

# 线光滑算法效率优化方法研究

严 浩<sup>1</sup> 沈 婕<sup>1 2 3</sup> 朱 伟<sup>1 2 3</sup>

(1. 南京师范大学 地理科学学院 江苏 南京 210046)

(2. 南京师范大学 虚拟地理环境教育部重点实验室 江苏 南京 210046)

(3. 南京师范大学 地理信息科学江苏省重点实验室 江苏 南京 210046)

**[摘要]** 线光滑算法是地图综合算法的重要组成部分, 现今发达的网络环境和大量的移动应用对其时效性提出了更高的要求, 但线光滑算法的效率评估和优化还缺乏理论基础和实验验证. 论文对线光滑算法进行了整理和分类, 选取了线性迭代法、正轴抛物线加权平均法、斜轴抛物线加权平均法、五点求导分段三次多项式法、三点求导分段三次多项式法和张力样条函数法等方法进行原理分析并实现. 采用代码优化的方法对正轴抛物线加权平均法进行了优化, 使其效率得到了一定提升.

**[关键词]** 线光滑算法 复杂度分析 效率优化

**[中图分类号]** P208 **[文献标志码]** A **[文章编号]** 1672-1292(2012)02-0087-06

## Research on Efficiency Optimization of Curve Smoothing Algorithms

Yan Hao<sup>1</sup>, Shen Jie<sup>1 2 3</sup>, Zhu Wei<sup>1 2 3</sup>

(1. School of Geography Science, Nanjing Normal University, Nanjing 210046, China)

(2. MOE Key Laboratories of Virtual Geographic Environment, Nanjing Normal University, Nanjing 210046, China)

(3. Key Laboratory of Geographic Information Science of Jiangsu Province, Nanjing Normal University, Nanjing 210046, China)

**Abstract:** Curve smoothing algorithms are significant parts of map generalization algorithms. Now the developed network environment and massive mobile services need higher-efficiency curve smoothing algorithms, but the evaluation and optimization of these algorithms lack theoretical basis and experimental proving. This paper summarized and classified 6 curve smoothing algorithms like the positive axial parabola interpolation method, the linear iterative method, the five-point method, the three-point method, the skew axial parabola interpolation method and tension spline interpolation. By making complexity analysis and efficiency evaluation, the positive axial parabola interpolation method was optimized with code optimization and its efficiency was improved.

**Key words:** curve smoothing algorithm, complexity analysis, efficiency optimization

地图综合是地图学的核心问题之一. 在网络和移动环境日益成熟的今天, 很多地图服务和应用对地图数据传输和处理的效率提出了很高的要求, 其中, 地图综合的线光滑算法的效率问题尤其突出. 目前已有的一些常用的线光滑理论, 如正轴抛物线加权平均法、斜轴抛物线加权平均法<sup>[1-3]</sup>、五点求导分段三次多项式法<sup>[4]</sup>、三点求导分段三次多项式法、张力样条函数法<sup>[5]</sup>, 以及相应的改进算法等<sup>[6-11]</sup>.

上述线光滑算法研究虽然理论较为成熟, 但仍缺乏对其算法效率的系统性评估和改进. 本文拟在线光滑算法效率方面进行研究, 首先对线光滑算法进行了分类, 选取了上述几个典型的算法进行详细的效率分析, 在此基础上提出了一些效率优化和改进的方法.

## 1 线光滑算法及其分类

### 1.1 线光滑算法的原理、效率分析和实现

本文选取了 6 种应用最广泛的线光滑算法进行原理分析和编程实现, 并尝试找出其中影响效率的

收稿日期: 2012-04-10.

基金项目: 国家自然科学基金(41071288, 41171350).

通讯联系人: 沈 婕, 博士, 副教授, 研究方向: 地图自动综合并行计算、电子地图与网络地图设计. E-mail: shenjie@njnu.edu.cn

部分.

(1) 线性迭代法: 基本思想是按照数据点的顺序依次取3个点,组成两个线段,在前后两个线段分别距中间点四分之一距离处取两个点,插入原始数据中,并删除原中间点,从而使得较尖锐的角被抹平.其效率提高的方式主要在于对存储点的数据结构的优化.

(2) 正轴抛物线加权平均法: 基本思想是按照数据点的顺序依次取4个点,前3个点拟合一条抛物线,后3个点拟合另一条抛物线,中间两点采用两条抛物线加权平均的方法生成一条新的曲线.该算法影响效率的主要因素是点数组的存取方式和每次循环中的存取效率,数组是最主要的需要优化的效率影响因素.

(3) 斜轴抛物线加权平均法: 基本思想与(2)类似,只是在计算抛物线之前需要进行坐标变换,以保证抛物线的最大曲率点落在数据点上.

(4) 五点求导分段三次多项式法: 基本思想是在相邻两点间确定一条三次多项式曲线,通过使该曲线具有连续的一阶导数来保证其光滑性,而每个点的导数满足AKIMA条件,由前后共5个点来确定.对算法影响最大的依然是点数据的存取方式,减少访问次数和提高循环效率是对算法效率提升的最好方式.

(5) 三点求导分段三次多项式法: 基本原理和(4)类似,其通过前后3个点来确定中间点的导数值.

(6) 张力样条函数法: 其构成是一个直线插值函数和两个双曲函数的线性组合,主要是求出张力系数,从而计算出每点的二阶导数.这部分是复杂的函数和矩阵计算,求解出张力系数后利用另一个循环,通过张力样条函数绘制出两点间的曲线.

6种算法的原理图和实现效果如表1所示.

线光滑算法的时间复杂度都比较低.通过对算法的分析,上述6种算法时间复杂度均为 $O(n)$ .针对相同的数据,算法执行时间大部分依赖于算法本身的计算原理和时间复杂度中不依赖于数据规模的常数级部分,因此优化也主要针对这部分进行.

## 1.2 线光滑算法分类

从数据拟合精度的角度来看,线光滑算法分为严格通过原始数据点和不严格通过原始数据点两类,具体算法分类如表2所示.严格通过原始数据点的光滑方法更符合地图综合质量要求,因此实际应用较多.在注重显示效果而对数据准确性要求不太严格的应用中,则可采用后者的方法.按照这种分类方式,线光滑算法分类如表2所示.

表2 基于拟合精度的线光滑算法分类

Table 2 Classification of curve smoothing algorithms based on fitting accuracy

分类	算法
严格通过原始数据点	正轴抛物线加权平均法,斜轴抛物线加权平均法,五点求导分段三次多项式法,三点求导分段三次多项式法...
不严格通过原始数据点	线性迭代法,张力样条函数法...

从数据量的角度来看,不同的算法各有优劣.有一些算法在不同数据量级的情况下可以通用,如线性迭代法<sup>[12,13]</sup>,该方法对数据本身的大小不敏感.还有部分算法对不同数据量的数据处理后效果会有差异,例如五点求导分段三次多项式法比较适合稠密的数据,正轴抛物线加权平均法<sup>[14]</sup>、三点求导分段三次多项式法等则适合较稀疏的数据<sup>[4]</sup>.对于数据量的敏感性大部分是由算法本身的特性所决定的,数据形式有很多,因此这种分类的角度不具有通用性.

线光滑算法的分类可以有很多种,在具体应用中,应根据数据特点和算法的特点,选取一个在显示效

果和计算效率上达到相对平衡的算法。

## 2 线光滑算法效率优化方法探索

### 2.1 效率优化概述

通过某种方法用更小的时间和空间开销获得相同的运行结果,都可称之为优化。算法的效率优化有多种方式,由于计算机系统的层次性,优化也是分层次的。具体可以分为 3 个层次或级别:系统级优化、算法级优化和代码级优化。其优化特点如表 3 所示。

表 3 算法优化级别及其优化特点  
Table 3 Levels of algorithm optimization

优化等级	优化特点
系统级优化	控制流和数据流优化为主,通过减少消息传递个数、平衡系统负载和发挥硬件最大性能等方式来进行优化
算法级优化	以算法选择、算法自身的时间空间优化为主,还有并行化和数据结构优化等
代码级优化	代码优化为主,主要是缓存相关的优化,通过函数调用方法、编译优化选项等来进行优化

### 2.2 线光滑算法代码级优化方法

本文主要选用代码优化方法,该方法比较快捷,但对编程语言和计算环境有较大依赖,且具有一定的主观性。代码级的效率优化主要是在程序的关键路径上进行优化,针对不同的数据类型和处理方法可以有很多方式,本文主要研究使用提前计算和循环展开优化技术<sup>[15]</sup>,并分析实验结果。

#### 2.2.1 提前计算

提前计算指的是把程序中需要多次用到的函数或者访址操作提前计算并保存到一个变量中,在需要使用时直接使用这个变量,从而避免了多次重复的复杂操作。例如,正轴抛物线加权平均法需要计算累积弦长作为抛物线的参数,前后两段抛物线有一段重合的弦长,且在计算弦长时调用的存取 shape 文件的数据结构比较复杂,因此选择这段代码提前计算。未优化代码如下:

```
/* 类调用层次为 shape 格式数据结构、第 m 条等高线、线结构、点结构和点的坐标值* /
stringA2 = stringA1 + sqrt( ( Shape. Contours [m]. Line. Points [n + 1]. x - Shape. Contours [m]. Line.
Points [n]. x) ...;
stringA3 = stringA2 + sqrt( ( Shape. Contours [m]. Line. Points [n + 2]. x - Shape. Contours [m]. Line.
Points [n + 1]. x) ...;
stringB2 = stringB1 + sqrt( ( Shape. Contours [m]. Line. Points [n + 2]. x - Shape. Contours [m]. Line.
Points [n + 1]. x) ...;
```

上述代码是分别计算前后两段抛物线的累加弦长,由于计算较繁琐,存取数据的数据结构层次也较多,因此在索引上会产生很大的开销,采用提前计算的方法将 3 段弦长提前计算,前两段分别命名为  $L1$  和  $L2$ ,代码可以简化为:

```
stringA2 = stringA1 + L1;
stringA3 = stringA2 + L2;
stringB2 = stringB1 + L2;
```

这样可以减少计算过程中对数组的多次索引,同时可以避免中间一段共享线段的重复计算。

#### 2.2.2 循环展开

循环展开指的是通过增加每次循环处理的数据量,从而减少总循环次数和条件测试次数的一种方法。把单次循环的代码改成如下形式,步长取为 3:

```
for( ; i < stringA2; i += 6)
{
    point1. x = A1 + B1* i + C1* i* i;
    point1. y = A2 + B2* i + C2* i* i;
    point2. x = A1 + B1* ( i + 3) + C1* ( i + 3) * ( i + 3);
```

```

point2.y = A2 + B2* ( i + 3) + C2* ( i + 3) * ( i + 3);
DrawLine( point1 ,point2);
}

```

其中 point1、point2 为两个存放临时点的 POINT 结构  $i$  为判断插值范围的变量,在循环外已定义和初始化  $A1、B1、C1$  和  $A2、B2、C2$  分别为前后两条抛物线的参数,DrawLine() 方法用于在屏幕上绘制线.该方法把操作对象线性化,且在一次迭代中访问线性数据中的一小组而非单独的某个,这样得到的程序将执行更少的迭代次数,于是循环开销就被有效地降低了.

### 2.2.3 其他优化方法

除上述方法外,还有很多代码优化的方法,如代码移动,原始代码如下:

```

/* 类调用层次为 shape 格式数据结构、第 m 条等高线、线结构、点结构和点结构的数据量 */
for ( int n=0; n < ( Shape. Contours [m]. Line. Points. size() - 3); n++ ) { ... }
将需要重复计算的部分移出循环条件测试语句,用一个变量代替:
int length = Shape. Contours [m]. Line. Points. size() - 3;
for( int n=0; n < length; n++ ) { ... }

```

这样就节省了多次测试条件的计算,该方法类似于提前计算方法.此外,还有提高指令级并行性、使用 inline 函数和局部变量等方法.

### 2.2.4 实验结果

本文选取正轴抛物线加权平均法作为测试算法.实验数据采用的是庐山 1:10 000 等高线数据,实验区点的数据量在 20 000 个左右.测试机的操作系统为 Windows 7 SP1,CPU 为 Intel Core2 DUO 双核,处理器频率 2.20 GHz,内存为 2 GB,一级数据缓存为 64 kB,一级指令缓存为 64 kB,二级缓存为 2 MB.IDE 使用的是微软的 Visual Studio 2010,建立 MFC 项目进行算法测试.

选取五点求导分段三次多项式法进行 CPU 监测,找出程序中的热点路径及不同函数对 CPU 的占用率,结果如图 1 和图 2 所示.可以看出,图形呈规律的波形,每隔固定周期有一个开销的峰值.这说明算法开销最大的部分是被重复计算的部分,即循环体.从 CPU 和资源占用率结果可以看出,算法部分的独占样本数百分比达到了 70.28%,屏幕显示占用了 15.94%,这两部分就占用了程序开销中 96% 的资源.CPU 占用率

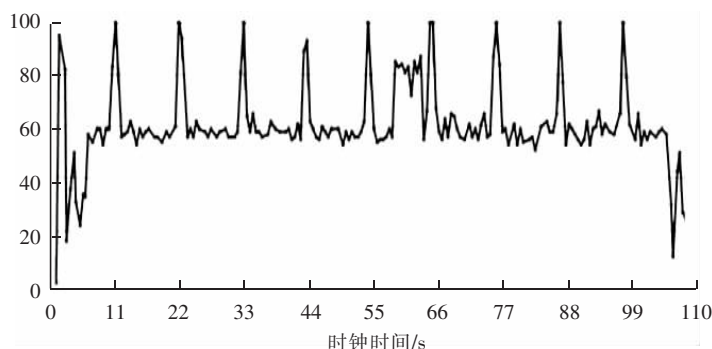


图1 随时间变化的 CPU 占用率

Fig.1 CPU occupancy rate by time

的周期性变化与算法中对数据的循环处理过程相符合,因此优化主要针对算法中这部分的循环体来进行.

热路径			执行单个工作最多的函数	
基于样本计数的开销最大的调用路径			采用独占样本最多的函数	
函数名	非独占样本百分比	独占样本百分比	名称	独占样本百分比
↗_wWinMainCRTStartup	100.00	0.00	CSAView::OnDraw(class CDC *)	70.28
↗_tmainCRTStartup	98.40	0.00	CDC::SetPixel(struct tagPOINT,unsigned long)	15.94
↗_wWinMain@16	98.40	0.00	未知框架	1.60
↗_AfxWinMain(struct HINSTANCE_,struct HINSTANCE_,_wchar_t*,int)	98.40	1.01	_ftol2_pentium4	1.26
⦿ CSAView::OnDraw(class CDC *)	93.30	70.28	[SA.exe]	1.14

图2 执行语句次数最多的函数

Fig.2 Fictions with the most statement

对点数据的量进行选取和划分,分别对优化程序和未优化程序进行时间对比,时间的单位为 s. 从结果可以看出,循环展开对程序的优化效果最大,测试时使用的展开度为 2,即减少了一半的语句判断开销. 提前计算和其他几种方法对性能的提升不大,可能是由于问题规模不够,所以其优势没有体现出来. 将优化前和优化后的性能—规模作图比较,如图 3 所示.

显然,随着数据规模的增大,优化后的速度更快,性能要优于优化前,用近似 CPE<sup>[15]</sup>(同复杂度下常数项随规模变化对算法时间的影响)的方法来进行量化,拟合出的性能函数的斜率优化前为 3.222 4,优化后为 3.033 4,性能有了小幅提升. 且从拟合数据来看(见图 4),该算法的性能曲线是近似线性的,因此也验证了其时间复杂度为  $O(n)$ .

从性能提升量的角度来看,性能只是呈提高趋势,提升量并不具有线性关系,但总体提升量稳定在 5% 到 15%,相比于原算法效率有了提高.

### 3 结语

本文对目前线光滑算法的类型进行了分类,对不同类型的算法的特点进行了归纳总结. 从线光滑算法效率评价的角度着手,选取并实现了若干线光滑算法,在此基础上对这些算法的复杂度进行了评估. 在没有改变算法复杂度的条件下,采用了几种代码优化的方法对程序进行优化并提高了程序的执行效率. 本文在这方面进行了初步的探索,整理了一种评价和提高算法效率的思路. 但是,本文只采用了代码优化方法对程序进行优化,效率的提高很有限,对线光滑算法的并行化以及算法级的优化还需要进一步探索和研究.

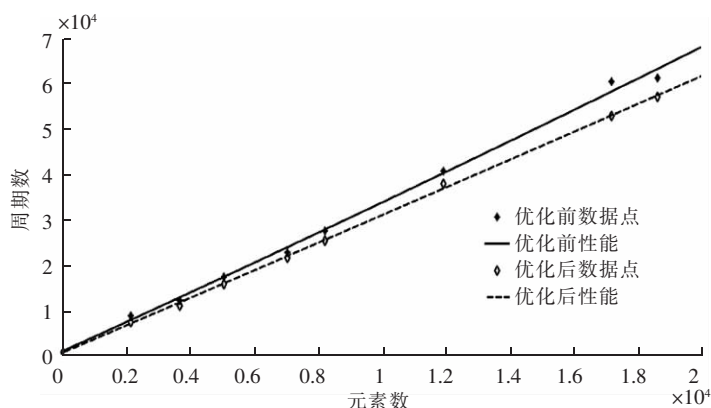


图 3 优化前后的性能—规模曲线

Fig.3 Performance-data volume curve of optimized algorithm and original algorithm

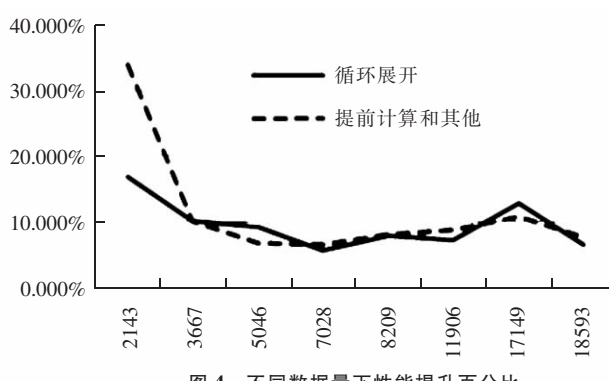


图 4 不同数据量下性能提升百分比

Fig.4 Percentages of performance promoting with different data sizes

### 【参考文献】(References)

- [1] 毋河海. 斜轴抛物线光滑插值[C]// 第三届全国地图学术会议论文集(上集). 北京: 测绘出版社, 1980.  
Wu Hehai. Curve smoothing interpolation by skew axial parabola[C]// The Third National Cartography Conference Papers Selection( Part 1). Beijing: Surveying and Mapping Press, 1980. ( in Chinese)
- [2] 潘正风, 罗年学, 黄全义. 近似斜轴抛物线加权平均插值法曲线光滑[J]. 测绘学报, 1991, 20( 1): 60-65.  
Pan Zhengfeng, Luo Nianxue, Huang Quanyi. Approximate interpolation by weighted average on skew axial parabola[J]. Acta Geodaetica et Cartographica Sinica, 1991, 20( 1): 60-65. ( in Chinese)
- [3] 李云锦, 钟耳顺, 黄跃峰. 斜轴抛物线插值的改进算法与近似算法[J]. 武汉大学学报: 信息科学版, 2009, 34( 12): 1 490-1 494.  
Li Yunjin, Zhong Ershun, Huang Yuefeng. An improved approximate interpolation by skew axial parabola[J]. Geomatics and Information Science of Wuhan University, 2009, 34( 12): 1 490-1 494. ( in Chinese)
- [4] 闫浩文. 计算机地图制图原理与方法[M]. 北京: 科学出版社, 2007.  
Yan Haowen. Principals and Methods of Computer-Aided Cartography[M]. Beijing: Science Press, 2007. ( in Chinese)

- [5] 迟文学,吴信才,于海洋,等. 张力样条函数在雨量等值线光滑中的应用研究[J]. 水文, 2007, 27(1): 22-25.  
Chi Wenxue, Wu Xincan, Yu Haiyang et al. Application of the tension spline function in rainfall isoline smoothing[J]. Journal of China Hydrology, 2007, 27(1): 22-25. (in Chinese)
- [6] Yu Kiyun. Performance improvement of bezier spline fitting for more accurate approximation of natural linear entities [J]. KSCE Journal of Civil Engineering, 1999(6): 181-193.
- [7] Liu Fang. New algorithm of complex curve spline interpolation[J]. Journal of Shanghai Jiaotong University, 2009(5): 834-836, 846.
- [8] Zeng Yong. A distance-based parameter free algorithm for curve reconstruction[J]. CAD Computer Aided Design, 2008(2): 210-222.
- [9] 张凤蛟. 快速曲线拟合的方法[J]. 延边大学学报: 自然科学版, 2006, 32(3): 208-211.  
Zhang Fengjiao. Fast curve fitting method[J]. Journal of Yanbian University: Natural Science Edition, 2006, 32(3): 208-211. (in Chinese)
- [10] 和正民. 关于计算机自动绘制光滑曲线的数学方法的探索[J]. 北京地质, 1998(1): 18-23.  
He Zhengmin. Discussion on the mathematical methods of curvesmoothing in automatic drawing with a computer[J]. Beijing Geology, 1998(1): 18-23. (in Chinese)
- [11] 邓曙光,李婉. 曲线光滑的张力样条插值法 VC 实现[J]. 工程地球物理学报, 2005, 2(5): 387-390.  
Deng Shuguang, Li Wan. Realization of smoothing curve with tension spline interpolation under visual C++ [J]. Chinese Journal of Engineering Geophysics, 2005, 2(5): 387-390. (in Chinese)
- [12] 王延亮,王明爽. 新型曲线光滑法——切线抹角法[J]. 测绘通报, 2005(3): 52-54.  
Wang Yanliang, Wang Mingshuang. The method of the new-type line smoothing—the tangent erased cape method[J]. Bulletin of Surveying and Mapping, 2005(3): 52-54. (in Chinese)
- [13] 文敏,唐新民,闫浩文,等. 一种改进型曲线光滑算法[J]. 测绘科学, 2011, 36(5): 181-183.  
Wen Min, Tang Xinmin, Yan Haowen et al. An innovatory algorithm for curve smoothing[J]. Science of Surveying and Mapping, 2011, 36(5): 181-183. (in Chinese)
- [14] De Boor C. High accuracy geometric hermite interpolation[J]. Computer Aided Geometric Design, 1987, 4(4): 269-278.
- [15] Randal E Bryant, David R O' Hallaron. 深入理解计算机系统[M]. 2版. 北京: 机械工业出版社, 2010.  
Randal E Bryant, David R O' Hallaron. Computer Systems—A Programmer's Perspective[M]. 2nd ed. Beijing: China Machine Press, 2010. (in Chinese)

[责任编辑: 严海琳]

(上接第86页)

- [10] 顾阿明,王彩会,徐玉琳,等. 苏锡常地区浅层地下水资源前景调查与开发利用示范[R]. 南京: 江苏省地质调查研究院, 2006.  
Gu Aming, Wang Caihui, Xu Yulin. Prospect investigation and development and exploitation demonstration of the shallow groundwater resource in Su-Xi-Chang area[R]. Nanjing: Institute of Geological Survey of Jiangsu Province, 2006. (in Chinese)
- [11] 张云,薛禹群. 一维非线性地面沉降模型参数敏感性分析[J]. 水文地质工程地质, 2005(3): 1-4.  
Zhang Yun, Xue Yuqun. Sensitivity of parameters in one-dimensional nonlinear model for land subsidence[J]. Hydrogeology and Engineering Geology, 2005(3): 1-4. (in Chinese)
- [12] 叶淑君,薛禹群. 应用沉降和水位数据计算上海地区弱透水层的参数[J]. 岩土力学, 2005, 26(2): 256-260.  
Ye Shujun, Xue Yuqun. Stress-strain analysis for storage coefficients and vertical hydraulic conductivities of aquitards in Shanghai area[J]. Rock and Soil Mechanics, 2005, 26(2): 256-260. (in Chinese)

[责任编辑: 严海琳]