

移动 GIS 切片地图展现技术^①

曹海涛¹, 贾博², 张波²

¹(国网电力科学研究院, 南京 210003)

²(宁夏电力公司信息通信分公司, 银川 750001)

摘要: 地图是 GIS(地理信息系统)操作的入口, 如何快速地显示地图是 GIS 首要考虑的问题. 快速展现地图主要涉及到地图数据的组织和地图展现技术两方面的问题. 目前, 以谷歌地图为代表的商业地图软件大都采用地图切片的方式组织地图数据, 这种方式被认为是快速且有效的. 本文采用上述方式组织地图数据, 使用切片并发控制技术和切片双缓存技术主要对移动 GIS 切片地图展现的相关技术进行研究, 旨在提供一种快速展现切片地图的方法. 基于本文的研究成果, 作者分别在 Windows Mobile 和 Android 平台上实现了移动 GIS 系统, 并成功应用于实际项目中.

关键词: 移动 GIS; 切片地图; 切片加载技术; 切片并发控制技术; 切片双缓存技术

Tile Map Display Technology of Mobile GIS

CAO Hai-Tao¹, JIA Bo², ZHANG Bo²

¹(State Grid Electric Power Research Institute, Nanjing 210003, China)

²(Information and Communications Branch of Ningxia Electric Power Company, Yinchuan 750001, China)

Abstract: Map is the entry of GIS, it is the prime consideration that how to show map quickly. Rapid map display mainly relates map data organization and map display technology. Presently, most of business map software represented by google map use map tile to organize map data, it is thought to be fast and useful. Using foregoing methods to organize map data and using tile concurrency control and tile double buffer technology, this paper is mainly about the tile map display technologies of mobile GIS and the purpose is to provide a method for showing tile map quickly. Based on the research findings, the authors have developed mobile GIS systems in windows mobile and android platform respectively, and the systems have been used in actual projects successfully.

Key words: mobile GIS; tile map; tile loading technology; tile concurrency control technology; tile double buffer technology

随着移动定位技术、无线通讯网络、地理信息系统、互联网技术的飞速发展, 基于位置服务(Location-Based Services, LBS)的应用也得到较快发展^[1]. 作为对 LBS 应用的支撑, 移动 GIS 的出现正是为了解决人们在移动设备上使用位置信息的需求. 它是继桌面 GIS 和 WebGIS 之后又一新的技术, 地图是移动 GIS 的核心部分, 而地图是对空间数据的展现. 传统的 GIS 系统通常采用矢量数据(如: shapefile)作为地图数据源, 随着地图切片技术的不断发展, 当前大多数

GIS 系统都采用切片数据作为数据源. 切片数据是按照一定的数据组织方式预先生成的, 与矢量数据相比它不需要在地图加载时实时渲染, 因此效率较高. 一些主流的商业地图应用, 如: 谷歌地图、百度地图, 都采用这种格式的地图数据.

切片数据是预先生成的, 通常这些切片数据是存放在地图服务器上的, 用户浏览地图时移动 GIS 系统自动将这些数据下载到移动终端上, 并实时显示在屏幕上. 如何将这些切片数据快速下载到移动终端上而

① 收稿时间:2013-06-18;收到修改稿时间:2013-07-22

不给服务端造成太大的压力,并且在用户不断操作地图时又能保证界面的响应速度,这是移动 GIS 地图展现系统必须要解决的问题;同时,移动设备使用的无线网络通常是不稳定的,如何在网络较差甚至没有网络的环境中浏览地图也是必须要解决的问题.针对上述问题,本文以当前比较流行的切片数据为数据源,对移动 GIS 切片加载技术和切片缓存技术进行研究,并将研究成果应用于实际项目中,旨在为用户提供流畅的移动 GIS 地图展现系统.

1 地图切片技术原理

地图切片技术(也称作地图瓦片技术)是一种地图预缓存技术.地图瓦片技术将配置好的一定坐标范围的地图,按照固定的若干个比例尺(瓦片级别)和指定图片尺寸,切成若干行及列的正方形图片,以指定的格式保存成图像文件,按一定的命名规则和组织方式存储到目录系统中或是数据库系统里,形成金字塔模型的静态地图缓存,地图切图所获得的地图切片也叫瓦片(Tile)^[2].

切片数据是按照影像金字塔的方式组织生成的.影像金字塔是一种栅格数据的组织方式,它是在统一的空间参考下,按照不同的分辨率将栅格数据分层表示,每一层具有相同的分辨率,不同层的分辨率从上到下按几何关系缩小,通常是按 2 倍的关系缩小,假设第一层的分辨率为 R ,则第 N 层的分辨率为 $R/2^{N-1}$,其中 $N \geq 1$;不同层的切片个数从上到下按几何关系增加,如果切片的分辨率是按 2 倍的关系缩小,假设第一层切片的个数为 1,则第 N 层切片的个数为 $2^{2*(N-1)}$.每一层的单个栅格图片称为切片,切片是正方形的栅格图片,大小通常为 256 像素×256 像素、512 像素×512 像素等,每个切片代表一定的空间范围.通常我们用商业地图软件(如谷歌地图、百度地图)看到的地图就是由切片根据空间关系拼接出来的.

按照上述规则,使用专业的地图切片软件(如: Arcgis 软件)生成切片地图.所有的地图切片存放在服务器上,对图片进行四叉树编码,客户端在进行地图浏览、缩放时,只需要动态计算所要加载的地图切片的 URL,然后访问那些地图切片,并下载到客户端的显示界面中^[3].关于切片的生成和切片服务的提供是服务端的内容,不是本文关注的重点.本文关注的是如何组织这些切片数据并在移动设备上展示,下

文将从切片加载技术和切片缓存技术两个方面探讨移动 GIS 的切片地图展现技术.

2 切片加载技术研究

通常我们用智能手机(如: Android 系统, IOS 系统的手机)浏览地图时会发现地图是一块块逐渐加载到手机界面上的(在网络较差的区域更加明显),这正是移动 GIS 切片地图加载的基本原理,即:切片是分步加载的.那么如何计算当前屏幕范围内要加载哪些切片,当用户移动、缩放地图时又要加载哪些切片呢?本文使用切片计算算法来解决这一问题,切片计算算法能够计算出用户移动或者缩放地图后,当前屏幕范围内应该显示的切片索引等信息,切片下载算法根据获取的切片信息从服务端下载切片地图,切片显示算法提供将获取的切片地图展示在系统界面上的策略.

2.1 切片计算算法

切片计算算法用于计算当前屏幕范围内所包含的切片信息.算法思想如下:

- ① 根据地图比例尺计算当前屏幕对应的地理范围.
- ② 通过比例关系计算出每个切片横向所占用的地理范围 X 、纵向所占用的地理范围 Y ,获取切片的宽度 W (像素)、高度 H (像素).
- ③ 根据切片的起始点、切片的尺寸以及当前屏幕的地理范围,循环计算出每个切片的横向编号 N_x 、纵向编号 N_y 以及切片的左上角点在屏幕上的坐标 (x, y) .

④ 根据切片在地图上的显示位置按照从左到右、从上到下的顺序将通过上述计算获取的切片信息保存在一个动态二维数组中,数组中的每个元素对应一个切片信息对象.在这里要特别注意的是这里用于保存切片的二维数组和数组中的切片信息对象都是可以复用的,只需要在第一次使用的时候申请内存空间即可,下次使用的时候只需要替换切片信息对象的具体内容即可;切片计算算法是需要被反复调用的(如:用户不断地拖动、缩放地图,每拖动一次就会执行一次切片计算算法),这样处理可以大大节约内存空间的占用,并节约频繁操作内存带来的系统开销.

2.2 切片下载算法

由于当前界面上显示的切片个数不至一个(其实界面上显示的切片个数为偶数,具体个数与切片的大小和屏幕分辨率相关),如果用单线程去实现切片下载势必会降低效率、造成用户等待时间过长的问

多线程下载则必须控制每次同时下载的切片个数, 否则并发量过大会给服务器带来较大的压力; 本文采用切片并发控制技术处理切片下载, 对切片的并发下载量进行控制, 以降低服务器压力; 同时过滤无效的切片请求, 以保证界面的快速响应. 假设每次地图界面更新时要向服务端请求 10 个切片, 在网络环境较差的区域, 用户连续 5 次拖动地图, 如果不做任何处理则必须从服务端下载 50 个切片, 而事实上最终展现在地图界面上的是第 5 次拖动地图后要下载的 10 个切片, 可以把之前 4 次拖动地图后要下载的 40 个切片看作是无效的切片请求. 如果不过滤掉无效切片请求, 则地图界面必须要等待这 40 个切片下载完成后才能更新, 在网络环境较差的情况下, 会严重影响用户体验.

切片下载算法的基本思想是: 使用切片并发控制技术, 控制每次同时下载的切片个数, 同时处理无效请求. 算法原理如下:

① 设计一个切片下载管理器 TileDownload Manager, 设置并发下载的最大切片个数 $\max\text{ ConcurrentTileCount}$, 获取界面上包含的切片个数 tileCount (给定设备屏幕分辨率和切片尺寸可以计算出切片个数).

② 通过 TileDownloadManager 的 download 接口添加切片下载请求 TileRequest, 用 Hash 表 TileRequest Container 保存切片请求, 判断当前添加的请求个数 $\text{currentRequestCount}$ 是否等于 $\text{tileCount}-1$, 如果是则说明界面上的切片都已经添加到下载队列这时转向步骤 3; 如果 $\text{currentRequestCount}$ 大于等于 tileCount 说明这是用户的下一次地图操作, 则清空 TileRequest Container(这时即使上次的请求尚未处理完, 也丢弃, 用这种方式来处理无效请求); 最后, 将切片请求添加到 TileRequestContainer. 具体算法流程如图 1 所示.

③ 判断是否是第一次执行切片下载, 如果是则开启 $\max\text{ ConcurrentTileCount}$ 个 TileDownloadThread 线程(该线程的具体内容见步骤 4), 否则唤醒正在等待下载的所有 TileDownloadThread 线程继续执行.

④ 设计一个切片下载管理线程 TileDownload Thread, 该线程内部包含一个无限循环(循环可以根据标志位退出), 在循环内部判断 TileRequestContainer 中是否还有未处理的请求, 如果没有则说明当前地图的绘制工作已经完成, 此时让 TileDownloadThread 线程等待, 否则从 TileRequestContainer 中取出一个未处理的请求, 执行具体的切片下载工作.

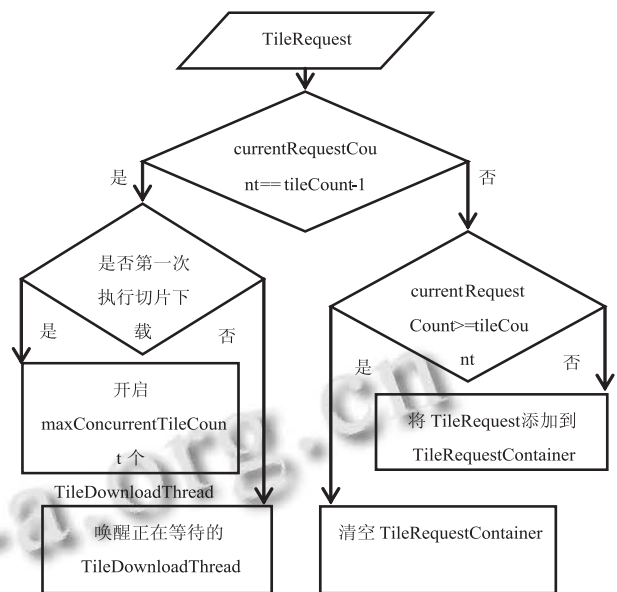


图 1 切片下载算法流程图

2.3 切片显示算法

切片显示算法提供了在地图界面上展示切片的策略. 算法思想: 每次切片下载完成后, 则立即将该切片渲染在界面上, 而不必等待其他切片是否下载完成. 这样可以保证界面的快速响应, 提高用户体验.

切片加载技术主要关注切片的计算、下载、显示问题, 但是为了高效展示地图切片, 还必须考虑以下问题: 下载过的切片还需要重新下载吗? 已经显示在界面上的地图切片下次显示时还需要重新加载吗? 切片缓存技术很好地解决了上述问题.

3 切片缓存技术应用

我们在用手机浏览地图时都会有这样的体验(特别是在网络信号较弱的地区): 第一次浏览某个地图区域时相对较慢, 但是再次浏览先前浏览过的区域时, 速度就大大提高; 同时, 在没有网络的地区也能够浏览之前浏览过的地图, 这就用到了地图缓存技术.

切片地图缓存技术是基于以下理论: 从本地文件中获取切片地图通常比直接从远程服务器上获取切片地图要快, 在网络信号较弱的地区这种优势更加明显; 从内存中获取切片地图要比本地文件中快的多, 理想的情况是把所有的缓存都放在内存中, 但是由于内存资源是稀缺的, 所以不能够把所有的缓存都放在内存中, 否则会造成内存溢出. 基于以上两点考虑, 本文设计出如下切片双缓存方案: 在内存中缓存有限张切

片地图, 缓存的切片个数要比界面上同时显示的切片个数大、所占内存空间要比设备的总内存空间小; 在文件中缓存切片地图, 可以保存成 png、jpg 等格式; 切片绘制时根据索引从内存缓存中获取切片, 如果内存中不存在则从文件中获取。

本文的切片地图缓存技术采用内存和文件双缓存机制, 设计思路如下:

① 设定内存缓存的最大容量为 maxCacheCapacity (假设单位为 KB), 切片地图下载成功后, 将切片保存在内存缓存中; 此时需要判断当前内存缓存的容量是否大于 maxCacheCapacity , 如果是, 说明已达到内存缓存的临界值 maxCacheCapacity , 此时释放整个内容缓存 $N\%$ 的空间(其中 $0 < N < 100$, 比如: 5% 的空间), 清除最早获取的 M 个切片(这 M 个切片所占空间为 $\text{maxCacheCapacity} \times N\%$, 获取这 M 个切片的一种实现方式是维护一个队列利用队列先进先出的特性, 依次执行出队操作直到所占内存满足需求), 然后再将切片保存在内存缓存中, 否则直接保存。

② 将切片保存在文件中, 将切片索引保存在数据库中(本文使用的是嵌入式轻量级数据库 SQLite, 保存在数据库中可以利用数据库的特性便于切片的查找), 同时需要维护数据库和切片文件的同步关系。

③ 地图渲染时, 先从内存缓存中查找是否存在指定切片, 如果存在则直接渲染, 否则查找文件缓存中是否存在指定切片, 如果存在则从文件中取出切片并读到内存中, 执行步骤 1, 然后渲染; 否则, 从远程服务器上下载切片并执行步骤 1、2, 然后渲染。

切片缓存技术是对切片展现技术的有力补充, 利用该技术避免了不必要的地图下载同时加快了切片的查找速度。这两个技术结合在一起, 构成了本文的地图展现策略, 实现过程如图 2 所示。

4 结语

本文首先介绍了切片地图的相关概念并介绍了地图切片技术原理, 重点探讨了使用地图加载技术和切片缓存技术在移动 GIS 系统中快速、高效地展现切片地图这一问题。

利用上述地图展现策略, 本文的作者分别在 Windows Mobile 平台和 Android 平台上实现了移动 GIS 地图浏览系统, 系统界面如图 3 所示。作者将该系统用于电力行业的移动作业项目中, 并取得了良好的

效果, 即使在网络环境较差的地区地图浏览也十分流畅。同时, 利用本文的切片缓存技术, 作者将移动 GIS 系统升级为支持离线切片地图浏览的移动 GIS 系统, 以便支持离线作业场景的需要。

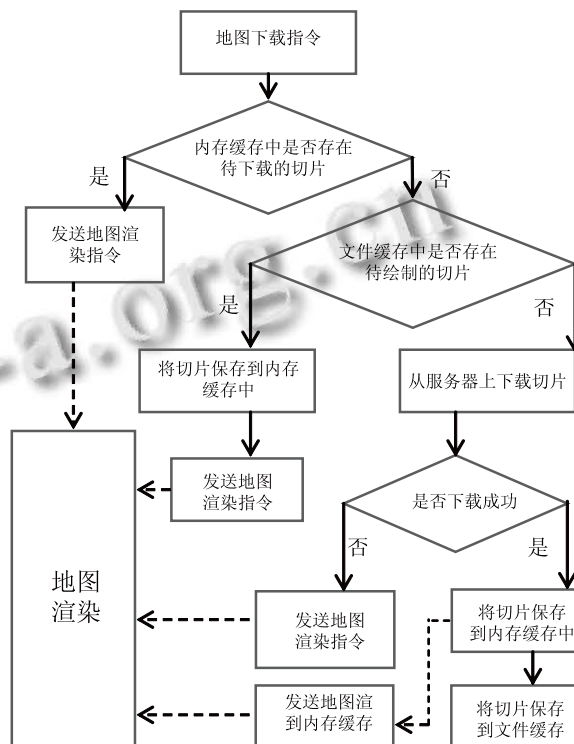


图 2 切片地图展现策略流程图

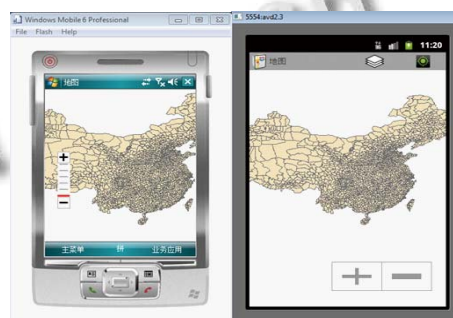


图 3 Windows Mobile、Android 版移动 GIS 客户端

参考文献

- 1 张园. 移动位置服务应用发展研究. 信息技术, 2011(2): 42-46.
- 2 黄梦龙. 瓦片地图技术在桌面端 GIS 中的应用. 地理空间信息, 2011, 9(4): 149-151.
- 3 赵大龙, 孙恒宇. 地图切片技术分析 with 简单实现. 测绘与空间地理信息, 2010, 33(1): 116-118.