

基于 VTK 技术的可视化研究

陈琰¹, 梁云¹, 杨海新²

(1.华南农业大学 信息学院, 广东 广州 510642; 2.天芝—敏迪通讯有限公司, 天津 300190)

摘要: 可视化工具包(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直接的技术支持。本文介绍了使用 VTK 环境的建立, 工作机制以及构成 VTK 对象模型的组成部分。最后给出了一个可视化的实现。

关键词: 可视化; 可视化工具包; 科学计算可视化; 面向对象

中图分类号: TP311 文献标识码: A 文章编号: 1009-3044(2007)11-21282-02

Research of Visualization Based on VTK

CHEN Yan¹, LIANG Yun¹, YANG Hai-xin²

(1.South China Agriculture University, College of Informatics, Guangzhou 510642, China; 2.Tianzhi-Mindi Communication Ltd., Tianjin 3000190, China)

Abstract: Visualization Toolkit (VTK) is an object-oriented visualization class library. It provides technological support for those who work at visualization. In the paper the environment and the mechanism, and structure of object models of VTK are introduced. Finally a visualization example for given.

Key words: visualization; visualization toolkit; object-oriented

1 可视化与 VTK 技术的产生

所谓可视化(Visualization), 《牛津英语词典》解释为: 构成头脑情景的能力或过程。X 片既为其一例。有人指出, 可视化是一系列的转换, 这种转换将原始模拟数据转换成可显示的图形图像, 这种转换的目的在于将信息转换成可被人类感应系统所领悟的格式。可视化成为一种技术和方法, 开始于利用计算机图形来加强信息的传递和理解。随后, 计算机图像处理技术和计算机视觉也成功的用来处理各种医学图像和卫星图片, 以帮助人们理解和利用各类图像数据。1998 年 2 月由 B.H. McCormick 等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了“科学计算可视化(Visualization in Scientific Computing, 简称 ViSC)”的概念[1,2], 从此标志着可视化的兴起。可视化工具包(VTK Visualization Toolkit)正是在这样的背景下产生的。

1.1 VTK 概述

伴随着可视化的广泛应用, 人们对于三维真实感图形的要求也逐步提高。机械工程师可以从二维平面图中得到自己设计的三维机械零件模型; 医生可以从病人的三维扫描图像分析病人的病理; 军事指挥员可以面对三维图形技术生成的战场地形, 指挥具有真实感的三维飞机、军舰、坦克并分析战斗方案的效果。这些三维应用已涉及建筑、产品设计、医学、地球科学、流体力学等人们生活的各个领域。为了计算机能够精确地再现这些物体, 必须能在三维空间描绘这些物体, 但这将要涉及到一些专业知识, 不利于其广泛的应用。为了便于使用, 国内外推出了许多优秀的可视化软件, 其中可视化工具包(VTK Visualization Toolkit) [3,4]以其强大的功能、方便的使用以及源码开放等特点受到了许多研究人员和组织机构的重视。

1.2 VTK 环境的建立

VTK 是由 Will Schroeder, Bill Lorensen 和 Ken Martin [4,5]三人一起开发的, 集计算机图形、图像处理和可视化处理于一体的软件系统。它容纳了图形图像和可视化领域的上百种算法, 采用面向对象技术开发, 可以跨平台使用。整个系统包括 600 多个类, 源代码超过 32500 行。图 1 中 a)图为建立一个应用程序的总体框架; b)图为 VTK 的技术框架[1,5]。

建立一个如图 1 中 a) 图的 VTK 应用程序框架, 可使用如

C++, JAVA, Tcl/Tk, Python 等多种编程语言调用, 如 b)图所示。下面主要以 JAVA 和 C++为例简要说明其环境的建立。首先从 vtk 的网站(<http://www.vtk.org/>)上下载最新的软件包安装。假定 vtkdir 为 vtk 的安装目录, 如 vtkdir=C:\vtk42。

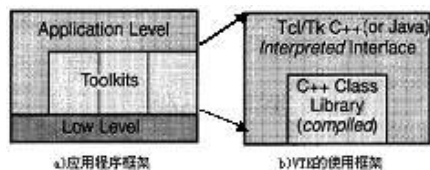


图 1 VTK 技术使用框架

VTK 在 JDK 下的使用方法:

- (1) 安装 JDK 到 C:\j2sdk1.4.1_02 上。
- (2) 设置 PATH: path=..., C:\j2sdk1.4.1_02\bin; vtkdir\bin;
- (3) 设置 CLASSPATH: classpath=.; D:\j2sdk1.4.2_04\lib\dt.jar; D:\j2sdk1.4.2_04\lib\tools.jar; vtkdir\bin\vtk.jar

(4) 编译 > javac cone.java

(5) 执行 > java Cone

VTK 在 C++中的使用方法:

(1) 安装 C++

(2) 设置 PATH: path=..., vtkdir\bin;

(3) 设置 vc: lib dictionary
include dictionary

(4) 添加 lib:

vtkjpeg.lib vtktiff.lib vtkzlib.lib vtkpng.lib vtkftgl.lib vtkexpat.lib
vtkimaging.lib vtkGraphics.lib vtkParallel.lib vtkhybrid.lib vtkFiltering.lib
vtkio.lib vtkrendering.lib vtkcommon.lib

2 VTK 的使用方法研究

VTK 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, 大致可以将其对象模型划分为两个方面[6,7], 图形模型对象(the Graphics Models)和可视化模型对象(the Visualization Models)。

2.1 图形模型(the Graphics Models)

整个图形模型主要表现了三维图形系统的本质特征, 利用了其它 3D 图形系统简单易于使用的特点, 同时也采用了一些目前已有的 GUI 的方法。整个图形模型主要分为九类对象:

收稿日期: 2007-04-08

作者简介: 陈琰(1977-), 女, 回族, 宁夏吴忠人, 华南农业大学信息学院, 助教, 硕士, 主要研究方向为多模人机交互及虚拟现实技术; 梁云(1981-), 女, 山东人, 华南农业大学信息学院, 助教, 硕士, 主要研究方向为图形图像处理技术; 杨海新(1979-), 男, 广东广州人, 天芝—敏迪通讯有限公司, 技术员, 学士, 主要研究方向为网络技术。

(1)渲染控制器: 定义与设备无关的坐标计算方法, 创建绘制窗口;

(2)渲染窗口(RenderWindow): 其基类为 vtkRenderWindow, 管理显示设备上的窗口, 一个或多个绘制方法可在渲染窗口上创建一个场景。渲染窗口是用户图形界面, 其中包括了设置渲染窗口的大小, 产生立体显示效果等的方法, 一共控制控制两个缓存;

(3)渲染器(Renderer): 基类为 vtkRenderer, 作用是管理光源、照相机和绘制对象等的位置、属性等; 提供了世界坐标系, 观察坐标系及显示坐标系之间的转换。建立好一个 Renderer 后将其加入到 RenderWindow 中即可将场景显示出来;

(4)灯光(Light): VTK 中的实现类为 vtkLight, 可在场景中照亮绘制对象; vtkLight 可通过调用参数的改变控制灯光的状态(开/关)、照射角度、照射强度、颜色等, 并支持点光源和平行光源。创建好一个灯光后将其加入到 Renderer 中即可;

(5)照相机(Camera): VTK 实现类为 vtkCamera, 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指定;

(6)角色(Actor): vtkActor 代表渲染场景中的绘制对象实体, 通过参数的调节可以设置角色的位置、方向, 渲染特性(Property), 引用(reference), 纹理影射(Texture)等属性, 并可对 Actor 进行放缩。角色的改变是通过一个 4×4 的变换矩阵实现的;

(7)属性(Property): VTK 实现类为 vtkProperty。要使三维物体具有真实感, 必须说明几何物体的一些特性, 例如: 光照特性(泛光, 漫反射, 镜面反射, 边缘颜色), 反射强度, 物体的灰度, 物体的绘制样式(点, 线框, 表面), 着色模式(flat, Gouraud, Phong)等; VTK 中正是通过属性模型来对其进行说明的。

(8)映射(Mapper): vtkMapper 指定了渲染数据和图形库中基本图元之间的联系。VtkMapper 的一些派生类通过一 LookupTable 映射数据并控制图形库中相应 Actor 图元的生成。一个或多个 Actor 可以使用相同的 Mapper。Mapper 有多个参数对其进行控制, 如 ScalarVisibility 标志可以设置 scalar 的数据是否影响相关的 Actor 的颜色; ScalarMode 可以控制 Actor 的颜色是 scalar point 还是 cell 的值; 渲染方式可以通过 (ImmediateModeRenderingOn) 和 (ImmediateModeRenderingOff) 控制, 前一项表示立即渲染, 后一项表示将渲染放于渲染列表中, 一般选择 ImmediateModeRenderingOn 方式进行, 因为如果选择 ImmediateModeRenderingOff 方式, 将会占用较多的系统资源, 影响系统的性能;

(9)变换(Transform): vtkTransform 是一个放置 4×4 变换矩阵的堆栈, 可对这一堆栈进行各种操作, 如: 平移, 缩放, 旋转等, 一般在栈的顶部进行。

实现过程主要由渲染控制器创建渲染窗口, 由渲染器在渲染窗口上绘制场景。场景包括绘制对象、映射、属性、变换、光源、照相机等。VTK 获取数据的方式是基于拉技术, 即在管道的下一对象请求数据时, 管道对象才进行数据集的相关处理, 因此可减少系统的运算负荷, 这对提高系统实时性大有好处。

2.2 可视化模型(the Visualization Models)

用 VTK 进行可视化应用是非常方便的, 它包括两个基本部分。首先, 建立适当的目标图形来演示数据; 其次, 建立一个数据流水线(data pipeline)来处理数据, 建立流水线(pipeline)就是将 Source、Filters 和 Mappers 连接起来。VTK 的可视化模型主要包括两类对象:

(1)数据对象

VTK 共有 5 种数据对象, 如图 2, 包括多边形数据(vtkPolyData), 结构点数据(vtkStructurePoint), 非结构点数据(vtkUnStructurePoint), 结构网格(vtkStructureGrid), 非结构网格(vtkUnStructureGrid)。

(a)多边形数据(vtkPolyData): 表示由顶点、直线、多边形即三角面片组成的几何体, 支持多种的原子类型, 如 vtkVertex, vtkPolyVertex, vtkLine 等。

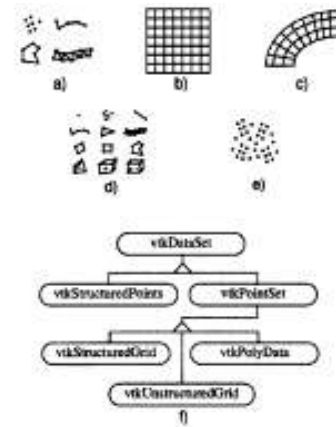


图2 VTK 的 5 种数据对象

(b)结构点数据(vtkStructurePoint): 是一包括表面形状及几何形状的几何体。

(c)非结构点数据(vtkUnStructurePoint): 指定了几何体的外观; 结构网格(vtkStructureGrid): 指定了几何体的结构。

(d)非结构网格(vtkUnStructureGrid): 可以为任意的 cell 类型的组合。

(e)数据对象继承关系。

(2)过程对象

VTK 中定义的过程对象根据其 pipeline 主要包括了数据源(Source), 过滤器(Filters)、映射(Mappers), 数据流水线(datapipeline), 过程如图 3 所示:

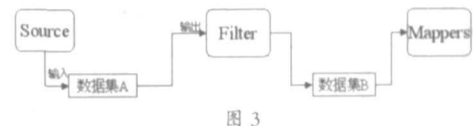


图3

数据源(Source): vtkSource 是所有数据源的基类, 其子类定义了许多数据源类型;

过滤器(Filters): vtkFilter 是各种 Filter 的基类, 从 vtkSource 中派生出来, 接收 Source 中的数据, 进行各种不同的 Filter 处理工作。Filters 为 VTK 的主要部件, 由其基类派生出了许多子类, 实现了许多图形学算法。例如网格细化(vtkSubdivideTetra), 压缩(vtkShrinkFilter), 映射(vtkReflectionFilter), 旋转(vtkRotationalExtrusionFilter), 提取边(vtkExtractEdges)等方法, 将其封装起来, 用户只需要编写简单程序接口调用就可, 并可以通过改变其参数来达到想要的效果;

映射(Mappers): vtkMapper 是所有 Mappers 的基类, 从 Filter 接受数据, 并把其映射为图形库中的基本图元。根据映射方式的不同, 有多个继承子类。

3 结论

VTK(Visualization Toolkit) 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的设计方法发展起来的, 它将我们在可视化开发过程中的一些常用的算法封装起来。以用户使用的方便性和灵活性为主要原则, 不仅具有强大的三维图形功能, 而且其设备的无关性使其代码具有良好的可移植性。从而在极大的改善可视化效果的同时又可以充分利用现有的图形库和图形硬件, 并能够更好的支持基于网络的工具比如 Java 和 VRML。掌握 VTK, 对于从事可视化技术的人员具有极好的应用价值。随着 Web 和 Internet 技术的发展, VTK 将会有着更好的发展前景。

参考文献:

[1] B. H. McCormick, T. A. DeFanti, and M. D. Brown, eds, "Visualization in Scientific Computing". Computer Graphics, Vol. 21, No. 6, Nov. 1987.

(下转第 1304 页)

4.2 计算开销

每次某个节点变化,则这个与节点相邻的所有节点与之共享数据的密钥都要更新。由于本方案采用的是二叉树的方式排列用户。因此每个用户的邻居节点集 E_{ij} 的数目 [1,3]。也就是说每当发生节点变化时只会有 1 至 3 个用户会发生变化。其他的用户则不用做改变。

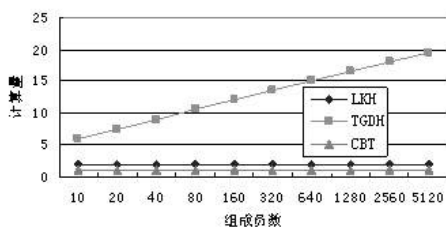


图5 成员加入时剩余组成员计算量对比

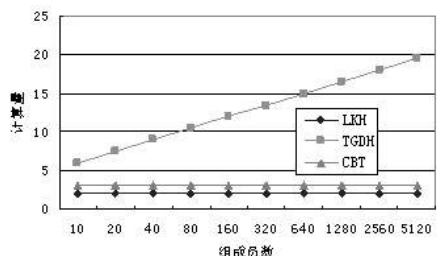


图6 成员退出时剩余组成员计算量对比

4.3 通信开销

当成员改变,组密钥需要更新的时候,GC产生新的组密钥 K_G 需要传送给所有组成员。对于每个客户节点,它只需要发送 K_G 给其子节点,而其子节点数最大为 2,因此,其最大的通信量为 2。而对于整个系统来说,把 K_G 传送给所有成员需要的通信次数为二叉树的边数 $n-1$ 。

CBT 有与其他方案不同的特点,就是新的组密钥 K_G 必须从根节点的组成员开始往下传递。而其他的方案都是用已知的参数直接计算组密钥,或者通过 GC 计算后传出去。在这里 CBT 的二叉树是完全二叉树,可以保证一定数量 n 个成员的情况下,树的深度最小。这里用 h 表示密钥树的深度。因为深度为 h 的二叉树至多有 $2^h - 1$ 个节点,当 $n = 2^h - 1$ 时可以计算出 $h = \log_2(n+1)$ 。也就是说,要经过 $\log_2(n+1)$ 次传播所有的组成员将得到新的 K_G 。这里所传播的层数是和 n 呈对数关系,因此,当 n 快速增长时 h 的增长幅度是很小的。在同样使用完全二叉树的情况下,CBT 所有的节点,都表示组成员,而 LKH, TGDH 只能用叶子节点表示组成员。因此 CBT 树的深度是最小的, $H_{CBT} < \min(H_{LKH}, H_{TGDH})$ 。对比 LKH 需要经过 H_{LKH} 次计算, TGDH 需要经过 $3H_{TGDH}/2$ 次计算才能得到新的组密钥,而 CBT 的传输次数比以上两种的计算次数都要少。其他的方案,比如 TGDH 里为了发布所有的 BK 也需要传送数据,有相应的传输次数。因此,虽然 CBT 存在数据传送的一个额外过程,整体密钥更新的速度不会比其他的方案慢。

我们用仿真实验来测试在理想环境下 CBT, TGDH 密钥更新时的密钥传输速度,并且做了比较。实验环境如下: 采用 java

(JDK1.5), windows2003(sp1)。每个成员都能够单独运行,因此我们用 java 程序产生多个线程代表组播中的成员,所有线程都是并发的,有效地模拟了实际的多用户环境。我们采用满二叉树来排列 CBT, TGDH 里的组成员。因为满二叉树可以用一维数组来存储,每个节点要找到自己所需节点的数据可以用简单的数组寻址访问完成,而不用像普通二叉树里那样进行遍历操作,极大地提高了运行速度。我们用 CBT, TGDH 里的算法对二叉树进行更新,观察更新的速度。我们用到了代表组成员数的 10 组数据,每组数据运行 5 次,取平均值,实验结果图 7。

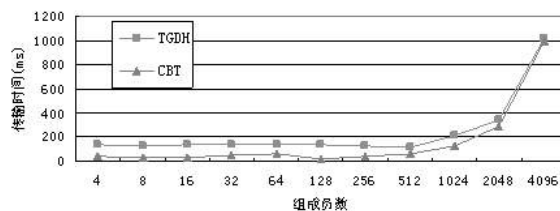


图7 CBT 和 TGDH 传输时间对比

4.4 系统的安全性分析

本方案里的要实现的安全性是组播成员变化时的前向,后向加密问题。当新成员加入组播时,GC 寻找一个位置安排这个新成员,然后该成员和与之相邻的 $m(1 \leq m \leq 3)$ 个节点进行密钥协商。经过 $\log_2(N+1)$ 轮传播,所有的成员都得到新密钥,保证以前的数据不会被新成员解密得到实现前向加密。与此相似,当新成员退出组播时也要及时更新密钥保证后向加密,同样经过 $\log_2(N+1)$ 轮次传播,所有成员计算出新密钥,保证了组播内容的机密性。

5 结论

我们给出了一种新型的组播密钥管理方案,给出了其详细的数学定义、原理及其实现方法。之后从存储开销,计算开销和通信开销出发,给出了该方案的性能分析,并将该方案与已知的几种经典方案做对比。从对比中可以看出,本章所述方案的具有比以往的方案更高效性和简单性的特点。最后,我们给出了所述方案的安全性分析。毕竟一种密钥管理方案的最终目的是要实现组播数据的安全访问。我们的方案也保证了最基本的前向加密、后向加密问题,因此,是一种安全可靠的方案。

参考文献:

- [1] Thomas Hardjono, Gene Tsudik. IP Multicast Security: Issues and Directions. Annales de Telecom, 2000, 6:324- 340
- [2] 徐明伟,董晓虎,徐恪.组播密钥管理的研究进展[J].软件学报,2004,15(1):141- 150.
- [3] Thomas Hardjono, Gene Tsudik. IP Multicast Security: Issues and Directions. Annales de Telecom, 2000, 6:324- 340.
- [4] Wong C, Gouda M, Lam S. Secure Group Communications Using Key Graphs. IEEE/ACM Transactions on networking, 1999, 8(1): 16- 30.
- [5] Kim Y, Perrig A and Tsudik G. Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups. In: 7th ACM Conference in Computer and Communication Security 2000. Jajodia: ACM Press, 2000, 235- 241.

(上接第 1283 页)

- [2] Nahum D. Gershon, Stephen G. Eick. "Information Visualization", IEEE Computer Graphics and Applications, 7- 8/1997, 29- 31.
- [3] The design and implementation of an object-oriented toolkit for 3D graphics and visualization Schroeder, W.J.; Martin, K.M.; Lorensen, W.E.; Visualization '96. Proceedings, 27 Oct. - 1 Nov. 1996 Pages:93- 100, 472.
- [4] Will Schroeder, Ken Martin & Bill Lorensen. The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics[M]. 2nd

Edition. PrenticeHall old Tappan, N.J, 1998.

- [5] <http://www.vtk.org/>.
- [6] Visualizing with VTK: a tutorial, Schroeder, W.J.; Avila, L.S.; Hoffman, W.; Computer Graphics and Applications, IEEE, Volume: 20, Issue: 5, Sept.- Oct. 2000 Pages:20- 27.
- [7] An application architecture for large data visualization: a case study, Law, C.C.; Henderson, A.; Ahrens, J.; Parallel and Large-Data Visualization and Graphics, 2001. Proceedings. IEEE 2001 Symposium on, 22- 23 Oct. 2001 Pages:125- 159.