

并行算法在三维景观可视化中的应用初探^{*}

Parallelism of Building 3D-Landscape Model

黄国满 何建邦

(中国科学院地理科学与资源研究所 资源与环境信息系统国家重点实验室 北京100101)

Abstract This paper studies the way to improve the speed of building 3D-landscape model with parallel algorithm. The author designs parallel algorithms for perspective transformation of DEM (with image), and a simulation is completed on series machine HP workstation and good results have been achieved.

Keywords 3D Landscape, Visualization, Parallelism

1 概述

当代高技术研究的许多领域都对计算机的性能提出了巨大的挑战,大型科学计算问题的求解,特别是科学计算可视化,要求计算机具有极快的运算速度^[1,2]。计算机图像处理更是经常需要对海量数据进行实时处理,对计算机的计算速度要求极高^[3-5]。传统的计算机是串行工作的,由于电脉冲传播速度以光速为极限,受到体积和散热等技术条件的限制,这类机器性能虽然在不断提高,但仍跟不上应用的需求。从六十年代起计算机科学家就开始模拟人类组织社会劳动的并行方式,将并行原理引入计算机结构设计,开拓了设计和生产高性能计算机的崭新道路^[6,7]。

随着并行计算系统的研制与发展,出现了对数值计算方法的一种新的分类:适合于串行计算机使用的传统数值计算方法称为串行算法,而面向并行计算机和向量计算机的数值计算方法则称为并行算法。并行算法在科学计算可视化领域得到广泛应用,同时也是科学计算可视化的一个重要研究方向^[8,9]。

在计算机图像处理,特别是地理信息系统和遥感应用中,景观模型的可视化及景观模型的结构分析显得越来越重要,而且景观模型的可视化常常要求进行实时处理,使用多CPU的图形工作站、微机群及与其相适应的并行算法已成为加快处理和显示速度的重要途径^[5]。现有的数字高程模型(DEM)透视变换及纹理覆盖的算法是串行的,速度慢,因此有必要对其并行算法进行研究。本文根据三维景观数据的自身特点对三维景观模型的数据处理和计算方法加以改造,将DEM及纹理影像分成若干条带进行并行处理,然后进行结果合并,达到了加快速度的目的。

2 DEM(叠加纹理影像)三维显示

DEM 三维显示分两步,第一步是按一定的变换规律求得DEM 格网点的屏幕坐标,以下以透视变换为例;第二步是绘制,绘制的重点在于消隐,这里以缓冲算法为例。

2.1 透视变换

这一步的目的,是将变换范围内任意一个三维点 (x, y, z) 按透视变换原理变换成平面点 (X, Y) 。

先不考虑高程的影响,如图1(a)所示,设视平线为 $Y=d$,左消失点为 (X_L, d) ,右消失点为 (X_R, d) ,分别连接点 $(x, 0)$ 和左消失点、点 $(y, 0)$ 和右消失点,两线交点坐标就是待求点 (x, y) 的透视变换坐标。如果纹理影像的分辨率比DEM的分辨率高,则DEM 格网内部各点的透视变换坐标可按双线性内插方法加密,如图1(b)所示。各加密点的高程由其所在格的四个角点的高程用双线性内插法求得。

接下来对各点Y坐标加入高程值的影响。设定一个高程夸大比例因子 m_z ,则对高程的点需要加入的影响值为:

$$z * (1 - Y/d) * m_z$$

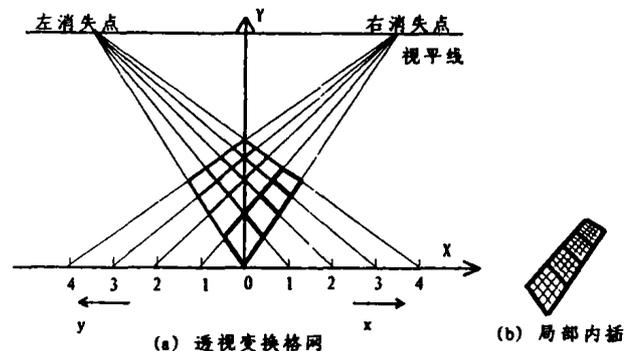


图1 透视变换

2.2 绘制

经过以上变换,就得到了由加密点及格网点构成的四边形集合。首先将整个缓冲区填上底色(比如黑色),然后将每个四边形填以纹理影像上对应点的颜色。根据缓冲算法,填充时由近到远,对已填色的元素则不再填^[10]。这样,就完成了DEM 叠加纹理影像的三维显示过程。

3 DEM(叠加纹理影像)三维显示

3.1 适合于分布存储环境的并行算法

从以上描述可以看出,这是一个纯串行的算法:较远的四边形必须等到较近的四边形填完之后才能填。要用并行算法来实现,必须针对不同的并行处理环境对它进行改造。以下先设计针对分布存储环境的并行算法。

^{*} 本文得到国家自然科学基金重大项目69896250、中国科学院“九五”基础性研究重大项目KJ951-B1-703支持。黄国满 博士研究生,长期在地理信息领域从事计算机图像处理研究与软件系统开发。何建邦 研究员,博士生导师,欧亚科学院院士,长期从事地理信息系统(GIS)和信息共享研究。

设 DEM 的格网点数为 $m \times n$, 并假定有 k 台处理机可用, 则可以将 DEM 分成 k 个条带, 各条带按其距离, 从近到远依次编号为 $1 \sim k$, 每个条带的大小为 $(m/k+1) \times n$, 每台处理机都分配能覆盖整个区域的同样大小的缓冲区, 处理一个条带。若 m 不能被 k 整除, 则多余部分分给前若干台处理机。因为各个条带之间的计算各不相干, 是异步算法, 而前几台处理机往往首先准备好数据, 先开始计算, 所以多余部分分配给它们有助于下一步(合并计算)的同步。

接下来的工作是将各条带的结果合并。合并可以采用并行算法中的结合扇入算法。

考虑形如:

$$A_k = a_1 \circ a_2 \circ \dots \circ a_k$$

的算式, 其中“ \circ ”是一种满足结合律的运算, 可以按图2所示的并行算法计算树直观地表示(以 $k=8$ 为例)。图中“ \square ”表示树的叶子, 代表操作数, “ \circ ”是节点, 代表一种运算, 节点上所标出的数字表示运算执行的步次。这种算法按其执行过程形象地取名为结合扇入算法。

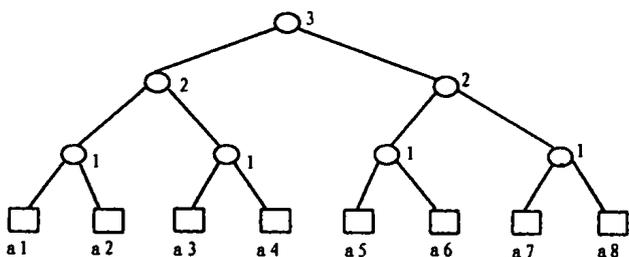


图2 结合扇入法

条带 $i(1 \leq i \leq k)$ 和条带 $(i+1)$ 的合并算法: 对条带 i 和条带 $(i+1)$ 影像重叠部分的外接矩形内的任意一个对应点, 若在条带 i 上颜色值不是底色, 则不作处理, 否则将该点在条带 $(i+1)$ 上的颜色值填到条带 i 的对应点上。这样, 条带 i 的缓冲区就是这两个条带合并的结果。显然, 这种算法是满足结合律的, 可以用上述的结合扇入法。



图3 四个条带的图像(为方便排版, 合成了一个图像)

合并运算时, 需要利用网络在各处理机之间传送数据。为了节省传送时间, 应该在传送之前对数据进行压缩, 传送完成后再解压缩。因为图像中存在大量的空白区域, 所以运行长度压缩法用在这里比较合适。特别是当把图像解压缩过程和条带合并过程结合在一起时, 原来对像素点是否底色的逐点判断变成了多点一次判断, 这样赢得的时间基本上能抵偿图像

压缩时所耗费的时间。

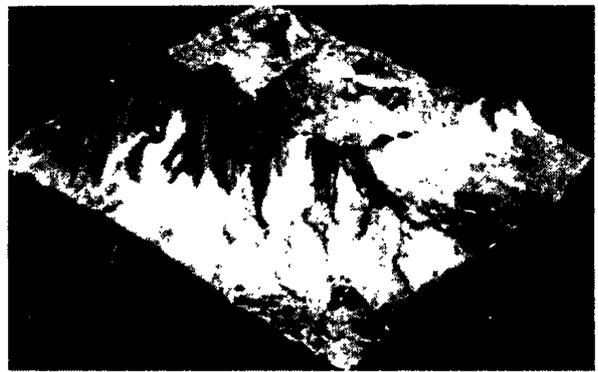


图4 合并后的图像

通常, 用以下两个概念来描述并行算法性能:

①加速。这是并行算法研究的最重要的概念之一。定义这个概念的目的在于度量一个并行算法的并行性质。并行算法对给定的串行算法的加速定义为:

$$S_p = \frac{\text{串行算法在单处理机上的执行时间}}{\text{并行算法在具有 } P \text{ 台处理机的系统上的执行时间}}$$

前面用到的结合扇入法所实现的加速约为 $S_p = p / \log_2 p$ 。

②效率。并行算法的效率用 E_p 表示, 定义为:

$$E_p = S_p / p$$

其中 p 为处理机个数。由于 $S_p \leq p$, 故 $E_p \leq 1$ 。理想的加速情况为 $S_p = p$, 此时 $E_p = 1$ 。一个较现实的目标是构造对 p 为线性加速的算法, 即 $S_p \approx cp$, 其中 c 为与 p 无关的在 $0 \sim 1$ 之间的一个常量。

以上并行算法在串行机 HP 工作站上进行了模拟试验。该试验假定处理机台数为4, 产生了如图3的四个条带图像, 它们的合并结果见图4。该算法在试验中实现的加速 $S_{p=4}$ 约为 3.6, 效率 $E_{p=4}$ 约为 0.9。

3.2 适合于共享存储环境的并行算法

对于共享存储类型的并行计算机, 可以对以上算法进行一些改造, 使其效率更高。将处理机在逻辑上排列成首尾相接的一个队列, 组成一条流水线, 每个处理机每次计算一个方格, 计算结果按次序存放在统一的缓冲区中。当某个处理机解算完它所承担那一行的第一个方格时, 下一个处理机就可以开始上一行的计算, 不必等到下一行全部解算完。

这是一种同步算法, 每一个方格的计算必须等到它的正下方那一个方格的计算结束后才能开始, 这样从第一台处理器开始运算到全部处理器都运行起来, 需要 $(k-1)$ 个方格的计算时间, 同样, 从有一台处理器空闲下来到全部运算结束, 也需要 $(k-1)$ 个方格的计算时间, 但这样的开销对于大体积的 $(m-1) \times (n-1)$ 个方格数据的计算量来说是微不足道的, 而且使用共享存储, 省去了类似条带合并的运算, 因此, 这一算法的效率比上一个算法效率要高。

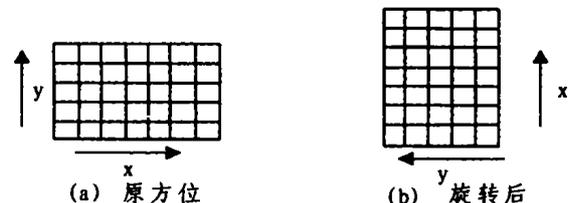


图5 优先顺序的变化

(下转第124页)

象的标记。服务端接收到请求以后,根据此标记找到对应的对象实例。

一个 TMiddlewareProfileBody 实例可以编码成一个字节流的封装。这个封装也就是对应于前面所定义的 Tagged-Profile 的成员变量 profile_data,并且此 TaggedProfile 的成员变量 tag 为 TAG_TMIDDLEWARE_IOP,客户根据此标记区分服务程序的地址信息。

5.3 Wrapper(对消息中间件的封装)

消息中间件是基于 C 语言实现的,为实现从 GIOP 到具体传输接口的映射,核心如虚连接的建立、消息的发送和接收以及具体协议相关的地址定义等,我们用面向对象语言对消息中间件进行封装。这样可以向 GIOP 层和核心提供一个公共的调用接口,从而最终支持协议槽规定的语义。

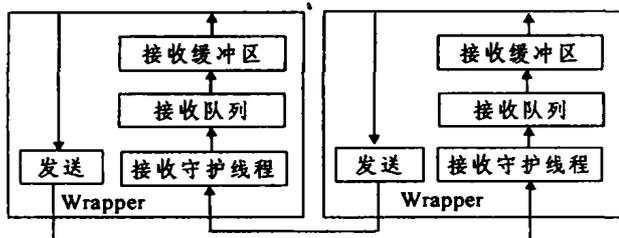


图5 Wrapper 的内部实现

(1)多线程封装 由于 ORB 是多线程的,因此对消息的发送和处理都应该提供对多线程的支持。在消息中间件的消息接收前,加入了一个接收队列,能够区分各自不同的线程号,最后再进行消息的接收和处理。

(2)端点和连接的封装 在这里主要将消息中间件的地址信息封装为被 ORB 中其它逻辑元素所能够识别的实体,向映像工厂、连接工厂、侦听器提供服务程序的地址信息。消息中间件需要定义一条虚连接,提供字节流消息的发送和接收。因此在连接的创建前,就应该完成和连接相关的消息队列的初始化,如消息中间件核心的启动等。

(3)消息的封装 CORBA 客户的请求必须按照 CDR 数据流格式进行封装,才能在网上进行传输和在不同的操作系统和不同的 ORB 之间进行通信。对消息的封装是在 GIOP 层完成的,在消息传输层只是需要对消息包进行传递。但并不是说消息是不能够改变的,如果消息需要特殊处理,如安全加

密,可以再对消息进行进一步的处理,并由对等通信实体来对消息进行逆操作。

结束语 常规的 ORB 缺乏强大的消息处理能力和消息的存储转发能力,而消息中间件在这方面有足够的支持,我们提出将消息中间件嵌入到 ORB 中,以弥补后者在这方面的不足。在这样的 ORB 中,基于 GIOP 协议的映射方式和动态协议槽,可以扩展 ORB 常规的通信能力,支持定制的协议构件。我们通过协议槽、消息中间件的 Wrapper 和互操作协议实现的消息传输协议构件,既可以支持面向对象的应用开发,又可以向前兼容传统中间件的应用,为相同平台不同 ORB 之间和不同平台 ORB 之间的消息传递提供一个良好的通信环境。所开发出的 S/W 产品,已经在省级范围的电力系统中得到成功的应用,效果良好。

参考文献

- 1 Object Management Group. The Common Request Broker: Architecture and Specification, Revision 2.4, Oct. 2000
- 2 Vinoski S. CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments. IEEE Communications Magazine, 1997, 14(Feb.)
- 3 Gokhale B N A, Schmidt S Y D C. Applying Patterns to Improve the Performance of Fault Tolerant CORBA. In: The 7 Intl. Conf. on High Performance Computing, ACM/IEEE, Bangalore, India, Dec. 2000. 17~20
- 4 O'Ryan C, et al. The Design and Performance of a Pluggable Protocols Framework for Real-time Distributed Object Computing Middleware. In: Proc. of the Middleware 2000 Conf. ACM/IFIP, Apr. 2000
- 5 Schmidt D C, et al. Software Architectures for Reducing Priority Inversion and Non-determinism in Real-time Object Request Brokers. Journal of Real-time Systems, special issue on Real-time Computing in the Age of the Web and the Internet, To appear 2000
- 6 Arulanthu A B, et al. The Design and Performance of a Scalable ORB Architecture for CORBA Asynchronous Messaging. In: Proc. of the Middleware 2000 Conf. ACM/IFIP, Apr. 2000
- 7 Li Gui, Yin Chao-Wan, Zheng Huai-Yuan. A Behavior Model of Object Interoperability. The Chinese Journal of Computers, 1999 (5)

(上接第107页)

需要指出的是,这里是按行处理优先来说明的,如果考虑到旋转、左右消失点变化等情形,则优先顺序是变化的。如图5,当旋转至(b)方位时,就应以列优先,而且列内部各方格的次序也与原来相反。

跟上一个算法一样,对此算法也在串行机 HP 工作站上进行了假定处理机台数为4的模拟试验,实现的加速 S_{p-4} 约为3.8,效率 E_{p-4} 约为0.95。

结论 本文初步探讨了并行算法在三维景观模型可视化方面的应用。并行算法是可视化系统的加速工具,通过它可以加快可视化系统的速度,改善可视化效果。在设计并行算法的过程中,必须考虑 CPU 的个数、存储方式等具体情况以及各 CPU 之间的负载平衡和网络通讯的时间耗费。在网络通讯中采用合适的压缩和解压缩技术,不仅能够缩短通讯时间,还可以将解压缩过程与结果合并过程融合到一块,进而改善整个算法的效率。

参考文献

- 1 Watt A, Watt M. Advanced Animation and Rendering Techniques - Theory and Practice, 1993
- 2 沈志宇,等. 并行编译方法. 北京:国防工业出版社,2000
- 3 Wood J. Visualization Contour Interpolation Accuracy in DEMs. In: Hilary M, H, et al., eds, Visualization in GIS, 1994
- 4 章孝灿,等. 遥感数字图像处理. 杭州:浙江大学出版社,1997
- 5 李军,李德仁. 分布式遥感图像处理中的若干关键技术. 武汉测绘科技大学学报,1999,24(1)
- 6 沈志宇,廖湘科,胡子昂. 并行程序设计. 长沙:国防科技大学出版社,1997
- 7 全惠云,高汉平,康立山,陈毓屏. 并行计算机程序设计导论. 武汉:武汉大学出版社,1998
- 8 邓俊辉,等. 基于虚拟机的并行体绘制. 软件学报,2000,11(8)
- 9 李晓明. 数据并行计算概念、模型与系统. 计算机科学,2000,27(6)
- 10 胡友元,黄杏元. 计算机地图制图. 北京:测绘出版社,1987