

文章编号:1002-3682(2015)01-0020-09

基于 OSG 的海底溢油多维 动态可视化技术研究*

李小龙¹, 曹金凤²

(1. 中海石油深海开发有限公司, 广东 深圳 518067;

2. 青岛理工大学 理工学院, 山东 青岛 266520)

摘 要:基于 MFC(Microsoft Foundation Classes)交互框架和 OSG(Open Scene Graph)三维渲染引擎,研究 MFC 与 OSG 集成和多线程交互关键技术,从 LOD 地形模型构建、海水仿真以及场景剪裁等几个方面对海洋环境三维立体场景进行构建和渲染。采用 GPU 高级着色语言(GLSL),结合 FBO(Frame Buffer Object)帧缓冲技术,对大规模溢油粒子系统的实时仿真绘制技术进行研究和实现,实验结果表明,该方法的运行效率比常规的粒子系统模拟方法(如 OpenGL 的 glBegin-glEnd 方法)有明显的提高。基于关键技术研究成果,开发实现海底溢油可视化系统,可通过便捷、友好、直观的可视化界面,对海洋环境立体场景和大规模溢油粒子进行高效、实时地模拟和再现。

关键词:三维海洋环境;溢油仿真;OSG;GPU;FBO

中图分类号:P229

文献标识码:A

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两个方面。在海底地形建模方面,Lindstrom 等^[1]和杜剑侠等^[2]基于规则网格数据和不规则网格数据的层次细节模型对地形高程数据进行可视化建模;贾俊涛等^[3]和陆丹等^[4]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。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 等^[5]通过求解流体 Navier-Stokes 方程真实地模拟了海面。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

* 收稿日期:2014-12-01

资助项目:工业和信息化部海洋工程装备科研项目——500 米水深油田生产装备 TLP 自主研发;自然科学基金委山东省海洋科学研究中心联合资助项目——海洋环境动力学和数值模拟(U1406404);中央级公益性科研院所基本科研业务费专项资金资助项目——波流耦合数值模式预报系统海量数据的远程动态可视化研究(2013G43)

作者简介:李小龙(1964-),男,高级工程师,主要从事海洋石油开发工程设计方面研究。E-mail:lixl10@cnooc.com.cn

(李 燕 编辑)

海洋环境的整体效果,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 虽然可以大范围地展现海底地形,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(Open Scene Graph)可视化库,将 MFC(Microsoft Foundation Classes)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,对海洋环境三维立体场景构建关键技术和大规模海底溢油粒子快速仿真技术进行研究。将用户交互框架、海底地形、海水与溢油粒子等要素进行集成,提供一个实时、动态、可交互的海底溢油三维可视化系统,清晰、直观地再现海底溢油场景,为从事海洋开发、资源勘测的工作人员提供一个专业的模拟仿真工具。

2 三维交互关键技术

2.1 MFC 与 OSG 集成

OSG 是一个基于 OpenGL 扩展封装的开源可视化图形库(www.osgChina.org),使用 C++ 语言编写而成^[6]。OSG 在封装的基础之上,建立一个面向对象的框架,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。

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,并且包含一个基于文档/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^[7]。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 集成的流程如图 1 所示。

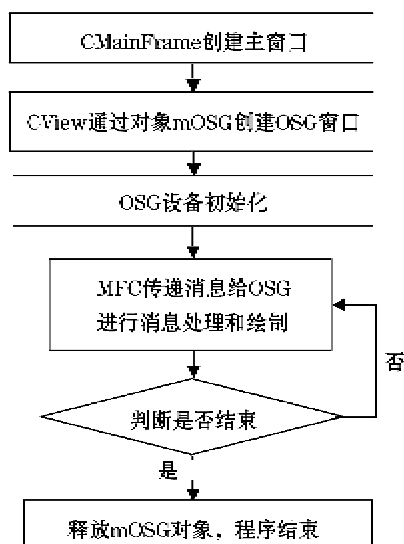


图 1 MFC 与 OSG 集成流程

Fig. 1 Flow chart of the integration of MFC and OSG

2.2 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, 具体的线程创建如图 2 所示。

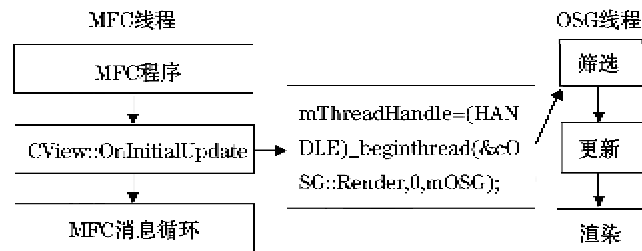


图 2 线程创建

Fig. 2 Thread creation

从图 2 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 2 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。线程之间的交互机制如图 3 所示。

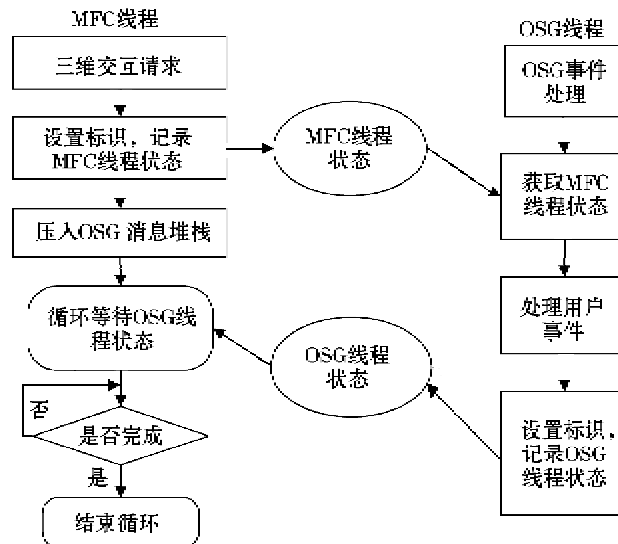


图 3 线程交互机制

Fig. 3 Thread interaction mechanism

3 海洋三维场景构建关键技术

3.1 海底地形模型的构建

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,我们利用 OSG 提供的 Virtual Planet Builder(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。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,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。本文采用高分辨率地形高程数据(图 4a)和纹理数据(图 4b),在对高程数据和影像数据格式化处理后,采用 VPB 工具命令(osgdem)生成海洋三维场景地形模型(图 4c)。

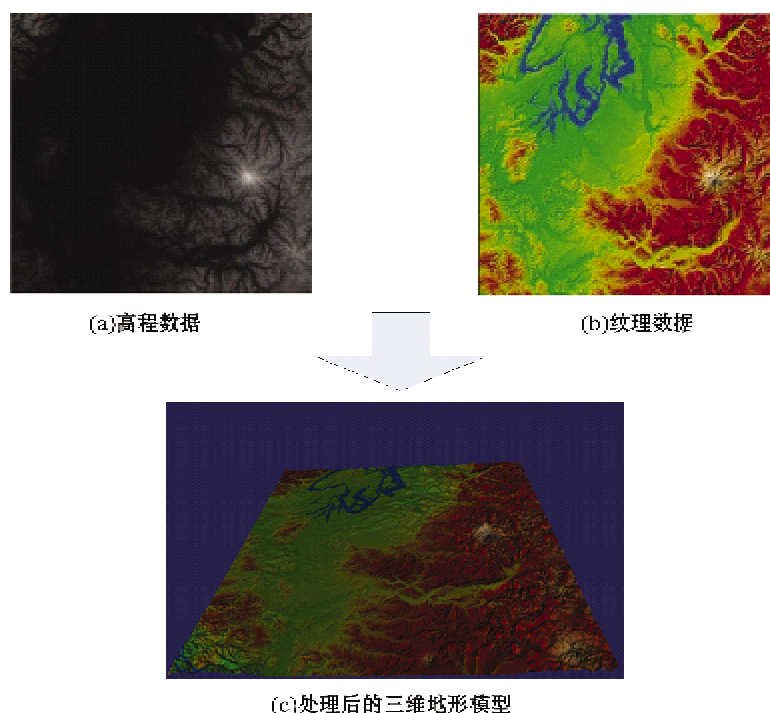


图 4 海底地形模型构建

Fig. 4 Modeling of the seabed terrain

3.2 海水仿真

海水的模拟技术应用十分广泛,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、气象预报等方面也具有重要意义。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(图 5a),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(图 5b)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(图 5c)。

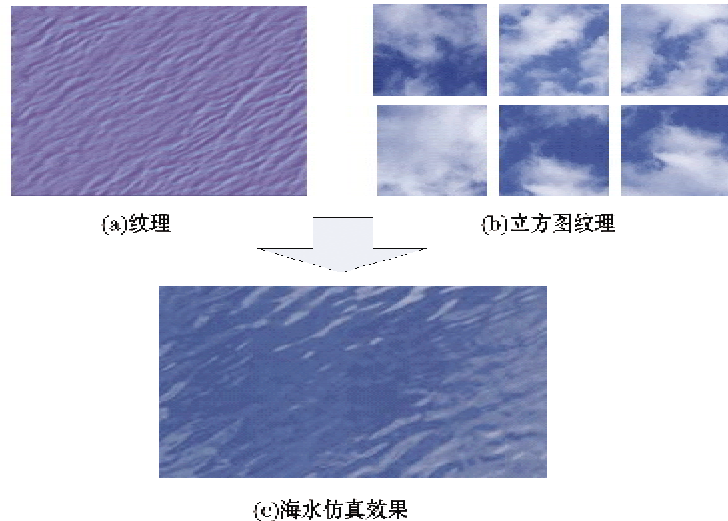


图5 海水仿真

Fig. 5 Seawater simulation

3.3 局部海洋三维立体场景构建

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。本系统根据用户指定范围,利用碰撞检测技术研究对海洋局部三维立体场景的构建技术,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,用一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,一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`: 表示一个线段的类,包括一个起点和一个终点;
- 2) `osgUtil::IntersectVisitor`: 接受线段的类,用于判别与节点的交集,其中的函数 `addLineSegment(line.get())` 用来添加一条线段到列表当中;
- 3) `osgUtil::IntersectVisitor::HitList`: 一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,那么就返回一个相交列表,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立体场景时,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,从这些交点向上构建海洋立体观察盒,4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的情况(即 2 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^[8-9]。

4 溢油模拟

伴随着海洋油气资源的勘探开发和海洋运输的迅速发展,操作性溢油和事故性溢油日益频繁,造成了巨大的环境破坏和经济损失,石油污染成为各个领域关注的焦点。由于溢油粒子运动规律非常复杂,具有多维动态性,目前溢油可视化工作更多的是基于海图等二维平台对溢油的数值模拟结果进行表达。将实际的多维动态现象采用二维的方式表示,具有很大的局限性,而利用三维可视化技术,结合三维海洋场景,可直观、逼真地再现溢油场景。由于溢油粒子包含多种类型,大小不同,特别是数量非常多,对每个溢油粒子进行定向、定位的实时再现,面临大数据量的粒子系统实时加载难题。

本文采用 GPU 编程的 GLSL 高级着色语言并结合 Frame Buffer Object (FBO) 帧缓冲技术对粒子系统进行实现,提高大规模溢油粒子系统的高效性和实时性。所谓帧缓冲区,其实是显卡内存中的一块,它的特别之处在于保存在该内存区块中的图像数据,会实时地在显示器上显示出来,实现流程如图 6 所示。

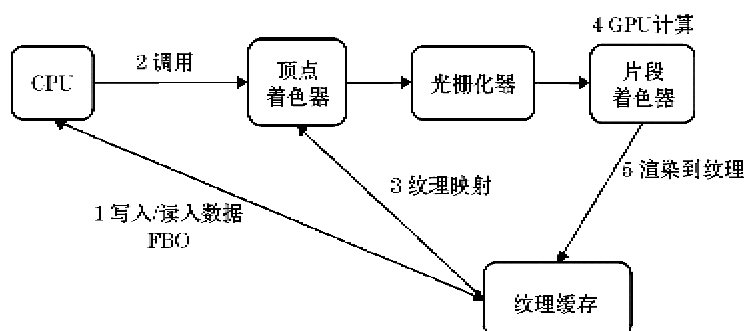


图 6 基于 FBO 的粒子系统绘制流程

Fig. 6 Flow chart of drawing particle system based on FBO

在这个过程中,每个时刻的油粒子位置数据保存成纹理,该纹理通过与颜色绑定绑定到 FBO 上,把纹理数据(溢油位置)传送到颜色缓冲区,在 GPU 中对颜色缓冲区中的数据通过 GLSL 的片段着色器进行修改,将修改后的数据从 FBO 获取传递到 CPU,更新每帧的油粒子位置。这样通过显卡内存 FBO 作为中间数据的传递,利用 GPU 进行并行计算,极大的提高了粒子仿真和渲染速度。

在 Intel Xeon E5 CPU、8GB 内存、1GB 显存、64 位 Windows 7 操作系统的计算机上,我们对基于 FBO 的粒子系统仿真绘制方法与常规的粒子系统仿真绘制方法(如 OpenGL 的 glBegin-g glEnd 方法)进行对比测试,测试结果如表 1 所示。根据测试结果可以看出,该方法能够实现近千万溢油粒子的快速计算和实时绘制,运行过程流畅,在实时性和粒子规模上比常规的仿真绘制方法有了很大改善。

表1 粒子系统绘制效率测试对比

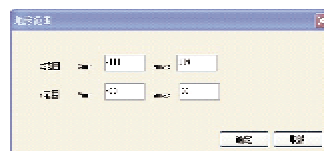
Table 1 The comparison of the particle system drawing efficiency test

粒子数量 / $\times 10^5$ 个	常规绘制方式 的帧速/帧 $\cdot s^{-1}$	FBO 绘制方式 的帧速/帧 $\cdot s^{-1}$
2	213	584
4	145	492
6	74	267
8	56	194
10	50	168
20	23	118
40	12	64
60	7	33
80	6	32
100	5	25

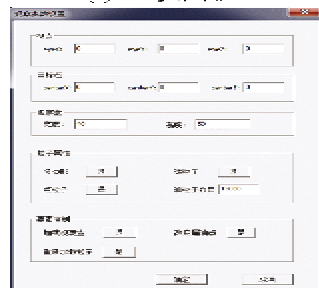
5 系统实现

基于 MFC 框架下的 OSG 三维交互、海洋三维立体场景构建和溢油模拟关键技术的研究,采用 Visual Studio 2008 开发环境和 OSG3.0 可视化库,对海底溢油可视化系统进行了开发和实现,并且在系统中模拟和再现海底三维溢油场景。

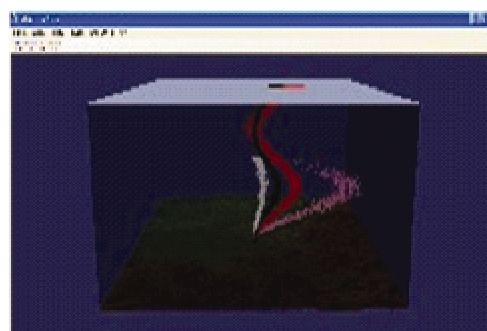
该系统可以根据用户自定义的场景范围和观察位置对显示范围参数(图 7a)和观察参数(图 7b)进行设置,根据参数动态构建海洋局部三维观察场景,并且在该场景下实现海底溢油的动态仿真(图 7c)。用户可以利用交互工具栏、鼠标和键盘灵活地对三维海洋场景进行交互操作,并且实时对显示参数、环境变量等信息进行显示,方便对海洋环境进行观察、仿真和模拟。



(a)显示参数设置



(b)观察参数设置



(c)海底溢油场景

图7 参数设置及系统场景

Fig. 7 Setting of parameters and scene of system

6 结 论

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,对海洋三维立体场景构建技术和大规模海底溢油粒子系统实时仿真绘制技术进行了研究和实现。通过友好的人机交互界面,将海面环境、海底地形和溢油粒子有机地结合在一起,为用户提供了一个实时、动态、可交互的三维可视化海底溢油仿真系统。实验结果表明,该系统能够对海洋立体环境和大规模溢油粒子进行实时、高效地模拟和绘制。

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;通过雾效、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,提高用户的沉浸感体验度。

参考文献:

- [1] LINDSTROM P, KOLLER D, RIBARSKY W, et al. Real-time continuous level of detail rendering of height fields[C]// Proceedings of the 23rd annual conference on computer graphics and interactive techniques. New Orleans USA: ACM Press, 1996:109-118.
- [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(13):211-213.
- [3] 贾俊涛,翟京生,孟婵媛,等. 基于海量多波束数据的海底地形模型的构建与可视化[J]. 测绘科学技术学报,2008,25(4):255-259.
- [4] 陆丹,李海森,魏玉阔,等. 多波束测深系统中的海底地形可视化技术研究[J]. 仪器仪表学报, 2012,33(2):450-456.
- [5] FOSTER N, METAXAS D. Realistic animation of liquids[J]. Graphical Models and Image Processing,1996,58(5):471-483.
- [6] 肖鹏,刘更代,徐明亮. OpenSceneGraph 三维渲染引擎编程指南[M]. 北京:清华大学出版社, 2010.
- [7] 侯俊杰. 深入浅出 MFC[M]. 武汉:华中科技大学出版社,2001.
- [8] SCHACHTER B. Decomposition of polygons into convex sets[J]. IEEE Transactions on Computers, 1978, C-27(11): 1078-1082.
- [9] 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.

Research on the Technology of Multi-dimensional Dynamic Visualization of Sub-marine Oil Spill Based on OSG

LI Xiao-long¹, CAO Jin-feng²

(1. CNOOC Deepwater Exploitation Co. Ltd., Shenzhen 518067, China;

2. School of Science, Qingdao Technological University, Qingdao 266520, China)

Abstract: Based on Microsoft Foundation Classes (MFC) user interface class and Open-SceneGraph (OSG) 3D rendering engine, key techniques of integration and multi-thread interactivity between MFC and OSG are researched, and a three-dimensional scene of marine environment is modeled and rendered with techniques such as LOD terrain model, seawater simulation and scene clip. Technology of real-time and dynamic simulation of large-scale submarine oil spill particle system is investigated and implemented by means of GPU high lever shader language GLSL and frame buffer object (FBO). The results show that this method is much more efficient than those (e. g. the OpenGL "glBegin-glEnd" mode) commonly used for the simulation of particle system. Based on the key technology researches a visualization system is developed for the submarine oil spill, which provides a convenient, friendly and intuitive visual interface for the efficient and real-time simulation and reappearance of the three-dimensional marine environment and the large-scale submarine oil spills.

Key words: three-dimensional marine environment; oil spill simulation; OSG; GPU; FBO