

面向嵌入式 GIS 的数据组织模型与存取机制

陈玉进¹, 李 泉¹, 柳 盛^{1,2}, 王春红¹

(1 南京睦步科技有限公司, 江苏 南京 210008)

2 南京师范大学 虚拟地理环境教育部重点实验室, 江苏 南京 210046)

[摘要] 地图显示效率一直是嵌入式 GIS 系统的核心问题。一方面, 嵌入式系统处理器性能低、内存容量小; 另一方面, GIS 数据量大、计算复杂。针对这一对突出的矛盾, 本文提出了一种新的 GIS 数据组织模型与存取机制, 在 I/O 阶段, 从逻辑和物理 2 个层面最大限度地减少对冗余数据的读取和处理, 保障地图的快速显示。

[关键词] 嵌入式 GIS, 纵向分级, 网格索引, Hilbert 排序, 缓存

[中图分类号] TP 311 [文献标识码] A [文章编号] 1672-1292(2008)04-0159-04

Data Organization Model and Access Mechanism Facing Embedded GIS

Chen Yujin¹, Li Quan¹, Liu Sheng^{1,2}, Wang Chunhong¹

(1 Nanjing Creable Technology Co., Ltd Nanjing 210008, China)

2 Key Laboratory of Virtual Geographic Environment of Ministry of Education, Nanjing Normal University, Nanjing 210046, China)

Abstract That the map shows efficiency has always been the core issue of embedded GIS system. On the one hand embedded systems processors have low performance and low memory capacity. On the other hand GIS has large data volume and complexity in calculation. Aiming at this pair of outstanding contradictions, the paper proposed a new organizational model and GIS data access mechanism, in the I/O phase, from the logical and physical levels, to minimize redundant data on the reading and processing, and secure the rapid display of the map.

Key words embedded GIS, vertical grade, grid indexing, Hilbert sort, cache

随着嵌入式软硬件技术的不断发展, GIS 运行平台也从 PC 扩展到了嵌入式设备, 尤其是智能手机的普及, 为嵌入式 GIS 在各行业的应用提供了条件。但是, GIS 数据量大、计算复杂, 给地图快速显示带来很大的压力, 因此, 在满足需求的情况下, 考虑用软件方法提高数据检索的效率、减少冗余数据的读取和处理, 提高必要数据的处理能力。本文主要从 GIS 数据的组织模型、存储和读取层面来阐述。

1 嵌入式 GIS 数据组织模型

1.1 纵向分级显示

GIS 数据按类别或等级不同分成不同的图层, 这些图层叠加在一起形成一幅层次和内容丰富的地图。而嵌入式设备屏幕通常都比较小, 能显示的地图内容和范围有限, 在同一屏幕上, 显示内容过多会造成图元拥挤不堪, 不但影响用户读图的效果, 而且用户等待时间过长, 给系统造成很大的负担, 因为, 显示内容过多, I/O 读取次数增加, 内存占用多, 地图坐标到屏幕坐标转换的计算量增大, 屏幕渲染的压力也很大。鉴于此, 地图需要按可视范围大小分级显示, 即窗口范围大, 只显示概要图层, 随着地图放大, 窗口范围缩小, 一些细节层次的图层逐步被显示出来。这样, 需要对地图图层按显示级别进行分组, 属于同一缩放级别的归为一组, 不同缩放级别的归为另外一组。具体实现就是, 给每个显示级别设置最大最小显示比例尺范围 max_zoom 和 min_zoom , 当当前显示比例尺符合条件 $max_zoom > win_zoom > min_zoom$ 时, 该级别对应的图层组被显示, 显示比例尺 = 当前可视范围实际宽度 / 窗口宽度, 当用户不断放大窗口显示地图时, 显示比例尺不断缩小, 以至于能够显示比较详细的显示级别图层组。

收稿日期: 2008-06-18

通讯联系人: 陈玉进, 硕士研究生, 研究方向: 嵌入式 GIS 和数字图像处理。E-mail: geochen@126.com

1.2 横向分块索引

嵌入式设备屏幕小,通常只显示地图上的 1 小块区域,屏幕之外的地图数据不参与显示,也就没有必要读入内存参与计算.这就涉及到从地图数据集合中,快速地选择出当前屏幕参与显示的数据子集.为此,需要对地图数据建立空间索引,空间索引数据常驻内存,通过空间索引获得地图数据的文件地址,再实际 I/O 读取所需要的地图数据.目前空间索引种类比较多,各有优缺点,实验结果表明,面向查询,网格索引效率最高,将整个图幅范围按网格进行划分,对跨网格的线、面事先依网格裁减,裁减后生成的多个子图元,分别存储到不同的网格索引单元里;由于线、面图形的不规则,其它索引都在不同程度上存在冗余数据的读取和处理,降低了整个地图显示的效率.调度的时候,物理上,非并发的 1 次 I/O 操作,只涉及到 1 个物理存储块;逻辑上,以 1 个网格索引里的数据为单位来调度.所以,理论上,逻辑层调度的单位数据量等于 1 个物理存储块的大小,效率达到最高,但通常无法做到,控制好逻辑层 I/O 数据读取的粒度,能保证较好的效率,这个参数需要针对不同数据进行测试.

1.3 数据组织模型

基于纵向上分级、横向上分块的策略,从逻辑层面上减少冗余数据的读取,为此,本文提出了一种新的数据组织模型——多图层共存同一逻辑文件、同一显示级别的多图层共存同一网格、1 个显示级别对应 1 个网格索引.

多图层共存同一逻辑文件 原本 1 个图层 1 个文件,现在若干个图层合并到一起形成 1 个文件,这样避免了显示图层过多带来的多次访问不同文件、造成 I/O 次数增多的弊端,使得显示效率与图层数量无关.

同一显示级别的多图层共存同一网格 纵向上,同一显示级别的多图层数据要一起被显示,横向上,同一网格里的数据也是一起被显示,因此,把同一显示级别的多图层数据组织到一个网格下,且逻辑上进行分图层组织.

1 个显示级别对应 1 个网格索引 以往 1 个图层 1 个索引,现在 1 个显示级别 1 个索引,首先判断某个级别是否需要显示,如果需要,则检索此级别对应的网格索引,把处于当前窗口内的这一显示级别的数据检索出来,予以显示.

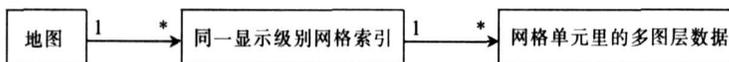


图 1 地图数据的组织逻辑模型

Fig.1 Logical model of map data organization

表 1 网格单元里多图层数据的字段格式

Table 1 Field type of multi-layer data in grid unit

字段名称	值类型	说明
所属图层编号	字节	图元所属图层编号
图元编号	整型	每个图元的编号
其它属性偏移量	整型	地图要素的其它属性在属性文件中的位置
图元名称	字符型	用于自动标注的图元名称
裁切后顺序号	整型	折线被网格裁切之后的顺序号
图元几何信息	自定义	图元的几何信息

2 嵌入式 GIS 数据文件存储

嵌入式 GIS 数据组织模型,是从逻辑层面阐述了减少数据读取和处理的规模、提高检索有效数据的效率,但不能最终保证数据读取性能,因为这个与数据文件存储方式有关. I/O 操作的单元是 1 个物理存储块,较大概率一起被读取的数据应尽量聚簇存储,以提高一次 I/O 操作获取有效数据的能力,如果数据文件记录存储无规律,或违背聚簇存储的原则,势必会增加 I/O 操作的次数,读取过多的冗余数据,降低整个系统的性能.基于此,需要分析地图数据聚簇的规律,并能实现数据聚簇存储.

同一显示级别的地图数据,是按网格索引组织的,网格里存储了属于同一显示级别、位于同一空间网格的多图层数据,网格之间按什么顺序来存储?空间数据一起被显示读取的概率是数据之间空间距离的

反函数, 也就是说, 距离近的地图数据之间被一起显示读取的概率大于之间距离远的, 所以, 这就要求网格索引中网格单元之间的存储顺序要遵循邻近原则: 靠得近的网格要临近存储. 但是, 网格索引是二维的, 存储器是一维的, 需要把二维的网格按照邻近原则映射到一维, 且保持邻近关系.

二维到一维的映射, 不能完全保持空间邻近关系. Hilbert 填充曲线, 是一种比较好的二维到一维的映射排序方法, 实现了网格数组坐标到一维排序序号的映射 $n = f(x, y)$, 通过这个映射函数, 就可以确定网格单元在一维排序中的位置, 且尽量照顾到了邻近原则.

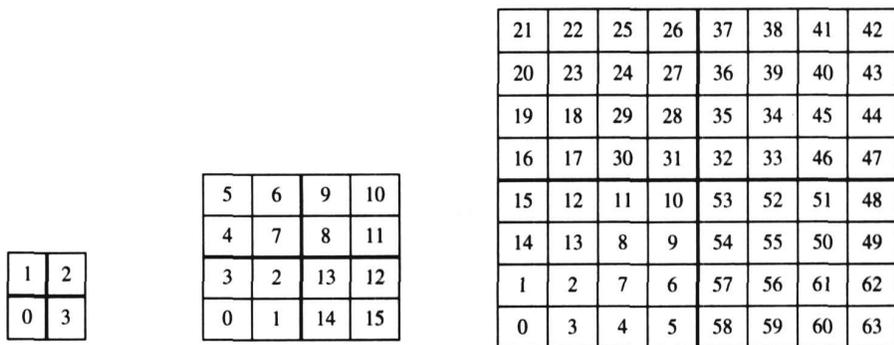


图 2 1、2、3 阶 Hilbert 填充曲线排序编码

Fig.2 Sorted encoding of 1-order, 2-order and 3-order Hilbert filling curve

Hilbert 填充曲线映射算法, 采用陈宁涛等所提出的迭代算法, 把 形的问题转化为具有 数特征的矩阵问题, 因而可以转化为矩阵运算, 通过复制旋转快速生成网格的编码. 按 Hilbert 映射排序, 实现了空间数据的聚簇存储, 从文件物理存储层面提高了 I/O 性能.

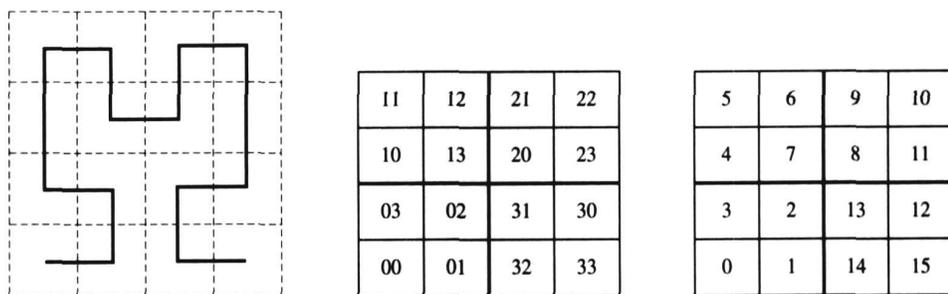


图 3 2 阶的 Hilbert 排列顺序

Fig.3 Sequential sorting of 2-order Hilbert filling curve

3 嵌入式 GIS 缓存机制

GIS 数据在嵌入式设备中以网格单位来管理, 加载 1 个网格单元, 淘汰 1 个网格单元. 设置缓冲网格单元数量上限值和下限值. 一旦缓冲网格单元的数量超过上限值溢出时, 立即启动淘汰算法, 使得缓冲区达到设定的下限值.

数据加载是被动的, 即不主动加载数据, 而是等到用户发出刷新地图命令时, 根据当前窗口可视范围来计算需要加载哪些数据, 然后在缓冲区中查找是否已经加载过, 没有加载则从文件中读取数据到缓冲区中. 图 4 所示, 为 1 次平移窗口底层所作的缓存操作. 左边的图是地图刚刚装载时的缓存状态, 虚线框表示当前地图的可视范围, a 色方框表示被加载过了网格数据, 右图显示用户向右平移了视口, 导致系统被动加载了 b 色部分的网格数据, 而之前已经加载的部分 a 色网格数据仍然在缓存中.

淘汰算法是依据数据空间位置进行的, 即先淘汰出离当前窗口显示数据最远的网格单元数据, 直到缓冲区大小达到下限或者只剩下当前需要使用的网格单元数量. 图 5 所示, 先淘汰 a 色部分网格数据, 再淘汰 b 色部分网格数据. 横向上如此淘汰数据, 纵向上依然如此.

4 结语

以上思路已在我们自主研发的嵌入式 GIS 平台 GridGIS Mobile SDK 得到应用 (如图 6), 实验表明, 此

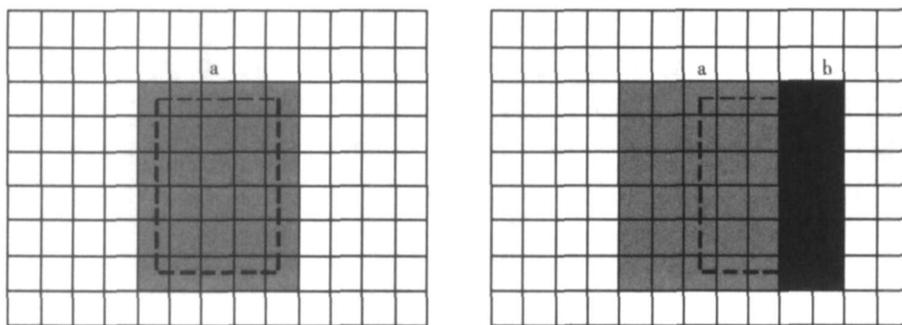


图 4 一次平移窗口操作的缓存机制

Fig.4 Cache mechanism of once translating window operation

方法极大地提升了地图显示的速度, 支持海量矢量地图数据的浏览, 解决了嵌入式 GIS 在各个行业应用中的效率瓶颈.

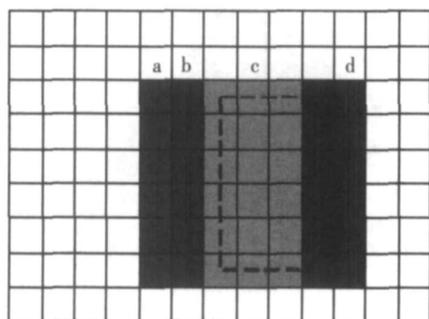


图 5 淘汰机制

Fig.5 Elimination mechanism

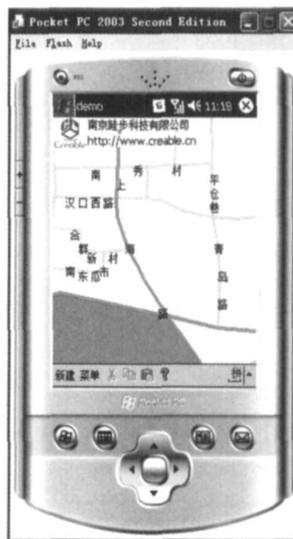


图 6 嵌段式 GIS 平台

Fig.6 Platform of embedded GIS

[参考文献] (References)

- [1] 胡泽明, 岳春生, 王志刚. 嵌入式 GIS 系统实时响应的软件方法实现 [J]. 测绘科学, 2007, 32(1): 98-99.
Hu Zeming Yue Chunsheng Wang Zhigang The software methods realization to improving real-time response in embedded GIS [J]. Science of Surveying and Mapping 2007, 32(1): 98-99 (in Chinese)
- [2] 夏颖, 张曙光, 张航. 空间数据检索在嵌入式 GIS 中的应用 [J]. 计算机应用, 2002, 22(12): 119-123.
Xia Yin Zhang Shuguang Zhang Hang Application of spatial data index in embedded GIS [J]. Computer Applications 2002, 22(12): 119-123 (in Chinese)
- [3] Chen Ningtao, Wang Nengchao, Shi Baochang. A new algorithm for encoding and decoding the Hilbert order [J]. Software-Practice and Experience, 2007, 37(8): 897-908

[责任编辑: 顾晓天]