

基于分形的多分辨率三维海浪实时仿真

彭 耿^{1,2}, 张立民¹, 艾祖亮¹, 邓向阳¹

(1.海军航空工程学院 电子信息工程系, 山东 烟台 264001;

2.国防科技大学 电子科学与工程学院, 长沙 410073)

E-mail: hjhy-penggeng@163.com

摘 要: 海浪的实时仿真与传统的计算机真实感图形的主要区别是, 它将绘制的实时性和真实感同时作为衡量系统有效性的重要指标。论文跟踪了国内外海浪仿真前沿, 结合具体的工程需求和软硬件环境, 通过建立数学模型, 基于海浪的分形特性, 提出并实现了满足模拟飞行训练需要的多分辨率三维海浪实时仿真。在编程中, 对实时性进行了研究并提出了相应的对策, 较好地满足了实时性要求。文中提出的海浪实时仿真方法和关键技术, 既可为模拟飞行训练提供一个虚拟海战场环境, 又可为其它自然景物仿真提供经验和方法。

关键词: 计算机应用; 海浪仿真; 分形; 实时

文章编号: 1002-8331(2006)33-0011-04 文献标识码: A 中图分类号: TP391.9

Real-time Simulation of Three-dimensional Multi-resolution Ocean Wave Based on Fractal

PENG Geng^{1,2}, ZHANG Li-min¹, AI Zu-liang¹, DENG Xiang-yang¹

(1.Dept. of Electronics & Inf. Eng., Navy Aviation Engineering Institute, Yantai, Shandong 264001, China;

2.School of Electronic Sci. & Eng., National University of Defense Technology, Changsha 410073, China)

Abstract: Distinguished with traditional computer realistic graphics, qualities of real time and fidelity are regarded as significant indexes of weighing system validity for real-time simulation of ocean wave simultaneously. It is shown that fractal is regarded as one character of ocean wave by studying plenty of papers. Tracking advancing front of ocean wave simulation home and abroad, associating with concrete engineering requirements and conditions of software and hardware, an algorithm of ocean wave simulation based on fractal is introduced, which satisfies the needs of flight training simulation. In the course of programming, the character of real time is researched and corresponding simplified means are presented. The method and key technologies of ocean wave real-time simulation advanced in this article can not only offer a virtual ocean battlefield environment for flight training simulation, but also present experiences and ways for other natural scenery simulation.

Key words: computer application; ocean wave simulation; fractal, real time

1 引言

海浪本身是一个十分复杂的自然现象, 它无论在时间上还是空间上, 都具有不规则性和无重复性。近些年来, 随着计算机软硬件技术的飞速发展, 人们尝试利用虚拟现实技术来生成真实感很强的海浪动画或兼顾真实感和实时性的虚拟海战环境, 这些研究对于海战、海区作业、海上游乐等都具有非常重要的现实意义。

国内外关于海浪的仿真研究归纳起来主要集中在三个方面: 基于物理模型、基于构造波形函数和基于数值模拟。基于物理模型的海浪仿真, 就是对海浪的波动从计算流体力学的角度加以研究, 从著名的流体动力学方程组—Navier-Stokes 方程组出发, 通过一系列近似和简化, 用数值方法求解流场来实现。基于构造波形函数的海浪仿真, 就是根据波浪的形状, 直接构造参数曲面来表示水表面形函数。基于数值模拟的海浪仿真, 就

是从海洋学的观测和研究成果出发, 利用海浪谱的相关公式, 在计算机上, 根据所需的海浪参数, 实现基于海浪谱的海浪数值模拟。

通过对国内外海浪仿真前沿方法的研究, 结合课题的软硬件环境和要求, 本文提出了一种满足某型飞机训练模拟器系统需要的海浪实时仿真算法, 则仅需考虑深水波, 即水深与波长之比大于 1/2, 此时海浪不受海底的影响, 仿真中就不用考虑海底这个因素, 大大降低了仿真的复杂性^[1]。

2 海浪的数学模型

2.1 仿真数据获取流程

海浪研究理论表明, 海浪波面可由无限多振幅不等、频率不等、初相位不等及传播方向不等的余弦波叠加而成, 最终得到的波面高度可表示如下^[2]:

基金项目: 国家部委基金项目; 中国博士后科学基金(2005038469)。

作者简介: 彭耿(1980-), 男, 博士生, 研究方向为计算机图形学、视景仿真和信息融合; 张立民(1966-), 男, 教授, 现为天津大学博士后, 研究方向为电子线路 CAD、分布式交互仿真和计算机图形学。

$$y=\eta(x,z,t)=H_{Tide}+\sum_{n=1}^{\infty}a_n\cos[w_n t+k_n(x\cos\theta_n+z\sin\theta_n)+\varepsilon_n] \quad (1)$$

在式(1)中, $a_n, w_n, k_n, \theta_n, \varepsilon_n$ 分别表示第 n 次谐波的振幅、频率、波数、在 XOZ 平面上的传播方向与 X 轴成的角度、初相。初相 ε_n 是在 $0-2\pi$ 上均匀分布的随机变量, 概率分布为 $p(\varepsilon)=1/2\pi$ 。波数与相应的波浪角频率间的关系为 $w_n^2=k_n g, H_{Tide}$ 表示潮高, 在这里可暂时赋值为 $H_{Tide}=0$ 。

对于式(1)的求解, 流程如图 1 所示。

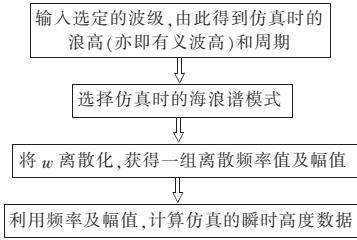


图 1 海浪仿真的高度数据获取流程图

2.2 仿真中的浪级、风和波高、周期的关系

引起海浪的原因很多, 最常见的是作用在海面上的风, 海浪的要素取决于风速、风时和风区的大小。对海浪等级的定义目前尚未统一^[3], Douglas 级数应用广泛, 但后来被国际气象组织(WMO)代码 75 代替, 第三种方式是蒲福风力等级, 它是用风速来描述的, 但并未体现浪级与周期间的关系。根据俄罗斯涌浪等级与波高、周期间的关系及前面的三种方式, 可以确定海浪实时仿真中的浪级、风、波高和周期间的关系, 如表 1 所示。

表 1 海浪实时仿真中的浪级、风和波高、周期的关系

浪级	相应的风级	浪高划分	仿真浪高/m	仿真周期/s
0	0	0.00	0.00	0.0
1	1	(0.00, 0.10)	0.10	1.0
2	2-3	(0.10, 0.50)	0.50	1.8
3	4	(0.50, 1.25)	1.25	3.5
4	5	(1.25, 2.50)	2.50	4.5
5	6-7	(2.50, 4.00)	4.00	6.5
6	8	(4.00, 6.00)	6.00	8.2
7	9-10	(6.00, 9.00)	9.00	9.7
8	10-11	(9.00, 14.00)	14.00	11.0
9	12	>14.00	16.00	13.0

2.3 海浪谱和仿真精度

海浪谱是一种描述海浪能量随频率和/或方向分布的物理量, 它在海浪的理论研究和实际应用中均不失为一种有效手段。海浪谱比较容易观测分析, 现在国内外已提出大量的海浪谱, 如皮尔逊—莫斯科维奇谱、ITTC 单参数波谱、ITTC 双参数波谱、JONSWAP 谱、中国沿海海浪谱等, 本文在模拟海浪时采用的是 ITTC 单参数波谱, 即:

$$S(w)=\frac{A}{w^5}\exp\left[-\frac{B}{w^4}\right] \quad (2)$$

各种海浪谱都是窄带谱, 它们的能量主要集中在某一频段 $[w_L, w_H]$, 特别是在高海情的情况下, 能量更是集中在某一频段。为了实现实时仿真, 提高仿真速度, 必须选择有限的频段来进行仿真, 即 $n \in [1, M]$, 忽略高次谐波, 仿真结果仍可以保持较高

的精度。

频段的确定取决于所要求的仿真精度。设在低频 w_L 与高频 w_H 外分别允许略去总能量的 μ 部分, 则有:

$$\mu=\frac{\int_0^{w_L} S(w)dw}{\int_0^{\infty} S(w)dw}=\frac{\int_{w_H}^{\infty} S(w)dw}{\int_0^{\infty} S(w)dw} \quad (3)$$

对于给定的仿真精度 $1-2\mu$, 代入式(3)即可获得仿真频段 $[w_L, w_H]$ 。

2.4 频率离散化和仿真参数的确定

选取海浪频谱 $S(w)$ 和计算出频段 w_L-w_H 后, 为便于计算机处理, 就必须对频率进行离散化, 即对频率区间 $[w_L, w_H]$ 进行划分, 在区间中选取代表频率进行仿真计算。频率的离散化有等分频率法和等分能量法两种。

等分频率法就是将频率范围 $[w_L, w_H]$ 内的角频率做 M 等间隔划分。这样的划分有两个缺点: 第一是得到的波面将出现周期性的重复, 与实际情况不符; 第二是海浪谱的能量集中于谱峰频率附近, 采用等频率划分必须使用较多谐波来叠加, 否则在谱峰附近采样点由于太少会导致仿真误差偏大。显然, 等频率划分不适合于海浪的实时仿真。

能量等分法是基于海浪的频谱, 设频率段范围为 $[w_L, w_H]$, 选定频率 $w_0(w_L), w_1, \dots, w_M(w_H)$, 使各频率间隔的能量相等, 即频谱曲线下的子面积相等。具体方法如下:

现在定义能量累积谱, 即:

$$E(w_i)=\int_0^{w_i} W(w)dw \quad (4)$$

由此则有:

$$\int_{w_{i-1}}^{w_i} S(w)dw=E(w_i)-E(w_{i-1}) \quad (5)$$

根据能量等分的思想, 令 $E(w_i)-E(w_{i-1})=X$ (常数), M 是能量等分的份数, 于是 $M \cdot X=E(\infty)$, 从而有:

$$X=\frac{1}{M}E(\infty) \quad (6)$$

$$E(\infty)=\int_0^{\infty} \frac{A}{w^5}\exp\left[-\frac{B}{w^4}\right]dw=\frac{A}{4B} \quad (7)$$

综上所述:

$$E(w_i)=\frac{iA}{4MB}, i \in [1, M-1] \quad (8)$$

$$w_i=\left[-\frac{B}{\ln(i/M)}\right]^{\frac{1}{4}}=[B/\ln(M/i)]^{\frac{1}{4}} \quad (9)$$

此时, 若 $w_1 < w_L$, 则取 $w_1=w_L$; 若 $w_M > w_H$, 则取 $w_M=w_H$ 。

$$X=\frac{A}{4BM} \quad (10)$$

波幅由频谱的定义就可算出为:

$$a_n=\sqrt{\frac{A}{4BM}} \quad (11)$$

于是得:

$$y=\eta(x,z,t)=\sum_{n=1}^M\sqrt{\frac{A}{2BM}}\cos[w_n t+w_n^2(x\cos\theta_n+z\sin\theta_n)]g+\varepsilon_n \quad (12)$$

为进行计算, 在频率区间内必须选择代表频率 w_n , 选取的方法有多种, 既可简单地取为 w_i, w_{i+1} 或 $w_n=(w_i+w_{i+1})/2$, 也可以在 w_i 和 w_{i+1} 之间随机选取。

随机相位采用混合同余法获得,即

$$x_i = Ax_{i-1}^* + c \pmod{M}, \varepsilon_i = \frac{x_i^*}{M} \quad (13)$$

在式(13)中, $M=2^k, k>2; A=1 \pmod{4}$, 即 $A=2^n+1, 1<n<35; c$ 为正整数; x_0^* 为非负整数, ε_i^* 即为循环周期为 2^k 的伪随机数。

本文所建立的随机海浪模型对海浪的描述复杂,其绘制出的外形与实际的波形非常接近,能满足高逼真度场景的需要。但其计算量偏大,应根据实际需要进行简化,否则计算将非常繁琐,根本无法满足实时绘制的需要。为此,可考虑海浪是在某一特定风向向下生成的,即风向角 θ_n 是固定的,这样的话就不用考虑方向谱,大大减少了计算量,默认状况下可设 $\theta_n=0$, 此时也就可以认为 θ_n 是随机分布的。这样除在具体参数的选定方法上不同外,式(12)实质与文献[4]中所用的频谱表示公式是一样的。

3 分形曲面的实现

3.1 海面的分形特性

由大量文献可知,海面是一个复杂的随机粗糙面,它是在大尺度的近似周期性的波浪上叠加着小尺度的波纹,即具有大尺度相关而小尺度相似的特征,并且空间各向近似同性。可见海洋表面呈现分形特征,基于这个特征,可以以实时获取的少量数据为依据,利用分形插值生成具有高度真实感的多分辨率仿真海浪曲面。

本文的基于分形的曲面生成算法中,采用基于 fBm 的分形插值方法中简单而快捷的随机中点位移算法。依据表面构网方式的不同,随机中点位移算法可分为三角形边界法、方形法、菱形法、菱形—方形法,其中三角形边界法和菱形—方形法是两种比较常用的方法,但三角形边界法的缺点是递归过程中各中点的坐标值不容易确定,递归的过程也不容易控制,故采用菱形—方形法。

3.2 菱形—方形法

菱形—方形法又称为 Diamond-Square 法,是将随机中点位移方法用于正方形(或矩形)网格面而生成分形特征曲面。此方法的原理如图 2 所示。取四个点 A、B、C、D 构成正方形,并首先给定这四个点的高度值,然后分别计算正方形中点 O 和各边中点 E、F、G、H 这 5 个新网格点的高度值。在图 2 中,新值显示成 ◆ 点,已经存在的点显示为 ● 点。具体计算方法如下:

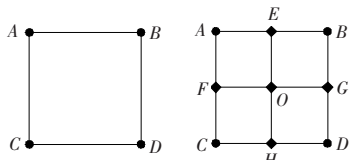


图 2 菱形—方形细分法

中点 O 的高度值为:

$$h_o = (h_A + h_B + h_C + h_D) / 4 + g_o \quad (14)$$

各边中点 E、F、G、H 的高度值为:

$$h_E = (h_A + h_B + h_o) / 3 + g_E \quad (15)$$

$$h_F = (h_A + h_C + h_o) / 3 + g_F \quad (16)$$

$$h_G = (h_B + h_D + h_o) / 3 + g_G \quad (17)$$

$$h_H = (h_C + h_D + h_o) / 3 + g_H$$

这些式子中的 g_o, g_E, g_F, g_G, g_H 分别是不同的随机值。

这样,原来的一个正方形就被分成了四个正方形。以上的细分过程每次都对新形成的正方形重复进行,直到新形成的正方形的边长小于指定的值为止。如果以上重复的次数足够多,将可以得到相当精细的数字高程模型,亦即可得到多分辨率的分形海浪仿真曲面。

Diamond-Square 是一种常用的中点随机位移算法^[5]。如前所述,这种算法是从种子点组成的正方形开始,通过采用若干次随机中点位移方法,不断细分,最后获得逼真的三维场景仿真图,具体描述如下:

假设用 5×5 的数组表示正方形(见图 3), (a) 的四个角上有了初始高度值,表示为 ◆ 点。在此基础上,细分的过程由 Diamond 步和 Square 步两个步骤构成。

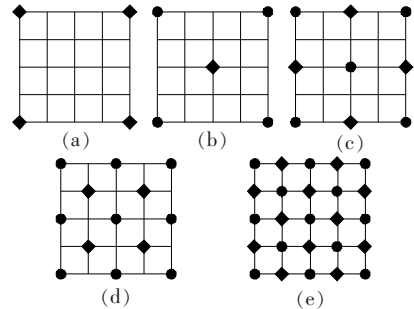


图 3 两次 Diamond-Square 迭代示意图

Diamond 步: 计算正方形中点的高度值(中点为两对角线交点), 中点的高度值是四个顶点的平均值再加上一个随机量计算得到的。在(b)中,新值显示成 ◆ 点,已经存在的点显示为 ● 点。这样,就已经得到了五个点。

Square 步: 计算各边中点的高度值,其计算方法是此中点所在边的两个顶点及正方形中点,共三个点高度的平均值再加上一个与 Diamond 步相似的随机量。经过这一步后,共得到了 9 个点的高度值。这样原来的一个正方形被细分成了多个新的正方形,新的正方形的个数是 2^i , 其中 i 是迭代的次数,由于目前是第一次迭代,所以正方形的个数是 $2^2=4$ 个。

(d)和(e)分别为第二次执行 Diamond 步和 Square 步的示意图。由于此时迭代次数是 2,所以可得到 $2^4=16$ 个正方形,共得到 25 个点的高度值。

Diamond-square 可以用迭代或递归的方法来实现。Diamond-Square 算法形成的浮点数构成的网格,可以用来填充二维分形高度值。这样,就实现了仿真海浪曲面的重建,这种插值方法构成了一个双二次曲面的效果。Diamond-Square 算法的效率较高,所得到的数字高程模型适合使用 $(n+1) \times (n+1)$ 的二维数组来 $H(n, n)$ 表示。当数字高程模型足够精细时, $(n+1) \times (n+1)$ 的二维数组中的每一个数值可以与显示平面中的每一个像素点所代表的高度值相对应。如果把这个二维数组中的某个元素 $H(i, j)$ 的值看成是点 A 在空间坐标的高度坐标,那么,元素 $H(i, j)$ 就唯一地确定了空间坐标系中的一个点 A, 其坐标为 $(i, j, H(i, j))$ 。

3.3 分形实现中的一些问题

(1) 随机值(或随机变量)的取值

一般情况下,随机位移量的数值计算一般采用如下公式:

$$\Delta i = scale \times 2^{-i/H^2} \times gauss() \quad (18)$$

或

$$\Delta i = d_i^H \times \delta \times \sqrt{1 - 2^{-2H-2}} \times \text{gauss} \quad (19)$$

其中: $scale$ 为比例系数,一般在 $(0, 1)$ 内取值; $\text{gauss}() \sim N(0, 1)$, 即为服从标准正态分布的随机函数; H, δ 均为抽取区域的高度变化特征参数值,且都与维数有一定的关系; i 为随机中点位移的递归层次; d_i 为每次递归细分后的线段间距。

上面的计算均要用到高斯函数 $\text{gauss}()$, 而该函数在 C 语言库函数中并不存在,需要依据概率论中的中心极限定理对多个 $\text{rand}()$ 随机数求和取平均来自行定义,因而降低了分形模拟生成的速度。为了达到实时性的要求,可采用如下的计算公式:

$$\Delta i = (\text{rand}() \% (r+1) - r/D) \quad (20)$$

在式(20)中, D 为分形维数, $\text{rand}()$ 为 C 语言库函数自带的伪随机函数, r 的取值与海面起伏特征有关。当然,该数学模型并非严格基于分形布朗运动。

(2) 裂缝或皱褶的消除

如果用式(20)来表达随机位移,仿真中将会发现皱褶现象。通过分析,可采用两个措施来解决:

一方面,每一次迭代过程所使用的随机位移应该是相关的,即在迭代过程中,中点随机位移量是应该逐渐减少的,把式(20)与迭代次数 i 关联起来,形成新的公式:

$$\Delta i = (\text{rand}() \% (r+1) - r/D) \times 2^{-i} \quad (21)$$

在具体的算法中,可取 $i=0, 1, 2$ 三个递归层次。

另一方面,在迭代过程中,要处理好相邻网格公共边的协调性,对于这个问题的解决,已经有很多文献,但在本文的算法实现过程中,由于采用规则的矩形网格,就不需要这些文献提供的方法,只要在编程时,对已生成高程值的中点赋予标记,以后迭代中不再生成该点的高程值即可。

(3) 真实感的增强

在构造分形曲面过程的每一步中引进的随机成分以简单线性迭代的方式加入,就使得分形曲面的精细结构不随分形曲面形态的改变而发生变化,即不能反映景物分形特征是全域不一致的,以致不能更好地表现出自然景物的随机性和丰富细节的生动性,也使得非规则几何形态精细结构的生成容易表现出人为的痕迹,缺乏真实感。对于这个问题,已有文献讨论,主要是采用一种非均匀插值位移分形的方法来解决。同时,真实的海浪曲面的分形是各向异性的,但按本文陈述的仿真方法却是各向同性的。

这两个问题的解决,虽然有助于真实感的进一步增强,但基于仿真的实时性与真实感之间的均衡,本文暂时不作考虑。

4 海浪实时视景仿真的实现

因为海浪场的数据量大,硬件环境也往往是 PC 机,所以海浪场的三维动态显示也是海浪工程应用中的重点和难点。必须在仿真的真实感和实时性之间作出很好的均衡,而且很少有文献涉及到海浪场的三维动态显示研究,所以针对不同的应用,需要不断尝试、总结,以追求更好、更快的显示方法。通过研究和探索,采取了以下一些方法来提高显示性能:

(1) 预先存储稳定状态值

对给定的初始状态值,在模拟时会出现计算突变,而且屏幕上显示的海浪仿真效果真实感非常差。为此,在实现时,应根据给定的初始状态值,预先进行计算,将稳定后的某次状态保

存在外部文件中,以后系统启动时,读入该状态值作为系统的初始状态。

(2) 采用双缓冲区的窗口

在双缓冲区的窗口执行绘图命令实际上是在画面以外创建一个场景,然后很快地交换到窗口的视图图中来,这种方式可以减少单缓冲区时产生的闪烁,OpenGL 要实现双缓冲区的窗口必须先使用函数 $\text{SetPixelFormat}()$ 设置窗口的像素格式支持双缓冲区;每次绘制完成后,使用函数 $\text{SwapBuffers}()$ 交换缓冲区,将图形绘制缓冲区变成窗口的当前缓冲区,以显示图形。

(3) OpenGL 显示列表

显示列表是存储起来的一系列 OpenGL 命令,引用显示列表的时候,按顺序执行其中的 OpenGL 命令。显示列表中的命令都是经过编译的,从而大大提高了执行的效率。在 OpenGL 中的显示列表通过 $\text{glNewList}(\text{ID}, \text{GL_COMPILE})$ 来定义,其中 ID 为显示列表的句柄。仿真海浪的曲面网格在显示之前都预先存储到显示列表中。

(4) Gourand 渲染

在 OpenGL 中,通过函数 $\text{glPolygonMode}(\text{GL_FRONT}, \text{GL_FLIP})$ 设置采用面模型,通过函数 $\text{glShadeModel}(\text{GL_SMOOTH})$ 设置 Gourand 渲染模式,顶点渲染伪代码自动根据三角形顶点颜色对三角面内点的颜色进行平滑处理。

(5) 绘制元素的类型应尽量采用 OpenGL 提供的 $\text{GL_TRIANGLE_STRIP}, \text{GL_TRIANGLE_FAN}, \text{GL_QUAD_STRIP}$, 避免使用单个绘制的 GL_POLYGON 类型。

(6) OpenGL 中的光照计算会很大程度上降低 GPU 的绘制速度,所以在程序中只使用一个无穷远处的点光源,并且在绘制某些场景时,可关闭光源。

(7) 如果想获取屏幕上图像的 RGB 值,写成 BITMAP 格式的文件时, glReadPixels 和 glDrawPixels 中的像素数据类型选择 GL_UNSIGNED_BYTE 。

(8) 在 OpenGL 程序中, $\text{glEnable}()$ 和 $\text{glIsEnabled}()$ 函数是经常被用到的,但如果使用不当,会大大影响程序的运行性能。为此可对其代码进行优化,提高程序执行的效率。

针对以上研究,本文利用 OpenGL 在 PC 机上实现了海浪实时仿真,取得了满意的显示速度和显示效果,如图 4 所示。



图 4 波数为 4 的多分辨率仿真海浪

5 结束语

本文结合模拟飞行训练的需要,通过对海浪时域、频域特性的分析、研究和对大量海浪及海洋观测资料的分析、统计,成功地实现海浪场数据的实时获取。为了使获得的数据更加兼顾实时性和随机性,在获得仿真数据的基础上,提出了基于分形的多分辨率海浪曲面生成算法,实现了具有高度真实感的仿真海浪曲面。最后在以上研究的基础上,根据实际应用的需要,精心设计算法,不断优化三维显示,成功地在 PC 机上实现了满

(下转 55 页)