

南水北调视景仿真系统中动态水环境的模拟

叶海建 刘旭东 罗辉

(中国农业大学 计算机网络中心,北京 100083)

摘要 动态水环境的模拟是南水北调视景仿真的技术难点。在分析现有动态水环境仿真模型的基础上,根据南水北调视景仿真的具体要求,利用水波动理论及计算机图形学的相关方法,实现了南水北调仿真系统视景仿真中动态水环境的仿真。实际模拟结果表明,仿真水面较为逼真,效果良好,能满足水力学仿真的需要。

关键词 仿真;计算机图形;动态水环境

中图分类号 TP 391.9

文章编号 1007-4333(2003)04-0058-05

文献标识码 A

Simulation about dynamic water environment of NSBD simulation system

Ye Haijian, Liu Xudong, Luo Hui

(Computer and Network Center, China Agricultural University, Beijing 100083, China)

Abstract Dynamic water environment simulation is the key of NSBD (North to South Water Transfer Project) scene simulation. By the analyzing of the water simulation method and the need of NSBD Simulation System, this paper give a way which is based on water wave theory and Computer Graphics to develop the dynamic water environment. It is good for the dynamic water environment.

Key words simulation; computer graphics; dynamic water environment

南水北调仿真系统的一个重要组成部分就是视景仿真。视景仿真要求与水利学仿真同步实时,其中要求动态水环境的实时仿真能体现东线水质的变化和中线调水渠道的运行情况。

近年来,国际上在水波和浪花的仿真建模方面已经进行了许多研究。Turner Whitted 是首先试图模拟水波的研究者之一,他使用光线跟踪方法模拟小水池中的反射波,由于光线跟踪方法的速度很慢,无法保证算法的实时性。Ken Perlin 用 Bump Mapping 法模拟海面的空中俯视图,但 Bump Mapping 法生成的波并不展现波浪实际的轮廓边界。Darwyn R. Peachey 用波形函数和相位函数来描述海面,从而使得不同波形的生成更加容易。Alain Fournier 用次摆线描述海洋表面。另外,在参考文献[1~3]中都建立了水动力模型。Peachey 采用高度场表示海面,从而可对几组波进行叠加,Fournier 采用参数曲面表示海面;他们还并考虑了水随着深度

变化所带来的折射及波浪的传播,解决了波峰卷曲的波浪建模问题,但都没有考虑水面随时间的变化情况。在模拟图形的绘制方法上,有的是通过大量粒子的相互作用来模拟流体^[4],有的是基于物理模型的精确方法直接模拟流体^[5],但需求解偏微分方程,达不到实时绘制的要求。目前国内也有不少人有关流体自然现象的仿真进行了研究,如,采用粒子系统模拟飞流瀑布落下后产生的浪花和固体掉入水中溅起的浪花,以及利用波谱函数对随机海面的海浪进行建模,等等。

对于南水北调仿真系统要求动态水环境的实时仿真必须体现东线水质的变化这一问题,主要采用改变动态水环境模型的材质来满足仿真的要求,因为水质按标准可分为6级,不同水质级别,可用不同的材质来表示。该部分内容比较简单,在此不详讨论。本文中主要介绍水力学实时动态仿真的解决办法。

收稿日期:2002-09-27

作者简介:叶海建,教授,主要研究方向为计算机图形图像技术及其应用

1 动态水环境模拟的常用模型

1.1 Bezier 曲面法

Bezier 曲面法的定义^[7]是：设 $p_{ij} (i = 0, 1, \dots, n; j = 0, 1, \dots, m)$ 为 $(m + 1) \times (n + 1)$ 个空间点列，则 $m \times n$ 次 Bezier 曲面定义为

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_{i,m}(u) B_{j,n}(v) p_{ij} \quad u, v \in [0, 1] \quad (1)$$

其中 $B_{i,m}(u) = C_m^i (1 - u)^{m-i}$, $B_{j,n}(v) = C_n^j (1 - v)^{n-j}$ ，是 Bernstein 基函数。依次用线段连接点列 $p_{ij} (i = 0, 1, \dots, n; j = 0, 1, \dots, m)$ 中相邻两点所形成的空间网格，称之为特征网格，而这些点可以称之为控制点（曲线为二维控制点，曲面为三维控制点）。

可以动态地给出 Bezier 曲面的控制点，用 OpenGL 函数生成动态水面（图 1）；此外，可以按一定的波形动态地改变控制点，并选取适当的色彩、材质、光照来满足水利学形态仿真的要求；但该视景仿真中需要模拟大面积的水面，所以在定义水面曲面的时候就需要多个控制点。从式（1）可以看出，曲面上的顶点是各控制点计算求和得到的，所以各个控制点的动作会影响整个曲面，曲面的波形不容易控制，并且随着控制点数目的增加计算量也将增加，这样就影响了图形的实时效果；如果减少控制点的数目，对于面积一定的水面，形态上波长增加，就会失去一定的逼真性。



图 1 Bezier 水面

Fig. 1 Bezier water surface

1.2 非均匀有理 B 样条 (NURBS) 曲面法

由双参数变量分段有理多项式定义的 NURBS 曲面^[7]为

$$S(u, v) = \frac{\sum_{i=0}^m \sum_{j=0}^n W_{ij} P_{ij} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^m \sum_{j=0}^n W_{ij} P_{ij} N_{i,k}(u) N_{j,l}(v)} \quad u, v \in [0, 1] \quad (2)$$

式中： P_{ij} 为矩形区域上特征网格控制点序列， W_{ij} 为相应控制点的权因子， $N_{i,k}(u)$ 和 $N_{j,l}(v)$ 为 k 次和 l 次的 B 样条基函数，它们是在节点适量 $S \{s_0, s_1, \dots, s_{m+k+1}\}$ 和 $T \{t_0, t_1, \dots, t_{n+l+1}\}$ 上定义的。令

$$R_{ij}(u, v) = \frac{W_{ij} N_{i,k}(u) N_{j,l}(v)}{\sum_{i=0}^m \sum_{j=0}^n W_{ij} P_{ij} N_{i,k}(u) N_{j,l}(v)} \quad (3)$$

其中 $R_{ij}(u, v)$ 是 NURBS 曲面的分段有理基函数。若在非均匀参数轴上定义的节点矢量 S 、 T 具有下述形式：

$$S \{ \{0, 0, \dots, 0\}_{k+1}, s_{k+1}, \dots, s_{m+p+1}, \{1, 1, \dots, 1\}_{k+1} \}$$

$$T \{ \{0, 0, \dots, 0\}_{k+1}, t_{k+1}, \dots, t_{m+p+1}, \{1, 1, \dots, 1\}_{k+1} \}$$

则由 S 、 T 定义的曲面就是 NURBS 曲面。通常设定权因子 $W_{00}, W_{0n}, W_{m0}, W_{mn} = 0, W_{ij} = 0, (i = 1, \dots, m - 1; j = 1, \dots, n - 1)$ ，这样可保证基函数为非负。

从 $R_{ij}(u, v)$ 的定义可知，控制点 P_{ij} 和 W_{ij} 仅影响区间 $[s_i, \dots, s_{i+k+1}]$ 、 $[t_i, \dots, t_{i+k+1}]$ 节点矢量范围内的曲面，所以通过控制点和权因子来灵活地改变曲面形状，用来模拟水面，就容易得到满意的波动效果；但同时由于控制点 P_{ij} 和 W_{ij} 只影响其相关区域，所以曲面的搭接和重叠处理相当麻烦，需要更多的存储空间和处理时间，如果要达到满意的实时效果，一般的微机能力是有限的，而对于南水北调这样的大场景，并不需要对水面进行过细的曲面描述。图 2 是用 OpenGL 的 NURBS 曲面相关函数实现的包括灯光和纹理图的真实感水面。



图 2 NURBS 水面

Fig. 2 NURBS water surface

按照 Bezier 曲面和 NURBS 曲面的构造原理，采用 OpenGL 函数模拟水面是非常方便的，并且在水面面积较小的条件下能达到较好的效果，但同时也存在以下问题：Bezier 曲面形状不易控制，NURBS 曲面计算量较大等。这样要满足大面积水面的动态仿真就比较困难。

2 利用水波动理论的动态网格生成动态水面

2.1 动态水面模型

一般情况下,可以将多数随机的正弦波叠加起来描述一固定点的波面。这种对随机变化的表面状态的计算,是基于对不同波数 k 、不同波频数 ω 和不同波传播方向的多重求和这一思想的。可以将水面分为动态水面和静态水面,动态水面由中心区域 $m \times n$ 个网格组成,其外围区域为静态水面(图3), $M \times N$ 表示水面的范围。

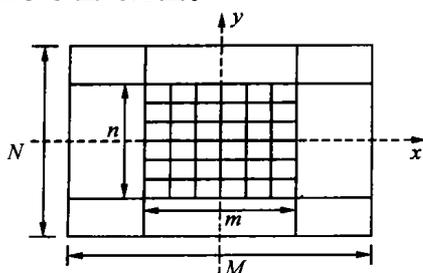


图3 网格模型

Fig. 3 GRID model

程序开始运行时,先计算出图3中每个网格点的物体坐标 (x_j, y_j) , 令 (S_x, S_y) 为视点在 t 时刻的世界坐标位置,将图3的海面网格模型的坐标原点移至视点位置,则网格节点的世界坐标为

$$x_{wj} = x_j + S_x, \quad y_{wj} = y_j + S_y \quad (4)$$

忽略波之间相互作用的非线性因素,当网格节点位于动态水面区域时,网格节点的垂向坐标为^[6]

$$(x, y) = H_{Tide} + \sum_{i=1}^n a_i \cos[k_i(x \sin \theta_i + y \cos \theta_i) + \omega_i t + \phi_i] \quad (5)$$

其中: x 和 y 分别为平行和垂直主导波传播方向的坐标; H_{Tide} 为水面的高度; θ_i 为分布在 $0 \sim 2\pi$ 之间第 i 个成分的随机相位角; θ_i 为波的传播方向; $k_i = \omega_i^2/g$ 为第 i 个成分的波数(由深水波数的色散关系得到); a_i 为第 i 个波的波幅。当网格节点位于静态海面区域时,网格节点的垂向坐标分量为

$$z_{wj} = H_{Tide} \quad (6)$$

由公式(5)可知,改变单元波的波幅 a_i 、角频率 ω_i 可以实现动态水面的流速仿真。

可见,将网格分类可以提高模型的实时性,而且网格模型是根据视点来实时计算的,因此模拟水面范围不必太大,可综合考虑视区与硬件情况而定。系统中采用 $10\,000\text{ m} \times 10\,000\text{ m}$ 的水面,其中动态

水面的范围为 $200\text{ m} \times 200\text{ m}$, 网格间隔为 10 m , 范围和网格间隔可根据成像系统硬件情况进行调整,以满足实时性的要求。

2.2 动态网格模型的实现

整个南水北调视景仿真系统的视景驱动程序,采用高层实时视景管理软件 OpenGVS 编写。OpenGVS 既为用户提供高层的场景管理 API,又提供了对点、线、面等基本图元的支持,并与其底层 OpenGL 有很好的接口。

为了实现动态网格模型,定义一个动态水面的结构 Dynamic.water:

```
typedef struct {
    ...
    vec3      *coord table;
    int       *dy.cindex;      动态区域结点索引
    int       *st.cindex2;    静态区结点索引
    int       *st.cindex3;    静态区边的索引
    int       *st.cindex4;    缓冲区结点索引
    float     dyn.range.x;    动态区 x 方向的
    范围
    float     dyn.range.y;    动态区 y 方向的
    范围
    float     sea.range.x;
    float     sea.range.y;
    float     step.x;        动态区 x 方向网格的步
    长
    float     step.y;        动态区 y 方向网格的步
    长
    int       num.coord.x;    动态区域结点坐标
    int       num.coord.y;
    int       n;             x 方向的网格数
    int       m;             y 方向的网格数
    float     height;
    ...
}Dynamic.water;
```

图4为生成的仿真水面正视网格。

在定义的对象 Dynamic.water 中,整个水面的大小由 sea.range.x 和 sea.range.y 确定,动态水面的区域则取决于 dyn.range.x、dyn.range.y、step.x、step.y 用来定义动态区域网格节点的间隔,coord table 用来存放仿真水面的各节点,如 coord table[i] 表示第 i 个节点, i 为该节点的索引。节点位置的计算,按动

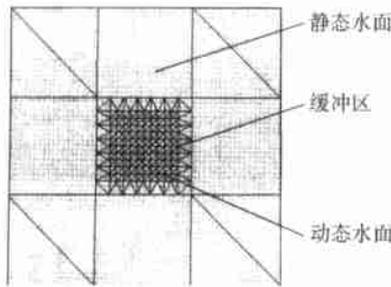


图 4 仿真(网格)水面

Fig. 4 Simulated water surface

态区、缓冲区和静态区的顺序进行,在计算每一个区域时又遵循一定的规则,这样就可以同时确定相应节点的索引。图 4 中,定义 $\text{dyn.range.y} = \text{dyn.range.x} = 50, \text{step.x} = \text{step.y} = 10$,各节点在相对坐标系中 x, y 平面的坐标按下列程序求得:

```

i = 0;      节点的索引
for(row = - m; row <= m; row++)      m
= (int) (dyn.range.y/ step.y) + 0.5);
{
    y = row * step.y;
    for(col = - n; col <= n; col++)      n =
(int) (dyn.range.x/ step.x) + 0.5);
    {
        coord.table[i][X IND] = col * step.x;
        coord.table[i][Y IND] = y;
        coord.table[i][Z IND] = 0.0;      缺省
    }
}

```

由以上程序可见,动态网格 $(-n, -m)$ 处节点索引为 0,是所有节点的第一个节点,根据图(4)所示 3 个不同区域的相对位置关系可以进一步算出缓冲区和静态水面各节点在相对坐标系中 (x, y) 平面的坐标,按计算顺序建立相应的索引,并将动态区域的索引存入 dy.index ,缓冲区域的索引存入 st.index ,而静态区域的索引存入 $\text{st.index2}, \text{st.index3}$;并且还要计算相应节点的纹理坐标和各网格的表面矢量,同时也要将表面矢量的索引存入 norm.index ,以便真实感水面的生成,这样在 x, y 平面的仿真水面就可以确定了。

要实现水面的实时仿真,就要实时计算仿真水面各节点的垂向分量,从而实现一个动态的三维网

格,而其静态区域各节点的垂向分量可以用式(6)计算得到。由式(5)可知,如果给定一组波,则有

$$(x, y) = H_{\text{Tide}} + \sum_{i=1}^n a_i \cos[x K_{xi} + y K_{yi} + it + \phi_i] \quad (7)$$

在此,如果给定 Num.waves 个波,分别定义: $W[\text{MAX WAVES}]$ 为角频率; $Kx[\text{MAX WAVES}]$ 为 K_{xi} ; $Ky[\text{MAX WAVES}]$ 为 K_{yi} ; $\text{Amp}[\text{MAX WAVES}]$ 为 a_i ,即第 i 个波的幅度; $\text{Phi}[\text{MAX WAVES}]$ 为 ϕ_i ,即波的随机相位角。那么就可以得到这 Num.waves 个波叠加后仿真水面各节点在垂向的分量;

```

wt[Num.waves] = W[Num.waves] * current.time;
current time 为系统计时
i = 0;      从索引为 0 的节点开始
for(row = 0; row < num.coord.y; row++)
num.coord.y = 2 * n + 1;
{
    for(col = 0; col < num.coord.x; col++)
num.coord.x = 2 * m + 1;
    {
        x = coord.table[i][DB.WEST.IND] + eye.x;
        y = coord.table[i][DB.NORTH.IND] + eye.y;
        z = 0.0;
        for(j = 0; j < Num.waves; j++)
        {
            ang = wt[j] + (Kx[j] * x) + (Ky[j] * y) + Phi[j];
            disp[j] = height + cosf(ang) * Amp[j];
            可取水面高度 height = 0;
            z += disp[j];
        }
        coord.table[i][DB.VERT.IND] = z;
        i++;
    }
}

```

仿真水面静态区域各节点的垂向分量由式(6)可知为: $z_{wj} = H_{\text{Tide}} = \text{height} = 0$,可以取缺省值,在此不必计算,水面的高度可以在视景驱动应用程序中动态确定。

利用 OpenGL 库函数;

```
glBegin();
glVertex3f(x,y,z);
glEnd();
```

可以生成如图 4 的网格面,同时由:

```
glColor4f(r,g,b,a);
glNormal3f(x,y,z);
glTexCoord2f(u,v);
```

可定义仿真水面的颜色、真实感纹理和表面矢量来生成具有真实感的动态仿真水面(图 5)。

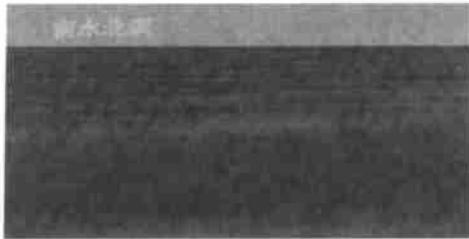


图 5 真实感水面

Fig. 5 Reality water surface

3 结束语

利用基于水波动理论建立的动态网格模型,能够对模拟水面进行较好的控制,并减少计算量,最终形成的虚拟场景中仿真水面较为逼真,效果较好,并能满足水力学仿真的需要。但因为波浪的形成比较

复杂,又有各种随机因素的影响,本文中所论述的模型及其实现条件都是在理想情况下进行的。随着计算机图形处理技术的不断提高,以及有关水动力现象和参数研究的不断完善,用计算机模拟出更加逼真的三维动态虚拟水环境是完全可能的。

参 考 文 献

- [1] Peachy D. Modeling waves and surf[A]. Proceedings of SIGGRAPH86[C], Louisiana:ACMSIGGRAPH,1986. 65 ~ 74
- [2] Fournier A, Reeves W. A simple model ocean waves[A]. Proceedings of SIGGRAPH86[C], Louisiana:ACMSIGGRAPH,1986. 75 ~ 84
- [3] Tsö.p, Barsky B. Modeling and rederi waves[J]. ACM Transactions on Graphics, 1987,6(3):191 ~ 214
- [4] Michael E, Goss A. Real time particle system for display of ship waakes[J]. IEEE Computer Graphics & Applications,1990,272(7):30 ~ 35
- [5] Michl K, Gavin M. Rapid, stable fluid dynamics for computer graphics[J]. Computer Graphics, 1990,24(4):49 ~ 54
- [6] 曾芬芳,黄建国.虚拟海洋场景中波浪的模拟[J].计算机应用,2000,21(4):33 ~ 36
- [7] 孙家广.计算机图形学[M].北京:清华大学出版社,1994.262 ~ 331

(上接第 35 页)

1) 采用面向安全的体系结构,通过 USB 和 DPRAM 与外电路接口,内部总线不外露。

2) 自主开发了 16 bit 的栈式结构 CPU 及其指令系统。

3) 支持对称和非对称加/解密,实测 1 024 bit 的 RSA 加/解密速率为 5 次 s^{-1} ,DES 加/解密速率达 2 Mbit s^{-1} 。

严格的软件仿真和物理验证结果表明,系统工作稳定,性能可靠,速度快;达到了设计要求。目前,

GIDC-C 密码算法芯片已通过验收,并已开始投产生产。

参 考 文 献

- [1] Jozwiak L. Quality-driven system on a chip design[J]. IEEE 2000 First International Symposium on Quality Electronic Design, 2000,6(3):93 ~ 102
- [2] 王志华,邓仰东.数字集成系统的结构化设计与高层次综合[M].北京:清华大学出版社,2000:80 ~ 101
- [3] 夏宇闻.复杂数字电路与系统的 Verilog HDL 设计技术[M].北京:北京航空航天大学出版社,1998:18 ~ 30