

基于线积分卷积算法的并行实现方法

冯 冲, 颜廷华

(中国海洋大学 信息科学与工程学院 计算机系, 山东 青岛 266100)

摘 要:线积分卷积(LIC)是矢量场可视化中一个强有力的工具。但其计算量过大、耗时过多,影响了它的应用。根据 LIC 算法的特点,提出了 LIC 算法的并行实现方法。由于流线跟踪和卷积计算的独立性,只需把输出图像以像素点为单位平均分配给各处理器节点进行计算。从进程之间没有通信,仅当从进程开始计算前和计算结束后,在主进程与从进程之间有数据传递,通信开销很小。最后应用 MPI 在 Linux 集群环境下实现了该算法,实验结果表明,该方法具有较高的并行度和加速比。

关键词:线积分卷积;矢量场可视化;并行计算

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2008)12-0022-03

A Parallel Algorithm for Line Integral Convolution

FENG Chong, YAN Ting-hua

(Department of Computer, College of Information Science
and Engineering, Ocean University of China, Qingdao 266100, China)

Abstract: Line integral convolution (LIC) is a powerful technique for vector field visualization. But LIC demands much running time, what limit its utilization. In order to enhance computation efficiency, presents a parallel algorithm for LIC depending on characteristic of LIC. Because the streamline tracing and convolution computing of every point is independent, all pixels of the output image are distributed to every node for being computed. No message delivers among the slave processes, and only a little of data delivers between a master process and the slave ones. The tests on the vector field visualization show that parallel algorithm has a good performance on Linux cluster.

Key words: line integral convolution; vector field visualization; parallel computing

0 引 言

矢量场可视化是科学计算可视化中最具挑战性的研究课题之一,它将科学和工程计算等产生的大规模矢量数据转换为图形、图像,并以直观的形式表示出来,使人们以直观形象的方式解释理解抽象科学数据中包含的客观规律,具有广泛的发展前景和应用领域。传统的矢量场可视化方法主要包括基于几何形状的矢量场映射方法,如箭头或其它图标、流线、流面等,以及许多不同的粒子跟踪方法。然而,图标法无法揭示数据内在的连续性,并且空间利用率较低,只适用于小规模矢量场。流线和粒子跟踪方法又完全依赖于源点的选择,否则会丢失矢量场的重要信息。

基于纹理的矢量场可视化方法解决了这些问题,以图像形式显示矢量场的全貌,可以表现细节变化,具有很大的优越性。Van Wijk^[1]在 1991 年的 SIGGRAPH 会议上提出基于纹理的点噪声(Spot Noise)沿着一条与局部矢量场方向相同的直线段对白噪声进行卷积来产生纹理,最早将纹理映射技术引入到矢量场可视化,但这种方法在矢量方向变化大的区域,很难准确表示出矢量的方向,甚至丢失重要信息。Brian Cabral 和 Leith Leedom^[2]在 1993 年的 SIGGRAPH 会议上提出线积分卷积(LIC: Line Integral Convolution),LIC 用滤波器沿流线卷积白噪声图像,它很好地可视化了流线的方向性,反映了整个矢量场的结构,克服了箭头、流线等可视化图像的混乱现象,对矢量场可视化具有极其重要的意义,但该方法最大的缺点就是耗时较长。于是多年来人们提出了很多基于 LIC 的改进算法,并取得了不错的效果,但是对该算法的并行实现方法的研究很少。

笔者论述了 LIC 的并行性,提出了 LIC 的并行实现算法,大大提高了计算速度。

收稿日期:2008-03-12

基金项目:山东省科技厅资助项目(2007BS01002)

作者简介:冯 冲(1982-),男,山东枣庄人,硕士研究生,研究方向为计算机图形图像及科学计算可视化;导师:魏志强,教授,博士生导师,研究方向为移动计算、图像处理与计算机视觉、智能机器人技术。

1 LIC 算法描述

LIC 算法是由 Cabral 和 Leedom^[2]提出的。用卷积来表示矢量场的方向源于一种运动模糊的思想。卷积是图像处理常用的方法,给定输入图像,在输出图像中每一个像素是输入图像中一个小区域中像素的加权平均,其中权值由一个函数定义,这个函数称为卷积核。

LIC 使用一维低通滤波器函数 $k(\omega)$ 作为卷积核沿流线卷积白噪声图像,合成纹理。输出纹理的每个像素值均通过积分卷积得到:首先基于该像素沿矢量正、反方向对称积分得到流线,将流线上所有像素对应的输入噪声值按卷积核参与卷积,结果作为输出纹理的像素值。输入的白噪声图像各点不相关,而 LIC 纹理在流线方向上相关,显示了矢量场的方向信息,假定流线用 $\sigma(s)$ 表示, s 是弧长参数,LIC 纹理在点 $x_0 = \sigma(s)$ 的强度为 $I(x_0)$,于是有

$$I(x_0) = \frac{1}{Z} \int_{s_0-L}^{s_0+L} k(s-s_0) T(\sigma(s)) ds \quad (1)$$

式中 $Z = \int_{-L}^L k(s) ds$ 为归一化参数, T 为白噪声, k 为滤波器,滤波器长为 L 。对输出图像的每个像素点,LIC 都要计算流线,然后用式(1)计算,得到该点的纹理强度。

这里用离散的思想描述 LIC 算法中粒子的轨迹,首先定义如下的速度矢量场^[3]:

$$V(x(t), t) = [v_x([x, y], t), v_y([x, y], t)] \quad (2)$$

其中 $V(x(t), t)$ 在实数范围内定义,则在一个时间段 $[t_k, t_{k+1}]$ 内,粒子轨迹可以表示为:

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} V(x(s), s) ds \quad (3)$$

有很多方法可以解出式(3),在这里采用精度最高的变长的四阶龙格-库塔方程求解。采用盒式卷积核函数计算,式(1)的离散形式为:

$$I(x_0) = \frac{1}{2n+1} \left(\sum_{i=-n}^n T(x_i) \right) \quad (4)$$

其中 $T(x_i)$ 为流线上各离散点噪声纹理值, n 为流线上的离散点个数。

2 LIC 算法的并行设计

近些年随着并行计算机的迅速发展,并行计算已经成为在更短的时间内解决相同问题的最重要的一种手段。要对 LIC 算法进行并行计算,必须首先对 LIC 算法进行并行性分析。

2.1 LIC 算法可视化处理的基本步骤

LIC 算法可视化处理的基本步骤:

1. 输入矢量场数据,利用矢量场生成一幅纹理图

像。

2. 对输出图像的每个像素点依次进行以下两步:

① 采用变长的四阶龙格-库塔方程进行流线跟踪;

② 将输入的纹理沿着流线进行卷积,得到该点的像素值。

3. 输出图像。

2.2 LIC 并行化的关键问题

整个算法的绝大部分计算量都集中在第 2 步上,该算法之所以耗时,主要是因为第 2 步的流线跟踪计算量极大。所以只要把第 2 步进行并行计算就会大大提高整个算法的速度。要把第 2 步进行并行计算,需重点考虑两个问题:一是计算任务的分配;二是通信问题。

1) 计算任务的分配。由于各像素点的流线跟踪和卷积计算都是相互独立的,这样就可以把 LIC 算法可视化的问题转化成非常充分的并行处理问题。即先分别独立地逐个点进行流线跟踪地计算,再利用式(4)快速地获得该点的像素值。所以,在理论上,输出图像有多少个像素点,就可以使用多少个节点进行独立计算。当然,节点数一般是远远少于像素点个数,所以需要根据节点数对整个计算任务进行分解即负载均衡^[4],就是尽可能保证每个节点负载均匀,每个节点都可以处理一部分负载,并且可以在节点之间动态分配负载,以实现平衡。对于一个大小为 $N \times M$ (像素) 的图像,设处理器节点数为 P ,如果将图像均匀地分至每个处理器节点中,则每个处理器节点的子图像大小为 $N \times M/p$,即每个节点处理的像素点个数为 $N \times M/p$ 。

2) 采用主从通信方式^[5]。首先各个从进程使用的数据和进行的计算完全是独立的,所以从进程之间不存在数据的传递,没有通信开销。由于每个节点进行流线跟踪和卷积计算时要用到整个矢量场数据和纹理图像,这就需要主进程把这些数据发给从进程,各个进程计算完以后,也要把结果发给主进程,这是主从进程之间,也是整个算法的主要通信量。

2.3 具体的并行算法

1) 指定任意的处理器节点(设标识为 0),从文件中读取矢量场数据,并生成纹理图像,然后把矢量数据和纹理像素值广播给所有处理器节点。

2) 各处理器节点根据矢量数据,对其划分的像素点逐个进行流线跟踪,然后根据纹理像素值用式(4)计算出该点的像素值。

3) 将各个节点运算的结果返回给处理器节点 0,然后处理器节点 0 根据各点的像素值输出图像。

3 实验结果与分析

为了对所提出的 LIC 并行算法的有效性进行验证,在 20 个节点的曙光 TC4000L 集群及 Linux 操作系统的实验环境和 MPI 并行编程环境下对渤海流场矢量数据进行了实验,积分长度为 100,输出图像大小为 562×376(如图 1 所示),并行实验结果如表 1 所示。

表 1 并行实验结果

节点数	计算时间(秒)	并行加速比	并行效率(%)
1	267.5	1.00	100
4	88.9	3.01	75
8	46.2	5.79	72
16	24.6	10.88	68
20	22.1	12.12	61

4 结束语

在矢量场可视化领域,LIC 算法具有很大的优越性,但其计算量过大,导致耗时过多,制约了它的实际应用,并行计算是解决该问题的有效途径。提出了 LIC 算法的并行实现算法,并运用到海洋流场可视化当中,实验结果表明,该算法设计合理,具有较高的并行度和加速比,使 LIC 算法的交互式探索成为可能,具有一定的实际意义。

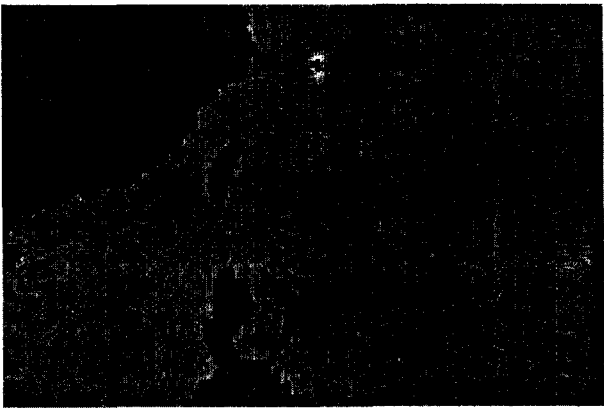


图 1 渤海流场的 LIC 纹理图

参考文献:

[1] Van Wijk J J. Spot noise: texture synthesis for data visualization[J]. Computer Graphics,1991,25(4):309-318.
[2] Cabral B,Leedom L. Imaging vector fields using line integral convolution[J]. Computer Graphics,1993,27(4):263-270.
[3] 余永胜,顾耀林. 一个基于 LIC 的矢量场可视化算法的改进[J]. 工程图学学报,2006(5):34-37.
[4] 陈国良. 并行计算-结构 算法 编程(修订版)[M]. 北京:高等教育出版社,2003.
[5] 都志辉. 高性能计算并行编程技术-MPI 并行程序设计[M]. 北京:清华大学出版社,2001.

(上接第 21 页)

ξ,当存在多个相同的最高测试率时,取支持向量少的那组参数作为最优参数。窗口长度 4096 个采样点。对每种音乐类型提取特征向量,使用不同的子带和子段的数目进行测试。实验中使用 6 种组合类型形式用 SVM 分类。实验结果见表 1。整个实验在 Visual C++ 6.0 平台下开发完成。

表 1 不同音乐类型间组合的分类错误率

子段数	子带数	错误率%					
		古典/流行	古典/摇摆	古典/爵士	爵士/流行	爵士/摇摆	摇摆/流行
1	16	16.3	6.7	22.3	26.7	14.1	19
1	32	16.9	6.7	20	26.3	11.9	14.1
2	64	17.5	7.0	21.0	26.4	10.3	16.2
4	64	19.6	7.2	19.6	26.6	9.4	17.2

4 结束语

音乐类型分类在多媒体应用中有重要的应用价值。文中采用 Morlet 小波变换作为音乐特征,SVM 分类音乐类型,分类正确率可达 85%,尤其对古典/摇摆这对类型分类表现最好。这表明该方法合理,分类性能较好。

参考文献:

[1] Burred J, Lerch A. A Hierarchical approach to automatic musical genre classification[C]//Int. Conf. on Digital Audio Effects (DAFx-03). London, UK:[s.n.], 2003:308-311.
[2] Li T, Oginara M, Li Q. A comparative study on content-based music genre classification[C]//in Proc. of the 26th annual int. ACM SIGIR conf. on Research and development in information retrieval. [s.l.]:ACM Press, 2003:282-289.
[3] Tzanetakis G, Cook P. Musical genre classification of audio signals[J]. IEEE Trans. on Speech and Audio Processing, 2002,10(5):293-302.
[4] Mallat S G. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation[J]. IEEE Trans Pattern Analysis and Machine Intelligence, 1989,11(7):674-693.
[5] Downie T R, Silverman B W. The discrete multiple wavelet transform and thresholding methods[J]. IEEE Trans on Signal Processing,1998,46(9):2558-2561.
[6] 王欢良,韩纪庆,张 磊.基于支持向量机的变异语音分类[J]. 哈尔滨工业大学学报,2003,35(4):389-393.
[7] Platt J C. Fast training of SVMs using sequential minimal optimization[C]//Scholkopf B, Burges C J C, Smola A J. Advances in Kernel Methods-Support Vector Learning. Cambridge, MA:MIT Press,1998.