

# 空间信息检索及其数据库概化技术\*

陆桑璐<sup>1,2</sup>, 周晓方<sup>3</sup>, 陈贵海<sup>1,2</sup>, 谢立<sup>1,2</sup>

<sup>1</sup>(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093);

<sup>2</sup>(南京大学 计算机科学与技术系,江苏 南京 210093);

<sup>3</sup>(澳大利亚昆士兰大学 计算机科学与电子工程系,澳大利亚)

E-mail: sanglu@nju.edu.cn

http://www.nju.edu.cn

**摘要:** 在一个空间数据库中,空间数据总是以最细节的信息进行储存,以满足不同应用的需求.制图学中的制图综合就是为不同的应用生成恰当的细节层面.随着空间数据库应用的日益广泛,以及越来越多的基于 Internet 的空间应用的出现,对空间数据的有效自动综合成为一个紧要问题.深入探讨了当前这一领域存在的问题,提出一种面向数据库的概化处理技术,利用空间索引机制  $z$ -value 进行扩展,并将其应用到空间数据的自动综合技术中,通过对空间数据库的设计和对空间操作的处理来实现一个有效的空间数据库自动综合系统.

**关键词:** 空间数据库;制图综合; $z$ -value

中图法分类号: TP311 文献标识码: A

近几年来,随着计算机的普及和 Internet 的飞速发展,电子地图、基于 Internet 的公共多媒体导购导游系统、汽车 GPS 自导航系统等应用相继出现,并具有越来越大的市场.这一切均离不开空间数据库(spatial database)的支持.一般情况下,空间对象总是以最细节的信息存放在空间数据库中,以满足一般应用的需求.然而,这样一些细节对于大部分应用来说,可能是没有必要的,并且过量的数据将引起较大的开销.

在保证信息真实及视觉效果的前题下,为不同的应用生成不同的细节层面,以减少空间数据集的规模和复杂度,这一过程在制图学中被称为制图综合<sup>[1]</sup>.它并不是简单地选择适合于一个应用的对象或对象的某部分,还需要维持这些空间对象本身所具有的语义信息,如拓扑关系以及综合后地图的整体视觉感受<sup>[1-5]</sup>.综合过程中,对象的重要性决定了在处理后的数据集中应该包括哪些对象以及如何显示.这正是困难所在:对象的重要性不仅取决于规模,也取决于这些对象相对于应用的意义,这是应用相关的,往往也是主观的.这作为制图学中的一个经典问题,一般是通过大量的人力预先针对不同的应用获取不同规模的地图.

在空间数据的制图综合过程中,除了质量以外,性能是另一个需要关心的问题.随着空间数据库应用的日益广泛,以及越来越多的基于 Internet 的空间应用的出现,对空间数据制图综合的效率的要求越来越成为一个紧要问题.与此同时,在 Internet 环境中为所有的应用提供人工干预和预先综合好的空间数据集,却已经越来越不可行了.

本文中,我们主要考虑空间数据制图综合的效率问题,并从数据库的角度,通过对空间数据库的设计以及空间操作来实现有效的制图综合系统,我们称其为数据库概化系统.下面,我们首先从传统的制图学角度讨论空间数据的制图综合问题.然后,从一个新的角度,通过对空间数据库的研究来考虑该问题.最后提出具体的数据库

\* 收稿日期: 2001-04-24; 修改日期: 2002-01-14

基金项目: 国家高技术研究发展计划资助项目(2001AA113050)

作者简介: 陆桑璐(1970 - ),女,云南昆明人,博士,副教授,主要研究领域为分布并行计算,空间数据库;周晓方(1963 - ),男,江苏南京人,博士,主要研究领域为空间数据库,信息可视化,并行数据库;陈贵海(1963 - ),男,江苏盐城人,博士,教授,主要研究领域为并行处理,空间数据库;谢立(1942 - ),男,江苏常熟人,教授,博士生导师,主要研究领域为分布式计算,并行处理,数据库.

概化新技术.

## 1 空间数据的制图综合问题

空间数据的制图综合问题一直是制图学和计算机科学中的热点问题<sup>[1]</sup>.我们首先从传统的制图学角度研究空间数据的制图综合问题.

### 1.1 制图综合过程

制图综合过程由一组基本操作组成<sup>[1,4]</sup>,它们可以简单地分为以下 3 类<sup>[5]</sup>:(1) 选择(selection)操作:从源数据集中确定代表性对象的子集.(2) 简化(simplification)操作:确定一个对象的哪些部分应该被显示.在这个过程中,一个重要的因素是保持简化后对象和源对象的相似性以及拓扑关系的维持.(3) 标记化(tokenization)和合并(amalgamation)操作:对于那些较为重要的对象,如果在制图综合后太小以至于不易辨认时,需要用可见的标记来代替.此外,可能也需要将几个对象合并为一个新对象,或者为了保证拓扑关系替代一些对象.

在整个制图综合过程中,第 1 步就是对象选择.在其后的简化、标记化和合并操作过程中所针对的对象都是由选择操作从数据库中选择出来的结果.因此,我们将重点放到对空间数据库的选择操作,然后再考察同样的技术对简化、标记化和合并操作的支持能力.

### 1.2 存在的问题

在一个空间数据库中,为不同的应用提供不同细节层面的实现方法一般分为两类.一类是在数据库中实际存储一个数据集的多份视图,每个视图对应于一类应用.也即,每个视图包括源数据库的一份制图综合数据集,对所有可能的应用其视图被预先计算并存储在数据库中.这种方法被称为多级数据库(multi-resolution database).由于一般来说空间数据集的数据量都很大(从 gigabytes 到 terabytes),因此需要一个巨大(甚至不可能)的额外存储能力.此外,可能产生的总查询数也是巨大的,对显示设备的要求也会较高.同时,当数据库修改后,要维持同一对象的多份不同层面的表示,其一致性也会引起较大的开销,问题也比较多<sup>[3]</sup>.

另一类方法是在数据库外简化数据集,也就是说,满足一个查询条件的所有数据对象都从数据库中读取出来,然后放到一个专门的程序中处理.用于制图学中生成高质量地图的大部分制图综合算法都属于此类<sup>[1,4]</sup>.这一方法能够将数据集减少到预期的复杂度.但是,这种为生成高质量打印地图而设计的方法不考虑性能问题,没有采取任何措施减少整个处理过程中耗时的主要因素——数据库中访问空间对象的开销.

## 2 面向数据库的方法

从上述讨论中我们可以看到,性能一直是阻碍空间数据库广泛应用的一个主要问题.由于对于大量级复杂空间对象的访问本身就是一个瓶颈,所以任何需要将所有的空间对象从数据库中先行获取才能处理的方法都是极为困难的.但另一方面,哪些对象以及对象的哪些部分在制图综合后会被需要,往往涉及到这些对象本身以及这些对象邻近的其他对象,这意味着这些对象已经从数据库中读取出来,已经是在应用制图综合算法的时候了<sup>[1,4]</sup>.这显然是一个矛盾.

在本文中,我们提出一种新方法来解决这一问题.该方法利用了数据空间和对象近似之间建立一个映射的空间索引技术  $z$ -value,该技术目前主要用于空间 DBMS(database management system)的快速存取<sup>[6]</sup>.我们的想法是,如果空间对象对于一个应用的重要性能能够通过它们的索引项来确定,空间数据的制图综合就能在空间 DBMS 中完成.索引项一般都要比空间数据本身更为简单,这将大大提高系统的效率.此外,还有另一个重要意义:对恰当的简化空间数据集的选择只会引起较小的额外开销.而传统的方法往往需要编写复杂代码的应用程序,以实现空间数据的制图综合.

### 2.1 基于Internet的空间应用

在一个基于 Internet 的空间应用的典型结构中,数据处理的整个过程主要包括下列步骤:

数据获取⇒数据编码⇒数据传输⇒数据解码⇒数据显示

数据库服务器                  网络                  客户端

对空间数据的制图综合处理既能提高数据库服务器端的性能(如减少编码和传输的数据量),又能提高应用客户端的性能(如减少画图显示的对象,且图形形状较简单).显然,一旦对象从数据库中检索出来,它们就能够在服务器端被简化-删除空间对象的部分数据,或找到某种合适的能够保持其特征的抽象.这类简化被称为对象综合(object generalization)<sup>[4]</sup>.以前的研究工作大部分都是针对对象综合的处理.

另一种制图综合我们称为数据库概化(database generalization),它是从数据库的角度,将制图综合作为数据库查询处理的一个整体部分完成.该概化同样从源数据集中删除一些对象,但考虑的主要是重要的空间对象,如显示时视觉上可辨认的对象<sup>[7]</sup>.在该方法中存在着一个转换规则:将用户的数据库请求转换为包含了对这些数据的最终用途进行考虑的请求.例如,对那些即使显示在 Web 浏览器上也不能被感知的对象,数据库概化根本就不会将它们从数据库中读取出来.这样,仅仅访问较小量数据的优势将延续到随后的各个步骤,如数据编码、传输等.但另一方面,这个概化过程应该仍保持源地图的特征,否则将会降低地图的质量.此外,由于该算法本身的开销也可能增加整个系统处理的开销,我们也需要考虑数据库概化算法本身的开销.因此,我们的目标是,设计一个能够实质性地减少数据量,同时又能以尽可能小的开销维持源数据集关键特征的有效的数据概化方法.

数据库内实现空间数据概化的方法将在第 3 节中详细地加以描述,在此之前,我们首先对对象近似、空间索引和基于此的空间数据处理策略<sup>[7]</sup>进行简短的介绍.

2.2 空间索引和z-value

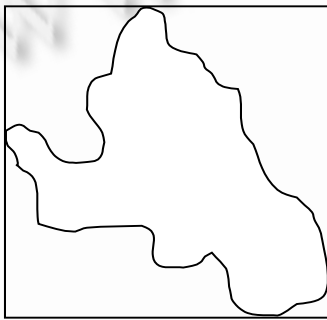


Fig.1 Minimum bounding rectangle  
图 1 最小包围矩形

对于一个任意复杂的空间对象,如多边形,一般用它的最小包围矩形 MBR(minimum bounding rectangle)来近似,如图 1 所示.然而,MBR 是一种较为粗略的近似,一个对象往往还需要被近似到比 MBR 更为精细的程度.例如,对于穿越整个数据空间的一条长路来说,它的 MBR 显然是一种不太合适的近似.z-value 技术基于空间填充曲线(space filling curve),通过将数据空间循环分解到更小的子空间(被称为 Peano Cell)来获得对于给定对象形状的近似<sup>[6]</sup>.作为一种空间索引机制,它已经被用于几个商业化的空间数据库系统,例如 Oracle<sup>[8]</sup>.

用 z-value 技术将数据空间循环分解到更小的子空间后,让每个子空间根据分解步骤依次得到一组数字,称为该子空间的 z-value.求解 z-value 的分解过程为:整个数据空间被分为 4 个相同大小的子空间,这 4 个子空间依照一定的次序被编号为 1~4.这些子空间能够进一步循环分解和编号.一个子空间由下列步骤确定其 z-value:

- (1) 初始空间的 z-value 为 1;
- (2) 一个 z-value 为  $z(z=z_1 \dots z_n, 1 \leq z_i \leq 4, 1 \leq i \leq n)$  的子空间,其 4 个子空间的 z-value 分别为  $z_1 \dots z_{n1}, z_1 \dots z_{n2}, z_1 \dots z_{n3}, z_1 \dots z_{n4}$ .

由于一个空间对象能够用与它重叠的一组子空间来近似,因此也得到惟一的一组 z-value.这样,空间对象能够被表示为一组数字(z-value),并且能够从该组数字中得到相应的近似空间信息.子空间有不同的大小,z-value 有不同的长度,显然,子空间越大,相应的 z-value 越短.这里,分辨率(resolution)是指最大的分解层次,它决定了 z-value 的最大长度.图 2 中的多边形可以由 7 个子空间近似,其分辨率是 4.这样,通过 z-value 我们可以将一个两维对象转换为一组一维点(z-value),因此可以用常见的一维访问

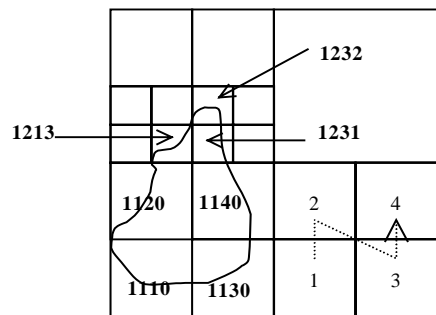


Fig.2 Polygon approximation using z-values  
图 2 Z-Value 对多边形的近似

方法,如 B+树来维持。

### 3 基于 $z$ -value 的数据库概化

下面我们将依次讨论如何利用  $z$ -value 来实现概化过程的基础操作:选择、简化、标记化和合并。

#### 3.1 对象选择

根据地图的视觉效果,一个合理的概化方法应能够确定重要的对象以及对象的重要部分。这里,有两类“重要”需要区分。一类是由查询请求的语义决定,如对于在某个特定区域查询农作物数据的用户来说,不会对其他无关请求的数据感兴趣,因此,农作物数据在这里被认为是重要的。对于这类重要数据来说,由用户在查询中明确指定相应的属性限制,就可以很容易地让低层 DBMS 实现对象访问操作。而另一种类型的重要数据则需要从应用和请求的设置中推断出来,是一类隐含的重要性限制。例如,对于一个给定的应用请求,可能推断出一些约束条件,如小于一定比例的对象不应该显示,因为它们太小以至于视觉不可辨别。因此,实际上对于交互式的空间应用,我们不需要直接显示所有满足用户指定条件的数据项,而是应该首先自动获得表示视觉上重要对象的隐含条件。对于满足下列条件之一的空间对象,我们定义为视觉上重要的对象:

(1) 大对象:规模超过某个尺度的对象。这不仅要考虑对象所占区域的面积,还应考虑其他的标准,如扩展性。例如,对于一条长路来说,虽然只具有相对较小的区域面积,但视觉上仍很重要,应作为大对象处理。

(2) 分布在空旷区域的对象:位于空旷区域的小对象,如沙漠中的一个小镇可能比都市里的一个市郊还要重要,尽管后者的面积可能更大。

(3) 代表性对象:对于都满足查询条件又不能由前两个标准区分的大批对象来说,可能有必要仅选择这些对象的一个子集作为全集的一个代表。例如,当不可能显示一个区域的全部土地块时,部分土地块需要作为代表来显示,以表示该区域的土地使用情况。

其中,第(1)个是度量约束,后两个则试图捕捉源数据集的整体视觉感受。

下面,我们将考虑如何利用  $z$ -value 来发现视觉上重要的对象。我们的目标是尽可能快速地从很大的数据库中确定可能是视觉重要的对象。这同以往的工作是有差异的(可以作为补充):以往的工作对简化单个复杂对象并保证最大的相似形更感兴趣。 $z$ -value 索引技术的良好特性可以用来确定视觉的重要性:

(1) 较大子空间的  $z$ -value 比较小子空间的  $z$ -value 要短。而大对象更可能由较大的子空间近似,因此能够快速地用  $z$ -value 的长短确定。这样就不需要在将所有的对象都读取出来后再比较大小,以确定较大的对象。

(2) 对于那些区域面积虽小但轨迹却很长的对象,例如道路河流,它们很可能由许多个  $z$ -value 近似,因此也是容易识别的。

(3) 由于子空间可以相互嵌套,当  $z$ -value 为  $z$  的子空间在另一个  $z$ -value 为  $z'$  的子空间内部时, $z$ -value 的一个特性是, $z'$  是  $z$  的前缀。这样,通过比较  $z$ -value 的前  $k$  位( $k$  可以由数据空间的空间扩展和显示区域的大小来获得),位于稀疏区域的对象可以很容易地识别出来。位于密集区域的对象也可以通过进一步的处理加以识别。

(4) 对于  $z$ -value 来说,有一个低层的坐标网格(grid)相对应,正是这个网格的大小和位置被编码为  $z$ -value。这样,代表性对象能够通过  $z$ -value 相应的低层网格(由视觉参数确定层次)选择出来。这比随机过滤更好,并且比计算这些对象的相对位置开销要小得多。

这样,对照视觉重要性的定义我们发现,虽然  $z$ -value 比它们表示的空间对象更简单,但已经包含了足够的信息以确定视觉重要性,而且上述的大部分操作很容易用 SQL 查询来表示,因此可以由 DBMS 高效执行。

通过上述技术,我们可以快速地确定视觉重要的对象。然而,对于一个视觉重要的对象,并不是每一部分都是视觉重要的。对于如何利用  $z$ -value 来识别视觉重要的部分,我们认为需要设计一些新的数据库存储结构,例如,当在一个子空间中对象所占的比例低于一个预定义的阈值时,能够忽略该部分的实际形状,而作为一个点返回。这个结构也将用于标记化,我们将在后面的相关部分进一步加以讨论。

#### 3.2 对象简化

空间对象的简化问题主要是,在尽可能保证对象简化前后相似的前提下,减少空间对象的细节层次。目前,

已有许多简化算法被提了出来<sup>[2]</sup>,尤其是单个多边形的简化已经被很好地解决了,其中一些算法不仅高效而且在维持数据相似形上也有较好的性能.然而,当这一问题涉及到一组对象时,情况就更加复杂了,因为还需要维持简化前后对象间的拓扑关系.例如,交叉的两条道路在简化后应该仍然在原来的交点处或附近相交,不能交叉到本不相交的位置;一条道路一侧的对象在简化后应仍然保持在同侧,而不能出现在另一侧.为了保证所有的拓扑约束在概化处理都能满足,许多对象需要经过至少 3 遍的处理:第 1 遍确定对象间的拓扑关系;第 2 遍应用概化操作;第 3 遍检查约束条件是否满足.如果不满足,这些步骤还需要用另一个参数集再次重复.

正如前面提到的, $z$ -value 对应于一个低层的坐标网格,因此我们可以将  $z$ -value 看做是向量空间数据用不同规模像素的栅格化(rasterization).像素的规模由近似对象的需要决定.这种对  $z$ -value 的理解能够更有效地处理空间操作,因为只有那些同样或彼此接近的像素(对象用像素来近似)才需要被处理.这能够有效地减少对开销更大的空间操作的需要.如  $z$ -value 可以用来迅速检查拓扑关系,虽然精确性不能完全保证.假设有两条线  $L$  和  $L'$ ,它们分别被两组  $z$ -value  $Z$  和  $Z'$  近似.我们能够想象,一条线的  $z$ -value 完全包住了这条线.当  $L$  和  $L'$  相交时,在  $Z$  和  $Z'$  中应该有一对  $z$ -value 是相同的或一个被嵌套在另一个内部,正如上面解释过的,这些关系都很容易利用  $z$ -value 检测.这样,对于简化后线条的  $z$ -value(这些  $z$ -value 可能与原值不同),仍然应该有一对  $z$ -value 是相同的或一个被嵌套在另一个的内部.这对于两条线简化的拓扑约束正确性是一个必要条件(虽然不是充分条件).显然, $z$ -value 的操作具有较低的开销,尤其当这些线是复杂线段时.这样,不正确的简化将能够以较低的开销被检测到.这种利用  $z$ -value 的新方法原则上类似于处理空间操作的标准方法——过滤-精化(filter-and-refine)<sup>[4,7]</sup>.简化后对于拓扑关系检查的处理能够局部化到线段的某个部分.

### 3.3 标记化

标记化是指将一个对象简化为一个点,所有细节都不出现.标记化是用来将一个很小但对于应用很重要的对象(如安全地图上的消防栓)显示为地图上一个标记的过程.对于那些需要被标记化的对象,其近似表示(如  $z$ -value)就已足够,因为对标记化来说,所需知道的全部信息就是那些对象的位置.其他空间细节,如几何形状,都是不重要的.

因此,如果一个对象(或对象的部分)的  $z$ -value 表示的区域比在显示设备上映射为一个点的区域还要小,那么该对象(或对象的部分)的细节能够安全地跳过(也就是说不需要从数据库中读取出来),如果该对象还很重要(由其属性决定),那么还需在其  $z$ -value 指示的位置放上一个标记.

### 3.4 多边形合并

多边形合并操作将一组多边形合并为一个新的多边形,它是原多边形的并集.对于在线空间分析处理、空间数据挖掘和空间数据可视化来说,这是一个重要操作.在这些应用中,多边形表示不同的空间对象,往往需要根据不同的标准或显示设备的限制被合并.一个直接的方法是将所有原多边形从数据库中读取出来,用计算几何的算法删除边界被不止一个多边形共享的那些部分.但当多边形数量较大时,该操作的处理开销极为庞大.考虑到并非所有的多边形都将影响合并后多边形的边界,我们提出了一个基于覆盖率的多边形合并算法<sup>[9]</sup>.它利用了  $z$ -value 确定内部多边形.如果子空间  $z$  覆盖的对象  $pid$  的  $z$ -value 索引项表示为  $(z, pid)$ ,我们加上覆盖率(occupancy ratio)  $\alpha$  对它进行扩充  $(z, pid, \alpha)$ ,也就是说,我们不仅记录哪些多边形被一个子空间覆盖,而且也记录每个多边形占据的比例.那些完全在一个被合并的多边形内部的子空间,能够简单地对每个子空间加上这些多边形的覆盖率.那些累计覆盖率为 100%的子空间是内部子空间.因此,只有那些和非内部子空间重叠的多边形需要从数据库中读取出来加以处理.这里,对象选择仍然可以用  $z$ -value.此外,一个小于 100%的阈值可以用来忽略那些较小的、在一个特别设备上不能被看见的合并多边形内部的空洞(这些对象不属于合并多边形).

这个基于覆盖率的  $z$ -value 扩展也可用于空间连接(join)的处理<sup>[10]</sup>.

从上述对简化、标记化和多边形合并的讨论我们可以看到,如果空间数据(空间数据的几何信息)按照  $z$ -value 分解结构分组存放,而不是像目前许多空间数据库实现的整块存放,将有利于一些主要性能的获得.换句话说,如果数据库的底层数据结构支持,利用  $z$ -value 可以找到一个对象的哪些部分需要被获取,那么只有这些数据需要从数据库中被读取出来.

## 4 结 论

空间数据的制图综合处理是空间信息检索中必不可少的一个步骤.与其他类型的数据不同,对空间数据的检索主要是为了可视化.由于并非所有的空间对象,或者说不是一个空间对象的全部信息都能被人眼所辨别,往往需要根据相对于应用的重要性,将一些对象过滤并转换为更加简单的形式.随着基于 Internet 空间应用的普及,有效的空间数据综合成为越来越重要的问题.在基于 Internet 的空间应用中,数据库操作一般都是最耗时的操作之一.如果能够利用优化的空间数据访问策略减少空间数据集,这个优势还将保持到其他的步骤,如数据简化和数据传输.本文基于此提出一个面向数据库的处理技术,通过在空间数据库内进行空间数据的概化处理以优化性能,而不同于以往大多将概化作为后数据库(post-database)的操作来处理.这一方法实现简单,开销小,并且还可用于其他更为广泛的空间操作,具有较好的应用前景.

### References:

- [1] McMaster, R.B., Shea, K.S. Generalization in Cartography. Association of American Geographers, Washington, D. C., 1992.
- [2] Jackson, C. Spatial data simplifications [Hons. Thesis]. University of Queensland, 2000.
- [3] Spaccapietra, S., Parent, C., Zimanyi, E. MurMur: a research agenda on multiple representations. In: Proceedings of the 1999 International Symposium on Database Applications in Non-Traditional Environments. Kyoto: IEEE CS Press, 1999. 373~384.
- [4] Weibel, R. Generalization of spatial data. In: CISM Advanced School on Algorithmic Foundations of Geographical Information Systems. 1996. 346~367.
- [5] Weibel, R.A. Topology of constraints to line simplification. In: Proceedings of the 7th International Symposium on Spatial Data Handling (SDH'96), 1996. 9A.1~9A.14.
- [6] Gaede, V., Günther O. Multidimensional access methods. ACM Computing Surveys, 1998,30(2):170~231.
- [7] Zhou, X., Krumm-Heller, A., Gaede, V. Generalization of spatial data for Web presentation. In: Proceedings of the 2nd Asia Pacific Web Conference (APWeb'99). Hong Kong: Computer Science Research, Education and Application Press, 1999. 115~122.
- [8] Oracle Inc. Oracle 8i spatial: experiences with extensible database. In: Oracle Technical White Paper, 1999.
- [9] Zhou, X., Truffet, D., Han, J. Efficient polygon amalgamation methods for spatial OLAP and spatial data mining. In: Proceedings of the 6th International Symposium on Large Spatial Databases (SSD'99). Lecture Notes in Computer Science 1651, Hong Kong: Springer-Verlag, 1999. 167~187.
- [10] Truffet, D., Orłowska, M.E. Two phase query processing with fuzzy approximations. In: Proceedings of the 4th ACM International Workshop on Advances in Geographic Information Systems (ACM-GIS'96). New York: ACM Press, 1996. 84~89.

## Spatial Information Retrieval and Database Generalization\*

LU Sang-lu<sup>1,2</sup>, ZHOU Xiao-fang<sup>3</sup>, CHEN Gui-hai<sup>1,2</sup>, XIE Li<sup>1,2</sup>

<sup>1</sup>(National Laboratory for Novel Computer Software Technology, Nanjing University, Nanjing 210093, China);

<sup>2</sup>(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China);

<sup>3</sup>(School of Computer Science and Electrical Engineering, University of Queensland, Australia)

E-mail: sanglu@nju.edu.cn

http://www.nju.edu.cn

**Abstract:** Spatial data are often stored in a database with the finest level of details. The problem of cartographic generalization of spatial data concerns deriving spatial data with proper level of details when the data are used for an application. With more and more computerized applications making use of shared spatial data sources, and the increasing number of applications emerging from Internet-based spatial applications, efficient spatial data generalization becomes a critical issue. In this paper, the issues in efficient spatial data generalization are investigated, with a unique focus on spatial database design and spatial operations processing. Several types of extensions and novel applications of a commonly used spatial indexing mechanism called *z*-values are proposed to achieve the goals.

**Key words:** spatial database; generalization; *z*-value

\* Received April 24, 2001; accepted January 14, 2002

Supported by the National High-Tech. Research and Development Plan of China under Grant No.2001AA113050