

文章编号: 2095-2163(2020)04-0208-04

中图分类号: TP391.1

文献标志码: A

# 基于嵌入式 GIS 的矢量地图数据处理技术研究

何 姣, 王 伟

(贵州电子科技职业学院 电气工程系, 贵州 贵安新区 550003)

**摘要:** 为提高嵌入式设备 GIS 的矢量地图数据准确度和地图显示速度, 解决因嵌入式设备内存小运算速度低而引起的卡顿、白页等现象。本文从改变数据结构和提出新型数据压缩算法着手研究, 提出了一种基于垂线距离法和道格拉斯-普克算法思想的新型矢量数据压缩算法。算法采用垂线距离法取出特征点, 用相邻三个特征点连接直线, 形成夹角斜率来判断特征点的有效性, 以有效特征点分割曲线, 重复上述过程, 最终由有效特征点组成压缩后的图像。试验证明, 新型矢量数据压缩算法更逼近原始图像, 算法复杂度较小, 压缩后的图像数据量明显降低。

**关键词:** GIS; 矢量地图; 压缩算法

## Research on vector map data processing technology based on Embedded GIS

HE Jiao, WANG Wei

(Department of Electrical Engineering, Guizhou College of Electronic Science and Technology, Guiyang 550003, China)

**[Abstract]** In order to improve the accuracy of vector map data and map display speed of embedded device GIS, and to solve the phenomenon of stuck and white pages caused by small memory operation speed of embedded device, this paper studies from changing the data structure and proposing a new data compression algorithm. A new vector data compression algorithm based on the vertical distance method and Douglas pucker algorithm is proposed. The feature points are extracted by the vertical distance method, and the validity of the feature points is judged by the angle slope formed by connecting three adjacent feature points to form a straight line. The effective feature points segment the curve, repeat the appeal process, and finally the effective feature points constitute the compressed image. It can reduce the amount of compressed data, improve the accuracy of compressed data, and effectively avoid the distortion of the whole image. Experiments show that the new vector data compression algorithm is closer to the original image, the algorithm complexity is smaller, and the amount of compressed image data is smaller.

**[Key words]** GIS; Vector map; compression algorithm

### 0 引言

随着社会科技的发展, 嵌入式设备已深入生活, 成为必不可少的生活娱乐设备。地图在嵌入式设备上的使用也越来越广泛, 地理信息系统技术(GIS)将地图栅格数据、矢量数据和空间数据库融合在一起, 是地图使用中信息最全面、应用最广泛的技术<sup>[1-3]</sup>。目前 PC 端的地理信息系统技术已非常成熟, 而嵌入式设备处理器运行速度慢、存储空间有限, 使得 GIS 在嵌入式设备上的应用受到了一定限制。随着人们生活中需求的数据量增多, 嵌入式设备上的地图加载速度越慢, 常常出现卡顿、空白区等现象。为加快嵌入式 GIS 地图数据的访问和获取更优的地图显示, 许多学者对地图矢量数据处理进行了研究, 提出了消息调度模型<sup>[4]</sup>、分区索引<sup>[5]</sup>、数据压缩<sup>[3]</sup>等研究方法。这些技术在一定程度上改善了地图矢量数据加载速度, 但仍不能满足社会发展

中数据量不断增大的应用要求。在硬件一定的条件下, 可通过提高矢量地图数据处理技术、提高数据压缩量、运算速度等方式来改善 GIS 在嵌入式设备上的应用。垂线距离法和道格拉斯-普克法是矢量数据压缩算法中的经典, 能够实现一定的矢量数据压缩。垂线距离法对矢量数据的压缩程度低, 并且容易失真; 道格拉斯-普克法压缩程度较好, 能够保持图像基本特征, 但运算复杂度太高, 有可能因为中心点选择不当, 引起矢量数据整体偏离, 造成严重失真<sup>[4-6]</sup>。

为了提高 GIS 矢量地图在嵌入式设备上的运行, 本文从改变数据结构、提出新的矢量数据压缩算法两方面进行了研究。为进一步提高数据压缩程度、降低算法运算复杂度、避免中心点选取不正确引起整体图像失真等现象, 提出了一种新型的矢量数据压缩算法。

**作者简介:** 何 姣(1991-), 女, 硕士, 助理讲师, 主要研究方向: 计算机控制技术; 王 伟(1991-), 男, 硕士, 助理讲师, 主要研究方向: 物联网技术、无线传感网络、计算机控制技术。

**通讯作者:** 王 伟 Email: 2504911553@qq.com

**收稿日期:** 2020-01-01

## 1 改变地理坐标数据精度

通常,地理坐标数据采用双精度(double)数据类型进行存储,该数据类型占用8字节(64位)空间。而3.8英寸的液晶显示屏仅可以显示 $240 \times 320$ 分辨率的图像<sup>[6]</sup>,也就意味着对一张图进行缩放,最多能够达到1:1000 00。double数据类型小数点后保留15位数字,远远超过嵌入式屏幕分辨率能够达到的要求。因此,设计地理数据为double数据类型,只是占用了嵌入式设备的内存空间而没有实际用处。而单精度(float)数据类型仅占用4个字节(32位),小数点后保留6位有效数字,完全能够满足屏幕最大缩放比例要求。将地理坐标数据的数据类型改为float型,相当于将数据存储空间压缩至1/2,既能满足坐标精度要求,又能有效节约嵌入式设备的存储空间,提高嵌入式设备应用程序处理速度,提高地图显示优化效果。

## 2 地理数据图层管理

与PC端相比较,嵌入式设备具有屏幕分辨率低、处理器响应速度慢、内存空间小等特点。一般情况下,在发布GIS系统地图时常选择使用矢量地图,将地图分成“瓦片”数据进行存储。在有限的屏幕范围内显示地图,首先加载粗大数据,当用户对地图进行放大时,加载更多详细数据,当用户对地图进行缩小时,释放掉多余数据,加载更多地图瓦片。设计空间地理数据库中的地理要素进行分层管理,文中以贵州省平坝区马尾松森林环境监测地图为例。地图用于森林环境监测系统,传感器节点所在位置是最受关注的对象,放在第一图层,默认打开地图时加载第一图层矢量数据。其余矢量数据,如:“小班”、“小班控制点”、“路径数据集”、“村”等要素放在优先级低的图层。当用户对选定区域进行放大时显示,缩小时关闭的情况下,利用加载这部分数据的空间资源,加载缓存更多地图瓦片数据,等待新的用户指令。

将空间数据库中的地理要素进行图层管理,能够避免浪费嵌入式存储空间和消耗CPU运行时间。除了将图层设置显示优先级外,可以在地图显示View上创建图层按钮控件,在相应的Fragment上使用PopupMenu创建按钮。用户可以自由选择加载需要的要素图层。使用PopupMenu开发组件创建自定义菜单,将菜单设置为全部可选。当选定需要的图层要素时,将加载相应的要素图层,其效果如图1所示。

功能实现的主要代码如下:

```
//使用动态地图服务创建一个动态地图图层
```



图1 显示或隐藏要素图层

Fig. 1 Show or hide feature layers

```
mMapImageLayer = newArcGISMapImageLayer  
(String URL);
```

```
mMapImageLayer.setOpacity(0.5f); //设置图层  
透明度
```

```
mMap.getOperationalLayers().add(mMapImage  
Layer); //添加为可操作图层
```

```
mLayers = mMapImageLayer.getSublayers(); //获  
取要素服务子图层
```

## 3 矢量数据压缩算法

矢量地图数据的线面数据通常存在大量的顶点,设备加载地图时,绘制大量的点将消耗较长时间,影响地图响应速度。在嵌入式GIS系统中,由于嵌入式设备自身的数据存储量小、CPU运行速度低的特点,对地图数据进行压缩是提高设备运行速度的有效方法。矢量数据压缩是一种有损压缩,通过删除冗余的数据点来简化地图,既能保持图像的基本特性又能简化数据量,提高设备运行速度。

### 3.1 垂距限值法

曲线为一群点组成,从曲线的一端开始取点,连续取出三个点,计算中间点到两端点连成的直线距离,若距离小于限定值 $\varepsilon$ , $d < \varepsilon$ 则舍去中间点, $d \geq \varepsilon$ 则保留中间点<sup>[7]</sup>。

### 3.2 道格拉斯-普克法

道格拉斯-普克法的基本思路是连接每一条曲线的首尾两端,求出曲线上所有点到这条线段的距离,找出最大距离 $d_{\max}$ 并设定阈值 $\varepsilon$ 。若最大距离 $d_{\max}$ 小于阈值 $\varepsilon$ ,则舍去这条线上的中间点,若最大

距离  $d_{\max}$  大于阈值  $\varepsilon$ , 则将这条线以该点分为两部分, 分别对这两个部分重复进行上述过程<sup>[8]</sup>, 如图 2 所示。

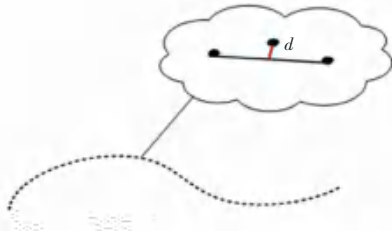


图 2 垂距限值法

Fig. 2 Vertical distance limit method

道格拉斯-普克法能够较好的保持曲线的基本特征, 设置的阈值越小, 曲线的还原性越高。道格拉斯-普克法运算过程中的递归运算方法, 使得数据运算量过大, 数据压缩率降低。设置阈值大, 运算量小, 还原度不高。并且, 如果其中有一个点出现偏差将会导致后续所有计算都偏离原来的方向, 引起图像的严重失真<sup>[9-10]</sup>。

综上所述, 文中采用一种新型的矢量数据压缩算法。基于道格拉斯-普克法的分割思想, 采用垂线距离法取出特征点, 用相邻三个特征点连接直线形成夹角斜率, 判断特征点的有效性, 以有效特征点分割曲线, 重复上述过程, 最终有效特征点组成压缩后的图像。新型的矢量数据压缩算法能达到降低运算量, 避免图像整体走向失真的目的。

### 3.3 一种新型的矢量数据压缩算法

#### 3.3.1 算法思想

为了进一步提高矢量数据压缩算法的准确度和降低道格拉斯-普克法的运算复杂度, 文中提出一种新的算法。该算法的基本思想是: 求出曲线上到端点直线垂线距离最大的点和距离, 通过计算, 取垂线距离以 2 的  $n$  次方递减的方法求出特征点, 用相邻三个特征点连接直线形成夹角斜率判断特征点的有效性, 以有效特征点分割图像。重复上述步骤, 最终求出曲线上所有满足条件的点, 组成压缩后的图像。

#### 3.3.2 算法实现

(1) 设定一个垂线距离限定值  $\varepsilon$ , 斜率限定值  $\theta$ 。

(2) 一条曲线连接首尾两端点为一条直线  $l$ , 求曲线上所有点到该直线  $l$  的距离, 找出最大距离  $D$ , 并保留此点。

(3) 以直线  $l$  作为  $x$  轴, 其垂线为  $y$  轴。向  $y$  轴正负两个方向作距离为  $D/2$  的直线  $l$  的平行直线  $l_{1/2}$  和  $l_{-1/2}$ , 取出交点。

(4) 以  $l_{1/2}$  为例, 判断直线  $l_{1/2}$  与曲线的交点个

数  $N$ , 若  $N=0$ , 说明  $y$  轴正方向上曲线不存在与直线  $l$  距离大于  $D/2$  的点, 若  $N=1$ , 说明  $y$  轴正方向上曲线存在一个与直线  $l$  距离等于  $D/2$  的点。若  $N \geq 2$ , 说明  $y$  轴正方向上曲线存在多个与直线  $l$  距离大于  $D/2$  的点。

(5) 分别以直线  $l_{1/2}$  和  $l_{-1/2}$  作为  $x$  轴, 以  $l_{1/2}$  为例, 若步骤 4 中直线  $l_{1/2}$  与曲线交点个数  $N \leq 1$ , 则只向直线  $l_{1/2}$  作为  $x$  轴的  $y$  轴负方向作距离为  $D/4$  的平行线。再重复步骤 4。若步骤 4 中直线  $l_{1/2}$  与曲线交点个数  $N > 1$  则需向  $y$  轴正负两个方向作距离为  $D/4$  的平行线。重复步骤 4。

(6) 按照距离以 2 的  $n$  次方递减的方式重复步骤 3-5, 直到  $D/2^n \leq \varepsilon$  时结束, 取出的点的位置保持不变。

(7) 计算相邻 3 个点连接而成的两条直线的斜率, 斜率差  $\Delta k$ , 若  $\Delta k \geq \theta$ , 则保留中间点作为分割点, 反之舍弃。

(8) 以分割点进行切割曲线, 将曲线分为若干段, 重复以上步骤 2~7, 直到步骤 2 中取到的最大距离  $D < \varepsilon$ , 结束数据压缩。算法执行过程如图 3 所示。

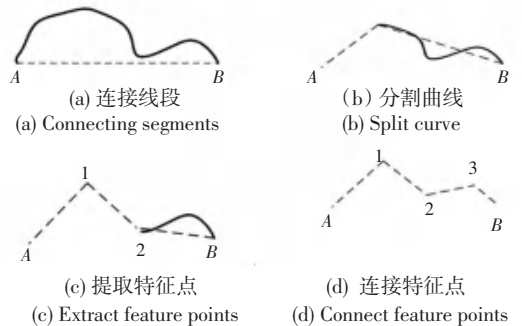


图 3 道格拉斯-普克法

Fig. 3 Douglas puck method

## 4 试验结果

试验采用的嵌入式设备主要配置参数为: ITOP 4412 嵌入式开发平台、微处理器芯片三星 Exynos 4412、电源管理芯片 S5M8767、64 位双通道 DDR3、EMMC 存储芯片、USB 接口扩展器(USB3505)以及 4 组板对板连接器组成。在嵌入式设备中实现新型的矢量数据压缩算法, 文中采用 Android5.0 操作系统、Android studio 应用开发平台、Java 语言编程。试验数据来源于 MapInfo 矢量数据, 比例尺精度 1:10 000, 电子地图距离 0.2 mm, 将  $\varepsilon$  值设置为 0.2 mm,  $\theta$  值设置为 1。研究区域为安顺市平坝区白云镇小河村马尾松林区, 其经纬度坐标为: (106°15'8"~106°15'49"E, 26°19'00"~26°19'32"N)。试验结果如表 1 所示。

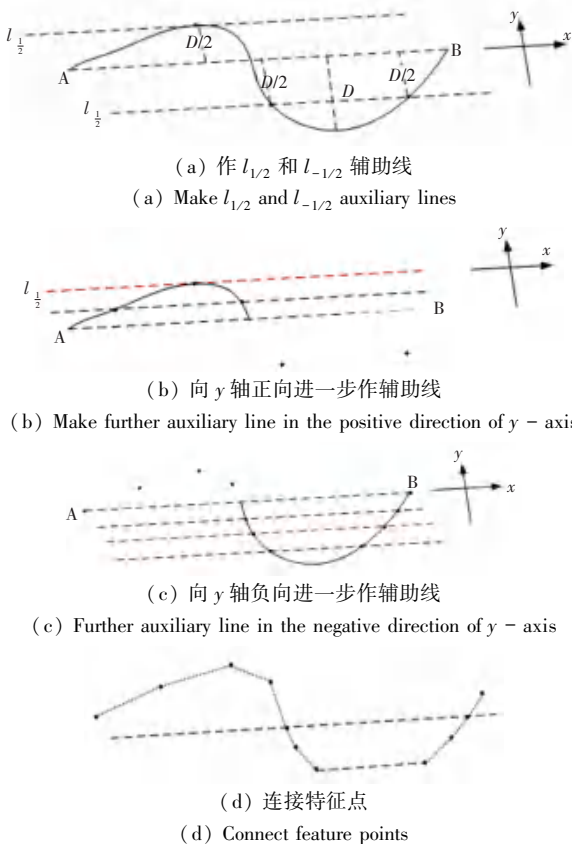


图4 新型矢量数据压缩算法示意

Fig. 4 A new algorithm of vector data compression

表1 压缩后曲线与原始曲线对比

Tab. 1 Comparison between compressed curve and original curve

算法	曲线长度/km	坐标平均值
原始曲线	6.154 268	106.154 722
道格拉斯-普克法	6.086 218	106.147 324
新型算法	6.154 645	106.154 687

由表1可见,新型算法的曲线长度和坐标平均值与原始曲线最接近。从算法复杂度上来看,新型

算法的算法复杂度为  $o(2n - 1)$ ,远低于道格拉斯-普克法算法复杂度  $o(n^3)$ 。算法复杂度越低,运算速度越快。新型的矢量数据压缩算法从压缩后矢量数据还原度和算法速度上都优于道格拉斯-普克算法。

### 5 结束语

文中提供的新型矢量数据压缩算法,弥补了道格拉斯-普克算法复杂度、失真较大的不足。经新型的矢量数据算法压缩后,结果更逼近原始矢量数据。由此可见,本文提出的算法复杂度低、运算速度快、适合在嵌入式设备上运行。

### 参考文献

[1] Kodali, Ravi Kishore, Soratkal, Sreerama. Trust modle for WSN [C]//1st International conference on Applied and Theoretical Computing and Communication Technology, 2015: 903-906.

[2] SHEN J, MOLISHCH A F. Indirect path detection based on wireless propagation measurements [J]. IEEE Transactions on Wireless Communications, 2012, 11(12): 4482-4493.

[3] Jing Lu, Dianjie Lu, Xiaoxia Huang. Channel Model for Wireless sensor networks in Forest Scenario [C]// IEEE 2010 2nd International Asia conference on Informatics in Control, Automation and Robotics, 2010: 476-479.

[4] 冯志红,赵拥军,李茂光. 消息调度模型下一种新型的双缓存地图漫游算法[J]. 测绘科学,2013,38(4):181-185.

[5] 冯鹏飞,于新文,张旭. 北京地区不同植被类型空气负离子浓度及其影响因素分析[J]. 生态环境学报,2015,24(5):818-824.

[6] 张成花. 嵌入式GIS中矢量地图快速显示方法研究[J]. 计算机工程与设计,2010,31(14):113-118.

[7] 杨安宁,周莹,吕康. 一种Android平台GIS软件的新型数据缓存策略的实现[J]. 计算机与现代化,2016.10:127-132.

[8] 姜代红. 基于影像金字塔的GIS地图动态漫游算法[J]. 计算机工程与设计,2013,5:95-100.

[9] 胡泽明,岳春生,李婧. 嵌入式GIS系统矢量地图数据压缩方法[D]. 信息工程大学,2013.

[10] 赵宇峰,孙伟华,王建国. 基于嵌入式GIS的地图显示算法研究[J]. 计算机与数字工程,2011,39(3):170-173.

(上接第207页)

[9] RASOULZADEH R, SHAHRI A M. Implementation of A low-cost multi-IMU hardware by using a homogenous multi-sensor fusion [C]//2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA). IEEE, 2016: 451-456.

[10] MARTIN H F S, GROVES P D, NEWMAN M, et al. A new approach to better low-cost MEMS IMU performance using sensor arrays[C]. The Institute of Navigation, 2013.

[11] PANAHANDEH G, SKOG I, JANSSON M. Calibration of the accelerometer triad of an inertial measurement unit, maximum likelihood estimation and Cramer - Rao bound [C]//2010 International Conference on Indoor Positioning and Indoor Navigation. IEEE, 2010: 1-6.

[12] NILSSON J O, SKOG I, HÄNDEL P. Aligning the forces—Eliminating the misalignments in IMU arrays [J]. IEEE Transactions on Instrumentation and Measurement, 2014, 63(10): 2498-2500.

[13] RASOULZADEH R, SHAHRI A M. Accuracy improvement of a

multi-MEMS inertial measurement unit by using an iterative UFIR filter[C]//2017 European Navigation Conference (ENC). IEEE, 2017: 279-286.

[14] GAN Q, HARRIS C J. Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion [J]. IEEE Transactions on Aerospace and Electronic systems, 2001, 37(1): 273-279.

[15] SEIFERT K, CAMACHO O. Implementing positioning algorithms using accelerometers[J]. Freescale Semiconductor, 2007: 1-13.

[16] YU C, LAN H, LIU Z, et al. Indoor map aiding/map matching smartphone navigation using auxiliary particle filter [C]//China Satellite Navigation Conference (CSNC) 2016 Proceedings: Volume I. Springer, Singapore, 2016: 321-331.

[17] 陈卫东,张飞. 移动机器人的同步自定位与地图创建研究进展[J]. 控制理论与应用, 2005, 22(3):455-460.

[18] GONCALVES L, DI BERNARDO E, BENSON D, et al. A visual front-end for simultaneous localization and mapping [C]// Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, 2005: 44-49.