

Research on Electronic Map Drawing Method Based on OpenGL and Google Maps Tile Data

Guotao Zhu, Jianping Cao

Naval Aviation University, Qingdao Branch, Qingdao Shandong
Email: zealotsparc@163.com

Received: June 6th, 2019; accepted: June 21st, 2019; published: June 28th, 2019

Abstract

This paper introduces the projection mode of Google Map and the organization mode of raster tile data. By reorganizing, merging and indexing the raster tile data of Google Map, an off-line and single map data package is constructed. Base on OpenGL graphics rendering engine and multi-threaded rendering technology, a high-speed and real-time 2-dimensional electronic map control is designed and implemented, which is applied to 2-dimensional battlefield situation, battle scenario editing and other components to avoid the problem that detailed vector data is precious and difficult to obtain, and it is difficult to implement the render and draw technology when using vector map

Keywords

Electronic Map, Raster Tile, Teal Time Drawings

基于OpenGL和谷歌地图的电子地图实时绘制方法研究

朱国涛, 曹建平

海军航空大学青岛校区, 山东 青岛
Email: zealotsparc@163.com

收稿日期: 2019年6月6日; 录用日期: 2019年6月21日; 发布日期: 2019年6月28日

摘要

介绍了谷歌地图的投影方式和栅格瓦片数据组织格式, 通过对谷歌地图栅格瓦片数据的重新组织、合并

和索引构建了离线的、单一的地图数据包文件, 基于OpenGL图形渲染引擎和多线程技术, 设计实现了高速实时的二维电子地图控件, 经实际应用满足了二维战场态势、作战想定编辑等组件对于二维电子地图实时显示的要求, 避免了使用矢量地图时详细的矢量数据珍贵且难以获取、渲染和绘制技术实现困难的问题。

关键词

电子地图, 栅格瓦片, 实时绘制

Copyright © 2019 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

电子地图即数字地图, 是利用计算机技术, 以数字方式存储和查阅的地图, 与传统地图相比, 电子地图具有界面友好、操作方便、交互性强、信息丰富、直观立体等特点, 是各种地理信息系统的核心, 广泛应用于测绘、规划、交通、旅游等各个行业和领域。

目前构建二维电子地图的手段主要分为两种: 以矢量地图数据为基础的矢量方式和以栅格图像数据为基础的栅格方式。其中矢量方式具有可以无级缩放和旋转、支持分层显示、支持交互式编辑等优点, 但大范围的、详细的、精确的矢量地图数据比较珍贵且难以获取, 基于矢量瓦片技术的矢量地图渲染技术虽然可以解决地图实时性的问题, 但是其矢量瓦片切分、数据组织、地图渲染和绘制都较为复杂[1][2]。与此同时基于栅格方式的互联网电子地图(诸如谷歌地图、百度地图等)的发展为用户开放且免费提供了内容和细节相对丰富的栅格地图数据, 本文将针对军事仿真应用的需求, 以批量下载的谷歌地图栅格数据为基础, 在离线环境下以 OpenGL 图形渲染技术为核心设计和实现了实时的二维电子地图控件。

2. 谷歌地图介绍

谷歌地图是指 GOOGLE 公司提供的覆盖全球主要国家和地区的电子地图服务, 其前身是 KEYHOLE 公司研发的一款电子地图软件, 2005 年被谷歌公司收购后, 其地图数据覆盖范围扩展至全球。谷歌地图提供三种类型的地图数据: 即普通电子地图、卫星地图和地形图, 并通过浏览器向用户以常见的栅格图像方式提供地图数据。自推出以来, 谷歌地图以其强大的功能, 简单方便的操作, 丰富的地理交通及商业信息资源, 覆盖全球的数据, 吸引了越来越多的用户。谷歌地图数据量非常庞大, 为解决在 WEB 环境下使用时, 网络带宽和浏览器处理渲染地图能力的不足, 谷歌地图采用了 WEB 墨卡托投影和瓦片分级切割的方案, 这也是目前互联网地图事实上的标准[3]。

2.1. WEB 墨卡托投影介绍

墨卡托投影是正轴等角圆柱投影, 由荷兰地图学家墨卡托于 1596 年创立。其投影方式为: 假想一个与地轴方向一致的圆柱切或割于地球, 按等角条件, 将经纬网投影到圆柱上, 按圆柱面展开为平面后, 即得本投影。WEB 墨卡托辅助球投影(EPSC: 3857)是将球面墨卡托投影公式运用于椭球面坐标的投影计算, 为计算简单和实现方便, WEB 墨卡托投影假设地球为球体, 其半径取为 WGS84 椭球的长半轴半径 6,378,137.0 米, 这样理论上精度可以控制在 0.33%之内。WEB 墨卡托投影如图 1 所示, 该投影坐标原点位于赤道和本初子午线的交点, 经线和经线之间相互平行且间隔相等, 对应的经度 ± 180 度其范围是士

20,037,508.34 米。其投影纬线和经线之间也相互平行, 间隔从赤道开始向两级逐渐增大, 为便于计算机处理, 提高显示效率, 取地球纬度范围也是 $\pm 20,037,508.34$ 米, 使地图呈正方形, 经过反算可得其对应的纬度范围是 ± 85.05 度[4] [5]。

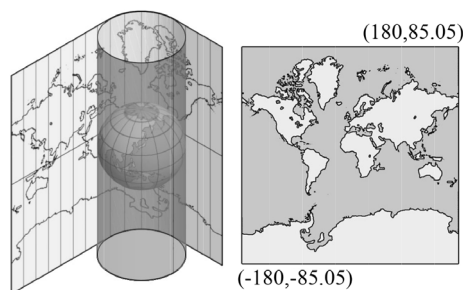


Figure 1. WEB Mercator projection

图 1. WEB 墨卡托投影

WEB 墨卡托投影在 2005 年发布的谷歌地图中首次使用, 考虑到相互之间的平台兼容, 随后跟进的微软 BING MAPS、在线地图服务公司的 MAPQUEST MAPS、雅虎公司的 YAHOO MAPS 均采用了谷歌地图的 WEB 墨卡托地图投影方式。

2.2. 栅格瓦片的组织方式

用户通过浏览器使用谷歌地图时, 看到的是一张铺面整个窗口的地图图片。但是实际上, 这张大的地图图片是由多个尺寸相同(通常是 256×256 像素)的小图片按照既定规则无缝拼接而成的。这些小图片就是栅格瓦片(简称瓦片), 瓦片按照如图 2 所示的金字塔结构组织, 每张瓦片都可以通过细节等级 LOD 级别、列号、行号三个要素唯一标记。瓦片地图金字塔模型是一种多分辨率层次模型, 从瓦片金字塔的顶层开始, 分辨率越来越高, 瓦片数量越来越多, 但每个层次表示的地理范围不变, 例如当 LOD 级别为 0 时, 整个地图就只有一个 256×256 像素的瓦片组成, 当 LOD 级别为 1 时, 地图分裂为 4 个瓦片, 这 4 个瓦片排列成两列两行, 依此类推, 每放大一倍, 每一块小瓦片都分裂为四块。因此当缩放等级为 n 时, 地图分裂为 $2n \times 2n$ 个瓦片, 这些瓦片排列成 $2n$ 行 $2n$ 列, 按照从左到右, 从上到下的顺序给瓦片编号, 即可以通过瓦片 LOD 级别、列号、行号定位瓦片。目前基本上所有的在线栅格地图均采用了瓦片地图的存储方式[6]。

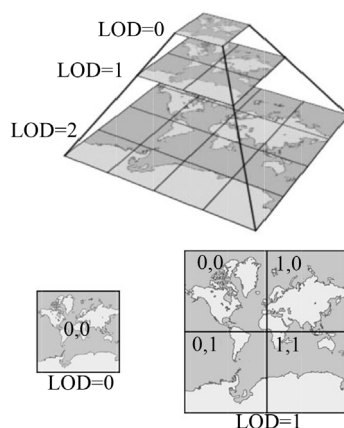


Figure 2. LOD level and pyramid structure of Maps

图 2. 地图的 LOD 等级和金字塔结构

3. 地图控件设计和实现

离线的、实时的二维电子地图需要首先获取地图的栅格瓦片数据, 然后按照地图的投影方式和瓦片的组织规则, 在 OpenGL 图形渲染环境中根据用户交互信息读取和渲染瓦片, 以多个瓦片进行无缝拼接成为用户可见的电子地图。因此如图 3 所示, 二维电子地图从结构组成上需要包括用户交互处理模块、栅格瓦片读取模块、地图渲染绘制模块以及为进行瓦片数据处理的瓦片处理工具。

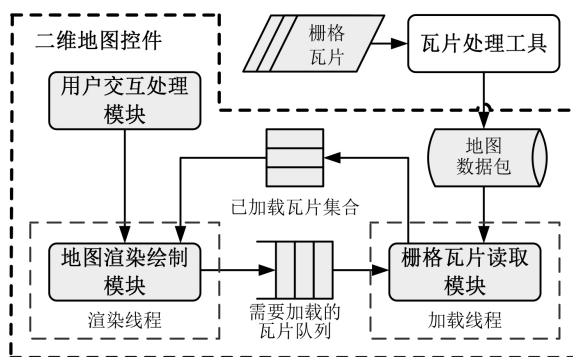


Figure 3. Structure composition of 2-dimensional electronic Map control

图 3. 二维电子地图控件的结构组成

1) 交互处理模块负责响应用户的地图平移、缩放、测距等交互操作, 这些操作将被封装成地图显示信息(地图显示范围、LOD 等级等)用于驱动地图渲染模块进行地图的更新显示;

2) 地图渲染绘制模块首先根据当前地图显示范围、LOD 等级等信息计算需要加载显示的瓦片单元, 并且将瓦片单元加入到一个瓦片队列用于驱动栅格瓦片读取模块加载栅格瓦片, 然后从带索引标识的加载完毕的瓦片集合中获取瓦片信息并以 OpenGL 图形渲染技术绘制和显示地图瓦片。由于实时性要求, 地图渲染绘制模块工作在一个单独的渲染线程中;

3) 栅格瓦片读取模块依次读取需要加载的瓦片单元信息, 从地图数据包中读取对应的瓦片数据, 然后将结果加入到已加载瓦片集合供地图渲染绘制模块使用。为避免读取仿真数据拖慢地图渲染绘制的速度, 瓦片读取模块也工作在一个独立的线程中;

4) 瓦片处理工具功能是根据指定的地图覆盖范围、细节等级 LOD 级别范围等参数, 对分散在不同目录下的地图栅格瓦片进行处理, 最终生成一个单一文件的地图数据包。

3.1. 地图数据包的文件格式

通过地图批量下载工具下载的电子地图瓦片是 JPG 或 PNG 格式的图片, 通常以图 4 所示的目录结构存放到磁盘上, 其中第一级目录表示瓦片的 LOD 等级, 第二级目录名称表示瓦片所在的列号, 第三级文件名称是瓦片所在的行号, 由于瓦片文件分散在不同的目录下, 在渲染和绘制时如果用户进行了缩放、平移等操作, 则程序将需要频繁的进行瓦片数据加载和卸载, 如果每次加载瓦片都需要进行一次比较耗时的文件打开或关闭操作, 则会影响整个地图控件渲染绘制和交互响应的实时性。

为此可以将所有的地图瓦片打包到一个文件中, 形成一个单一文件的地图数据包, 然后在地图控件启动时打开地图数据包文件, 之后每次需要加载瓦片时都从地图数据包中读取, 就可以大大加快地图瓦片加载的速度, 提高地图交互响应的实时性。根据需求, 地图数据包的文件格式如图 5 所示, 它由三个部分组成, 分别是文件头、瓦片数据和索引数据。

1) 文件头: 包括地图数据包的文件版本、瓦片数据缩放范围(即对应地图瓦片的 LOD 级别的最小值

和最大值)、瓦片索引数据起始位置在地图数据包文件中的偏移量、瓦片索引数据的长度以及地图边界(分别用经度和纬度的最小值、最大值表示)。

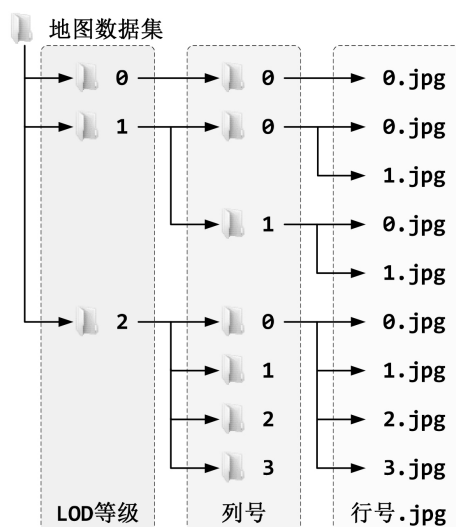


Figure 4. Map data sets tile files storage

图 4. 地图数据集瓦片文件存储方式

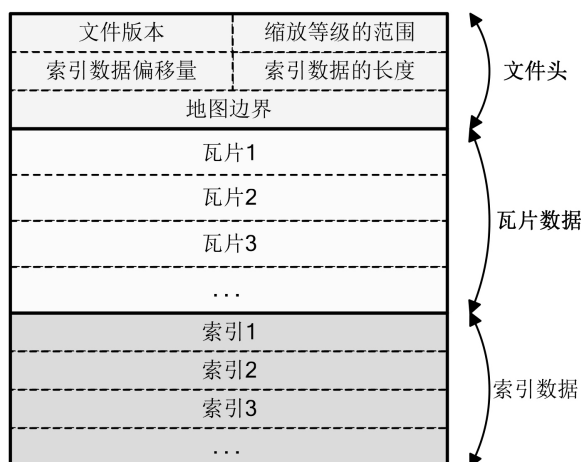


Figure 5. File format of Map packet

图 5. 地图数据包的文件格式

2) 瓦片数据:即将一定覆盖范围和LOD级别内的所有栅格瓦片文件依次集中存放的地图瓦片数据,其存储方法为:依次读取代表地图瓦片的JPG或PNG格式图片文件的内容,然后依次写入到地图数据包的瓦片数据块中。

3) 索引数据:为加载瓦片时快速定位瓦片地址而设计,其内容包括两部分:瓦片ID和瓦片信息,瓦片ID的内容包括瓦片的LOD级别、列号、行号、类型(普通地图、卫星地图或者地形图),瓦片信息的内容包括瓦片数据在地图数据包中的偏移量和长度。在C++中瓦片索引数据的格式如下:

```
struct mapTileIndex
{
    struct mapTileID
    {
```

```

unsigned short nLOD; //细节等级
unsigned short nROW; //瓦片行号
unsigned short nCOL; //瓦片列号
unsigned short nType; //地图类型
} _id;
struct mapTileINFO
{
    unsigned __int64 _pos; //瓦片位置
    unsigned int _len; //瓦片长度
} _info;
};

```

通过上述定义, 控件加载在地图数据包时, 可以首先根据文件头中的索引数据偏移量确定在索引数据在地图数据包文件中位置, 然后依次读取索引数据并将其存储起来作为加载地图瓦片的索引。

3.2. 栅格瓦片读取

栅格瓦片数据经过瓦片处理工具的处理, 分散的瓦片地图数据集被打成单一的地图数据包文件, 当二维电子地图控件启动时, 栅格瓦片读取线程随之启动。之后在该线程的每一次循环中程序都按照以下流程进行处理, 直至线程随二维电子地图控件关闭而结束:

1) 在线程循环开始时, 首先检查需要加载的瓦片队列是否为空, 如果为空, 则延迟 1 毫秒后重新进入线程循环, 否则进行步骤 2。

2) 从队列中取出一个瓦片, 然后依据瓦片标识(包括 LOD 级别、列号、行号)等信息在索引数据中二分法查找瓦片信息, 然后根据瓦片数据在地图数据包中的偏移量和长度, 从地图数据包文件中加载指定的瓦片数据。

3) 瓦片数据加载完毕后, 将瓦片数据(即栅格图像数据)添加到瓦片集合供地图渲染绘制模块使用, 然后再次跳转到步骤 1, 直至需要加载的瓦片队列处理完毕。

3.3. 地图渲染绘制

为保证实时性, 地图的渲染绘制也在一个独立的渲染线程中完成, 渲染线程的主体是一个渲染循环, 在每一次循环中程序均要完成绘制准备、瓦片绘制、瓦片清理等工作:

一、绘制准备

在绘制准备阶段, 首先需要获取控件客户区的大小(即地图显示区域的大小)、当前的 LOD 级别、窗口中心对应的地图位置等信息, 然后计算得到需要参与绘制的瓦片数组, 计算方法如图 6 所示。

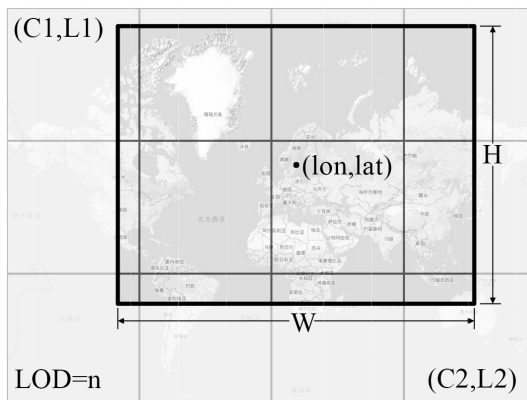


Figure 6. Tiles calculation which participation in rendering
图 6. 参与渲染的瓦片计算

1) 首先, 已知当前 LOD 级别 n , 可以计算瓦片在 WEB 墨卡托投影坐标下的瓦片的宽度和高度均为 $T = 2 * 20,037,508.34 / 2^n$, 由于瓦片像素尺寸为 256×256 , 所以可以计算每个像素对应的 WEB 墨卡托投影坐标下的宽高为 $L = T / 256$ 。

2) 然后, 根据窗口中心的 WEB 墨卡托投影坐标(lon,lat)和控件客户区域的宽度 W 和高度 H , 可以计算客户区左上角和右下角对应的地图 WEB 墨卡托投影下的坐标分别为(wmX_LT, wmY_LT)和(wmX_BR, wmY_BR), 计算代码如下:

```
double wmX_LT=lon-W*L/2;
```

```
double wmY_LT=lat+H*L/2;
```

```
double wmX_BR=lon+W*L/2;
```

```
double wmY_BR=lat-H*L/2。
```

最后, 分别计算客户区左上角和右下角对应的瓦片列号、行号, 部分计算代码如下:

```
int C1=(int)floor((20037508.34+wmX_LT)/T);
```

```
int L1=(int)floor((20037508.34-wmY_LT)/T);
```

```
int C2=(int)ceil((20037508.34+wmX_BR)/T);
```

```
int L2=(int)ceil((20037508.34-wmY_BR)/T)。
```

计算完毕后, 所有处于 C1 到 C2 列、L1 到 L2 行中间的瓦片都需要参与绘制。

二、瓦片绘制

在瓦片绘制阶段, 需要依次遍历步骤 1 计算所得的需要参与显示绘制的瓦片数组, 然后对每个瓦片按如下步骤进行绘制显:

1) 计算瓦片在 OpenGL 中绘制显示必需的位置信息。

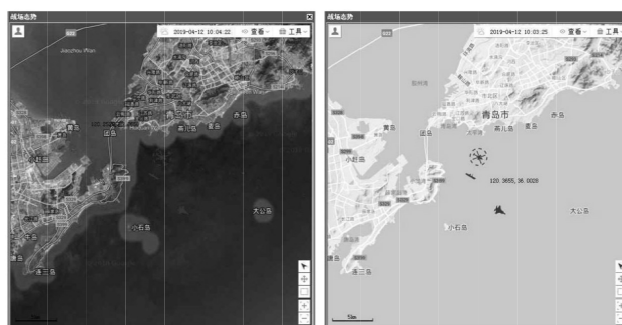
2) 根据瓦片标识从瓦片集合中查找瓦片, 如果查找成功, 则继续步骤 3, 否则跳转到步骤 4。

3) 根据步骤 2 查找所得的瓦片像素数据, 生成并绑定对应的纹理, 然后使用 OpenGL 绘制在指定位置绘制瓦片, 然后跳转到步骤 1 继续遍历。

4) 将瓦片标识添加到需要加载的瓦片队列, 驱动瓦片加载线程进行瓦片数据加载。然后跳转到步骤 1 继续遍历。

三、瓦片清理

在瓦片清理阶段, 程序需要再次遍历瓦片集合, 检查瓦片集合中的瓦片是否需要参与当前的显示绘制(即判断瓦片是否在需要参与显示绘制的瓦片数组中), 如果瓦片不需要参与显示绘制, 则释放瓦片纹理和图像数据, 并从集合中删除瓦片。



1. 卫星图

2. 地形图

Figure 7. Drawing effect of map control based on OpenGL

图 7. 基于 OpenGL 实现的地图控件绘制效果

4. 结论

由于保密需要, 军事仿真领域对二维电子地图的基本要求是能够离线使用, 本文在理解 WEB 墨卡托投影方式和瓦片文件组织结构基础上, 通过对瓦片的重新组织, 基于 OpenGL 图形渲染技术设计实现了实时的二维电子地图控件, 其界面如图 7 所示, 控件在硬件配置为 Intel Core i7-6700HQ 处理器, 英伟达 GeForce 940MX 显卡, 8 GB DDR4 2133 MHz 内存, 7200 转机械硬盘, 地图数据对应的 LOD 等级为 8 到 15 级, 地图最大分辨率为 5 米左右, 地图控件大小为 1920 × 1080 像素时, 实测地图控件的刷新率稳定保持在 50 到 60 赫兹。实测表明该控件可以维持较高的渲染效率, 虽然具有不能无级缩放和旋转、栅格化之后丢失地理信息等问题, 但是其数据易于获取、显示内容丰富、操作界面友好、信息直观立体, 因此在二维战场态势、作战想定编辑等方面具有一定的应用价值。

参考文献

- [1] 孙晨龙, 霍亮, 高泽辉. 基于矢量瓦片的矢量数据组织方法研究[J]. 测绘与空间地理信息, 2016(4): 122-124.
- [2] 唐雅玲, 马晨燕, 宋佳, 等. 矢量 WEB 网络地图应用研究[J]. 测绘与空间地理信息, 2018(2): 87-90+93+96.
- [3] 张国伟. 基于谷歌地图的智慧路灯管控系统设计及实现[D]: [硕士学位论文]. 合肥: 合肥工业大学, 2017.
- [4] 何碧容, 蔡倩. 基于 WEB 墨卡托投影的导航电子地图设计[J]. 计算机测量与控制, 2017, 25(1): 119-122.
- [5] 李鹤元, 陈刚. 基于改进 WEB 墨卡托投影的瓦片地图服务设计与实现[J]. 测绘工程, 2016, 25(2): 11-16.
- [6] 李长春, 蔡伯根, 上官伟, 等. 基于 WEB 墨卡托投影的地图算法研究与实现[J]. 计算机应用研究, 2012, 29(12): 4793-4796.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-2286, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>
期刊邮箱: sea@hanspub.org