

一个基于知识的地理信息系统: KBGIS-II

T. 史密斯等

1. 简介 地理信息系统 (GIS) 可以看作一个数据库系统, 其中大部分数据有空间索引, 还有一组过程作用其上, 用于回答有关数据库所描述的空间实体的询问。Smith 等 (1987) 认为一个通用的大型 GIS 应该满足五点设计要求, 同时提出并描述了为满足这些要求而应采用的四条通用原则。本文描述的基于知识的 GIS (KBGIS-II) 正是按照这五点要求来设计和实现的。在详细描述 KBGIS-II 之前, 先简要叙述一下这些原则和要求。

设计和实现 GIS 的要求和原则 以往的研究表明, 大多数 GIS 的设计与实现应满足下列通用要求: (a) 能处理由空间索引数据组成的大型多层次异构型数据库; (b) 能在上述数据库中对大范围的空间对象的存在性、位置及属性进行查询; (c) 能处理交互系统的查询; (d) 系统配置灵活, 保证系统能容易地改装成满足各种特殊应用和用户的新系统; (e) 在系统使用过程中, 能用一种方法“学习”知识库中的空间对象。

可能性 (模糊论) 等。对于何时使用何种测量去评价数据的不精确性以及如何在 GIS 中处理它们的问题还有待于进一步研究。

5. 用户如何处理这些知识? 这个问题很大, 但我们可以限制在两个方面。一是与用户查询有关。因为我们应该去分析从数据库中提取数据的选择标准。用户可以按他们的选择标准或查询分组。

另一方面是有关数据处理算法。如数据的转换。因为原始数据常不能用, 必须进行转化。无论特定的转化是有意义的还是与数据的语义无关的, 象这样的算法在原理上取决于 FDS, 而使用的效果取决于算法的结构和使用的数据库结构。

七、地理信息理论的潜在作用 本文虽只提出了五个中心问题, 却引出了很广阔的讨论领域。这里只作了肤浅的解释。这些问题相当于打开潘朵拉之盒的钥匙。当然有些问题已有答案了, 但在这领域中还不多。发展信息理论主要基于以下考虑:

——好的理论能帮助 GIS 的用户了解他们要求的可能性或不可能性。另外理论对系统完成数据处理、或具体地说对选择各自的处理步骤也很有好处。

——许多 GIS 或系统的组成部分在市场上是有用的, 但用户通常不知道选择系统的标准。他们常常几乎任意挑选或根据他们同事选用的系统去挑选。另外也很难对系统能力进行标准评价。

——如果用户不买系统而是自己建立, 那么就必须制定设计标准和设计策略。系统最后必须进行设计标准检验。

——在这篇文章的开头就提到了新一代的地理信息系统不仅能处理数据, 而且要处理知识。因此我们必须清楚地确定怎样使用和应使用什么数据。

很明显, 地理信息理论对建立解决这些问题的方法是很重要的。当然我们不能等理论成熟以后去解决问题。但是理论方面的工作将会开拓我们的视野, 对完成这些工作是很有利的。同时好的理论系统对新一代地理信息专家的培养也是很重要的。

唐小龙, 冀译译自《ITC Journal》, 1989—1

为了便于设计和实现满足上述五点要求的 GIS, Smith 等提出了如下四条通用原则:

(a) 系统地运用在计算机科学各领域中开发的技术和方法; (b) 综合运用在计算机视觉、图象理解、数字制图和遥感等方面开发的方法和过程; (c) 应用各种过程来减少答复询问时的搜索开销; (d) 提供某些特征, 以便 GIS 可以容易地改装成适合特殊应用和用户的新系统。

在过去的二十年中虽然开发了大量的 GIS, 但是其中没有一个系统能完全满足上述要求。而我们设计和实现的 KBGIS-II 能完全或部分地满足这五点要求。同时, KBGIS-II 还遵循了为满足这些要求而给出的四条原则, 正因为这两点, 我们说 KBGIS-II 代表了新一代 GIS。该系统能处理用光栅或矢量形式表示的多层次数据集合。它能“一步”(从用户角度看)找到数据库中的复杂空间对象, 因为它所采用的搜索过程能将搜索开销减少到最低限度。用户可以用多种方式对系统加以扩充, 同时该系统的结构允许调用 KBGIS-II 中的其它系统。最后, KBGIS-II 还具有归纳学习能力, 允许系统自动修改知识库。

2. KBGIS 总观 本节概述了 KBGIS-II 的主要功能和结构, 以及为使系统满足上述要求所采用的方法。

(1) 系统功能 KBGIS-II 可以在用户控制下执行四个主要功能:

(a) 系统在查询方式下答复涉及空间对象的询问, 这些空间对象在数据库中常常用隐含形式表示。目前主要有两种查询方式, 它们的作用相反。第一种查询的一般形式 (FIND 位置[#情况][空间对象][空间窗口]):

当系统依据所要满足的各个实例, 在给定的窗口中找到所描述的空间对象的空间位置集合时, 查询便得到满足。空间对象是用下文中定义的空间对象语言 (SOL) 来说明的。相反的查询采用下列一般形式 (FIND 对象[空间窗口][对象类]):

当系统在指定的空间窗口中找到所有属于给定的空间对象类的空间对象时, 查询便得到满足。

一大类空间数据库查询都可以用查询方式 (1) 和 (2) 来表示, 其中包括有关各种决策任务的查询, 即寻找满足各种限制或最优条件的位置集合。譬如可以用第一种查询来解决推销员问题。另外, 通过进一步处理查询方式 (1) 和 (2) 的输出结果, 能够容易地满足一类更广泛的查询, 例如在给定区域中查询空间对象的统计结果。

(b) 系统在学习方式下修改和扩充知识库。在第一种学习形式下, 即在查询方式缺省的情况下, 系统用选出的新发现的空间对象子集的位置来扩充知识库。在第二种学习形式下, 系统首先归纳学习如何定义新的空间对象, 然后将新对象的定义及有关信息加入系统知识库中。第二种形式目前需要用户干预。

(c) 用户可以在编辑方式下修改和扩充空间对象语言 (SOL) 以及与之相联系的过程, 还可以修改系统的知识库。

(d) 在跟踪方式下, 用户可以对系统执行过程的每一步进行跟踪。可以在查询, 学习和编辑方式下激活跟踪方式,

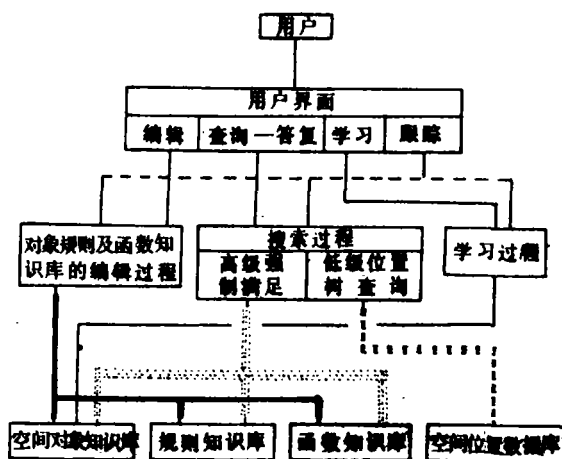


图1 KBGIS-II 结构图

本文中不再作进一步讨论。

(2) 系统结构 图1给出了系统的基本结构。其中用户界面是控制系统I/O行为的普通模块,含有用户查询分析。四个过程集分别对应于系统的四个主要功能。函数知识库包含定义SOL的函数知识,可以被用户修改。空间对象知识库包含空间对象知识(如空间对象的定义和各种启发性知识)而位置树数据库则包含了基本空间数据的层次。

(3) KBGIS-II与GIS要求 要使系统能够处理大型多层次数据库,既要依赖于软件系统,又要依赖于运行该系统的硬件。要使系统能响应对复杂空间对象的查询,就要依靠SOL所采用的搜索过程及知识库和数据库的结构。要想提高搜索效率,就要依靠搜索过程和选择的数据结构,而要求系统具有灵活性,就要依靠可供用户使用的编辑器。遵循上述要求来设计软件,必然造成目前采用的硬件(VAX11-750)不太适合于所承担的任务,而且使得在交互速度下所能处理的数据库大小受到限制。

3. 空间对象语言 在描述图1中给出的系统各组成部分之前,有必要先描述一下KBGIS-II中用于表示对象的空间对象语言(SOL)。选择SOL很重要,因为:(a) SOL定义了系统可以学习和查询的空间对象类;(b) 选择SOL有助于容易地执行各种计算任务(如搜索);(c) SOL可以展示系统数据库中空间对象搜索问题的计算复杂性。在本节中, SOL是用它表达空间对象的能力来描述的。

SOL的一个重要特点是给用户以灵活性。如同其它类似的基于谓词演算的语言。SOL语法相当简单,推理机制也是众所周知的。但是,用户有权选择对大量谓词、函数、变量和常量的定义,以便提供适合于给定空间区域的表达能力很强的语言。

(1) 定义SOL 我们将空间对象定义成一个空间位置集合加上一个刻划这些空间位置特征的属性集合。在其最基本形式中,位置被定义成由最小空间单元,或者称为“像素”的某种汇聚组成的集合,这些像素将数据库中表示的区域分割开来。位置不一定是一组相连的像素。位置的这种定义可以扩充到包含多个位置集合。

定义SOL要用到三类属性:(a) 像素属性,或记作PPROP,用于刻划数据库中单个像素的特征。空间数据库的每一层函数对应一个PPROP。PPROP的例子有土地利用、地理及海拔高度。显然,土地利用,地理及海拔高度都刻划了单个像素或某个像素集中的每个像素的特征;(b) 像素组属性,或记作GPROP,用于刻划构成某一位置的像素集的特征,但它并不描述该像素集中单个像素的特征。GPROP的例子有大小、形状及方向;(c) 关系属性,或记作RPROP,用于描述两个位置之间或两个位置的属性之间的关系。RPROP的例子包括距离、方向及面积。

在SOL中,我们用刻划所给空间位置集合特征的三类属性的合取来描述空间对象。这三类属性可以用谓词来表示,谓词的解释依赖于一个空间位置与一组属性值之间的关系,或两个空间位置与一组属性值之间的关系,或两个空间位置的属性值之间的关系。PPROP和GPROP这两种属性可以用如下形式表示: $EQUAL[(U - FUNCTION LOC1) VAL]$; 而属性RPROP既可以表示成: $EQUAL[(B - FUNCTION LOC1 LOC2) VAL]$, 又可以表示成: $EQUAL[(B - FUNCTION <LOC1的函数> <LOC2的函数>) VAL]$ 。在上述各定义中, LOC1 是一个表示位置的常量或者变量; VAL是表示某一属性值的常量或变量; U—FUNCTION是位置的一元函数; B—FUNCTION是两个位置的二元函数; EQUAL是指明语句真假值的谓词。

下面给出这三类属性的实例。

(a) 描述一个土地利用属性值为农业的位置的PPROP谓词为: EQUAL[(土地LOC 1) 农业], 当变量LOC 1约束为某个位置 (即一组空间索引), 该位置的每个空间索引的土地利用属性值为农业时, 该谓词得到满足。根据空间数据库某一适当的层次上存贮的信息, 可以确定PPROP谓词的真假值。

(b) 描述一个面积为50到60的结果单元的位置的GPROP谓词为: EQUAL[(面积LOC 1) (50 60)], 若变量LOC 1约束为一个面积为50到60个像素的位置, 则该谓词得到满足。GPROP谓词的真假值可以用计算或存贮信息来确定。系统为每个GPROP准备了一个计算相应属性值的函数。

(c) 描述一个由距离为10到20结果单元的两个位置组成的对象的RPROP谓词为: EQUAL[(距离LOC 1 LOC 2) (10 20)], 当约束为LOC 1和LOC 2的两个位置被10到20个单元隔开时, 该谓词为真。系统用一个函数来计算每个RPROP的属性值。

该语言也允许对两组空间索引作关系比较, 用到的数学比较运算有EQ、GT、LT、GE 和LE, 分别对应于=、>、<、≥和≤。举个例子, 假设一个空间对象的某个组成部分的面积大于其他组成部分的面积, 则可以写成:

EQUAL{(GT(面积LOC 1) (面积LOC 2))TURE }

上面讨论的任何谓词都可以用逻辑连接符 ∧ (与) 和 ∨ (或) 来连接。如果想把逻辑否定 (¬) 与任意谓词PPROP或GPROP联系起来, 只要用谓词NDT-EQUAL代替上述表达式中的谓词EQUAL。有一个居住带 (LOC 2) 环绕商业中心的城市模型, 这个简单例子可以用SQL语言来描述: EQUAL[(土地LOC 1) 商业区]。

∧ EQUAL[(面积LOC 1) (30 40)]; ∧ EQUAL[(土地LOC 2) 居住区]; ∧ EQUAL[(面积LOC 2) (50 60)]; ∧ EQUAL[(包容LOC 2 LOC 1) TURE]。

这里要强调一点, 系统中表示空间对象的函数和自变量集合, 可以由用户通过各种编辑器来定义。

(2) 空间对象层次 利用“TYPE”这一特殊的GPROP, 可以定义高层空间对象, 即用基本的P-、G-和R-PROP定义的空间对象。于是, 用户可以对空间对象进行编辑排序, 从而在空间对象上引入层次结构的概念。TYPE最简单的应用是给一个空间对象赋名, 该空间对象被定义成带有特定值的PPROP、GPROP 和 RPROP的合取。下面是一个高层次空间对象的例子: [(TYPE X) GEOL-OBJ 1]。

⇔ ∧ [(土地 X1) 森林]; ∧ [(面积 X1) 大]; ∧ [(形状 X1) 图形]; ∧ [(地理 X2) 4]; ∧ [(海拔 X2) (50 100)]; ∧ [(面积 X2) 中等]; ∧ [(距离 X1 X2) (60 100)]; ∧ [(方向 X1 X2) 北]。(谓词EQUAL是隐含的, 在该语句中省略了)。该定义的含义为: 任意两个满足语句右边给出的一元或二元限制的位置X1和X2的集合, 都构成高层对象GEOL-OBJ 1的一个位置X。可以用某种适当的方式来选择位置X与X1、X2之间的关系。例如, X可以是X1与X2的凸面, 可以是X1与X2的逻辑和, 也可以是X1与X2的中心。位置X1上的一元限制由面积和形状这两个GPROP函数给出, 而位置X2上的限制则由面积这一GPROP函数来表示。位置X1和X2上的限制由距离和方向这两个RPROP函数来指定。

一般情况下, 可以利用属性TYPE, 根据其它高层次对象与其它PPROP、GPROP及RPROP的合取来定义高层次空间对象。

利用属性TYPE给高层次空间对象赋名, 可以达到两个目的:

(a) 它提供了一种用已定义的对象定义新对象的方便的速记符号。因而, 如果已定义

了两个对象, 名字分别为LAND-1和LAND-2, 就可以定义一个新的高层次空间对象LAND-3: $[(\text{TYPE } X) \text{ LAND-3}]$ 。

$\Leftrightarrow \wedge [(\text{TYPE } X1) \text{ LAND-1}]; \wedge [(\text{TYPE } X2) \text{ LAND-2}]; \wedge [(\text{距离 } X1 \text{ } X2) (20 \text{ } 30)]$ 。

(b) 属性TYPE可以将新发现的高层次对象位置存贮在以对象名和位置为索引的数据库中。位置索引是用辨别网来实现的, 每个高层对象都有自己的辨别网。这些数据结构将在下文中描述。

于是, 任何高层次空间对象都可以看作一棵树的根结点, 这棵树完全伸展后产生的叶结点是PPROP, GPROP和RPROP。基于这种观点, 每个高层空间对象的复杂性量度都可归结为考虑以该对象为根结点的树的高度, 该树上每一层对象成分的个数, 以及每一层上空间关系(谓词RPROP)的复杂性。

从描述空间对象搜索过程这一目的出发, 区别高层空间对象与原始空间对象非常方便。如果在空间对象数据库中定义了一个对象, 它的名字为TYPE的属性值, 则将该对象归为高层次空间对象。而原始空间对象的概念是指任何用PPROP的合取表示的象素的连续集。由此可见, 任意高层次空间对象最终都可以用原始空间对象及一组恰当的RPROP和GPROP来定义。

4. 用户界面 该用户界面允许用户选择系统的四个主要功能(查询、编辑、学习和跟踪), 并提供适当的输入、输出。目前, 该系统的大部分用户输入采用键盘方式, 而系统查询的输出则显示在图形设备上。

(1) 查询 在查询方式下, 用户可以在两种基本查询(1)和(2)中任意选一个。要进入这两种类型的查询, 既可以通过交互方式, 也可以采用文件方式。

(2) 编辑 在编辑方式下, 用户既可以用函数编辑器修改SOL及与之有关的过程, 也可以利用对象编辑器来修改系统的知识库。

(3) 学习 在学习方式下, 用户可以让系统通过所给的例子来学习一个新空间对象的定义。这些实例既可以由系统搜索生成, 也可以由用户提供。

5. 知识库与数据库

(1) 空间对象知识库 空间对象知识库存贮系统已知的所有对象的定义及有用的信息。该知识库是通过槽与填充数据结构及辨别网数据结构实现的。有关对象定义, 启发性搜索, 对象分类及对象复杂性的信息, 以及在搜索空间对象时直接引用的低层搜索过程都存贮在该知识库中。而有关已发现的空间对象已知的位置信息则存贮在辨别网数据库中。

表1给出了存贮在槽与填充数据结构中的槽名和信息。每个槽中的信息可以借助空间对象知识库编辑器来增加、修改或删除。而存贮在该数据库中的信息可以在系统的归纳学习方

表1 空间对象的数据结构

槽 名	内 容
特征类型	线性和基于区域之间的特征差别。
父 定 义	该对象的定义, 用空间对象语言描述的析取范式。
定 义	用该对象定义的一组较高层次对象。
启发性信息	位置与该对象位置及该对象的空间关系性质都是上下文有关的其他对象。
复 杂 性	搜索该对象的新实例的复杂性量度。
大 小	该对象的线性与区域大小的近似值。
过 程	可直接在图象上操作, 而无需求助于对象定义的低层算法的指针。

式下进行修改。

辨别网数据库用于存贮系统在回答用户询问的过程中产生的空间对象已知实例的位置。每个对象都有自己的辨别网。辨别网的核心是数据库的空间对象名和想要的位置树地址。

(2) 函数知识库 函数知识库存贮系统在搜索空间对象时用到的函数信息。用于测量空间对象的GPROP与RPROP属性值的函数信息存贮在该知识库中。

用户可以借助函数编辑器来增加、修改和删除数据库中的信息。每个GPROP和RPROP函数都存贮有各自的有关传播限制能力的信息, 计算复杂性, 子过程名, 对称性, 范围, 以及学习与学习有关的信息。表2展示了在知识库中存贮的槽名和信息。系统正是用这些信息来控制空间对象的搜索, 并在学习方式下产生新信息。

表2 函数数据结构

传播性	指出在GPROP和RPROP情况下, 函数能否在搜索过程中进行转化, 以便传播限制。
复杂性	函数计算复杂性量度, 用于计算用恰当的GPROP或RPROP定义的空间对象的复杂性。
对称性	是一元(RPROP)函数对称属性信息, 允许搜索控制根据动态对象优先改变自变量。
范围	用相应的GPROP或RPROP定义对象时由用户根据需要指定的值的性质。
子过程	在系统管理中由函数调用的子过程名。

(3) 位置树数据库 位置树数据库存贮了基于区域的PPROP及线性特征的空间分布信息, 这些信息属于数据库覆盖区域的位置信息。

A. 区域数据 基于区域的PPROP的原始输入可以构造位置数据库, 它由每一层(如土地利用、地质或海拔高度)上的光栅影图组成。用于数据存贮的概念数据库模型为四叉树结构。这种数据结构的基础是将空间递归地分为四个象限, 该结构在有关文献中已作过广泛的讨论。位置树数据库允许对主题信息的多个层次进行编码, 以便扩充四叉树概念。位置树内部结点的每个层次上存贮的信息不止一种。PPROP可以表示土地利用、地质、海拔高度等原始象素属性, 位置树有不同的层次对应于数据库中每一个这样的PPROP。位置树中的结点是三维框架结构。数据库中每个PPROP配有一个槽。因而每一层(槽)都是一个包含下列各槽的框架:

(a) 值槽: 存贮出现在结点表示的区域中的数据值。每个PPROP最多有50个离散值。在树的各个中间结点上存贮了出现在该结点下面的一组值(包括每个值的区域范围)。在数据值被存贮到Tanimoto和Pavlidis(1975)所描述的锥形数据结构中之前, 不对数据值求平均值。得到每个数据值的区域范围后, 就可以动态地计算结点的颜色。从而就可以根据一个专用的数据值将结点分为黑、白或灰三种, 该数据值由可变的百分数门槛值决定。

(b) 分布槽: 存贮用结点表示的区域中每个数据值的区域范围信息。分布槽可用于存贮不止一种描述各种数据值分布的统计。可以根据分布槽中存贮的信息, 采用灵活的标准来计算结点的颜色。

在满足查询的搜索过程中, 用搜索标志来标记搜索过程所访问的各个结点。这些搜索标志字段的定位是在搜索时产生的动态过程。作为一个查询部分的每个原始对象(连续区域)有唯一的搜索标志字段。搜索标志字段中存贮的信息只在动态查询范围内有效。查询结束后可以删除该字段, 并释放该字段所占的空间。

B. 线性数据 基于线性数据的原始输入由各线性特征(如道路、河流)的二维光栅图象组成。这些数据通过边界条件过程转化为矢量形式, 该结果矢量以空间索引形式保存, 作为

位置树数据库的高层结点属性。每个线性特征的矢量表示由一系列直线段构成。这些线段存放在数组中,只有光标可以识别段与段之间的分点。正是这些存贮在位置树数据库结点中的光标,确保有效地找回数据库中任意指定区域内的溪流子集或其他线性特征。

6. 编辑器 KBGIS-II 有两种编辑器,即函数编辑器和空间对象知识库编辑器(简称对象编辑器)。前者允许修改函数知识库,而后者允许用户修改空间对象知识库。这些编辑器都是菜单驱动的,用户可以选择下列五种方式的任一种来改变知识库。

(a) 在增加方式下,对象编辑器可以产生一个新的空间对象。它首先询问用户新对象的“特征类型”。然后调用对象定义包,指导用户用 SOL 构造对象的“父定义”槽。除了对象的定义外,编辑器还询问其它的信息,如类型、启发性信息、线性和面积大小。在这种方式下,通过函数编辑器可以增加一个新的 GPROP 或 RPROP。由用户指定含有函数定义的文件名。如果新函数能传播限制,则文件中要含有一个能返回新搜索窗口的函数。除了函数定义外,编辑器还询问有关的参数,如复杂性、对称性及区域等。

(b) 在删除方式下,对象编辑器可以删除知识库中的空间对象。系统只允许删除当前不构成知识库中其他任何空间对象定义的成分的那些对象。函数编辑器可以删除函数知识库中的 GPROP 和 RPROP。

(c) 在修改方式下,用户可以修改所选空间对象的槽或一个函数的槽的内容。系统在允许用户修改之前,保证逻辑一致性。

(d) 在显示方式下,用户可以浏览知识库,检查所选择的对象或函数的所选择的成分。

(e) 在帮助方式下,可以为用户使用编辑器提供帮助。

(f) 在结束方式下,用户可以保存当前的改动。

7. 空间搜索

(1) 搜索原则 从前面的讨论可以清楚地看到,搜索空间对象的过程既位于一般 GIS 的中心,又位于特殊的 KBGIS-II 的中心。Smith 和 Peuquet 总结了五条原则以强调 KBGIS-II 的搜索过程,在此重申一下这些原则,并增加一条:(a) 在数据结构及用于这些数据结构的搜索过程中使用层次分解;(b) 提供不同的搜索策略,以便可以针对给定的搜索情形,选择效率最高的策略;(c) 应用最先最优搜索过程,以便运用领域知识减少回答询问时所需搜索的位置集合;(d) 使用强制——满足方法;(e) 使用循环;(f) 为满足查询,动态地修改系统的知识库。在下文中具体描述搜索过程时,将看到这些原则的应用。

(2) 搜索过程 为方便起见,先简要概述一下基于上述 6 条原则的搜索过程,该过程应在 KBGIS-II 中,以满足查询类型(1)。首先由用户输入一个查询,搜索过程开始分析该查询,并检查其语法正确性,同时用户时刻做好修改准备。然后系统将查询(高层)对象改为用语义网络来表示,语义网络中的联系反映了要满足的查询子对象之间的 RPROP 关系(或限制)。该网络随着启发性知识而增大,同时结点上的子对象按序排列。再将强制满足过程按指定顺序作用到各结点上。这时在系统知识库中首次出现对满足关系限制和空间限制的已知子对象的查询。如果这次查找不能满足用户的查询,则要在结点的子对象上循环调用该搜索过程。当搜索到系统的位置树数据库时,该循环过程结束。当数据库的某一搜索使查询最终得到满足,而且该搜索的计算开销相当大时,将结果存贮到系统的知识库中,以便在以后的搜索中使用。

在上述搜索过程中,利用强制满足过程来满足在定义对象时用到的所有一元(GPROP)和二元(RPROP)限制,从而给出了空间查询方法的精髓,许多科研人员研究了一般的强制

满足问题 (CSP), CSP可以表述为: 假设有 m 个变量, 每个变量有一个相应的取值范围, 另外还有一组限制关系, 每个限制关系包含一个变量子集, 要求找出所有可能的 m 元元组, 使得每个 m 元元组都是一个满足限制关系的 m 个变量的例化。

这里只考虑了一元或二元限制关系的离散和有穷的CPS。

传统的CPS方法要用到回溯。将变量按连续顺序进行例化, 例化时要用到从取值范围的连续表示中选出的标签。因此回溯对应于联合搜索空间的深度优先搜索, 并通过测试中间谓词的真假值, 尽早地终止不成功的搜索。一旦某个谓词的变量被例化, 便可测试出该谓词的真假值。如果值为真, 则继续测试和例化过程, 否则将搜索过程退回到最后被例化的, 并且在其取值范围内仍有未试过的值的变量。再将该变量例化为它的下一个值。

回溯虽然有其内在优点, 即用单个失败来消除生成和测试搜索空间的实质部分, 但它效率很低。现在已经提出多种改进这种搜索过程的建议, 例如对网络的结点、弧和路径的一致性处理。以及利用预看过程削减搜索空间的向前测试树搜索。

空间搜索的重要特点是, 通过传播空间限制, 有效地削减变量的取值范围。由于空间关系隐含在数据组织中, 所以这是可以实现的。在下一节中要更详细地讨论这种空间搜索方法, 首先讨论控制搜索的高层搜索过程, 然后讨论强制——满足过程, 最后讨论搜索位置树数据库的低层搜索过程。

(3) SOL和搜索过程 我们可以通过SOL与搜索过程的关系来看SOL的结构。首先, 由于采用了只包含一元 (PPROP, GPROP) 和二元 (RPROP) 关系的语言, 故允许用邻近结构表示空间对象的语义网络。这类表示有一个通用的空间解释, 并提供一个能正常运用强制满足技术的数据结构。其次, 由于在SOL中使用了谓词TYPE, 因而允许在满足查询的过程中按通常方法进行循环调用。

(4) 空间对象搜索的复杂性 如上所述, SOL可用于刻划空间对象搜索的计算复杂性, 例如搜索某一对象所需的计算时间可以用测量对象大小的函数来表示。下文中的一个简单而有启发性的讨论告诉我们, 一般说来空间对象搜索是一个很困难的计算问题。用一个例子可以说明, 采用上文定义的具有高度搜索复杂性的SOL, 能够容易地构造表示形式极为简单的空间对象。

一个空间对象可以包含几个子对象, 这些子对象以一定方式连接, 形成一个关联图; 同时可以用子对象的数目(n)来衡量空间对象的大小。子对象之间的联系可以用某种RPROP来表示。我们进一步假设每个子对象的特征可以用呈现等概率的两个值的某种GPROP来描述。如果假定各子对象随机分布于空间数据库中, 那么任一指定位置满足某个GPROP限制的的概率为 $1/2$ (因而构成一个对应于子对象的实例)。于是在由RPROP指定的配置中, n 个位置都满足GPROP限制的的概率为 $(1/2)^n$ 。在不提供预处理, 并假定子对象在空间数据库中随机分布的情况下, 有必要检查每个位置的 n 元元组 (位于RPROP指定的配置中), 以检测GPROP限制是否得到满足。在寻找给定GPROP值的一组结点的对象之前, 还要搜索平均 $O(2^n)$ 次。另外, 在某些情况下, 搜索可能要花较长的时间, 而用SOL容易表示更加复杂的对象。

如果有搜索过程的附加信息, 则可以减少搜索时间。若子对象在数据库中不是随机分布的, 则可以通过预处理以及 (或者) 在搜索过程中建立的有关对象分配模式的启发性知识来形成附加信息。上例中的启发性知识可以由每个子对象的存贮窗口组成, 在每个子对象上有一个位置的概率高于数据库中其他位置的概率, 该位置满足GPROP (PPROP) 限制, 并使

之成为相应子对象的实例。类似地，父空间对象的存贮窗口将用该窗口表明，位于由RPROP指定的空间配置上的位置的 n 元元组满足GPROP限制的概率更高。因此在该存贮窗口内部的子对象位置之间存在一种可开发的相关性。

尽管这种预处理过程可能提高搜索速度，但由前面的讨论必然得出如下结论：用SOL描述的任意空间对象的搜索，是一个具有高度复杂性的问题。

8. 高层对象搜索 高层对象搜索过程用于搜索任意高层对象的位置，可以满足类型(1)的查询。首先调用该过程查找“查询”对象的实例。如果该“查询”对象的后代中还有其他高层对象，则递归调用该过程。搜索过程的递归层数不受限制。

空间搜索的第一步是利用已知的对象位置知识来减小搜索窗口的大小。其实现方法是，通过访问对象知识库，找到与搜索的目标对象上下文有关的其他高层对象。然后系统判断搜索窗口内是否存在这些辅助对象的已知实例。如果答案肯定，则在每个辅助位置周围建立子窗口，并作为搜索的可能区域。这样便建立了一系列窗口，系统依次在所有这些窗口中搜索目标对象，直至找到规定数目的该对象实例。对于任意一个窗口，高层对象搜索过程可以用两种方法完成该任务：

(a) 根据已知实例的空间索引知识库可以找回空间数据库的指定窗口中该对象的已知位置。存贮在该知识库中的已知位置集合并不完全，它依赖于过去搜索的历史。在任何时刻，该集合通常只包含以隐含形式存在于空间数据库中的部分对象实例。

(b) 通过窗口中的搜索过程可以发现该对象的新位置。搜索具有 m 个子对象的高层对象的新位置，一定要找到 m 个位置，分别对应于高层对象的 m 个子对象，以保证这组位置满足定义父对象的所有一元和二元限制。如果该查询要求找到 n 个父对象实例，那么必须找出 n 个集合，每个集合包含 m 个位置。下一节将给出KBGIS-II中实现的强制满足过程的设计细节。

在强制满足过程中确定 m 个子对象中的任一个子对象采用哪个候选位置的任务，是确定高层空间对象位置这一任务的循环说明。循环终止于确定原始空间对象位置的任务。由于未存贮这些对象的已知实例，故总是通过搜索空间数据库来确定它们的位置。该搜索包含确定满足一个谓词PPROP析合范式的一组连续的四叉树块，它是通过一个合适的区域增长过程实现的。后面一节将具体给出这种原始对象搜索过程。

高层对象的搜索可以用图2中展示的树来表示。例如，考虑在一个窗口中找出根 O 这一高层对象的新位置。假设已用启发性知识对上面描述的窗口大小进行了限制。一个父对象的子对象数目不受限制，随对象的TYPE而改变，但在本例中个数取3。作为强制满足过程输入部分的子对象 o_1 ， o_2 和 o_3 的位置可以是已知实例的空间索引数据库中的已知实例，或搜索发现的新位置。

例如，搜索 o_1 的新位置是就要递归应用该任务，这时 o_1 是父对象，而 o_{11} 、 o_{12} 和 o_{13} 是子对象。例如，递归终止于 o_{11} ，这是一个原始对象，并且可以直接

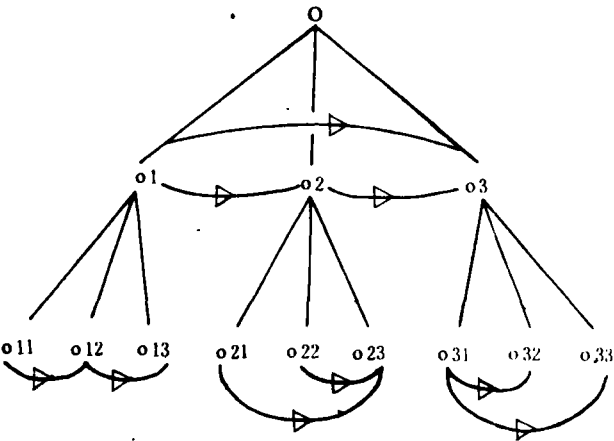


图2 高层对象搜索

在空间数据库中找到。

除了存在特殊目的搜索过程的情况下，上述过程之后都接着搜索所有定义的高层对象的新实例。有关这些过程的信息都存贮在空间对象知识库中，可用于控制过程。在这些情况下，直接调用特殊的搜索函数。如果要查询的对象是正在搜索的某个父对象的子对象，则将返回的实例交给强制满足过程。这就是搜索具有线性特征的被定义对象的方法。这种连接外部搜索过程的能力，使系统能使用高效的特殊一目标算法来搜索用户定义的对象。在这些情况下，用户可以通过函数编辑器为系统提供有关特殊一目标函数的知识。

9. 强制满足 图2展示了在层次结构中任意中间层次上的强制满足任务。具体地说，就是把该过程看作子对象 o_1 , o_2 和 o_3 上的操作。子对象同时受一元（GPROP）和二元（PPROP）限制的约束。父对象 O 上的高层对象搜索将其定义转换成一个如图2所示的语义网络，而这个以 o_1 , o_2 和 o_3 为结点的网络又传给强制满足过程。每个结点都通过空间关系（限制弧）连接到它的兄弟结点上，并立即通过父结点和子结点连接到在层次结构中位于该结点之上或之下的结点上。当在 o_1 , o_2 或 o_3 的任一个结点上递归调用该搜索过程时，才生成子结点。强制满足过程只与空间关系有关，并在兄弟结点（如 o_1 , o_2 和 o_3 ）的集合上执行。

可以将这种空间对象的强制满足问题映射到上文中描述的一般强制满足问题。变量表示一个父对象的 m 个子对象的位置，而每个变量的取值范围是子对象候选位置的集合。空间搜索问题的一个特点是：强制满足过程可以拥有关于变量取值范围（每个子对象的候选位置集合）的部分知识。

空间强制满足策略可以在搜索子对象的新位置的递归方法与搜索已找到的子对象位置的一致性分配的回溯方法之间作出选择。在搜索任一子对象的新位置的过程中，都是从已发现的其他子对象位置开始传播空间限制，以减少搜索空间。在任何时候，强制满足过程都是在窗口内每个子对象的已找到的位置子集上执行的。该过程搜索这一空间的目的是找到一致性分配。如果搜索过程失败了，下一个任务就是搜寻可以分配给子对象的更多的标志。选择搜索哪一个子对象，以及选择在原始窗口的哪个子窗口中进行搜索，都是为了最大程度地提高发现对应于父对象位置的一致性分配的概率。一旦找到了某些子对象的新位置，就在扩大了变量取值范围上继续强制满足过程。该过程就在强制满足与搜索新的子对象位置之间来回切换，直到找出所需数目的对应于父对象实例的一致性分配，或者该过程宣告失败。

10. 原始对象搜索 原始搜索过程的任务是确定原始对象位置的指定数目。原始对象的每个例子对应于搜索窗口中一个连通区域。原始对象表示成只含 PPROP 谓词的合取范式形式。下面给出这种表达式的例子： $\{(土地\ X)\ (10\ 11)\} \vee \{(地质\ X)\ (1\ 2)\}$;
 $\wedge \{(\text{海拔}\ X)\ (50\ 90)\} \vee \{(\text{方位}\ X)\ (30\ 40)\}$ 。

依据所要求的任务，原始对象搜索过程有两种策略可以选择：（a）如果寻找原始对象的少量单独的实例，则使用通过种子扩张（SEEP EXPANSION）增长的区域；（b）如果在一个窗口中寻找某个原始对象的全部实例，则使用通过相连成分标签（CONNECTED COMPONENT LABELLING）增长的区域。

对于每种策略，原始对象搜索过程还可以在位置树数据库中选择一个截断的分解层。在该分解层上每个结点都标为黑色或白色。

根据满足特定谓词 PPROP 的结点面积，针对具体的原始对象，可以将位置树数据库上的每个结点分为白色、黑色或灰色。

位置树上的每个结点依据它在树中的高度,相应地有个面积。设 N 为位置树的层数。然后设层 N 上有个结点,若把 N 层看作最低层,则该结点高度为 0 ,面积为 1 个单元(像素),高度为 H (层: $N-H$)的结点,面积为 $(2^H)^2$ 个单元。

要选择满足谓词的面积,就要采用下列方式的严格限制。对于面积为 Y 个像素的结点,只有当它具有大于 $(Y-X)$ 个满足谓词的像素时,才能把它看作黑色;当它具有 X 到 $(Y-X)$ 个满足谓词的像素时,它为灰色;而当它具有小于 X 个满足谓词的像素时为白色。这种判定规则保证对应于某个层次的面积为 X 个单元的结点只可能是黑色或白色,以防止区域增长算法进入树的更深处,并将分解固定在所需层次上。选择 X 为 0 可以使搜索在完全消解的情况下执行。

如果一个过程只要求树中一个单层,比如没有父—子链的光栅锥,则可以采用下列选择规则。处于某些分解层上的结点,如果其面积的百分之 X 满足指定的 **PPROP** 谓词,则把该结点看成黑色的,如果满足谓词的面积为 0 到百分之 X ,则被视为白色。这种规则可使每一层被独立地看成所要的分解层上的一个光栅。

原始对象搜索过程的第一步是选择一个适合于区域增长的策略及一个合适的分解层。策略及分解层的选择依赖于:(a)要求的实例数目;以及(b)与搜索窗口有关的所求目标的平均大小。调用原始对象查询过程的强制满足过程通过涉及原始对象及已搜索过的查询对象的其他子对象的二元空间关系的传播,来减小搜索窗口。在这种方式下,要特别引起重视的是传播**RPROP**限制,来调用原始对象搜索过程。种子扩张的第一步是在搜索窗口中系统地搜索初始种子。该搜索是利用基于对象大小的启发性知识来完成的,它指明了黑色结点可能出现的深度。这种启发性信息可用于控制种子的搜索,在所选定的深度上将树的深度优先搜索转换成广度优先的搜索。一旦找到一个种子,它就使用种子扩张过程来增长,该过程能在窗口中找到完整的区域范围。随后的过程保证返回增长区域的最大块表示。该过程反复执行,直到长出所需的种子数。

每个访问过的结点都用一个查询标志来标记。允许上述过程系统地搜索和生长种子,直至整个窗口都搜索完毕,或已找到所需的实例数。

如果要对对象进行彻底搜索,就要在搜索窗口中应用相连成分标签算法。这是传统的使用四叉树数据结构的团状加色区域增长算法的一种应用。自顶向下地应用该过程来标记黑色结点及合并相连成分。只有当发现灰色结点时,该过程才进入下一分解层,并且只考虑该灰色结点的子结点。用这种方法,可以最大程度地利用位置树数据库的层次树结构。当分解层的所有结点均为黑色或白色时,该过程就不能用进入更深层了。该过程返回搜索窗口内所有相连区域。

11. 学 习 在**KBGIS-II**中实现学习过程的主要目的是缩短查询搜索时间,实现方法有两种,或者记住前面的搜索结果,或者学习更准确的定义对象,以便迅速减小搜索空间。因此学习可以分为死记硬背式学习和归纳学习。

(1) **死记硬背式学习** 死记硬背式学习允许系统记住已搜索过的对象实例,因而当再次要求搜索同一对象时,系统可以取出以前的例子,而不必再搜索。系统只存贮预定义的高层对象。

已知实例存贮在由每个已定义的空间对象的辨别网组成的单独的数据库中。该数据库是空间对象知识库的一个组成部分。用于存贮实例的辨别网基本上是个基于指针的四叉树。实例存贮在最小包容块中,即完整包容该实例的最低结点。每个对象存贮在不同的辨别网中。

该数据库还有另外一种辨别网,称为对象树,用于存贮根据位置索引的对象名。如果对象X的名存贮在结点Y中,则暗示在四叉树窗口内的位置树数据库中存贮对象X的对应于Y的一个或多个实例。这种信息在回答类型(2)的查询时很有用。

要回答对四叉树窗口中某个对象位置的查询,就要返回存贮在查询结点下的子树中的全部实例。如果低层搜索返回对象的一个新实例,则把它加到合适的辨别网中,同时修改对象树。显然,并非所有实例都能保存起来,因为空间需求是单调增加的,因此,找到一个新实例后,系统必须决定是否应该存贮该实例,决定依赖于各种因素。如果一个对象复杂程度低,能容易地搜索到,因而不存贮到对象库中。如果一个对象是利用其它高层对象递归定义的,则必须决定是否应该存贮于对象或父对象。此外,决定还取决于由其子对象重建该对象的开销。除了复杂性,另一个存贮实例的标准是对象的查询频率。

存贮对象有二种方式,或用精确位置,或用矩形近似。一个对象的近似可以用指定的面积、离心率、中心和方向来表示。

(2) 归纳学习 归纳学习的作用是根据一组已知的实例,给出一个对象的新定义,它可以提高对象搜索的效率。为了学习一个对象的新定义,既可以由用户给出实例,也可以由系统自己产生新的实例定义。因为在实例生成过程中不可能包容所有的 PPROP, RPROP 和 GPROP, 用户须指定适当的值,系统则利用这些属性产生定义。

KBGIS-II 的归纳学习子模块以 INDUCE (Hoff 等1983) 为基础。INDUCE 是具有一般目的的归纳学习程序,它接受一组输入规则,然后产生一条或多条与输入规则一致的更简单更一般的输出规则。给定一组输入规则之后,INDUCE 首先找出最有希望的子句(是通用的),再把新子句加到这些子句中,直到获得事件的一组一致性归纳,这样便找到了一组可替换的一致性归纳。在得到这个替代后,它便扩大函数的反馈信息,并从归纳集中选出最好的一个,同时删除被它归纳的规则。通过改变参数,可以改变最佳规则的选择标准及输出规则的数目。

INDUCE 能提供背景知识,而用户可以给归纳增加算术和逻辑规则。除了背景规则,INDUCE 还能增加新函数、等值谓词及极端谓词。

INDUCE 的语言不同于 KBGIS-II 的语言。因此要用翻译器将对象定义从一种语言转换成另一种语言。

A. KBGIS-INDUCE 这种翻译器将一组用 KBGIS-II 语言书写的规则转换成 INDUCE 的形式。除了句法的翻译,还要执行下列任务:

(a) 目前实现的 INDUCE 中不允许有析取,因此如果输入规则中有析取,就要把它分成两条规则,例如 $(X1 \vee X2) \wedge Y \Rightarrow [d=1]$ 变成: $X1 \wedge Y \Rightarrow [d=1]$; $X2 \wedge Y \Rightarrow [d=1]$ 。

在 KBGIS-II 中 GPROP 和 RPROP 没有析取,但 PPROP 可以有含两个层之析取的子句。因而翻译器为每个析取生成一条新规则,例如,当输入规则如下:

$[(TYPE\ O_1)\ LAKE]$ 。

$\wedge \{[(土地\ O_2)\ 21] \vee [(地质\ O_2)\ 22]\}; \wedge \{[(海拔\ O_2)\ (100\ 200)] \vee [(坡度\ O_2)\ (20\ 40)]\}$ 输出为:

$[TYPE\ (O_1) = LAKE][土地\ (O_2) = 21]\ [海拔\ (O_2) = 100..200] \Rightarrow [d=1]$

$[TYPE\ (O_1) = LAKE][地质\ (O_2) = 22]\ [海拔\ (O_2) = 100..200] \Rightarrow [d=1]$

$[TYPE\ (O_1) = LAKE][土地\ (O_2) = 21]\ [坡度\ (O_2) = 20..40] \Rightarrow [d=1]$

$[TYPE\ (O_1) = LAKE][地质\ (O_2) = 22]\ [坡度\ (O_2) = 20..40] \Rightarrow [d=1]$ 。

(b) INDUCE不能处理实型数,而且取值范围也不能太大(典型值 ≤ 100);因此 GPROP 和 PPROP 的值应适当地规范化。

B. INDUCE-KBGIS 这种翻译将一组用 INDUCE 的语言书写的规则转换成 KBGIS 的形式。目前实现的 KBGIS 语言不能与 INDUCE 的语言完全兼容;我们将忽略掉无法转换成 KBGIS 语言的输入子句。

如果系统学会了一个新对象的定义,就把它直接存贮到空间对象知识库中。否则,系统对新旧定义加以比较,如果新的更好,则修改空间对象知识库。在目前的实现中,如果是由于语言的不兼容性,有时系统可能无法处理 INDUCE 输出。在这类情况下,用户可以借助知识库编辑器,解释输出结果,并修改知识库。

12. KBGIS 的应用实例 已用由 USGS 准备的一个多层次数据集对该系统进行了测试。数据集中表示的区域采用 Black Hills 的正方形区域。将该区域表示成一组数据层,每个有 512×512 个 50m 的像素。数据层包括土地利用、地质、地形、流域、方位/坡度及运输。现在我们简要描述一下 KBGIS-II 的两个实际应用,第一个是一个查询的例子,第二个是一个归纳学习过程的例子。

(1) 第一种查询的例子 假设系统要解决的问题是在 512×512 的区域上找到最坏情况的潜在山崩位置。该空间对象用自然语言描述如下:

最坏情况的潜在山崩位置是这样—个区域:其方位是西方 ($270^\circ - 360^\circ$),有较缓的坡度 (小于 20°),而且其地质是 Spearfish 页岩, Sundance 页岩, Niobrara 石灰石,或者 Pierre 页岩。该区域必须位于距离第一级溪流源头 200 英尺以内,距离地形边线 800 到 2000 英尺以内,距离对应于 Opeche 页岩和 Minnekanka 石灰石的各地质地层之界面 200 英尺之内,或者距离一个陡峭地形断坡 (30° 或更大) 200 英尺以内。

用 SOL 表示,则该查询写成: {AND[(TYPE O—1) 断坡]}。

[(TYPE O—2) (地质界面—19—20)]; [(TYPE O—3) 边缘]; [(TYPE O—4) 溪流源头—1]; [(TYPE O—5) 期望—滑动]; [(距离 O—1 O—2) (0 200)]; [(距离 O—2 O—5) (0 200)]; [(距离 O—3 O—5) (0 1000)]; [(距离 O—4 O—5) (0 200)]。

假设系统拥有足够的辅助知识,那么只要具备系统中对象定义的入口及搜索命令(1),就足以让系统产生完全搜索过程,认识到这一点非常重要。例如,需在知识库中定义子对象 O—i, $i=1, 5$, 以及相关属性,过程和已知实例。再如,将期望—滑动定义为: (AND)。

[地质 O—1 (21 22 31 33)]; [坡度 O—1 (0—36)]; [方位 O—1 (270 360)]。其中地质、坡度和方位都是 PPROP。

为回答该查询,要访问五个数据层(或四叉树)。在遇到这种特殊空间对象的时候,将搜索过程定位在窗口中实例数最少的线性特征上。用限制的传播来构造窗口,在这些窗口中搜索连续的对象。用区域增长过程在限制最强的窗口中寻找空间对象期望—滑动。

(2) 归纳学习的例子 系统要解决的第二个问题涉及四种矿床的位置。KBGIS-II 仅知道四个 32×32 个像素区域的位置,不知道有关这些区域的其他信息。系统根据第二种查询依次检查每个区域。在特殊情况下,假设系统有一组 PPROP (地质、海拔、坡度和土地利用) 一组 GPROP (大小) 和一组 RPROP (距离、方向和限制),系统则以这些 PROP 的形式(例如用 SOL 形式)返回四个区域的特征。这些查询的输出随后又作为归纳学习系统的输入,再依次产生四种归纳描述,作为归纳学习系统的输出。

返回的归纳描述可以用 SOL 表示成: {AND[(限制 O—3 O—1) T]}。

[(方向 O—1 O—2) 4]; [(大小 O—1) (256 896)]; [(大小 O—2) (128 512)]; [(大小 O—3) (1024)]; <AND{OR[(地质 O—1) (34 37 38)]}>;<AND{OR[(地质 O—2) (14 16 17)]}>;<AND{OR[(土地 O—3) (42)]}>。

用自然语言的描述表明每个区域的一般特点是由三个原始子对象组成（其中一个子对象的三种类型之一是岩性学；还有一个子对象中土地利用为常绿森林）。另外，每个子对象都以某种大小的属性为特征，而不同的子对象之间存在方向和包容关系。

必须强调指出，若给定一组区域及刻划每个区域特征的一组PROP，系统便在自动方式下执行一个过程序列。另外，归纳得出的空间对象的SOL定义及其各种属性被自动加到系统的知识库中，因而可以在今后寻找该对象的查询中用到。

13 概要和结论 本文描述的 KBGIS-II 最近已在加利福尼亚大学圣巴巴拉分校VAX 11/750上VMS操作系统的支持下实现并运行，该系统用 Common Lisp, Pascal 和 C 编程。应强调一点，目前系统的属性是调查得到的，而且计划继续开发KBGIS-II。同时讨论用目前系统执行的搜索及准备为系统扩充的搜索。

(1) 需求与原则 系统目前能够处理大型、多层次、异构型的空间索引数据库。满足上述所有系统需求的软件设计必然使目前的硬件成为限制可以用交互速度处理的数据库大小的一个因素。如果把该系统移植到更合适的硬件（如LISP机）上，则可在很大程度上解决这一问题。系统能够回答可以用空间对象语言（SOL）表示的类型(1)和(2)的所有查询。虽然对该问题的研究还在不断深入，但从减低搜索空间对象的平均复杂性的角度来说，查询处理效率看起来还是相当高的。硬件的缺陷导致系统在查询复杂空间对象时无法真正达到交互。KBGIS-II 对应用领域和用户领域都具有灵活性。

可以如此评价为使系统满足五点要求而采用的四组原则：

(a) 系统的开发经受了：由于坚持有原则地使用软件工程技术而导致的失败，虽然也从数据库管理技术的系统应用中受益（在构造和存贮位置树数据库时，这里空间图象数据被分割成可寻找的区域，并根据需要将各区域分页）；算法和复杂性理论的应用（在构造空间搜索过程时）；使用AI技术（在建立知识库，设计空间搜索过程及应用学习过程时）；以及应用计算机图形技术（用于系统输出）。

(b) 将计算机视觉和图象处理技术结合起来，使系统能够处理GIS中不具有代表性的某种查询，同时使系统把图象和数字制图数据结合起来。

(c) 在查询一节中（§7.1）讨论的6条原则，与标准的基于光栅的彻底搜索过程相比，使系统响应查询的计算开销大为减少。

(d) 提供了各种编辑器，使系统易于改装成适用于不同空间领域和用户的新系统。

(2) 系统性能调查 对系统处理有关大型地质数据库的各种查询能力的调查目前正在进行。主要调查工作涉及到搜索过程的效率及改变各种参数对搜索效率之影响的经验分析。

(3) 系统扩充 目前正在实施修改KBGIS-II的计划，目标是提高系统目前处理能力的功效，并扩充这些能力。将进行的扩充包括：

(a) 增加类似于ARC/INFO等目前流行的系统中的计算机制图能力和多边形处理函数；

(b) 增加“模糊”空间对象的定义和“模糊”推理；

(c) 增加一个数据库和专门处理遥感数据的函数，及该数据库与KBGIS-II之间的一个界面，增加图象的地图引导解释过程，提供能连接数字化制图与图象数据库两种查询的数据

面向目标空间数据库——实例研究

P. 米尔恩等

1. 引言 地理信息系统 (GIS) 所关心的是可以想象得到的最广泛的信息。其研究对象是现实世界, 而不是一些商业上 (例如商业数据库) 的抽象模型 (虽然诸如此类的数据库可以成为GIS的组成要素)。GIS的应用范围非常广泛, 包括保护、开发和其它主要工业、国防、紧急服务、环境管理、设施管理、土地管理和规划。

涉及土地信息管理的组织机构与从硬拷贝而来的几何数据库紧密相关, 同时, 先进的数据库更新与转换技术正在使巨型地理数据库成为现实。日常生活中, 许多GIS的应用都涉及几个横跨不同组织机构的数据库。应用的广泛性和目标的复杂性超出了数据库技术的能力, 使得其必须进一步深化。因此, GIS正在推动数据库技术的发展。

为GIS用户服务的工业团体正在迅速地演化, 目前, 已经突破了原有的专门化的应用领域, 例如地理数据处理、环境中开发的数据的采集技术。由于涉及图形数据类型的操作需要, 这些工业团体已经抛弃了传统的数据库的发展方向。目前, 面向目标的数据库技术正在适应工程需要 (例如GIS)。可以预料, GIS将在适合其需要的一般目的数据库平台上得到发展。

本文讨论了基于具有商用可能的、面向目标的数据库管理系统的地理数据库 (OODBMS) 发展中的一些问题。澳大利亚空间信息中心 (CSIRO) 信息技术室开发了这种数据库。这种数据库要求能够支持地理目标和它们属性的交互输入、检索和显示。图 1 (略) 给出了一个例子。

用作设计的概念数据模型是以空间数据转换标准草案 (SDTS) 中的数据模型为基础的。§ 2 中作出了描述。§ 3 中叙述GIS设计。§ 4 中讨论涉及 OODBMS 的一些问题。§ 5 中阐述 OODBMS 和数据库设计。§ 6 中讨论一些中间运行结果。

2. 空间数据转换标准 空间数据转换标准的建立 (美国地质调查局1991) 使现今GIS协会感兴趣的数据概念模型达成了一致意见。用实体和类型的术语来描述这种模型, 它的反面是: 目标和类型, (参§ 4), 实体可以考虑为一个限定类型的空间现象。可是, 如图 2 中格网概念模型所描述的数据聚集体并不是对应于一种具体的真实世界中的实体。图 2 显示了这个模型的典型部分, 这个模型涉及到SDTS称之为的空间目标, 用位置、形状、大小和拓扑关系来描述。其它属性 (包括实体) 通过关系模型结合在一起。

空间实体的概念模型必须用图形形式表示成地理系统的内部模型 (例如数据模型)。如果面向目标的数据库被用于这个系统, 那么空间实体即用图形形式表示成复杂目标。例如。一个线目标的数据组成可以是它的起始结点、终止结点、确定此线的中间点的集合、以及一些可打印属性。

SDTS概念模型的非空间成分是一个关系模型, 这个模型专门负责建立空间目标和实体结构和过程。

- (d) 为系统提供答复有关时间变化探测一类查询的能力; 以及
- (e) 提供允许系统的归纳学习过程独立进行操作的过程和控制结构。

宛铭译自《IJGIS》1987, 第1卷, 第2期