一种大规模矢量地图数据实时简化的方法

杨 玲[1)] 张立强[1)] 何子琳[2)] 陈 晨[1)]

[1)](北京师范大学地理学与遥感科学学院,遥感科学国家重点实验室,北京 100875)

[2)](中国移动北京公司,北京 100007)

摘 要 提出利用帧缓存和 Voronoi 图实现大规模矢量地图的快速简化以及用模板缓存剔除 Voronoi 图中因自相交而重叠的区域,避免了简化后要素间错误相交、自相交,点线位置改变和多边形邻接关系改变的拓扑错误,有效保持了简化前后拓扑关系的一致性。该方法的简化时间复杂度始终控制在一定范围内。该研究有助于提高多尺度、大尺度矢量数据融合的精度和效率。

关键词 矢量地图 多分辨率模型 简化

中图法分类号:TP391.41,P282 **文献标识码**:A **文章编号**:1006-8961(2009)06-1007-05

A Real-time Simplication Method for Large Vector Map Data

YANG Ling[1)], ZHANG Li-qiang[1)], HE Zi-lin[2)], CHEN Chen[1)]

[1)](*State Key Laboratory of Remote Sensing Science, Jointly Sponsored by Beijing Normal University and the Institute of Remote Sensing Applications of Chinese Academy of Sciences, Beijing* 100875)

[2)](*Beijing Mobile Companay, China, Beijing* 100007)

Abstract This paper presents an efficient way to simplify large-scale vector maps while keeping their original topology relationships. By using the frame buffer and Voronoi diagram, we achieve a rapid simplification of large-scale vector maps while avoiding topolopy relationship errors, including intersections, self-intersections, points' sidedness and polygons' adjacency changes. This study will help improve the accuracy and efficiency of multi-scale, large-scale vector data generalization.

Keywords vector map, multiresolution models, Simplification

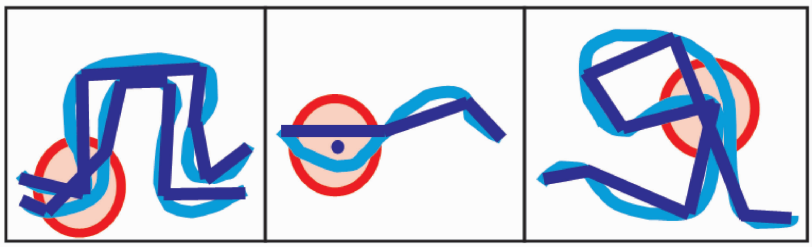
1 引 言

随着测绘、遥感及相关技术的发展,人们获取空间数据的数据量以摩尔定律的速度增长,远远超出了计算机内存的增长速度。对高分辨率、空间数据来说,通过简化融合,建立多细节层次模型(LOD)可加快可视化及网络传输速度。2 维矢量地图是遥感技术研究中存储、管理和分析的主要对象,并且有广泛的应用。本文在保持拓扑关系不变的前提下,研究了矢量地图的快速简化算法。该研究有助于提高多尺度、大尺度矢量数据融合的精度和效率,加快大规模矢量数据的可视化速度,并且在网络环境下,实现矢量数据的动态渐进传输。

对矢量地图的简化研究最早起于 1973 年,Douglas 和 Peuker 提出了 Douglas-Peuker 折线简化法[1],效率较高,且可保持线要素的重要几何特征。但在简化过程中会导致拓扑关系发生改变,造成简化后的线要素出现如图 1 所示的 3 种拓扑关系不一致[2]。对于面要素来说,若以多边形为基本单位,则简化后还可出现要素间邻接关系的改变。为避免上述现象,需在简化过程中加入一定的约束条件,如

收稿日期:2008-12-03;**改回日期**:2009-03-26

第一作者简介:杨 玲(1984~),女。现为北京师范大学地理学与遥感科学学院硕士研究生。主要研究方向为空间数据的 3 维显示和传输。E-mail: yangling0531@126.com



(a) 要素相交 (b) 点线位置关系 (c) 自相交

图 1 折线简化法可能导致的三种拓扑不一致现象[2]

Fig. 1 Incompatible topological relationships

Estkowski 证明了保持拓扑关系的复杂性,并提出了避免错误相交的算法[3],即先用 Douglas-Peuker 算法生成简化的结果,然后用“绕远之路(detour)”替换交叉的地方。Mustafa 等人提出了基于 Voronoi 图方法[4],利用了图形硬件初步获得了 Voronoi 图的计算和简化结果,避免了线要素之间错误的相交,但却不能避免自相交。Mantler 等人也利用了 Voronoi 图求得了折线的安全集,在安全集内应用 Douglas-Peuker 算法可避免自相交错误的出现[5],此算法计算 Voronoi 图的时间复杂度为 $O(n\log n)$,计算安全集的时间复杂度为 $O(n)$,加上 Douglas-Peuker 算法本身的时间复杂度 $O(n^2)$,使得对于大规模数据的简化耗时较长。Yang 等人利用基于三角形剖分的多分辨率模型的矢量数据简化和网络传输的算法[6],能够较好地实现矢量数量的简化和传输,但是,多分辨率模型的编码带有较大的数据冗余,影响了简化的效果。Bertolotto 构建了一个多比例尺数据和渐进传输的概念模型[7]。该模型仅仅考虑矢量数据在拓扑结构变化的情况,并没太多地考虑曲线数据的问题。吴焕萍利用 Delaunay 三角网,将空间剖分为包含曲线的互不重叠区域[8]。曲线化简位移的最大范围即为安全集合的范围。由此保证曲线综合的拓扑关系一致性。

本文主要利用并改进了 Mustafa 等人提出的方法[4],在快速简化的同时,有效地避免了拓扑不一致现象的产生。

2 大规模矢量地图简化算法

2.1 基于 Voronoi 图的简化方法

Douglas-Peuker 折线简化算法的基本思想:若线要素 c' 是线要素 c 的简化结果,则 c' 距 c 的距离应小于 ε,此算法时间复杂度为 $O(n^2)$。基于此连接 c 上不相邻两点 (v_i, v_j) 生成一条捷径 $\overline{v_iv_j}$,若 $\overline{v_iv_j}$ 与原线段组 $(\overline{v_iv_{i+1}}, \cdots, \overline{v_{j-1}v_j})$ 的距离小于 ε,则认为此捷径和 c 接近,可作为简化结果。捷径可分为 3 类,第 1 类与 c 接近,第 2 类与 c 不接近,但是位于 c 的宽为 ε 的缓冲区内的,第 3 类位于 c 的缓冲区外。首先利用帧缓存和 Voronoi 图,剔除第 3 类捷径,使得剩余捷径数目远小于初始时的数目;然后检查剩余的捷径距原线段组的距离,以剔除第 2 类捷径;最后由第 1 类捷径生成简化结果 c'。

在 Douglas-Peuker 简化算法和 Mustafa 等人工作的基础上,提出如下基于 Voronoi 图的简化方法。首先清空各个帧缓存,使其值均为 0。在模板缓存中渲染整幅地图的宽为 ε 的 Voronoi 图,每个线要素的 Voronoi 区域的模板缓存值与要素序列号 i 一一对应。连接要素 c_i 上不相邻的两顶点,在颜色缓存中渲染此捷径。渲染时开启模板检测,要求模板缓存值不等于 i 的部分才能渲染到颜色缓存中,并且每条捷径与一种颜色一一对应。定义所有第 3 类捷径组成集合为 $\boldsymbol{S}_i$,初始时,$\boldsymbol{S}_i$ 包括所有捷径。渲染完所有要素的所有捷径后,逐像素检查颜色缓存,若一像素的 RGBA 值不全为 0,则将与此颜色对应的捷径从 $\boldsymbol{S}_i$ 中剔除。颜色缓存检查完毕后,逐一计算 $\boldsymbol{S}_i$ 中剩余的捷径和其对应原线段组的距离,若大于 ε,则也从 $\boldsymbol{S}_i$ 剔除。最后按照最短路径,将 $\boldsymbol{S}_i$ 中剩余捷径相连,即得到简化结果 c'_i。对于面要素,可将其看作线要素加以简化。形成封闭的线段组。

因模板缓存最大值为 256,相应的要素个数不能大于 256。假设颜色缓存有 RGBA 4 个通道,各占 8 bits,则要求渲染的捷径总数不大于 $2^{32}-1$。若不满足这两个条件,可对原数据分批次简化。

此方法与几何方法相比,耗时较小,简单而有效地保持了拓扑关系。

2.2　保持拓扑关系一致性

上述过程利用了 Voronoi 图,可避免简化后矢量要素间错误的相交。如图 2 所示,红色线段$\overline{v_iv_{i+2}}$距线 c_1 的距离小于 ε,可作为原线段组的简化结果,这使得 c_1 简化后与 c_2 错误地相交。但若采用 Voronoi 图方法,$\overline{v_iv_{i+2}}$会经过 c_2 的 Voronoi 区域(图 2 中的绿色区域),则不会被选作简化结果。

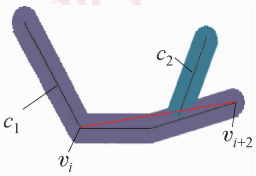


图 2　保持线要素之间的拓扑关系

Fig. 2　Avoid intersections between polylines

针对图 1(b)所示的位置改变现象,即线要素被简化后,位于线一侧的点 v 可能会错误地位于线的另一侧,Mustafa 等人在简化时增加了约束条件,即判断简化后的线与原来线所围成的范围内是否包括点 v,若包括,则表明点 v 和线的位置关系在简化后会被改变,此时需修改简化结果。解决此问题可避开上述几何算法,而可利用 Voronoi 图快速实现。具体地,在生成线要素的 Voronoi 图后,生成点要素宽度为 ε 的缓冲区,并且在渲染其缓冲区时,减少其深度值,使其在模板缓存中覆盖原 Voronoi 图,得到完全渲染。则经过点缓冲区的捷径将从 $\boldsymbol{S}_i$ 剔除。如图 3 所示,捷径$\overline{v_iv_{i+2}}$部分超出线 c_1 的 Voronoi 区域,而位于点的缓冲区内,因此不能作为简化结果。图 4 为该方法的实验结果。对于图 1(c)所示错误,线要素在简化后产生自相交,主要是因为线要素生成的 Voronoi 图自相交(图 5 所示,黄色部分为 Voronoi 图自相交区域)。

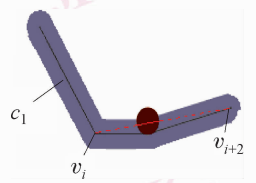
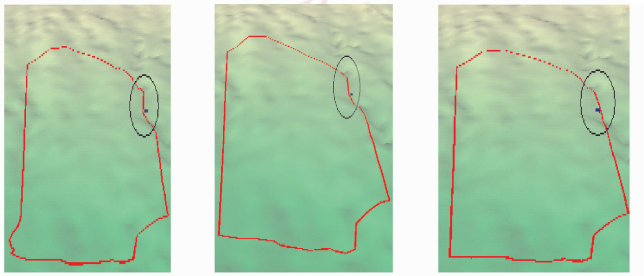


图 3　保持线要素和点要素之间的拓扑关系

Fig. 3　Keep points' sidedness

本文提出利用模板缓存去除此类区域。先求得由一组四边形组成的线要素的缓冲区,宽度为 ε(如图 6 所示)。在生成矢量地图的 Voronoi 图后,关闭深度缓存的写操作,在此基础上渲染每个要素的缓冲区。图 7 为利用线缓冲区避免要素自相交的实验结果,假设矢量要素 c_i 对应的模板缓存值为 i,则渲染缓冲区时,要求模板缓存值大于或等于 i 的地方



(a) 未简化的线要素　(b) 渲染点要素的 Voronoi图后的简化结果　(c) 未加约束的简化结果

图 4　利用 Voronoi 图保持线点间拓扑关系

Fig. 4　Keeping points' sidedness by using ε-Voronoi diagram.

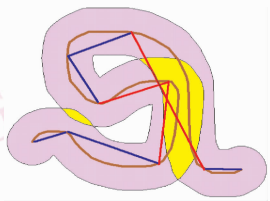


图 5　线拓扑关系保持

Fig. 5　Self-overlapped ε-Voronoi regions may cause self-intersections

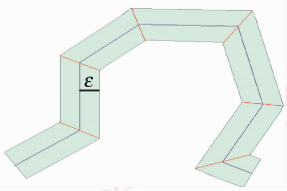
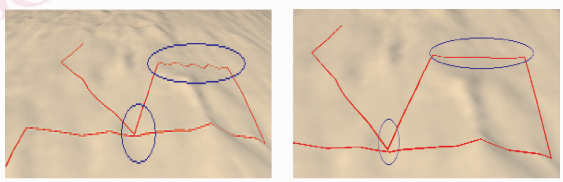


图 6　生成线缓冲区

Fig. 6　The ε-buffer of a polyline



(a) 原始线要素　(b) 简化结果

图 7　利用线缓冲区避免要素自相交

Fig. 7　The simplified result which avoids self-intersections by using the ε-buffer of the polylines

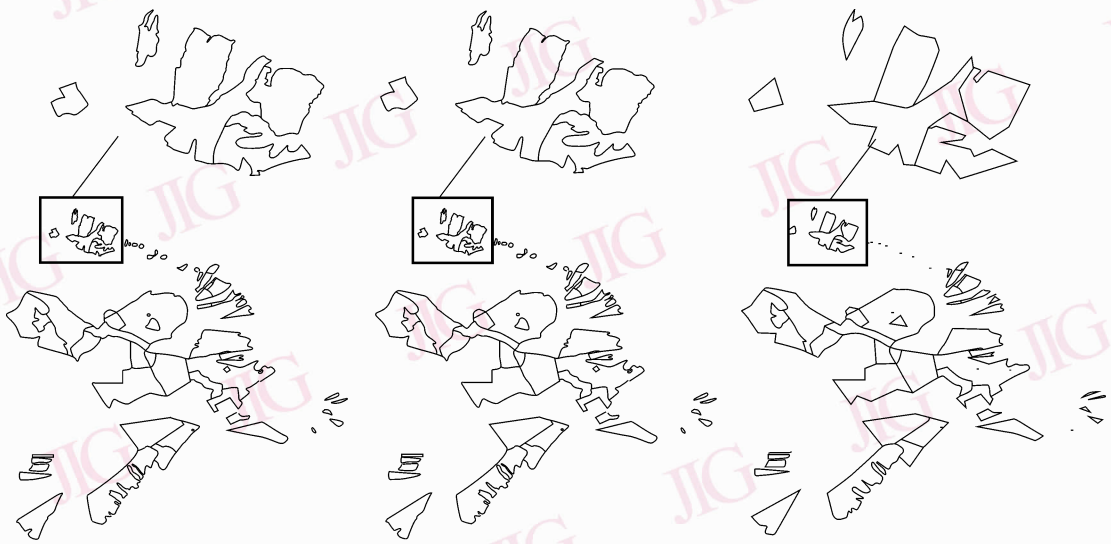
其缓存值加 1,其他地方值不变。故渲染的结果为该要素的 Voronoi 图中没有相交区域的模板缓存值为 $i+1$,相交的区域值大于 $i+1$。此后在渲染捷径时,要求模板值不为 $i+1$ 的地方才可通过检测得到渲染。此外,还有一种拓扑关系需要保持,即两个多边形共用一条边时,若简化以多边形为基本单位,则此公共边将被简化两次,两次简化结果可能不同,使得这两个多边形边界不再完全重合。为消除这种错误,对于可以得知邻接关系的面状数据,简化时应以组成多边形的弧段为基本单位。

3 实 验

为了证明本文方法的可行性,进行实验验证。机器配置为:CPU 为 Intel(R) Pentium(R) 4,速度为 3 993 MHz,内存大小为 768 MB,显卡型号为 Intel(R) 82865G Graphics controller。开发语言为VC++.NET 和 OpneGL。矢量地图为表示夏威夷岛的狩猎区,包含 63 个面要素,25 220 个顶点。通过 ArcMAp 建立拓扑关系后,面要素被转化为 132 个弧段,简化时以弧段为基本单位。

图 8 显示了简化结果。第一次简化时,设缓冲区宽度 ε 为 20 m,简化后保留了 3 824 个顶点,简化时间约为2 s;第二次简化缓冲区宽度 ε 这 100 m,简化程度约为 4% ,简化时间约为 3.5 s。

此实验结果可以得出,在保持简化前后拓扑关系一致和满足数据精度的前提下,本文方法能够快速地对矢量数据进行简化,同时也能够按照简化过程的逆变换实现数据的快速重建,较好地实现了大规模量矢量地图数据的综合。



(a) 原始(25 220顶点) (b) 简化(3 824顶点) (c) 简化(1 183顶点)

图 8 矢量图简化

Fig. 8 Vector maps simplification

4 结 论

本文主要研究了大数据量矢量地图简化方法。借助图形硬件和 Voronoi 图实现了矢量数据简化,并保持了正确的拓扑关系。提高了大数据量矢量地图的动态简化速度。本文的研究有助于提高多尺度、大尺度矢量数据融合的精度和效率。

下一步工作主要是在本文算法的基础上,研究大规模矢量地图的网络传输以及多尺度矢量数据融合。

参考文献(References)

1 Douglas D H, Peucker T K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature[J]. The Canadian Cartographer, 1973, **10**(2): 112-122.

2 Zhan H S, Li G X. Progressive transmission of vector map data based on polygonal chain simplification[J]. Lecture Notes in Computer Science, 2006, **4282**: 908-917.

3 Estkowsk R, Mitchell J S B. Simplifying a polygonal subdivision while keeping it simple[A]. In: Proceedings of the 17th ACM

Symposium on Computational Geometry[C], Boston, Massachusetts, USA, 2001:40-49.

4 Mustafa N, Krishnan N S G, Varadhan S. Dynamic simplification and visualization of large maps [J]. International Journal of Geographical Information Science, 2006, **20**(3):273-320.

5 Mantler A, Snoeyink J. Safe sets for line simplification [R]. In: Abstracts of the 10th Annual Fall Workshop on Computational Geometry[C], Stony Brook, New York, USA, 2000.

6 Yang B S, Purves R, Weibel R. Efficient transmission of vector data over the internet [J]. International Journal of Geographical Information Science, 2007, **21**(2):215-237.

7 Bertolotto M, Egenhofer M J. Progressive transmission of vector map data over the World Wide Web[J]. GeoInformatica, 2001, **5**(4): 345-373.

8 Hu Huan-pin, Pan Mao, Zhang Chuan-ming, *et al*. Progressive transmission of GIS vector spatial data over Internet[J]. High Technology Letters. 2006, **16**(3): 290-295. [吴焕萍,潘懋,张传明等. GIS 矢量数据的网络渐近传输研究[J]. 高技术通讯. 2006, **16**(3):290-295.]