

Google Map 数字栅格地图算法及应用

寇曼曼^{1,2}, 王勤忠¹, 谭同德²

(1. 河南省地震局, 河南 郑州 450018;

2. 郑州大学 信息工程学院, 河南 郑州 450001)

摘要: Google 地图是 Google 向全球提供的电子地图服务, 包括丰富的地图和黄页资源, 特别是局部详细的卫星地图。通过具体分析 Google Map 不同坐标系下的坐标转换、Web 墨卡托投影方式、地图瓦片 URL 组织结构, 系统地研究了 Google 地图的算法及实现, 并在 VS2008 环境下用 C# 语言实现了任意缩放系数下指定地图类型 URL 的合成和对应图片的下载。文中列出了部分程序代码以帮助读者进行后续开发设计, 更有效地将 Google 地图集成到应用系统中。实验证实, 该方法能够快速准确地下载指定范围的地图。

关键词: 栅格地图; 墨卡托投影; 瓦片; 地理信息系统

中图分类号: P208

文献标识码: A

文章编号: 1673-629X(2012)04-0204-03

Research on Google Map Algorithm and Application

KOU Man-man^{1,2}, WANG Qin-zhong¹, TAN Tong-de²

(1. Earthquake Administration of Henan Province, Zhengzhou 450018, China;

2. School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: Google Maps which offer web mapping are famous for their rich map and yellow page resource in the world. According to the analysis of the Google Map's coordinate, projection and URL's elements, the algorithm and implementation of Google Map is systematically studied. With the help of VS2008, finish the composing algorithm of URL under any zoom and map types, and download corresponding tiles in the C# language. The program in the paper may help better design and integrate Google Map function into application system. Making use of this method is able to download the designated maps quickly and exactly.

Key words: raster map; Mercator projection; tile; GIS

0 引言

数字栅格地图(DRG, Digital Raster Graphic)一般是根据现有纸质、胶片等地形图经扫描和几何纠正及色彩校正后, 形成在内容、几何精度和色彩上与地形图保持一致的栅格数据集。还有一类是以 Google Map、Google Earth 为代表的以互联网为载体的数字栅格地图, 它以其丰富的地图和黄页资源, 特别是覆盖全球的卫星地图而著称。

地理信息系统可以方便地对空间数据进行采集、分析、管理、输出等, 并具有区域分析、多种要素分析和动态预测的能力^[1]。通过研究 Google 地图的算法及实现, 可以为地理信息系统的应用集成提供参考。文献[2]就是针对 Google Map 地图算法和应用的研究。

由于时间的推移及其他原因, Google Map 的瓦片分割策略已经与文献所给出的方法有所不同。因此, 文中再次探讨 Google Map 数字栅格地图的算法及应用, 并在 VS2008 环境下, 用 C# 语言实现了任意缩放系数下指定地图类型 URL(统一资源定位符, Uniform / Universal Resource Locator 的缩写)的合成和对应图片的下载。

1 Google Map 数学模型

1.1 坐标系

Google Map 定义了多重坐标系统, 分别是地理坐标系、投影坐标系、像素坐标系以及瓦片坐标系, 各个坐标系统之间能够互相转换。Google Map 的地理坐标系^[3]以经纬度为单位。经过墨卡托投影后生成投影坐标系, 以米为单位。这两个坐标系都是以本初子午线和赤道的交点为原点, 横轴往东为正, 纵轴往北为正, 可通过墨卡托投影的公式进行转换。Google Map 是由分辨率大小 256 * 256 的图片无缝拼接而成的, 称

收稿日期: 2011-08-29; 修回日期: 2011-12-02

作者简介: 寇曼曼(1981-), 女, 河南邓州人, 工程师, 硕士, 研究方向为图像处理与分析、计算机应用; 王勤忠, 高级工程师, 研究方向为软件工程、计算机网络等; 谭同德, 教授, 博士, 研究方向为计算机图形学、虚拟现实、CAD。

之为瓦片。Google Map 定义了瓦片坐标系^[4],坐标系的原点在西经 180 度,北纬 85.05113 度,即是原来墨卡托投影坐标系的左上角,坐标系横轴往东为正,纵轴往南为正。Google Map 还定义了像素坐标系,经纬度坐标与像素坐标是一一对应的关系。像素坐标系来源于瓦片坐标系,是将每张瓦片分割成 $256 * 256$ 个像素点,所以像素坐标系的最大值是瓦片坐标系最大值的 256 倍。投影坐标系可根据 1.3 中的公式转换到像素坐标系。

1.2 地球椭球体数学模型

地图投影就是按照一定数学法则,将地球椭球面上的经纬网转换到平面上,使地面点位的地理坐标与地面上相对应的点位的平面直角坐标或平面极坐标间建立起一一对应的函数关系^[5,6]。地图投影的拟定和计算,一般均假定地球表面为旋转椭球面,并称其为地球椭球面或参考椭球面。

如图 1 所示,地球椭球体的形状和大小是由其长半径 a_e (赤道半径) 和短半径 b_e (极轴半径) 决定,通常取 a_e 值为 6378137m, b_e 值为 6356752.3m。由于地球是一个赤道略宽两极略扁的不规则的梨形球体,故其表面是一个不可展平的曲面,所以运用任何数学方法进行这种转换都会产生误差和变形,为按照不同的需求缩小误差,就产生了各种投影方法。

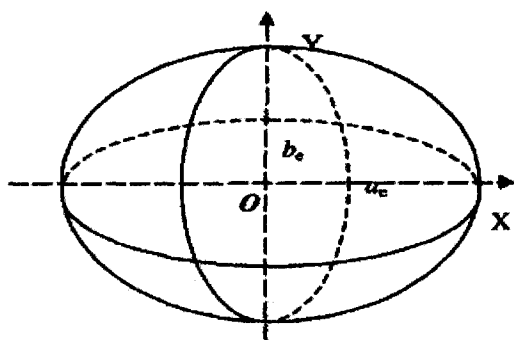


图 1 地球椭球体示意图

1.3 墨卡托 (Mercator) 投影

墨卡托投影^[7],是一种“等角正切圆柱投影”,荷兰地图学家墨卡托 (Gerhardus Mercator 1512-1594) 在 1569 年拟定。假设地球被围在一个中空的圆柱里,其赤道与圆柱相接触,然后再假想地球中心有一盏灯,把球面上的图形投影到圆柱体上,再把圆柱体展开就是一幅标准纬线为零度 (即赤道) 的“墨卡托投影”绘制出的世界地图。

墨卡托投影没有角度变形^[8],由每一点向各方向的长度比相等,它的经纬线都是平行直线,且相交成直角,经线间隔相等,纬线间隔从赤道向两极逐渐增大。墨卡托投影的地图上长度和面积变形明显,但标准纬线无变形,从标准纬线向两极变形逐渐增大,但因为它

具有各个方向均等扩大的特性,保持了方向和相互位置关系的正确。在地图上保持方向和角度的正确是墨卡托投影的优点,这样在查询地物的方向时就不会出错。

简化后的墨卡托投影公式如下 (公式中 θ 为经度, φ 为纬度, R_e 为地球半径)

$$x = R_e * \theta$$

$$y = R_e * \ln \left(\tan \left(\frac{\pi}{4} + \frac{\varphi}{2} \right) \right)$$

当 y 等于 πR 时,投影后的图正好为一个正方形,可得出纬度为 85.05113 度,所以 Google Map 的最大纬度值为 ± 85.05113 度,最大经度值为 ± 180 度。

1.4 Google Map 地图投影

Google Maps 所使用的地图投影,常被称作 Web Mercator,它与常规墨卡托投影的主要区别就是把地球模拟为球体而非椭球体^[9]。以整个世界范围,赤道作为标准纬线,本初子午线作为中央经线,两者交点为坐标原点,向东向北为正,向西向南为负。由于赤道半径为 6378137 米,则赤道周长为 $2 * \pi * R = 2 * 20037508.3427892$,因此投影坐标系中 X 轴的取值范围: $[-20037508.3427892, 20037508.3427892]$, Y 轴的取值范围也限定在 $[-20037508.3427892, 20037508.3427892]$ 之间。地理坐标和像素坐标的转换关系如下:

$$x_p = \text{centerP} + \theta * \frac{\text{totalP}}{2\pi}$$

$$y_p = \text{centerP} - 0.5 * \ln \left(\frac{1 + \sin(\varphi)}{1 - \sin(\varphi)} \right) * \frac{\text{totalP}}{2\pi}$$

$$\text{centerP} = 2^{\text{Zoom}+7}, \text{totalP} = 2 * \text{centerP}$$

其中 Zoom 为地图缩放系数, θ 为经度, φ 为纬度, x_p, y_p 为像素坐标。

对应的瓦片坐标为:

$$x_T = \frac{x_p}{256}$$

$$y_T = \frac{y_p}{256}$$

其中 x_p, y_p, x_T, y_T 均取整数。

1.5 Google Map 瓦片 URL 地址

目前,使用最普遍的一种全球影像数据组织方法是金字塔模型法^[10,11]。层次最高为 0 级,即为金字塔的顶部,此时影像和地形数据的分辨率最低;层次越低,数据分辨率越高。Google Map 是由分辨率大小为 $256 * 256$ 的图片无缝拼接而成的,在初始等级时,整个地球只投影在一张 $256 * 256$ 的图片上。每放大一级每张图片都会分裂成四张,比例尺则减小为原来的一半。最后随着缩放等级的增大,瓦片自顶向下呈几何级数的增长,形成一个金字塔状。其实 Microsoft

Virtual Earth, Yahoo Maps 等都采用类似的组织结构。唯一不同的是编码方式不一样,比如 Google 的瓦片坐标系 XYZ,微软的四叉树,OSGeo 的 TMS 等^[12]。

Google Map 拥有包括地图、遥感图、地形图及独立的地标等多种地图形式。每个地图瓦片均由独立的 URL 地址唯一确定,如 URL `http://mt2.google.cn/vt/lyrs=m@132&hl=zh-CN&gl=cn&x=1670&y=812&z=11&s=Galileo`,其中参数 `mt2` 表示服务器名,Google Map 有 `mt0~mt3` 四个服务器,可以分别从各个服务器获取图片,减轻了服务器的压力。`lyrs` 代表地图类型;`@` 后面代表版本号;`hl` 为语言;`x,y,z` 分别代表瓦片的 x 坐标, y 坐标以及缩放等级 `zoom`。

对于每个 URL 可以分解为基本 URL 及 x,y,z 组成,基本 URL 由地图源、地图类型及版本号确定,在一定周期内除了版本号经常变化外基本 URL 一般不变,下面为几个常用数据源的地图基本 URL:

`http://maps.google.cn`

`baseUrls[] = { "http://mt0.google.cn/vt/lyrs=m@132&hl=zh-CN&gl=cn&", "http://mt1.google.cn/vt/lyrs=m@132&hl=zh-CN&gl=cn&", "http://mt2.google.cn/vt/lyrs=m@132&hl=zh-CN&gl=cn&", "http://mt3.google.cn/vt/lyrs=m@132&hl=zh-CN&gl=cn&" }`

`http://maps.google.tw`, `http://maps.google.com`, `http://maps.google.jp`

`baseUrls[] = { "http://mt0.google.com/vt/lyrs=m@132&hl=zh-CN&", "http://mt1.google.com/vt/lyrs=m@132&hl=zh-CN&" }`

其中版本号对应的 132 会随时间变化较快,同样对于卫星影像、地形数据及地标数据也均有相应的基本 URL,在此不再赘述。

完整的 URL 可以通过函数 `getTitleUrl` 获取,以下为 C# 代码。

```
// itile 为包含瓦片坐标 x,y 的结构
public string getTitleUrl(string[] baseUrls, itile tile,
int zoom)
{
    int i = (tile.x + 2 * tile.y) % baseUrls.Length;
    string salt = "Galileo".Substring(0, (tile.x * 3 +
tile.y) % 8);
    string middleSalt = "";
    if (tile.y >= 10000 && tile.y < 100000)
        middleSalt = "&s=";
    return baseUrls[i] + "x=" + tile.x.ToString() + mid-
```

dle

```
Salt + "&y=" + tile.y.ToString() + "&z=" + zoom.ToString() + "&s=" + salt;
}
```

2 应用设计

有了完整的 URL 在 C# 中利用 `HttpWebRequest` 类可实现瓦片地图的下载。在获取瓦片地图的基础上,首先应用图像拼接技术,然后结合运用图像处理和模式识别方法可以开发特定的 GIS 应用系统。

3 结束语

文中给出了 Google Map 数字栅格地图在任意缩放系数下指定地图类型 URL 的合成和对应图片的下载方法,可以为专用地理信息系统 (Geographic Information System, GIS) 的集成提供参考,在此基础上开发适用于不同领域的专项 GIS 应用系统。

参考文献:

- [1] 李东平,姚 远. GIS 的发展趋势与数字地震应急救援的实现技术[J]. 计算机技术与发展, 2011, 21(1): 214-217.
- [2] 崔金红,王 旭. Google 地图算法研究及实现[J]. 计算机科学, 2007, 34(1): 193-195.
- [3] 梁世文,王春光,尹志永. ArcGIS 坐标系统探讨[J]. 地理空间信息, 2010, 8(4): 152-154.
- [4] Google Maps API Documentation [EB/OL]. 2011. http://code.google.com/intl/zh-CN/apis/maps/documentation/javascript/v2/overlays.html#Google_Maps_Coordinates.
- [5] 韩丽君,安建成. 地图投影及其在 GIS 中的应用[J]. 科技情报开发与经济, 2009, 19(8): 136-138.
- [6] Yang Qihe. Principle and Method of the Transformation of Map Projection[M]. Beijing: Publishing House of PLA, 1989.
- [7] 孙 达,蒲英霞. 地图投影[M]. 南京: 南京大学出版社, 2009.
- [8] Williams R T. Lambert and Mercator map projections in geology and geophysics[J]. Computers and Geosciences, 1995, 21(3): 353-364.
- [9] 许 辉,马晓鹏. 基于 Web 墨卡托投影地理信息系统设计与实现[J]. 电脑编程技巧与维护, 2011(8): 41-43.
- [10] 陈 刚,柯希林,朱 军. 基于金字塔模型的海量地形数据管理算法[J]. 测绘科学与工程, 2007, 27(1): 15-18.
- [11] 李少梅,阚映红,王 颖,等. Google Earth 影像变换与应用研究[J]. 测绘科学与工程, 2010, 30(2): 11-14.
- [12] Tiles ala Google Maps: Coordinates, Tile Bounds and Projection [EB/OL]. 2011. <http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/>.