

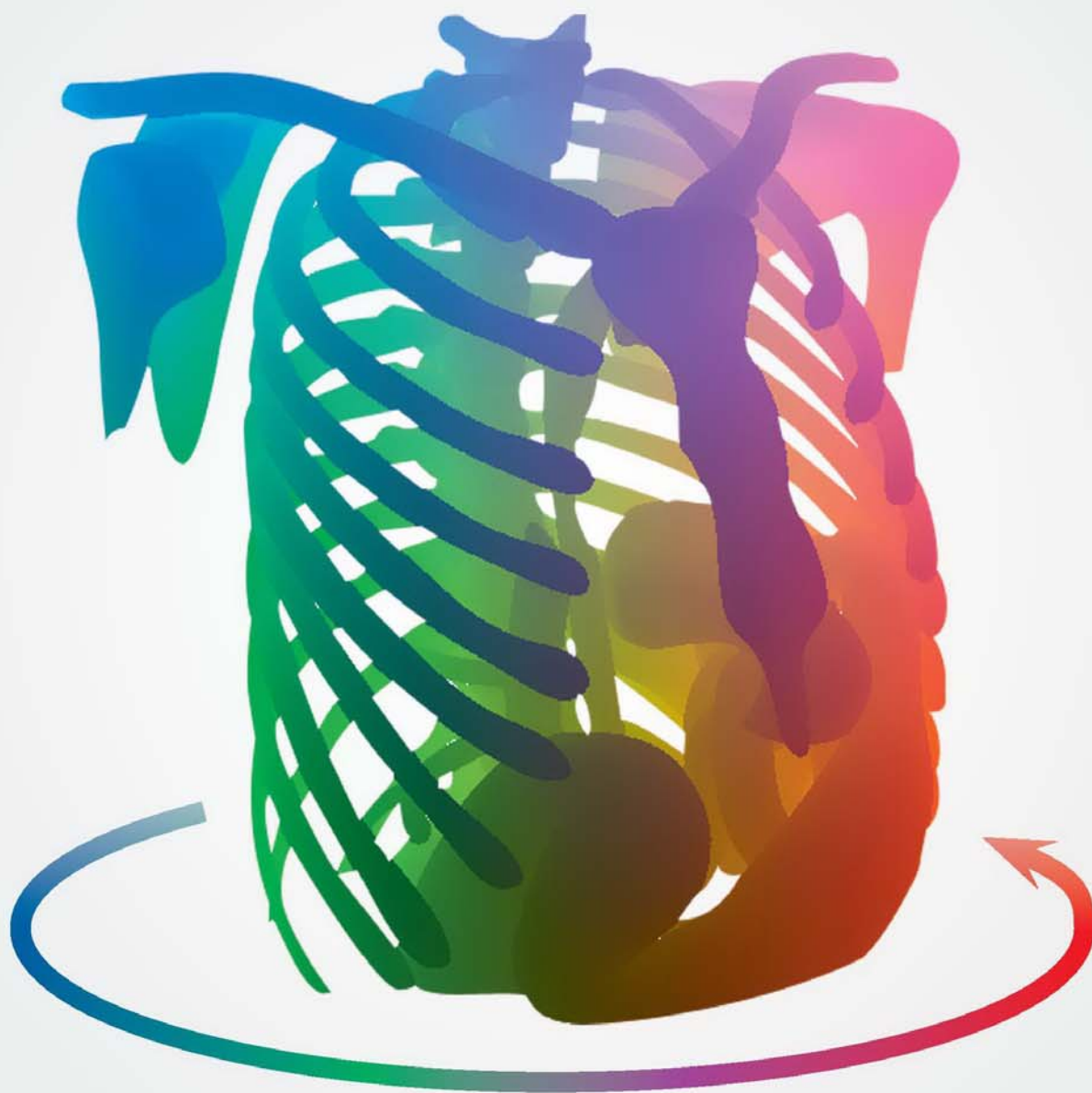
JOURNAL OF IMAGE AND GRAPHICS

主办: 中国科学院遥感与数字地球研究所
中国图象图形学学会
北京应用物理与计算数学研究所

中国图象图形学报

2015
02
VOL.20

ISSN1006-8961
CN11-3758/TB



等值面光线跟踪 P193



第20卷第2期（总第226期）

2015年2月16日

中国精品科技期刊
中国国际影响力优秀学术期刊
中国科技核心期刊

版权声明

凡向《中国图象图形学报》投稿，均视为同意在本刊网站及CNKI等全文数据库出版，所刊载论文已获得著作权人的授权。本刊所有图片均为非商业目的使用，所有内容，未经许可，不得转载或以其他方式使用。

Copyright

All rights reserved by Journal of Image and Graphics, Institute of Remote Sensing and Digital Earth, CAS. The content (including but not limited text, photo, etc) published in this journal is for non-commercial use.

主管单位 中国科学院

主办单位 中国科学院遥感与数字地球研究所

中国图象图形学学会

北京应用物理与计算数学研究所

主 编 李小文

编辑出版《中国图象图形学报》编辑出版委员会

邮政信箱 北京9718信箱

邮 编 100101

电子信箱 jig@irsa.ac.cn

电 话 010-64807995

网 址 www.cjig.cn

广告经营许可证 京朝工商广字第0361号

总 发 行 北京报刊发行局

订 购 全国各地邮局

海外发行 中国国际图书贸易集团有限公司

（邮政信箱：北京399信箱 邮编：100048）

印刷装订 北京科信印刷有限公司

Journal of Image and Graphics

Title inscription: Song Jian

Monthly, Started in 1996

Superintended by Chinese Academy of Sciences

Sponsored by Institute of Remote Sensing and Digital Earth, CAS

China Society of Image and Graphics

Institute of Applied Physics and Computational Mathematics

Editor-in-Chief LI Xiaowen

Editor, Publisher Editorial and Publishing Board of Journal of Image and Graphics

P.O.Box 9718, Beijing, P.R.China

Zip code 100101

E-mail jig@radi.ac.cn

Telephone 010-64807995

Website www.cjig.cn

Distributed by Beijing Bureau for Distribution of Newspapers and Journals

Domestic All Local Post Offices in China

Overseas China International Book Trading Corporation
(P.O.Box 399, Beijing 100048, P.R.China))

Printed by Beijing Kexin Printing Co., Ltd.

CN 11-3758/TB

ISSN 1006-8961

CODEN ZTTXFZ

国外发行代号 M1406

国内邮发代号 82-831

国内定价 50.00元



多尺度活动网格在云场景仿真中的应用(第0202页)



自然环境视频中基于显著鲁棒轨迹的行为识别(第0245页)



虚拟手抓持力觉生成算法真实性的评价(第0280页)

图像处理和编码

引入神经网络中间神经元的快速小波图像压缩

张海涛, 张永霖 0159

拟正态分布扩散的图像平滑

周先春, 汪美玲, 周林锋 0169

可选特征的快速分形图像编码

袁宗文, 鲁业频, 杨汉生 0177

图像分析和识别

融合多尺度码本的全局编码图像分类

董振宇, 赵杰煜, 祝军 0183

计算机图形学

结合K-D树和Shell的快速动态等值面光线跟踪法

罗月童, 石放放, 张伟, 朱会国 0193

多尺度活动网格在云场景仿真中的应用

范晓磊, 张立民, 张建廷, 徐涛 0202

第十届和谐人机环境联合会议专栏

低资源语言的无监督语音关键词检测技术综述

杨鹏, 谢磊, 张艳宁 0211

快速离散Curvelet变换域的图像融合

杨勇, 董松, 黄淑英 0219

高效视频编码中变换跳过模式的快速选择

王宁, 张永飞, 樊锐 0229

主动学习的多标签图像在线分类

徐美香, 孙福明, 李豪杰 0237

自然环境视频中基于显著鲁棒轨迹的行为识别

易云, 王瀚漓 0245

特征变换和数据集分块的行人比对

沈洋, 林巍晓, 殷惠清 0254

移动手持环境下触摸式目标选择技术对比

辛义忠, 李岩, 袁伟强, 郑谦 0264

虚拟环境的用户意图捕获

程成, 赵东坡, 卢保安 0271

虚拟手抓持力觉生成算法真实性的评价

杨文珍, 许艳, 颜传武, 吴新丽, 张昊, 孟闯 0280

交互式乐器演奏的六自由度力觉渲染方法

童号, 史有姣, 王党校, 张玉茹 0288

CONTENTS

JOURNAL OF IMAGE AND GRAPHICS



Multi-scale moving grids and its application in clouds scene simulation(P0202)



Action recognition from unconstrained videos via salient and robust trajectory (P0245)



Validation the Authenticity of Virtual fingers Grasping Haptic Algorithm(P0280)

Image Processing and Coding

- Fast wavelet image compression introducing neural network of relay neuron
Zhang Haitao, Zhang Yonglin 0159
- Image smoothing algorithm based on matching normal distribution diffusion
Zhou Xianchun, Wang Meiling, Zhou Linfeng 0169
- Optional features fast fractal image coding algorithm
Yuan Zongwen, Lu Yepin, Yang Hansheng 0177

Image Analysis and Recognition

- Image classification based on global coding combined with multi-scale codebook
Dong Zhenyu, Zhao Jieyu, Zhu Jun 0183

Computer Graphics

- Fast isosurface ray tracing method by combining K-D tree and Shell
Luo Yuetong, Shi Fangfang, Zhang Wei, Zhu Huiguo 0193
- Multi-scale moving grids and its application in clouds scene simulation
Fan Xiaolei, Zhang Limin, Zhang Jianting, Xu Tao 0202

Column of HHME'2014

- Survey on unsupervised spoken term detection for low-resource languages
Yang Peng, Xie Lei, Zhang Yanning 0211
- Image fusion based on fast discrete Curvelet transform
Yang Yong, Tong Song, Huang Shuying 0219
- Fast transform skip mode decision for high efficiency video coding
Wang Ning, Zhang Yongfei, Fan Rui 0229
- Online multi-label image classification with active learning
Xu Meixiang, Sun Fuming, Li haojie 0237
- Action recognition from unconstrained videos via salient and robust trajectory
Yi Yun, Wang Hanli 0245
- Pedestrian re-identification based on feature transformation and dataset classification
Shen Yang, Lin Weiyao, Yin Huiqing 0254
- Comparison of target selection techniques in mobile handheld environment
Xin Yizhong, Li Yan, Yuan Weiqiang, Zheng Qian 0264
- Intent understanding for virtual environments
Cheng Cheng, Zhao Dongpo, Lu Baoan 0271
- Validation the authenticity of virtual fingers grasping haptic algorithm
Yang Wenzhen, Xu Yan, Yan Chuanwu, Wu Xinli, Zhang Hao, Meng Chuang 0280
- Interactive simulation of music instrument playing by a six degree-of-freedom haptic rendering approach
Tong Hao, Shi Youjiao, Wang Dangxiao, Zhang Yuru 0288

中图法分类号: TP391 文献标识码: A 文章编号: 1006-8961(2015)02-0193-09

论文引用格式: Luo Y T, Shi F F, Zhang W, Zhu H G. Fast isosurface ray tracing method by combining K-D tree and Shell [J]. Journal of Image and Graphics, 2015, 20(2): 0193-0201. [罗月童, 石放放, 张伟, 朱会国. 结合 K-D 树和 Shell 的快速动态等值面光线跟踪法[J]. 中国图象图形学报, 2015, 20(2): 0193-0201.] [DOI:10.11834/jig.20150205]

结合 K-D 树和 Shell 的快速动态等值面光线跟踪法

罗月童, 石放放, 张伟, 朱会国

合肥工业大学计算机与信息学院可视化与协同计算研究室, 合肥 230009

摘要: 目的 K-D 树和 Shell 常被用于加速等值面光线跟踪法。如果 Isovalue 保持不变时, Shell 方法效率更高, 否则 Shell 方法需要重构 Shell, 而 K-D 树方法速度更快。提出一种结合 K-D 树和 Shell 的动态等值面光线跟踪方法, 其关键是如何实现两者的平稳切换。**方法** 首先改进基于 K-D 树的等值面光线跟踪算法, 使得该方法在绘制过程中渐进地构建 Shell 数据结构。在 Isovalue 发生变化后, 首先使用改进的基于 K-D 树的等值面光线跟踪算法进行绘制, 并在绘制过程中渐进地构建新的 Shell 数据结构, 从而平滑地过渡到绘制效率更高的基于 Shell 的等值面光线跟踪算法。**结果** 实验中, 在 Isovalue 动态变化时本文方法的效率接近 K-D 树方法; 但用户在进行缩放、旋转等操作时, 本文方法能达到 Shell 方法的速度。实验结果表明本文方法结合了两者的优点。**结论** 提出了一种快速 Isovalue 光线跟踪算法, 综合运用 K-D 树和 Shell 两种数据结构, 在 Isovalue 保持不变和动态变化两者场合都实现了较高的绘制速度。

关键词: 体数据可视化; 等值面; 光线跟踪; K-D 树; Shell 数据结构

Fast isosurface ray tracing method by combining K-D tree and Shell

Luo Yuetong, Shi Fangfang, Zhang Wei, Zhu Huiguo

Visualization & Cooperative Computing Division, School of Computer & Information, Hefei University of Technology, Hefei 230009, China

Abstract: **Objective** K-D tree and Shell are commonly utilized to accelerate isosurface ray tracing. Shell-based isosurface ray tracing method is more efficient than K-D tree is if the isosurface is fixed. Otherwise, the K-D tree-based isosurface ray tracing method is more efficient because the shell-based isosurface ray tracing method needs to reconstruct the shell. To utilize the advantages of both methods, this paper presents a fast isosurface ray tracing method by combining K-D tree and Shell. The main point of the proposed method is how to switch between the two methods smoothly. **Method** The K-D tree-based isosurface ray-tracing method is first improved to allow a progressive construction of the shell during rendering. The improved K-D tree-based isosurface ray tracing method is used when the isosurface changes, and a new shell is progressively constructed to allow the switch to a faster shell-based isosurface ray-tracing method. **Result** The presented method and K-D tree-based isosurface ray tracing method have similar speed when the isosurface changes frequently. The presented method can also achieve a similar speed as that of shell-based isosurface ray tracing method when only the zoom in/zoom out/rotation operation is performed by the user. Result shows that the presented method combines the advantages of both methods. **Conclusion** The presented fast isosurface ray tracing method utilizes both K-D tree and Shell and thus can achieve a high rendering speed for fixed isosurface scene and dynamic isosurface scene.

收稿日期: 2014-07-24; 修回日期: 2014-09-23

基金项目: 国家自然科学基金项目 (11005028, 11305205, 61370167, 61305093)

第一作者简介: 罗月童 (1978—), 男, 副教授, 2005 年于合肥工业大学获工学博士学位, 主要研究领域为科学计算可视化、计算机图形学和计算机辅助设计。E-mail: ytluo@hfut.edu.cn

Supported by: National Natural Science Foundation of China (11005028, 11305205, 61370167, 61305093)

Key words: volume visualization; isosurface; ray tracing; K-D tree; Shell data structure

0 引言

等值面绘制是一种最重要的体数据可视化方法,存在两类主要实现方法:一类是等值面提取^[1],这类方法首先使用移动立方体^[1]、移动四面体^[2]等算法提取中间几何图元(等值面面片),然后利用传统面片绘制方法进行绘制;另一类是光线跟踪法^[3-4],这类方法通过从屏幕所发出射线与等值面的交点进行绘制,无需提取中间几何图元,且可以自适应屏幕的复杂性,因而越来越受关注^[5-7]。

等值面光线跟踪法的基本思想是从每个屏幕像素沿视线方向发射1条光线,然后判断光线首先进入的体素是否包含给定 Isovalue,如果有则计算光线与等值面的交点并绘制,否则进入下一个体素继续追踪,直至离开体数据或与等值面相交,如图1(a)中光线在追踪到等值面前需穿过10个体素。因为穿过每个体素都要进行求交运算,且

体数据的体素一般非常多,按基本思想实现的等值面光线跟踪法一般很慢。因此利用空间分割思想加速等值面光线跟踪,如图1(b)所示为一种空间分割方式,光线只需穿过3个区域即可达到等值面,能大幅度提高光线跟踪进度。Wald等人^[8]基于K-D树实现体数据空间分割,K-D树一旦建立可实现所有等值面的绘制,但光线跟踪过程中对K-D树需要通过层次搜索才能确定相邻区域,如图1(c)是表示体数据的K-D树,图1(b)中A、B区域在K-D树中的位置分别如图1(c)所示,当光线离开区域A进入相邻的区域B时,文献[8]要执行如图1(c)的箭头所示的层次搜索K-D树才能确定B区域。如何减少K-D树的层次搜索操作是相关工作的重要努力方向,如文献[9]和文献[10]均提出了各种改进方案,虽然有较大改善,但都无法完全回避K-D树的层次搜索操作。K-D树的层次搜索操作严重影响了这类方法的速度,尤其影响在GPU这类共享内存多核硬件的实现效率,因为GPU等硬件实现树遍历的效率很低^[11]。

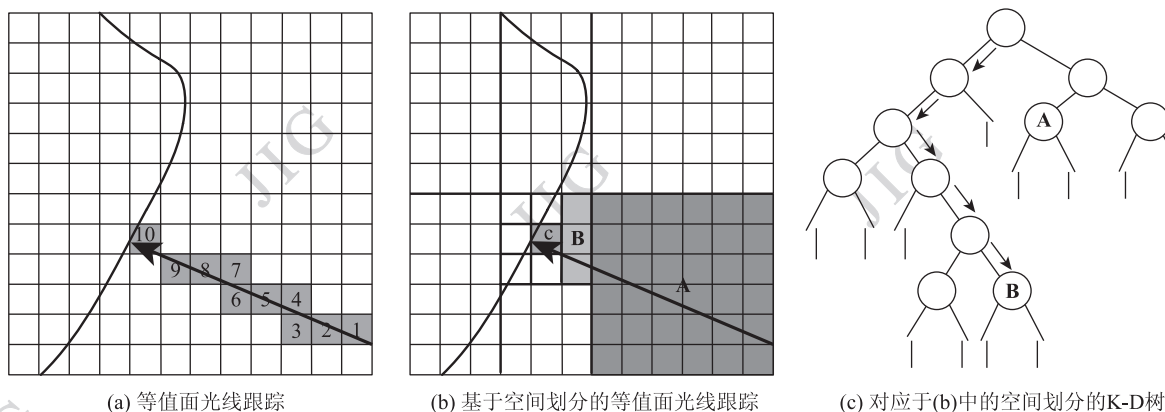


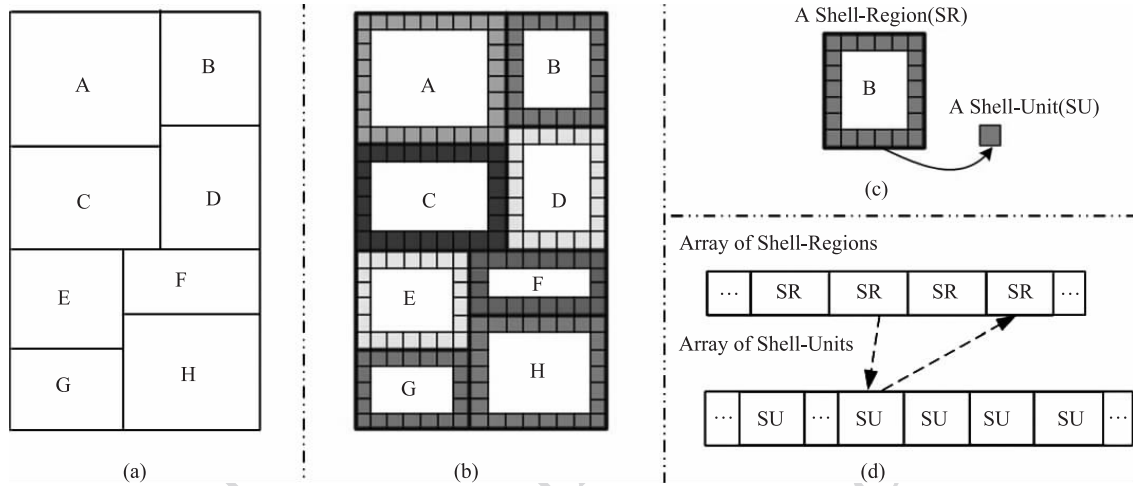
图1 不同光线跟踪法的原理

Fig. 1 The Principle of different ray tracing methods

Shell是Xiao等人^[11]所提出的一种新的空间分割数据结构(如图2所示),基于Shell数据结构可以根据光线和区域边界的交点直接确定光线要进入的下一个区域,消除了树的层次搜索操作,文献[11]也表明了基于Shell的光线跟踪法具有很高的效率。但体数据的Shell与Isovalue相关,当Isovalue变化后因需要重新建立Shell而严重影响渲染速度,所以基于Shell的等值面光线跟踪方法不能

满足动态改变Isovalue的需求。

本文的基本思想是将K-D树和Shell相结合,在等值变化后使用基于K-D树的方法绘制,等值不变时利用基于Shell的方法进行快速绘制。该方法的关键是在等值变化后如何建立Shell,通过改造基于K-D树的等值面光线跟踪算法的实现——利用基于K-D树的等值面光线跟踪算法中的信息逐步建立Shell,同时尽量少地影响渲染算法的效率。

图2 Shell 数据结构^[11]Fig. 2 Shell data structure^[11]

1 相关工作

1.1 基于 K-D 树的等值面光线跟踪法

为提高等值面光线跟踪的效率,从减少光线与体元的求交次数方面入手,基于 K-D 树的等值面光线跟踪法被提出^[8],通过空间划分体数据创建 K-D 树,在光线跟踪过程中通过计算光线与 K-D 树节点的交点情况,直接跳过一些不包含等值面的体元,实现光线的跳跃式跟踪;为了更好地体现 K-D 树的优势,进一步加快光线跟踪的速度,Foley^[9]提出了在

GPU 上的基于栈的 K-D 树遍历算法 Kd-restart 和 Kd-backtrack, Kd-restart 在查找相邻节点时,每次从根节点开始重新层次搜索 K-D 树, Kd-backtrack 在查找相邻节点时,从当前光线所在节点逐步向上回溯查找;为了更进一步减少相邻节点查找的时间消耗, Hughes 等人^[10]提出了一种基于非栈的 K-D 树回溯算法 Kd-jump,通过对分支节点的标记,在查找相邻节点时,从当前节点直接回溯到上一层的标记节点,无需逐步回溯;3 种 K-D 的节点搜索方式如图 3 所示,但这些方法都不可避免地需要对 K-D 树的层次搜索或回溯过程,仍然需要消耗较多的时间。

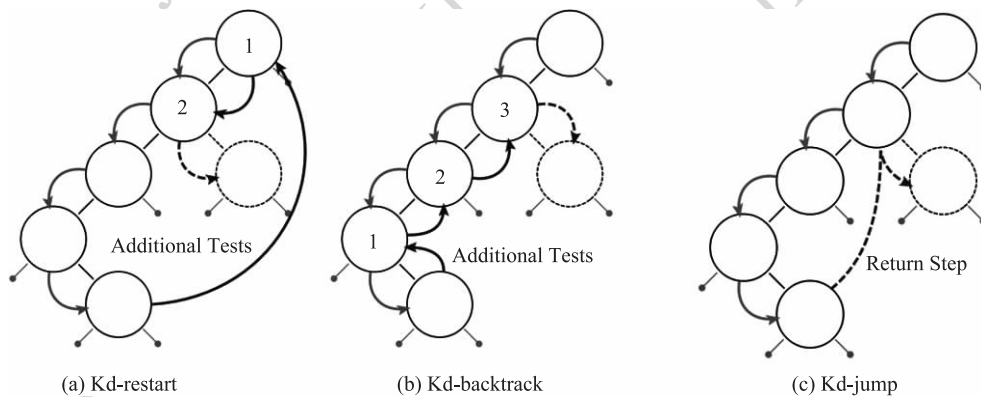


图3 K-D 树的遍历算法

Fig. 3 K-D tree traversal algorithm

1.2 Shell

Shell 是由 Xiao 等人^[9]为解决 3 维曲面的遍历问题提出的一种新的基于空间划分的数据结构,通过预先将相邻区域的相邻关系保存起来,然后在使

用时检索 Shell 直接进行相邻区域的定位。如图 2 所示, Shell 包含 Shell-Region 和 Shell-Unit 两部分, Shell-Region 用于保存划分有的区域空间, Shell-Unit 用于保存所在 Shell-Region 经由该 Shell-Unit 相邻的

Shell-Region。文献[9]同时介绍了如何在光线跟踪法中有效地使用 Shell 快速定位相邻区域,首先通过如表面启发式模型(surface area heuristic)^[12]的代价模型划分体数据空间创建 Shell-Region,然后依据 Shell-Region 创建其表面的 Shell-Unit 指向其相邻的 Shell-Region,在使用时通过查找光线与 Shell-Region 交点所在 Shell-Unit 来定位相邻的 Shell-Region,从而可以实现快速的光线跟踪。

2 K-D 树和 Shell 相结合的快速等值面光线跟踪法

高绘制效率能够确保用户对体数据友好的交

互,从而方便用户及时地观察和分析体数据。本文的目标是实现更为快速的等值面光线跟踪法,算法的总体流程如图 4 所示。首先创建 K-D 树,然后依据 K-D 树和 Shell 进行等值面光线跟踪,此过程中主要分为两部分:一是当 Isovalue 变化重置 Shell 后,此时 Shell 中数据皆为初始值,无法用于检索,使用基于 K-D 树的光线跟踪进行绘制,并同时通过更新 Shell 中数据来逐步建立 Shell;二是当 Shell 建立完成后,此时过渡到基于 Shell 的光线跟踪进行绘制。通过此方式很好地结合了 K-D 树和 Shell,实现了 K-D 树向 Shell 的平滑过渡,关键在于如何设计数据结构从而实现 K-D 树方法向 Shell 方法的平滑过渡。

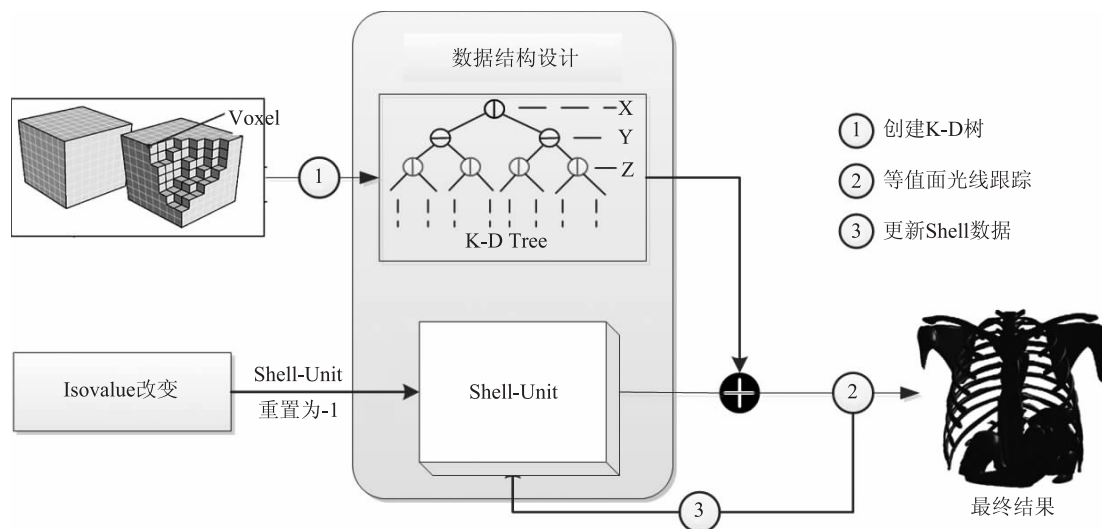


图 4 本文算法的总体流程图

Fig. 4 The overall flowchart of the proposed algorithm

2.1 数据结构设计

实现基于 K-D 树的光线跟踪向基于 Shell 的光线跟踪的平滑过渡,关键在于能够在基于 K-D 树的光线跟踪过程中利用 K-D 树的信息建立 Shell,即建立 Shell-Region 和 Shell-Unit。

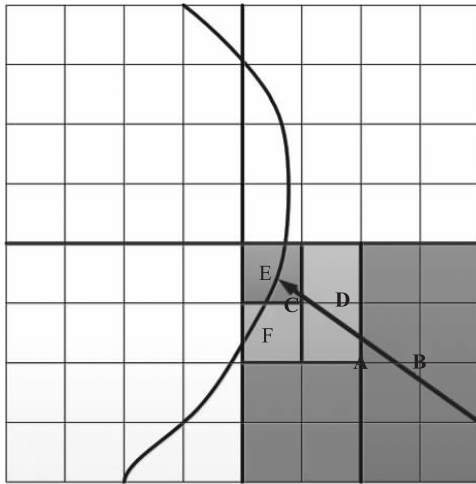
Shell-Region 和 K-D 树都是实现对体数据空间划分的方法,K-D 树是直接对体数据空间进行逐层划分,Shell-Region 则依据启发式代价模型进行最优划分,确保光线跟踪过程中相邻区域查找次数最少。依据 K-D 树和 Shell-Region 的划分原则,可以将 K-D 树节点作为 Shell-Region,为了确保 Shell-Region 的最优化,选取合适的 K-D 树节点作为 Shell-Region。满足以下原则之一则为合适的节点:

1) 节点是完全不包含 Isovalue 的具有最大间隔

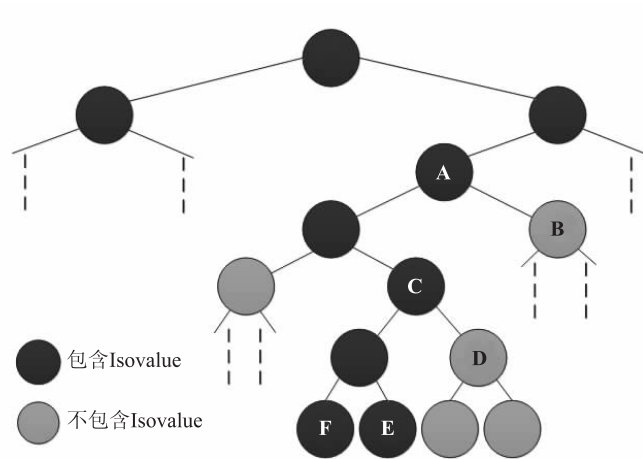
区间的节点,如图 5 中节点 B 和节点 D;

2) 节点是包含 Isovalue 的叶子节点,如图 5 中节点 E 和节点 F。

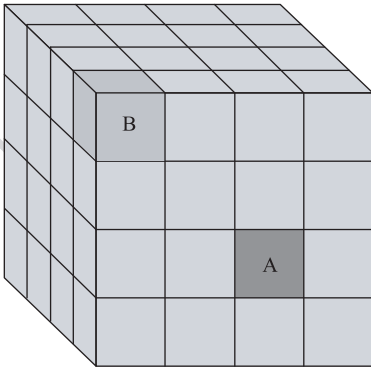
Shell-Unit 对应 Shell-Region 边界面上的最小单位,而 Shell-Region 是由体数据的体元组成的,即 Shell-Region 边界体元的外围面都是该 Shell-Region 的 Shell-Unit。如图 6(a)(b)都是被选取的满足最优节点要求的作为 Shell-Region 的 K-D 树节点,图 6(a)是完全不包含 Isovalue 的最大间隔节点,图 6(b)是包含 Isovalue 的叶子节点。图 6(a)中 K-D 树节点中间部分体元 A 只需设置最外围的 1 个面作为 Shell-Unit,而边角体元 B 则需设置 3 个面作为 Shell-Unit,而图 6(b)中作为叶子节点的体元 C 则 6 个面都需设置 Shell-Unit;从而可以看出,体元的



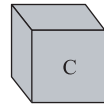
(a) 空间划分



(a) K-D树

图5 Shell-Region 的选取
Fig. 5 Shell-Region selection

(a) 最大间隔节点



(b) 叶子节点

图6 Shell-Unit 设计

Fig. 6 The design of Shell-Unit

每个面都是潜在的 Shell-Unit, 所以对 $M \times N \times K$ 的体数据存在 $6 \times M \times N \times K$ 个可能的 Shell-Unit。

综上所述, 本文算法的数据结构包括两个部分: 完整 K-D 树和 $6 \times M \times N \times K$ 个可能 Shell-Unit, 其中 K-D 树的每个节点也是 1 个可能的 Shell-Region。

2.2 结合 K-D 树和 Shell 的等值面光线跟踪

本文结合 K-D 和 Shell 实现动态的等值面光线跟踪法, 在 Isovalue 变化后使用基于 K-D 树的方法进行等值面绘制并同时逐步建立 Shell, 然后逐步平滑过渡到基于 Shell 的方法进行等值面绘制。在此过程中, 核心问题在于如何基于 K-D 树建立 Shell, 以及如何逐步由 K-D 树向 Shell 进行平滑过渡。

2.2.1 基于 K-D 树更新 Shell 数据

Shell 更新是在基于 K-D 树的等值面绘制过程

中完成的, 为了尽量减少 Shell 更新对基于 K-D 树的等值面绘制效率的影响, 利用 GPU 的并行能力在光线跟踪过程中依据每条光线进行独立的更新。Shell 更新主要分为两个部分: Shell-Region 的更新和 Shell-Unit 的更新。由 Shell-Region 的设计可知, Shell-Region 的更新是通过在基于 K-D 树的等值面光线跟踪过程中层次搜索 K-D 树, 在查找等值面所在的叶子节点同时, 判定在此过程中与光线相交的节点是否为最优节点, 标记最优节点作为 Shell-Region。本文 Shell-Region 是隐含存在的, 并未显示创建, Shell-Unit 直接指向最优 K-D 树节点。而 Shell 更新的重点在于选取了 K-D 树最优节点后如何设置相关的 Shell-Unit。

为了方便在 GPU 上使用 Shell-Unit, 将 Shell-Unit 定义为数组结构, 可以通过数组下标定位 Shell-Unit 位置。Shell-Unit 对应的是体元的面, 而每个面可通过该面的最小顶点坐标和面的偏移量唯一确定, 其对应的 Shell-Unit 在数组中的位置为

$$6 \times (z \times M \times N + y \times M + x) + offset \quad (1)$$

式中, (x, y, z) 表示最小坐标值, $offset \in \{0, 1, 2, 3, 4, 5\}$, 其中 0—5 分别对应 Shell-Unit 的上、下、左、右、前和后六个面。

为了进一步减少 Shell 数据更新对基于 K-D 树的等值面绘制效率的影响, 每次只更新光线与 Shell-Region 相交的交点的 Shell-Unit。在更新 Shell-Unit 时, 如果直接更新交点所在的节点的 Shell-Unit, 需重新层次搜索 K-D 树查找与该 Shell-

Unit 相邻的最优节点,在保证节点最优性的同时还要保证节点的相邻性,查找过程较为复杂,这并不是本文所期望的。由 Shell-Unit 的定位可知,两节点相邻时,相邻的两体元的相邻面是重合面,即它们具有相同的最小坐标顶点值,唯一可通过偏移量来区分,当重合面作为其中一个相邻体元的左面时,必是另一个相邻体元的右面,如图 7 所示, A、B 是两相邻的 K-D 树节点, A 是已选取出的最优节点, B 是存在的未显示选取出的最优节点,深色部分是它们的相邻面, a、b 分别是 A、B 相邻面上的体元的外围面,此时,如果设置 a 的 Shell-Unit,必须显示的找到 B 节点的位置,不可避免地需要额外的查找工作。由图 7 可知, a、b 面是重合面,由面的偏移量的设计模式可知,它们的偏移量的值差距为 1,即两体元重合面对应的 Shell-Unit 在数组中的位置是相邻的。计算光线与节点的交点值,对交点值取整,即为交点所在体元面的最小顶点坐标值,依据交点值亦可判断出交点所在体元面的类型,获得偏移量值,从而可计算出 a 面所对应的 Shell-Unit 位置,也就得到了与其重合的 b 面对应的 Shell-Unit 位置,将其设置指向 A 节点,即完成了与 A 节点相邻的 B 节点上 Shell-Unit 的设置,反其道而行之,避免了直接设置 A 节点 Shell-Unit 所需额外的相邻节点的查找。

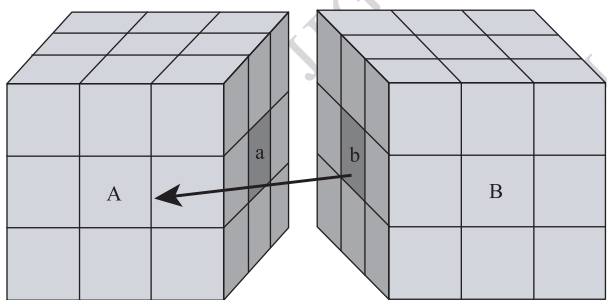


图 7 Shell-Unit 的设置

Fig. 7 Setting of Shell-Unit

2.2.2 基于 K-D 和 Shell 的光线跟踪

在 Isovalue 改变后,由于重置了 Shell 中数据,此时只能使用基于 K-D 树的方法绘制,当完成 1 次绘制后,Shell 中部分数据已更新,此时,应体现这部分数据的价值,在进行再次绘制时,不仅仅基于 K-D 树的方法,而是基于 K-D 树和 Shell 混合的方法。在进行光线跟踪查找相邻节点时,首先计算光线与当前节点的出射点坐标值定位交点所在

的 Shell-Unit,若该 Shell-Unit 值不为 -1,则表明该 Shell-Unit 已经保存了相邻节点的信息,则可以根据 Shell-Unit 信息直接定位相邻节点,若无相邻节点信息,则仍使用基于 K-D 树的方式查找相邻节点,在查找过程中进行必要的 Shell-Unit 更新。此方式很好地体现了已更新的那部分 Shell 的作用,加快了绘制效率,而随着 Shell 更新的不断完成,通过 Shell 查找相邻节点的成功率越来越高,绘制的效率也不断提升,直到 Shell 完全更新完成,此时直接转换到基于 Shell 的等值面绘制,实现了 K-D 树向 Shell 的平滑过渡, K-D 树和 Shell 混合的绘制流程如图 8 所示。

3 实验结果与分析

实验的配置环境如下: 2.93 GHz Intel I3 CPU, 2.0 G 内存, NVIDIA GeForce GTS450 GPU, 编程环境为 Microsoft Visual Studio 2008, 程序主框架采用 C++ 编写, 渲染相关功能采用 CUDA 实现, 采用了 CUDA SDK 5.5 与 OpenGL 3.1 等函数库。

选用了合成数据球、医学 CT 数据等体数据对本文算法进行了验证,由于硬件环境的限制以及该方法对 GPU 内存的较大的消耗,使用的测试数据均为 $128 \times 128 \times 128$ 。

3.1 绘制结果分析

图 9 中,红色像素部分表示该投射光线参与了 Shell 更新,否则以等值面与光线交点坐标值作为颜色。从图 9 中可看,初次绘制时,由于 Shell 还未建立,所有的投射光线都参与了 Shell 更新,而旋转 30° 后只有显示新内容的像素的投射光线参与了 Shell 更新;旋转 60° 后,由于角度偏移较大,较多像素的投射光线会穿过新的 Shell-Unit,从而参与了 Shell 更新;当旋转 360° ,回到初始位置时,由于 Shell 中数据已更新完成,光线跟踪时,只需检索 Shell 即可完成等值面定位,而无需 Shell 更新。

3.2 时间性能分析

表 1 中, Isovalue 变化是指 Isovalue 变换瞬间的绘制效率, Isovalue 不变是指 Shell 数据完全建立后的绘制效率。从表 1 可知,在 Isovalue 变化时,基于 K-D 树的快速等值面光线跟踪法的绘制效率最高,在 Isovalue 不变时,基于 Shell 的快速等值面

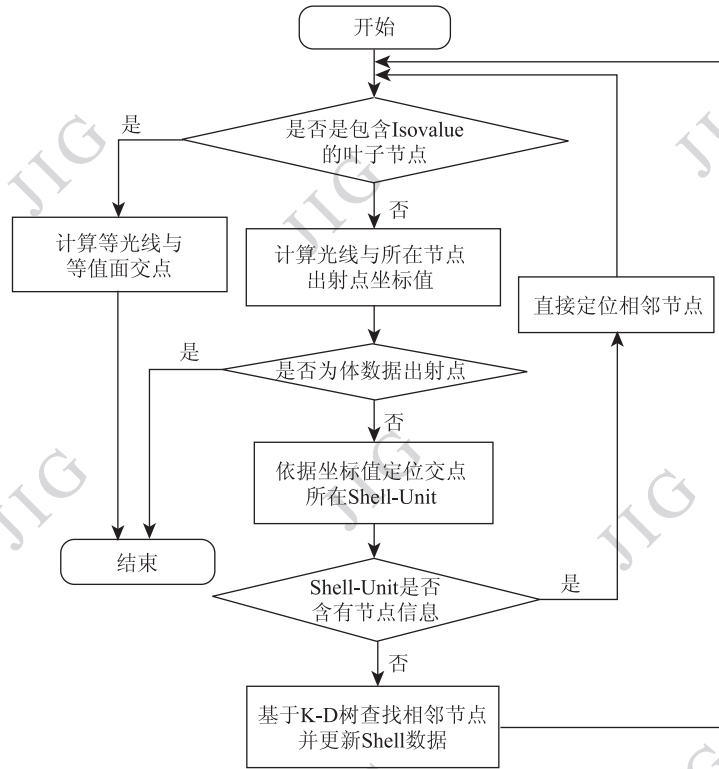


图 8 基于 K-D 树和 Shell 的光线跟踪过程

Fig. 8 The flowchart of K-D tree and Shell based ray tracing

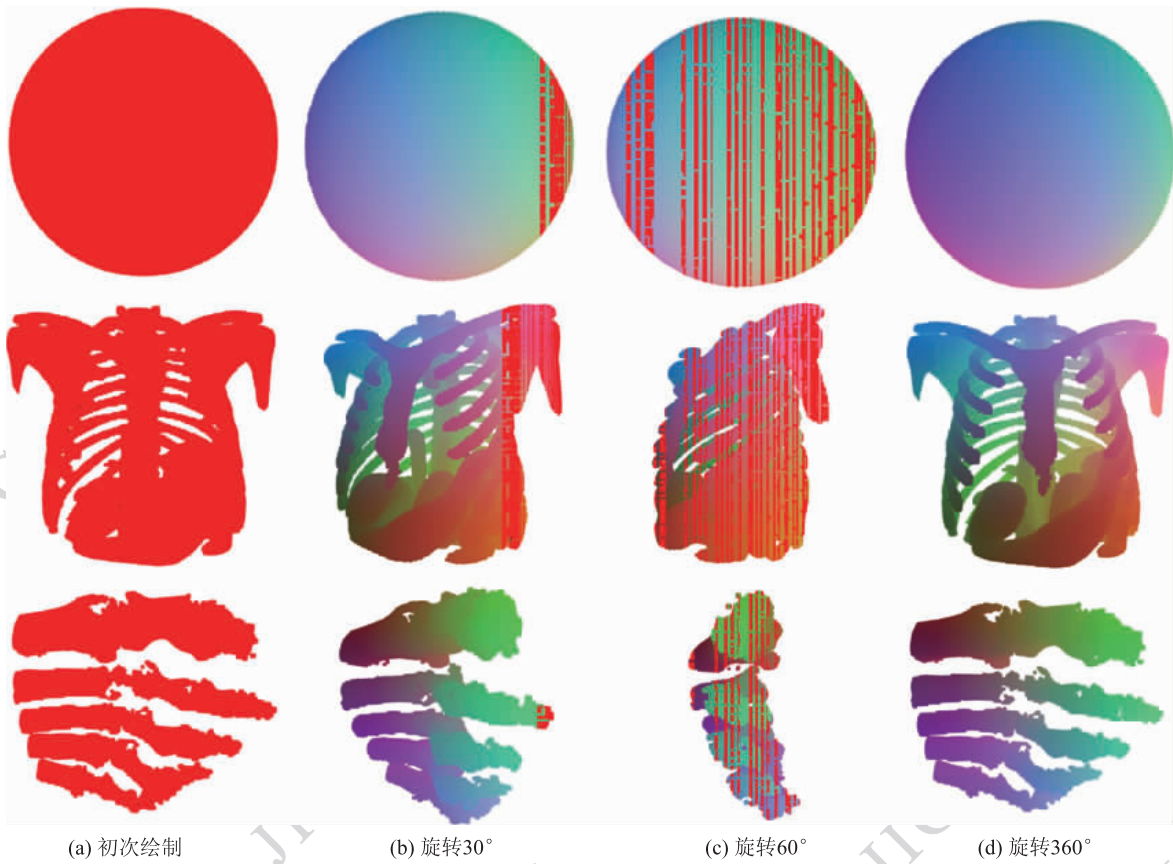


图 9 更新 Shell 过程的示意图

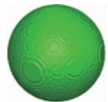
Fig. 9 Illumination of Shell updating

光线跟踪法的绘制效率最高,而本文目标是实现在 Isovalue 变化和不变时都保持最高的绘制效率。从表 1 可知,本文算法在 Isovalue 变化的绘制效率基本达到了基于 K-D 树的等值面绘制效率,而当完成了 Shell 数据建立后, Isovalue 保持不变,其绘制效率达到了原有的基于 Shell 的等值面绘制效

率。从图 10 也可以看出,本文算法在从基于 K-D 树方法绘制到基于 Shell 方法绘制的过渡过程中,由于 Shell 中数据的不断完善,绘制的效率也在不断提高,直到达到基于 Shell 的绘制效率,实现了 K-D 树向 Shell 的平滑过渡,始终保证了等值面的快速绘制。

表 1 不同绘制方法时间性能的比较

Table 1 Time performance comparison of different methods of rendering

体数据	绘制方法	效率/(帧/s)	
		Isovalue 变化	Isovalue 不变
 球	等值面光线跟踪法	38	38
	基于 K-D 树的快速等值面光线跟踪法	48	48
	基于 Shell 的快速等值面光线跟踪法	6	55
	结合 K-D 树和 Shell 的快速动态等值面光线跟踪法	45	55
 胸腔	等值面光线跟踪法	30	30
	基于 K-D 树的快速等值面光线跟踪法	42	42
	基于 Shell 的快速等值面光线跟踪法	5	50
	结合 K-D 树和 Shell 的快速动态等值面光线跟踪法	40	50
 脚	等值面光线跟踪法	32	32
	基于 K-D 树的快速等值面光线跟踪法	43	43
	基于 Shell 的快速等值面光线跟踪法	5	54
	结合 K-D 树和 Shell 的快速动态等值面光线跟踪法	40	54

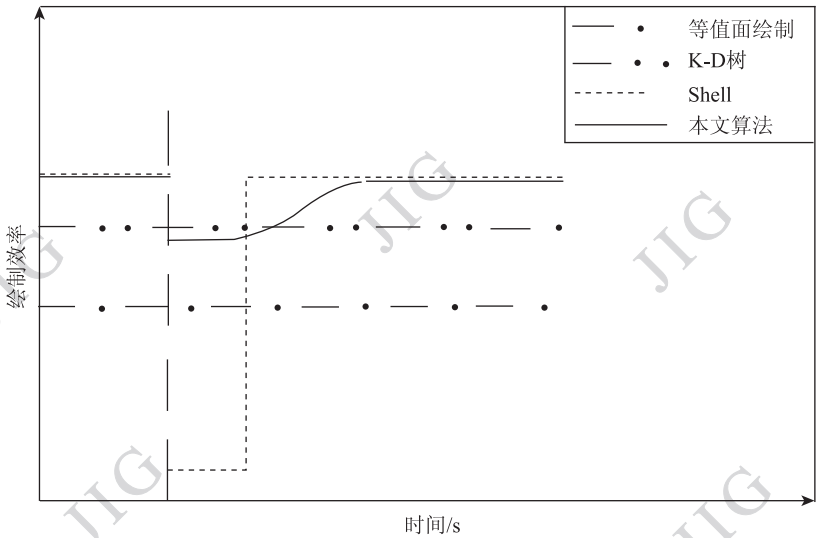


图 10 脚数据 Isovalue 变化前后时间效率对比

Fig. 10 Time efficiency comparison of before and after changing the foot data Isovalue

4 结 论

等值面光线跟踪法因无需中间几何图元、能自适应屏幕复杂性、同时进行多等值面显示等优点,而被广泛使用。本文提出结合 K-D 树和 Shell 的快速动态等值面光线跟踪法,通过在 Isovalue 变化时采用基于 K-D 树方法进行绘制并逐步建立 Shell 数据,在建立过程中使用 K-D 树和 Shell 混合的方法进行绘制,当 Shell 数据建立完成后,使用基于 Shell 的方法进行绘制,实现了 K-D 树向 Shell 平滑的过渡,很好地结合了 K-D 树适合 Isovalue 动态变化的场景和 Shell 快速定位相邻区域能力,同时避免了 Isovalue 变化时 Shell 方法绘制效率低的问题。使用合成数据、医学 CT 数据等体数据的测试结果都表明,在 Isovalue 动态变化时本文方法的效率接近基于 K-D 树方法;而用户基于缩放、旋转进行浏览时,本文算法能达到基于 Shell 的方法的速度。

本文算法需要在 GPU 上创建和保存 K-D 树和 Shell 数据,所需的内存空间较大,特别对一些较大的体数据,对 GPU 内存空间的需求问题更为突出,GPU 本身的内存空间限制了本文方法所能处理的体数据的大小,如何压缩 K-D 树和 Shell 数据的大小从而能处理更大的体数据是下一步研究的方向。

参考文献 (References)

- [1] Lorensen W E, Cline H E. Marching cubes: a high resolution 3D surface construction algorithm [C]// Proceedings of ACM Siggraph Computer Graphics. New York: ACM, 1987, 21 (4): 163-169.
- [2] Akio D, Koide A. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells [J]. IEEE Transactions on Information and Systems, 1991, 74 (1): 214-224.
- [3] Parker S, Shirley P, Livnat Y, et al. Interactive ray tracing for isosurface rendering [C]// Proceedings of the Conference on Visualization'98. Los Alamitos, CA, USA: IEEE, 1998: 233-238.
- [4] Parker S, Parker M, Livnat Y, et al. Interactive ray tracing for volume visualization [C]// ACM SIGGRAPH 2005 Courses. New York: ACM, 2005: #15.
- [5] Lin C C, Ching Y T. An efficient volume-rendering algorithm with an analytic approach [J]. The Visual Computer, 1996, 12 (10): 515-526.
- [6] Marschner S R, Lobb R J. An evaluation of reconstruction filters for volume rendering [C]// Proceedings of the Conference on Visualization'94. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994: 100-107.
- [7] Šrámek M. Fast surface rendering from raster data by voxel traversal using Chessboard distance [C]// Proceedings of the Conference on Visualization'94. Los Alamitos, CA, USA: IEEE, 1994: 188-195.
- [8] Wald I, Friedrich H, Marmitt G, et al. Faster isosurface ray tracing using implicit kd-trees [J]. IEEE Transactions on Visualization and Computer Graphics, 2005, 11 (5): 562-572.
- [9] Foley T, Sugerman J. KD-tree acceleration structures for a GPU raytracer [C]// Proceedings of the ACM Siggraph/Eurographics Conference on Graphics Hardware. New York: ACM, 2005: 15-22.
- [10] Hughes D M, Lim I S. Kd-jump: a path-preserving stackless traversal for faster isosurface raytracing on gpus [J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15 (6): 1555-1562.
- [11] Xiao K, Chen D Z, Hu X S, et al. Shell: A Spatial Decomposition Data Structure for 3D Curve Traversal on Many-Core Architectures [M]. Algorithms-ESA 2013. Berlin Heidelberg: Springer 2013: 815-826.
- [12] MacDonald J D, Booth K S. Heuristics for ray tracing using space subdivision [J]. The Visual Computer, 1990, 6 (3): 153-166.