

海洋环境三维场景关键技术研究*

宋 蒙¹, 苏天赞^{2**}, 王国宇¹, 李家钢³, 李新放²

(1. 中国海洋大学, 山东 青岛 266100; 2. 国家海洋局第一海洋研究所, 山东 青岛 266061; 3. 中海石油研究总院, 北京 100027)

摘要: 为了满足海洋科研和工程开发工作对海洋环境进行直观模拟和表达的应用需求, 本文运用 MFC 交互框架和 OSG 三维渲染引擎, 研究节点访问器、回调机制、事件处理以及消息传递等三维交互关键技术, 从地形模型构建、海水仿真、坐标映射以及场景剪裁等几个方面对海洋环境三维场景构建和渲染技术进行研究和开发, 实现了界面友好的海洋环境三维交互场景, 以便进一步对海洋环境特征和现象进行仿真和模拟。

关键词: OSG; MFC; GPU; 海洋环境仿真; 三维可视化

中图分类号: P715

文献标志码: A

文章编号: 1672-5174(2014)08-108-06

约占地球表面积 71% 的海洋, 与人们的生活息息相关, 在陆地资源日益紧缺的今天, 海洋已经成为当今世界关注的焦点。随着海洋科研、勘探、能源开发活动的开展, 获得的海洋环境信息迅猛增长, 如何对这些信息进行可视化仿真和分析, 从而更加直观地认识和理解海洋, 是目前研究的热点。近些年来, 海洋环境可视化研究工作主要包括海底地形建模和海水仿真 2 个方面。在海底地形建模方面, Lindstrom 等基于规则网格数据和层次细节模型对海底地形进行可视化建模^[1-2]; Lee, Schachter 等基于不规则三角网和层次细节模型实现海底地形的建模和绘制^[3-4]。海水仿真建模主要集中在利用海流、海温、盐度、密度等信息进行水体要素表达, 其中 Foster 通过求解流体 Navier-Stokes 方程真实的模拟了海面^[5]; Tessendorf 采用 FFT(快速傅里叶变换)反演海浪谱^[6]。目前, 这些研究和应用工作大多集中在海洋环境要素的独立建模和绘制, 如海底地形、海面、海流、海浪等, 没有表现出海洋环境整体效果, 对于面向空间要素交互建模和显示的海洋环境三维场景可视化系统的研究工作并不多见。Google Ocean 虽然可以大范围的展现海底地形, 但没有实现对海洋水体要素的可视化表达; 北京国遥新天地推出的 Ev-Globe 软件在海面模拟、海洋信息可视化方面作了一些尝试; osgOcean 能够模拟大范围海面、海底环境, 但是对显卡要求较高, 并且渲染速度和交互方式不理想。

本文采用 VC++ 开发平台和 OSG 可视化库, 将 MFC 人机交互的优越性与 OSG 的高效渲染特性相结

合, 基于现有的研究成果, 对海洋环境三维场景构建关键技术进行研究, 将用户交互框架、海底地形与海水等要素进行集成, 构建海洋环境三维可视化场景, 为海洋科研、开发和管理人员提供 1 个实时、逼真、可交互的海洋环境要素可视化模拟和仿真工具。

1 系统总体层次框架

海洋环境三维场景可视化系统主要由以下 3 个层次: 人机交互层、显示控制层、绘制渲染层。人机交互层利用 MFC 提供的菜单栏、工具栏、对话框、浮动栏等友好的用户交互界面, 方便用户的控制参数输入和信息反馈; 显示控制层主要通过 OSG 三维引擎利用树状结构对场景数据进行组织管理, 采用节点访问、回调机制、场景交互控制关键技术为用户提供 1 个实时、动态、可交互的三维环境; 绘制渲染层主要对海底地形、海水以及其他海洋环境要素进行绘制和渲染, 如对海底地形高程值以不同颜色进行着色、实现地形光照阴影效果、实时海面动态波光粼粼效果、对海底局部场景进行三维剪裁和构建等, 实现海洋三维交互场景。

2 三维交互关键技术

2.1 MFC 与 OSG 集成

OpenSceneGraph(OSG)是 1 个基于 OpenGL 扩展封装的开源可视化图形库, 使用 C++ 语言编写而成^[7]。OSG 在封装的基础之上, 建立 1 个面向对象的框架, 主要为图形图像应用程序的开发提供场景管理和图形渲染优化功能, 使得编程者可以摆脱底层的繁

* 基金项目: 海洋公益性行业科研专项“‘山东半岛蓝色经济区’建设的海洋空间布局优化技术体系及决策服务系统应用示范”项目(201205001); 国家科技重大专项“南海北部陆坡地区地质灾害风险评价预测研究”项目(2011ZX05056-001-01)资助

收稿日期: 2013-03-11; 修订日期: 2013-12-10

作者简介: 宋 蒙(1986-), 女, 硕士生。E-mail: smengq@163.com

** 通讯作者: E-mail: sutianyan@fio.org.cn

杂建模,更便于应用程序的开发和管理,作为一款高性能的、开源的渲染引擎,被广泛应用于虚拟现实、虚拟仿真、科学和工程可视化等领域。

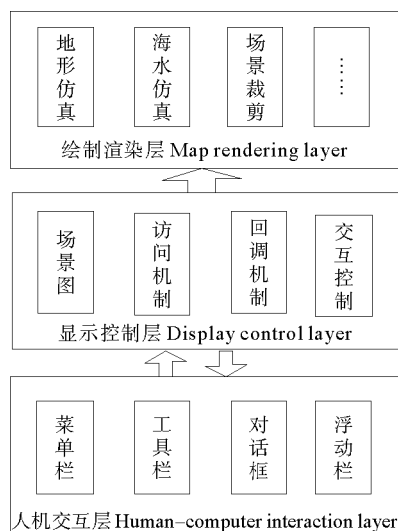


图1 系统总体层次框架图

Fig.1 The overall level frame system

OSG 提供了场景管理和图形渲染有关的可视化接口和对键盘与鼠标等设备的操作器接口,但是缺乏用户交互界面,不便于用户进行对象管理、参数设置、消息传递与控制等操作。MFC(Microsoft Foundation Classes)是微软公司提供的类库,以 C++ 类的形式封装了 Windows 的 API,并且包含 1 个基于文档/视图结构的应用程序框架,程序的数据由文档对象来维护,通过视图对象提供给用户^[8]。因此,基于 MFC 用户交互框架,集成 OSG 可视化图形库,能够构建具有友好用户交互界面的三维可视化场景。针对 MFC 和 OSG 各自的功能特点和内部消息机制,需要在 MFC 框架的视图类(CView)中定义一个 COSG 类对象 mOSG,负责对三维可视化场景中的对象进行管理和渲染,MFC 框架负责提供交互界面、传递消息和参数给 mOSG 对象、显示 mOSG 对象输出的渲染场景等功能。MFC 和 OSG 集成的流程图见图 2。

2.2 场景图构建

场景图又称视图图,是一种简单有效的用来组织和存储三维场景的数据结构,它保存着场景中所有物体及其相互关系,场景图形采用一种自顶向下的、分层的树状数据结构来组织空间数据集,以提升渲染的效率。OSG 三维场景管理的基本单位是节点,节点的类型有组节点(Group)、节点(Node)、叶节点(Geode)。最顶端是根节点,根节点具有一个或多个子节点,各个子节点又有零个或多个子节点。叶节点(Geode)只能作为子节点,不能包含其他节点。场景组织结构见图 3。

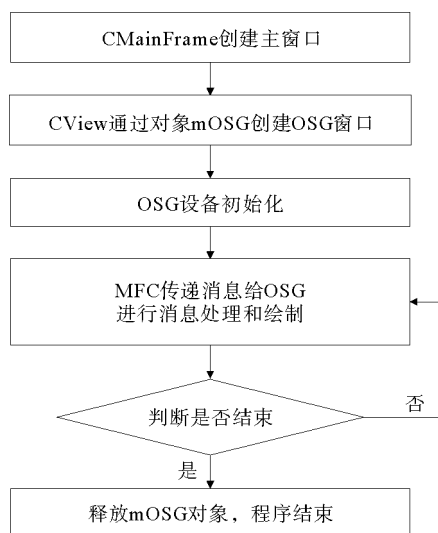


图2 MFC与OSG集成流程

Fig.2 Integrated process of MFC and OSG

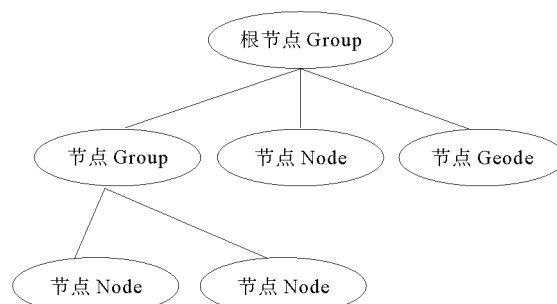


图3 场景图构建

Fig.3 Construction of scene graph

2.3 访问机制

在三维交互过程中,需要快速地从场景中找到目标节点进行查询、渲染和移动等操作。利用 OSG 提供的 NodeVisitor 类,能够遍历访问整个场景的每个图形节点,进而对指定的节点进行具体的操作。遍历的方式有 TRAVERSE_ALL_CHILDREN(向下遍历所有节点)、TRACERSE_ACTIVE_CHILDREN(向下遍历活动节点)、TRAVERSE_PARENTS(向上遍历父节点)和 TRAVERSE_NONE(只遍历本身节点)等。访问机制最关键的 2 个函数是 apply 和 accept。accept 函数将访问器与要访问的对象进行关联,apply 函数决定遍历的方式。用户可以创建新的继承于 NodeVisitor 的类,重写 apply 函数来改变遍历方式,采用“双重分派”技术(见图 4)来实现访问器与节点之间的交互。

2.4 回调机制

OSG 通过回调类来实现对场景图形节点的更新和用户与场景对象的交互。回调可以被理解成是一种用户自定义的函数,根据遍历方式的不同(如更新、拣选、绘制等),回调函数将自动地执行。回调可以与个别的

节点或者选定类型(及子类型)的节点相关联。在场景图形节点的各次遍历中,如果遇到的某个节点已经与用户定义的回调类和函数相关联,则这个节点的回调将被执行。下面,以创建一个粒子系统为例,利用回调机制实现粒子的运动(见图5)。

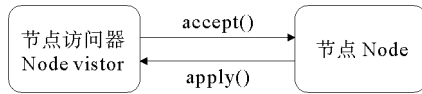


图4 “双重分派”技术

Fig. 4 “Dual distribution” technology

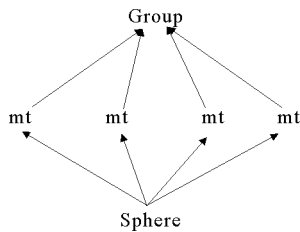


图5 粒子系统

Fig. 5 Particle system

整个粒子系统作为1个根节点 Group,下面是各个粒子对象 Sphere。定义根节点 Group 的回调函数 setUpdateCallback (osg::NodeCallback),在函数中对每个粒子的位置变量 mt(Matrix Transform 类型)进行修改。这样,粒子系统创建后,根节点通过执行节点回调函数,对每个粒子位置进行修改和绘制,从而实现粒子运动。

2.5 多线程消息传递机制

根据 MFC 与 OSG 框架集成原理,MFC 在子线程中创建 OSG 对象,具体的线程创建和访问机制如图6所示。

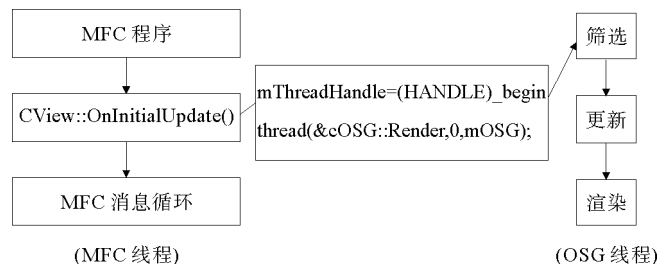


图6 线程创建

Fig. 6 Thread creation

从图6可以看出,MFC 框架在 CView 类里启动了 OSG 的渲染线程,所以在整个程序里存在了两个线程:MFC 交互框架线程和 OSG 线程,2 个线程都具有独立

的消息处理机制来管理和操作对象。如果在 MFC 框架下直接采用传统的类对象调用方式对 OSG 线程中的共享节点进行控制,则会出现访问冲突,导致系统异常退出。因此,在 MFC 与 OSG 集成技术下,需要利用多线程消息传递机制,通过 MFC 框架主进程向 OSG 子进程发送图形节点访问消息,由 OSG 线程内部的消息处理机制完成该消息相应的操作。目前主要的实现方法有以下3种:

- (1)在 MFC 里自定义事件,压入 OSG 的事件队列,在 OSG 的渲染线程里进行处理;
- (2)使用 command buffer 的策略,MFC 层传递 1 个指令到 OSG,然后在 OSG 线程里解析并执行它;
- (3)在 MFC 下设置变量来触发 OSG 执行回调并完成操作。

其中(1)和(2)方式都是通过 OSG 事件处理机制来完成,具体通过重载虚函数 handle (const osgGA::GUIEventAdapter&ea, osgGA::GUIActionAdapter&aa) 来实现。该函数的第 1 个参数代表 MFC 框架消息的提供者,如鼠标事件(PUSH、RELEASE、DRAG、MOVE)、键盘事件(KEYDOWN、KEYUP)等,第 2 个参数是用来对 GUI 进行反馈,它可以让 GUIEventHandler 根据输入的事件让 GUI 进行一些动作。handler 的返回值决定这个事件是否继续让后面的 GUIEventHandler 处理,如果返回 true,则系统会认为该事件已经处理,就不再传递给下一个事件处理器;如果返回 false,后面的 GUIEventHandler 还有机会继续对这个事件进行响应,具体实现见图7。

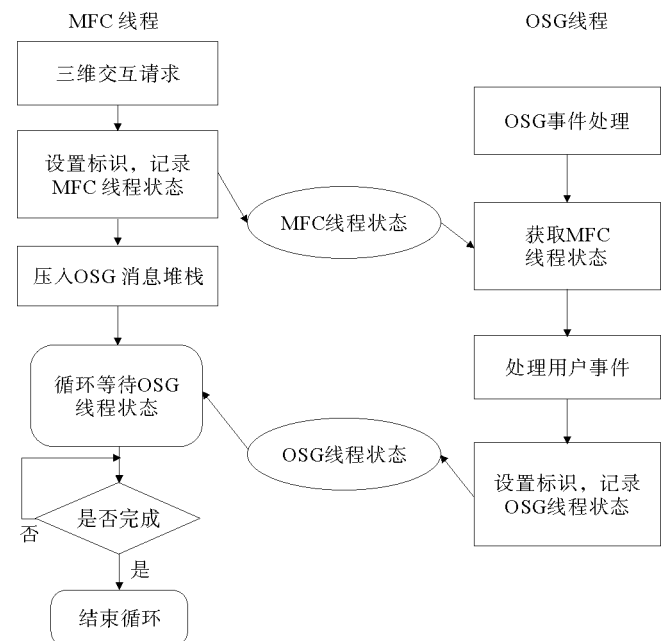


图7 线程交互机制

Fig. 7 Thread interaction mechanisms

2.6 交互控制

把现实三维对象在计算机屏幕上显示需要完成目标场景视图的变换过程:视点变换、模型变换、投影变换、视口变换,这些变换都是通过矩阵操作完成的。首先,通过模型变换对模型的位置、大小、角度进行调整;然后,通过视点变换设置视点的方向和位置;在投影变换下设置裁剪视景体对场景进行裁剪,将在视景体内的场景投影到二维平面内;最后,将投影结果在视口变换下放到合适的窗口下进行显示(见图8)。

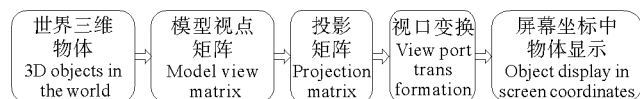


图8 三维图形显示过程

Fig. 8 Three-dimensional graphics display process

为了提供友好的用户交互功能,利用矩阵变换原理和多线程消息传递机制,实现三维场景的人机交互操作。通过对模型矩阵变换的方式,实现工具栏及键盘对图形节点的访问和交互操作;通过视点矩阵变换的方式对相机操作器 TrackballManipulator 的成员变量 `_center`、`_rotation`、`_distance` 等进行设置,改变观察矩阵,灵活实现对场景的移动、缩放、旋转操作;对投影矩阵操作实现裁剪视景体的调整,完成场景内对象有选择的显示;操作视口矩阵完成场景图形在计算机中显示位置、大小的调整。利用上文提到的 MFC、OSG 多线程关键技术,将鼠标、键盘、工具栏消息通过 OSG 在高层为用户提供的统一接收用户各种操作的 `osgGA` 图形抽象层接口,由 OSG 负责与不同底层窗口通信,完成交互控制。

3 海洋三维场景构建关键技术

3.1 海底地形模型的构建

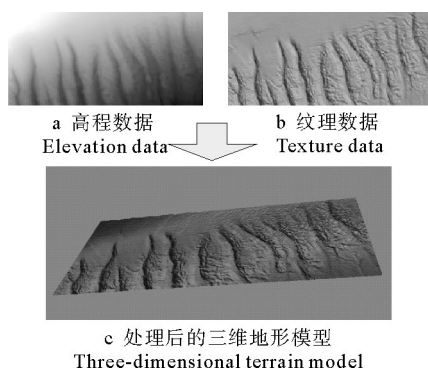


图9 海底地形模型构建

Fig. 9 The construction of the undersea terrain model

海底地形是海洋三维场景构建的基础地理背景。为了提高海底地形的显示效果和渲染效率,本文利用

OSG 提供的 VirtualPlanetBuilder (简称 VPB) 工具对海底地形数据进行建模。该工具可以读入多种类型的地理影像和高程数据,并利用层次细节法 (LOD) 构建各种规模的分块分页地形数据模型。对于单个高程数据或影像数据,文件不宜过大,本研究对文件的大小限制为 4GB,如果超出,则需要重新分割。VPB 采用四叉树结构^[9]进行 LOD 级别划分:0 级是最粗糙的级别,展示地形的整体概貌,1 级将整体划分成 4 部分,每一部分又可以分成下一级别的四部分,以此类推。OSG 在进行三维场景显示时,自动根据场景对象与观察者的距离调用相应精度的数据模型进行显示,模型越靠近观察者,显示级别越高,数据越精细,显示范围会相对缩小,以减轻系统绘制场景的负担,提高显示效率和速度。本文采用如下海底峡谷高分辨率地形高程数据(见图 9a)和纹理数据(见图 9b),在对高程数据和影像数据格式化处理之后,采用 VPB 工具命令 (`osgdem`) 生成海洋三维场景地形模型(见图 9c)。

3.2 海水仿真

海水的模拟技术应用十分广泛,不仅在计算机图形学,军事领域,虚拟现实,电影制作、动画及特效,海洋工程领域等具有非常重要的意义,在动力海洋学,流体力学,波动力学,气象预报等也具有非常重要的意义。本系统采用 GLSL 语言,利用 GPU 加速绘制技术对海水进行可视化仿真。首先建立海面网格模型,然后在海面网格的高度场上叠加纹理贴图(见图 10a),通过 Shader 中的 Vertex Shader 对象实时更改每个网格顶点位置、法线、切线,然后传到 Pixel Shader 中进行插值,得到海面每一个像素的法向量,进而计算出水的折射、反射颜色,并进行调制生成水的最终颜色。本文将立方图纹理(见图 10b)作为环境贴图,实现海水对天空环境的反射和对海底的折射,通过在法线上的扰动实现海水波动的效果,绘制出逼真的海面环境(见图 10c)。

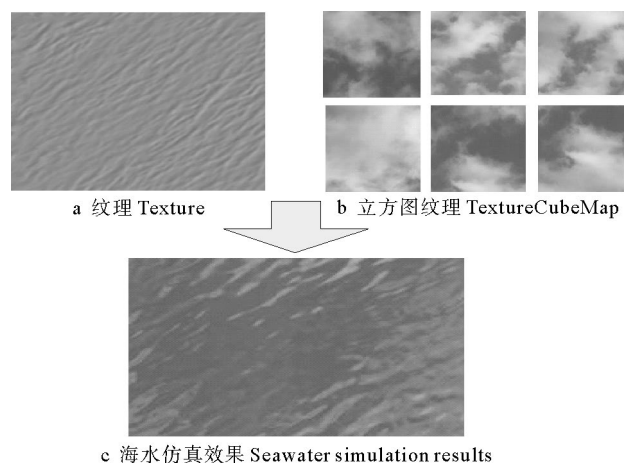


图10 海水仿真

Fig. 10 Seawater simulation

3.3 海洋局部三维场景构建

目前对海面的仿真大都是对整个海面场景的展示,如 osgOcean,一方面在绘制效率上对显卡性能要求较高,另一方面不便于对局部区域进行观察。本系统利用碰撞检测技术研究对海洋局部三维场景的构建技术,根据用户指定范围,动态实现任意地理位置的局部海洋场景构建。

碰撞检测技术主要采用射线体检测算法,用 1 条射线与场景中的物体包围盒进行相交测试,如果有交点则认为射线与物体发生碰撞。OSG 中进行碰撞检测的核心函数接口基本都由 osgUtil 提供支持,1 条线段和场景中的几何体相交的检测由几个关键的类定义:

(1)osg::LineSegment:表示 1 个线段的类,包括 1 个起点和 1 个终点;

(2)osgUtil::IntersectVisitor:接受线段的类,用于判别与节点的交集,函数 addLineSegment(line.get())用来添加 1 条线段到列表当中;

(3)osgUtil::IntersectVisitor::HitList:1 根线段很有可能与场景中的很多物体或者与同一个物体有多次的相交,那么就返回 1 个相交列表,如果在这条线段上没有发生相交,就保持空。

本文在运用相交检测技术构建局部海洋场景时,首先用一系列等间隔的直线与海底地形模型相交,得到交点集。然后,从这些交点往上构建海洋立体观察盒,四个侧面的底边以这些交点为边界,顶面利用海水仿真技术进行海面绘制。利用裁剪技术,把整个海洋场景限制在剪裁盒范围内,只有在剪裁盒范围内的才显示。

在利用相交点构建侧面的时候,可能会出现凹多边形情况(即两点的连线不在该多边形内)。为了避免出现这种情况,需要按照交点连线顺序进行分格化处理,通过指定相应的绕线规则把凹多边形分解为简单的凸多边形^[10-13]。

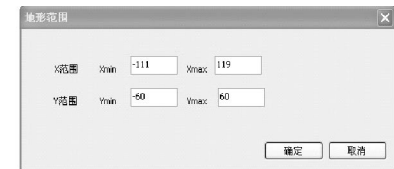
3.4 系统场景组织与检索

在 OSG 场景图高效组织方式下,将海底地形、裁剪盒、海面、粒子系统节点按照树状结构进行管理,利用访问机制可以快速的对场景中的物体以自顶向下或是自顶向上的顺序按照某一属性(节点 ID 或是名字标识等)进行搜索查询,对找到符合条件的节点可以进行目标操作,如得到节点的详细信息、修改属性、删除、添加船只等操作。通过相交检测,依据交点信息对场景内的对象(如船只、粒子等)进行拾取,并得到物体的三维坐标、属性等信息。

4 系统实现

本文基于 MFC 框架下的 OSG 三维交互和海洋三

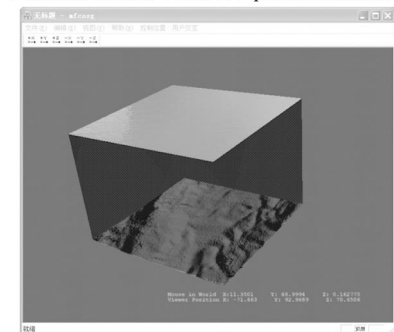
维场景构建关键技术的研究,采用 Visual Studio 2008 开发环境和 OSG3.0 可视化库,对海洋环境三维可视化系统进行了开发和实现。该系统可以根据用户自定义的场景范围和观察位置对显示范围参数(见图 11a)和观察参数(见图 11b)进行设置,根据参数动态构建海洋局部三维观察场景(见图 11c)。用户可以利用交互工具栏、鼠标和键盘灵活地对三维海洋场景进行操作,并且实时对显示参数、环境变量等信息进行显示,方便对海洋环境进行观察、仿真和模拟。



a 显示参数设置 Displayed parameter setting



b 观察参数设置 Observed parameter setting



c 三维局部海洋环境场景

Three-dimensional scenes of the local marine environment

图 11 参数设置及系统场景

Fig. 11 Parameter setting and system scene

5 结语

本文基于 MFC 框架,利用 OSG 场景图形管理和绘制功能,结合 GPU 加速绘制技术,通过友好的人机交互界面,将海面环境和海底地形有机地结合在一起,为用户提供了 1 个实时、动态、可交互的三维可视化海洋仿真系统。基于该系统,可以进一步开展海底溢油仿真、海底灾害模拟、海洋环境要素可视化等应用,具有比较好的应用前景。目前,系统还存在一些不足之处有待进一步完善,如结合海流、海温、盐度、密度等信息对海洋水体可视化表达;同过雾效、光照、刻蚀等技术增强海洋环境的逼真度;利用碰撞检测技术克服场景交互中存在的视点与物体碰撞情况,为用户提高沉浸感。

参考文献:

- [1] Lindstrom P, Koller D, Ribarsky W, et al. RealTime Continuous Level of Detail Rendering of Height Fields[M]. New Orleans: SIGGRAPH M' 96, 1996.
- [2] 杜剑侠, 李凤霞, 战守义. LOD 算法研究及其在地形实时显示中的应用[J]. 计算机工程与应用, 2005, 41(3): 211-213.
- [3] Lee D T, Schachter B J. Two algorithms for constructing a Delaunay Triangulation[J]. Journal of Computer and Information Science, 1995, 9(3): 219-242.
- [4] 孟浩, 刘惠媛. 三维海底地形动态绘制方法研究与实现[C]. 哈尔滨: 哈尔滨工程大学自动化学院年会, 2008.
- [5] Foster N, Metaxas D. Realistic Animation of Liquids[C]. // Graphical Models and Image Processing. New York: Academic Press, 1996, 58(5): 471-483.
- [6] Jerry Tessendorf. Simulating Ocean Water[C]. // Proceeding of ACM SIGGRAPH, New York: ACM Press, 1999: 348-367.
- [7] 肖鹏, 刘更代, 徐明亮. Open Scene Graph 三维渲染引擎编程指南[M]. 北京: 清华大学出版社, 2010.
- [8] 侯俊杰. 深入浅出 MFC [M]. 第 2 版. 武汉: 华中科技大学出版社, 2001.
- [9] 柯希林, 曾军. 动态 LOD 四叉树虚拟地形绘制[J]. 测绘通报, 2005(6): 12-15, 34.
- [10] Schachter B. Decomposition of polygons into convex sets[J]. IEEE Trans on Computers, 1978, C-27(11): 1078-1082.
- [11] Keil M. Decomposing Polygons into Simpler Components[M]. Toronto: University of Toronto, 1983.
- [12] 肖忠晖, 卢振荣, 张谦. 简单多边形凸单元剖分的编码算法[J]. 计算机学报, 1996, 19(6): 477-481.
- [13] 金文华, 饶上荣, 唐卫清, 等. 基于顶点可见性的凹多边形快速凸分解算法[J]. 计算机研究与发展, 1999, 36(12): 1455-1460.

Research of the Key Technology of Three-Dimensional Marine Environment

SONG Meng¹, SU Tian-Yun², WANG Guo-Yu¹, LI Jia-Gang³, LI Xin-Fang²

(1. Ocean University of China, Qingdao 266100, China; 2. The First Institute of Oceanography, SOA, Qingdao 266061, China; 3. Research Center of China National Offshore Oil Corporation, Beijing 100027, China)

Abstract: In order to satisfy the application requirements of marine scientific research and engineering development for intuitive simulation and expression of marine environment, based on MFC interaction framework and OSG 3D rendering engine, we researched key three-dimensional interactive technologies such as node accessing, callback mechanism, event handling, messaging and so on. Then, three dimensional marine environment was constructed and rendered from terrain model building, seawater simulation, coordinate mapping and scene cutting aspects, which can realize friendly interface on three-dimensional interactive scene of marine environment and facilitate further simulation and modeling of marine features and phenomena.

Key words: Open Scene Graph(OSG); MFC; GPU; marine environment simulation; 3D visualization

责任编辑 陈呈超