# HilCluster:一种简单有效的R-树压缩技术

陈学工,张 厅,张文艺,张驰伟

(中南大学 信息科学与工程学院,湖南 长沙 410083)

**摘 要**:传统的Hilbert Packed R-树是利用Hilbert值对空间实体依次进行压缩,算法简单快速,然而空间位置上邻近的空间实体的Hilbert值并不一定相邻,使得在数据分布不均匀时,查询效率开始下降;递归聚类的算法虽然解决了以上问题,但是它计算复杂,而且容易造成R-树的不平衡,以至降低了存储利用率和检索的效率。文中对两种方法加以综合,提出了一种新的批量加载R-树的算法—HilCluster。实验结果表明,新算法不仅继承了Hilbert Packed R-树构造过程时间消耗低、存储利用率高的优点,还使得查询效率进一步提高。

**关键词**:R树;批量加载;聚类

**中图分类号**:TP311.132 **文献标识码**:A **文章编号**:1673-629X(2007)06-0110-03

# HilCluster: a Simple and Efficient Algorithm for R-Tree Packing

CHEN Xue-gong, ZHANG Ting, ZHANG Wen-yi, ZHANG Chi-wei

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

**Abstract**: Traditional Hilbert Packed R-Tree packs the spatial objects by the turn of their Hilbert values. Though the algorithm is simple and rapid, the spatial objects which neighboring in their spatial location do not necessarily neighboring in their Hilbert values. Because of this, its query efficiency decline for the data distributed unevenly. The algorithm clustering recursively resolved the problem, but it has high construction expense, and the R-Tree constructed by means of it usually is imbalance, which result in low space utilization and efficiency. In this paper, united above two methods, proposed a new bulk-loading algorithm for the R-Tree called HilCluster. The experimental data indicates that the new algorithm not only inherit Hilbert Packed R-Tree's low construction expense and high space utilization percent, but also has better performance in searching.

**Key words**: R-Tree; bulk-loading; clustering

## 0 引 言

空间数据库是存储和管理空间信息的数据库系统。为了快速、有效地处理存储于空间数据库中的海量空间数据,专家学者提出了大量的基于磁盘的空间索引方法。其中,由Guttman于1984年提出的R-树是目前最流行的动态空间索引结构,并得到广泛应用。传统的R-树的构建方法是从空树开始,利用插入算法逐个插入记录而生成。这种被称为OBO(one-by-one)的方法,由于要动态维护空间索引结构,导致R-树的创建过程耗时巨大,存储利用率也不高(70%左右)。与此相应的是,GIS空间数据操作的特点是:插入、删除操作相对较少,而查询操作则比较频繁。因此,专家学者开始寻求高效的R-树批量操作技术。常用的批量装载算法有两种:一种是基于分形曲线构建静态R-树的压缩算法,如Hilbert packed R-树[1];另一种是采用递归聚类的方法构造静态R-树的压缩算法,如文献[2,3]中论述的算法。前者虽然具有算法简单、建树速度快和存储利用率高的优点,但是由于在映射过程中破坏了实体之间的相邻性,Hilbert值不能完整地反映各空间实体间的位置关系,在数据分布不均匀时会产生结点之间的大量重叠,严重影响R-树的检索性能;后者的优点是,都能够很好地将空间实体进行聚类,但它构造R-树的过程时间消耗很高,由于聚类的结果不可预计,聚类结果中对象分布不均,因此必须对聚类结果进行调整,否则将造成R-树的极不平衡,降低了存储利用率和检索效率。而调整本身是一个复杂的过程,而且调整必然破坏原有的空间关系。鉴于此,文中对以上两种方法进行综合,先将空间实体划分为 $k$ 个聚类,然后对这 $k$ 个聚类分别建立Hilbert R-树。由于这 $k$ 个聚类之间的交叠很少,故下层的各Hilbert R-树之间基本上不存在交叠,因此新算法进一步提高了R-树的检索性能。

**收稿日期**:2006-08-18

**作者简介**:陈学工(1965-),男,湖南长沙人,副教授,博士,研究方向为地理信息系统等。

## 1　相关工作

### 1.1　R-树原理

R-树是一种采用对象界定技术的高度平衡树,树中用对象的最小外包矩形(MBR)来表示对象。设 $M$ 为每个结点所能容纳的最大实体数,$m$ 为每个结点容纳的最小实体数,且 $2 \leqslant m \leqslant M/2$,则R-树必须满足下列条件:

a.根结点若非叶子结点,则至少有2个子结点;

b.每个非根结点包含的实体个数介于 $m$ 和 $M$ 之间;

c.所有叶子结点在同一层次上。

影响R-树索引查询效率主要取决于两个参数:覆盖和交叠[4]。若要得到一个高效的R-树,覆盖和交叠都应达到最小。而且交叠的最小化比覆盖的最小化更加关键。

### 1.2　Hilbert R-树

Hilbert R-树[1]是利用Hilbert曲线对已知空间数据对象进行更好的一维排序,以获得优良的压缩效果。其中,大量的实验表明"2D-C"(利用空间对象MBR中心点的Hilbert值进行排序)变体性能最优。

Hilbert R-树采用压缩技术,存储利用率很高,接近100%;创建过程消耗低,时间复杂度为 $O(n^2)$;查询性能较好,不仅优于Roussopoulos等提出的Packed R-树,而且优于R-树所有动态版本,如 $R^+$ 树、$R^*$ 树等[5]。但是,由于Hilbert曲线不能完全反映出空间实体的分布情况,在从高维空间向一维空间映射的过程中不可避免地存在着信息的损失,在空间位置上相近的点,它们的Hilbert值并不一定相近,这样就造成了Hilbert R-树结点之间的大量交叠。如图1(a)所示,图中在空间位置上相邻的点不一定处在同一结点中,从而造成了结点之间较多的交叠。

### 1.3　基于递归聚类的索引树生成算法

聚类是提高空间数据查询处理性能的一个非常有用的特性。使用聚类技术,可以将逻辑上相关的对象存储在相同的磁盘页面,达到减少磁盘的访问次数的目的。递归聚类的思想就是将空间实体从上而下或从下而上层层聚类,使得空间上相近的结点都处于同一结点中,尽可能减少结点之间的交叠,尤其在数据分布不均匀的情况下,聚类的优势更加明显。

但是,递归聚类有其固有的缺点。文献[2]中提出一种自上而下的递归聚类算法。虽然它在查询效率上较SS树有所提高,但是它创建过程消耗很高,时间复杂度为 $O(n^3)$,而且由于它没有对聚类结果进行有效的调整,导致结点中实体的数目不可预测,存储利用率较低。如图1(b)所示的是由一自上而下的递归聚类算法产生的R-树,图中显示有较多的结点中只存有单一数据点,从而造成了存储利用率较低。文献[3]中提出一种自下而上的递归聚类算法。创建过程的时间复杂度也是 $O(n^3)$;算法中虽然对自然聚类进行了相应的优化、调整,但是调整过程复杂繁琐,调整的过程中不可避免地破坏了原有的实体分布关系,造成结点的覆盖和交叠面积的增大。

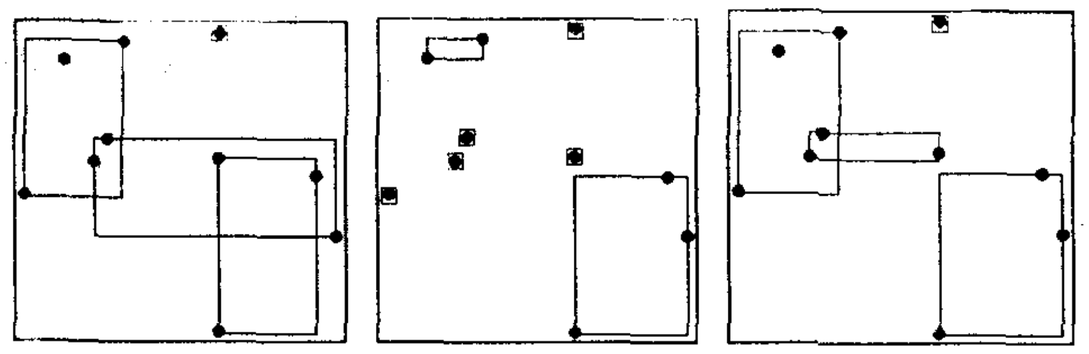
## 2　HilCluster R-树构造方法

HilCluster的构造思想就是试图将Hilbert R-树构造方法与k-min聚类技术相结合,既要减少在数据分布不均的情况下结点之间的交叠(k-min聚类的优点),又要降低构造树的时间复杂度,提高存储利用率(Hilbert R-树的优点)。

构造过程分两个步骤:

Step1:用空间实体MBR的中心点代表每一个空间实体,采用常用的聚类算法k-min方法对这些空间点进行聚类,得到 $M$($M$ 为R-树中每个结点所能容纳的最大实体数)个聚类。

Step2:对产生的 $M$ 个聚类进行判别,如果某个聚类中点的个数不大于 $M$,则不作处理;否则求出它们的Hilbert值,利用构造Hilbert R-树的方法,构造Hilbert子树。



(a) Hilbert R-树　(b) 递归聚类R-树　(c) HilCluster R-树

图1　结点的分布情况比较

## 3　性能分析和评价

设总的空间点个数为 $n$,R-树中每个结点所能容纳的最大实体数为 $M$。HilCluster R-树的创建费用主要由两部分组成:

(1) 采用k-min方法产生 $M$ 个聚类的费用:$M * n * t$($t$ 为聚类时迭代次数,可以看作远小于 $n$ 的常数);

(2) 对各聚类分别构造Hilbert R-树的费用:由于每个聚类中空间点的个数不可预测,假设分别为 $n_1, n_2, n_3, \cdots, n_M$,则费用为 $n_1^2 + n_2^2 + n_3^2 + \cdots + n_M^2 \leqslant n^2$。

将两步综合起来,HilCluster R-树的创建过程时间复杂度为 $O(n^2)$,总时间接近Hilbert R-树。

从存储利用率来说,在最坏的情况下,聚类结果中有 $M-1$ 个孤立点,其他 $n-M+1$ 个点在一个聚类中,对这个聚类结果构建 Hilbert R-树,当 $M \leqslant n$ 时,HilCluster R-树的存储利用率仍然接近 Hilbert R-树。

从查询效率来说,在构造 Hilbert R-树之前,对数据进行聚类处理,使得邻近的数据都处在同一个聚类中,大大减少了结点之间的交叠,特别是对于分布不均匀的数据,这种优势应该更加明显,后面的试验将重点对查询效率进行测试。

## 4 实验与结果

实验中选用 Hilbert R-树和自上而下的递归聚类产生的树与文中提出的 HilCluster R-树进行对比实验。

实验环境是奔腾Ⅲ 850MHz CPU,128M 内存,金钻 30G(7200 转)硬盘,Windows2000 操作系统。使用 Visual C++ 7.0 进行算法实现。

选用了两类数据进行实验:一类是模拟数据集,它是计算机模拟产生的随机数,包括均匀分布和正态分布两种;另一类是实际数据集,它来自美国地图中的各城镇的地理坐标。

实验过程如下:

Step1:产生随机样本。样本的个数依次为 10000~100000;样本的分布类型可以为均匀分布或正态分布。

Step2:对产生的样本进行点查询性能测试。测试方法为随机产生 1 个空间点,然后对点进行查询,记录查询过程中访问结点次数,这样循环 5000 次,将每个点的查询过程中访问结点的次数相加,最后求得平均次数。实验数据如图 2 所示。

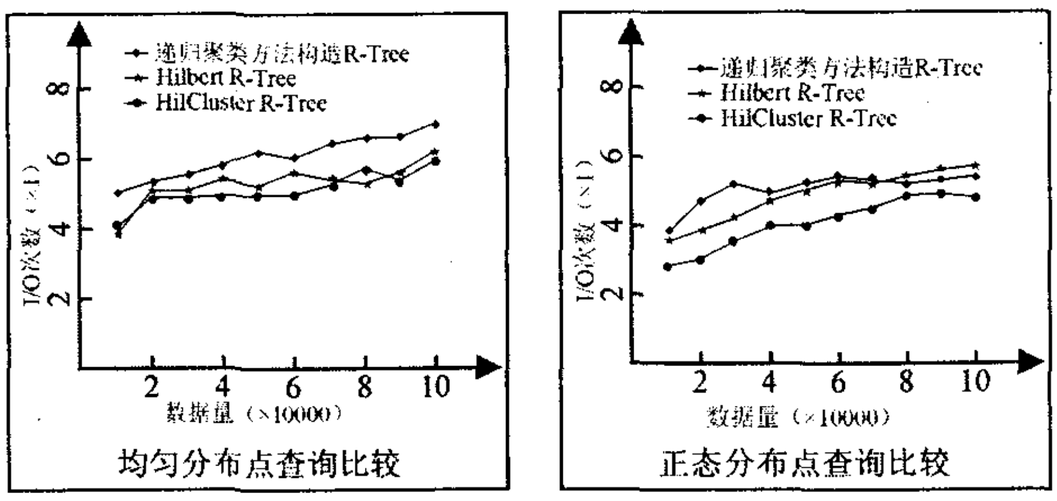



图 2 点查询性能随数据量变化实验

Step3:对产生的样本进行区域查询性能测试。测试方法为首先确定查询区域大小占实际空间大小的比例(设查询区域面积为 $Q$),然后随机产生一点作为区域的左上角,随机产生一个整数作为区域的长 $l$,则 $Q/l$ 为区域的宽;随机产生 5000 个这样的矩形区域,分别进行查询测试,并求得访问结点的平均次数。实验数据如图 3 所示。

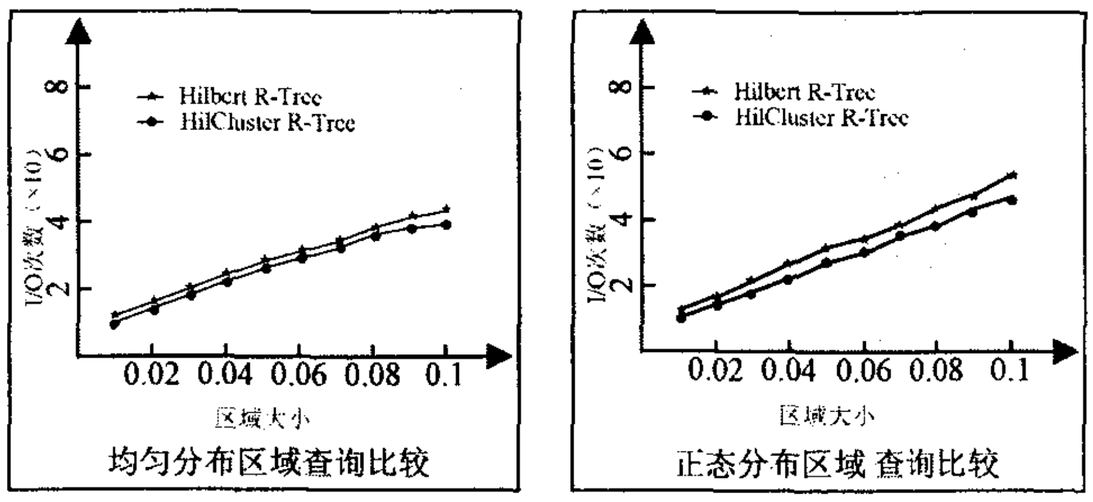
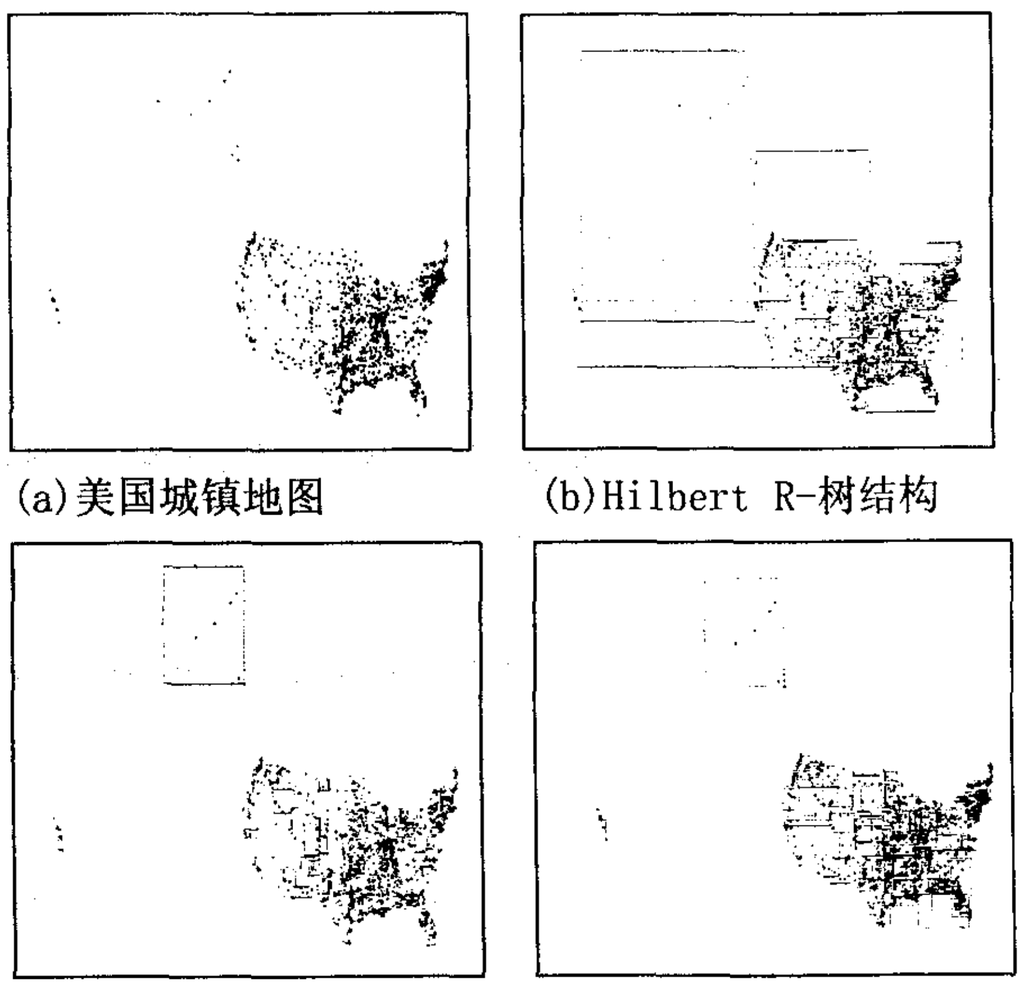



图 3 区域查询性能随查询区域大小变化实验

Step4:采用 Step2 和 Step3 中所述测试方法,对美国地图数据进行索引性能比较。图 4 中分别描述了采用不同方法对美国地图中城镇坐标建立索引后的结构图,图中只画出了各 R-树中的叶子结点的 MBR。图 5 中给出了查询测试的结果,其中(a)为点查询测试数据,(b)为区域查询测试数据。



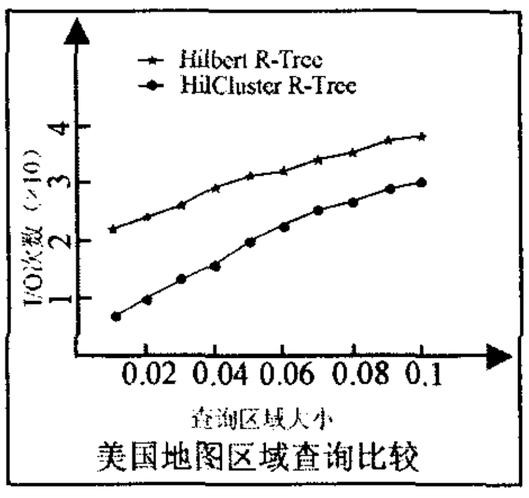
(a)美国城镇地图 (b)Hilbert R-树结构

(c)递归聚类 R-树结构 (d)HilCluster R-树结构

图 4 对美国地图建立索引结构比较

| R-树方法 | 访问结点平均次数 |
|---|---|
| Hilbert R-树 | 1.5 |
| 递归聚类 R-树 | 0.7 |
| HilCluster R-树 | 0.2 |

(a)美国城镇地图点查询效率比较




(b)美国城镇地图区域查询效率比较

图 5 美国地图查询效率比较

(下转第 115 页)

类别;对于拒识的样本,采用就近原则确定样本的类别。

## 4　实验和结论

文中所使用的实验数据是中文自然语言处理开放平台中提供的中文文本分类语料库,选取了计算机、交通、教育、经济、体育、艺术、政治七类共 2162 篇文本,并在各类别中按 1:1 的近似比例随机抽取学习样本集和测试样本集。文本首先通过分词,共得到 79360 个词组,然后通过 $x^2$ 统计量方法进行特征降维,从中取出 CHI 值较高的 1500 个词组作为特征库,特征值只占词组总数的 1.8901%。实验结果如下所示,其中表 1 以词组在文本中是否出现作为特征向量中各维的值(0 或 1),表 2 以词组在文本中出现的频率作为特征向量中各维的值,以精确率和召回率作为评价标准[7]。

表 1　以词组是否出现作为向量值

| 类别 | 学习数 | 测试 | 精确率 | 召回率 |
|---|---|---|---|---|
| 计算机 | 99 | 101 | 87.3585% | 92.0792% |
| 交通 | 99 | 115 | 91.3462% | 82.6087% |
| 教育 | 109 | 111 | 75.9398% | 90.9910% |
| 经济 | 162 | 163 | 90.4348% | 63.8037% |
| 体育 | 219 | 231 | 92.2080% | 92.0780% |
| 艺术 | 124 | 125 | 75.4718% | 96.0000% |
| 政治 | 270 | 235 | 91.4163% | 90.6383% |
| 总正确率 | 1082 | 1081 | 86.8640% | |

从实验结果可以看出,在文中所使用的方法中,以词组在文本中出现的频率作为特征向量中各维的值,其效果较以词组在文本中是否出现作为特征向量中各维的值有明显提高。从文本各类别的精确率和分类总正确率上可以看出,使用 CHI 方法降维,可以在特征向量维数较低时仍能取得较好分类效果,从而可以大大降低分类的时空开销。可见,将 CHI 方法与交叉覆盖算法结合可得到一个效果较好的文本分类器。

表 2　以词组出现频率作为向量值

| 类别 | 学习 | 测试 | 精确率 | 召回率 |
|---|---|---|---|---|
| 计算机 | 99 | 101 | 92.2330% | 94.0594% |
| 交通 | 99 | 115 | 87.4016% | 96.5217% |
| 教育 | 109 | 111 | 91.1504% | 92.7928% |
| 经济 | 162 | 163 | 93.2836% | 76.6871% |
| 体育 | 219 | 231 | 95.5947% | 93.9394% |
| 艺术 | 124 | 125 | 93.7985% | 96.8000% |
| 政治 | 270 | 235 | 89.5161% | 94.4681% |
| 总正确率 | 1082 | 1081 | 91.9519% | |

**参考文献:**


[1] 刘丽珍,宋瀚涛.文本分类中的特征抽取[J].计算机工程,2004,30(4):14-15.

[2] 陈　涛,谢阳群.文本分类中的特征降维方法综述[J].情报学报,2005,24(6):600-604.

[3] Yang Y, Pedersen J O. A comparative study on feature selection in text categorization[C]//In:Proceeding of the 14th International Conference on Machine Learning(ICML'97). San Francisco: Morgan Kaufmann Publishers,1997:412-420.

[4] 张　铃,张　钹.M-P 神经元模型的几何意义及其应用[J].软件学报,1998,9(5):334-338.

[5] 张　铃,张　钹,殷海风.多层前向网络的交叉覆盖算法[J].软件学报,1999,10(7):737-742.

[6] 吴　涛,张燕平,张　铃.前向神经网络交叉覆盖算法的一种改进[J].微机发展,2003,13(3):50-52.

[7] 程泽凯,林士敏.文本分类器准确性评估方法[J].情报学报,2004,23(5):631-636.


(上接第 112 页)

## 5　结　语

文中将 Hilbert R-树构造方法与 k-min 聚类技术相结合,提出了一种先对数据采用 k-min 聚类方法进行聚类,然后分别对 $k$ 个聚类采用 Hilbert R-树构造方法构造 R-树的新的压缩算法——HilCluster。经分析得出该压缩算法时间消耗低、存储利用率高,而且查询效率高;实验数据进一步表明,无论是点查询性能还是区域查询性能,由 HilCluster 算法构造的 R-树比前面提到的两种压缩算法都有较大优势,尤其是在数据分布不均匀的情况下,优势更加明显。

事实上,Hilbert R-树构造方法本身也属于一种聚类方法,文中就是将两种聚类方法有机结合起来应用于 R-树的构造过程。然而 k-min 方法需要人为地给出一个参数($k$ 值),将数据集合分为 $k$ 个聚类,这也不一定完全符合实际数据分布的特点,因此引进一种能自动聚类的聚类算法很有必要。

**参考文献:**


[1] Kamel I, Faloutsos C. On packing R-trees[C]//In: Proceedings of CIKM. Washington, DC, USA: [s. n.], 1993: 490-499.

[2] 王　颖,汪晓岩.基于递归聚类索引树的剪枝相似检索算法[J].合肥工业大学学报:自然科学版,2000,23(4):597-600.

[3] 黄继先,鲍光淑.一种面向 GIS 的静态 R-树数据组织方法[J].中南大学学报:自然科学版,2005,36(3):491-495.

[4] 谢昆青.空间数据库[M].北京:机械工业出版社,2004:118-123.

[5] 张明波,陆　锋.R 树家族的演变和发展[J].计算机学报,2005,28(3):289-300.