

基于 VTK 的 可视化技术的研究

黄姗姗 王博亮 闵小平

关键词 可视化 面向对象 绘制 CT

摘 要 工具包(Visualization Toolkit)是一个面向对象的可视化类库,它为从事可视化工具开发的广大科研工作者提供强大的技术支持。为此,必须了解和掌握 VTK 的框架结构、构成 VTK 对象模型和组成成分,以及 VTK 在医学可视化应用中的实现。

Research of Visualization Based on VTK / Shan-shan HUANG, Bo-liang WANG, Xiao-pin MIN // China Digital Medicine. -2008 3 (1): 31 to 34

Keywords visualization, object-oriented, plot, CT

Abstract Visualization Toolkit is an object-oriented visualization class library. It provides technological support for those who work at visualization. In the paper the structure and the elements of object models of VTK are introduced. Finally a visualization example applied in iatrology for given.

Corresponding author Communication Engineering Department, Xiamen University, Xiamen 361005, Fujian Province, P.R.C. Project supported by National Natural Scientific Fund(No: 30770561)

1 引言

科学计算可视化(Visualization in Scientific Computing)^[2]是1987年由B.H.McCormick等人根据美国国家科学基金会召开的科学计算可视化研讨会的内容撰写的一份报告中正式提出来的,这标志着可视化的兴起。可视化工具包(Visualization Toolkit)就是在这样的背景下产生的。

可视化^[6]是运用计算机图形学和图像处理技术将科学计算过程及计算结果的数据转换为图形及图像在屏幕上显示出来并进行交互处理的理论和方法和技术。近年来伴随着计算机硬件水平的提高和可视化理论及方法的不断完善,可视化应用已经涉及到医学、地球科学、产品设计、建筑等各个领域。

由于可视化开发的过程中涉及到一些专业知识,所以希望能有一个专门的可视化开发工具能够简化这一过程以利于其广泛应用。本文引入了一个功能强大、使用方便、源代码开放的可视化开发工具(VTK),它大大提高了我们的工作效率。

国家自然科学基金项目批准号:30770561

厦门大学信息科学与技术学院通信工程系,在读硕士研究生,361005,福建省厦门市思明区

2 VTK 概述

VTK (Visualization Toolkit) ^[1] 是一个开放源代码、面向对象的软件系统。VTK 从功能上分为 3D 计算机图形显示 (Graphics Model)、图像处理 (Imaging Processing Model) 和可视化处理 (Visualization Model)。它包括了图形图像和可视化领域内的上百种算法, 可以跨平台使用。VTK 是一个目标库, 可以很容易嵌入到应用程序中并在此基础上开发自己的库函数, 从而建立独立的大型应用系统。

2.1 VTK 层次结构 (system architecture) VTK 主要有两个部分组成: 利用 C 语言编写的编译后的核心层 (C++ 类库) 和按照一定规则生成的支持脚本语言的解释层 (包括 Java, Tcl/Tk, vtkInterpreted Interface)。

在 VTK 编译层中, 有关图像处理相关数据结构、算法、时间关键性的功能等被封装成 C++ 对象, 为 VTK 其它类或程序所调用。编译后的 VTK 动态链接库, 在与相应脚本语言绑定以后, 形成这些解释型语言的接口, 称为解释层部分。利用解释层接口可以很方便的使用 Tcl/Tk, Python/Tk, Java AWT 等解释型语言进行快速开发或试验 VTK 的新功能, 极大的提高了效率。图 1 说明 VTK 在图像处理程序里所处的层次及内核与其它高级语言的关系。

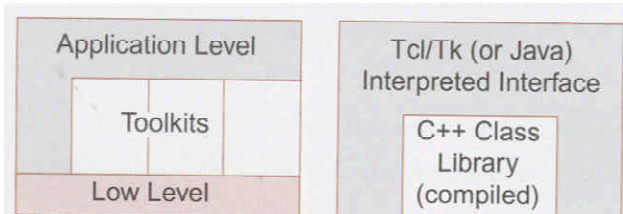


图 1 VTK 的层次框架

2.2 VTK 库的框架结构 VTK 是一个面向对象的可视化系统, 它将可视化过程中常用的算法封装起来, 隐藏了具体的细节。用面向对象技术的对象模型图表示 Visualization Toolkit 的框架结构, 如图 2 所示:



图 2 VTK 的框架结构

VTK (Visualization Toolkit) 采用的是流水线 (pipeline) 的机制, 几乎可以对任何类型的数据进行处理, 并且提供了许多相应的类对各种类型的数据进行转换或处理, 所以我们可以根据需求建立起自己的可视化流程。

vtkObject 是 VTK 的基类, 它为整个可视化流程提供基本的方法。

vtkSource 是 vtkObject 的派生类和 vtkFilter 的父类, 它

为整个可视化流程定义具体的行为和接口。

vtkFilter 是 vtkSource 的派生类, 经过 filter 的处理后, 原始的数据能够转换成可以直接用某种算法模块进行处理的形式。

vtkMapper 是 vtkObject 的派生类, 它将经过 filter 处理后的应用数据映射成几何数据, 为原始数据和图像数据之间定义接口。

VTK 有一个特点就是对于任何进行可视化的数据它都是通过角色在窗口中表现出来, 所以建立目标图形应该做的工作包括:

- 建立 actor (交互操作者), 使用 vtkActor 类来绘制。几何数据的属性通过方法 SetMapper () 告诉演员。
- 建立 renderer (演示者), 使用 vtkRender 类将结果在窗口中显示出来。演员通过方法 AddActor () 添加到演示者中。
- 建立 renderwindow (绘制窗口), 使用 VtkRenderWindow () 类来绘制。演示者通过方法 AddRenderer () 添加到窗口中。通常使用类 vtkRenderWindowInteractor 可以对目标图形进行交互操作。

2.3 VTK 图形处理 VTK 是使用面向对象技术实现的, 包含了大量的对象模型 (Object Models), 根据其继承关系, 可以将对象模型大致划分为: 图形模型对象 (Graphics Models) 和可视化模型对象 (Visualization Models)。

2.3.1 图形模型 (Graphics Models) 图形模型利用了其它 3D 图形系统的一些简单易于使用的特点, 同时也采用了一些目前的 GUI 新方法, 在图形模型中主要的对象有如下 7 种。

2.3.1.1 渲染器 (Renderer) 和渲染窗口 (Render Window)

vtkRenderer 类和 vtkRenderWindow 类都是基类, 它们结合起来管理图形和窗口系统之间的接口。vtkRenderWindow 类用于在显示设备上生成一个窗口, 可以设置窗口的大小, 也可以产生立体显示效果。支持双缓存。渲染器 (Renderer) 对象用于控制目标的渲染过程, 它把几何体, 灯光以及摄像机观察角度转换成图像的形式。当生成场景后, 要把渲染器加到渲染窗口中去:

```
vtkRenderer *ren1 = vtkRenderer::New();
vtkRenderWindow *renWin = vtkRenderWindow::New();
renWin->AddRenderer(ren1);
```

2.3.1.2 灯光 (Light) VTK 中的实现类为 vtkLight。它提供了灯光的位置, 灯光的开关状态, 亮度, 颜色。也可以设置光源的照射角度和衰减程度。

```

vtkLight *light=vtkLight::New();
light->SetColor(1,0,0);
light->SetFocalPoint(camera->GetFocalPoint());
light->SetPosition(camera->GetPosition());
在设置好光源后,要把光源加到场景(Renderer)中:
ren->AddLight(light);

```

2.3.1.3 相机(Camera) VTK中的实现类为vtkCamera。相机的作用是定义观察者的位置、聚焦点和其它有关属性,参数可由调用者根据需要设定。

```

vtkCamera *cam=vtkCamera::new();
cam->SetFocalPoint(0,0,0);
cam->SetPosition(0,0,1);
cam->SetViewUp(0,1,0);
cam->ResetCamera();

```

或者想使用一个已经存在的相机(例如演示者自动装载的相机),可以使用:

```
cam->GetActiveCamera();
```

2.3.1.4 角色(Actor) vtkActor是基类。Actor代表渲染场景中的实体,通过一个 4×4 的变换矩阵可对Actor位置、方向以及大小进行设置。还可设置Actor的渲染特性(Property)引用(例如Mapper)纹理影射(Texture)等属性。

```

vtkActor *sphereActor = vtkActor::New();
sphereActor->GetProperty()->SetColor(1,0,0);
sphereActor->GetProperty()->SetAmbient(0.875);
sphereActor->GetProperty()->SetDiffuse(0.0);
sphereActor->GetProperty()->SetSpecular(0.0);
sphereActor->SetMapper(sphereMapper);

```

2.3.1.5 映射(Mapper) vtkMapper派生类通过一个向上观察板(Lookup Table)来映射数据和控制图形库中基本图元的生成。vtkLook-upTable是vtkScalarsToColors的子类。一个或多个Actor可以使用同一个Mapper。Mapper有多个参数对其进行控制。

2.3.1.6 特征(Property) VTK中的实现类是vtkProperty。它指定了几何物体的光照及其它特性。例如:光照特性(泛光,漫反射,镜面反射,边缘颜色)漫反射强度、物体的灰度、物体的表现形式(点、线框、表面)着色模式等等。

2.3.1.7 变换(Transform) Transform是一个 4×4 变换矩阵的堆栈。有很多对这个矩阵进行操作的方法:平移、缩放和旋转等。

2.3.2 可视化模型(Visualization Models) VTK利用了数

据流程模型,在这个模型中,各个模块在网络中连接起来,利用模块对数据进行一系列操作。这种模型的特点是适用于不同的数据类型和不同的算法,所以灵活性很强。

VTK使用数据流方法把原始信息转换成图形数据。在这个方法中有两个基本对象:数据对象(Data Object)和流程对象(Process Object)。

2.3.2.1 数据对象(Data Object) VTK的数据对象有很多种数据类型,vtkDataSet类是父类,有严格的数据结构。VTK支持的数据类型有:多边形数据(vtkPolyData)结构网格(vtkStructuredGrid)非结构网格(vtkUnstructuredGrid)结构点(vtkStructurePoint)非结构点(vtkUnstructuredPoint)。

2.3.1.2 流程对象(Process Object)一般指的是过滤器(Filter),对数据对象进行操作产生新的数据对象。流程对象表示了系统的运算法则,当数据在网络中流动时对数据进行操作。流程对象可以分为:

源对象(Source) vtkSource是基类,它指定了源目标的行为和接口。源对象的产生可以有两种方法:从文件读入源数据,程序内部产生源数据。当流程处理完毕后,源对象会自动删除源对象输出的数据。

过滤器对象(Filter) vtkFilter是基类,是vtkSource的派生类。它接受源数据的输出数据,在这里需要说明,过滤器的输出数据类型可能与输入数据类型不同。

映射对象(Mapper) vtkMapper是基类,接受过滤器的对象输入的数据并把数据映射成基本图元。

2.3.3 可视化应用 用VTK进行可视化应用主要包括两个步骤。

2.3.3.1 建立合适的目标图形来演示数据

2.3.3.2 建立数据流水线(data pipeline)来处理数据,也就是把Source,Filter,Mapper连接起来

数据对象和流程对象结合起来产生了可视化流程,如图3,其中箭头指向数据流的流向。

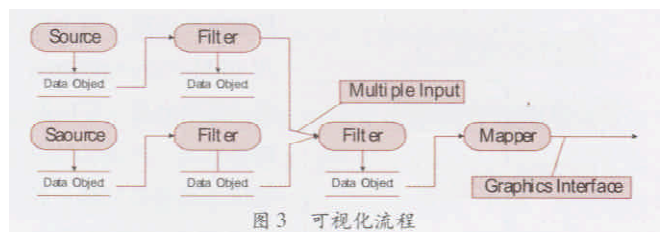


图3 可视化流程

3 VTK 可视化实例

下面用计算机断层扫描CT来分析利用VTK实现数据可视化^{[4][3]}。我们要处理的是人体头部组织的切片数据,由切

片数据重建骨骼表面。采用的算法是基于表面绘制的移动立方体法(Marching Cube)[5]。

3.1 重建算法介绍 表面绘制法是将感兴趣的部分以等值面的方式抽取出来便于利用真实感技术,通过旋转和变换光照效果生成高质量的三维图像,并可对其观察和分析。这种绘制方法速度快,适合于实时性要求高的情形,如交互操作、图像引导手术等。它适合于绘制表面特征分明的组织和器官(例如由CT数据生成骨骼三维图像)。在表面绘制算法中较为常用的是MC法(Marching Cube)。其算法是在数据体中将位于两个相邻切片上的8个相邻的体素构成一个立方体,然后逐个处理数据场中的立方体,分类出与等值面相交的立方体,采用插值计算出等值面与立方体边的交点,根据立方体每一顶点与等值面的相对位置将等值面与立方体边的交点按一定方式连接生成等值面,从而作为等值面在该立方体内的一个逼近表示。

3.2 重建流程 根据图4可以看出利用VTK进行表面绘制的流程如下。

3.2.1 读取切片数据并将其转换为开发工具 Visualization Toolkit 所支持的一种数据表达形式,然后根据其物理结构建立起相应的模型。我们给CT数据建立抽象的等值面模型。最后将物理组件与抽象模型结合起来建立CT数据的可视化,以帮助用户正确理解数据。我们所要进行处理的是有结构点阵数据,其拓扑和几何都是隐含的。所以只需要知道

数据的维数数据源和数据空间,利用Visualization Toolkit 中的 vtkVolume-16Reader 就可以很方便的读取切片数据。

3.2.2 用Marching Cubes 算法对所读取的数据进行处理 首先利用 vtkM-archingCubes 类来提取出某一CT值的等值面。但是这时的等值面其实仍只是一些三角面片,还必须由 vtkStripper 类将其拼接起来形成连续的等值面,这样就把读取的原

始数据经过处理转换为应用数据,也即由原始的点阵数据转换为多边形数据。然后由 vtkPolyDataMapper 将其映射为几何数据,并将其属性赋给窗口中代表它的演员。

3.2.3 将数据显示出来 首先利用 vtk-Renderer 类和 vtkRenderWindow 类建立绘制者和绘制窗口,然后告诉绘制者将要在绘制窗口中进行显示的演员,将绘制者加入绘制窗口。最后将绘制的结果显示出来。

最终表面绘制的结果如图5(a)(b)所示。

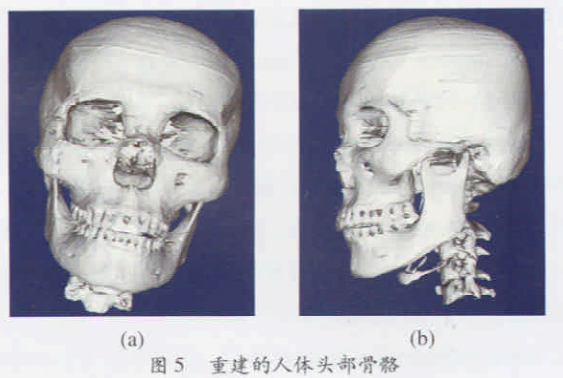


图5 重建的人体头部骨骼

4 小结

VTK 是一个用于可视化应用程序构造与运行的支撑环境,它是在三维函数库OpenGL的基础上采用面向对象设计方法发展起来的。它强大的可视化功能与MFC图形界面功能相结合可快捷开发出高性能的可视化程序。掌握好VTK对于从事可视化工作的研究者具有十分重要的意义,相信随着Web和Internet技术的发展,VTK将会有更广阔的发展前景。CDM

参考文献

- [1] 王延华,洪飞,吴恩华等.基于VTK库的医学图像处理子系统设计 and 实现[J].计算机工程与应用,2003,(8):205-207
- [2] B. H. McCormick, T. A. DeFanti, and M. D. Brown, eds., Visualization in Scientific Computing. Computer Graphics, Vol. 21, No. 6, Nov. 1987
- [3] 吕维雪.医学图像处理.北京:北京高等教育出版社[M],1989
- [4] Sabe P. A Rendering Algorithm for Visualization 3D Scalar Fields. Computer Graphics, August 1988, 22(4): 51-58
- [5] 唐泽胜.三维数据场可视化[M].北京:清华大学出版社,1999
- [6] 石教英,蔡立文.科学计算可视化算法与系统[M].北京:北京科学出版社,1996

【收稿日期:2007年10月22日】

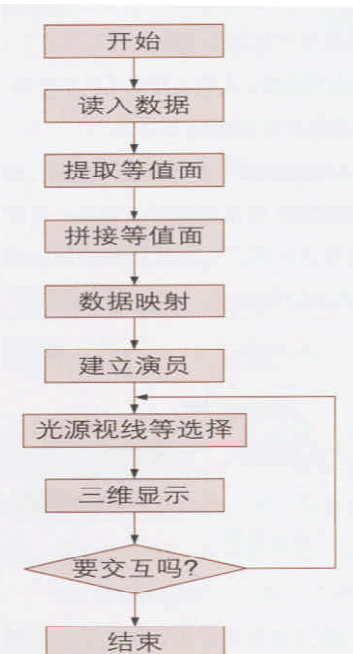


图4 利用VTK进行表面绘制的流程