

引用格式: Huang Ruosi, Li Chuanrong, Feng Lei, *et al.* Geometry based Rendering of Vector Data with WebGL [J]. Remote Sensing Technology and Application, 2014, 29(3): 463-468. [黄若思, 李传荣, 冯磊, 等. 基于几何的 WebGL 矢量数据三维渲染技术研究[J]. 遥感技术与应用, 2014, 29(3): 463-468.]
doi:10.11873/j.issn.1004-0323.2014.3.0463

基于几何的 WebGL 矢量数据三维渲染技术研究

黄若思^{1,2}, 李传荣¹, 冯磊¹, 唐伶俐¹

(1. 中国科学院光电研究院定量遥感信息技术重点实验室, 北京 100094;

2. 中国科学院大学, 北京 100049)

摘要: 针对现有 3D WebGIS 系统矢量数据三维可视化在渲染速度与渲染效果方面存在明显不足的问题, 利用新一代 Web 三维绘图标准 WebGL 的优势, 研究了基于几何的 WebGL 矢量数据三维渲染技术, 提出了在水平方向与垂直方向上进行矢量数据渲染优化的方法, 并在此基础上开发了基于 WebGL 的矢量数据三维渲染原型系统, 对渲染方法与渲染效果进行了验证, 证明了基于几何的 WebGL 矢量数据三维渲染方法的可行性与有效性。

关键词: WebGL; 细节层次技术; 矢量数据渲染; 3D WebGIS

中图分类号: P 208 文献标志码: A 文章编号: 1004-0323(2014)03-0463-06

1 引言

矢量数据作为地理信息系统的两种基本数据类型之一, 具有数据结构紧凑、冗余度低、表达精度高、图形显示质量好并且有利于网络传输和检索分析等优点, 在 GIS 尤其是网络地理信息系统中得到了广泛应用。随着 WebGIS 的发展, 地理信息场景的表现形式也逐渐由二维向三维方向发展, 如何在三维 WebGIS 系统(3D WebGIS)中快速、有效地渲染矢量数据成为当前 GIS 空间数据可视化的一个重要研究方向。

早期的 3D WebGIS 是基于虚拟现实建模语言(VRML)实现的^[1], 客户端浏览器通过安装 VRML Viewer 专用插件实现 Web 三维渲染。由于 VRML 语言几乎不支持其他所有的结构化数据编程语言, 且不可移植, 协同性与稳定性都很差, 无法满足 3D WebGIS 三维渲染的应用需求。鉴于 VRML 在 3D WebGIS 应用中的诸多不足, Web3D 联盟于 2001 年 8 月发布了新一代的三维图像标记语言 X3D, 它

是 VRML 的升级版, 具有跨平台、可扩展等优点, 虽经近 10 a 的发展, 但其制作工具和开发环境都比较落后, 也没有完善的功能包, 客户端浏览器仍需安装专用插件才能进行三维渲染, 时至今日 X3D 在 WebGIS 中应用^[2]也并不广泛。

WebGL 的出现为 3D WebGIS 的快速发展创造了新的契机, 其具有跨平台、与浏览器无缝集成、可访问 GPU 硬件加速等特点, 为快速、高效实现 Web 三维渲染提供了一种新的技术实现途径。国内冯磊等^[3]基于 WebGL 技术开展了 3D WebGIS 实现的初步研究, 验证了 WebGL 在 3D WebGIS 中的能力与优势。国外基于 WebGL 技术的各类系统也一次次地向人们展示了 WebGL 所带来的惊奇效果^[4]。然而, 从目前公开的文献资料来看, 基于 WebGL 的矢量数据三维渲染研究较少, 使用该技术的 3D WebGIS 系统矢量数据三维可视化在渲染速度与渲染效果方面存在明显的不足, 本文利用 WebGL 的优势, 通过研究基于几何的 WebGL 矢量数据三维渲染方法, 在保持矢量数据原始信息的同

收稿日期: 2013-04-12; 修订日期: 2013-05-27

基金项目: 863 计划项目(2013AA122105)。

作者简介: 黄若思(1989—), 男, 湖北武汉人, 硕士研究生, 主要从事地理信息系统应用研究。E-mail: hrs@aoe.ac.cn。

通讯作者: 李传荣(1956—), 男, 湖南醴陵人, 研究员, 主要从事遥感卫星地面应用系统研究。E-mail: crli@aoe.ac.cn。

时,提升了网络环境下矢量数据渲染的速度与效果,为 3D WebGIS 矢量数据三维可视化提供了一种新的技术途径。

2 WebGL 简介

WebGL 是 Khronos 协会于 2011 年 2 月发布的一种新的 Web 三维绘图标准,是 OpenGL ES 2.0 的 Web 版,属于 HTML5 标准的一部分,具有跨浏览器、跨平台、可访问 GPU 加速等特点。相比以往的 VRML、X3D 等网络三维可视化技术,遵从 HTML5 标准的客户端浏览器原生集成了对 WebGL 的支持,用户无需额外安装专用浏览器插件,即可在 Web 页面中实现与桌面程序性能相当的三维渲染功能。AMD、NVIDIA、高通、Mozilla、Google、OPERA 和苹果等多家半导体和浏览器厂商都是这一标准的制定者,也是这一标准的积极倡导者,目前 Mozilla 的 Firefox 浏览器、Google 的 Chrome 浏览器、OPERA 的 OPERA 浏览器、苹果的 Safari 浏览器等均实现了对 WebGL 的支持。

传统的 Web 页面通常包含 HTML、CSS 与 JavaScript 文件,基于 WebGL 的应用程序在传统 Web 应用的基础上,还包含用来展示的三维模型数据与着色器源代码。这些三维模型数据和着色器的源代码通过 JavaScript 调用 WebGL API 的方式提交给 WebGL 渲染管线,渲染管线调用 GPU 资源实现 Web 环境下的三维可视化快速渲染。WebGL 进行三维可视化渲染管线如图 1 所示。

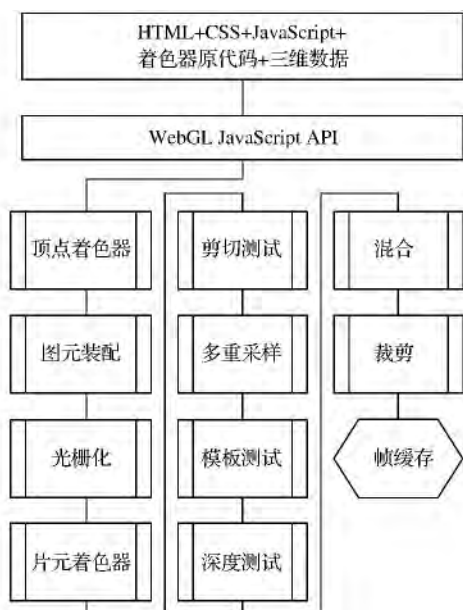


图 1 WebGL 渲染管线图

Fig. 1 Graphic pipeline of WebGL

3 基于 WebGL 的矢量数据重组织

当前主流的矢量数据三维渲染方法包括基于几何的方法与基于纹理的方法。基于几何的矢量数据渲染方法是利用二维矢量数据生成三维几何模型进行渲染;基于纹理的方法是将二维矢量数据首先渲染到图像,然后把图像作为地形纹理映射到相应地形区域实现的。基于几何的方法相比基于纹理的矢量数据渲染方法,矢量数据原始信息仍然得以保持,易于实现选择、查询、进行空间分析等操作。因此,本文使用这种方法实现网络环境下 WebGL 矢量数据三维渲染。

矢量数据通常包含几何信息和属性信息。其中几何信息由顶点坐标(点数据)或顶点坐标串(线数据、面数据)表示。通常情况下,矢量数据本身并不包含高程信息(Z 值),而几何体在三维场景中表达必须同时具备 XYZ 3 个坐标参数,因此矢量数据的三维渲染首先需要计算每个顶点的高程值 Z。常用的高程值求解算法包括控制点提取法^[5]、等高线提取法^[6]、DEM 提取法等^[7]。由于本文利用规则三角网地形作为矢量数据渲染的基础,因此采用 DEM 提取法计算矢量顶点的高程值,其方法如式(1)所示。

$$\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix} = 0 \quad (1)$$

其中: (x_1, y_1, z_1) 、 (x_2, y_2, z_2) 、 (x_3, y_3, z_3) 为矢量顶点 (x, y, z) 所在地形格网三角形的 3 个顶点坐标。

通过 DEM 提取法计算出矢量数据顶点高程值后,就可以根据各顶点原有的拓扑关系构造点、线、面的三维几何体。对于构造的点对象可以直接在三维场景中进行渲染,而对于线对象和面对象直接渲染会产生穿越地形的的问题,如图 2 所示^[8]。

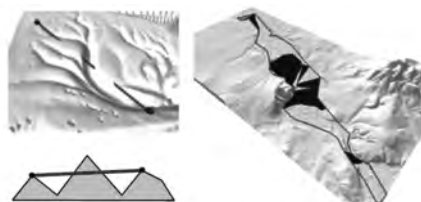


图 2 矢量线和面对象渲染问题示意图

Fig. 2 Problems of rendering lines and polygons

解决上述问题需要根据矢量线或面所在位置的地形格网数据做进一步处理。

3.1 矢量线对象的顶点加密

针对构成矢量线对象的每一条线段按照如下方

法分别进行处理:假设矢量线段两个端点分别为 P_1 和 P_2 , 计算 P_1P_2 与地形格网在二维投影面上的所有交点(图 3(a)), 然后根据式(1)计算这些交点的高程值, 最终将这些交点依次插入到 P_1P_2 之间, 形成新的线段组合(图 3(b))。

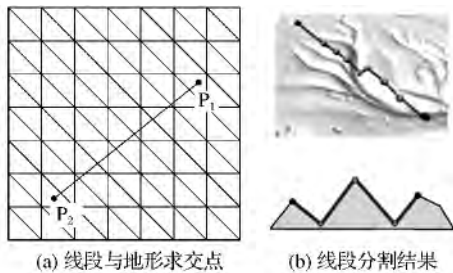


图 3 矢量线段处理示意图

Fig. 3 Processing of vector line segment

3.2 矢量面对象的离散化

矢量面对象渲染较为复杂, 如果对其边界采用矢量线对象的方法处理之后直接三角化, 其内部仍然会存在穿越地形的问題。本文设计了一种新的基于地形格网的矢量面对象三角形剖分方法, 能够形成贴合在地形表面的一系列三角面片。

首先, 将矢量面的边界以上述矢量线对象的方式进行处理, 得到紧贴地形的新边界。

然后, 根据新的边界将地形格网分为如图 4 所示的面对象内部格网(以下简称内部格网)、面对象边界格网(以下简称边界格网)和面对象外部格网 3 部分。内部网格是指包含在矢量面对象内, 但与矢量面对象边界不相交的地形网格集合; 边界网格是指与矢量面对象边界相交的地形网格集合; 剩余的网格为面对象外部格网。

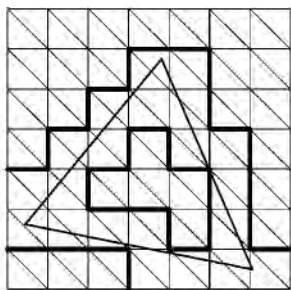


图 4 矢量面对象渲染地形格网分类示意图

Fig. 4 Categories of Terrain for vector rendering

最后, 将面对象与边界格网、内部格网的相交区域进行三角化处理, 处理结果合并即构成填充该矢量面对象的三角形集合。

面对象与内部格网相交区域三角化处理比较简单, 直接拷贝格网三角形即可。面对象与边界网格

相交区域存在如图 5 所示的 4 种情况, 需要分别进行处理。

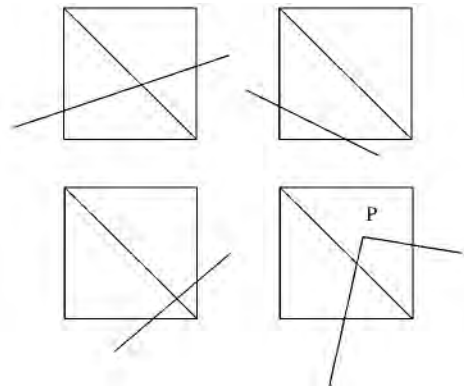


图 5 格网与边界相交情况示意图

Fig. 5 Intersections between the vector boundary and the terrain grid

这里仅对第一种相交的情况进行说明, 其他情况处理方法类似, 如图 6 所示, 矢量面边界与格网三角形的交点记为 P_1 、 P_2 、 P_3 , 若格网顶点 A 和 B 位于矢量内部, 则矢量面对象与该格网交叠区的三角化处理结果为 $\triangle AP_1P_2$ 、 $\triangle ABP_2$ 与 $\triangle BP_2P_3$; 若格网顶点为 C、D 位于矢量内部, 则三角化处理结果为 $\triangle CP_2P_3$ 、 $\triangle CDP_2$ 与 $\triangle DP_1P_2$ 。

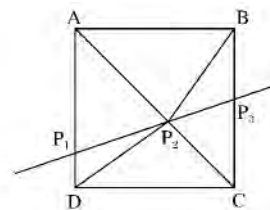


图 6 边界格网中矢量面处理示意图

Fig. 6 Process of vector polygon in boundary grids

通过上述方法计算得到的构成矢量点、线、面对象的几何体与地形完全重合, 在三维渲染时会产生深度冲突(z-fighting)问题, 出现闪烁现象。这里可以在计算矢量对象顶点高程值的时候, 添加一个深度偏移量(z-offset), 保证在渲染时矢量对象始终位于地形上方。

4 WebGL 环境下矢量数据渲染优化

大比例尺的矢量数据通常包含较多的顶点数据, 这些数据经过第 3 章中方法的处理后又会增加大量的顶点。虽然 WebGL 能够通过硬件加速进行快速数据渲染, 然而仍有必要通过优化的方法继续提高矢量数据渲染的效率与能力。本文通过两种途径进行矢量数据渲染的优化: 在预处理阶段, 使用

LOD 技术 (Levels of Detail, LOD)^[9] 结合道格拉斯—普克算法,对原始矢量进行分层简化,以减少原始数据量;在渲染阶段,针对每个 LOD 层的数据计算顶点时,基于深度偏移值动态设置方法对顶点进行处理,以减少计算量及渲染的顶点数。

4.1 基于 LOD 技术的矢量数据简化

文中矢量数据 LOD 层次简化采用了道格拉斯—普克算法,该算法由 Douglas 等^[10]于 1973 年提出,它是曲线简化最经典的算法,也是 GIS 中最常用的矢量数据简化方法^[11]。算法的基本思想是:将一条曲线首末顶点虚连一条直线,求出其余各顶点到该直线的最大距离 D_m 。将 D_m 与限差 D 相比,若 $D_m < D$,则将直线上中间点全部舍去;若 $D_m \geq D$,则保留 D_m 对应的坐标点,并以该点为界,把曲线分为两部分,对这两部分重复使用上述方法,直至最终无法作进一步的压缩为止,如图 7 所示^[12]。

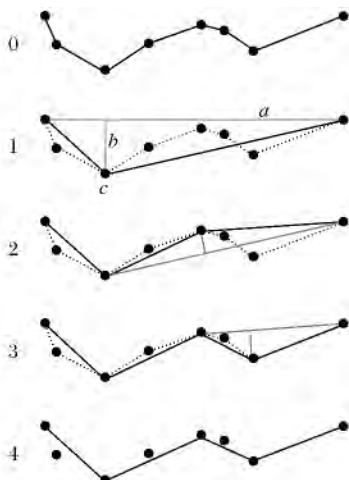


图 7 道格拉斯—普克算法示意图

Fig. 7 Schematic diagram of Douglas-Peucker algorithms

使用道格拉斯—普克算法设定不同的限差 D ,即可生成不同 LOD 细节层次的矢量数据。

4.2 基于深度偏移值动态设置的顶点计算

在 WebGL 环境下,渲染矢量数据时,根据视点与地形的距离不同,访问不同的 LOD 层次数据。每一个 LOD 层都需要根据第 3 章中的方法实现三维渲染。在第 3 章中,为了解决 z -fighting 问题,采用了对顶点设置深度偏移值的方法。如图 8 所示,对顶点设置深度偏移值后,可以在地形相对平坦的区域,对部分顶点数据进行合并,而不影响渲染效果。基于此,本文提出了基于深度偏移值动态设置的矢量数据顶点计算方法,减少了顶点计算的次数及 WebGL 渲染的数据量,提高了渲染效率。



图 8 矢量深度偏移与简化示意图

Fig. 8 Z-offset and simplification of vector data

方法的基本思想是(以矢量线数据为例):根据视点与地形的距离,设定深度偏移值 dz ,计算矢量原始线段内的地形顶点到此线段的 Z 方向最大距离 H_{\max} ,并与 dz 进行比较,若 $H_{\max} < dz$,则说明该线段顶点经过深度偏移设置后,不会产生穿越地形的现象,因此,该线段内所有地形顶点不必参与顶点插值计算。若 $H_{\max} \geq dz$,则以该点为界,把线段分为两部分,对这两部分重复使用上述方法,直至不再产生穿越地形的现象为止,如图 9 所示。视点与地形的距离越远,则设定深度偏移值 dz 越大,系统的计算量就越小,渲染效率越高。

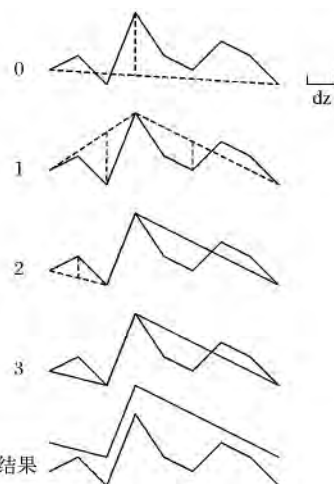


图 9 基于深度偏移的矢量顶点计算方法示意图

Fig. 9 Vector data optimization based on z-offset

对于矢量面也使用同样的方法,不同之处在于将点与线段的距离比较替代为地形格网三角面片与原始多边形平台的距离比较,达到渲染三角面片数量的目的。

5 实 验

为了验证基于 WebGL 的矢量数据渲染方法的可行性与渲染效果,本文基于上述方法开发了基于 WebGL 的矢量数据三维渲染原型系统,并进行了实验验证。系统为浏览器/服务器架构(B/S),如图 10 所示。其中,服务器端包括 Web 服务器、矢量数据发布服务器和数据库服务器,客户端为各种支持 WebGL 运行的 HTML5 浏览器访问终端。

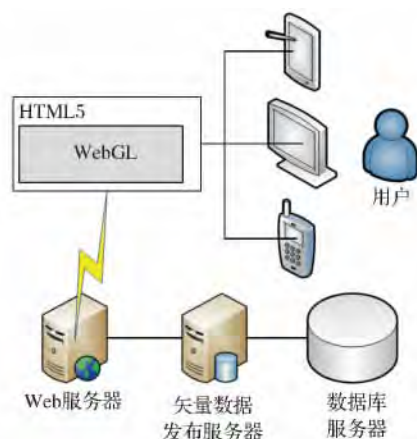


图 10 基于 WebGL 的矢量数据三维渲染系统结构图

Fig. 10 System structure figure of WebGL-based vector data 3D rendering

实验采用一台计算机,既作为服务器又作为浏览器访问终端,实验软硬件环境如表 1 所示。

表 1 实验软硬件环境配置表

Table 1 Environment configuration table

环境	参数
CPU	Intel Core 2 Duo E4500, 2 200 MHz
内存	2 GB
显卡	NVIDIA GeForce GT 220 (1024 MB)
操作系统	Windows XP
数据库	PostgreSQL 8.4
Web 服务器	Apache tomcat 6.0.35
矢量数据发布服务器	GeoServer 2.3
浏览器	chrome 22.0.1229.92

实验使用北京市昌平区沙河地区的地形数据作为背景数据,在其上进行了矢量点、线、面的渲染实验,渲染效果如图 11 所示,矢量数据渲染结果紧贴地形表面,但没有出现穿越问题,也没有出现闪烁现象。

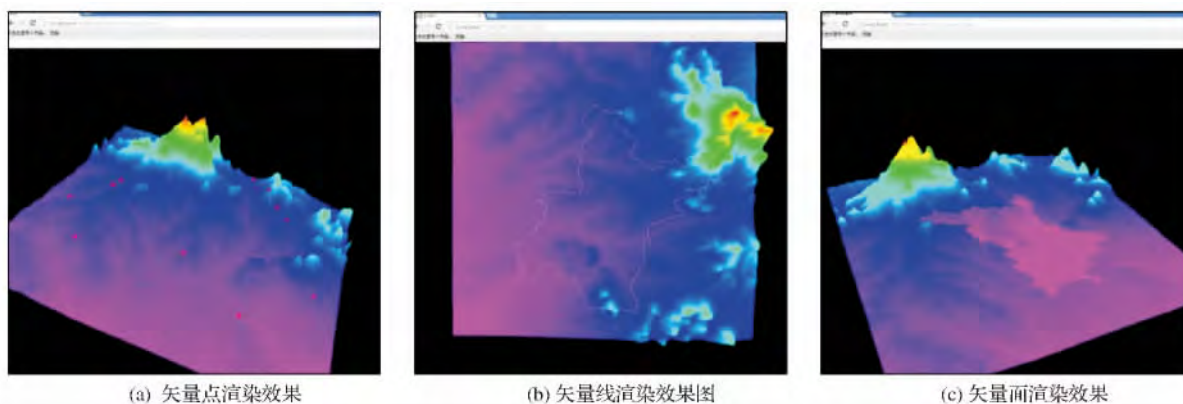


图 11 渲染效果图

Fig. 11 Rendering effect figure

同时对图 11(b)中矢量线数据进行了渲染优化实验。原始矢量渲染线段数为 2 792,每秒渲染帧数(FPS)为 30;当视点拉远,优化后渲染矢量线段数为 1 318,FPS 为 60,渲染速度提升了一倍。

6 结 语

本文针对现有 3D WebGIS 系统矢量数据三维可视化在渲染速度与渲染效果方面存在明显不足的问题,研究了基于 WebGL 的矢量数据三维渲染技术,实现了网络环境下高品质的矢量数据三维渲染,在此基础上提出了 WebGL 矢量数据渲染优化方法,提升了网络环境下矢量数据渲染的速度。该研究为网络环境下矢量数据三维可视化提供了一种快速、高品质的实现技术途径,在可预见的将来,随着遵从 HTML5 标准的网页浏览器的广泛推广,该研

究成果将会很快得到普及应用,为互联网用户提供类似桌面三维 GIS 体验效果的 3D WebGIS 应用。

参考文献(References):

- [1] Schilling A, Zipf A. Generation of VRML City Models for Focus based Tour Animations: Integration, Modeling and Presentation of Heterogeneous Geo-data sources[C]//The Eighth International Conference on 3D Web Technology, Web 3D 2003, Saint Malo, France, 2003:39-48.
- [2] Song Wei, Li Hua. Research of 3D Web GIS System based on X3D[J]. Computer Engineering and Design, 2005, 26(11): 2920-2921, 2941. [宋蔚, 李华. 基于 X3D 的三维 Web GIS 研究[J]. 计算机工程与设计, 2005, 26(11): 2920-2921, 2941.]
- [3] Feng L, Wang C L, Li C R, et al. A Research for 3D WebGIS based on WebGL[J]. Computer Science and Network, 2011.
- [4] Sloup P. WebGL Earth[D]. Masaryk University, 2011.
- [5] Zhai Wei, Chi Zhongxian, Du Jinlian. Research on Automatically

- Constructing DEM based on City Vector Map[J]. Journal of Dalian University of Technology, 2003, 43(2): 252-255. [翟巍, 迟忠先, 杜金莲. 基于二维矢量地图自动构建 DEM 方法研究[J]. 大连理工大学学报, 2003, 43(2): 252-255.]
- [6] Wang Xiaoying. 2D Vector Terrain Map Elevation Information Extraction and Real-time DEM Generation and Application in Mine area[D]. Xuzhou: China University of Mining and Technology, 2005. [王晓英. 矿区二维矢量地形图的高程信息提取、实时 DEM 生成及应用[D]. 徐州: 中国矿业大学, 2005.]
- [7] Wang Xiaoying, Wu Kan, Wang Ling. A Research on 2D Vector Map Levation Information Extraction Method[J]. Mine Surveying, 2006, (4): 11-14. [王晓英, 吴侃, 王伶. 二维矢量地图高程提取方法研究[J]. 矿山测量, 2006, (4): 11-14.]
- [8] Kersting O, Döllner J. Interactive 3D Visualization of Vector Data in GIS[C]//ACM International Symposium on Advances in Geographic Information Systems. New York, USA, 2003.
- [9] Clark, James H. Hierarchical Geometric Models for Visible Surface Algorithms[J]. Communications of the ACM, 1975, 19: 547-554.
- [10] Douglas D, Peucker T. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature[J]. The Canadian Cartographer, 1973, 10(2): 112-122.
- [11] Cai Jun. Research on Object Simplification and Scale Sampling Method in Lod Model for Vector Data[D]. Changsha: Central South University, 2009. [蔡俊. 矢量数据 LOD 模型中实体简化与尺度采样方法研究[D]. 长沙: 中南大学, 2009.]
- [12] Bonham-Carter G. Geographic Information Systems for Geoscientists: Modelling with GIS[M]. New York: Pergamon Press, 1994.

Geometry based Rendering of Vector Data with WebGL

Huang Ruosi^{1,2}, Li Chuanrong¹, Feng Lei¹, Tang Lingli¹

(1. Key Laboratory of Quantitative Remote Sensing Information Technology, Academy of Opto-Electronics, Chinese Academy of Sciences, Beijing 100094, China;

2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: As it is not well supported by existing 3D WebGIS system with respect to the effect of vector data rendering results, a geometry based approach to render vector data, which takes the advantage of the new WebGL technology, is proposed in the paper. And a optimization method for vector data rendering acceleration is also given. Then a prototype system of 3D-rendering of vector data is developed to verify the rendering effect. At last, a test is carried out to prove the feasibility and effectiveness of the method proposed in the paper.

Key words: WebGL; Vector data rendering; Level-of-details; 3D WebGIS