

一种 webGIS 数据可视化方法^①

肖寒冰¹ 方路平² (1. 浙江工业大学 计算机学院 浙江 杭州 310023; 2. 浙江工业大学 信息学院 浙江 杭州 310023)

摘要: 对 AJAX 技术和基于 JavaScript 的矢量图形可视化方法进行了讨论, 提出了基于这两种技术组合的一种 webGIS 平台设计方案, 即采用瓦片图来表示基础地图信息, 用 JVGL 来实现其他地理信息的二维可视化。

关键词: AJAX; webGIS; JVGL; 可视化方法

Visualization Method of WebGIS Data

XIAO Han-Bing¹, FANG Lu-Ping² (1. College of Computer Science and Technology Zhejiang University of Technology, Hangzhou 310023, China; 2. College of Information Engineer Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: The working principle of AJAX and the visualization method of webGIS data, and another JavaScript-based vector graphics principles of visualization methods are introduced. The combination of the two proposed a new type of webGIS platform design program, using tile-based map to indicate the map information, using JVGL to other two-dimensional visualization of geographic information

Keywords: AJAX; webGIS; JVGL; visualization method

地理信息可视化的内涵相当丰富, 其研究的内容主要是二维和三维矢量图形的可视化。在 webGIS 平台中, 矢量图形(特别是二维矢量图形)的实现方法与其平台的实现方式密切相关。

当前, webGIS 的实现方式各不相同, 从 CGI 架构到 .NET 架构, 在不同架构下的平台, 其图形可视化方式不尽相同。广泛流行的 Google Map 是基于 AJAX 的, 它的信息可视化采用的是贴图方式, 无论是地图还是用户信息, 都以 256*256 的 JPG 方式展现出来, 其地图切片更是以金字塔型瓦片图为人们知晓。而其他的一些 GIS 平台则以 VML 或者是 FLASH 以矢量图方式实现地理信息的可视化。

1 AJAX原理及webGIS数据表示

1.1 AJAX 概述

AJAX 是 Asynchronous JavaScript and XML(异

步 JavaScript 和 XML)的缩写。严格意义上说, AJAX 不是一种技术, 而是多种技术的集合, 其核心是 XMLHttpRequest, JavaScript 和 DOM。它们各有其独到之处, 合在一起, 就成为了 web2.0 中的一种革命性的技术。AJAX 可以使得页面发出请求却无须刷新整个页面, 既可以使客户端更小, 也可以让用户在使用 webGIS 时感觉如同使用普通桌面软件一样轻松快捷^[1,2]。

1.2 AJAX 的工作原理

传统的 web 应用交互流程是这样的: 用户发出一个 HTTP 请求, 然后等待服务器应答, 服务器在收到请求以后, 执行相应的操作, 生成 HTML 页面表示的结果, 然后返回给用户浏览器。这个过程不断重复, 直到用户退出这个 web 应用。在每次发出请求以后, 浏览器都要等待服务器的响应结果, 在这个时间间隔

^① 基金项目:浙江省财政厅专项(2008C0417)

收稿时间:2010-03-22;收到修改稿时间:2010-05-13

内,无法响应用户的其他操作。并且,浏览器因为刷新页面,还会出现白屏现象。

AJAX 中通过 XMLHttpRequest 来实现异步交互。在交互过程中,用户无需等待交互的结果,依然可以进行其他操作,浏览器也可以响应用户的操作。等到交互结果到来以后,浏览器可以只更新页面中特定的部分,而不用全部更新。

由于异步响应的存在,所以由于页面刷新导致的白屏现象,在很大程度上可以避免。

1.3 webGIS 上的数据表示

相对于传统 GIS,在 webGIS 实现中,我们要解决三个方面的问题^[3]: 1)对于空间数据的表示; 2)实现空间数据在服务器和客户端之间的传输; 3)实现空间数据所表达的信息在客户端的呈现。

1.3.1 空间数据的表示

现在地理信息数据库主要通过地理标记语言(Geography Markup Language, GML)来存储和管理地理信息。GML 是开放 GIS 联盟(Open GIS Consortium, OGC)的推荐标准。它是基于 XML 的语法规集,用于地理空间信息的建模、存储和传输。它完全适合在网络上传输,也便于转换为其他基于 XML 的语言。

以下就是由 GML 表示的街区的地理信息:

```
<Feature fid="142" featureType = "district"
Description = "A middle school">
  <Polygon srsName="epsg:27354">
    <LineString srsName="epsg:27354">
      <CDATA>
        491888.000000459,5458045.00063358
        491904.000000458,5458044.00063358
        491908.000000462,5458064.00063358
        491924.000000461,5458064.00063358
        491925.000000462,5458079.00063359
        491977.000000466,5458120.00063360
        491953.000000466,5458017.00063357
      </CDATA>
    </LineString>
  </Polygon>
</Feature>
```

1.3.2 服务器端和客户端的通信

客户端和服务器端的交互主要由 XMLHttpRequest 完成。在客户端有交互需求时,便

会生成一个 XMLHttpRequest,然后等待服务器端发送所需数据,得到以后,根据取得的数据更新部分内容。以下是 XMLHttpRequest 对象的概述。

首先是创建 XMLHttpRequest 对象,下面是简单的创建 XMLHttpRequest 对象的示例代码^[4]。

```
var xmlhttp;
function createXMLHttpRequest(){
  if(window.ActiveXObject)
    xmlhttp=new ActiveXObject("
Microsoft.XMLHTTP" );
  else if(window.XMLHttpRequest)
    xmlhttp=new XMLHttpRequest();
}
```

在创建了 XMLHttpRequest 对象以后,就要设置 onreadystatechange 属性,即回调函数。这个属性是用来指定当 readyState 属性改变时的事件处理句柄。之后便可以使用 send()方法来向服务器发送请求。当服务器返回信息以后,便可以通过回调函数对返回的数据进行处理。

1.3.3 信息的表示

在本文中,通过两种方式来表示空间数据信息。

首先是利用瓦片图来表示基础的地图信息。将已有的卫星影像数据分片,或者是将地理信息转为二维栅格图形以后分片。采用文献^[5]中的方式组织地图分片。

此类信息很容易由用户导入。只需要拥有符合国家标准比例的图像,然后进行分片,并且根据比例尺大小命名分片后的图片,便可以导入。

另外使用基于 JavaScript 的矢量图形库来显示少量的易变更的数据。

2 JVGL库的原理、功能和特性

2.1 JVGL 的原理

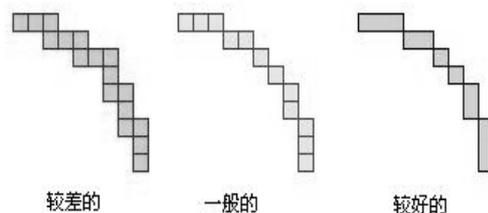


图 1 优化像素个数^[6]

JVGL 是一个遵循 GNU 的 Lesser General Public License (LGPL), 提供高性能 JavaScript 图形函数功能的自由软件^[6], 它完全由 DHTML 和 JavaScript 实现。

由于 HTML 仅仅支持矩形元素, 并不支持直线、椭圆等非矩形图形, 所以非矩形元素必须通过分割, 变为若干矩形元素来表示。由于这些矩形元素都是像素级大小, 将其组合形成图形以后, 边缘并不明显。JVGL 就是通过将若干个 HTML 中的 DIV 元素, 缩小到像素级大小, 并将其染上相同的背景色, 通过这些元素的组合在页面上绘制各种图形。

为了达到绘图性能最佳的目标, JVGL 通过最大化任一 DIV 中的能包含的像素数这种方式来减少 DIV 的个数, 如图 1 所示。

2.2 JVGL 库的函数

JVGL 提供了三类函数。首先是设置函数, 比如设置所画图形的宽度, 颜色等。其次是绘图函数。还有就是辅助函数。这三类函数封装了用于 HTML 绘图的直线、折线、椭圆等功能, 并且可以设置颜色、线条类型宽度等等。表 1 展示了 JVGL 的几种主要函数、参数说明以及示例。

表 1 JVGL 的绘图函数

函数名	函数参数说明	实例
setColor("#HexColor")	用以设定画笔的颜色。可以使用 16 进制的 RGB 色, 也可以使用 HTML 能识别出的颜色名。	jg.setColor("#ff0000"); jg.setColor("red");
setStroke(Number)	设置画笔的宽度, 以像素为单位。或是设置成为虚线画笔。当设置画虚线时, 宽度为 1 个像素。	jg.setStroke(3); jg.setStroke(Stroke.DOTTED);
drawLine(X1, Y1, X2, Y2)	画(X1,Y1)到(X2,Y2)的线段	jg.drawLine(20,50,453,40);
drawPolyline(Xpoints, Ypoints)	画出点列所表示的折线。Xpoints, Ypoints 分别储存点列的 XY 坐标。 var Xpoints = new Array(x1,x2,x3,x4,x5); var YPoints = new Array(y1,y2,y3,y4,y5);	var Xpoints = new Array(10,85,93,60); var YPoints = new Array(50,10,105,87); jg.drawPolyline(Xpoints,Ypoints);
drawRect(X, Y, width, height)	画左上角坐标是(X,Y), 宽度是 width, 高度是 height 的空心矩形。边缘宽度可以由 setStroke() 设置, 默认为 1。	jg.drawRect(20,50,70,140);
fillRect(X, Y, width, height)	画左上角坐标是(X,Y), 宽度是 width, 高度是 height 的实心矩形。	jg.fillRect(20,50,453,40);
drawEllipse(X, Y, width, height)	画以左上角是(X,Y), 宽 width, 高 height 的矩形为外切矩形的空心椭圆, 边缘宽度可以由 setStroke() 设置, 默认为 1。	jg.drawEllipse(20,50,70,140);
fillEllipse(X, Y, width, height)	画以左上角是(X,Y), 宽 width, 高 height 的矩形为外切矩形的实心椭圆。	jg.drawEllipse(20,50,70,140);
fillArc(X, Y, width, height, start-angle, end-angle)	画出一个扇形, start-angle 是起始角度, end-angle 是结束角度。其他参数同上。	jg.fillArc(20,20,41,12,270.0,220.0);

但是并不能用 JVGL 直接进行地理信息的可视化, 必须经过转化。绝大多数的地理信息是通过 GML 格式

储存。GML 是基于 XML 的语言类型。GML 通过标签来表现其几何特征, 通过这一点可以建立起一套由

GML 标签到 JVGL 图形函数的映射关系,如表 2 所示。

表 2 GML 和 JVGL 的映射关系

GML 的几何特征	JVGL 接口函数
点	drawLine
线段	drawLine
折线	drawPolyline
曲线	采用 drawPolyline 近似
多边形	drawPolygon
面	drawPolygon

2.3 对 JVGL 的改进

由于需要尽可能减少 DIV 的个数,而不能每个像素就是一个 DIV 元素,所以简单易行的扫描线算法并不合适。JVGL 的作图算法是以 Bresenham 快速算法为基础进行改良的。在此根据所需功能,对于某些算法细节进行改变。

由于使用 JVGL 的主要目的是对于部分地理信息进行二维可视化处理,所以需要添加部分功能,以适应地理信息的显示。

在已有的 JVGL 中,所有的图形都是以 DIV 元素形式表示出来的。而这些元素的共同特征是不透明。而在二维可视化处理中,很多情况下,需要半透明的元素。这样可以表现出叠加等效果。以下是建立半透明 DIV 元素的 JS 函数代码。其中四个参数 x,y,w,h 表示的是所要建立的 DIV 元素的位置和宽高。

```
function _mkDivTrslct(x, y, w, h)
{
    this.htm += '<div style='
"position:absolute;+
    'left:' + x + 'px;'+
    'top:' + y + 'px;'+
    'width:' + w + 'px;'+
    'height:' + h + 'px;'+
    'clip:rect(0,'+w+'px,'+h+'px,0);'+
    'filter:alpha(opacity=30);opacity:
0.7;'+
    'background-color:' + this.color +
    (!jg_moz? 'overflow:hidden' : '')+
    '><\div>';
}
```

下图是两种不同效果的对比图。

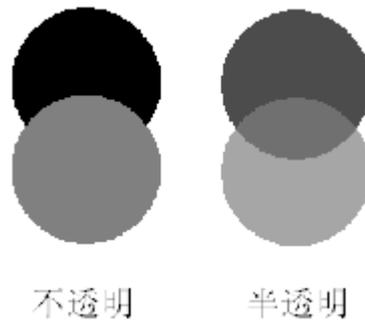


图 2 不同透明效果的对比

3 新型轻量级webGIS平台设计

现有的 webGIS 模式,或者是类似 Google map 方式,在服务器端将所有的地理信息转换为栅格图形,然后以瓦片图的方式展现出来;或者是将所有的地理信息转换为矢量图形,以 VML 或者是 SVG 等方式展现。这两种处理方法各有优缺,瓦片图在比例变换以后,需要把地图全部刷新。而矢量图一旦载入,便可以自由放大缩小。但矢量图只适合于胖客户端。

3.1 瓦片图和矢量图的归一化处理

一般 webGIS 平台分为表现层、逻辑层和数据层三层。在三层结构中,浏览器端承担着表现的任务。浏览器是地理信息可视化的载体,是人机交互的接口。它由以地图为核心的各种地理信息元素所组成。在不同的平台上,地理信息的可视化方式不尽相同,但是对于浏览器而言,都是一个个 HTML 占位元素。

JVGL 所表示的位置要于地图位置相同,就是要两者所表示的经纬度相同。通过以下方式进行归一化处理。

$$pixelX = \frac{longitude}{360} \times tileSize \times 2^{zoom} + tileSize \times 2^{zoom-1}$$

$$pixelY = tileSize \times 2^{zoom-1} - \frac{longitude}{360} \times tileSize \times 2^{zoom}$$

其中 pixelX 和 pixelY 表示的是屏幕上的坐标,而 longitude 和 latitude 表示的是经纬度。tileSizeX 和 tileSizeY 表示瓦片图的大小,一般来说 tileSizeX=tileSizeY=256。Zoom 表示瓦片图的的级数。

3.2 地图图层和 JVGL 的叠加

在地图与 JVGL 进行归一化以后,便可以在浏览器中进行显示了。为了使用户数据不至于干扰地图信

息, JVGL 一般使用的是上文提到的改进后的半透明方式。在特殊情况下, 也可以使用不透明方式。

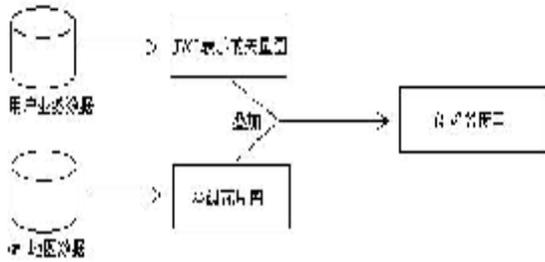


图3 图层的叠加

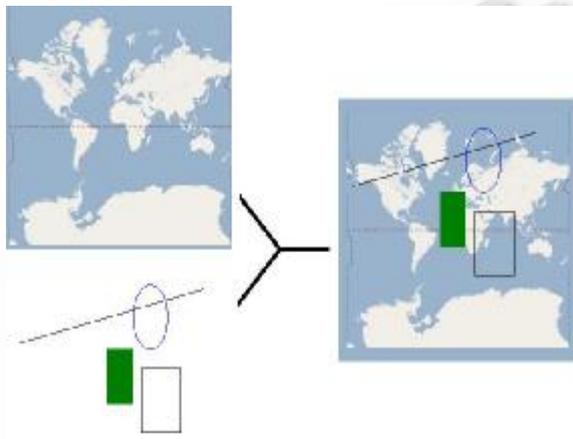


图4 图层的叠加的实际效果展示

4 结束语

本文提出的基于瓦片图+JVGL 的 webGIS 平台设计方法相对于 google map, 省去了在服务器端将用户数据转为栅格图并且传输的资源。将用户数据一次性载入, 无论比例尺放大缩小, 浏览器都可以根据比例绘制图形。但由于 JVGL 是基于 DIV 绘图, 由于 HTML 限制, 当所需绘制图形较多时, 会导致浏览器响应速度降低。今后的研究内容将加强对算法的改进, 对于 DIV 元素的重叠进行识别, 然后将不同时间段生成的 DIV 元素合并, 降低浏览器响应变慢的可能。

参考文献

- 1 Crane D, Pascarello E, James D. Ajax in Action. USA: Manning Publications, 2005:1 - 306.
- 2 Asleson R, Schutta NT. Foundations of Ajax. USA: A press, 2005:106 - 405.
- 3 沈静, 吴健平, 戎恺. 基于 WebService 的 WebGIS 的设计与应用. 遥感技术与应用, 2004:138 - 142.
- 4 柯自聪. Ajax 开发精要——概念、案例与框架. 北京: 电子工业出版社, 2006.
- 5 巫细波, 胡伟平. Google Maps 运行机制以及应用研究. 华南师范大学学报(自然科学版), 2009:107 - 110.
- 6 Zorn W. High Performance JavaScript Vector Graphics Library. [2010-03-28] http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm, 2009.