

基于 VTK 的体绘制系统实现

李 婧, 李昌华

(西安建筑科技大学 信息与控制工程学院 陕西 西安 710055)

摘 要: 可视化是在信息化进程中发展起来的一个全新领域, VTK 是一个功能强大的可视化工具包。体绘制技术作为可视化最活跃的领域之一, 其算法不断得到改进和完善。简要介绍 VTK 的功能和应用, 描述几种不同的绘制算法, 并在此基础上用 TCL 语言实现一个简单的基于 VTK 的体绘制系统, 给出具体设计实现过程。该系统实现体绘制技术中 4 种不同的算法, 为 VTK 工具包更好地应用于可视化领域提供了实践基础。

关键词: 可视化; VTK; 体绘制; 三维重建

中图分类号: TP391

文献标识码: B

文章编号: 1004-373X(2008)12-088-03

The Implement of Volume Rendering System Based on VTK

LI Jing, LI Changhua

(School of Information & Control Engineering, Xi'an University of Architecture and Technology, Xi'an, 710055, China)

Abstract: Visualization is an emerging filed that developed in information progress. VTK is a visualization toolkits with strong functions. Volume rendering technology, as an active filed, its algorithm is improving and becomes perfect. In this paper, it introduces the function and application of VTK, describes several different rendering algorisms. It realizes a simple volume rendering system which uses TCL, and provides complete process. This system accomplishes four different algorisms about volume rendering. It provides the practical basis that VTK toolkit applied in visualization.

Keywords: visualization; VTK; volume rendering; 3D reconstruction

1 引 言

随着社会信息化的发展和网络应用的日益广泛, 人们所获得的信息源越来越庞大, 对于大型数据甚至海量数据的处理需求日益迫切。科学计算可视化(Visualization in Scientific Computing)就是在这种背景下提出并发展起来的一个全新的领域。科学计算可视化的本质是运用计算机图形学和图像处理技术, 将科学计算过程中产生的数据及计算结果转换为图像, 在屏幕上显示出来并进行交互处理, 其核心是三维数据场的可视化^[1]。体绘制是可视化最活跃的研究子领域之一, 它不借助中间图元, 而是直接将三维空间的离散数据转换为二维图像。十多年来, 在广大科研人员的努力下, 已经发展了一些体绘制技术来分析、理解和绘制数据场中包含的物体, 并且已在医学、天文、地理、气象、航天等计算机模拟中得到广泛应用。本文主要简单介绍可视化工具包 VTK 的功能及应用, 并具体实现一个基于 VTK 的利用 TCL & TK 语言开发的体绘制系统。

2 可视化工具包 VTK

VTK(The Visualization Toolkits)是一个基于面向对象方法设计的、功能强大的可视化和图形图像处理的工具箱。它是在三维函数库 OpenGL 的基础上发展起来的, VTK 构造在 C++ 语言之上, 它不仅基于 C++ 类库, 还支持脚本语言 TCL & Tk, Java, Python, 支持 Windows, Unix 等操作系统。VTK 能够支持和处理多种表示格式的数据, 如有规则的或无规则的点阵(point sets), 图像(image), 体元数据(volume)等^[2]。此外 VTK 还将在可视化开发过程中会经常遇到的细节屏蔽起来, 并将一些常用的算法进行封装, 同时, 用户还可以在 VTK 基类的基础上开发自己的类库。自从 1993 年问世以来, VTK 凭借其开放灵活的特性受到越来越多人的青睐, 不断地被改进和完善。

2.1 VTK 的绘图模式

VTK 的绘图模式可用于 2D, 3D 等一般图形的处理, 它主要有 9 种基本对象:

角色(Actor): VTKActor 代表渲染场景中的绘制对象实体, 通过调节参数可以设置角色的位置、方向、渲染特性、纹理影射等属性。

灯光(Light): VTKLight 类提供了灯光的位置、灯光的状态(开关)、亮度、颜色等。同时也可以设置光源的照

射角度以及衰减程度。

照相机(Camera): VTKCamera 类用来设置视点的位置以及焦点位置及其他属性。

属性(Property): Property 主要用来说明几何物体的一些特征。比如光照特性、着色模式、物体灰度等。

映射(Mapper): Mapper 指定数据与基本图元之间的关系, VTKMapper 的派生类通过一个向上观察板(Look-upTable)来映射数据, 同时控制与图形库相连的图元的生成。

变换(Trasform): VTKTrasform 是一个放置 4×4 变换矩阵的堆栈, 可以对这个矩阵进行平移、缩放、旋转等操作。

渲染窗口(RenderWindow): 其基类为 VTKRenderWindow, 用于在显示设备中生成一个窗口, 渲染窗口是用户图形界面, 可以设置渲染窗口的大小, 也可以产生立体显示效果, 支持双缓存。

渲染器(Render): 基类为 VTKRenderer, 主要用于控制目标的渲染过程, 管理光源、照相机位置等属性。

窗口交互(Render Window Interactor): 在 VTK 中, VtkRenderWindowInteractor 是场景交互方法的一种, 它作为一种简单的工具可以用来操作相机、选取对象、调用用户定义的函数等。

2.2 VTK 的可视化模式

可视化模式主要用于可视化过程中的数据模式如 CT, MRI 的数据输入及处理。VTK 使用数据流的方式将源数据转换为图像数据。这里包含数据对象和处理对象 2 类基本对象。数据对象用来表示各种类型的数据, 它由几何结构、拓扑结构以及属性数据(如向量、标量)组成。处理对象一般分为源处理对象(Source)、过滤器(Filter)、映射器(Mapper)三类, 它用来操作数据对象并生成新的数据对象。处理对象表示系统中对数据进行处理的方法。将数据对象和处理对象连接在一起就形成 VTK 中的可视化流水线^[3], 如图 1 所示。

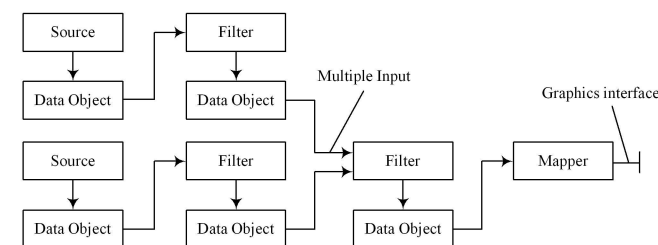


图 1 VTK 的可视化流水线

3 体绘制系统介绍与实现

这里使用 TCL &TK 语言, 利用 VTK 函数库实现一个简单的体绘制系统。通过这个系统主要介绍如何通过调用 VTK 函数库实现图像重建, 并且通过一个切片数据

bolt.slc 向大家展现不同的绘制方法所表现出的不同的效果。它可大体分为前台交互及后台计算 2 部分。其中前台交互是通过 Tk 实现的, 后台的计算则以 TCL 为开发平台, 通过对 VTK 中封装的对象及方法的调用来实现。在整个系统中, 对三维数据的绘制使用 4 种不同的方法: 最大密度投影法; 合成法; 等值面法; 二维纹理映射法。

3.1 TCL &TK

TCL 代表“Tool Command Language”, 是一种简单的脚本语言, 它几乎在所有的平台上都可以解释运行, 功能强大。TCL 主要用于发布命令给一些交互程序如文本编辑器、可视化工具软件、仿真工具等。它语法很简单, 并有很强的扩充性。TK 是对 TCL 的图形工具箱扩展。TK 发行版与 TCL 的发行版是在一起的^[4]。在创建 TK 小部件时, 几乎很少使用命令, 一半以上都是按钮或文本小部件的变体。选用 TCL &Tk 作为开发语言, 是因为它灵活的特性和简单的语法可以很方便地与 VTK 进行融合。

3.2 常用绘制算法介绍

(1) MC 算法: Marching Cubes 算法是一种应用很广泛的由体密度数据重构三维等值面的方法。MC 算法也被称为“等值面提取”(Isosurface Extraction), 其本质是将一系列二维的切片数据看作一个三维的数据场, 从中将具有某种域值的物质抽取出来, 以某种拓扑形式连接成三角面片。MC 算法流程如下: 每次读出 2 张切片, 形成 1 层, 2 张切片上下对应的 8 个点构成 1 个立方体, 然后按从左到右, 从前到后的顺序处理 1 层中的立方体, 抽取每个立方体中的等值面, 然后从下到上直至处理到 $n-1$ 层, 所以称为移动立方体算法。MC 算法属于面绘制, 也可以称为间接体绘制。

(2) 光线投射算法: 光线投射算法(Ray Casting)是直接体绘制中最常用的算法, 它的基本思想是: 从屏幕上的每一像素点发出 1 条视线, 这条视线穿过三维场的体元矩阵, 沿这条视线等距设置采样点, 将距离采样点最近的 8 个数据点所组成体素的颜色值及不透明度进行三次线性插值, 求出该采样点的不透明度及颜色值。然后可以按从前到后或从后到前的合成公式对所有采样点的颜色和不透明度进行合成。当对所有像素点都进行以上过程后, 就会得到此数据场的体绘制结果图像。

3.3 四种绘制方法分析与实现

(1) 最大密度投影法: 假设每一个体元都是能够发光的立方体, 沿着光线方向, 选择每条与体元相交的直线上的全部点中的最大的密度值作为图像投影平面密度值, 这就是最大密度投影(MIP)算法。在 MIP 描绘过程中, 主要实现光线路径上的点中最大密度点投影的功能。这一过程在 VTK 的定义中属于光线投射算法的三维数据体绘制。在 VTK 实现过程中首先通过调用 vtkSLCReader 函数读取切片数据并设置读取路径, 这里与 Slice 方法不同

的是,是通过设计转换函数实现颜色和透明度设置。在这里通过调用 `vtkPiecewiseFunction` 函数和 `vtkColorTransferFunction` 函数将体数据值转换成不透明度和颜色等光学属性,并使用 `vtkVolumeProperty` 函数对其颜色、不透明度等属性进行加载,接着调用 `vtkVolumeRayCastMIPFunction` 函数加载绘制方法,然后使用 `AddVolume` 将产生的 `mip_volume` 及通过类 `vtkCubeSource` 定义的 `Actor`; `outline_mapper` 加入到 `ren2` 中。最后调用 `vtkRenderer` 进行投影。在 MIP 中有 2 种方法可以用来改变最大值操作的行为: `SetMaximizeMethodToScalarValue()` 方法或 `SetMaximizeMethodToOpacity()` 方法。如果采用第一种方法,则沿着射线的每一个采样点的标量值均被考虑,其中具有最大标量值的采样点被选择,然后这个标量值被传递给颜色和透明度转换函数用来生成最后的射线值;如果调用第二种方法,则沿着射线的每一个采样点的不透明度都将被计算,其中具有最大不透明度的采样点将被选择。在这里采用第一种方法,即通过调用 `vtkVolumeRayCastMIPFunction` 类中的 `SetMaximizeMethodToScalarValue` 方法来实现最大密度投影设置。效果图如图 3 所示。

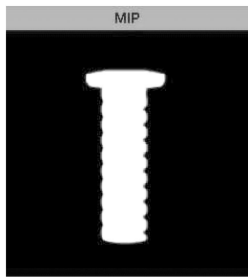


图 2 MIP 方法效果图



图 3 Composite 方法效果图

(2) 合成算法:合成算法(Composite)是对每条光线上的每个像素点的密度进行加权求和,将结果作为图像投影像素的密度值。在 Composite 描绘过程中,主要实现光线路径上所有点的密度加权求和的投影功能。这一过程在 VTK 的定义中同属基于光线投射算法的三维数据体绘制。它的实现过程基本类似于 MIP,在这里借助于类 `vtkVolumeRayCastCompositeFunction` 对取得“光线路径上所有点的密度之和的投影”这一功能做了定义。在 `vtkVolumeRayCastCompositeFunction` 中,有 2 种方法可以用来改变插值和分类的顺序,分别是: `SetCompositeMethodToInterpolateFirst` (插值优先)方法和 `SetCompositeMethodToClassifyFirst` (分类优先)方法。这种设置仅在采用三线性插值的时候才会产生影响。在第一种情况中,先执行插值,以此决定采样点的标量值,然后在将这个值用于分类;在第二种情况中,将先执行分类,之后在执行插值。插值优先通常能产生较好的效果,因此在这里采用默认的第一种插值优先法。效果图如图 3 所示。

(3) 等值面法:等值面法(Isosurface)是利用表面的信息对图像进行绘制。虽然该方法同属基于光线投射算法的三维数据体绘制,但是其本质更接近于面绘制。在实现过程中借助 `ShadeOn` 方法及 `SetInterpolationTypeToLinear` 方法对阴影功能和插值功能进行定义。并使用类 `vtkVolumeRayCastIsosurfaceFunction` 中的方法 `SetIsoValue` 设置等值面的初始值。在 Isosurface 描绘过程中,主要实现光线路径上所有密度相同的点组成的等值面的投影功能。效果图如图 4 所示。

(4) 二维纹理映射法:二维纹理映射二维纹理映射的体绘制是将体数据沿正交方向生成序列二维图像,产生二维纹理切片,然后将这些二维纹理切片依次堆叠完成体绘制过程。纹理映射算法同属于体绘制算法,因此同样需要设置不透明度、颜色、梯度 3 个传递函数来控制三维体的外观。与光线投射算法不同的是,它调用的映射类型为二维纹理映射。在实现过程中通过 `vtkPiecewiseFunction` 函数将体数据值转换成不透明度和颜色等光学属性,使用 `vtkVolumeProperty` 函数对其颜色、不透明度、阴影、插值等功能进行加载,调用 `vtkVolumeTextureMapper2D` 函数定义二维纹理映射,使用 `setCamera` 设置视点位置,并使用 `[ren1 GetActiveCamera] Zoom` 来设置放大系数,最后调用 `Render` 进行绘制。效果图如图 5 所示。

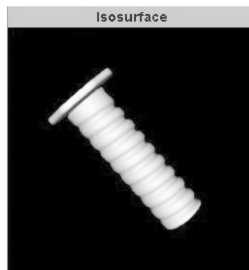


图 4 Isosurface 方法效果图

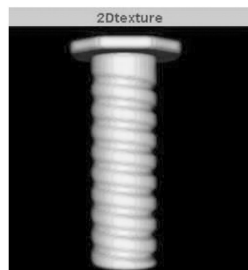


图 5 二维纹理映射效果图

3.4 结果讨论

试验结果表明, Slice 方法由于是切片投影,数据量相对比较小,因此成像速度快,并可以很清楚的观测到不同的切片面关于不同数轴的绘制效果。但是它重建的图像表面不光滑,容易出现锯齿效应,这可能是由于图像像素间的距离过大造成的,因此可通过插值计算来缓解这个现象。MIP 算法不需要太多的参数设置,实现起来比较简单,但是它对图像的细节显示得不清楚,在细节的描绘上不如 Composite 算法,这是由于投影密度值选择方法的不同造成的,因此只能用于对细节要求不高的情况下。Isosurface 算法是利用表面的信息进行重建,因此它的表面绘制效果很逼真,能够很清楚直观的显示物体表面信息。

通过 VTK 提供的封装算法,在一个普通的 PC 机上实现图像的三维重建只需几秒钟的时间,还可以利用鼠标

(下转第 97 页)

- Multicasting. In: Neuvo Y, ed. Proc. of the IEEE Int'l Conf. on Communications CICCJ. Helsinki: IEEE Communications Society, 2001; 2: 327-2: 331.
- [9] Bless R, Wehrle K. DS Multicast Router Extension, 2001, draftbless-diffserv-mcast-routerext-00. txt, July.
- [10] Agarwal A, Wang K B. Supporting QoS in IP Multicast Networks. Computer Communications, 2003 (26): 1: 533-1: 540.
- [11] Acharya A, Griffoul F. IP Multicast Support in MPLS. ATM Workshop, 1999. IEEE Proceedings, 1999, 211-218.
- [12] Ooms D, Hoebke R. MPLS Multicast Traffic Engineering, 2002, draft-ooms-mpls-multicast-te-01. txt, February.
- [13] Su H C, Hwang R H. Multicast Provision in a Differentiated Services Network[C]. 15th International Conference on Information Networking, 2001; 189-196.
- [14] Stoica I, Zhang H. Per Hop Behaviors Based on Dynamic packet States. IETF Internet Draft. < draft-stoica-diffserv-dps-00. txt >, February, 1999.
- [15] Stoica I, Zhang H. Providing Guaranteed Services without Per Flow Management. In ACM SIGCOMM' 99. <http://www.cs.cmu.edu/~hzhang> 1999.
- [16] Borgonovo F, Fratta L. End-to-End QoS Provisioning Mechanism for Differentiated Services. IETF Internet Draft. July 29, 1998. < draft-Borgonovo-qos-ds-00. txt > .
- [17] Detti A, Listanti M, Salsano. Supporting RSVP in a Differentiated Service Domains: An Architectural Framework and a Scalability Analysis. In: IEEE International Conference on Communications (ICC' 99), Vancouver, British Columbia, Canada, 1999; 204-210.
- [18] Cheng D. RSVP-TE: Extensions to RSVP for Multicast LSP Tunnels. draft-cheng-mpls-rsvp-multicast-er-00. txt, 2001. 10.
- [19] Chung J M. RSVP-TE. Extensions for MPLS Multicasting Services. 2002, draft-chung-mpls-rsvp-multicasting-00. txt, February.
- [20] Faucheur F L, Wu L, Davie B, *et al.* Multi-Protocol Label Switching (MPLS) Support of Differentiated Services, RFC3270, 2002.
- [21] Moh M, Wei B, Zhu J H. Supporting Differentiated Services with Per-class Traffic Engineering in MPLS[C]. Tenth International Conference on Computer Communications and Networks, 2001; 354-360.
- [22] Jing Z, Li L. Supporting Differentiated Services in MPLS based ATM Switches[C]. APCC/ OECC' 99, Fifth Asia-Pacific Conference on Communications and Fourth Optoelectronics and Communications Conference, 1999, 1; 91-99.

作者简介 陈琳 男, 1972 年出生, 湖北荆州人, 博士 副教授。主要研究方向为多媒体通信、计算机网络、无线网络、网络服务质量和网络安全。

(上接第 90 页)

直接拖动三维模型, 实现旋转、缩放、平移等操作, 便于用户根据需要从不同的视角观察物体, 增强了系统的交互能力。与 OpenGL 和 Matlab 相比, VTK 具有使用灵活简单、重建速度快, 可视化效果好等优点。

4 结 语

在可视化发展过程中, VTK 凭借其开源、灵活、可封装、可移植等特性被越来越多的开发人员所采用, 它极大地简化了程序, 为开发人员节省了很多的时间和精力, 在图像的三维重建中得到了广泛应用。本文所实现的系统就是一个将 VTK 应用到体绘制系统中的典型例子。VTK 将一些常用到的体绘制算法进行封装, 系统通过直接调用这些 VTK 基类实现切片数据的三维重建可视化, 通过 4 种不同的绘制方法, 展现了不同的绘制效果。这对

于帮助人们更好更深层次地分析、理解、掌握图像信息产生了极大的推动作用。随着今后信息化进程的发展, 可视化领域将会有更广阔的发展空间, 体绘制技术也将会日益改进和完善。

参 考 文 献

- [1] 唐泽圣. 三维数据场可视化[M]. 北京: 清华大学出版社, 1999.
- [2] Will Schroeder. The VTK User's Guide[M]. Kitware, 2002.
- [3] Will Schroeder. Visualization Toolkit[M]. 3rd Edition Kitware, 2002.
- [4] 王道义. TCL/TK 组合教程——双语教材+多媒体教室[M]. 北京: 电子工业出版社, 2001.
- [5] 陈琰, 梁云, 杨海新. 基于 VTK 技术的可视化研究[J]. 电脑知识与技术, 2007, 2(11): 1: 282-1: 283, 1: 304.

作者简介 李婧 女, 1982 年出生, 山东菏泽人, 硕士研究生。主要从事图像三维重建可视化方面的研究工作。

李昌华 男, 教授, 博士生导师, 现任西安建筑科技大学信息与控制工程学院院长。