

Web 图形技术在煤矿监测中的应用

周方晓^{1,2}, 李昌华¹, 丁有军¹

(1 西安建筑科技大学 信息与控制工程学院, 陕西 西安 710055; 2 西安建筑科技大学 土木工程学院, 陕西 西安 710055)

摘 要: 为了在监测中使用 Web 图形, 需解决网页局部刷新、图形生成和传送问题. 用 AJAX 技术实现网页局部刷新, 用 GDI+ 技术生成曲线图并转化成图片返回给用户, 同时还给出了提高效率的措施. 使用结果表明: AJAX 技术提高了显示效果, 生成的曲线图准确并且可以在允许时间内得到查询响应.

关键词: Web 图形; AJAX; 图形设备接口; 煤矿监测

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2011)04-0162-05

Application of Web Graphics Technology to Coalmine Monitoring

ZHOU Fang-xiao^{1,2}, LI Chang-hua¹, DING You-jun¹

(1 School of Information and Control Engineering,

Xi'an University of Architecture and Technology, Xi'an 710055, China;

2 School of Civil Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: In order to apply web graphics to monitoring, it needs to solve problems of local updating web pages, creating graph and transferring it. We use AJAX to implement local updating, and GDI+ to create graph and then convert into picture for users. Methods for promoting efficiency are given. The result shows that AJAX improves effect on display, indicates that graph built is accurate and can receive reply within the time allowed.

Key words: web graphics; AJAX; GDI+; coalmine monitoring

1 引言

所谓 Web 图形, 是指用户使用浏览器接收来自 Web 服务器的响应数据, 并在网页中以图形方式显示数据. 与其它图形方式相比, Web 方式具有不需安装客户端程序的优点, 只要有浏览器及必要插件就可以显示图形, 因此使用和维护都很方便. 另外由于 Web 所具有的分布式特点, 为用户跨地域使用提供可能. Web 图形这些优点为煤矿监测人员带来了很大的便利.

使用 Web 图形为浏览器端的用户带来了方便, 但对服务器端却提出了更高的要求. 服务器端需要解决实时数据的动态刷新、图形的生成及传送等. 为了显示实时数据, 需动态刷新网页. 刷新一次就需服务器传送一次网页的全部数据. 对于实时监控网页,

它所需更新的只占整个网页的很小一部分. 因此整页更新不仅浪费带宽而且影响显示效果. 本系统采用了 AJAX 技术, 使得网页局部刷新成为可能.

生成 Web 图形的技术有: GDI+、SVG 及 Adobe Flex2 等. GDI+ 的图形绘制功能更加强大, 在浏览器端也不需要另装插件. 因此本系统采用 GDI+ 生成图形.

2 系统简介

2.1 系统架构

采用 B/S(浏览器端/服务器端)架构. 在服务器端有两类服务器: 数据库服务器、Web 服务器. 前者负责存储现场采集的数据, 后者负责处理 Web 事务. Web 服务器接受来自浏览器的 Web 请求, 并从数据库服务器查询数据, 经处理后返回 Web 响应.

收稿日期: 2010-06-08; 修回日期: 2010-08-11

基金项目: 国家自然科学基金项目(50878176)

浏览器端则用浏览器打开返回的 Web 响应,并以图形的形式显示给用户.见图 1 所示.

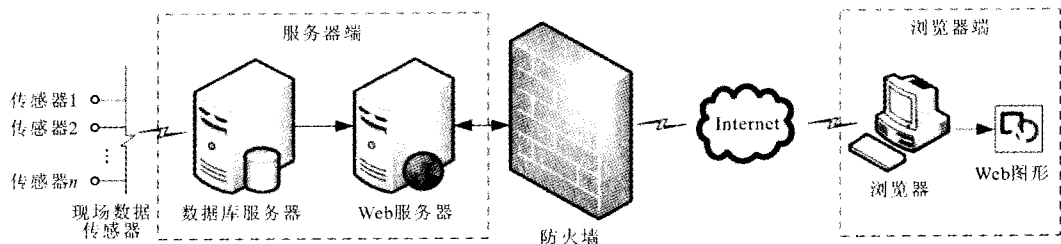


图1 系统架构图

2.2 系统功能

系统与 Web 图形有关的主要功能如下:

- (1) 实时读取和显示井下传感器传来的动态信息;
- (2) 超限报警功能:当情况异常超过报警线时,系统可进行屏幕提示或声音报警;
- (3) 绘图功能:系统可以绘制指定时间范围内的历史数据图,用曲线图形式显示.

2.3 系统环境

- 服务器:.. Net2. 0 Framework + AJAX Extensions + IIS;
- 浏览器:IE6. 0 及以上;
- 开发工具: Asp. Net(C#).

3 Web 图形的实现

3.1 实时数据的动态显示

3.1.1 AJAX 技术

AJAX 全称为“Asynchronous JavaScript and XML”(异步 JavaScript 和 XML),是一种创建交互式网页应用的网页开发技术. AJAX 不是指一种单一的技术,而是有机地利用了一系列相关的技术^[1]. 它能在不更新整个页面的前提下维护数据. 这使得 Web 应用程序更为迅速地回应用户动作,并避免了在网络上发送那些未改变的信息. 图 2 对比了传统 Web 模型与使用 AJAX 的 Web 模型^[1]. 可以看出,二者最大区别在浏览器端, AJAX 模型多了一个 AJAX 引擎. 有了这个引擎使得许多原本需要服务器端处理的事件在浏览器端就可以处理. AJAX 引擎发给 Web 服务器的请求也与传统模型不同:传统模型按整个网页发送请求,而 AJAX 仅针对网页某个感兴趣的局部^[2]. 虽然这种针对局部的方式需要的请求次数比传统模型多,但每次请求的数据量小,总体看它是节约带宽的. AJAX 引擎收到服务器传回的数据后,仅对网页中发生改变的部分作修改.

因此可以看出使用 AJAX 的优点:①占用较小

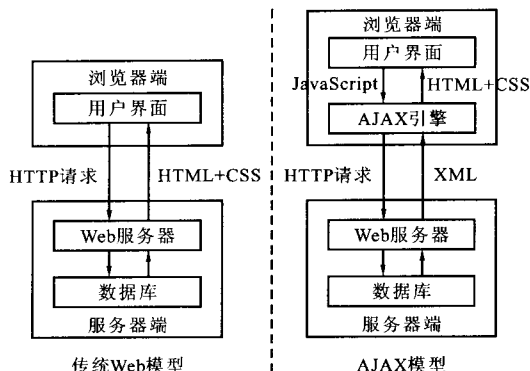


图2 传统 Web 模型与 AJAX 模型对比

的带宽;②响应较快;③显示时,不必刷新整个页面. 避免因刷新过快而出现页面闪烁现象.

本系统采用 AJAX 技术的重要原因是:它提供了网页局部刷新的功能. 解决了网页定时刷新出现的闪烁现象,大大提高了显示效果.

3.1.2 用 AJAX 实现网页局部更新

ASP. NET 提供了对 AJAX 技术的支持:在其 AJAX Extensions 包中包含了 Timer、ScriptManager 和 UpdatePanel 等 Web 服务器控件,利用这些控件实现基于 AJAX 的局部刷新网页的代码如下:

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
<ContentTemplate>
<asp:Timer ID="T1" runat="server" OnTick="OnTimer" Interval="5">
</asp:Timer>
<asp:Label ID="Label1" runat="server" Height="30px" Width="40px" Style="left: 62px; position: absolute; top: 82px; z-index: 100;" BackColor="#FFFFC0" BorderStyle="Ridge" Font-Size="X-Small" ForeColor="White"></asp:Label>
</ContentTemplate>
</asp:UpdatePanel>
```

代码由两大部分组成: ScriptManager 控件和 UpdatePanel 控件. 首先, ScriptManager 控件是每个 AJAX 网页必须包含的. 实际上, 它是 AJAX 网页的核心, 负责链接网页和 Script 脚本文件同时定位网页中局部更新的位置. 其次, UpdatePanel 控件是一个容器, 我们把需要局部更新的内容放在其中. 在 UpdatePanel 控件中包含了 Timer 控件, 此控件设置刷新时间间隔, 并映射定时响应函数 OnTimer. 接下来是 Label 控件, 它负责显示动态数据. 整个处理过程是: 由 Timer 控件在每个时间间隔到来时, 激发 OnTimer 函数. 在这个函数中, 它向服务器发送读取实时数据的请求, 服务器按此请求查询数据库并将查询结果返回给浏览器. OnTimer 将结果写入 Label 控件, 刷新 Label 控件, 从而更新显示数据.

综上, AJAX 的 ScriptManager 控件会定位网页中需局部刷新的位置, 即 Label 控件在网页中所占区域. 每次定时更新数据时, 仅刷新这一小部分区域, 从而获得局部刷新效果.

3.2 超限报警

超过设定的报警值, 则需提示用户数据超限. 系统从视觉和听觉两方面提示用户. 由于实时数据显示在 Label 控件内, 改变此控件的颜色即可提醒用户. 每次刷新时都会判断实时数据是否大于报警值, 如大于则改变 Label 控件的颜色, 使之由蓝转红. 除此外, 超限后还可以调用 System. Media. SystemSounds 类中的 Beep. Play() 函数调用蜂鸣告知用户. 对于历史数据曲线, 为区分超限区域与正常区域, 可先用水平线画出报警值, 位于报警值之上的曲线用红色画笔绘制, 否则用绿色画笔.

3.3 历史数据曲线的绘制

3.3.1 GDI+ 绘制图形

GDI+ 作为 .NET Framework 的组成部分, 它是微软为用户提供的与设备无关的图形开发程序库. GDI+ 的 System. Drawing 命名空间中有一个 Graphics 类, 此类提供了绘制弧、直线、多边形、文本等十余种基本图形的方法^[3]. 另外, 此命名空间中还有一个 Image 类, 它提供了对图像操作的方法. 本系统利用这两个类来生成历史数据曲线.

3.3.2 曲线数据的处理流程

浏览器端的用户在网页中输入需要查询的数据的类型和起止日期, 然后发送 HTTP 请求. 服务器接到请求后, 以数据类型和起止日期作为关键字查询数据库. 从数据库得到数据集中的每条记录对应

历史曲线的一个转点, 经坐标变换将记录转变为转点坐标. 使用 Graphics 类的绘图函数以及转点坐标绘制出曲线图. 由于曲线图是矢量图, 而一般网页无法直接读取矢量图, 因此我们采用 GDI+ 中 Image 类, 将矢量图转换成图片, 然后再传送给用户. 整个处理流程如图 3 所示.

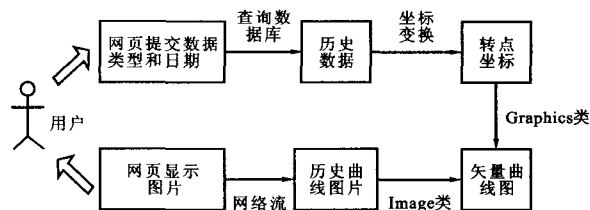


图3 数据处理流程图

图3中, 数据处理可分成两步: ①历史数据转变成屏幕中曲线的转点坐标, 这需要坐标变换计算; ②将转点连接生成曲线, 同时通过 Web 传送给用户.

3.3.3 坐标变换

坐标变换分两步实现: 计算世界坐标, 转换为屏幕坐标.

首先建立坐标系. 根据计算机图形学中光栅化概念, 矢量图形最终要映射到屏幕的一个区域内, 此区域称之为画布, 见图4. 在画布内建立两个坐标系: ①以画布左上角点 O 为原点, 向右为 X 轴正方向, 向下为 Y 轴正方向建立屏幕坐标系 XOY . 屏幕坐标系为 GDI+ 默认坐标系. ②画布内一点 O' 为原点, 向右为 X' 轴正方向, 向上为 Y' 轴正方向建立世界坐标系 $X'O'Y'$. 世界坐标系内的虚线矩形框, 为绘图有效区, 生成的图形完全包含在其内, 见图4. 之所以采用两个坐标系, 前者便于图形的绘制, 后者便于计算和显示.

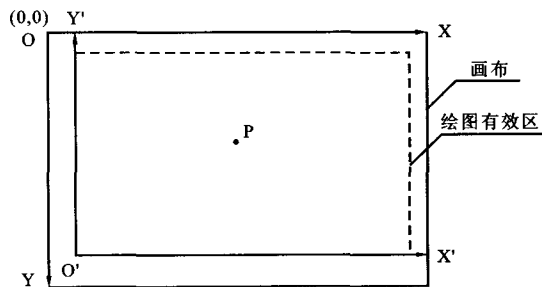


图4 坐标变换

数据集的每条记录由数据值和日期时间两个字组成. 数据值和日期时间是有范围的, 根据实际情况, 事先确定数据值最大最小值 ($V_{\min}$ 和 $V_{\max}$). 而时间范围则通过用户输入的起始时间 (T_{start} 和 T_{end}) 得到. 根据线性关系, 可以对数据值 V 和时间 T 进行单位化. 然后计算坐标值, 假设 $T_{\text{start}} - T_{\text{end}}$ 在横轴代

表的实际长度为 LT 个单位, $V_{\max} - V_{\min}$ 在纵轴代表的实际长度为 LV 个单位, 则任一点 P 在世界坐标系下的坐标 (x', y') 为:

$$\begin{aligned} x' &= \frac{T - T_{\text{start}}}{T_{\text{end}} - T_{\text{start}}} \times LT \\ y' &= \frac{V - V_{\min}}{V_{\max} - V_{\min}} \times LV \end{aligned} \quad (1)$$

其中日期时间不是十进制, 需要转换为秒为单位再行计算. 而被乘数即为单位比例.

画布对应屏幕设备, 尺寸单位为像素. 在屏幕坐标系下, 设 O 为 $(0, 0)$, O' 为 (x_0, y_0) . 绘图有效区长 LT 个像素, 宽 LV 个像素. 则任一点 P 在屏幕坐标系下的坐标 (x, y) 为:

$$\begin{aligned} x &= x' + x_0 \\ y &= y_0 - y' \end{aligned} \quad (2)$$

使用式(1)、(2), 就可以将一条记录转换为屏幕坐标系下的一个转点坐标, 供 GDI+ 的函数在绘图有效区内绘制曲线.

3.3.4 曲线图的生成和传送

得到转点的屏幕坐标后, 用 Graphics 类中的 DrawLine() 函数依次连接转点得到曲线. 除曲线外, 还需绘制坐标轴、刻度、题目、文字标示等, 都可以用 Graphics 类中的函数绘制. 最终, 形成完整的曲线图. 接着调用 Image 类, 将曲线图保存为网络流, 传回用户. 核心代码如下:

```
Image i; //图像
Graphics g = Graphics.FromImage(i); //图形
DrawTitle(g); //绘制标题
DrawCoordSys(g, startDate, endDate); //绘制坐标轴
系统
foreach(转点 pt) //绘制曲线
{
    g.DrawLine(Pen, ppt.x, ppt.y, pt.x, pt.y);
    ppt = pt; //ppt 为 pt 前一个点
}
i.Save(webStream, encoder)
```

代码中有两个对象: 图像 Image 类对象 i 和图形 Graphics 类对象 g , 二者通过 FromImage 函数关联起来. 通过 g 绘制标题、坐标轴和曲线, 生成矢量曲线图. 最后一行代码为调用 Image 类的 Save 函数, 使用指定的编码方式, 将 g 保存到 i 的 webStream 流中. 此流通过网络传送到浏览器用户, 浏览器对流解码并转换为图像显示在浏览器中.

3.4 提高效率的措施

(1) AJAX 技术使用 AJAX 引擎, 减少了网络

传送的数据量, 缩短了处理时间;

(2) 等差间隔取点. 每条记录标准间隔时间是 20s, 如果查询的起止日期相差很多, 那么将会有很多条记录需要处理. 事实上, 无论起止日期相差多少, 曲线都在相同大小的画布中显示. 而在有限时间轴内绘制太密集的数据点会造成点的堆积. 因此读取所有数据记录是不必要的. 采用等差间隔的方法读取数据记录: 比如查询一天的历史曲线按 20s 间隔读取数据记录, 查询两天则按 40s 间隔读取记录, 三天为 60s, 依此类推. 等差间隔取点保证了总的数量基本不变. 另外, 为了减少失真, 还需添加一些特征点, 如每天的最大最小值点.

(3) 设置专用的临时文件夹, 避免相同查询的重复处理. 第一次查询结果得到后, 所生成的图片文件以起止日期为文件名存放至服务器专门的临时文件夹中. 以后查询时首先按起止日期查找临时文件夹中有无此文件, 如有则直接传送给浏览器. 省去了再次进行数据库查询和图形生成的时间.

4 实例分析

本系统已运用到某煤矿的粉尘监测系统中.

用户首先须通过登录页输入用户名和密码, 然后进入图 5 所示的实时数据动态显示网页. 此网页每五秒更新一次. 从实际运行情况看, 刷新平稳且没有闪烁现象. 如有超限数据, 会用红色 Label 来显示它, 否则为蓝色.

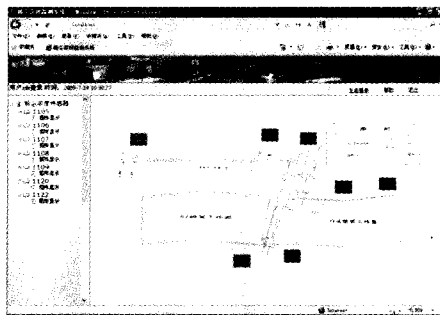


图5 实时动态数据显示页

点击显示页的传感器导航栏, 将弹出超链接的相应传感器的历史数据曲线图, 见图 6. 此图显示的是 1106 号传感器某两天时间内历史数据曲线. 画布中 X 轴为时间轴, Y 轴为粉尘浓度值. 水平线为报警线, 位于报警线之上的曲线用红色笔绘制, 之下用绿色. 画布图像为 GIF 格式, 尺寸为 750×450 像素, 图像文件大小为 20.4KB. 曲线图的查询响应时间由请求传送时间、数据库查询时间、曲线图生成时间及曲线图片的回送时间组成, 影响因素较多, 因此

文中未给出具体的实测值. 但从用户使用效果看, 基本在用户容忍范围内($<5s$)得到响应. 而且还可以看到: 第一次查询的响应时间会慢些, 但以后同条件下的查询时间大大加快, 原因就是采用了专用临时文件夹措施.

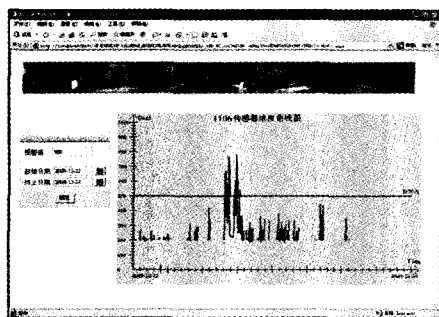


图 6 历史曲线图

5 结束语

从系统实际使用情况看, 文中的方案是有效可行的. 在实时数据动态显示网页中, 矿区平面图也是动态的, 这个问题在文献[4]中有所提及, 但它采用的是基于 DWG 文件的方法. 如何绘制 Web 形式的动态平面位置图, 是下一步需要做的工作.

由于 Web 方便灵活的特点, 它为全方位、跨地域监测提供了可能. 鉴于当今网络迅猛发展的趋势, 基于 Web 的监测方式将成为煤矿监测中的重要手段. 文中为在煤矿监测中使用 Web 图形提供了可行技术方案, 对促进 Web 监测的发展有一定积极意义.

参考文献:

- [1] Jesse James Garrett. Ajax: a new approach to web applications[EB/OL]. (2001-05-18)[2010-06-08]. <http://adaptivepath.com/ideas/essays/archives/000385.php>.
- [2] 陈婷, 周家纪, 杨宁. 基于 VRML-AJAX 的分布式虚拟现实应用[J]. 微电子学与计算机, 2006 (1): 272-273.
- [3] Mahesh Chard. GDI+图形程序设计[M]. 北京: 电子工业出版社, 2005.
- [4] 潘志刚, 卢建军, 王晓路. 基于 DWG 的煤矿井下定位监测矢量图形系统设计[J]. 煤炭科学技术, 2009 (10): 79-81.

作者简介:

周方晓 男, (1971-), 博士研究生. 研究方向为计算机图形学、软件工程.

(上接第 161 页)

- [3] Wren C R, Azarbaye J A, Darrell T P. Real-time tracking of the human body[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19 (7): 780-785.
- [4] Stauffer C, Grimson W E L. Adaptive background mixture models for real-time tracking[C]// PPComputer Vision and Pattern Recognition. CO, USA: IEEE, 1999: 246-250.
- [5] Zoran Zivkovic, Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction[J]. Pattern Recognition Let-

ters, 2006, 27(5): 827-832.

- [6] Stauffer C, Grimson W E L. Learning patterns of activity using real-time tracking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22 (8): 747-757.

作者简介:

蒋明 男, (1958-), 硕士, 副教授. 研究方向为现代数据库与图像处理.

潘姣丽 女, (1986-), 研究方向为现代数据库和图像处理.