Douglas-Peucker 算法在无拓扑矢量数据压缩中的改进

谢亦才[1],李 岩[2]
XIE Yi-cai[1],LI Yan[2]

1.华南师范大学 计算机学院,广州 510631
2.华南师范大学 空间信息技术与应用研究中心,广州 510631
1.Scholol of Computer,South China Normal University,Guangzhou 510631,China
2.Spatial Information Research Center,South China Normal University,Guangzhou 510631,China
E-mail:xieyicaiwlz@tom.com

XIE Yi-cai,LI Yan.Improvement on Douglas-Peucker algorithm for non-topology vector data.Computer Engineering and Applications,2009,45(32):189-192.

Abstract: The paper analyzes the reason of graphic distortion phenomenon when compressing the non-topology vector graphics by classical Douglas-Peucker algorithm.The reason is that the common boundary is compressed with Douglas-Peucker algorithm more than one time.Based on this,an improvement on Douglas-Peucker named common boundary objected Douglas-Peucker improving algorithm is put forward.To implement it,first this paper designs a new algorithm for extracting common boundary between two polygons.Then it adopts the idea of OOP,packages the common boundary to be a class with other information showed in figure two.Finally,on the basis of the class,it applies classical Douglas-Peucker to the common boundary and non-common boundary of polygons respectively to compress vector graphics.At last,the validity of the new compressing algorithm is proved by experiment with SVG graph.And the advantages in space and time efficiency are analyzed by contrast with the other Douglas-Peucker improving algorithm showed in table one and two.

Key words: Douglas-Peucker algorithm;vector data compressing;Scalable Vector Graphics(SVG);common boundary objected Douglas-Peucker improving algorithm

摘 要:分析了常规 Douglas-Peucker 算法压缩无拓扑矢量数据时产生公共边"裂缝"现象的原因,即公共边被两次或可能更多次压缩,而每次运用 Douglas-Peucker 算法压缩时所选择的初始点和终点不同造成的。为此,提出了公共边对象化 Douglas-Peucker 改进算法。为实现此算法,首先设计了新的公共边提取算法来提取公共边,然后使用 OOP 技术,把公共边的相关信息封装成类,最后根据公共边对象提供的信息对多边形的公共边和非公共边分别进行 Douglas-Peucker 压缩。以广东省行政界线的 SVG 矢量图为实验对象验证了该算法的有效性,分析了该算法相对于其他 Douglas-Peucker 改进算法在所需辅助空间和时间效率上的优势。

关键词:Douglas-Peucker 算法;矢量数据压缩;可缩放矢量图形(SVG);公共边对象化 Douglas-Peucker 改进算法
DOI:10.3778/j.issn.1002-8331.2009.32.060 **文章编号**:1002-8331(2009)32-0189-04 **文献标识码**:A **中图分类号**:TP391

1 引言

随着 WebGIS 的迅速发展,海量空间数据在目前带宽有限的网络上的传输速度慢的问题越来越突出,因此有必要对空间数据进行压缩。

对空间数据的压缩技术可分为无损压缩和有损压缩。无损压缩算法已经很成熟,最常用的有 LZW 算法。而有损压缩,特别是对矢量图形(如:Mapinfo 的 MID 和基于 XML 的 SVG 矢量数据)的有损压缩,近年来引起了不少学者的关注。目前,矢量数据压缩算法主要有:距离控制类算法,如垂距限值法、Douglas-Peucker 算法(部分文献称之为 Spliting 算法);角度控制类算法,如角度限值法;黄培之 1995 年提出的具有预测功能的曲线矢量数据压缩方法[1];以及新兴的基于小波技术的压缩方法。由于基于小波技术的压缩算法,压缩后的数据在边界处会出现变形,且压缩过程复杂、对计算机要求高等缺点,而距离计算相对于角度计算在执行效率方面的优势,使得垂距限值法,特别是 Douglas-Peucker 算法(简称 DP 算法)的应用较普遍。

DP 算法最基本的思想是通过删除一条曲线上的非特征点而保留特征点来减少数据量。它被提出后,为了适应应用要求,很多学者对它进行了改进:(1)为了提高时间效率,John Hershberger 等在文献[2]以及 P.K.Agarwal 等在文献[3]中对 DP 算法实现进行改进,大大减少了时间复杂度;(2)在压缩诸如公路等弯曲度很大的曲线时,有可能使本来没有自相交的曲线压缩

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60842007);广东省百项工程项目(No.2005B30801006)。
作者简介:谢亦才(1981-),男,硕士生,主要研究方向:图形图像处理,WebGIS;李岩(1955-),女,教授,硕士生导师,主要研究方向:空间信息技术应用、图形图像处理。
收稿日期:2008-12-01 **修回日期**:2009-03-06

后会自相交,为解决此问题,Wu S.T 等人在文献[4]中提出了无自相交 DP 算法;(3)最近几年,陈飞翔等人以提高压缩比和减少误差为优化目标,在文献[5]和文献[6-7]分别引进遗传算法和动态规划算法,但文献[5]和文献[6]只针对单条曲线的优化压缩,没有考虑多条曲线时的情况,文献[7]虽然适用于多条曲线的优化压缩,但它没有考虑曲线间的相邻等拓扑关系,导致出现如实验结果图 5 所示的部分"裂缝"现象;(4)在压缩用闭合曲线表达的多个相邻多边形时, 会出现如第 2.2 节那样的"裂缝"现象。因此,在空间数据压缩中有必要提出一种方案来解决此问题。文献[8-9]采取先提取相邻多边形的公共边,然后利用常规的 DP 算法对其等效数据进行压缩,压缩完毕后再重建数据,按照原始的数据格式进行存储。这两种方法需要很大的辅助空间存储等效数据[8](或元数据[9])。文献[10]采取先把存在相邻关系的两个多边形分成公共边和非公共边,即进行逻辑分段(在公共边的两端作记号),然后再对相邻两个多边形的各自的公共边和非公共边分别进行分段压缩。这种方法需要对公共边进行重复压缩,导致时间效率低。为此,设计了一种公共边对象化的 DP 改进算法,有效弥补了上述两方面的不足,又解决了常规 DP 算法压缩无拓扑矢量数据时会产生"裂缝"的问题。

2 理论与方法

2.1 压缩算法相关数据结构及其特点

当前主流 GIS 的空间数据格式主要通过两种数据模型来表示矢量多边形图层。第一种是拓扑数据结构,该结构的基本元素是"弧段",弧段的两个端点是结点,中间有任意多个中间结点,而多边形则是由一系列首尾相连的弧段组成的,相邻的多边形共享公共弧段, 整个系统分别维护各自数据的结点表、弧段表和多边形表。第二种是简单数据结构(俗称无拓扑数据结构),如 MapInfo 的 MID 格式、SVG。在简单数据结构中,地理实体仅被抽象为点、线、面三种基本类型,每个空间对象只记录、维护自己所有的图形信息和属性,且每个对象都是自包含或独立对象,没有相邻等拓扑信息[11]。

这两种数据结构主要的差异有:(1)前者相邻多边形之间的公共边界被数字化且只存储一次;而后者要存储两次,由此产生冗余和两多边形被 DP 压缩后公共边不一致。(2)前者每个多边形有邻域信息,便于进行邻域处理,如邻域搜索;而后者每个多边形自成体系,缺少邻域信息。

2.2 常规 Douglas-Peucker 算法及其不足

DP 算法的基本思想是: 对任何一条由坐标点序列组成的线段或封闭曲线(表示一个多边形),选择其中两点分别作为始点和终点,然后虚连接这两点形成一条线段,再计算曲线上的其他任一点到这条虚线段的距离并找到最大距离 d_{max},使 d_{max} 与阈值 D 比较:若 $d_{max}<D$,则曲线上的始点和终点之间的中间点全部舍去;若 $d_{max}>D$,则保留 d_{max} 对应的坐标点,并以该点为界,把曲线分为两部分,再分别对这两部分重复使用这种方法,直到始点和终点之间无中间点为止。

从上述思想可知:DP 算法对初始条件(初始点)较为敏感[12],即对同一曲线压缩时,若选择的始末坐标点不同,压缩后保留的坐标点也不同。例如,假设有两个相邻多边形 A 和 B,如图 1(a)所示:选取阈值 D,对于多边形 A,选择坐标点 2 和坐标点 5 作为始点和终点,对于多边形 B,以坐标点 6 和坐标点 13 作为始点和终点。当对多边形 A 和 B 进行压缩时,很可能坐标点 7 到线段 $L(2,5)$ 的距离小于 D 而被舍去,但它到线段 $L(6,13)$ 的距离大于 D 而保留,导致压缩后 A 的坐标序列为(1,2,3,4,5,6,8,9,1),而 B 的坐标序列为(1,13,12,11,10,6,7,8,9,1)(假设其他点保留),这样看起来两多边形之间公共边出现"裂缝"(如图 1(b)所示)。

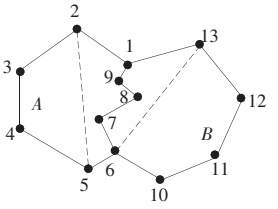

图 1(a) DP 前两相邻多边形

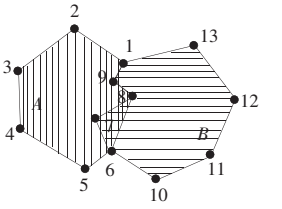

图 1(b) DP 后两相邻多边形

2.3 对常规 Douglas-Peucker 算法的改进

为解决上述"裂缝"问题,很多学者对 Douglas-Peucker 算法进行了改进。文献[13]提出将对有公共边的两多边形进行分段压缩的思想, 但在该文没有提出具体如何分段及压缩后如何重组的问题;文献[8]提出更具体的解决办法:先提取相邻多边形的公共边,按照一定的逻辑结构生成等效数据,再利用常规 DP 算法对等效数据进行压缩,压缩完毕后再重建数据,按照原始的数据格式进行存储;文献[9]提出切分重组的思想,与文献[8]思想类似,只是把等效数据换成元数据。这两种方法需要很大的辅助空间存储等效数据[8](或元数据[9]);为了减少辅助空间,文献[10]提出了逻辑分段压缩的思想: 首先将多边形公共边界的两个端点作为约束点处理(做好分段标记),使得多边形从约束点处逻辑上分成几段;然后利用常规 DP 算法进行分段压缩,使每一多边形公共边界压缩时的初始点和结束点一致,从而保证了无拓扑多边形数据的一致性压缩,但这种方法需要对公共边进行重复压缩,导致时间效率低。为了既减少辅助空间又节省压缩时间,对 DP 算法进行改进(称为公共边对象化 DP 改进算法)。

3 公共边对象化 DP 改进算法的实现

为实现该算法思想,先要提取相邻多边形的公共边,并封装成类,再分别对公共边和非公共边进行压缩。

3.1 提取公共边

要对公共边和非公共边分别进行常规 DP 压缩,首先必须提取出公共边。文献[8-10,13]中都使用深度搜索匹配算法来提取公共边。该文设计的提取公共边算法源于文献[14]中的字符串的模式匹配算法(KMP 算法),算法基本步骤如下:

(1)将坐标串 s_1 和 s_2 分别存入两个数组(设数组名为 s_1、s_2)中,一个单元存放一个坐标(x 和 y 值)。把这两个数组比作两把刻度尺,将 s_2 固定,s_1 的尾部和 s_2 的头部对齐。设初始公共坐标串 *ComMax* 为空。

(2)s_1 向右移一个单位,若 s_1 的头部移到 s_2 的尾部,则转向第 4 步;否则,依次比较重叠部分的坐标串中的坐标。若有相同坐标串时,则记下其在各自坐标串中的初始位置、结束位置和这个公共坐标串并转向第 3 步;若重叠部分没有坐标相同,则继续执行第 2 步。

(3)将此次重叠部分公共坐标串长度跟 *ComMax* 比较,若此次的公共坐标串更长,则将它赋给 *ComMax* 并更新初始和结束位置;否则 *ComMax*、初始和结束位置不变,转向第 2 步。

(4)返回公共坐标串及公共坐标串在各自坐标串中的初始

和结束位置，算法结束。

3.2　公共边对象化压缩算法思想及其流程

为实现该算法思想，使算法相对于文献[9]，减少辅助空间，相对于文献[10]，提高时间效率。为此，采用 OOP 技术，创建公共边类，公共边类的结构片断如图 2 所示。每提取出一段公共边就用此公共边类将其实例化，即创建了公共边对象，并将此公共边的各种标记信息初始化。然后把对象存入动态数组。当有此公共边的相应多边形 j 要压缩时，查看 Compressed 标记是否为 true，若是，则直接将压缩过的公共边替换多边形 j 中的此段公共边，再压缩 j 中的非公共边；若 Compressed 标记为 false，则分段压缩多边形 j，并将压缩后的公共边坐标存入动态数组 *ComCoord-Vector* 中，并修改对象中相应的 *Compressed_j* 标记为 true。当 *Compressed_i* 和 *Compressed_j* 都为 true 时，说明共用此公共边的两多边形都压缩完了，可把此对象销毁，以减少空间。

```
public class ComCoordinate{ //用来记录一次调用求公共坐标后得到的公共坐标的始末位置，以及在后面利用它的 Compressed_i 进行分段压缩时判断它是否已被压缩过.
private
    int Curve_i;              //有公共边的第 i 条曲线.
    boolean Compressed_i;     //有公共边的第 i 条曲线是否已被压缩过了.
    int Curve_j;              //有公共边的第 j 条曲线.
    boolean Compressed_j;     //有公共边的第 j 条曲线是否已被压缩过了.
    int ComStart_i;           //有公共边在第 i 条曲线中的初始位置.
    int ComStart_j;           //有公共边在第 j 条曲线中的初始位置.
    int ComEnd_i;             //有公共边在第 i 条曲线中的结束位置.
    int ComEnd_j;             //有公共边在第 j 条曲线中的结束位置.
public
    Vector ComCoordVector=new Vector(100,50); //用于存贮公共坐标，存的是第 i 曲线上的公共位置开始的坐标.
    boolean Compressed;       //用于表示此公共坐标有没有被压缩过，若压缩过则其值为 true
```

图 2　公共边类结构片断

借助上面设计的 ComCoordinate 类，设计该算法的基本思想为：逐次对图形中所有多边形用最小外接矩形 MBR 两两判断是否存在公共边，若不存在，则调用常规 DP 算法直接对其压缩；若存在公共边，则调用该文设计的公共边提取算法提取公共边，若该公共边是第一次提取到，则将公共边生成一个公共边对象（对象中记录有共享该公共边的两条曲线的编号、公共边在各自多边形中的初始和终端位置、共享该公共边的两条曲线是否已压缩完毕标志等），并将该对象放入动态对象数组中，然后分别对非公共边和公共边调用常规 DP 算法进行压缩；若该公共边对象已存在，则先只压缩非公共边，然后直接用前面已压缩过的公共边替换此公共边（避免了再次压缩），并把此公共边对象销毁以减少辅助空间。每一次把压缩完的曲线写回源文件中。算法的流程图如图 3 所示。

4　实验与分析

为了验证算法的可行性，并且与文献[9]和文献[10]以及文献[7]改进算法进行比较，在如下环境中实现了这三种 DP 改进算法。硬件环境：AMD Athlon 3000+，2.0 GHz，内存 512 MB；软件环境：Windows XP 系统，Java1.5 编程语言，DOM 接口和 Eclipse3.2 开发平台。

以广东省行政界线的 SVG 矢量数据为实验对象，实验效果

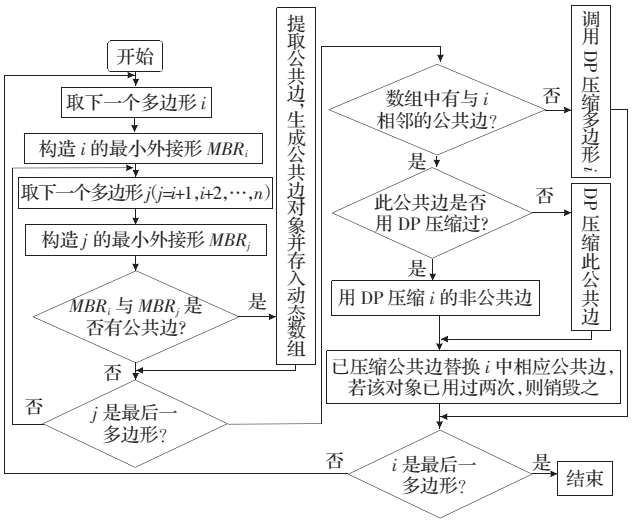

图 3　公共边对象化 DP 改进算法流程图

对比如图 4(a)（用常规 DP 压缩后的图）和图 4(b)（该文算法压缩后的图）所示，从图中可以明显看到：用常规 DP 压缩时，在阈值为 10 时，由于没有特殊处理公共边，导致公共边压缩结果不一致，出现“裂缝”，放大图片肉眼都可看到更多“裂缝”；而采用该文改进算法后，即使在阈值是 10 的 10 倍的情况下也不会出现公共边“裂缝”现象。压缩性能指标如表 1 所示，从表中可知该文算法的压缩比是比较大的，跟文献[9]的算法的压缩比几乎相等，主要是因为压缩原理都是基于 DP 算法的，压缩比（压缩比=压缩后文件大小/压缩前文件大小）的大小主要取决于常规 DP 算法和原始数据的冗余程度。

图 4(a)　阈值为 10 常规 DP 压缩效果图

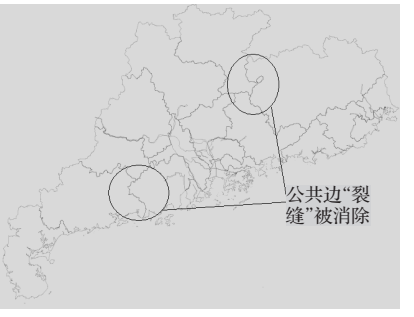

图 4(b)　阈值为 1 000 改进 DP 压缩效果图

除了消除“裂缝”外，从表 1 的对比结果可知：该算法与文献[9]相比，大大减少了算法的辅助空间。因为“在提取公共边生成等效的元数据时，占用了几乎与原数据相等的存储空间”[9]。而该文算法用动态数组来存放公共边对象（在对象中记录有关公共边信息），当公共边所在的两个多边形都被压缩完时，就销毁它，这

表1 公共边对象化DP改进算法与文献[9]的算法压缩效果对比表

	公共边对象化DP改进算法				文献[9]的算法			
阈值	1	10	100	1 000	1	10	100	1 000
压缩时间/s	59	58	52	48	72	70	65	63
辅助空间/KB	212	206	202	197	2 448	2 446	2 439	2 430
压缩比	0.440	0.426	0.307	0.161	0.440	0.423	0.309	0.160

样大大节省了空间。因为根据人的习惯,相邻多边形一般会记录存放在一起,所以在压缩时,相邻的一个多边形压缩完后,另一个很快就会被压缩,于是它们的公共边对象被销毁,所以在动态数组中不会堆积太多公共边对象,从而达到减少辅助空间的目的。

与文献[10]相比,从表2的对比结果可知:由于该算法减少了不必要的对公共边的重复压缩,节省了压缩时间。因为文献[10]通过逻辑分段,分成公共边和非公共边压缩,压缩时对一个多边形分两段压缩,而公共边在另一个相邻多边形压缩时会再次压缩,所以公共边实际被压缩了两次。而在一个地图中,除了最外围边界,里面的边界大部分都是两个多边形共用的,即这种公共边占总多边形边的数量的比例很大,所以文献[10]重复压缩导致浪费的时间很多。而该文算法对公共边在第一个多边形中压缩后,第二个多边形可直接将其公共边压缩结果复制过来,更能节省不必要的压缩时间。

表2 公共边对象化DP改进算法与文献[10]的算法压缩效果对比表

	公共边对象化DP改进算法				文献[10]的算法			
阈值	1	10	100	1 000	1	10	100	1 000
压缩时间/s	59	58	52	48	82	80	75	73
压缩比	0.440	0.426	0.307	0.161	0.440	0.425	0.307	0.162

为了验证该文算法相对于文献[7]在消除“裂缝”方面的优越性,选择相同的阈值1 000和压缩比0.161,得到的实验结果效果如图5所示。从图中可以看出,用文献[7]的算法并不能完全消除“裂缝”。究其原因,先看文献[7]的算法的总体思想:首先对每一条曲线(又称一个实体或多边形)进行基于动态规划的DP压缩,并计算其误差代价函数$D(Nb,h)$;然后根据每条曲线压缩后的误差代价函数$D(Nb,h)$分配压缩后要保留的Ne个坐标点($Ne=Nb*\eta$,其中Nb为曲线压缩前原有坐标点数,η为事先设定的压缩比,为具有可比性,在此η取值为该文算法在阈值1 000时得到的压缩比);最后,对所有曲线保留的Ne个坐标点重新进行动态规划,得到整个地图最后的压缩结果图。而在这三个步骤中的第一步中首先要通过DP计算出一条参考压缩曲线(又称路径),这里用DP时阈值取与该文算法相同的1 000。整个算法过程的基础还是DP算法,没有特别处理相邻多边形(曲线)压缩时的“裂缝”问题,只是在算法的第三步对所有曲线进行总体的动态规划算法,这不能完全消除DP算法压缩无拓扑相邻多边形时的“裂缝”现象。

5 结语

对矢量数据的压缩,DP算法因其压缩比较高和保真性好而被普遍采用。但由于它对公共边进行压缩时会出现“裂缝”现象。为此通过把公共边的坐标及其相关信息封装成类,采用常规DP算法进行压缩,取得了两个效果:

图5 阈值为1 000压缩比为0.161时文献[7]压缩效果图

(1)解决了常规DP算法压缩时出现的公共边“裂缝”问题;

(2)减少了压缩时间和辅助空间。

该文提出的公共边对象化矢量数据压缩算法设计合理科学,但和DP有损压缩算法一样,需要用户自己选择阈值。阈值过大,虽然能得到高压缩比,但失真较明显;阈值过小,则保真效果较好,但得不到理想的压缩比。所以将在后面的工作中,研究如何让算法根据矢量数据特点自动选择合适的阈值,以取得高压缩比与保真效果之间的平衡。

参考文献:

[1] 黄培之.具有预测功能的曲线矢量数据压缩方法[J].测绘学报,1995,24(4).

[2] Hershberger J,Snoeyink J.An O($n\log n$) implementation of the Douglas-Peucker algorithm for line simplification[C]//Proceedings of the Tenth Annual Symposium on Computational Geometry,1994-06:383-384.

[3] Agarwal P K,Har-Peled S,Mustafa N H,et al.Near-linear time approximation algorithms for curve simplification[J].Algorithmica,2005,42:203-219.

[4] Wu S T,Mercedes Rocfo Gonzales Marque.A non-self-intersection Douglas-Peucker algorithm[C]//Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing(SIBGRAPI'03),IEEE,2003.

[5] 陈飞翔,于文洋,李华.基于GA的矢量数据压缩优化算法[J].计算机工程与应用,2007,43(34):185-187.

[6] 陈飞翔,周治武,张建兵.基于动态规划算法的矢量数据压缩改进算法[J].计算机应用,2008,28(1):168-170.

[7] 陈飞翔,李华,于文洋.基于多实体的矢量数据压缩改进算法[J].计算机工程与应用,2008,44(19):200-202.

[8] 王净,江刚武.无拓扑矢量数据快速压缩算法的研究与实现[J].测绘学报,2003,32(2):173-177.

[9] 张胜,朱才连,钟世明.Douglas—Peucker算法的改进及应用[J].武汉理工大学学报:交通科学与工程版,2005,29(5):671-674.

[10] 吴正升,成毅,郭婧.基于约束点的无拓扑多边形数据压缩算法[J].测绘科学技术学报,2006,23(3):202-207.

[11] 薛胜,潘懋,王勇.多边形叠置分析算法研究[J].计算机工程与应用,2003,39(2):57-60.

[12] 应申.曲线的一致性化简及曲线相交的研究[D].武汉:武汉大学,2002:26-28.

[13] 翟战强,管华,王双亭.一种快速空间矢量数据压缩方法[J].计算机工程,2003,29(2):94-95.

[14] 严蔚敏,吴伟民.数据结构(C语言版)[M].北京:清华大学出版社,1996:79-84.