

# 空间数据在嵌入式导航系统中的索引

沈永增 姚萌萌 周 巍 (浙江工业大学 信息工程学院 浙江 杭州 310014)

**摘要:** 研究了基于嵌入式 Linux 平台电子地图数据的空间索引,介绍了空间索引的基本方法。通过分析常见的空间索引算法如 B 树、R 树、四叉树等算法的特点,结合嵌入式电子地图数据特点,设计了基于 Hilbert 编码的格网空间数据索引机制,并改进了格网索引的数据结构,从而提高了索引效率;并使用嵌入式数据库 SQLite 管理空间数据,达到了预期效果。

**关键词:** 电子地图数据;空间索引;Hilbert;格网索引;SQLite

## Spatial Data Index in Embedded Navigation System

SHEN Yong-Zeng, YAO Meng-Meng, ZHOU Wei

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** This paper deals with electronic map data index based on Embedded Linux System, and introduces the basic ways about spatial index. By analyzing the common indexed algorithms, such as B Tree, B+Tree, R Tree, Quadtree and so on, it designs the spatial data indexed structure based on Hilbert grid and modifies the data structure. Index efficiency is improved because of the modified structure. Finally, it uses the SQLite database to manage the spatial data and achieves expected effect.

**Keywords:** electronic map data; spatial index; Hilbert; grid index; SQLite database

## 1 引言

嵌入式 GPS 导航系统需要大量的空间数据支持电子地图的绘制、路径规划、搜索数据等功能实现。嵌入式导航系统的硬件设备和 pc 机相比,在速度和效率上相差很大。所以,如何从大量空间数据中快速检索、提取所需的空间数据来满足嵌入式 GPS 导航系统中地图的搜索地物点、地图快速漫游以及多尺度显示、路径的规划等功能成了重要的问题。

空间索引是指依据空间对象的位置和形状或空间对象之间的某种空间关系按一定的顺序排列的一种数据结构,其中包含空间对象的概要信息,如对象的标识、外接矩形及指向空间对象实体的指针。作为一种辅助性的空间数据结构,空间索引介于空间操作算法和空间对象之间,它通过筛选作用,大量与特定空间操作无关的空间对象被排除,从而提高空间操作的速度和效率。空间索引的性能的优劣直接影响空间数据

库和地理信息系统的整体性能<sup>[1]</sup>。

## 2 空间索引算法研究

根据对象描述和索引结构的处理,针对空间数据本身的特点以及空间数据的点查询、范围查询、最近邻域查询的特点,国内外学者开发了很多空间索引技术。从空间目标映射方式可以把空间索引技术分为二类<sup>[2]</sup>:

### 2.1 基于空间目标排序的索引方法

这种方法的思想是:把空间数据多维的目标采用某种策略映射成一维的目标,这种映射要较好的保持多维空间目标的邻近关系。用一维的值给多维的空间目标排序。这种排序技术很多,比如:peano 曲线、位置键、Z-排序等。多维的空间目标转换为简单的数字,用常规的数据库系统可以管理空间目标的属性与几何信息。

基金项目:科技厅面上项目(2007C30008)

收稿时间:2009-08-07;收到修改稿时间:2009-09-12

## 2.2 专门的索引方法

这种方法是在系统中加入专门的外部空间数据结构，来对空间数据索引。常用的方法有：

### (1) 不允许空间重叠的索引法

这种方法将空间数据按某种方法比如二叉树、四叉树、格网划分成彼此不相交的子空间，然后对属于这些子空间的目标分别存储在对应的磁盘页或数据通中。而这种方法必须采用 2 种技术：目标复制、目标裁剪。

### (2) 允许空间重叠的索引法

这种方法的基本思想是：按某种策略将索引空间层次划分成多级的子空间，这些子空间允许重叠从而使点与非点状物完全包含在某子空间之内。

采用专门的索引方法一般是自顶向下、逐级划分空间的各种数据结构空间索引，比较有代表性的包括 B 树、B+树、R 树系列、四叉树等，结构较为简单的格网空间索引有着广泛的应用。

B 树索引结构式关系数据库中用的最广泛的索引。嵌入式数据库 Sqlite、ysql 都采用 B 树和 B+树索引机制。B 树和 B+树在一维的数据搜索中是一种很高效的搜索机制，并且读磁盘的次数很少。但 B 树和 B+树不适用复杂的多维的空间数据索引和管理。比如，搜索面状物和线状物实体时，B 树和 B+树很难高效的索引和管理。把多维的空间信息映射到一维的索引域是选择关系数据库管理空间数据常用的方法。

R 树是一种动态平衡树，基于对象划分的数据结构。它允许划分的子空间重叠，而且查询效率高，树的每个结点是一个区域，叶结点是实体最小外接矩形的集合。R 树索引机制不划分图像，只利用插入和删除操作分组空间对象，建立树数据结构。R 树索引机制主要问题是结点分裂操作比较复杂，这会降低索引效率。在空间数据目标数目很多的时候，一些空间实体的外接矩形会出现大量的重叠，使内存中出现大量重复存储，同时树的深度会增加，降低索引效率。

四叉树索引是基于空间划分组织索引结构的索引机制，是一种面向主存空间的索引技术。四叉树索引是一种重要的索引。层次四叉树索引中决定一个空间对象的层次依赖于该对象的最小外接矩形(MBR)的大小，而不是它的位置。传统的四叉树索引中，非叶结点不能存储实体对象信息，这样在叶结点中，会有很多实体对象的重复存储。后来，规定非叶结点亦可存

储空间对象信息，前提是该结点完全包括对象的最小外接矩形，而其子结点不包含该对象的最小外接矩形(相离或相交)，这对于减少重复对象的存储有很大帮助。层次四叉树层次间的指针操作比较复杂，随着四叉树层次的深入，影响索引效率。线性四叉树是采用一定的编码方式对空间对象覆盖的范围编码，进行四叉树分割，线性四叉树的编码方式很多，而且编码方式会影响索引效率。线性四叉树索引仍然很复杂，不容易建立和维护。

人们一直没有停止对空间索引算法的研究，不断出现改进的索引算法。各种算法都有优点，但在一定条件下也可能出现瓶颈问题。在实际应用中，要选择合适的空间索引机制。本文结合嵌入式 linux 的电子地图数据本身的特点，使用格网索引，这种索引方法结构比较简单，查询效率比较高，查询精确。格网索引使用也比较广泛。

## 3 空间索引算法研究

### 3.1 格网划分与编码

格网索引的基本思想是将图幅纵横分成若干相等或不相等的格网，将落于该格网内的空间对象放入该格网对应的存储桶中。当用户进行空间查询时，首先计算出用户查询对象所在格网，然后再在该格网中快速查询所选空间实体，这样一来就大大地加速了空间索引的查询速度。格网索引思路比较简单明了，容易理解和实现<sup>[3]</sup>。格网划分是将一幅图分成  $M \times N$  个网格如图 1 所示：

00	01	02	03
10	11	12	13
20	21	22	23
30	31	32	33
40	41	42	43

图 1 4 × 5 格网

每个网格可以用行列号组织，并按行列号排序。如 Grid[0,1]表示第 0 行第一列的格网、Grid[2,3]表示第 2 行第 3 列的格网。不过为了便于建立空间索引的线性表，将每个网格按曲线或编码进行二维到一维

的映射编码，建立编码与空间实体的关系，做为格网索引文件。

本文用 Hilbert 空间排列码进行映射，这种映射需要将地图划分成  $N \times N$  个网格。Hilbert 空间排列码最早由 Faloutsos 和 Rosman 提出，这种算法基于空间目标格网行列数的二进制操作<sup>[4]</sup>。

阶数为  $n$  的 Hilbert 空间排列码生成算法描述如下：

- (1) 读入网格的行列号  $r, l$ ，转换长度为  $n$  的二进制表达形式；
- (2) 将二进制的  $r, l$  按位互相交叉，构成一个长度为  $2n$  的二进制串  $S$ ；
- (3) 将串  $S$  从左到右按每两位分开，形成二进制串  $S_i (i=1, \dots, n)$ ；
- (4) 按照一定形式给出  $S_i$  每两位二进制所对应的十进制数值，并按照顺序加入到数组  $D[]$  中，例如：“00”等于“0”，“01”等于“1”，“10”等于“2”，“11”等于“3”；
- (5) 对于  $D$  中的每个值  $D[i]$ ：如果存在  $i$ ，使  $D[i]=0$ ，如  $D[j]=1$ ，则  $D[j]=3$ ；如  $D[j]=3$ ，则  $D[j]=1 (j>i)$ 。如果存在  $i$ ，使  $D[i]=3$ ，如果  $D[j]=0$ ，则  $D[j]=2$ ；如  $D[j]=2$ ，则  $D[j]=0 (j>i)$ ；
- (6) 将数组中每个值按(5)转换成二进制表示，自左向右连接所有的串，并计算其十进制。

假设一个  $N \times N$  格网 第  $i$  个网格表示为  $Buck[i]$ ， $Buck$  与  $Grid$  的关系为：

$Buck[Hilbert(N,r,l)] = Grid[i,j]$ ，其中  $r$  表示网格的行号， $l$  表示网格的列号， $Hilbert(N,r,l)$  根据划分网格的阶数和  $r, l$  求得编码，“ ”表示一一对应关系。这样建立一一对应关系，可以用唯一标识 ID 码表示网格，方便存储，节省嵌入式设备中有限的存储资源。对空间对象检索时，只对含有空间对象的网格进行检索。而且许多实验证明，用 Hilbert 空间排列码对于空间数据来说，有很好的聚类性。

对于任一点状物  $Point\_obj$ ，设其对应网格为  $Buck[m]$ ，则有  $Point\_obj \in Buck[m]$ 。对于任一线状物  $Line\_obj$ ，设其对应网格的编码集合为  $\{k_1, k_2, k_3, \dots, k_n\}$ ，则对任意  $m \in \{k_1, k_2, k_3, \dots, k_n\}$ ，有  $Line\_obj \in Buck[m]$ 。对于任一面积状体  $Area\_obj$ ，设其所覆盖的网格号集合为  $\{k_1, k_2, k_3, \dots, k_n\}$ ，则对任意  $m \in \{k_1, k_2, k_3, \dots, k_n\}$ ，有  $Area\_obj \in Buck[m]$ 。

### 3.2 格网索引结构的改进

格网索引机制的数据结构有网格列表以及各网格对应的对象单链表组成，如图 2 所示。在一般的格网索引机制中，网格的结点包含其编码以及指向第一个实体的指针。这样存在网格内的实体有点状物、线状物、面状物，而且这些实体的 ID 号全部在一个单链表中。如果查询一个面状物，不管链表中的结点是不是面状物都要查询，这样必然会影响查询的时间。如果在网格的结构体上，原来的指针去掉，加入指向第一个线状物的指针、指向第一个面状物的指针和指向第一个点状物的指针，这样在查询时，没必要查询不同类型的实体对象，从而提高了查询效率。地图数据中的面状物、线状物不像点状物只有一个点组成，线、面实体由连续的节点组成。线、面实体结构体中，加入该实体中属于该网格的节点集合在磁盘上数据库中的地址信息：起始记录和偏移记录，这样会加快访问外存的速度，加快显示、更新电子地图的效率。在绘制地图时，显示不同尺度的地图需要显示不同的对象，所以对象结构体还要加上级别标识。改进的网格结构体结构如图 3 所示。各网格指向某类型的第一个对象结点的指针若不为空，则表示该网格内存在该类型的实体对象；否则，网格内不存在该类型的实体对象<sup>[5]</sup>。

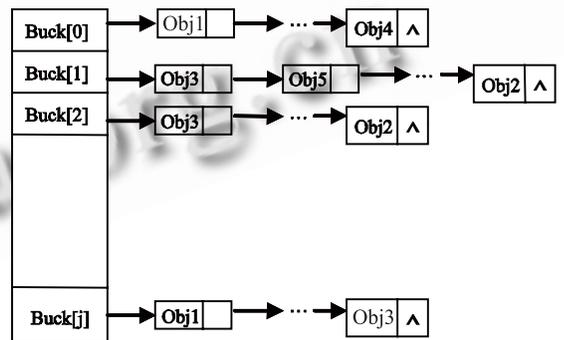


图 2 格网索引基本结构

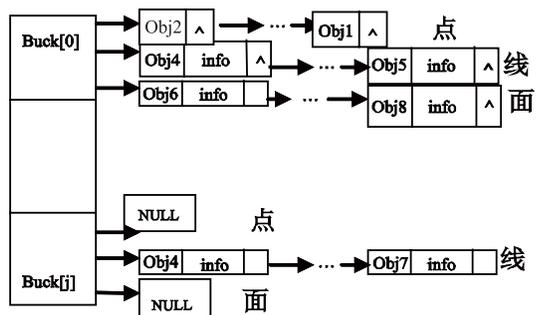


图 3 改进的格网索引基本结构

### 3.3 格网索引的可行性分析

从图 2、图 3 中可以看出，如果一个实体跨越多个格网，则实体编码会同时出现在这些与被跨越的格网一一对应的桶内，造成在内存中数据存储的冗余。但是这种冗余在一定程度上是有价值的。对于线状物来说，跨越的格网，说明了该线状物表示的路段的轨迹，而对于面状物来说，跨越的格网，说明了该面状物覆盖的区域；其次，格网对线对象而言，虽有内存存储的冗余，有些学者通过分析认为，基于格网的索引结构整体应比基于 MBR 的索引结构效率高。

格网索引机制的关键在于对一幅图的格网划分。划分机制直接影响到算法的时间效率和空间效率，需要根据软硬件情况来合理划分每个网格的大小。如果网格的数量过多，精度可以保证但是时空效率相对来说比较低；如果网格的数量较少，一个网格里面的实体对象也很多，同样影响时空效率。本文通过建立基于行政单位的三级索引，解决格网数量过多的问题，即网格属于某个地区的编码。面状物的格网划分，是根据面的最小外接矩形和格网的边界相交情况划分。嵌入式电子地图数据对于面的处理是比较少的，可以把面抽象成一个点在格网中索引<sup>[6]</sup>。

### 4 SQLite在索引机制中的应用

SQLite 是开源的完全免费的嵌入式数据库，运行中占有内存很小(少于 250KB)，支持 2T 的数据量，支持大部分 SQL92。SQLite 是一款轻巧的数据库系统。SQLite 用的是 B+ 树索引机制，查询速度较快，在 PC 机上查询效率上甚至超过 MySQL。

本文建立的格网索引中，空间数据可以使用 SQLite 管理。如表 1：

表 1 格网 ID 索引表结构

字段	类型	字段说明
Grid	int	格网 ID
Area_num	int	所属地区编码
Geoid	int	对象 ID

在上面的表中，对表中字段 Grid 建立索引，以便加快查找速度。测试证明，在 SQLite 中，添加索引要比不添加索引查询快很多。在搜索某一点信息时，可以根据该点坐标获取其所在的网格 ID，然后从磁盘读入该网格的实体对象的数据显示到 LCD 上。假设地图

被分为  $N \times N$  个格网，则格网的行为  $i$ ，列为  $j$ ，则  $ID = Hilbert(N, i, j)$ 。

### 5 结论

本文在通过分析空间索引算法的基础上，根据嵌入式 Linux 平台的 GPS 导航电子地图的特点，设计了一个基于 Hilbert 码的格网索引机制，并改进了格网索引数据结构，提高了索引效率。在实际运用中，使用了嵌入式数据库 SQLite 管理空间数据信息，并用 QT/Embedded 实现了电子地图。本文开发板配置：s3c2440、64M ram、64M nandflash、480 × 272 tft Lcd。下面是杭州地图数据测试的结果。



图 4 开发板上显示界面

表 2 从 sqlite 中读取数据所需时间

设备	网格数(记录总数)	时间(ms)
Pc	9个网格(23000)	20 - 60
开发板	9个网格(23000)	100 - 300

### 参考文献

- 郭薇,郭菁,胡志勇.空间数据库索引技术.上海:上海交通大学出版社, 2006.
- 张宏.地理信息系统算法基础.北京:科学出版社, 2006.
- 张为舟.嵌入式电子地图开发方法的研究[硕士学位论文].武汉:华中师范大学, 2006.
- 王永杰,孟令奎,赵春宇.基于 Hilbert 空间排列码的海量空间数据划分算法研究.武汉大学学报, 2007(7): 650 - 652.
- 方钰,何启海.面向移动导航设备的数字地图分级组织与预取策略.小型微型计算机系统, 2006,(7):1339 - 1343.
- 成洁,吕遵明,敖雪.基于嵌入式 Linux 的嵌入式 GIS 的设计与实现.电子工程师, 2006,32(11):69 - 71 .