计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
44(11):1980~1985, 2007

利用分区和距离实现高维空间快速 KNN 查询

梁俊杰[1,2]　王长磊[3]
[1](湖北大学数学与计算机科学学院　武汉　430062)
[2](华中科技大学计算机科学与技术学院　武汉　430074)
[3](湖北省公安厅行动技术总队　武汉　430072)
(ljjhubu@163.com;ljj@dameng.com)

Indexing Bit-Code and Distance for Fast KNN Search in High-Dimensional Spaces

Liang Junjie[1,2] and Wang Changlei[3]
[1](*School of Mathematics & Computer Science, Hubei University, Wuhan* 430062)
[2](*College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan* 430074)
[3](*Action and Technology Department of Public Security of Hubei Province, Wuhan* 430072)

Abstract In the recent literature, a variety of index structures have been proposed to facilitate high-dimensional KNN queries, among which the techniques of approximate vector presentation and one dimensional transformation can efficiently break the curse of dimensionality. Based on the two techniques above, a novel high-dimensional index, called bit-code and distance based index (BD) is proposed. On the basis of the data distribution in high-dimensional spaces, the BD partitions the surface of dimensionality in a special way, such that all the data points are divided into lots of partitions according to some cluster centroids. By the definitions of bit code and transformation function, a high-dimensional vector can be first approximately represented and then transformed into a one-dimensional vector, the key managed by a special B^+-tree. During the process of KNN search, the BD enables two levels filtering: the transformation function prunes away points that do not share similar distance ranges, while the bit code component filters away points by the lower bounded distance. A fast algorithm is presented for KNN search, by which one can greatly reduce the number of distance computations and the cost of the tree search. By using both synthetic and real data, the results of extensive experiments demonstrate that the BD outperforms the existing index structures for KNN search in high-dimensional spaces.

Key words high-dimensional vector space; KNN search; bit code; approximate vector; index structure

摘　要　在高维空间 KNN 查询算法中,近似向量和一维转换表示法能有效克服维数灾难,结合这两种思想,提出一种基于区位码和距离的索引结构(BD)以实现快速 KNN 查询. 根据高维空间向量分布特点,合理分区使得大量分布在空间表面的点尽可能地划分到不同的分区中,提高检索剪枝效率. 引入区位码概念和转换函数,将高维向量近似表示并转换为一维数值形式,组织成 B^+ 树索引. 利用快速 KNN 查询算法,实现两层过滤,缩小搜索范围,降低树搜索代价. 采用模拟数据和真实数据,大量实验验证了 BD 比其他同类索引具有更高的检索效率.

关键词　高维向量空间;KNN 查询;区位码;近似向量;索引结构

中图法分类号　TP391.3

收稿日期:2006-05-25;**修回日期**:2007-07-03
基金项目:国家"八六三"高技术研究发展计划基金项目([2005]555)

近几十年来,在很多领域出现了高维数据应用,例如数据仓库和基于内容的多媒体信息检索,数据以高维向量的形式存在,有的维度甚至高达几百或几千维,并且数据量通常也较大. KNN 查询(K-nearest neighbor search)是这类应用中常见的检索方式,给定查询对象,要求在高维空间中查找出与查询对象距离最近的 K 个点[1]. 在实际查询过程中,例如在电子商务、医疗诊断等领域,用户不仅对检索精度有一定的要求,对检索响应时间也会提出很高的要求,这样必须保证系统有较高的检索效率,因此有必要研究高维向量空间的快速 KNN 查询方法.

在高维向量空间中实施 KNN 查询,影响检索效率提高的一个重要因素是高维向量距离计算的代价相当大,为了减少距离计算,目前已有很多文献对此提出了不同的解决方法,其中以通过分割数据或空间建立索引为主要思路[2-3],但是这类方法往往不可避免地遭遇'维数灾难'困扰,不能很好地应用于高维(超过几十维)数据空间检索[4]. 因此,最近几年提出一种新思路:简化高维向量表示形式. 例如利用近似向量代替高维向量的表示方法,或者将高维向量转换为一维表示形式等,降低维度增加带来的不利影响. 但是,目前提出的这类索引结构都只是采用其中一种思想而设计的,并且存在不同方面的问题,影响了在实际中的应用.

本文结合近似向量表示法和一维向量转换法两者思想,提出一种新的索引结构实现高维空间的快速 KNN 检索,称为基于区位码和距离的索引结构(bit-code and distance based index, BD). 在构造 BD 索引结构时,充分考虑到高维空间数据分布特点,随着维度的增加,空间表面体积所占比重增大,分布在表面的点越来越多,而空间内部的点越来越少,从而对向量空间进行合理分区,这样可以将大量分布在空间表面的点尽可能地划分到不同的分区中,便于检索时剪枝处理,提高 KNN 查询速度.

1 相关工作

高维空间的复杂性大大影响了检索效率的提高,目前已有很多文献都对这一问题的解决提出了不同的方法,大致可以分为 3 类:

1) 数据空间分割法. 如 R-树、X-树、SR-树和 TV-树等. 这些索引树在维数较低的情况搜索效率较好,一旦维数超过一定范围时,检索效率会急剧下降,甚至不如顺序扫描,这就是所谓的'维数灾难'现象. 主要是高维向量带来的影响,随着维度的增加,最近邻的距离会很快接近最远邻的距离[5-6],使得这类方法在查询时需要访问所有分割块.

2) 近似向量表示法. 通过采用简单向量近似地表示原高维向量,达到简化搜索空间的目的[4,7-8],如与本文索引有关的两个例子 VA-file[4]和 BID[8]. VA-file 利用数据压缩加速顺序扫描,基本思想是通过将数据空间分割成 2^b 个矩形单元格,对每个单元格采用一个长度为 b 个位(bit)的字符串标识,这样所有的高维点向量就可以采用其所属的单元格标识符近似表示. 因此,VA-file 的 KNN 搜索就只需在这些近似向量中进行顺序扫描,但是 VA-file 的搜索效率受数据分布影响较大. 和 VA-file 不同的是,BID 利用与参考点的相对位置关系将高维向量近似表示成位码字符串,查询时根据查询点和某个点之间的位码不相同位数,判断该点是否为候选点,实现主存环境的近似 KNN 查询. 虽然 BID 查询过程不需要距离计算,检索速度较快,但是不能保证较高的 KNN 检索精度.

3) 一维转换表示法. 将高维向量转换成一维表示形式,金字塔技术(pyramid-technique, PT)[9]和 iDistance[10]体现了这种思想. 金字塔技术将 d 维数据空间划分成 $2d$ 个金字塔,根据每个点所在的金字塔序号 i 以及距离顶点的高度 h,通过函数($key = i + h$)转换得到它的关键字值(key),并将它们组织成 B^+ 树索引结构. 金字塔技术的检索效率受维度影响不大,但是由于它只适用于矩形范围检索,限制了在实际中的应用. iDistance 利用与参考点的距离将高维点向量转换为一维表示,由于提供了灵活的空间划分和参考点选择方式,使得 iDistance 具有较好的检索性能. 但是,由于转换方式引起的不同点可能具有相同一维数值,并且随着维度的增高(>30),这一现象越来越明显,因此检索过程会引入过多的误中点,对这些点的距离计算大大降低检索效率.

2 基于区位码和距离的索引结构

基于区位码和距离的索引结构综合利用近似向量表示法和一维向量转换法的思想,首先将数据空间划分成 $2^{d'}$ ($d' < d$)个分区,每个分区用一个位码字符串表示,则任意高维点向量都可以近似表示为所属分区的位码字符串;然后利用与参考点的距离,通过转换函数进一步将高维向量表示成一维数值

形式(key);最后将所有的 key 值组织成 B^+ 树索引.因此,BD 索引同时具有近似向量表示法和一维向量转换法特点,key 值中既包含近似向量(区位码)信息,又包含一维向量(距离)信息.在进行 KNN 检索时,首先利用查询点所在位置确定初始查询范围;然后根据区位码快速将与查询范围不相交的分区剪枝;最后对剩下的分区,利用一维向量的排序和过滤思想,进一步排除非候选点.因此,利用 BD 索引能够快速缩小需要搜索的范围,尽可能减少高维向量距离计算次数,从而实现快速 KNN 查询.

2.1 向量分区

对于 d-维向量空间,根据数据分布可以划分成几个不同的分区,每个分区用一个区位码表示.为了实现分区,我们可以选择一个参考点,根据与参考点的位置关系对数据空间在某些维 $d'(<d)$ 上进行划分.究竟在哪些维上进行划分,可以利用主成分分析方法(principal component analysis)[11]确定,这样做的好处是:1)划分维的选择与数据分布无关,使得 BD 索引既适用于均匀分布又适用于非均匀分布数据.2)可以通过调整 d' 的大小和选择,对数据空间进行合理分区,不至于出现过多的空分区或分区不均匀.因此,首先将数据通过主成分分析方法转换到主成分空间,由于前几个主成分代表数据变化最大方向,将其选为划分维可以得到较好的分区效果.为了降低 I/O 代价,可以将分区划分得与页面大小相同,所以根据一个数据页面所能存放的点向量个数 f,得到 d' 的估算公式,$d'=\lfloor \log_2 N/f \rfloor$,其中 N 表示空间点向量总数.

参考点的选择可以分为两种情况:对于均匀分布,将空间中心点选为参考点;对于非均匀分布,可以利用常用的聚类算法(如 BIRCH,CURE[11])将数据分为几个类,各个类新的中心点选为参考点.为简化距离计算,在选中参考点后,将空间坐标原点移到参考点上,这样参考点的坐标为 $O(0,0,\cdots,0)$.

为了叙述方便,在本文后续章节假设数据分布已经转换为 d-维主成分单位空间 $[0,1]^d$,两点间的距离采用欧氏距离公式度量.根据划分维的选择策略,空间分区是在前 d' 维上进行的划分.

定义 1. 向量分区.高维向量空间任意一点 $P(p_1,p_2,\cdots,p_d)$ 的区位码可以表示为 $S(s_1,s_2,\cdots,s_{d'})$,其中

$$s_i=\begin{cases}1, & p_i\geqslant 0,\\ 0, & \text{otherwise},\end{cases}\quad (1\leqslant i\leqslant d').$$

由定义 1 可以看出点 P 的区位码 S 是由 P 的前 d' 维数值所决定的,也可以说,S 的第 i 维 s_i 即为 P 的第 i 维 p_i 的区位码,因此一个 d-维向量就可以用一个长度为 d' 的位码字符串表示,并且同一分区内的所有点具有相同的区位码.根据参考点和划分维 d' 可以将数据空间划分为 $2^{d'}$ 个分区,例如二维向量空间可以划分为 4 个分区,每个分区的区位码如图 1 所示:

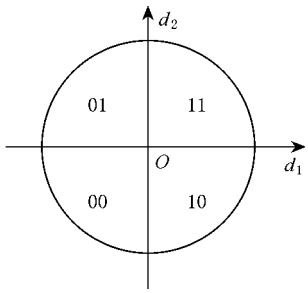

Fig. 1 2-dimensional space partitioning.

图 1 二维向量空间分区划分

定义 2. 向量 key 值.假设任意一点 $P(p_1,p_2,\cdots,p_d)$ 的区位码为 $S(s_1,s_2,\cdots,s_{d'})$,则 P 的 key 值可以利用下面定义的转换函数求得:

$$key_P = Value(S)\times C + Dist(P,O),$$

其中 $Value(S)$ 表示 P 的区位码数值;$Dist(P,O)$ 是 P 和参考点间的距离;C 是一个常数.

需要说明的是,在定义 2 中 $Value(S)$ 是分区 S 的数值表示,可以采用 $Value(S)=2^0\times s_1+2^1\times s_2+\cdots+2^{d'-1}\times s_{d'}$ 公式计算,也可以用其他的公式计算,只要该公式能区分不同分区的数值即可.在 $[0,1]^d$ 空间 $Dist(P,Q)=(\sum_{i=1}^{d}p_i^2)^{1/2}$ 的最大值为 $\sqrt{d}$,所以常数 $C=\lceil\sqrt{d}\rceil$ 就可以将不同分区的 key 值区分开.根据定义 2 可以将一个高维向量转换为一维数值(key)表示,并且这种 key 值中既包含区位码信息,又包含距离信息,具有近似向量和一维转换两种方法的特点.

2.2 KNN 查询

高维空间的 KNN 查询,大多采用逐步扩大查询范围直到检索结果集大小达到 K 为止的方法.但是由于查询点位置和数据分布密度的不同,使得这类方法的初始查询半径很难确定,影响查询效率.根据 BD 空间划分特点,设计一种快速 KNN 查询方法:首先确定查询点 Q 所在分区,在该分区内查找 Q 的 K 个最近邻点;然后以该结果集作为初始

查询范围,确定与其相交的其他分区,以及各个相交分区内需要搜索的 key 范围.在具体搜索过程中,需要不断修改查询范围,使得快速生成精确 KNN 最终结果(算法主要步骤见图 2).

```
Algorithm BD_KNN(Q)
  initialize S; /* answer set */
  examine the partition Q falls into;
  set the query sphere as center Q and radius the distance of the
  farthest object in S from Q;
  L: determine the intersected partitions; /* using Lemma 1 */
     determine the affected ranges; /* using Lemma 2 */
     examine these ranges in a suitable order;
     if S is modified then goto L;
```

Fig. 2 KNN querying algorithm of BD index.

图 2 BD 索引的 KNN 查询算法

与逐步扩大查询范围方法相比,这种查询方法具有多方面的好处:1)避免了对 B^+ 树同一个分支多次搜索的操作;2)符合空间分布特点,最近邻点与 Q 同在一个分区内的概率较大;3)控制查询范围恰当,使得在搜索较小范围的情况下得到精确结果;4)能快速得到近似 KNN 结果,因为对各个相交分区的搜索是为了使 KNN 结果更加精确,所以在精度要求不高的情况下,可以中止对相交分区的搜索,以近似 KNN 结果集返回.

根据区位码可以快速判断哪些分区与查询范围相交,然后利用 key 值进一步确定相交分区内需要搜索的范围. 以二维空间为例,如图 3 所示,Q 为查询点,R 为查询半径,则与之相交的分区是 10,11,在这两个分区内需要搜索的区域在图中用灰色表示.

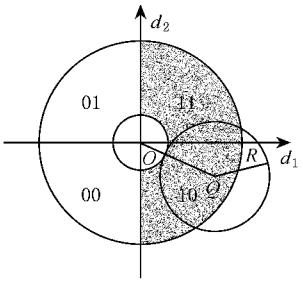

Fig. 3 KNN query in 2-dimensional space.

图 3 二维向量空间 KNN 查询

定理 1. 分区与查询范围相交. 若存在一个查询范围,表示为以 $Q(q_1,q_2,\cdots,q_d)$ 为中心,以 R 为查询半径的圆,假设点 Q 的区位码为 $S(s_1,s_2,\cdots,s_{d'})$,则与该查询范围相交的分区 $T(t_1,t_2,\cdots,t_{d'})$ 必满足下列条件:

$$t_i=\begin{cases} s_i, & |q_i|\geqslant R,\\ 0\text{ or }1. & \text{otherwise},\end{cases}\quad (1\leqslant i\leqslant d').$$

证明. 因为查询范围和分区 $T(t_1,t_2,\cdots,t_{d'})$ 相交,表示查询范围内至少存在一点 $P(p_1,p_2,\cdots,p_d)$ 位于分区 T 内;所以,对于 $\forall i\in[1,d']$,$|p_i-q_i|\leqslant R$ 成立.

根据 q_i 的大小分为下列两种情况:

1) $|q_i|\geqslant R$

若 $q_i>0$,则由 $|p_i-q_i|\leqslant R$,得 $p_i>0$;

同理,若 $q_i<0$,则 $p_i<0$.

所以,当 $|q_i|\geqslant R$ 时,P 和 Q 在 i 维上的区位码相同(定义 1),即 $t_i=s_i$.

2) $|q_i|<R$

由于 $-R<q_i<R$,根据 $|p_i-q_i|\leqslant R$,得 $p_i>0$ 或 $p_i<0$.

所以,当 $|q_i|<R$ 时,P 在 i 维上的区位码可能为 0 或 1,即 $t_i=0$ or 1. 证毕.

定理 2. 相交分区内需要搜索的 key 范围. 若查询点 $Q(q_1,q_2,\cdots,q_d)$,查询半径 R,分区 $T(t_1,t_2,\cdots,t_{d'})$ 与该查询范围相交,则 T 内需要搜索的 key 值范围为

$$[Value(T)\times C+Dist(Q,O)-R,\ Value(T)\times C+Dist(Q,O)+R].$$

3 实验结果与讨论

为了验证 BD 索引技术的效果,分别使用人工模拟数据和真实数据进行实验,因为 BD 索引综合了近似向量表示法和一维向量转换法,所以实验中采用代表这两种思想的典型索引 VA-file 和 iDistance 作为比较对象,虽然目前根据这两种思想的索引还有很多,但是我们选择的是与 BD 索引相似的索引结构进行比较和分析.

3.1 模拟数据实验

实验 1(图 4)考察了数据量对查询时间的影响. 实验中使用的数据是 20 维的向量,数据量从 5 万条到 30 万条,KNN 查询结果集大小 $K=10$. 可以

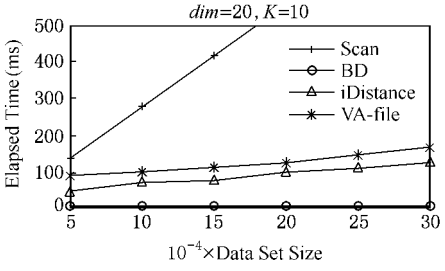

Fig. 4 Influence of data set size on elapsed time.

图 4 数据量对查询时间的影响

看出,无论是利用近似向量表示法还是一维向量转换法,检索性能都比顺序扫描好很多,这是因为顺序扫描需要计算的距离次数随数据量呈线性增长,而利用索引可以使需要搜索的范围缩小,减少距离计算. 因此,在下面的实验中,我们重点比较 BD 和 iDistance,VA-file 之间的区别.

根据图 4,随着数据量的增加,BD 的检索性能比只用一种思想设计的索引 iDistance 和 VA-file 都要好,查询花费时间至少减少 80%,并且随数据量增加这种差距更加明显. 这说明综合两种思想构造的 BD 索引可以同时具备两者的优点,既能利用近似向量加快扫描速度,又能利用一维向量快速缩小搜索范围,从而大大减少距离计算. 另外 BD 索引采用了快速 KNN 检索算法,减少了 B^+ 树搜索,也有利于提高检索速度.

实验 2(图 5)考察了维度对查询时间的影响,实验中生成了维度分别为 10,20,30,40,50,60 的数据,每个数据文件的数据量都是 10 万,KNN 结果集大小 $K=10$.

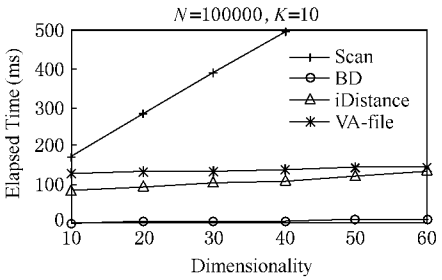

Fig. 5 Influence of dimensionality on elapsed time.

图 5 维度对查询时间的影响

可以看出,维度对于 BD,VA-file 和 iDistance 的查询时间影响都比较小,维度由 10 变化到 60,查询时间的增加均小于 50%. 但是,随维度增加 BD 的查询时间增长显得更加缓慢;并且与 VA-file,iDistance 相比,BD 查询时间始终少出 90%. 维度的增加给 BD 带来了更多的优势. 从理论上分析,这是因为:随着维度的增加,数据分布稀疏现象愈加明显,分布在空间表面的点越来越多,VA-file 对数据空间均匀分割的单元个数快速增加,造成查询时需要扫描的单元增多,相应地时间也增多;iDistance 由于采用简单的一维距离数值代替高维向量搜索,维度增加使得更多的点具有相同的一维数值,这样检索时难免会引入过多的误中点,影响了检索速度. 而从 BD 索引值的构造过程可以看出,分区只是在某些维上进行一次分割,这样不至于产生过多的分区,并利用距离对各个分区进一步细分,因此 BD 对高维空间稀疏数据的划分更加合理,避免了检索时只采用一种表示形式带来的问题,提高了检索剪枝效率.

3.2 真实数据实验

采用来自 68040 幅图像提取的 32 维颜色特征数据,这组实验(图 6、图 7)主要考察结果集大小对查询时间的影响. 随着查询结果集增大,BD,VA-file 和 iDistance 查询时间缓慢增加,但是总体上 BD 查询时间明显少于 VA-file 和 iDistance 所用时间,只有 1/6 左右. BD 索引利用在部分主成分维上进行分区的方式,可以将数据合理地划分到有限个不同的分区中,便于检索时剪枝,使得 BD 更加适用于非均匀分布数据.

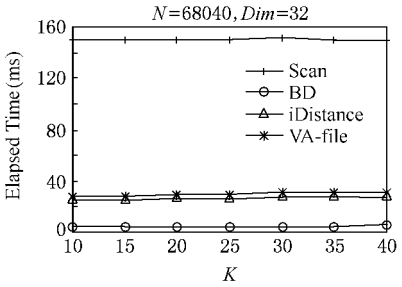

Fig. 6 Influence of searching domain on elapsed time.

图 6 查询范围对查询时间的影响

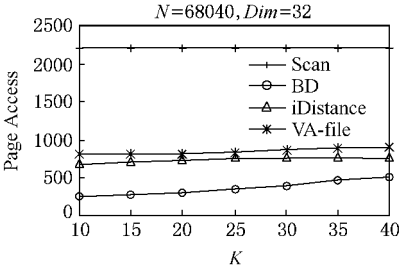

Fig. 7 Influence of searching domain on page access.

图 7 查询范围对页面访问次数的影响

综合可知,BD 的查询效率远远优于 SCAN,iDistance 和 VA-file,尽管它们都是高维数据空间检索表现较好的索引结构,特别是随着数据量增大或维度增高这种优势更加明显,并且 BD 索引结构不受数据分布影响,能很好地适用于各种数据分布情况,所以 BD 是一种有效的针对高维空间 KNN 检索的索引结构.

4 总 结

本文结合近似向量和一维向量转换两种思想,提出基于区位码和距离的索引结构(BD),在充分考

虑到高维空间向量分布特点的前提下,对高维空间进行合理分区,根据区位码和转换函数实现高维向量近似表示和一维数值表示形式,利用快速 KNN 查询算法大大提高检索效率. 实验证明,BD 的检索性能优于其他同类索引结构.

参 考 文 献

[1] E Chavez, G Navarro, R Baeza-Yates, *et al*. Searching in metric spaces [J]. ACM Computing Surveys, 2001, 33(3): 273-321

[2] M J Fonseca, J A Jorge. Indexing high-dimensional data for content-based retrieval in large databases [C]. The 8th Int'l Conf on Database Systems for Advanced Applications, Kyoto, 2003

[3] G R Hjaltason, S Hanan. Index-driven similarity search in metric spaces [J]. ACM Trans on Database Systems, 2003, 28(4): 517-580

[4] R Weber, H Schek, S Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces [C]. The 24th Int'l Conf on Very Large Data Bases, New York, 1998

[5] K Beyer, J Goldstein, R Ramakhrisnan, *et al*. When is "Nearest neighbor" meaningful [C]. The 7th Int'l Conf on Database Theory (ICDT'99), Jerusalem, Israel, 1999

[6] A Hinneburg, C C Aggarwal, D A Keim. What is the nearest neighbor in high dimensional spaces [C]. The 26th Int'l Conf on Very Large Data Bases, Cairo, Egypt, 2000

[7] Dong Daoguo, Liu Zhenzhong, Xue Xiangyang. VA-Trie: A new and efficient high dimensional index structure for approximate k nearest neighbor query [J]. Journal of Computer Research and Development, 2005, 42(12): 2213-2218 (in Chinese)
(董道国, 刘振中, 薛向阳. VA-Trie:一种用于近似 k 近邻查询的高维索引结构[J]. 计算机研究与发展, 2005, 42(12): 2213-2218)

[8] B Cui, *et al*. Exploring bit-difference for approximate KNN search in high-dimensional databases [C]. The 16th Australasian Database Conf, Newcastle, Australia, 2005

[9] S Berchtold, C Bohm, H P Kriegel. The pyramid-technique: Towards breaking the curse of dimensionality [C]. ACM SIGMOD Int'l Conf on Management of Data, Seattle, Washington, 1998

[10] C Yu, *et al*. Indexing the distance: An efficient method to KNN Processing [C]. The 27th Int'l Conf on Very Large Data Bases, Roma, Italy, 2001

[11] S Guha, R Rastogi, K Shim. Cure: An efficient clustering algorithm for large databases [C]. ACM SIGMOD Int'l Conf on Management of Data, Seattle, Washington, 1998

Liang Junjie, born in 1974. Since 2002, she has been Ph. D. candidate in computing science from Huazhong University of Science and Technology in Wuhan, Hubei Province. Her current research interests include multimedia database, content-based image retrieval, and high-dimensional indexing.

梁俊杰,1974 年生,博士研究生,主要研究方向为多媒体数据库、基于内容的检索和高维索引结构.

Wang Changlei, born in 1974. Since 2005, he has been a master candidate in computing science from Shanghai Jiaotong University. His current research interests include multimedia database, image processing, *etc*.

王长磊,1974 年生,硕士研究生,主要研究方向为多媒体数据库、图像处理.

Research Background

In recent years, the technology of high-dimensional indexing has found application in a huge range of application domains, such as multimedia information processing, Web data mining, and meteorological, geographic and medical data analysis. K-nearest neighbor search algorithm (KNN) is a widely used indexing method. Among a variety of index structures which facilitate high-dimensional KNN queries, the techniques of approximate vector presentation and one dimensional transformation can efficiently break the curse of dimensionality. In this paper, a high-dimensional index called bit-code and distance based index(BD) is proposed. Based on this method, we have done extensive experiments, using both synthetic and real data. The results demonstrate that the BD outperforms the existing index structures for KNN search in high-dimensional spaces.