

矢量图形数据压缩技术的研究与应用

魏 国

(中海油能源发展装备技术有限公司, 天津 300000)

摘 要: 随着互联网时代的高速发展, 矢量图形相关各类应用软件也趋向网络化。由于受网络传输速度、浏览器速度等诸多因素的影响, 为提高用户体验、减缓加载延迟, 对矢量图形进行压缩传输变得尤为重要。文章通过对矢量图形数据的研究, 提出一种新的矢量图形数据压缩技术、针对该技术研究新的寻址技术和分页存储方法, 结合 DWG 在线看图类软件与基于矢量图形的业务系统在实际应用中的效果可知, 该压缩技术克服相关瓶颈, 压缩效率得到很大提高。

关键词: 图形压缩; 数据压缩; 网络延迟

中图分类号: TP391.41 **文献标识码:** A

Research and application of vector graphics data compression technology

WEI Guo

(CNOOC Energy Technology & Services Co., Ltd., Tianjin 300000, China)

Abstract: With the rapid development of Internet era, all kinds of application software related to vector graphics also tend to network. Due to the influence of many factors, such as network transmission speed, browser speed, vector graphics improve the user experience and slow down the loading delay, it is very important to compress and transmit. By studying vector graphic data, a new vector graphic data compression technique is proposed in this paper. New addressing technology and page storage method are researched for this technique. Utilizing DWG online viewing software and vector graphics-based business system, its practical application effect shows that the compression technology overcomes the relevant bottleneck and the compression efficiency is greatly improved.

Key words: picture compression; compression of data; network delay

0 引言

矢量图形的应用范围覆盖我国机械、建筑、交通运输等多个行业, 在工程设计及实践过程中^[1], 工程人员常采用专业制图软件绘制或查看总体、工艺、详图等矢量图纸, 因为桌面应用软件处理文件速度快, 多数软件开发时对图形数据文件压缩需求并不高, 甚至可以直接用明文保存矢量数据, 但随着互联网技术的不断发展, 例如项目管理、资产管理、图纸管理甚至图形设计等业务模块趋向网络化, 很多桌面应用软件都实现了在线使用, 由于此类在线应用软件受网络传输速度、浏览器速度的影响, 将矢量图形进行压缩传输变得越来越重要^[2], 也成为了图形类在线应用软件的设计瓶颈。

目前所处的互联网时代, 矢量图形压缩技术要解决好两方面的问题, 其一是压缩率要高, 其二是解压速度要快, 设计一种压缩方案需要兼顾这两者之间的平衡。当今软件产品多数以浏览器为平台加载, 实现了跨平台、无插件、云端使用, 目前包括 IE、Chrome、火狐等主流浏览器在最新 HTML5^[3] 技术下仍未实现与桌面程序同级别的多线程性能, 其掌控 CPU、GPU、内存的能力也大不如桌面程序, 因此采用特别复杂的压缩算法压缩后的数据在浏览器端即需要花费更多资源进行解码, 这是会造成软件延迟,

收稿日期: 2018-09-28

作者简介: 魏国(1983-), 男, 本科, 工程师, 研究方向为计算机信息技术、大数据分析。

降低用户体验感,而数据压缩率低则下载时间长,同样影响软件使用,加之浏览器前端代码与远程服务器后台交互过程中实时大量并发互传图形数据^[4],延迟情况会更加严重,因此权衡压缩率与解码速度两个因素合理设计图形数据压缩方案变得尤为重要。

1 压缩技术

1.1 矢量图形数据压缩简介

以通用图形格式 DWG^[5]来举例,明文 LINE(3.854 2.56 0.15683)(8.5 0, 0.15683)释义为从一个空间点至另一个空间点做一条空间直线,括号内分别是两个点的三维坐标,如果直接传输这段文字,则需要用 40 个 Byte(字节),1Byte = 8bit(比特),则一共用去 320bits,用代号 13 取代 LINE 则节省了 2Byte,解码时读到类型为 13 即 LINE,再者取 0 来看它单独占用了 1byte 即 8bit,但如果仅存储 2Bit(00),如果读一个 double(双浮点)类型时先读 2Bit,如果是 00 则值就位 0,这样就省去了 6Bit,再者第二个空间点仅记录与第一个点的差值,则 z 坐标 0.15683(6Byte = 48Bit)相同即为 0,这样就节省了 46Bit。

对比传统的通用压缩技术^[6],即将所有字节变为 bit 后应用算法寻找 0 和 1 组成的串码重复规律,以上此类矢量图形数据压缩专用技术^[7]比传统的通用压缩技术有以下三点优势。其一,传统方式归根结底还是一种数据或公式字典,字典内容、判别公式、算法越多则压缩率越高,其解码时也就越占 CPU、速度变慢,最后解出来的字节串还需分隔读取后使用,而专用方式采用可顺序读取的工作方式,不但并保持了数据的整体性,而且按顺序提取数据方便;其二,网络传输兼容性方面,传统方式需要将所有数据均收取完才能解压,若网络传输错误则需要全部重传数据,而专用方式可分段读取解压,边读边解压的优势是网页端尤其重要的,大大降低了用户的操作延迟度;其三,在实时互动的图形操作系统中,用户大量传输单条图形数据,传统压缩技术因为数据总量小其压缩率并不高,而专用技术仍保持不变的压缩率。

1.2 数据压缩技术

针对各种类型的数据需要制定不同的压缩方案,矢量图形中常见的数据^[8]一般可以归纳为 Short(无符号整形)、Long(长整型)、float(单精度浮点)、double(双精度浮点)、Char(字符型),再加之矢量图形的一些特点,例如水平线、竖直线这类一个坐标轴相同差值为 0,即 0 比较多;多数值得小数位并不

多,即 float 类型比 double 类型多等等,参照通用 DWG 格式的压缩方式,对各种类型的压缩解决方案如下。

1.2.1 符号位压缩法

举例一个被压缩的整数,先读两个 Bit 作为标志位,标志位是 00 则读后面 2 个 Byte 组成一个整数值,可表示最大整数为 $2^8 - 1$;若标志位是 01 则读后面 1 个 Byte,可表示最大整数 $2^8 - 1$;若标志位是 10 则表示值 0,若标志位是 11 则表示值 256,这种编码采用了适当的字节数,通常用于表示图元类型号、256 种颜色,其 0 和 256 两种颜色通常代表最常使用的白色和黑色,可直接从标志位读出值兼顾了软件的使用特点。

压缩浮点数也采用相似方法设置 2Bit 的标志位,当标志位 00 时读取 8byte 浮点数,当标志位 01 时值为 1,标志位 10 时值为 0,其压缩设计兼顾到了图像三轴坐标差、图像平移、拷贝经常出现 0 的情况,也兼顾到缩放比例经常为 1 的特点。

1.2.2 模块化压缩法

这种方法多用于存储较大整数,例如表达在 1G 文件中的地址,从 0 至 10243,可以将每个 Byte 视为一个模块,当第一个模块首位 Bit 为 1 则有第 2 个模块,当第 2 个模块首位 Bit 为 1 则有第 3 个模块,直到某模块首 Bit 为 0 则结束,最后将去除了标志位的 bit 合在一起构成一个整数,当解压获取模块后也立即计算出值。

1.3 寻址技术

与数据库的设计相似,图形数据所包含的各种实体元素(例如点、线、面、圆、椭圆、弧线、文字等)、非实体元素(例如字体、图层、线型等)类型列表放置在最前端,或称为 ObjectMap(图元地图),解压时首先获得 ObjectMap,再根据所需获得的目标图元通过遍历表得到关于此图元详细数据的地址,然后根据地址取数据,对于大图形数据,采用寻址技术使得所有数据不必全部解压,仅需根据需求局部解压,大大提高了查询效率。

寻址技术^[9]必须结合校验码使用,即在传输过程中丢失数据或文件损坏造成某元素数据段缺了 n 个字节,那从 ObjectMap 中获得地址相对于真实地址就会提前 n 个字节造成读取错误,如果每段数据有结束校验码则除了丢失数据图元外其余经过修复仍是可用的,预留纠错能力的数据更适应网络化应用。

1.4 分页存储方法

当增加图元时,最佳的数据暂存方案是在末尾

增加数据,以减少向文件中间寻址插数据操作,最佳的删除、修改数据方案是将原图元标志位设成失效然后再末尾添加数据,最后正式保存时做数据全面处理,这种在末尾增加数据的技术称为 Page(数据页)保存^[10]。

图像数据被分割成若干个 Page,每个 Page 最大存储字节数为 512K,超过了就放入下一页,每页有详细的页头信息来说明页内容,分页保存有三点优势:其一,在多数情况下用户绘制和修改新图元可在一个 Page 数据范围内工作,增加了数据传输速度;其二,分页存储在数据丢失、损坏的情况下减少重传数据,且重传丢失页数据时正确页的数据可继续处理;其三,针对一个页面可再采用传统的二级制压缩方法进行数据压缩,增强数据压缩量,解压时在多线程工作环境下多页面即实现了解压与图形加载同步进行。

2 技术应用实例

2.1 DWG 在线看图类软件

DWG 数据文件作为一种工程设计专用图像文件格式,是制图绘图行业的事实工业标准,尤其在桌面制图 CAD 软件中大量采用,只要遵循其规定的压缩方法,开发的软件即可满足通用数据交换的要求,在线看图类软件主要实现 DWG 压缩格式的解压及网页显示,其技术流程如图 1 所示。

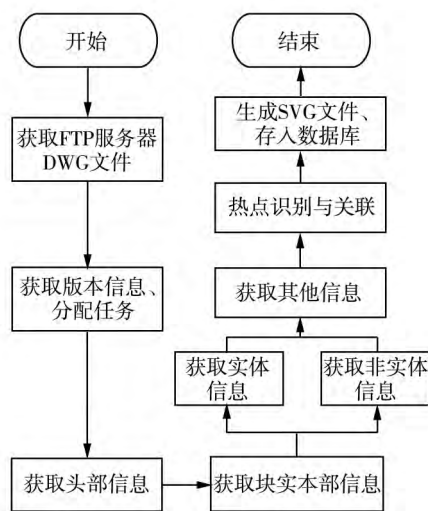


图 1 DWG 在线看图软件技术流程图

登录 FTP 服务器获取相关文件,FTP^[11]即文件传输协议(File Transfer Protocol),采用客户/服务器模式,客户机与服务器之间通过 TCP 建立连接,实现客户机与服务器之间文件的双向传输。本文在 DWG 数据文件网页在线显示技术的实现中采用 FTP 客户端软件对服务器进行访问获取相关信息。

由于不同版本的 CAD 软件生成的 DWG 格式文件不同,因此成功登录服务器获取 FTP 文件后,根据获取到的 DWG 格式文件对应的 Auto CAD 版本信息^[12],对解析任务合理分配给多台解析服务器,分别获取 DWG 格式文件的头部信息、块实体部信息、实体部信息、非实体部信息及其他信息,经过对各部分解析后对 DWG 格式文件中的热点信息进行识别与划分并建立连接,最后生成信息相互关联的 SVG 文件^[13]存入数据库实现 DWG 格式文件的在线显示。

DWG 数据结构的实体部包含了 DWG 格式文件中的实体元素点、线、圆、弧、块等所有实体信息,其实体对象的一般数据结构如下:

MS: Size of object, not including the CRC

//实体对象 bit 大小

BS: Object type//实体对象类型

RL: Size of object data in bits

//实体对象数据 bit 大小(R2000 Only)

H: Object's handle//操作识别号

BS: Size of extended object data, if any

//实体对象的扩展数据 bit 大小

X: Extended object data, if any

//实体对象的扩展数据

B: Flag indicating presence of graphic image.

RL: Size of object data in bits

//实体对象数据 bit 大小(R13 - R14 Only)

X: Object data (varies by type of object)

//实体对象的具体数据

X: Handles associated with this object

//该实体对象对应的操作识别号

RS: CRC//校验码

不同的实体对象其利用 C#编程语言^[14]在 Visual Studio 中的二次编码不同,例如对于 DWG 数据文件中的线型实体而言,需获取每条线所对应的 x 起止点、y 起止点、z 起止点,其具体信息的二次编码程序如下所示:

```

oneLine R_Line( byte[] heByte, ref int ibyte,
ref int ibit, CommonEntityData coment) {

```

```

    bool zIsZero = ToolkitReadValue. B( heByte, ref
ibyte, ref ibit); //是否 z 为 0

```

```

    double startPtx = ToolkitReadValue. RD( heByte,
ref ibyte, ref ibit); //x 起点

```

```

double endPtx =

```

```

ToolkitReadValue. DD( heByte, ref ibyte, ref ibit, st-
artPtx); //x 终点

```

```

double startPty =
ToolkipReadValue. RD( heByte ,ref iblyte ,ref ibit); //
y 起点
double endPty =
ToolkipReadValue. DD( heByte ,ref iblyte ,ref ibit ,st-
artPty); //y 终点
if ( zIsZero == false) {
double startPz =
ToolkipReadValue. RD( heByte ,ref iblyte ,ref ibit); //z
起点
double endPz =
ToolkipReadValue. DD( heByte ,ref iblyte ,ref ibit ,st-
artPz); //z 终
doublevalue = ToolkipReadValue. BT ( heByte , ref
iblyte ,ref ibit); //线宽
doublevalue =
ToolkipReadValue. BE( heByte ,ref iblyte ,ref ibit ,ref a ,
ref b ,ref c); //变形比例
CHandleData comh =
CHandleData( heByte ,ref iblyte ,ref ibit ,coment); //
读取尾部 Handle
return new oneLine( coment , comh , startPtx , st-
artPty , endPtx , endPty); }

```

在工程设计及实施过程中,通过对 DWG 格式文件进行二次编码后即可摆脱专业设计软件对 DWG 数据文件的束缚,通过浏览器实现 DWG 数据文件图像网页显示,给工程设计人员、工程管理人员、现场项目施工人员等的工作带来了极大的便利。

2.2 基于矢量图形的业务系统

在矢量图形在线显示的基础上,可将网页中将图元表现为可链接、可点击的热点,基于此超链接图元实现图纸浏览、资产管理、文档管理、设备管理、设备预警、跳转显示等业务功能。在人机交互过程中还可以实现图元的增删改操作,实现例如框选多个图元将之定义为超链接、删除某个图元、复制图元到任意位置等,在前台进行图形操作的同时不断和远程服务器进行通讯,实时传送图形数据,需要解决的关键问题如下。

2.2.1 使用图形操作命令语句及反命令语句加快数据传送速度

在常用绘图操作中,对于图元的修改执行撤销、前进操作是必不可少的,例如删除一个图元,如果直接删除图元数据则无法实现撤销操作,因此在前端设备内存中需同时存储整体数据、临时数据、当前命令集合、前进命令集合,如执行删除的命令行行

数据可设计语句为 DEL ID,则程序工作流程为语句加入存储命令集合并执行命令(即删除视图构件、修改临时数据);当执行撤销操作时,程序工作流程是执行最后一行命令行的反操作(即从整体数据中将 ID 对应的构件写入临时数据、视图绘制构件),另将当前命令集合最后一行移动至前进命令集合;若再执行前进操作,程序流程为移动前进命令集合最后一行至当前命令集合并执行,最后对图形文件的保存即将压缩后的命令行数据上传服务器,这样对于大规模图形数据文件的数据交互中避免了图形临时数据(即全部数据)传输延迟,并且命令行语句完全适用图形数据专用压缩方法。

2.2.2 采用多线程加快解压速度

以 HTML5 Web Worker 提供的后台线程开发网页在线图形操作系统^[15]为例,当加载图形数据时往往造成用户使用延迟,主要原因有以下两点:其一,是等待全部解压完成时间长;其二,图形显示与用户操作共用线程时图形显示占用 CPU 资源多表现为卡顿。

采用多线程^[16]减少主线程的 CPU 占用率并加快解压速度,例如划分 5 个线程,1 条是主线程也是用户操作线程和图形显示线程,4 条是后台解压线程,分管解压不同的 Page 页面并向主线程发送绘图申请。

3 结束语

本文参照 DWG 格式数据压缩方法对数据压缩技术进行了介绍和分析,尤其在互联网时代大量基于云平台的图形操作系统的出现,对图形文件压缩提出更高的要求。

参考文献:

- [1] 刘小婷. 基于立体重构的工程制图作业智能批改技术的研究与实践[D]. 镇江: 江苏科技大学, 2016.
- [2] 冯瑛琪. 分析新时期建筑管理方法的智能化应用[J]. 建材与装饰, 2018(28): 139-140.
- [3] 刘从军, 楚嘉琦. Local Storage 存储机制性能的研究与改进[J]. 信息技术, 2018, 42(9): 96-100.
- [4] 张宏涛. 基于 HTML5 技术的移动学习平台建设与应用[J]. 轻工科技, 2018(4): 64-65.
- [5] 梁洪波, 王占丰, 秦岭. 基于先验知识的图纸转换技术研究[J]. 云南电力技术, 2017(5): 64-66.
- [6] 张轶, 黄成. 计算机图形学与图形图像处理技术的应用[J]. 电子技术与软件工程, 2018(10): 59-60.
- [7] 韩立敏, 田泽, 张骏, 等. 图形处理器流水线数据压缩技术研究综述[J]. 云南电力技术, 2018(3): 648-653.
- [8] 蒋丽, 邹时林. CAD 到 GIS 数据的属性匹配和转换的研究[J]. 测绘地理信息, 2017(3): 58-62.
- [9] 刘志, 梁凌霄. 单向条件接收系统特征值寻址技术及其应用探讨[J]. 中国传媒科技, 2016(7): 54-55.

(下转第 128 页)

据。从图中可以看出,校正后的最大静态漂移距离最多为 0.15m,两端最大距离比原始数据减小了 50% 以上。由此可见,该校正算法对于减少静态定位漂移是有效的。

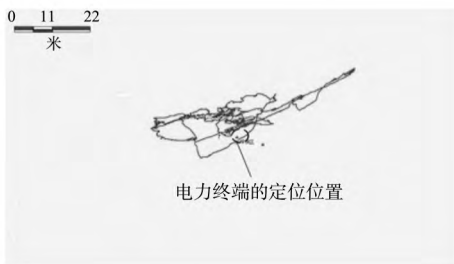


图5 偏移校正前的电力终端的静态定位和动态轨迹



图6 偏移校正后的电力终端的静态定位和动态轨迹

表 2 统计了此次实验中偏移校正前后电力终端的平均定位精度,其中 dX 、 dY 分别表示经纬度 2 个坐标分量的平均定位精度。从表 2 可以发现,偏移校正后电力终端的二维坐标分量的定位精度均有所提高,进一步证明了本文提出的偏移校正算法的优越性。

表 2 偏移校正前后电力终端平均定位精度对比/m

	dX	dY
偏移校正前	0.38	0.17
偏移校正后	0.14	0.15

3 结束语

在本文中,提出了一种易于实现的校正算法来减少 BDS 的静态定位漂移。静态定位是整个定位

系统的重要组成部分。误差源在理论方法上不易分析,严格按照北斗系统(BDS)精密点定位(PPP)的工作原理进行建模。因此,提出了一种移动平均算法,其中不需要为处理该问题建模误差源。实验结果表明,与原始数据相比,校正数据中最大静态漂移距离减少了近 70%,两端距离的最大值减小了 50% 以上。因此,可以得出结论,所提出的静态 BDS 定位漂移校正算法在降低位置漂移方面是有效的,并且在实际应用中易于实现。

参考文献:

- [1] 董建军,徐爱功,高猛,等. GPS 实时精密单点定位精度与收敛性分析[J]. 导航定位学报, 2018, 6(3): 92-97.
- [2] 张瑜,李爽,朱一帆,等. 北斗系统对流层延迟改正模型精度分析[J]. 雷达科学与技术, 2018, 16(4): 366-370.
- [3] 张兆龙,王跃钢,腾红磊,等. 高阶 DMCKF 的 BDS/GPS 精密单点定位算法[J]. 现代防御技术, 2018, 46(3): 48-53, 92.
- [4] 常志巧,胡小工,郭睿,等. 北斗导航卫星相位中心修正[J]. 武汉大学学报: 信息科学版, 2018, 43(6): 860-866.
- [5] 邓豆豆,李灯熬,廖述剑,等. 北斗/GPS 组合系统精密单点定位算法研究[J]. 电视技术, 2018, 42(6): 64-66, 73.
- [6] 黄祥雄,廖超明,唐丹. 基于广西 BDS 观测数据的 PPP 算法研究与精度分析[J]. 海洋测绘, 2018, 38(2): 33-37.
- [7] 齐光松,祝会忠,徐宗秋,等. BDS 不同轨道卫星精密单点定位性能分析[J]. 测绘科学, 2017, 42(12): 44-51.
- [8] 杨勇,黄海生,李鑫,等. 基-2FFT 和基-4FFT GPS 信号快速捕获算法研究[J]. 信息技术, 2017, 41(12): 85-89.
- [9] 张继海,广伟,袁海波,等. 北斗测距信号评估与精密单点定位应用研究[J]. 仪器仪表学报, 2017, 38(11): 2707-2714.
- [10] 徐博,李昊,欧钢. 伪卫星辅助北斗定位精度改进方法[J]. 国防科技大学学报, 2017, 39(5): 56-60.
- [11] 闻志国,陈影,庞振江,等. 一种面向电力终端的北斗定位模块研究[J]. 宇航计测技术, 2017, 37(5): 22-29.
- [12] 邹绪平,王相军. 北斗二代卫星导航系统定位优化仿真研究[J]. 计算机仿真, 2017, 34(6): 80-83, 360.
- [13] 王伟强,汪传建,余晓平,等. 基于 WebGIS 与北斗/GPS 的放牧监测系统[J]. 江苏农业科学, 2017, 45(7): 202-206.
- [14] 吴子岳,高亚东,王董测,等. 基于北斗系统的履带移动机器人定位设计与实现[J]. 全球定位系统, 2016, 41(5): 71-74.
- [15] 高猛,徐爱功,祝会忠,等. 非差观测模型的北斗系统实时动态定位算法[J]. 测绘科学, 2016, 41(3): 3-7.

责任编辑: 杨静

(上接第 124 页)

- [10] 夏春梅. 分页存储管理系统中内存有效访问时间的计算[J]. 滨州学院学报, 2015(6): 94-96.
- [11] 党海燕,麻小娟,詹金珍. 基于 FTP 协议多用户远程数据的访问[J]. 滨州学院学报, 2017(1): 109-111.
- [12] 王蕊. 基于 ObjectARX 和 SQL Server 的 AutoCAD 图形数据存储[J]. 电子测试, 2016(12): 87-88.
- [13] 赵吴鹏. 考虑电网谐波的 SVG 控制策略研究[J]. 信息技术, 2017, 41(5): 137-140.

- [14] 李志权,王旭,朱震震,等. C++ .net 2010 下基于 ActiveX 的 Web 控件开发过程中的常见问题及解决方法[J]. 无线互联科技, 2015(2): 140-142.
- [15] 李忠玉,甘成,许小旭,等. C#智能终端点到系统的开发与实现[J]. 电脑知识与技术, 2018, 11: 69-71.
- [16] 孙经纬,孙广中,詹石岩,等. SA*: 一种多线程路径规划算法[J]. 地球信息科学学报, 2018(6): 753-761.

责任编辑: 丁玥