

doi:10.3969/j.issn.1673-9833.2014.01.017

# 基于三维标量场的光线投射并行体绘制算法

殷智慧, 李朝奎, 杨武, 张强, 严雯英

(湖南科技大学 地理空间信息湖南省工程实验室, 湖南 湘潭 411201;  
湖南科技大学 地理空间信息技术国家地方联合工程实验室, 湖南 湘潭 411201)

**摘要:** 针对大规模三维标量场相关算法计算量大且数据绘制速度慢、不能满足用户实时需求的问题, 在研究和总结三维标量场和体绘制关键技术的基础上, 对体绘制的原理进行了深入探讨, 重点研究了体绘制中的光线投射算法。利用集群多节点的并行优势, 对体绘制的光线投射算法进行了改进, 采用MPI并行编程模式, 成功地将该算法运行于PC集群环境中, 并应用于三维城市建模, 充分发挥了集群的高性能计算, 大大提高了绘制效率。实验结果表明, 该算法具有良好的并行效率和可视化效果。

**关键词:** 三维标量场; 光线投射算法; 体绘制; MPI

中图分类号: TP391

文献标志码: A

文章编号: 1673-9833(2014)01-0081-06

## The Parallel Ray Casting Volume Rendering Algorithm Based on 3D Scalar Fields

Yin Zhihui, Li Chaokui, Yang Wu, Zhang Qiang, Yan Wenying

(Hunan Province Engineering Laboratory of Geospatial Information, Hunan University of Science and Technology, Xiangtan Hunan 411201, China; National & Local United Engineering laboratory of Geospatial Information Technology, Hunan University of Science and Technology, Xiangtan Hunan 411201, China)

**Abstract:** In view of mass computation of large-scale 3D scalar field correlation algorithm and slow data rendering, and on the basis of research and summary of key technologies of 3D scalar fields and volume rendering, the principle of volume rendering is discussed, focusing on the ray casting algorithm of volume rendering. The ray casting algorithm of volume rendering is improved with the parallel advantages of multi-node cluster. And by means of MPI parallel programming mode, the algorithm is run on the PC cluster environment and applied to 3D city modeling. It fully develops the high performance computing of cluster and greatly improves the rendering efficiency. The experimental results show that the algorithm makes good parallel efficiency and visual effects.

**Keywords:** 3D scalar fields; ray casting algorithms; volume rendering; MPI

### 1 三维标量场

三维标量场是用来研究科学计算可视化的一种

场, 是指场中每一个点的属性值都可以以一个标量来表示的空间场<sup>[1]</sup>。随着时间的变化, 标量场值也发生相应的变化。自然界中的三维标量场一般都是空

收稿日期: 2013-12-25

基金项目: 国家自然科学基金资助项目(41271390), 湖南科技大学研究生优秀学位论文培育基金资助项目(S120036), 湖南省研究生科研创新基金资助项目(CX2013B405)

作者简介: 殷智慧(1987-), 女, 河北张家口人, 湖南科技大学硕士生, 主要研究方向为并行绘制, 空间数据调度, E-mail: zhihui61993@126.com

通信作者: 李朝奎(1967-), 男, 湖南汉寿人, 博士, 湖南科技大学教授, 博士生导师, 主要从事3D GIS空间定量分析及辅助决策应用的研究, E-mail: chkl\_hn@163.com

间连续的<sup>[2]</sup>,但是为了方便存储,通常以离散点云数据进行存储,在物理结构中,采取笛卡尔坐标来表示大规模三维场景数据,本文从三维标量场角度出发,研究并行绘制技术,既是三维标量场可视化研究的内容,也是并行绘制技术创新点。

现实中的大规模三维场景数据是连续的,而研究组在研究过程中获取的数据,是通过各种包括低空测量、高空遥感形式获得的离散数据,而将这些三维离散数据点转换为二维屏幕图像数据点也是离散的。基于这些,三维标量场绘制技术可以定义为:将离散分布的三维数据场,按照一定的规则转换为图行设备屏幕上的二维图像,从而绘制出各类地物的分布情况<sup>[3]</sup>。图1为某一区域在三个坐标方向均匀取样得到的标量场。

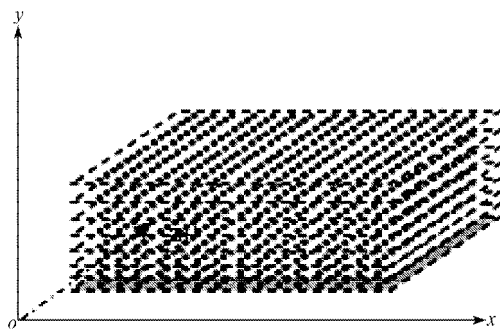


图1 规则三维标量场

Fig. 1 Rules of 3D Scalar Fields

三维标量场可视化包含两种绘制方法,分别是面绘制和直接体绘制。由于面绘制方法首先需要建立中间几何模型,并且只能描述场景的局部信息,而体绘制却能表现地物空间的细节信息,且不需要建中间模型。鉴于体绘制这种优势,本文对体绘制算法展开深入研究。

## 2 体绘制技术

在体绘制过程中,将大规模三维场景数据投射到二维平面形成绘制图像,主要有两种方法:一种是基于图像空间划分的体绘制算法,另一种是基于物体空间内容划分的体绘制算法<sup>[4]</sup>。典型的以图像空间为序的体绘制算法为光线投射算法;以物体空间为序的体绘制算法有错切变形体绘制算法、体元投射法、Splating 体绘制算法、足迹表法等<sup>[5]</sup>。

基于物体空间内容划分的前提是采取合理的方法对数据场进行剖分,得到若干个数据子块,再将这些数据子块分发到每一个计算节点。在以物理空间为序的体绘制算法中,每一条射线的计算都与所划分的数据子块相关,因此,每条射线的计算都需

要几个节点进行数据存储,增加了节点的负担,限制了通信的效率。加之投影屏幕图像的形成,不单单涉及简单的数据子块,它与整个数据相关,因而在存储过程中一般在每个节点重复拷贝整个数据场。这种方法的优点是任务间的数据相关性小,通讯性能比较高。但是,冗余存储数据会要求每个节点需要有较大的存储空间,旋转视线的不同加大了编程的难度,大大降低了绘制速率。

而基于图像空间的划分则是通过对投影屏幕进行划分,将屏幕划分成包含多个三维标量场体数据投影像素点的图像子块,并将形成这些图像子块所对应的标量场数据存储在计算节点上,作为本地访问数据。每个节点对应于一个图像子块,将全部的数据场数据存储在远程服务器上。在节点需要时,通过管理节点分发数据给所需要的绘制节点。节点间具有很高的独立性,但是考虑到观察视点的不同,在实际绘制过程中需要重采样,可能因光线问题引起实际操作则需要访问三维数据场中的每一个数据,从而使得节点间必须进行频繁通信。这种方法的优点是存储消耗小,缺点是通讯量比较大,降低了绘制效率。鉴于以上分析,本文主要探讨基于图像空间划分的光线投射体绘制算法。

## 3 光线投射体绘制原理

体绘制可以追溯到上个世纪70年代,最早用在医学领域,而众多的体绘制算法中,比较常用的光线投射体绘制,又叫后投影(back-ward projection)<sup>[6]</sup>,是在1987年由 Marc Levoy<sup>[7]</sup>提出的,它是以图像空间为序的体绘制算法。光线投射体绘制算法过程如图2所示。

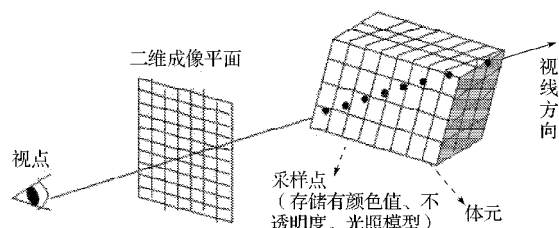


图2 基于光线投射的体绘制过程

Fig. 2 Volume rendering process based on ray casting

算法基本思想可以描述为:从视点出发,沿视线方向发射光线,光线穿过投影成像平面和三维场空间,且光线与三维数据场相交,沿着交线进行采样。采样点的选取要遵循一定的规则,必须以同样的间隔进行采样。然后分别对这些采样点进行内插计算(最常用的是三线性插值方式),从而计算出每个采

样点数据的采样值(它包含每个采样点的颜色值和不透明度值)。最终二维屏幕上图像的形成,是将这些采样点采取由前至后的方式或由后至前的方式,进行颜色值和不透明度值的累积而形成的。正是由于光线投射算法的这一特点,使得对三维场景数据进行绘制时,能够保留更多的地物细节,并获得较高质量的信息,且计算过程最明确<sup>[8]</sup>。

以上是从光学知识角度分析体绘制合成过程,下面对光线投射算法的实现进行进一步分析,光线投射算法的具体实现过程如下。

#### 1) 对三维标量场体数据进行读取和预处理。

其目的是判断三维场景所在的立方体的可见性,并确定三维场在平面上的投影区域。为了区分三维场的数据类别,或者找出同类物质不同属性值,通常采取两种方法进行分类:阈值法和概率法。本文采用阈值法,即对数据灰度值的变化范围进行等级划分进而分类。在存储的时候,以头文件形式存储分类信息,这种分类存储办法能够很好地表示多种物质的不同分布和同一物质的不同属性<sup>[9]</sup>。阈值法可描述为:根据对全部采样点的取值进行统计后,设定若干阈值 $d_i (i=1, 2, \dots, n)$ ,如果各采样点的数值以 $(x_i, y_j, z_k)$ 表示,则将满足下列条件的采样点归入同一类中: $d_i \leq f(x_i, y_j, z_k) \leq d_j^{[10]}$ 。

2) 通过主节点把投影屏幕图像进行划分,依次分配给各个计算节点。

3) 设置三维场景参数和绘制属性值,对投影屏幕上的每个像素进行绘制,具体步骤如下。

a) 在投影成像平面选择一个像素点 $P(i, j)$ ,并将该点二维坐标转换为三维坐标,从视点出发,使视线方向穿过成像屏幕像素点,获得视线与三维场景相交。

b) 按特定的步长进行重采样,得到采样点颜色值 $(R, G, B)$ 和不透明度值 $(Alpha)$ 。在体绘制中,三维标量场数据的重采样是关键步骤。一般情况下,从光线穿过标量场投影到屏幕成像的过程中,从某个角度发出的光线可能产生两个问题:第一,标量场中被遮挡的体素可以透过缝隙投影到投影平面而造成屏幕显示错误;第二,屏幕上的某些区块可能没有体素投影到屏幕而造成该块显示背景色。重采样的目的就是为了避免由于以上两个问题导致错误绘制成像。

c) 合成采样点和不透明度值,得到最终的颜色值和不透明度值。在给采样点赋值时,为获得具有内部详细结构的三维数据场,需要对采样点的分类结果赋予不同的颜色值和不透明度值。通常,在不

失图像颜色的前提下,遵循用户的需求来赋值,一般地, $Alpha$ 的取值在0~1之间: $0 \leq Alpha \leq 1$ , $Alpha=1$ 表示所要计算的体素是完全不透明的, $Alpha=0$ 表示该体素是完全透明的。值得注意的是,最终得到的图像中的颜色并不是物体的真正颜色,而是合成的,将其概括为伪色彩。

d) 重复步骤a), b), c), 计算投影成像平面的所有像素点合成后的颜色值和不透明度值;

e) 判断步骤d)是否将所有像素点进行投影,如果没有,则继续步骤2;否则,进行下一步的操作。

#### 4) 最终图像合成显示。

在最终图像合成过程中,按照一定的规则进行图像合成。常用的合成算法主要有两种:一种是由后向前对图像进行合成,也就是沿发射光线从后往前合成发射光线上的多个采样点的颜色值和不透明度值,以求得最后的图像;另一种是由前向后对图像进行合成<sup>[11]</sup>。本文采用第一种图像合成算法,其具体实现可描述为:按照采样点从后往前的顺序,设采样点从初始颜色值到最终颜色值依次为: $c_0, \dots, c_i, \dots, c_n$ ,不透明度值依次为: $1, \dots, a_i, \dots, 0$ ,则透明度值为 $1 - a_i$ ,最终合成的颜色值 $C$ 用公式描述为:

$$C = c_0(1-a_1)(1-a_2)\dots(1-a_n) + c_1a_1(1-a_2)(1-a_3)\dots(1-a_n) + c_2a_2(1-a_3)(1-a_4)\dots(1-a_n) + \dots + c_{n-1}a_{n-1}(1-a_n) + c_n a_n$$

光线投射体绘制实现思路如图3所示。

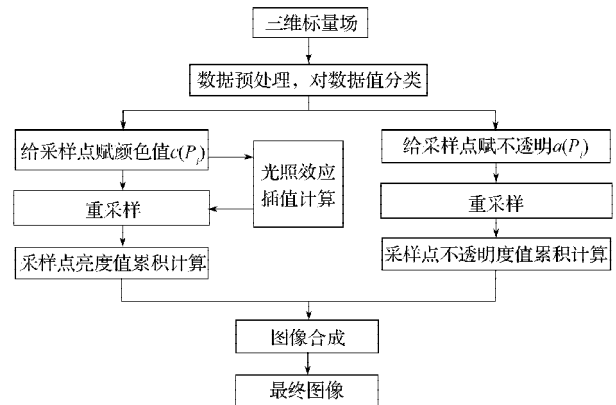


图3 光线投射体绘制实现思路

Fig. 3 The implementation idea of algorithm in ray casting volume rendering

## 4 基于集群的光线投射并行体绘制算法

光线投射算法绘制的可视化图像精度最高,但缺点是存储容量较大,多年来,许多科研工作者通过存储压缩和数据分块的办法来解决这一瓶颈,但往往不可避免地带来一些其它问题,比如有损图像质量等。针对以上问题,考虑到光线投射过程中的

每一条投射光线都是彼此独立的,可采用并行计算的办法解决以上瓶颈。本文在深入分析光线投射算法的基础上,对算法进行了改进,在执行重采样前引入包围盒概念,并在此基础上探讨集群环境下基于图像空间划分的光线投射并行体绘制算法。所谓的包围盒,就是一个标准的长方体<sup>[1]</sup>。假设该包围盒存在于三维标量场中,首先判断穿过屏幕像素点的入射光线与该包围盒的6个面是否有交点。如果有交点,分别计算入射点和出射点,然后根据采样点存储的颜色值和不透明度值,利用三线性差值方法计算贡献值;如果没有交点,则表明标量场中的数据对屏幕所对应的像素点无贡献值,则可略去此步骤的计算,由此可简化算法实现。

除此之外,考虑到图像空间划分后的数据结构相关性较小,可以将每一个屏幕图像子块分配到计算节点,也就是采用集群并行的方式,将屏幕图像子块作为计算节点的任务,然后对分配到每个节点的图像空间子块所对应的三维标量场数据进行插值计算。整个并行计算过程采用 Master-Slaver 的并行编程模式,头节点(Master 进程)负责任务划分和分配,计算节点(Slaver 进程)进行屏幕图像的像素绘制,各节点绘制的图像最后交给头节点进行图像合成。图4为光线投射并行体绘制算法实现流程图。

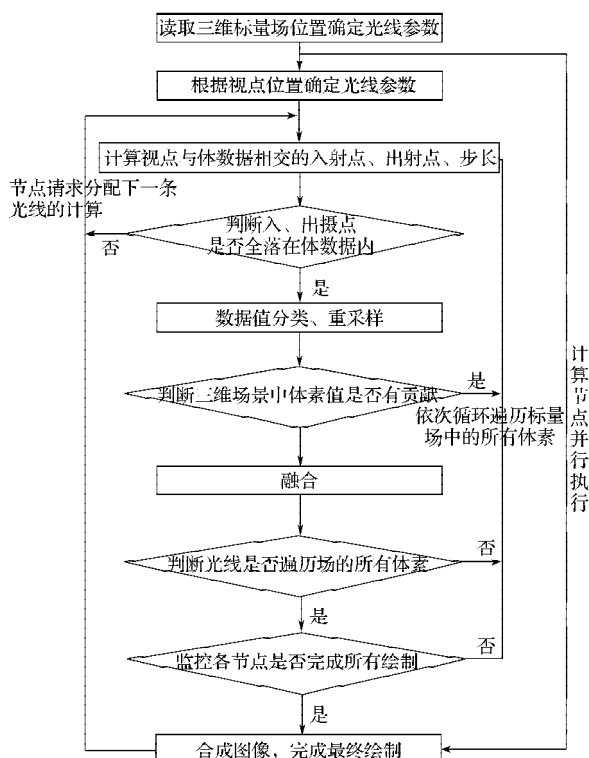


图4 光线投射并行体绘制算法的流程

Fig. 4 The flow chart of parallel ray casting volume rendering algorithm

### 1) 屏幕图像划分及节点间数据分发的步骤

Step 1 MPI初始化。通过MPI\_Init函数进入MPI环境,并完成所有的初始化工作。

Step 2 首先,头节点为每个计算节点分配数据,然后各个计算节点进行图像绘制,并将绘制好的图像数据传送给头结点。

Step 3 获取进程编号。调用MPI\_Comm\_rank函数,获得当前进程在指定通信域中的编号,将自身与其他程序区分。

Step 4 获取指定通信域的进程总数。调用MPI\_Comm\_size函数获取指定通信域的进程个数,确定自身完成任务比例。

Step 5 获取机器名。调用MPI\_Get\_processor\_name函数获取相应的机器名。

Step 6 消息发送和接收。通过MPI\_Send函数发送一个消息到目标程,通过MPI\_Recv函数用于从指定进程接收一个消息。

Step 7 头节点接收各计算节点传来的数据,并合成最终绘制图形。

### 2) 每一个节点所对应的光线投射体绘制算法实现步骤

Step 1 设置不透明度值为0,像素初始颜色值设为背景色,计算像素点 $p(x, y)$ 对应的物体空间坐标 $P(x, y, z)$ ;

Step 2 从像素点发射光线,并判断视线与体数据是否相交。如果相交,跳转到步骤3;如果不相交,则剔除空体素;

Step 3 根据当前射线上采样点坐标,用三线性插值法求出当前采样点的颜色值,通过循环递归的方法,计算所有采样点坐标的颜色值和不透明度;

Step 4 累计不透明度值和颜色值,并不断更新灰度值和颜色值;

Step 5 重复以上步骤2~4,直到完成整个图像的绘制。

## 5 实验及结果分析

### 5.1 并行环境

实验的并行计算环境是Window HPC Server 2008高性能计算集群系统,配置了符合MPI(multipoint interface)标准的消息传递库,即将MPICH2库集成到Visual Studio 2010环境中,使用MPI来实现各节点之间的通信。采用4个计算节点。每个节点的配置为:16个Intel(R)Xeon 2.4 GHz CPU;512 kB L2 Cache;2 GB DDR RAM,节点间采用KVM进行多电脑切换。

## 5.2 实验结果与分析

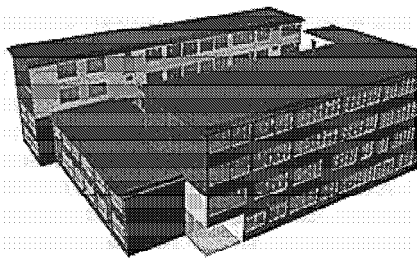
本实验选择建筑模型数据为数据源,选择 Windows HPC Server 集群环境为实验环境,对不同规模的建筑物数据进行体绘制。其算法基本思路是:各计算节点可独立计算穿过屏幕该像素点和子数据的光线段的绘制,节点间不存在通讯消耗,各节点可生成部分图像,且合成部分也可并行进行。为验证本文算法的效率,将本文算法与传统的体光线投射算法进行了对比,所得实验结果如表 1 所示。

表 1 算法运行时间对比

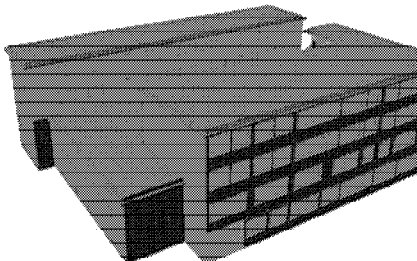
建筑物 模型数据	空体 素率 / %	数据规模	传统光线 投射算法	本文算法
			绘制时间 / s	绘制时间 / s
1	62	256*256*128	6.7	1.2
2	60	512*512*160	10.3	3.8
3	57	512*512*256	12.7	4.9

注:空体素率为  $\frac{\text{空体元数}}{\text{总体元数}} \times 100\%$ ; 体元<sup>[12]</sup>是数据场中的一个四面体,是数据场的最基本组成单位;空体元<sup>[13-14]</sup>是指非透明度取值等于 0 或很低的体元,空体元通过传递函数映射后,完全透明,对最终图像没有贡献。

由表 1 不难看出,当数据规模为 512\*512\*160 时,使用传统光线投射算法耗时 10.3 s,使用本文算法用时 3.8 s。改进后的算法除了能够运行在并行环境中,而且增加了无效光线的判断和空体素的剔除,且各子任务间相互独立计算,这样通过减少图像合并时间,进而缩短了绘制时间,同时也提高了绘制效率。对于建筑物模型数据 2,两种算法的绘制效果如图 5 所示。



a) 传统光线投射算法绘制效果 (绘制帧率为 18 帧/s)



b) 本文算法绘制效果 (绘制帧率为 27 帧/s)

图 5 建筑物数据的体绘制对比效果图

Fig. 5 The contrast diagrams of volume rendering effect about building data

对比效果图不难看出,采用本文提出的算法对相同的体数据进行绘制,绘制图像结果的质量有明显差别,绘制帧率提高到 27 帧/s,得到的效果图更加细腻逼真,同时绘制时间也有所降低。对比传统的光线投射算法,本文所提出的算法具有良好的效率和可视化效果。

## 6 结语

本文在 HPC 集群环境下讨论了基于光线投射的并行体绘制算法,算法采用 Master-Slave 计算模型,由头节点 (Master 进程) 负责屏幕图像的分配、接收和合并显示,计算节点 (Slave 子进程) 负责相应的子图像的体绘制计算。算法采用 MPI 并行编程模式,选择 3 台 PC 节点对改进的算法进行实验验证,实现了对图像的并行数据划分,解决了单机耗时长的问题,在一定程度上提高了绘制速度,但是实时性并不是很高。此外,本文的研究仅仅局限于规则数据场,而没有考虑不规则数据场的划分与任务分配以及并行绘制。对算法的进一步改进将是今后研究的重点。

### 参考文献:

- [1] 于荣欢,邓宝松,吴玲达,等.时变三维标量场并行计算与绘制框架研究[J].计算机科学,2012,39(3):244-251.  
Yu Ronghuan, Deng Baosong, Wu Lingda, et al. Parallel Computing and Rendering Framework of Time Variable 3D Scalar Fields[J]. Computer Science, 2012, 39(3): 244-251.
- [2] 于荣欢,邓宝松,吴玲达,等.三维标量场并行等值面提取与绘制技术[J].计算机辅助设计与图形学报,2012,39(3):183-186.  
Yu Ronghuan, Deng Baosong, Wu Lingda, et al. Parallel Isosurface Extracting and Rendering of 3D Scalar Fields[J]. Journal of Computer-Aided Design & Computer Graphics, 2012, 39(3): 183-186.
- [3] 姚丽娇.科学计算中的标量场可视化技术[D].沈阳:东北大学,2009.  
Yao Lijiao. The Scalar Fields Visualization Technology in Scientific Computing[D]. Shenyang: Northeastern University, 2009.
- [4] 洪振刚,罗省贤.多层次并行体绘制算法的研究与应用[J].计算机工程与科学,2009,31(增刊1):221-224.  
Hong Zhengang, Luo Shengxian. Research and Application on Hierarchical Parallel Volume Rendering Algorithm[J]. Computer Engineering & Science, 2009, 31(S1): 221-224.

- [5] 朱金亮. 基于足迹法的三维地震数据并行可视化方法研究[D]. 南京: 南京理工大学, 2008.  
Zhu Jinliang. A Method of Parallel Visualization in 3D Seismic Data Based on Footprint Method[D]. Nanjing: Nanjing University of Science and Technology, 2008.
- [6] 刘如娟. 基于体绘制的三维数据场可视化技术研究[D]. 太原: 太原理工大学, 2003.  
Liu Rujuan. The Visualization of the 3D Sets Based on Volume Rendering Technology[D]. Taiyuan: Taiyuan University of Technology, 2003.
- [7] Levoy M. Display of Surface From Volume Data[J]. IEEE Computer Graphics and Application, 1988, 8(3): 29-37.
- [8] 徐赛花, 张二华. 基于CUDA的三维数据并行可视化[J]. CT理论与应用研究, 2011, 20(1): 47-54.  
Xu Saihua, Zhang Erhua. CUDA-Based Parallel Visualization of 3D Data[J]. CT Theory and Applications, 2011, 20(1): 47-54.
- [9] 张薇薇, 张 桦. 光线投射体绘制算法关键技术研究[D]. 天津: 天津理工大学, 2006.  
Zhang Weiwei, Zhang Hua. Research on Key Technique of Ray Casting Volume Rendering Algorithm[D]. Tianjin: Tianjin University of Technology, 2006.
- [10] 唐泽圣, 陈 莉, 邓俊辉. 三维数据场可视化[M]. 北京: 清华大学出版社, 1999: 1-253.  
Tang Zesheng, Chen Li, Deng Junhui. Visualization in 3D Data Fields[M]. Beijing: Tsinghua University Press, 1999: 1-253.
- [11] 叶 良, 单桂华, 迟学斌. 基于CUDA加速的光线投射法研究[C]//图像图形技术研究与应用学术会议论文集. 北京: 北京交通大学出版社, 2010: 235-240.  
Ye Liang, Shan Guihua, Chi Xuebin. CUDA-Accelerated Ray-Casting Volume Rendering[C]//IGTA2010. Beijing: Beijing Jiaotong University Press, 2010: 235-240.
- [12] 朱 爽, 顾耀林. 基于体元投影的一种非规则数据场体绘制方法[J]. 计算机工程与应用, 2008, 44(15): 68-70.  
Zhu Shi, Gu Yaolin. Volume Rendering Algorithm of Irregular Volume Based on Cell Projection[J]. Computer Engineering and Applications, 2008, 44(15): 68-70.
- [13] 湛海新, 缪 琳, 马丙辰. 基于直接体绘制的三维数据场交互可视化系统SinoVis[J]. 计算机应用, 2004, 24(4): 67-69.  
Chen Haixin, Miao Lin, Ma Bingchen. SinoVis: An Interactive Direct Volume Rendering-Based 3D Data Visualization System[J]. Computer Application, 2004, 24(4): 67-69.
- [14] 彭 伟, 李建新, 闫 镔, 等. 基于改进空体素跳跃法的光线投射算法[J]. 计算机工程, 2012, 38(2): 264-266.  
Peng Wei, Li Jianxin, Yan Bin, et al. Ray Casting Algorithm Based on Improved Empty Voxel Skipping Method[J]. Computer Engineering, 2012, 38(2): 264-266.

(责任编辑: 申 剑)