

文章编号:1671-6833(2011)02-0088-05

## 应用折射反射纹理的实时水面仿真技术

谭同德, 赵 硕, 赵红领

(郑州大学 信息工程学院, 河南 郑州 450001)

**摘 要:** 水面效果可以大大增强虚拟现实环境的沉浸感, 设计了一种实时水面仿真算法, 该算法采用两个三角形单元模拟水面的方法, 综合运用了渲染到纹理、凹凸纹理、投影纹理映射、纹理扰动技术来模拟水的反射、折射、波动、倒影效果; 并提出了一种改进的纹理扰动技术, 设置 Alpha 值确定应该扰动的纹理, 避免了扰动错法, 使水面更加逼真. 实验表明, 该方法在系统绘制的真实感和实时性上取得了较满意的优化效果, 较好地解决了虚拟环境中水面实时绘制问题.

**关键词:** 实时水面; 渲染到纹理; 凹凸纹理; 反射折射纹理; 纹理扰动

**中图分类号:** TP391

**文献标志码:** A

### 0 引言

水面模拟是实时计算机图形学的研究热点之一. 现在流行的水面模拟算法有 3 种: 基于网格的欧拉法<sup>[1]</sup>、基于平滑粒子流体动力学的拉格朗日法<sup>[2]</sup>和基于平面的反射折射算法. 前两种算法模拟真实, 但是要想得到逼真的效果需要将水面大量细分, 计算量大, 不利于大面积水面的实时模拟. 笔者采用基于平面的反射折射算法, 运用渲染到纹理(Render To Texture, RTT)的技术, 使用两个三角形单元模拟大面积的水面, 解决了有物体在水面前导致视觉走样及扰动走样的问题. 该方法模拟大面积水面效率高, 效果与性能都很好, 可广泛地应用在 3D 游戏及系统仿真<sup>[3]</sup>中.

### 1 水面模拟相关研究

#### 1.1 渲染到纹理技术

纹理是在内存中的一段线性地址, 其中存储了颜色和其他信息, 这张纹理被称为渲染目标(render target). 渲染到纹理技术的实质就是对这段线性内存进行多次渲染, 达到要求后, 再读回渲染后的纹理数据, 粘贴到物体表面, 产生物体的真实感效果.

#### 1.2 获得折射场景纹理

##### 1.2.1 渲染水下折射场景

模拟真实的水面, 需要在每一帧中根据视点的

方向, 将水下折射场景渲染到渲染目标中. 为此, 首先附加一个额外的裁剪面, 这个裁剪面与水面重合, 裁剪掉场景中所有水面以上的部分, 将水面以下部分正确渲染到一个渲染目标  $S_1$  中(见图 1).

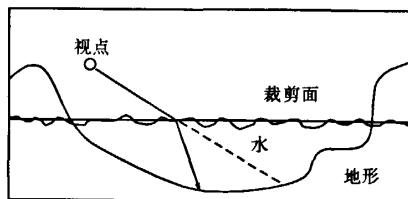


图 1 经过折射的水下场景

Fig. 1 The refractive scene undersurface

##### 1.2.2 根据水深修正场景亮度

水下物体的亮度会随着距水面深度的增加不断减小, 人眼对此的直观反应是水下物体离水面越深, 表面颜色越暗淡. 所以, 渲染目标  $S_1$  中的颜色通道  $r, g, b$  中的亮度值应该依其距水面的深度递减. 为了计算每个颜色通道的这种亮度变化, 我们使用经验公式:

$$F_c = \exp(-d \cdot n) \quad (1)$$

式中:  $F_c$  为最终的颜色亮度值;  $d$  是当前距水面的深度; 而  $n$  的值取颜色通道亮度值的一半的倒数, 即  $n(r, g, b) = (2/R, 2/G, 2/B)$ ;  $R, G, B$  分别为颜色通道相应分量的亮度.

##### 1.2.3 将最终场景投影到纹理中

由上两步的工作, 可以得到一张亮度正确的

收稿日期: 2010-06-20; 修订日期: 2010-11-20

基金项目: 河南省自然科学基金项目(0611051900)

作者简介: 谭同德(1950-), 男, 河南郑州人, 郑州大学教授, 博士, 主要研究计算机图形学及虚拟现实.

水下场景的纹理.但是在生成纹理之前,场景还处于世界空间中,坐标范围也不是标准的纹理空间范围 $[0,1]$ ,所以要把场景变换到投影坐标空间中,并把坐标范围变换为 $[0,1]$ ,这样才可以得到投影到水面上的正确的折射纹理.相关的投影矩阵公式为:

$$M_p = M_v \cdot M_o \cdot M_t \quad (2)$$

式中: $M_o$ 是世界坐标空间中的摄像机(即视点)的视点矩阵; $M_t$ 是摄像机的投影矩阵; $M_o \cdot M_t$ 将场景由世界空间变换到投影空间中,投影空间的范围为 $[-1,1]$ .为了把坐标由范围为 $[-1,1]$ 的投影空间映射到范围为 $[0,1]$ 的纹理空间,必须再乘以一个特殊的矩阵 $M_v$ ,这个矩阵将平移和缩放场景,使其由投影空间转换到纹理空间.

### 1.3 获得反射场景纹理

为了逼真的模拟反射场景,在每一帧都要把有倒影的物体翻转到水下去后渲染到一个渲染目标 $S_2$ 中<sup>[5]</sup>.因此,需要将世界坐标空间中的原始视点 $P_o$ 以水面为对称轴翻转,翻转的同时附加一个裁剪面,将水面以下的场景裁剪掉,如图2所示.

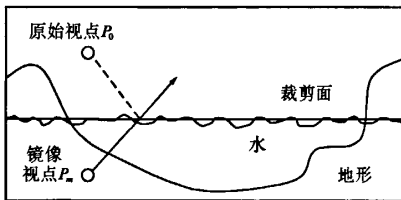


图2 反射场景的原始视点翻转

Fig.2 The view point reversed in reversing scene

翻转原始视点 $P_o$ 的位置,把整个场景以水面为对称轴翻转到水面下,得到新的镜像视点 $P_m$ 的视点矩阵 $M_c$ 如下:

$$M_c = M_m \cdot M_o \quad (3)$$

设原始视点 $P_o$ 的位置为 $(x, y, z)$ ,高度为 $h$ ,则镜像视点 $P_m$ 位置变为 $(x, -y + 2 \cdot h, z)$ ,矩阵 $M_m$ 可以表示为:

$$M_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 2h & 0 & 1 \end{bmatrix} \quad (4)$$

水上的反射场景是用镜像摄像机渲染的,为了将反射场景投影到水面上,投影矩阵<sup>[6]</sup>根据上述讨论变化为:

$$M_p = M_c \cdot M_o \cdot M_t \quad (5)$$

### 1.4 渲染水面

#### 1.4.1 菲涅耳效果<sup>[7]</sup>

水面的反射折射效果主要取决于观察者的视点,当视点向量和水面法向量成 $90^\circ$ 时,水面将呈现100%的反射效果;当二者成 $0^\circ$ 时,水面将呈现100%的折射效果.数学模型如下:

$$C = WC_r + (L - W) \cdot C_L \quad (6)$$

$$W = w(\alpha) = \max(\cos\alpha, 0) \quad (7)$$

式中: $W$ 的范围是 $[0,1]$ ;  $\alpha$ 为视点向量与水面法向量所成的角度;折射的分量系数为 $W$ ,反射的分量系数为 $L - W$ .当 $\alpha$ 从 $0^\circ$ 到 $90^\circ$ 变化时,折射分量减少而反射分量增加.  $C_r$ 变量是折射的颜色,  $C_L$ 变量是反射的颜色,它们的颜色值分别从折射贴图 $S_1$ 与反射贴图 $S_2$ 中查找.

#### 1.4.2 凹凸映射技术

采用法向量凹凸纹理映射<sup>[8]</sup> (Bump Mapping)技术来模拟水面的波动.计算水面凹凸映射时预先准备好一张预处理过的水面的法向量纹理图,首先将传入的光源向量单位化,然后查询纹理法向量图获得向量值并映射到 $[-1, 1]$ 区间即为相应顶点的法向量,把单位化的光源向量和顶点法向量做点乘获得一个算子,这个算子得出了该点接收反射光照和折射光照的系数,该系数乘以该点本身的颜色得到最终颜色.用该算子查询获得的颜色乘积作为输出颜色,便获得了凹凸明暗的水面颜色效果,模拟了水面的褶皱与波纹;然后使用菲涅耳效果处理水面纹理并与之前的折射反射纹理相混合,就会得到最终的水面效果.

#### 1.4.3 纹理扰动

真实水面的波动会造成水面倒影的扭曲.单靠环境映射模拟反射水面周围的环境,利用凹凸纹理展现水面的凸凹不同不足以展示这种效果,笔者通过在光照投影过程中动态的卷动法向量图的 $xz$ 坐标对传入的反射、折射纹理进行纹理扰动,来获得水面倒影的扭曲效果,最终水面的颜色由反射纹理、折射纹理及水面颜色混合而成.

## 2 水面模拟方法的研究改进

### 2.1 视觉走样修正

经分析研究,上述水面模拟技术存在视觉走样的缺陷,当一个物体挡在水面前面时,如果纹理扰动值过大,就会出现错误.如前所述,扰动的纹理包括视点所能看到的所有几何体,如果一个物体处于水面前面,且处于视点的可视范围内,那么这个物体将会被渲染到场景纹理中,而在渲染水

面对这张纹理的每个像素不加区别地进行扰动,导致了视觉走样,使人误认为物体的一部分也发生了折射,如图3(a)所示。

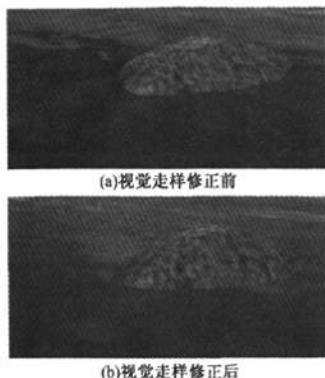


图3 视觉走样修正

Fig.3 Visual aliasing Amendment

改进的渲染算法是区别应该扰动与不应该扰动的像素,只扰动应该扰动的像素。具体步骤如下:①将纹理的 Alpha 通道清为白色,正常渲染场景。②将水面的 Alpha 通道绘制为黑色。③只扰动 Alpha 通道为黑色的部分,即只扰动水面而不扰动水面的物体,获得了正确的效果。图3(b)为实验效果,水面前方有一块小岛,小岛上的纹理并没有错误的扰动。

## 2.2 反射贴图修正

反射贴图技术同样存在缺陷:在每一帧都要把有倒影的物体翻转到水下去后渲染,这种不加区别地对所有水面上场景的翻转导致在扰动反射贴图时出现了扰动走样(图4(a))。原因是反射场景是有层次的,比如对于水面附近的场景来说大致分为3个层次:水面边沿的水岸及岩石、水岸后的植被、植被后面的房屋或背景。这些层次是从前到后依次遮挡的,对它们不加区别地进行纹理扰动,再加上折射、凹凸纹理对纹理像素的近似处理就会导致走样。图4(a)是走样:山后的背景天空经扰动后出现在山与水连接的地方,产生了走样。

为了修正扰动走样,需要对视觉走样修正进行改进,分层次渲染:①将纹理的 Alpha 通道清为白色,正常渲染反射场景。②将水面的 Alpha 通道绘制为灰色。③从倒数第二层次开始,从后向前依层次绘制水面的 Alpha 通道,绘制的颜色均比灰色深,且每次颜色深度递增,第一层加深为黑色。

这样一来得到的反射纹理可以根据 Alpha 通道准确地判断场景纹理所在的层次,然后依层次

对水面进行纹理扰动(设图4中有2个层次:地形、天空)。

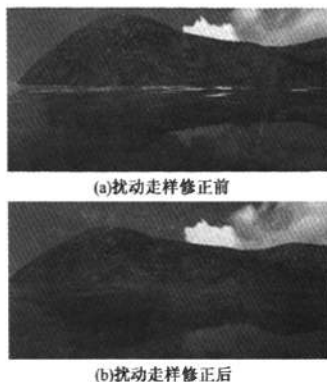


图4 扰动走样修正

Fig.4 Disturbance aliasing Amendment

对于水面中某像素点:

①若该点处于层次2(天空)中,则该点可以被层次1(地形)遮挡,扰动后的像素值即为该点的最终颜色。②若该点处于层次1中,则该点不能被层次2遮挡,需要判断:若扰动过来的像素值处于层次1中,扰动后的像素值即为该点的最终颜色;若扰动过来的像素值处于层次2中,抛弃扰动后的像素值,该点值不变(即不扰动)。

使用上述扰动修正算法仿真,取得了不错的效果,如图4(b)所示。

## 2.3 实时水面仿真算法流程

根据上述分析,对算法进行修改,改进后的实时水面仿真算法流程如图5所示。

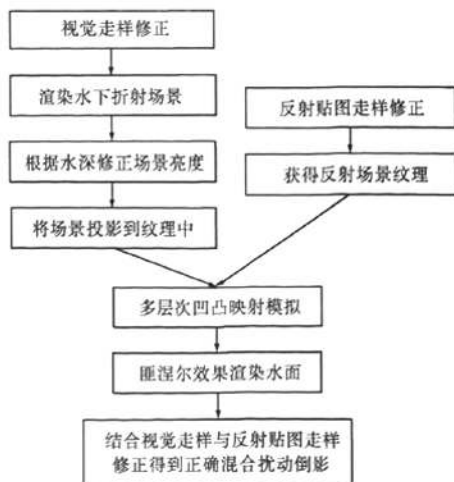


图5 实时水面仿真算法流程图

Fig.5 Algorithm procedure of real-time water surface simulation

先对水下的折射场景进行视觉走样修正,再渲染水下折射场景,根据水深计算出折射场景各点的亮度,把计算好的场景投影到一块纹理区间中.采用并行计算同时对反射贴图进行分层,将渲染后的反射场景投影到另一块纹理区间中.然后对折射场景和反射场景进行凹凸映射,合并在一起,按照菲涅尔效果混合折射和反射的场景.最后利用视觉走样修正和反射场景分层设定的 Alpha 值对场景进行混合扰动,渲染出正确的逼真水面.

### 3 水面整体实验实现

整体的水面实验效果如图 5,实验数据如表 1、表 2 所示.实验硬件平台为:CPU P4 3.0 G,内存 512 MB,显卡 GeForce 7800 GTX (256MB),操作系统 Windows-XP-sp2,开发工具 VS. net2003,3D 图形开发库为 Dire-ctX 9.0c,CG 库为 Cg2.0.



图 6 水面整体实验效果图

Fig.6 Overall experimental results

表 1 是改进前后算法性能的对比数据,可以看出:随着场景中三角图元的增加,渲染水面对整个场景绘制帧速率的影响减小,影响帧速率的主要因素是场景中待渲染的三角图元个数.采用本文的改进算法随着场景中三角图元的增加优势更为突出.

表 1 基于平面的水面绘制算法改进前后性能对比

Tab.1 Algorithm performance contrast

三角图元个数	帧速率/(帧·s <sup>-1</sup> )		
	未进行水面绘制	原始水面绘制算法	改进水面绘制算法
15 514	71	58	56
29 762	45	42	41
43 522	36	34	34

对比改进前后的算法性能可见:在三角图元个数较少的情况下改进后的算法帧速率略低于原始算法,但是其逼真效果远高于改进前算法;在场景复杂度较高时(图中是 43 522 个图元)算法改进前后的帧速率基本相同.

表 2 为 3 种流行的水面模拟算法的对比数据,采用要模拟的水面面积大小来对算法的性能进行度量,其中 1 m<sup>2</sup> 水面由 1 000 × 1 000 个网格

(欧拉法)或 1 000 × 1 000 个粒子(拉格朗日法)组成,改进的水面绘制算法由两个三角图元覆盖整个水面.由表中数据可知:在面积为 1 m<sup>2</sup> 时,改进后的水面绘制算法帧速率为 1 522 帧/s,远大于欧拉法与拉格朗日法,但是拉格朗日法效果的逼真度要高于欧拉法和本文使用的算法;当水面面积增加为 1 000 × 1 000 m<sup>2</sup> 时,欧拉法的帧速率骤降为 36 帧/s,而使用拉格朗日法渲染的程序以当前配置已无法运行,本文采用的改进水面绘制算法仍可流畅运行在 340 帧/s;当水面面积达到 10 000 × 10 000 m<sup>2</sup> 时,欧拉法帧速率仅有 4.5 帧/s,使用拉格朗日法的程序不能运行.欧拉法及拉格朗日法无法胜任大规模的水面模拟,而本文使用的改进算法依然可以达到 26 帧/s 的帧速率,保持流畅的运行效果.

表 2 3 种水面模拟算法性能对比

Tab.2 Three Water surface simulation algorithm performance comparison

水面面积/m <sup>2</sup>	帧速率/(帧·s <sup>-1</sup> )		
	欧拉法	拉格朗日法	改进水面绘制算法
1 <sup>2</sup>	538	34.2	1 522
100 <sup>2</sup>	109	2.6	912
1 000 <sup>2</sup>	36	无法运行	340
10 000 <sup>2</sup>	4.5	无法运行	26

可见:基于平面的反射折射技术的改进算法适用于中大规模的水面模拟,在较大规模水面模拟时,只有基于平面的反射折射技术的改进算法可以保证实时流畅运行.

### 4 结论

实时水面模拟是计算机图形学和虚拟现实领域研究的热点,笔者使用纯纹理渲染的方法,通过反射、折射、凹凸纹理、纹理扰动、菲涅尔效果这些技术的综合应用,在不过大增加系统负担的情况下实现虚拟水面的绘制;针对算法中存在的视觉走样与反射贴图走样的缺陷,提出了视觉走样修正及对场景分层次渲染的反射贴图走样修正算法,解决了上述缺陷,既增强了场景真实感,又保证了绘制速度,在系统绘制的真实感和实时性上取得了较满意的优化,较好地解决了虚拟环境中水面实时绘制问题.

### 参考文献:

- [1] 虞宏毅.大规模水场景的实时绘制[D].杭州:浙江大学计算机学院,2007.
- [2] 廖敏.实时可交互自然水面的模拟研究[D].上海:

- 上海交通大学计算机科学与技术系,2007.
- [3] 涂超. 基于位图的大规模地形景观生成技术研究[J]. 郑州大学学报:工学版,2003,24(1):46-49.
- [4] 马骏,朱衡君,龚建华. 基于 Cg 和 OpenGL 的实时水面环境模拟[J]. 系统仿真学报,2006,18(2):472-474.
- [5] YANN L. Realistic natural effect rendering [EB/OL]. <http://www.gamedev.net/reference/articles/article2138.asp>,2004-09-7/2009-12-23.
- [6] HUN Y K. Virtual camera position [EB/OL]. <http://www.gamedev.net/reference/articles/article2023.asp>,2003-02-13/2008-12-23.
- [7] 洪伟,刘亚妮,李骑.Cg 教程可编程实时图形权威指南[M]. 北京:人民邮电出版社,2004:179-209.
- [8] MATTHIAS W. Fresnel reflection [EB/OL]. [http://developer.nvidia.com/object/fresnel\\_wp.html](http://developer.nvidia.com/object/fresnel_wp.html),2004-07-14/2008-12-23.

## Real-time Water Surface Simulation based on Refraction and Reflection Texture

TAN Tong-de, ZHAO Shuo, ZHAO Hong-li

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** The effects of water surface can greatly enhance the sense of immersion in virtual reality. This paper described an algorithm of the fast real-time surface simulation. The simulation of water surface for any scale was constructed by only two triangles, some technologies including Render-to-Buffer, Bump Mapping, Projective Texture Mapping and Texture Perturbation, were used to simulate the water surface with effect such as reflection, refraction, wave, and inverted reflection. Then the method of Texture Perturbation was proposed, which will make the surface more realistic. The experiments show that the optimum effect between realistic and real-time performance is obtained and the question of rendering a water surface in real-time in virtual environment is resolved well.

**Key words:** real-time water surface; render-to-buffer; bump mapping; refraction and reflection texture; texture perturbation

(上接第 87 页)

## A Novel Channel Estimation Method for OFDM Time-Varying Channel based on The Particle Filter

MU Xiao-min, CAO Li-guo, LU Yan-hui

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** A joint estimation method of the model parameter and time-varying channel states, which was based on the particle filter (PF) algorithm, was proposed for OFDM system. By modeling the time-varying channel for AR model of dynamic and time-varying coefficient, the model parameter was estimated by introducing the particle filter algorithm and kernel smoothing contraction technology. Besides, channel states were estimated later. Finally, it realizes the joint estimation of the channel states and the model parameter. Simulation results show that the above mentioned method has more significantly improved in the estimation precision and system performance compared with the traditional channel estimation method with AR model of constant coefficient.

**Key words:** time-varying channel; dynamic estimation; particle filter; kernel smoothing contraction