三角不等式来减少距离计算的次数,从而提高相似检索的效率.采用的具体方法为[10]:

设 I 为 S 中的任一个图像,$K=\{K_1,K_2,\ldots,K_n\}$是一个键对象集合(K_i 称为键对象),利用三角不等式可得,

$$d(I,Q)\ge\max_{1\le s\le m}\left|d(I,K_s)-d(Q,K_s)\right|. \tag{1}$$

由式(1)可知,对于任意的 $s(1\le s\le n)$,均有 $d(I,Q)\ge|d(I,K_s)-d(Q,K_s)|$成立,从而可得到 I 与 Q 之间距离的一个下界.

考虑一个大数据库 $S=\{I_1,I_2,\ldots,I_n\}$和一个非常小的键对象集合 $K=\{K_1,K_2,\ldots,K_n\}$.若对任意的 s 与 $t(1\le s\le n,1\le t\le m)$,我们预先计算好 I_s 与 K_t 之间的距离 $d(I_s,K_t)$,那么对于相似检索 Query(Q,t)来说,只需要计算$\{d(Q,K_1),d(Q,K_2),\ldots,d(Q,K_m)\}$,并且应用不等式(1)就可以得到相应的距离下界.显然,若能证明 $d(I_s,Q)>t$ 成立,则我们就可以将 I_s 从 Q 的候选匹配集中去除掉.经过这种过滤后,对剩下的对象可采用线性搜索方法逐个计算其距离,并将满足条件的对象放入检索结果集中.在这种基于三角不等式的相似检索策略中,借助“距离下界”就可以将那些因与查询对象的距离太远而不可能成为候选匹配的对象过滤掉,从而减少了查询中距离计算的次数.采用这种策略的检索算法,只需进行 $m+u$ 次距离计算(u 为过滤后剩余对象的个数)和 $O(mn)$次简单运算.不难看出,只要满足 $m+u\le n$,就可以节省大量的距离计算费用,从而显著地提高相似检索的效率.

2 η-最优化划分与 opt-树索引结构

首先,我们讨论数据集的划分.为了方便解释,在不引起混淆的情况下,我们对“图像”、“图像的特征向量”以及“点”和“特征向量”不加区别.

定义 **1**(η-最优化划分). 设 D 为一个高维数据集,d 为其上的一个距离度量.在数据空间中取两个点 $C1$ 和 $C2$($C1$ 和 $C2$ 可以属于 D,也可以不属于 D),借助这两个点可以将 D 划分为两个子集 $D1$ 和 $D2$,使得对于 D 中任意一点 X,若 $d(X,C1)\le d(X,C2)$,则将 X 划分到 $D1$ 中;否则将其划分到 $D2$ 中.

若该划分满足下列条件(1),则称其为“平衡划分”;对于给定的小正数 $\eta>0$,若该划分满足下列条件(1)～(3),就称为“η-最优化划分”,相应的点 $C1$ 和 $C2$ 被称为该划分的“代表点”.

(1) abs(|$D1$|−|$D2$|)最小,即 $D1$ 和 $D2$ 中所包含的数据点的个数相差最少,其中|•|为求集合基数算子.

(2) 设 $D'1=\{X|d(X,C1)-d(X,C2)\le 2\eta\wedge X\in D2\}$,$D'2=\{X|d(X,C2)-d(X,C1)\le 2\eta\wedge X\in D1\}$,则要求该划分使 $D'1\cup D'2$ 所包含的数据点最少.

(3) $d(C1,C2)>\eta$.

由定义1可知,η-最优化划分要解决的关键问题是如何选取代表点,这是一个优化问题.由于η是η-最优化划分的一个参数,我们就将这种数据集分割方法称为"η-最优化划分".采用一般优化问题的求解方法,例如模拟退火方法[11]、遗传算法[12]等,就可以对数据集进行平衡划分或η-最优化划分.

对于给定的数据集,我们可以采用平衡划分或η-最优化划分对其进行递归划分,据此建立相应的 opt-树(优化树).opt-树索引结构的基本思想是采用一种平衡划分或η-最优化划分方法,将数据空间 I 划分成两个子集,递归地对每个子集进行同样的划分,直到划分出的每个子集仅包含指定数目的数据点为止.因此,opt-树是一种二叉树结构,它表示了对数据空间的一个递归划分过程.opt-树的每个内部节点代表了数据空间的一次划分,它的结构定义为

$$Node:(C1,C2,R_{ptr},L_{ptr}).$$

这里,$C1$ 和 $C2$ 是该划分的两个代表点,R_{ptr} 和 L_{ptr} 分别为左、右子树的指针.

opt-树的叶节点的结构定义为

$$Leaf:(E_1,E_2,\ldots,E_k),E_i:(D_1[i],D_2[i],P_i),$$

其中 P_i 为叶节点中存放的 k 个数据点(k 为叶节点的扇出(fanouts)),而数组 $D_1[i]$和 $D_2[i]$分别存放 P_i 与其父节点的一个代表点之间的 k 个距离值,存储这 $2k$ 个距离的目的是为了在检索中利用三角不等式减少距离计算的次数.

2.1 opt-树的建立

设 I 是一个包含有 n 个对象的数据集 $I=(O_1,O_2,\ldots,O_n)$,d 为一个距离度量,opt-树的建立算法如下:

输入:数据集 I

输出:opt-树 V

(1) 若$|I|=0$,则建立空树,算法返回.

(2) 否则,

(2.1) 使用一种平衡划分或η-最优化划分方法将数据集 I 划分成两个子集 D_l 和 D_r(该划分的代表点分别为 $C1$ 和 $C2$),并且

$$D_l=\{O_i \mid d(C1,O_i)\le d(C2,O_i)\wedge O_i\in I\},D_r=\{O_j \mid d(C2,O_j)<d(C1,O_j)\wedge O_j\in I\}$$

(2.2) 分别以 D_l 和 D_r 作为树 V 根节点的左、右子树.

(2.3) 若 D_l 或 D_r 为叶节点,则计算距离 $d(Ci,O_j)$,并将其存放到叶节点的距离数组 $D_i[j]$中,算法返回.

(3) 分别对 D_l 和 D_r 递归地进行如上处理,得到相应的 opt-树 V.

由上述 opt-树的建立过程不难看出,opt-树是一种真正基于距离的索引结构,它是一棵二叉树结构,容易将其组织成页面存入外存中(内部节点的扇出为2,叶节点的扇出为 k,k 是opt-树的一个设计参数).如果使用距离计算次数作为度量准则,则建立 opt-树的计算复杂度为 $O(n\log_2 n)$.

定理 **1**. 设 D 为一个数据集,d 是其上的一个距离度量,$D1$ 和 $D2$ 分别为使用平衡划分或η-最优化划分得到的两个子集,$C1$ 和 $C2$ 为该划分的代表点.考虑相似查询 Query(q,t)(q 为查询对象,t 为阈值),我们有:

在 $d(q,C1)\le d(q,C2)$的情况下,若存在一个点 $x\in D2$,使不等式 $d(x,C1)-d(x,C2)\le 2t$ 成立,则必须搜索 $D1$ 和 $D2$ 来执行相似查询 Query(q,t);否则,Query(q,t)仅需搜索 $D1$.

在 $d(q,C2)<d(q,C1)$的情况下,若存在一个点 $x\in D1$,使不等式 $d(x,C2)-d(x,C1)\le 2t$ 成立,则必须搜索 $D1$ 和 $D2$ 来执行相似查询 Query(q,t);否则,Query(q,t)仅需搜索 $D2$.

证明:在第 1 种情况下,即 $d(q,C1)\le d(q,C2)$.因为 d 是一个距离度量,由距离度量的定义可得下列两个不等式成立:$d(q,C1)+d(C1,x)\ge d(q,x)$,$d(q,C1)+d(q,x)\ge d(C1,x)$,由这两个不等式可推得

$$d(q,C1)\ge|d(q,x)-d(C1,x)|, \tag{2}$$

和

$$d(q,C2)\le|d(q,x)+d(C2,x)|. \tag{3}$$

由不等式(2)和(3)可得,$d^2(q,C1)\geq[d(q,x)-d(C1,x)]^2$ 和 $d^2(q,C2)\leq[d(q,x)+d(C2,x)]^2$.

由假设条件 $d(q,C1)\leq d(q,C2)$可推得

$$[d(q,x)-d(C1,x)]^2\leq[d(q,x)+d(C2,x)]^2,$$

即

$$\begin{aligned}-2d(q,x)d(C1,x)+d^2(C1,x)\leq 2d(q,x)d(C2,x)+d^2(C2,x),\\ 2d(q,x)[d(C1,x)+d(C2,x)]\geq d^2(C1,x)-d^2(C2,x),\\ d(q,x)\geq(d(C1,x)-d(C2,x))/2.\end{aligned}\tag{4}$$

从不等式(4)可知,若不等式 $d(x,C1)-d(x,C2)>2t$ 成立,我们有 $d(q,x)>t$.也就是说,若 $D2$ 中的任一点 x 满足不等式 $d(x,C1)-d(x,C2)>2t$,那么 x 就不可能成为查询对象的侯选匹配.因此,我们仅需搜索 $D1$ 来执行查询 Query(q,t).否则,若存在一个点 $x\in D2$ 使不等式 $d(x,C1)-d(x,C2)\leq 2t$ 成立,此时我们不能确定 $d(q,x)$是否小于阈值 t,从而必须同时搜索 $D1$ 和 $D2$ 来执行查询 Query(q,t).

同理可证,在第 2 种情况下结论成立. □

推论 **1**. 设条件同定理 1,则

(1) 在 $d(q,C1)\leq d(q,C2)$的情况下,若不等式 $d(q,C2)-d(q,C1)\leq 2t$ 成立,则必须搜索 $D1$ 和 $D2$ 来执行相似查询 Query(q,t);否则,Query(q,t)仅需搜索 $D1$.

(2) 在 $d(q,C2)<d(q,C1)$的情况下,若不等式 $d(q,C1)-d(q,C2)\leq 2t$ 成立,则必须搜索 $D1$ 和 $D2$ 来执行相似查询 Query(q,t);否则,Query(q,t)仅需搜索 $D2$.

证明:考虑第 1 种情况,即 $d(q,C1)\leq d(q,C2)$.

令$\delta'=d(q,C2)-d(q,C1)$,对于任意的 $x\in D2$,设 $d(x,C1)-d(x,C2)=\delta>0$(见定义 1).

利用三角不等式,可得

$$d(q,x)>d(x,C1)-d(q,C1),\quad d(q,x)>d(q,C2)-d(x,C2),$$

由此推得,

$$2d(q,x)>[d(x,C1)-d(x,C2)]+[d(q,C2)-d(q,C1)]=\delta+\delta',$$

即

$$d(q,x)>(\delta+\delta')/2.$$

若$\delta'=d(q,C2)-d(q,C1)>2t$,则 $d(q,x)>(\delta+\delta')/2\geq t+\delta/2>t$.由此可知,此时的 $x\in D2$ 不可能成为 q 的候选匹配.由 x 点的任意性可知,Query(q,t)仅需搜索 $D1$.

若$\delta'=d(q,C2)-d(q,C1)\leq 2t$,我们不能保证 $d(q,x)>t$ 一定成立,也就是说,必须搜索 $D1$ 和 $D2$ 来执行 Query(q,t).

同理可证,在第 2 种情况下结论成立. □

2.2 **opt**-树的检索

借助于推论 1,我们可以给出 opt-树的检索算法,它是一个 opt-树的深度优先搜索算法.

opt-树的检索算法.

输入:opt-树 V

输出:检索结果集 *result*

(1) 取当前节点;

(2) 若当前节点为叶节点,则对于叶节点中的每个数据点 P_i,

(2.2) 取距离数组 $D1$ 和 $D2$;

(2.2) 若 $\max\{|d(q,C1)-D1[i]|,|d(q,C2)-D2[i]|\}>t$,则 P_i 为非候选匹配(无须计算距离即可将其排除);
否则,计算 $d(q,P_i)$.若 $d(q,P_i)\leq t$,则将其放入检索结果集 *result* 中;

(2.3) 算法返回;

(3) 若当前节点为内部节点,则

(3.1) 计算距离 $d(q,C1)$和 $d(q,C2)$;

(3.2) 若 $d(q,C1)\le d(q,C2)$,

若 $d(q,C2)-d(q,C1)\le 2t$,则递归搜索其左右子树;

否则,递归搜索左子树;

(3.3) 若 $d(q,C2)<d(q,C1)$,

若 $d(q,C1)-d(q,C2)\le 2t$,则递归搜索其左右子树;

否则,递归搜索右子树;

从 opt-树的检索算法不难发现,opt-树可以提高相似查询的搜索速度,其查询效率不会因维数的增加而明显下降,它是一种基于距离的多维空间索引结构.应当指出,由于 opt-树是一种二叉查找树结构,对于大容量数据集来说,其层次(高度)较多,再加上需要递归搜索子树,因而影响了 opt-树的检索性能.为了提高 opt-树的检索性能,可行的改进方法是采用增加 opt-树节点扇出(fanouts)的方法以降低树的高度,从而达到减少查询中距离计算次数的目的(有关采用这种技术来改进 opt-树的方法,我们将另文讨论).事实上,改进 opt-树的方法并不仅限于通过增加节点扇出这种途径,下面我们将讨论另一种有效的解决方案.

考察“η-最优化划分”定义中的条件(2),对于用户查询 Query(q,t),假设 $d(q,C1)\le d(q,C2)$且 $d(q,C2)-d(q,C1)\le 2t$,由推论定理 1 可知,该查询必须对 $D1$ 和 $D2$ 均搜索.显然,只要我们选取的η值满足$\eta\ge t$,$D'1$ 实际上就包含了 $D2$ 中对象 q 的所有可能的匹配.也就是说,查询 Query(q,t)只需搜索 $D1$ 和 $D'1$ 即可.分析条件(2),它实际上是为了保证“η-最优化划分”划分出的两个子集彼此最大限度地“分离”.理想情况下,$D'1$ 和 $D'2$ 中只包含很少的数据点或为空集.因此,在大多数情况下,查询 Query(q,t)对 $D2$ 的搜索是低效的.一种提高 opt-树检索性能的方法就是仅搜索 $D1'$而避免对 $D2$ 的搜索.注意到上述分析结果成立的前提条件是$\eta\ge t$,由于 t 代表了一个查询的阈值,而不同查询可能具有不同的阈值.能否找到一个这样的η,对于所有可能的用户查询 Query(q,t)均满足$\eta\ge t$ 呢?对于图像检索应用来说,我们可以找到一种确定η的方法.

考察阈值型图像检索 Query(q,t),对于某一种图像特征(例如图像的颜色直方图),在图像相似的意义下,阈值 t 应限制在一定的范围内.也就是说,在图像相似的意义下,阈值 t 应具有确定的上界.因此,给定一个图像数据库 I 以及相应的距离度量 d,我们可以建立相应的基于参数η的 opt-树,参数η可以根据图像特征的性质来选取(参数η的选取方法将在第 3 节加以讨论).

2.3 η-树

这里,我们给出一种改进的 opt-树索引结构——η-树,它保证每个查询仅沿η-树中的一条路经搜索.

定义 **3**(**η-对称冗余存储**). 设 $D,D1,D2,C1$ 和 $C2$ 的含义同定义 1.设 $D'1=\{x|d(x,C1)-d(x,C2)\le 2\eta\wedge x\in D2\}$, $D'2=\{x|d(x,C2)-d(x,C1)\le 2\eta\wedge x\in D1\}$,将子集 $D1'$和 $D2'$中的数据点分别重复存放到子集 $D1$ 和 $D2$ 之中,得到两个扩展子集 $\overline{D1}$和 $\overline{D2}$,即 $\overline{D1}=D1\cup D'1$, $\overline{D2}=D2\cup D'2$.我们将上述对划分的处理称为“η-对称冗余存储”.

从定义 3 不难看出,经过对一个划分进行η-对称冗余存储处理,我们将得到两个相互覆盖的扩展子集,其冗余度是由参数η和具体的数据分布确定的.对于η-对称冗余存储,我们使用不等式 $d(C1,x)-d(C2,x)>2\eta$或 $d(C2,x)-d(C1,x)>2\eta$来排除那些因与查询对象距离太远而不可能成为候选匹配的那些数据对象.这种处理策略本质上是基于概率的,但它可以保证对于相似检索问题是正确的.这样,我们就得到定理 2.

定理 **2**. 设 $D,D1,D2,\overline{D1},\overline{D2},C1$ 和 $C2$ 的含义同定义 3,并且对任意的用户查询 Query(q,t),有$\eta\ge t$,则下列结论成立:若 $d(q,C1)\le d(q,C2)$成立,则相似查询 Query(q,t)只需搜索 $\overline{D1}$;否则,相似查询 Query(q,t)只需搜索 $\overline{D2}$.

证明:若 $d(q,C1)\le d(q,C2)$,要证明查询 Query(q,t)只需搜索 $\overline{D1}$,由定义 3 可知:只需证明对于 $D2$ 中任意一点 $x\in D2$,若 $x\notin\overline{D1}$,则 x 不可能是 q 的候选匹配.

由定义 3 可知,若 $x\notin\overline{D1}$, $x\in D2$,则 $d(x,C1)-d(x,C2)>2\eta$.

使用“推论 1”证明中相同的方法可得,$d(q,x)>(\delta+\delta')/2$($\delta,\delta'$的含义同推论 1).

由此可推得,$d(q,x)>\eta+\delta/2>\eta\ge t$,从而有 $d(q,x)>t$.即 x 不是 q 的候选匹配.亦即 Query(q,t)只需搜索 $\overline{D1}$.

同理可证,若 $d(q,C2)<d(q,C1)$,Query(q,t)只需搜索 $\overline{D2}$. □

定义 **4**(**η-树**). 设 I 为一个图像特征向量集,选取参数η,并采用某种η-最优化划分方法将特征空间划分为两

个子集 DL 和 DR,再采用η-对称冗余存储分别对子集 DL 和 DR 进行扩展,即 $\overline{DL}=DL\cup DL'$,$\overline{DR}=DR\cup DR'$.对每个扩展子集($\overline{DL}$ 和 $\overline{DR}$)进行同样的递归划分和处理,直到划分出的每个子集仅包含指定数目的数据点或划分出的子集"足够小"为止(若数据子集 $DL(DR)$满足 $DL'=DR$ ($DR'=DL$),则称 $DL(DR)$足够小),我们把经这种数据空间划分而得到的二叉树结构称为η-树.

η-树的节点结构与 opt-树类似,在此不再赘述.由定理 2 可知,基于η-树索引结构的相似检索是相当容易的,只需沿η-树中的一条路经搜索即可,从而显著地提高了η-树的查询效率.

η-树的检索算法(设相似检索为 Query(q,t)).

输入:η-树

输出:检索结果集 *result*

(1) 若当前节点为叶节点,则对于叶节点中的每个数据点 P_i,

(1.1) 取距离数组分别存入 $D1$ 和 $D2$;

(1.2) 若 $\max\{|d(q,C1)-D1[i]|,|d(q,C2)-D2[i]|\}>t$,则 P_i 为非候选匹配(无须计算距离即可将其排除);否则计算 $d(q,P_i)$.若 $d(q,P_i)\le t$,则将其放入检索结果集 *result* 中;

(1.3) 算法返回.

(2) 若当前节点为内部节点,则

(2.1) 若 $d(q,C1)\le d(q,C2)$,则递归搜索左子树;

(2.2) 若 $d(q,C2)<d(q,C1)$,则递归搜索右子树.

3 参数η的选取

η-树索引结构的检索性能依赖于多个因素,其中包括图像集的特性、距离度量、参数η的选取等.在这些因素中,参数η的选取尤为重要.如果选取的参数η太大,数据的冗余度将增加,从而导致存储效率降低和检索性能下降;相反地,图像检索的漏检率就会增加.从参数η在η-树中所起的作用不难看出,对于给定的图像特征,η代表了相似图像之间的一个距离阈值.即对于两幅图像,如果它们的距离大于η,则这两幅图像是不相似的.因此,η的值可以根据图像特征集的统计特征来确定.η值的确定分如下 3 步进行:

Step 1. 在图像集中随机地选取 n 对相似图像,n 对于图像集来说是足够大的;

Step 2. 计算每对图像之间的距离,根据这些距离的分布来确定一个合适的图像相似阈值 T,例如取这些距离的均值为 T.

Step 3. 根据用户的查询需求,由 T 导出参数η.

由于 T 的值是根据相似图像的距离分布来确定的,因而不同的图像集可能导出不同的 T.考虑图像查询 Qurey(q,t),其中的 t 表示了用户对检索结果的要求.考察用户查询中 t 值的分布特征,t 的取值可以近似地模型化为一个 Gaussian 分布 $N(T,\sigma)$.我们可以通过实验或参数估计方法来确定σ的值.根据 3σ-规则,参数η的值为 $\eta=T+3\sigma$.

需要指出的是,为了确定参数η的值,我们可以在图像数据库中选取一个具有统计意义的子集,并在其上以离线方式计算得到.另外,如果图像库足够大,那么当新图像加入数据库后参数η无须重新计算.

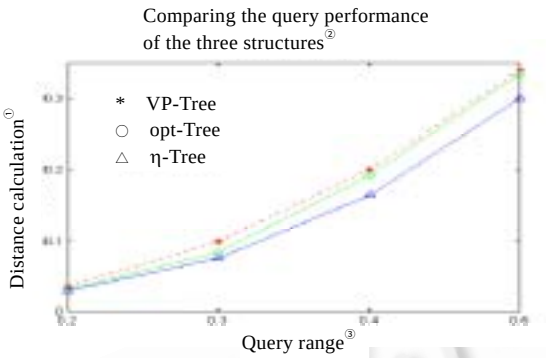
4 实验结果与讨论

在实验中,我们对基于η-树的内查询(即所有索引都存放在内存中的查询)进行了测试,并与 VP 树索引结构的检索性能进行了比较和分析.3 种索引结构:opt-树、η-树和 VP 树均使用 windows 环境下的 C 语言实现.对于高维尺度空间中的相似检索问题,由于距离计算的费用高,我们就使用"距离计算次数"作为算法计算复杂度的度量准则.另一方面,由于我们实现的是内存索引结构,所以这里我们并不考虑磁盘的 I/O 操作.

在实验中,首先选取两个数据集,一个数据集 DS_1 包含 10 000 个均匀分布的 10 维向量,另一个是由 1 150 幅彩色图像的颜色直方图组成的数据集 DS_2(该图像集是从 Internet 获得的,采用 HSV 彩色模型,并且使用 V 分量建立 256 维颜色直方图作为图像的特征向量,直方图距离度量采用 L_1 距离);然后随机地生成若干个用于查询的

点数据;最后计算平均检索费用,即一个检索所需的距离计算次数的平均值.

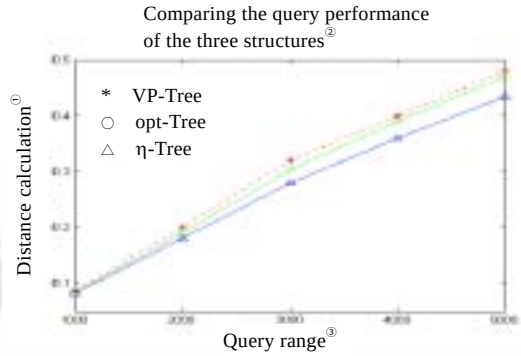
对于数据集 DS_1 和 DS_2,分别建立相应的 opt-树、η-树和 VP 树.我们随机地选取 100 个查询数据点,计算这 100 个查询的平均距离计算次数,并据此来比较这 3 种索引结构的检索性能.图 1、图 2 分别给出了数据集 DS_1 和 DS_2 上 3 种索引结构的检索性能比较(其中"检索费用"分别是由检索中距离计算次数除以 50 000 和 1 150 后得到的).实验结果表明,η-树具有最好的检索性能,特别是对于较大的检索范围尤为如此.对于所有的查询范围,opt-树的检索性能均优于 VP 树.



①检索费用,②3 种索引结构比较,③查询范围.

Fig.1 The comparison of retrieval performance on DS_1

图 1 数据集 DS_1 上的检索性能比较



①检索费用,②3 种索引结构比较,③查询范围.

Fig.2 The comparison of retrieval performance on DS_2

图 2 数据集 DS_2 上的检索性能比较

5 结束语

针对尺度空间上的相似检索问题,我们引入了数据集的两种分割方法:"平衡划分"和"η-最优化划分",并据此提出了一种基于距离的相似索引结构 opt-树,它是一种二叉树结构,并且支持相似检索.为了提高 opt-树的检索性能,我们对其进行了改进,提出了η-树索引结构,它支持对具有任意分布特征的大容量图像库的快速相似检索.对于任意的图像集,根据性质以及图像的分布特征可以确定一个合适的参数值 η,在此基础上就可以建立相应的η-树.对于图像检索来说,利用它可以过滤掉那些与例子图像的距离太大以至于不可能成为候选匹配的那些图像,使查询的搜索范围仅限在η-树中的一条路经上进行,从而显著提高了相似检索的效率.

References:

[1] Bentley, J.L. Multidimensional binary search trees used for associative searching. Communications of the ACM, 1975, 18(9):509~517.

[2] Guttman, A. R-Tree: a dynamic index structure for spatial searching. In: Yormark, B., ed. Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press, 1984. 47~54.

[3] Beckman, N., Kriegel H.P., *et al*. The R*-tree: an efficient and robust access method for points and rectangles. In: Garcia-Molina, H., Jagadish, H.V., eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press, 1990. 322~331.

[4] Berchtold, S., Keim, D.A., Kriegel, H.P. The X-tree: an index structure for highdimensional data. In: Vijayaraman, T.M., Buchmann,A.P., *et al*., eds. Proceedings of the 22th International Conference on VLDB. CA: Morgan Kaufmann Publishers, 1996. 28~39.

[5] White, D.A., Jain, R. Similarity indexing with the SS-tree. In: Proceedings of the 12th International Conference on Data Engineering. 1996. 516~523.

[6] Uhlmann, J. Satisfying general proximity/similarity queries with metric trees. Information Processing Letters, 1991,40:175~179.

[7] Baeza-Yates, R., Cunto, W., Manber U., *et al*. Proximity matching using fixed-queries trees. In: Gochemore, M., Gusfield, D., eds. Proceedings of the 5th Symposium on Combinatorial Pattern Matching. Lecture Notes in Computer Science 807, Springer-Verlag, 1994. 198~212.

[8] Brin, S. New neighbor search in large metric space. In: Dayal, U., Peter, P.M.D., *et al*, eds. Proceedings of the VLDB'95. CA: Morgan Kaufmann Publishers, 1995. 574~584.

[9] Ciaccia, P., Patella, M., Zezula, P. M-Tree: an efficient access method for similarity search in metric space. In: Jarke, M., Karey, M.J., eds. Proceedings of the VLDB'97. CA: Morgan Kaufmann Publishers, 1997. 426~435.

[10] Andrew, P.B., Linda, G.S. A flexible image database system for content-based retrieval. Computer Vision and Image Understanding, 1999,75(1/2):175~195.

[11] Szu, H.H., Hartley, R.L. Fast simulated annealing. Physics Letters A, 1987,122:157~162.

[12] Goldberg, D.E. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989.

A Multidimensional Index Structure for Fast Similarity Retrieval*

FENG Yu-cai, CAO Kui, CAO Zhong-sheng

(*College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074, *China*)

E-mail: ck2896@263.net

http://www.dm2.com.cn

Abstract: A typical example of similarity search is to find the images similar to a given image in a large collection of images. This paper focuses on the important and technically difficult case where each data element is represented by a point in a large metric space. As distance function employed is metric and distance calculations are assumed to be computationally expensive, it is necessary to index data objects in the metric space such that less distance evaluations are performed to support fast similarity queries. Based on the optimal partition method that uses representative points to partition the data space into subsets in a hierarchical manner, a novel distance-based index structure opt-tree and its variant η-tree are proposed. In order to fully support the content-based image retrieval, the optimal strategies for the partition of data space and data redundancy storage, which are called η-optimal partitioning and η-symmetric redundancy storage respectively, are adopted in the η-tree index structure to achieve the high performance of the similarity retrievals. In this paper, the decisions and the algorithms which led to opt-tree and its variant η-tree are discussed in detail, and the experimental results show that this index structure is effective.

Key words: high-dimensional index structure; similarity retrieval; metric space; distance function; distance-based; content-based image retrieval

* Received March 23, 2001; accepted July 18, 2001

Supported by the National High Technology Development 863 Program of China under Grant No.863-511-920-001; the Defence Pre-Research Project of the 'Ninth Five-Year-Plan' of China under Grant No.15.4.1