计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
45(2):310~317, 2008

一种基于 Z 曲线近似 k-最近对查询算法

徐红波[1] 郝忠孝[1,2,3]
[1](哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)
[2](齐齐哈尔大学计算机科学与技术系 齐齐哈尔 161006)
[3](哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)
(x_h_b@tom.com)

An Approximate k-Closest Pair Query Algorithm Based on Z Curve

Xu Hongbo[1] and Hao Zhongxiao[1,2,3]
[1](*College of Computer Science and Technology, Harbin University of Science and Technology, Harbin* 150080)
[2](*Department of Computer Science and Technology, Qiqihar University, Qiqihar* 161006)
[3](*College of Computer Science and Technology, Harbin Institute of Technology, Harbin* 150001)

Abstract The k-closest pairs query is one of the important operations of spatial database. The k-self-closest pair query algorithm based on R*-tree (k-self-CPQ) and brute-force method could achieve better performance in low-dimensional space, but their performances suffer greatly in high-dimensional space, so the reduction of the dimensionality is the key to the problem. Space-filling curve has been extensively used as a mapping scheme from high-dimensional space into linear space, and imposes a linear order of points in the space. It is like a thread that goes through all the points in the space. Hilbert curve, Gray curve, and Z curve are three important space-filling curves. The mapping of Z curve could apply to high-dimensional space easily. Based on Z curve, a method of the reduction of the dimensionality, a notion of minimum grid, and an approximate k-closest pair algorithm under the L_t-metric ($t = 1, \cdots, \infty$) are presented. It uses multiple shifted copies (*ZL-set*) of the data point sorted according to their position along Z curve. Using the length of minimum grid, it optimizes the procedure of scanning *ZL-set*. The algorithm is efficient and simple to implement. Experimental results, obtained by using real and synthetic data sets in high-dimensional space, indicate that its performance is better than that of the k-self-CPQ and brute-force methods, and the quality of approximate k-closest pair is better than that of theoretical analysis.

Key words Z curve; minimum grid; reduction of dimensionality; approximate k-closest pairs

摘 要 k-最近对查询是空间数据库中重要操作之一. 在低维空间中基于 R* 树分枝限界最近对查询算法(k-self-CPQ)和 Brute-Force 算法的查询效率较高,而在高维空间中其性能急剧恶化,降低空间维度成为解决问题的关键. 依据 Z 曲线构造过程,将高维空间分割成大小相等的网格,以此将网格中的点映射到线性空间中. 提出了基于网格划分的降维方法及最小网格概念,给出了基于 Z 曲线近似 k-最近对查询算法. 利用最小网格的边长,算法优化线性扫描过程. 实验结果表明在高维空间中算法性能优于 Brute-Fore 和 k-self-CPQ,且近似 k-最近对质量较好.

关键词 Z 曲线;最小网格;降维;近似 k-最近对

中图法分类号 TP311.13

收稿日期:2006-11-22;**修回日期**:2007-07-19
基金项目:黑龙江省自然科学基金项目(F00-06)

计算几何学广泛地应用到统计学、模式识别、数据挖掘和空间数据库等领域中. 高维空间计算几何学问题逐渐成为空间数据库科研人员的研究热点. 高维空间计算几何学问题包括 k-最近邻、k-最近对、k-最远对和反向最近邻等问题. k-最近对问题是指从点集中找到 k 对距离最近的点. k-最近对查询操作是空间数据库中重要操作之一.

采用分治策略,文献[1]提出时间复杂度为 $O(n \log n)$递归算法,算法将二维空间分割成两个分别包含 $n/2$ 个点的子空间,在子空间上递归调用自身获得局部解,而全局解的求得只需计算分别位于两个子空间相邻区域的点组成的点对,且点对之间的距离不超过局部解中点对距离最大值. 文献[2]对平面点集最近点对分治算法进行了改进,使原来归并时最多需要计算 $3n$ 对点对的距离,改进为最多只需计算 $2n$ 对点对的距离.

基于网格划分方法,文献[3-4]提出时间复杂度为 $O(n \log n + k)$网格划分算法. 网格必须满足两个条件:1)至少存在一个网格,其中至少包含两个点;2)每个网格中至多包含 2^d 个点. 若 δ 为网格边长,则条件 1)保证最近对距离不大于 $d\delta$;条件 2)保证每个点只与常数个点比较. 算法执行过程分为近似阶段和枚举阶段. 在近似阶段中条件 $20^d(k+2n) \leqslant n(n-1)/2$ 必需满足,当 $k=1$ 时 $d \leqslant \log_{20} n$,枚举阶段的复杂度为 $\Omega(5^d k + 4^d n \log n)$.

采用分枝限界方法,文献[5-6]提出两个点集上 k-最近对查询算法(当两点集相等时算法为 k-self-CPQ). 算法将两个点集分别存储在两棵 R* 树中,从根结点开始向下深度优先遍历两棵 R* 树,通过启发式规则和更新策略避免对大量结点的遍历操作.

文献[7]证明基于 R 树及其变种树查询算法的执行时间指数依赖于维数 d,当 $d>10$ 时其执行时间大于 Brute-Force 算法. Brute-Force 依次计算点集中每个点与其他点的距离,从中选择 k 对距离最近的点作为 k-最近对. 当点数较大时点集存储在磁盘中. Brute-Force 进行大量磁盘 I/O 操作,执行时间与点数的平方成正比. 在高维空间中缺少高效查询算法,这被称为"维数灾难"[8].

在线性扫描基础上,本文提出一种基于 Z 曲线近似 k-最近对查询算法. 依据 Z 曲线构造过程,将高维空间分割成大小相等网格,将网格中的点映射到线性空间. 算法在点集上进行最多 $(d+1)$ 次线性扫描,每次扫描计算包含当前点与其 k 个后继点的最小网格,当最小网格的边长大于队列中候选解的最大值时停止当前点的扫描操作,继续扫描下一点.

模拟实验表明在高维空间中算法的执行时间优于 Brute-Force[7] 和 k-self-CPQ[6],且近似 k-最近对的质量较好.

1 基础知识

空间填充曲线是一种将 d 维空间映射成一维空间的方法. 它像一条线穿过高维空间中每个离散单元,且只穿过一次. 它按照线性顺序对这些单元进行编号. 空间填充曲线主要有 Hilbert 曲线、Z 曲线和 Gray 曲线. Hilbert 曲线的数据聚类特性最优,Z 曲线最差. 另一方面,Hilbert 曲线的映射过程最复杂,Z 曲线最简单[9].

空间填充曲线是一种分形曲线. 从整体上看,分形曲线的几何图形是处处不规则的,但在不同尺度上图形的规则性又是相同的,其局部形状又和整体形态相似,它们从整体到局部都是自相似的. 空间填充曲线的构造过程是依次填充曲线的过程.

定义 1. Z 曲线定义为 d 维空间 $\boldsymbol{R}^d$ 与一维空间 $\boldsymbol{I}$ 之间的一一映射,记做 $Z:\boldsymbol{R}^d \rightarrow \boldsymbol{I}$.

若点 $p \in \boldsymbol{R}^d$,则像 $Z(p) \in \boldsymbol{I}$,称像 $Z(p)$ 为点 p 的 Z 值.

定义 2. Z 曲线的阶定义为在曲线构造过程中填充操作的次数.

基本 Z 曲线是大小为 2×2 且阶为 1 的曲线. 为了获得 i 阶 Z 曲线,则将基本 Z 曲线的网格 R_0,R_1,R_2 和 R_3 由 $i-1$ 阶 Z 曲线填充. 1 阶 Z 曲线(见图 1(a))的 4 个网格由 1 阶 Z 曲线填充得到 2 阶 Z 曲线(见图 1(b));1 阶 Z 曲线的 4 个网格由 2 阶 Z 曲线填充得到 3 阶 Z 曲线(见图 1(c)).

Z 曲线的构造过程确定从点坐标到 Z 值之间映射关系. d 维 m 阶 Z 曲线的映射过程:若 $p=((p_{11}\cdots p_{1m})_2,\cdots,(p_{d1}\cdots p_{dm})_2)$,则点 p 的 Z 值 $Z(p)=(p_{11}\cdots p_{d1}\cdots p_{1m}\cdots p_{dm})_2$. 图 2 给出 2 维 3 阶 Z 曲线的映射过程.

定义 3. 设 $p=(p_1,\cdots,p_d)$,$q=(q_1,\cdots,q_d)$,点 p 到点 q 的距离函数定义为 $Dist_t(p,q)=(|p_1-q_1|^t+\cdots+|p_d-q_d|^t)^{1/t}$ $(1 \leqslant t < \infty)$,当 $t=\infty$ 时 $Dist_\infty(p,q)=\max\limits_{1\leqslant i\leqslant d}\{|p_i-q_i|\}$.

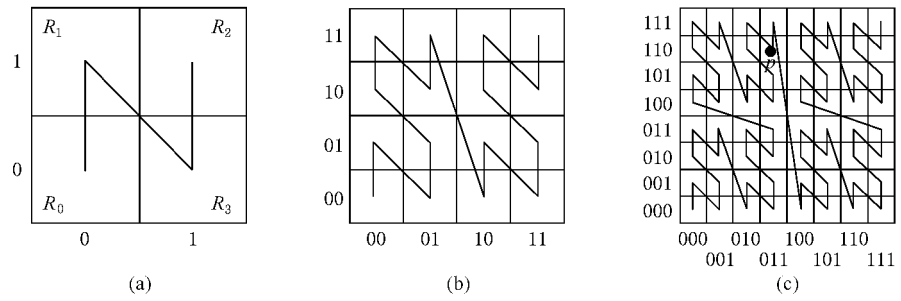

Fig. 1 Z curves of order 1, 2 and 3. (a) Z curve of order 1; (b) Z curve of order 2; and (c) Z curve of order 3.

图 1 1,2 和 3 阶 Z 曲线.(a) 1 阶 Z 曲线;(b) 2 阶 Z 曲线;(c) 3 阶 Z 曲线

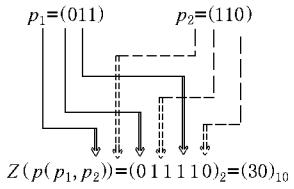

Fig. 2 Bit-interleaving operation of $p(3,6)$.

图 2 点 $p(3,6)$的位交插操作

距离函数是衡量空间中两点之间距离的标准,当 $t=1$ 时其表示曼哈顿距离;当 $t=2$ 时其表示欧几里德距离;当 $t=\infty$ 时其表示车比雪夫距离.

2 问题定义

定义 4. 设点集 $P(|P|=n)$,$1\leqslant k\leqslant n(n-1)/2$,$k$-最近对定义为 $CP_k(P)=\{\langle p_1,q_1\rangle,\cdots,\langle p_k,q_k\rangle\mid(p_1,\cdots,p_k,q_1,\cdots,q_k\in P)\wedge(\forall p,q\in P-\{p_1,\cdots,p_k,q_1,\cdots,q_k\})\ Dist_t(p,q)\geqslant Dist_t(p_k,q_k)\geqslant\cdots\geqslant Dist_t(p_1,q_1)\}$.

定义 5. Z 网格定义为 d 维 m 阶 Z 曲线在第 k $(1\leqslant k\leqslant m)$次填充中将空间 $\boldsymbol{R}^d=[0,1)^d$ 分割成的超立方体,几何形状由主对角线上两个端点 $\boldsymbol{S}=(s_1,\cdots,s_d)$和 $\boldsymbol{T}=(t_1,\cdots,t_d)$确定,记做 $Z\text{-}Grid=\langle S,T\rangle$,边长为 $r=t_i-s_i=2^{-k}(1\leqslant i\leqslant d)$.

在二维空间中,m 阶 Z 曲线将边长为 1 的正方形分割成 $2^{2k}(1\leqslant k\leqslant m)$个大小相等的网格,每个网格的边长 $r=2^{-k}$;Z 曲线将网格中的点连接成链,相邻网格的连接是通过前一网格尾结点与后一网格首结点连接形成.在 d 维空间中,m 阶 Z 曲线将边长为 1 的超立方体分割成 $2^{dk}(1\leqslant k\leqslant m)$个大小相等的网格,每个网格的边长 $r=2^{-k}$.每个网格包含一段且仅一段连续 Z 曲线.

定义 6. 设 d 维点 $v(j)=(j/(d+1),\cdots,j/(d+1))$,点集 $P(|P|=n$ 且 $(\forall p\in P)\ p\in[0,1)^d)$,$Z$ 表定义为 $Z\text{-}List_j=\{\langle p_1,Z(p_1+v(j))\rangle,\cdots,\langle p_i,Z(p_i+v(j))\rangle,\langle p_{i+1},Z(p_{i+1}+v(j))\rangle,\cdots,\langle p_n,Z(p_n+v(j))\rangle\mid(p_1,\cdots,p_n\in P)\wedge(Z(p_i+v(j))\leqslant Z(p_{i+1}+v(j)))(i=1,\cdots,n-1)\}\ (j=0,\cdots,d)$.

Z 表中第 i 个元素记为 $Z\text{-}List[i]$;Z 表的集合记为 $ZL\text{-}set=\{Z\text{-}List_0,Z\text{-}List_1,\cdots,Z\text{-}List_d\}$. $ZL\text{-}set$ 将 d 维空间的区域从 $[0,1)^d$ 扩展成 $[0,2)^d$.

定义 7. 点 p 和点 q 的最小网格定义为包含点 p 和点 q 最小的 Z 网格,记做 $MinGrid(p,q)$.

同时 $MinGrid(p,q)$代表包含点 p 和点 q 的最小网格的边长.

例 1. 二维空间中点的分布如图 3 所示.最小网格边长的计算过程:依据曲线阶 m,将点坐标转换成长度为 m 二进制串;对每维二进制串从左向右进行扫描,比较对应位置上的值是否相等,若相等则继续扫描,否则终止扫描过程,记录相等次数.当所有维扫描完后,求相等次数的最小值 Min,则最小网格的边长为 $1/2^{Min}$.具体计算过程如图 4 所示.

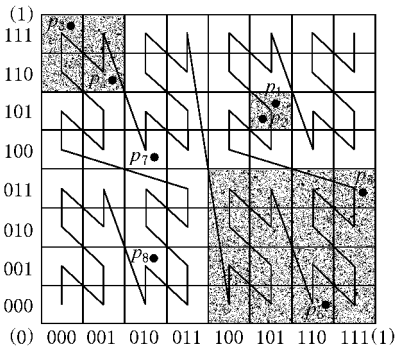

Fig. 3 Minimum grids of Z curve of dimension 2 and order 3.

图 3 2 维 3 阶 Z 曲线的最小网格(阴影区域)

从 Z 曲线构造过程知 Z 表 $Z\text{-}List_j$ 具有两条性质:1)若将 $\boldsymbol{R}^d$ 分成大小相等网格,则每个网格中包含一段且仅一段 Z 表片段;2)若 $R_i(j)=MinGrid(Z\text{-}List[i],Z\text{-}List[i+j])(1\leqslant i<i+j\leqslant n)$,则当 i 取特定值时 $R_i(j)$为自变量 j 的单调递增函数.

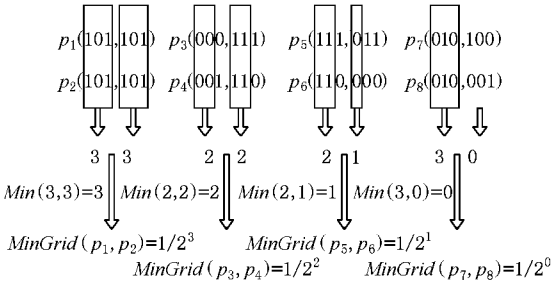

Fig. 4 Procedure of calculating $MinGrid(p,q)$.

图 4 最小网格边长的计算过程

3 近似 k-最近对查询算法

定义 8. 优先级队列定义为 $Q=\{\langle p,q,r\rangle \mid p,q\in P,\ r=MinGrid(p,q)\}$,且元素依 r 值升序排序.

初始化操作 $Init(Q)$ 动态分配存储 k 个元素的堆内存空间;投影操作 $P(Q)=\{\langle p,q\rangle \mid \langle p,q,r\rangle \in Q\}$;最大值操作 $Max(Q)=\max\{r \mid \langle p,q,r\rangle \in Q\}$;更新操作 $Update(\langle p,q,r\rangle,Q)$ 分为 3 种情况:1)当 $\langle p,q\rangle \in P(Q)$ 时,r 取较小值;2)当 $\langle p,q,r\rangle \notin Q$ 且 Q 未满时,将 $\langle p,q,r\rangle$ 插入到 Q 中相应位置上;3)当 $\langle p,q,r\rangle \notin Q$ 且 Q 已满时,若 $r<Max(Q)$,则将 $\langle p,q,r\rangle$ 插入到 Q 中相应位置上,否则,空操作. 队列 Q 存储在算法执行过程中扫描到的候选最近对.

算法的基本思想:对 Z 表中每个点,依次计算包含当前点和其 k 个后继点的最小网格的边长 r. 在扫描过程中,若 $r>Max(Q)$,则停止扫描当前点,否则,继续扫描其后继点.

算法 1. 近似 k-最近对查询算法 $AKCP(P,k)$.

输入:点集 $P=\{p_1,\cdots,p_n\}$、最近对数 k.

输出:近似 k-最近对 $ACP_k(P)$.

```
begin
① Init(ZL-set);
  Init(Q);
② for t=0 to d do
③   for i=1 to n-1 do
      j=i+1;
      p=Z-List_t[i].point;
④     while(j≤min(i+k,n))
        q=Z-List_t[j].point;
        r=MinGrid(p,q);
⑤       if(r<Max(Q))
          then Update(⟨p,q,r⟩,Q);
          else break;
        j=j+1;
      endwhile
    endfor
  endfor
  return P(Q);
end
```

定理 1. AKCP 近似地求出 k-最近对,时间复杂度为 $O(dnk)$,空间复杂度为 $O(dn)$.

证明. 终止性. 步骤①初始化表集 $ZL\text{-}set$ 和队列 Q. 步骤②执行 Z 表扫描,其中:t 为 Z 表指针,i 为当前点指针,j 为后继点指针. 步骤④扫描当前点 p 的 k 个后继点 q,计算最小网格 $MinGrid(p,q)$ 的边长 r. 若 r 小于队列 Q 的最大值(步骤⑤),则终止当前点的扫描,否则,继续扫描后继点($j=j+1$),同时更新队列. 故队列中存储算法执行过程中查询到的候选最近对.

近似性. 近似 k-最近对的质量由定理 2 保证.

分析. 步骤②执行 $(d+1)$ 次,步骤③执行 $(n-1)$ 次,步骤④最多执行 k 次,故时间复杂度为 $O(dnk)$;$ZL\text{-}set$ 包含 $(d+1)$ 个 Z 表,每个 Z 表包含 n 个点,故空间复杂度为 $O(dn)$. 证毕.

首次执行算法时需要初始化表集 $ZL\text{-}set$. 在后续查询中 $ZL\text{-}set$ 无需重新建立,且支持点集的动态变化.

4 误差分析

定义 9. 设 $\boldsymbol{p}=(p_1,\cdots,p_d)\in \boldsymbol{R}^d$,$p\in Z\text{-}Grid$,$r$ 为 $Z\text{-}Grid$ 的边长,若 $(ar\leqslant \bmod(p_i,r)<(1-a)r)\vee(2ar\leqslant \bmod(p_i+ar,r))$ $(i=1,\cdots,d,\ 0\leqslant a<0.5)$,则称点 p 是 a 中心于 $Z\text{-}Grid$.

引理 1[10]. 设 d 为偶数,$r=2^{-m}$ $(m\in N)$,$p\in[0,2)^d$,存在 $j\in\{0,\cdots,d\}$,使得 $p+v(j)$ 是 $1/(2d+2)$ 中心于 $Z\text{-}Grid$.

证明. 反证法. 假设 $p+v(j)$ $(j=0,\cdots,d)$ 均不是 $1/(2d+2)$ 中心于 $Z\text{-}Grid$. 由定义 9 知对每个 j,存在对应值 $i(j)\in\{1,\cdots,d\}$,使得

$$\bmod(p_{i(j)}+j/(d+1)+r/(2d+2),r)<r/(d+1) \quad (1)$$

将式(1)左右两边同时乘以 $(d+1)/r$,式(1)化为

$$\mathrm{mod}((d+1)2^m p_{i(j)} + 2^m j + 1/2, d+1) < 1 \quad (2)$$

j 从 $\{0,\cdots,d\}$ 中取值,而 $i(j)$ 从 $\{1,\cdots,d\}$ 中取值,由鸽巢原理可知,存在两个不同的数 j 和 j',使得 $i(j)=i(j')$. 设 $z=(d+1)2^m p_{i(j)}+1/2=(d+1)2^m p_{i(j')}+1/2$,式(2)化为 $\mathrm{mod}(z+2^m j, d+1)<1$ 和 $\mathrm{mod}(z+2^m j', d+1)<1$ 成立,即 $2^m j \equiv 2^m j' (\mathrm{mod}(d+1))$ 成立,2^m 与 $d+1$ 互为素数,故 $j=j'$ 成立,与假设矛盾. 证毕.

引理 2. 设 $0<\varepsilon<1/(2d+2)$,$C_\varepsilon(q)$ 是以点 q 为中心且边长为 2ε 的超立方体,存在 $j\in\{0,\cdots,d\}$,$C_\varepsilon(q+v(j))$ 包含在边长为 $r(r/(4d+4)\leqslant\varepsilon<r/(2d+2))$ 的 Z-$Grid$ 中.

证明. 当 r 满足 $r/(4d+4)\leqslant\varepsilon<r/(2d+2)$ 时,r 惟一确定. 若将 $[0,2)^d$ 分割成大小相等且边长为 r 的 Z-$Grid$,则由引理 1 知,存在 $j\in\{0,\cdots,d\}$,$q'=q+v(j)$ 是 $1/(2d+2)$ 中心于 Z-$Grid$. 由定义 9 知 $\mathrm{mod}(q_i', r)\geqslant ar=r/(2d+2)$ $(i=1,\cdots,d)$,故 $C_{r/(2d+2)}(q')$ 包含在 Z-$Grid$ 中. 由 $\varepsilon<r/(2d+2)$ 知,$C_\varepsilon(q')$ 也包含在该 Z-$Grid$ 中,引理成立. 证毕.

定理 2. $\delta_k^*\leqslant\delta_k\leqslant d^{1/t}(4d+4)\delta_k^*$.

证明. 设 δ_k 为队列 Q 的 k 个点对中两点之间距离最大值,δ_k^* 为 CP_k 的 k 个点对中两点之间距离最大值,显然,$\delta_k^*\leqslant\delta_k$. 设 $\boldsymbol{p}=(p_1,\cdots,p_d)$,$\boldsymbol{q}=(q_1,\cdots,q_d)\in[0,1)$,由 $|p_i-q_i|\leqslant 1(i=1,\cdots,d)$ 知 $Dist_t(p,q)=(|p_1-q_1|^t+\cdots+|p_d-q_d|^t)^{1/t}\leqslant(1^t+\cdots+1^t)^{1/t}=d^{1/t}$,故点集 P 中任意点对之间距离小于 $d^{1/t}$,即 $d^{1/t}\geqslant\delta_k$. 若 $\delta_k^*\geqslant 1/(4d+4)$,即 $(4d+4)\delta_k^*\geqslant 1$,则 $d^{1/t}(4d+4)\delta_k^*\geqslant\delta_k$,下面证明当 $\delta_k^*<1/(4d+4)$ 时命题成立. 设 $CP_k(P)=\{\langle s_1,q_1\rangle,\cdots,\langle s_k,q_k\rangle\}$,$x_i=Dist_t(s_i,q_i)$ $(i=1,\cdots,k)$,$\delta_k^*=\max\{x_1,\cdots,x_k\}$. 由 $\delta_k^*<1/(4d+4)$ 知 $0<x_i\leqslant\delta_k^*<1/(4d+4)<1/(2d+2)$ $(i=1,\cdots,k)$. 由引理 2 知,当 $0<x_i<1/(2d+2)$ 时存在 $j_i\in\{0,\cdots,d\}$,$C_{xi}(s_i+v(j_i))\subseteq Z_i-Grid$,其中:$r_i/(4d+4)\leqslant x_i<r_i/(2d+2)$,即 $r_i\leqslant x_i(4d+4)$,且 $q_i+v(j_i)\in Z_i$-$Grid$. 设 $r_{\max}=\max\{r_1,\cdots,r_k\}$,假设 $Max(Q)\leqslant r_{\max}$,故队列 Q 中任意点对之间的距离 $Dist_t(p,q)\leqslant((r_{\max})^t+\cdots+(r_{\max})^t)^{1/t}=d^{1/t}r_{\max}\leqslant d^{1/t}(4d+4)\delta_k^*$,即 $\delta_k\leqslant d^{1/t}(4d+4)\delta_k^*$,下面证明当 $Max(Q)\leqslant r_{\max}$ 时命题成立. 设 $s_i'=s_i+v(j_i)$,$q_i'=q_i+v(j_i)$,设 Z-$List_{ji}$ 中点 q_i' 为点 s_i' 的后继点,存在两种情况:1)在算法执行过程中所有的点对 (s_i', q_i') $(i=1,\cdots,k)$ 均被扫描过,显然 $Max(Q)\leqslant r_{\max}$;2) Z-$List_{ji}$ 上存在点对 (s_i', q_i') $(1\leqslant i\leqslant k)$ 没有被扫描过. ①算法扫描了点 s_i' 的 k 个后继点且点 q_i' 不在其中,由 Z 表的性质 1 知 k 个点对必被包含在 Z_i-$Grid$ 中,故点对的 r 值小于 r_i;②在扫描点 q_i' 之前,算法扫描点对 (s_i', t),(s_i', t) 的 r 值大于 $Max(Q)$,算法停止对点 s_i' 的后继点的扫描,此时,$Max(Q)<r\leqslant r_i\leqslant r_{\max}$. 综上所述,命题成立. 证毕.

5 实验结果

实验环境:2.6GHz 奔腾Ⅳ处理器、256MB 内存、40GB 硬盘、Windows XP 操作系统、Visual C++.net 2003 编程工具和 Microsoft Access 2002 数据库管理系统.

测试数据选取从 Corel 图片库[①] 中提取出的高维特征向量所组成的数据文件:ColorHistogram(CH) 和 LayoutHistogram(LH)($d=32$,$n=10000$);由随机函数产生、满足均匀分布、维数为 64,128,256,512,640,768 的点组成的数据文件,数据量从 1000~5000.

距离函数选择欧几里德距离($t=2$).

查询结果的质量是评价近似算法的重要指标. 标准 ε 从距离上衡量近似 k-最近对的质量.

定义 10. 设 $CP_k(P)$ 为 k-最近对,$ACP_k(P)$ 为近似 k-最近对,ε 定义为

$$\varepsilon=\left(\sum_{(p,q)\in ACP_k(P)} Dist_t(p,q)-\sum_{(p,q)\in CP_k(P)} Dist_t(p,q)\right)\Big/\sum_{(p,q)\in CP_k(P)} Dist_t(p,q).$$

实验 1. 对曲线阶 m 的不同取值,AKCP 在两份数据文件上分别进行 1000-最近对查询,统计 ZL-set 创建时间(如图 5 所示)、执行时间(如图 6 所示)、近似 k-最近对的质量 ε(如表 1 和表 2 所示).

ZL-set 可以重复使用,且支持点集的动态变化,如点的插入、删除和更新操作等. Z-$List$ 通过数据库中的表进行存储,且在 Z 值字段上建立索引,实

① http://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.data.html

现对 Z 表快速线性遍历.

查询 CH 和 LH 上 1000-最近对,Brute-Force 的执行时间为 453s 左右,k-self-CPQ 的执行时间为 686s 左右,而 AKCP 的执行时间在 100s 左右(如图 6 所示).

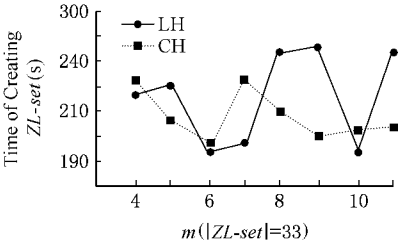

Fig. 5 Relation of order of curve and time of creating *ZL-set*.

图 5 曲线的阶与创建 *ZL-set* 所需时间之间的关系

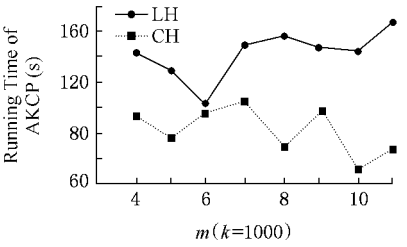

Fig. 6 Relation of order of curve and running time of AKCP.

图 6 曲线的阶与 AKCP 的执行时间之间的关系

从表 1 和表 2 知,曲线的阶越大,近似 k-最近对的质量 ε 越小;当曲线的阶取到一定值时,近似 k-最近对和 k-最近对之间的距离相差非常小. 在大多数应用本身就是一种'相似'查询的情况下,近似 k-最近对的质量可以满足实际要求.

Table 1 Quality of AKCP(P) Based on CH

表 1 基于 CH 的近似 k-最近对的质量

m	k									
	1	2	3	4	5	10	20	30	40	50
4	29.0	26.7	22.5	22.1	22.6	19.6	19.7	18.3	16.8	16.1
5	4.82	3.41	3.00	3.00	3.12	3.01	1.95	1.73	1.93	2.01
6	0.772	1.26	0.963	0.888	0.687	0.780	0.580	0.553	0.583	0.476
7	0.183	0.116	0.415	0.366	0.355	0.419	0.520	0.493	0.553	0.462

Table 2 Quality of AKCP(P) Based on LH

表 2 基于 LH 的近似 k-最近对的质量

m	k									
	1	2	3	4	5	10	20	30	40	50
8	22.8	20.8	13.2	5.77	5.27	2.19	1.79	0.928	0.803	0.746
9	12.5	9.41	9.90	3.80	2.04	0.413	0.116	0.402	0.410	0.569
10	5.45	2.53	1.19	0.458	0.008	0.413	0.116	0.402	0.410	0.569
11	0.151	0.00	0.00	0.014	0.008	0.413	0.116	0.402	0.410	0.569

实验 2. 在人工合成数据上比较 AKCP 同 Brute-Force、k-self-CPQ 在执行时间上的差异,结果见表 3、表 4 和表 5,其中:M 表示 R^* 树中结点分枝数. 从表 3 知,AKCP 的执行时间最短,随着参数 d 的增加,执行时间保持在 5.4s 左右,原因在于算法使用高维点的 32 维坐标值近似代替高维点. 由 ε 值可知,近似 k-最近对的质量较好. k-self-CPQ 的执行时间递增较快;从表 4 知,随着参数 n 的增加,k-self-CPQ 和 Brute-Force 的执行时间急剧增大,而 AKCP 的执行时间线性增长;从表 5 知,随着参数 k 的增加,AKCP 的执行时间变化不大,原因在于算法中步骤⑤能够提前结束对后继点的扫描. 随着算法的执行,队列 Q 中点对的最小网格的边长逐渐变小,后继点的扫描数远远小于 k.

Table 3 Relation of Running Time and d

表 3 参数 d 与执行时间的关系

$t(s)$	d					
	64	128	256	512	640	768
k-self-CPQ	12.51	24.45	49.1	97.62	103.6	156.2
Brute-Force	4.672	7.469	8.218	9.938	10.17	11.81
AKCP	5.438	5.375	5.391	5.5	5.422	5.578
ε	0.3915	0.2373	0.1535	0.1068	0.2520	0.0894

Note: $n=1000$, $k=100$, $M=50$, $m=16$, $|ZL\text{-}set|=33$

Table 4 Relation of Running Time and n

表 4 参数 n 与执行时间的关系

$t(s)$	n				
	1000	2000	3000	4000	5000
k-self-CPQ	100.4	401.4	929.5	1630	2534
Brute-Force	10.12	41.67	90.34	158.7	248.1
AKCP	5.453	10.53	15.68	21.41	25.87
ε	0.1101	0.1208	0.1308	0.1288	0.1307

Note: $d=512$, $k=100$, $M=50$, $m=16$, $|ZL\text{-}set|=33$

Table 5 Relation of Running Time and k

表 5 参数 k 与执行时间的关系

$t(s)$	k				
	50	100	200	500	1000
k-self-CPQ	154.4	154.7	154.5	155.1	155.7
Brute-Force	11.81	11.7	11.79	11.75	11.84
AKCP	5.422	5.438	5.609	5.75	6.04
ε	0.0956	0.0878	0.083	0.772	0.0748

Note: $d=768$, $n=1000$, $M=50$, $m=16$, $|ZL\text{-}set|=33$

6 小 结

本文提出一种基于 Z 曲线近似 k-最近对查询算法,用来解决已有算法应用到高维空间时查询效率急剧恶化问题. 实验结果表明 AKCP 克服了"维数灾难"问题,算法可行且有效. 下一步工作是将其他空间填充曲线应用到解决 k-最近对问题当中,研究其他曲线与 Z 曲线之间性能差异.

参 考 文 献

[1] J L Bentley, M I Shamos. Divide-and-conquer in multidimensional space [C]. In: Proc of the 8th Annual ACM Symp on Theory of Computing. New York: ACM Press, 1976. 220-230

[2] Zhou Yulin, Xiong Pengrong, Zhu Hong. An improved algorithm about the closest pair of points on plane set [J]. Journal of Computer Research & Development, 1998, 35(10): 956-960 (in Chinese)
(周玉林,熊鹏荣,朱洪. 求平面点集最近点对的一个改进算法[J]. 计算机研究与发展, 1998, 35(10): 956-960)

[3] H P Lenhof, M Smid. Enumerating the k closest pairs optimally [C]. In: Proc of the 33rd IEEE Symp on Foundation of Computer Science. Piscataway, NJ: IEEE Press, 1992

[4] T Chan. On enumerating and selecting distance [C]. In: Proc of the 14th Annual ACM Symp on Computational Geometry. New York: ACM Press, 1998. 279-286

[5] A Corral, Y Manolopoulos, Y Theodoridis, *et al*. Closest pair queries in spatial databases [C]. In: Proc of the 2000 ACM Int'l Conf on Management of Data. New York: ACM Press, 2000. 189-200

[6] A Corral, Y Manolopoulos, Y Theodoridis, *et al*. Algorithms for processing k-closest-pair queries in spatial databases [J]. Data & Knowledge Engineering, 2004, 49(1): 67-104

[7] R Weber, H Schek, S Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces [C]. In: Proc of the 24th Int'l Conf on Very Large Data Bases. Berlin: Springer-Verlag. 194-205

[8] K Beyer, J Goldstein, R Ramakrishnan, *et al*. When is "nearest neighbor" meaningful? [C]. In: Proc of the 7th Int'l Conf on Database Theory. Berlin: Springer-Verlag, 1999. 217-235

[9] S Shekhar, S Chawla. Spatial Databases: A Tour [M]. Englewood Cliffs, NJ: Prentice Hall, 2003

[10] T M Chan. Approximate nearest neighbor queries revisited [C]. In: Proc of the 13th ACM Symp on Computer Geometry. New York: ACM Press, 1997. 352-358

Xu Hongbo, born in 1980. Received his B. S. and M. S. degrees from the College of Computer Science and Technology, Harbin University of Science and Technology in 2002 and 2005 respectively. Ph. D. candidate in computer application from HUST. His main research intersects include high-dimensional index structure and spatial query algorithm.

徐红波,1980 年生,博士研究生,主要研究方向为高维空间索引结构和空间查询算法.

Hao Zhongxiao, born in 1940. He is professor and Ph. D. supervisor. His main research interests include database theory and application.

郝忠孝,1940 年生,教授,博士生导师,主要研究方向为数据库理论及应用(haozx@hrbust.edu.cn).

Research Background

The work of this paper is supported by the Natural Science Foundation of Heilongjiang Province of China under grant No. F00-06. The k-closest pairs query is one of the important operations of spatial database. The k-self-closest pairs query algorithm based on R^*-tree (k-self-CPQ) and brute-force method could achieve better performance in low-dimensional space, but their performance suffers greatly in high-dimensional space. So the reduction of the dimensionality is the key to the problem. Space-filling curve has been extensively used as a mapping scheme from high-dimensional space into linear space, and imposes a linear order of points in the space. It is like a thread that goes through all the points in the space. Hilbert curve, Gray curve, and Z curve are three important space-filling curves. The mapping of Z curve is the simplest among them. Using Z curve, this paper presents a notion of minimum grid, and an approximate k-closest pairs query algorithm. Using the length of minimum grid, it optimizes the procedure of linear scan. Experiment results show that its performance is better than that of k-self-CPQ and brute-force in high-dimensional dimension.

中国计算机学会
全国第 3 届语义 Web 与本体论学术研讨会(SWON2008)
征文通知
(2008 年 9 月 19~21 日　西安交通大学　西安)

语义 Web 吸取人工智能、信息论、哲学、逻辑和计算复杂性等学科的研究成果,力图对 Web 上信息的表示和获取方式进行改进,以解决目前使用 Web 时存在的瓶颈. 语义 Web 的核心思想是通过增加一些语义信息,使得计算机能参与到自动处理 Web 信息的过程,并为实现智能化的 Web 应用提供必要的技术基础.

全国语义 Web 与本体论学术研讨会(SWON)是中国计算机学会暨电子政务与办公自动化专委会主办的系列会议. SWON 2008 会议将于 2008 年 9 月在西安召开. 会议目的是为语义 Web 学术界和工业界提供一个交流平台,反映国际国内关于语义 Web 的最新研究成果和进展. 会议录用论文中将选择部分以英文方式继续由《东南大学学报(英文)》(EI 源刊)正刊出版,其余论文将由核心期刊《计算机科学》专刊和清华大学出版社出版. 会议期间除进行会议论文交流外,还将邀请著名学者作特邀报告,并继续评选大会优秀学生论文.

征文范围(包括但不限于)

语义 Web 语言与工具	语义 Web 知识表示	语义 Web 知识管理	语义 Web 推理
语义 Web 服务	语义 Web 安全	语义 Web 挖掘	语义信息标注
语义检索和查询	本体学习与元数据生成	本体存储与管理	语义集成和映射

来稿要求

① 本次会议主要通过网上投稿,尽量不要通过 Email 投稿,拒收纸质稿件. 严禁一稿多投.

② 本次会议只接受英文稿,一般不超过 6000 字,为了便于出版论文集,来稿必须附中英文摘要、关键词、资助基金与主要参考文献,注明作者及主要联系人姓名、工作单位、详细通信地址(包括 Email 地址)与作者简介. 稿件要求采用 WORD 或 PDF 格式,按《东南大学学报》(英文版)格式编排.

联系信息

① 投稿地址:http://wisa2008.xjtu.edu.cn;联系人:东北大学　王国仁(swon2008@mail.neu.edu.cn)

②大会网站:http://wisa2008.xjtu.edu.cn,http://cse.seu.edu.cn/pcegoa/

③ 会务情况:西安交通大学　齐勇　赵天海 (wisa2008@mail.xjtu.edu.cn)

重要日期

征文截止日期:2008 年 4 月 1 日

录用通知发出日期:2008 年 4 月 25 日

正式论文提交日期:2008 年 5 月 15 日

会议召开日期:2008 年 9 月 19~21 日