

# 基于球面的海浪建模与绘制技术研究

李苏军 蒋杰 杨冰 吴玲达

(国防科学技术大学多媒体研究开发中心, 长沙 410073)

**摘要** 由于传统的海浪建模与绘制方法多是基于平面进行建模与绘制,较少细致考虑球面曲率对海浪建模和绘制的影响,因此无法在球形表面绘制出海浪在不同条件下形状、色泽等的变化。为了实现无界海洋的虚拟自由漫游,提出了一种考虑球面影响的、任意视点下的海浪建模与绘制方法,其核心是球面海洋模型和屏幕细分的自适应算法,同时基于海洋学知识以及解析几何和数字地球特点,首先解决和消除了模型生成中存在的计算量大和高纬度“压缩现象”等问题,并建立了球面海浪模型;然后,依靠可逐点计算的海浪生成模型,利用屏幕细分自适应算法将计算限制在与视点相关的可视球面海洋区域,并使几何分辨率依视点变化;接着采用“夹球”技术,解决了海面生成中存在的“边缘裂缝”现象;最后,采用GPU技术绘制出不同观察距离和方向下的真实感海浪场景,进而实现了用户在无界海洋上的自由漫游。

**关键词** 海浪建模与绘制 球面海洋模型 视点相关细节层次模型 屏幕细分自适应算法

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2009)04-0744-09

## Resear on Sphere Based Modeling and Rendering of Ocean Waves

LI Su-jun, JIANG Jie, YANG Bing, WU Ling-da

(Multimedia R&D Center, National University of Defense Technology, Changsha 410073)

**Abstract** Traditional plane based modeling and rendering of ocean wave lacks in consider ations on the effect of globe curvature in detail, thus fails to realistically simulate the change of shape, color of the ocean waves on sphere. In this paper, considering sphere shapes influence and free viewpoint, a new method of modeling and rendering of ocean wave is presented. Its kernel is the modeling of ocean waves on sphere and the screen-subdivision algorithm. The ocean wave model based on the knowledge of ocean, analytic geometry and the characters of digital globe is set up first. It reduces the computation obviously and resolves the “compress phenomenon” at high latitude. Then, relying on a procedural ocean wave model, the method restricts computations to the visible part of the ocean surface using the screen-subdivision algorithm and adapts the geometric resolution depending on the viewing distance. The “border slit” on screen can be removed by the “Clamp Sphere”. Finally, the realistic scenes are real-time rendered based on GPU. It allows the user to interactively fly over an unbounded animated ocean.

**Keywords** ocean wave modeling and rendering, sphere based modeling of ocean waves, view-dependent levels of detail, screen-subdivision adaptive algorithm

收稿日期:2007-05-28; 改回日期:2007-09-14

第一作者简介:李苏军(1972 ~ ),男。现为国防科学技术大学系统工程专业博士研究生。研究方向为虚拟现实、图形图像处理。

E-mail: lisujun2004@163.com, lsj2008@nudt.edu.cn

## 1 引言

真实感海面的建模与绘制是计算机图形学研究的热点和难点之一<sup>[1-2]</sup>。在虚拟战场、航海训练、计算机游戏动画、海洋渔业、海洋预测预报、影视娱乐等领域具有广泛的应用。

20世纪80年代早期,便利用计算机对海浪建模和绘制进行了研究。1978年,Blinn提出了凹凸纹理映射方法(bump mapping)<sup>[3]</sup>。该方法虽然比在光滑表面添加纹理好,但不能生成真实感较强的海面,逼真性较差。针对凹凸纹理映射方法存在近视点真实感不强的问题,1980年,Fishman和Schachter提出了高度场方法<sup>[4]</sup>,Max后来又加以改进<sup>[5]</sup>。虽然文献[5]方法能较好地生成连续帧动画,但不能生成如波峰向前卷曲等复杂的海浪景象。1986年,Fournier等人采用Gerstner-Rankine模型<sup>[6-7]</sup>方法生成了包括折射场景在内的具有一定真实感的海浪场景<sup>[8]</sup>。该方法不需大量的网格转换计算,但需合理地调整参数和添加一些背景噪声。1995~2000年,Foster和Chen等人采用Navier-Stokes方程模拟海浪<sup>[9-11]</sup>。该方法虽能生成真实感较强的海浪,但因边界条件及受力随时空变换、计算量大等原因,使方程的求解非常困难,目前还不能满足实时性要求较强的应用。2001年,Premoze和Ashikhmin提出了一个基于物理的方法,用来生成水波<sup>[12]</sup>,并用非实时光线传播方法计算了海洋的复杂光学效果。2001年,Jensen和Tessendorf等人分别借鉴海洋统计和经验模型,采用大量正弦波叠加,通过快速傅里叶变换(FFT)模拟海浪<sup>[13-14]</sup>。该方法的绘制结果具有一定程度的真实感,但由于需要采用规则矩形粗网格进行实时绘制和要避免FFT方法产生的视觉重复性,因此降低了图像的质量和真实感。2004年,Johanson采用Perlin噪声和投影网格方法<sup>[15]</sup>,实时生成了真实感较强的海面,该方法虽可以产生真实感很强的水面,但需要进行噪声的复杂计算,对效率影响很大,且仅适用于小面积海面的生成。近年来随着GPU技术的迅猛发展,许多文献中的方法都利用GPU技术来加速海浪绘制,从而提高了实时性<sup>[15-20]</sup>。

国内对于海洋场景真实感绘制方面的研究相对较少。徐迎庆等人采用水力学方程组建立海面模型<sup>[21]</sup>,生成了真实感较强的海面夜色,还用粒子系

统模拟了瀑布浪花的飞溅效果,但方法的实时性较差。李广鑫、王磊等人采用Perlin噪声作为水面高度场的激励源,分别使用准均匀B样条曲面和NURBS曲面来构造连续动荡变化的水波面<sup>[22-23]</sup>;王长波等人提出了基于元胞自动机(cellular automata)的实时海浪模拟方法<sup>[24]</sup>,尹勇等人提出了基于浪级划分和海浪谱的实时模拟方法<sup>[25]</sup>。鄢来斌、解翠等人分别采用视场分割算法、改进的实时海浪几何模型来生成动态多分辨率海面网格<sup>[26-27]</sup>,提高了海洋模拟的实时性。皮学贤、陆志慧等人采用噪声分形面和GPU技术生成了真实感较强的实时动态海洋场景<sup>[28]</sup>。

但上述工作中的大多数海浪模拟是基于平面,且采用波谱方法生成<sup>[29]</sup>,较少考虑地球曲率影响、数字地球表面可视区域计算等问题,不能直接应用于数字地球平台上。因此就迫切需要研究在数字地球上进行海洋建模和绘制技术,即球面海浪建模与绘制技术。球面不同于平面,是造成可视区域计算和动态连续多分辨率采样较难的主要原因,其中较显著的难点包括不同经纬度条件下的球形表面海浪建模、球形表面可视区域计算、连续动态多分辨率网格生成等。

针对以上问题,本文提出了一种考虑球面曲率影响和任意视点条件下的可视海洋区域海浪建模与绘制方法,其核心是球面海洋模型和屏幕细分自适应算法。该算法首先基于海洋学知识以及解析几何和球面特点,建立了球面海浪模型;然后,采用屏幕细分自适应算法生成任意视点、视向条件下可视区域的连续多分辨率网格;最后,采用GPU技术实现真实感海浪场景的实时绘制。

## 2 球形海洋表面建模

### 2.1 基于Perlin噪声的平面海洋表面模型

1985年,Perlin提出Perlin噪声函数<sup>[30]</sup>。Perlin噪声函数是指能在空间中产生连续噪声的函数。所谓连续噪声,是指对于空间中的任意两个点,当从一个点移动到另外一个点时,噪声的值是平滑变化的。

图1展示了如何将八阶Perlin噪声叠加成一个分形噪声的过程。噪声的特性由每一阶噪声的放大系数决定。其生成算法描述如下:

$$h_{\text{noise}}(x, y) = \sum_{i=0}^{N-1} a^i \cdot \frac{n(2^i \cdot x, 2^i \cdot y)}{2^i} \quad (1)$$

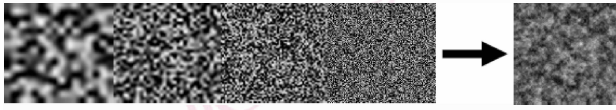


图 1 噪声分形面

Fig. 1 The surface of noise fractal

式中,  $i$  表示第  $i$  个被叠加的 Perlin 噪声函数;  $a^i$  为加权幅度, 它的大小决定了分形面的幅度, 一般加权幅值与频率成反比。

上述方法可构造一个均匀的噪声分形面。由于海面网格点的位置是动态变化的, 因此海面噪声高度场必须动态生成。根据海面网格点的平面坐标  $(x, y)$  可以利用式(2)得到该点的高度值。

$$h_{\text{heightfield}}(x, y) = h_{\text{plane}}(x, y) + h_{\text{noise}}(x, y) \cdot N_{\text{plane}} \quad (2)$$

其中,

$$-180 \leq x \leq 180, -90 \leq y \leq 90$$

$h_{\text{heightfield}}$  为平面上点  $(x, y)$  的海面高度值;

$h_{\text{plane}}(x, y)$  为平面上点  $(x, y)$  的标准海面高度;

$h_{\text{noise}}(x, y)$  为点  $(x, y)$  的噪声高度值;

$N_{\text{plane}}$  为点  $(x, y)$  切平面的法向量。

利用式(2)生成的平面海面噪声网格如图 2 所示。当  $y=0$  时, 点  $(x, y)$  是一系列点, 利用  $f(x, y)$  可得到该点的噪声高度值; 当  $z = \pm 90$ , 点  $(x, y)$  也是一系列点, 利用  $f(x, y)$  也可得到该点的噪声高度值。

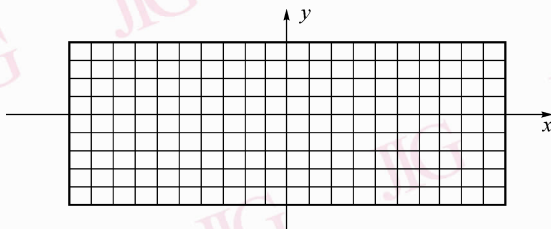


图 2 平面海面噪声网格

Fig. 2 The height noise grid of plane ocean surface

## 2.2 考虑球面影响的海洋表面模型

由于地球是球形, 若要构建真正的地球海洋表面模型, 就必须在 3 维球面上进行建模, 这样求解海洋表面各点  $(x, y, z)$  的计算量就是  $3^N$  ( $N$  为点的数目), 再加上海洋是动态的, 其海面高度时刻随时间变化, 必须考虑时间因素  $t$ , 因此生成动态海洋表面各点  $(x, y, z)$  的计算量就是  $4^N$ 。为了减小计算量, 就必须降维, 由于地球各处的半径是已知的, 即可从

海洋数据库中查到海洋表面各点的高度, 因此, 可以地球的经纬度为自变量, 采用 2.1 节的方法来建立海洋表面模型(式(3)), 以便将海洋表面模型降到 2 维。

$$H_{\text{heightfield}}(\alpha, \beta) = H_{\text{surface}}(\alpha, \beta) + h(\alpha, \beta) \cdot N_{\text{surface}} \quad (3)$$

式中,  $\alpha$  表示地球经度值,  $-\pi \leq \alpha \leq \pi$ ;

$\beta$  表示地球纬度值,  $-\pi/2 \leq \beta \leq \pi/2$ ;

$H_{\text{heightfield}}(\alpha, \beta)$  为球面上点  $(\alpha, \beta)$  的海面高度值;

$H_{\text{surface}}(\alpha, \beta)$  为球面上点  $(\alpha, \beta)$  的静止高度;

$h(\alpha, \beta)$  为球面上点  $(\alpha, \beta)$  的噪声高度值;

$N_{\text{surface}}$  为点  $(\alpha, \beta)$  的切平面的法向量。

如果以经纬度为自变量建模, 则在实现“降维”的同时, 又带来了新的问题, 即生成的球面海洋在高纬度存在明显的“压缩”现象(如图 3(a)所示), 即随着纬度增加, 相同距离上的海浪波数增加, 出现波浪“压缩”现象, 如南北极点处, 一个点处有无数个点的坐标值; 同时在南北极附近, 相同距离内有数倍于赤道处的波数, 这显然与实际情况不符, 因此就得解决“压缩”现象。

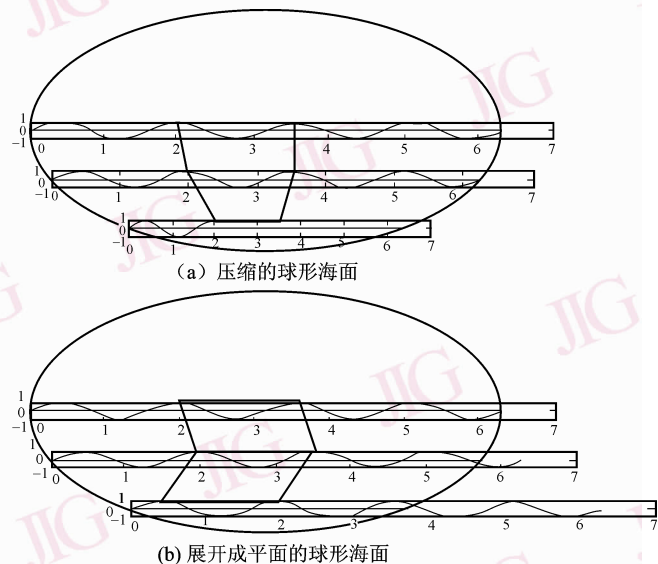


图 3 球形海面

Fig. 3 Spherical ocean surface

解决“压缩”现象就是使不同纬度的相同距离内具有相同的波数(此处指同一波长波)。处于纬度  $\alpha$  和纬度  $\beta$  相同经度内的弧长比为  $\cos \alpha / \cos \beta$ ; 而处于赤道和纬度  $\beta$  相同经度内的弧长比为  $1 / \cos \beta$ 。根据这一规律, 本文在球面海浪模型中通过加入缩

放因子  $\cos \beta$ , 将经度自变量按其所处纬度进行缩放(见式(4)), 使之与球面对应的弧长相一致, 这就解决了“压缩”现象(如图3(b)所示), 构建的球面海洋表面噪声模型如下:

$$h_{\text{noise}}(\alpha, \beta) = h(\alpha \cdot \cos \beta, \beta) \quad (4)$$

式中,  $h(\alpha \cdot \cos \beta, \beta)$  表示经纬度  $(\alpha, \beta)$  处的海面噪声高度值。

根据式(4)可求得点  $(\alpha, \beta)$  处的噪声高度值, 实际上该点的高度值为

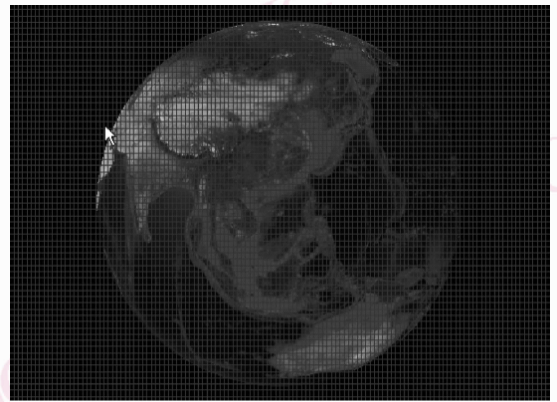
$$H_{\text{heightfield}}(\alpha, \beta) = H_{\text{surface}}(\alpha, \beta) + h(\alpha \cdot \cos \beta, \beta) \cdot N_{\text{surface}} \quad (5)$$

利用式(5)即可生成球面海面网格。如在赤道,  $\beta = 0$ , 点  $(\alpha, 0)$  就是一系列点, 利用  $h(\alpha \cdot \cos \beta, \beta)$  即可得到该点的噪声高度值  $h(\alpha, 0)$ ; 在南北极点,  $\beta = \pm \pi/2$ , 点  $(\alpha \cdot \cos \beta, \beta)$  就是一个点  $(0, \beta)$ , 利用  $h(\alpha \cdot \cos \beta, \beta)$  得到的该点的噪声高度值为  $h(0, \beta)$ ; 而对于其他位置, 即  $-\pi/2 < \beta < 0$  或  $0 < \beta < \pi/2$ , 则利用  $h(\alpha \cdot \cos \beta, \beta)$  即可得到长度为  $\alpha \cdot \cos \beta$  的弧上各点的高度值。利用式(5)可得到点  $(\alpha, \beta)$  的高度值  $H_{\text{heightfield}}$ , 实际上就是该点到球心的距离。高度值不断变化, 即距离不断变化, 这样就生成了动态的球面海浪。

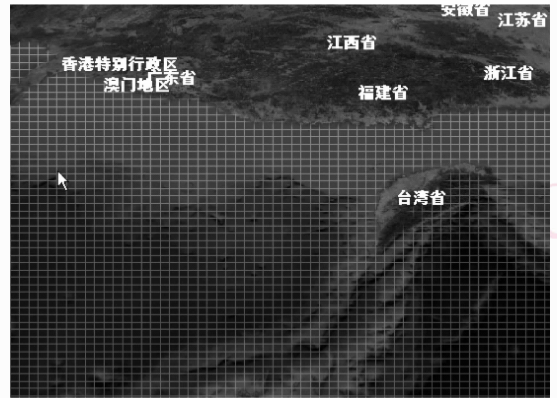
### 3 屏幕细分自适应网格算法

先前的海面模拟多采用规则网格, 没有考虑 LOD(levels of detail)。一些研究虽考虑了 LOD(如文献[16]、[31]所示), 但不是过于简单, 就是不能直接在数字地球上应用。为了实现对数字地球上的任意可视区域内的海面进行模拟, 本文提出了一种视点相关的球面动态多分辨率网格自适应策略, 即不是模拟一个规则、预定义的海面区域, 而是将网格采样对应于每一个球面元素的投影面积。其基本思想是在静止状态时, 球面海洋的每一个网格都覆盖与屏幕上同样大小的区域<sup>[15, 31]</sup>(如图4所示), 若将屏幕上显示的海面区域均匀细分成四边形的格子, 并将每个格子后向透视投影到静止海面, 则可在球形海面上获得对应的连续多分辨率网格。

其实现方法为, 首先在透视体内构建与视线方向垂直的透视平面, 并将该平面等分成小网格; 然后将视点与网格点连线, 其与球面相交得到的第1个交点即为球面的可视交点, 而由这些可视交点所组成的网格, 即为可视区域连续多分辨网格。该算法



(a)



(b)

图4 球面网格

Fig. 4 Sphere ocean surface grid

有以下6个特点:

- (1) 能对场景进行实时计算, 对于不在视野范围内的海面会被自动裁剪掉;
- (2) 对任意方向, 随着离视点距离的增大, 采样点间距增大;
- (3) 能实时生成 LOD 网格, 离视点近的海面则划分的网格小, 而离视点远的海面则划分的网格大;
- (4) 虽然海面网格瓦片离视点距离不同, 大小也不同, 但每个瓦片在屏幕上的投影面积相等;
- (5) 整个海浪表面被划分成很有限的瓦片, 以支持稳定帧速;
- (6) 由于相机运动导致网格在海面上的连续变化, 因此计算生成图像的每个地方都具有足够的分辨率。

由于该算法划分上符合人眼的观察属性, 即使得近处的海面拥有较高的面积划分密度, 而远处的海面则越来越低, 从而使近处海面具有较高的几何真实, 而对远处海面则自动进行平滑处理和简化, 因

此,该算法是视点相关的海浪 LOD 模型生成算法。

### 3.1 坐标空间变换

如图 5 所示,建立屏幕坐标系  $sot$ 。对于屏幕  $ABCD$ ,其范围为  $s \in (-1.0, 1.0), t \in (-1.0, 1.0)$ 。

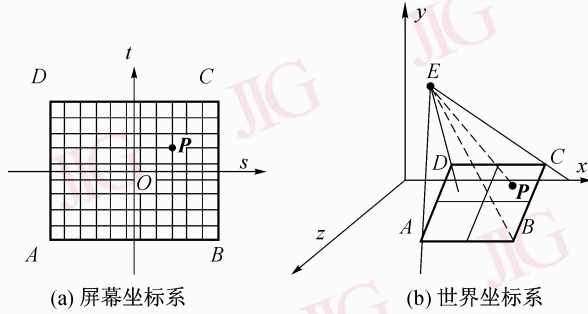


图 5 屏幕坐标系与世界坐标系

Fig. 5 Screen coordinate system & world coordinate system

本文中的矩阵和点的定义如下:

$M_{View}$  为视矩阵;

$M_{Perspective}$  为透视矩阵;

$M_{Proj} = M_{View} \cdot M_{Perspective}$  为相机投影矩阵;

$M_{InvProj} = [M_{Proj}]^{-1}$  为相机投影矩阵的逆;

$P_S = (s, t, z_s)$  为屏幕空间的任意一点,其中,  $-1 \leq z_s \leq 1$

$P_W = (x, y, z)$  为世界空间中任意一点。

屏幕空间任意一点  $P_S$  变换到世界坐标系下的点  $P_W$  的变换式为

$$P_W = M_{InvProj} \cdot P_S \quad (6)$$

式中,  $-1 \leq z_s \leq 1$ 。

世界坐标系下的任意一点  $P_W$  变换到屏幕  $ABCD$  中间的点  $P_S$  的变换式为

$$P_S = M_{Proj} \cdot P_W \quad (7)$$

由于变换矩阵  $M_{InvProj}$  和  $M_{Proj}$  与任何采样点  $P_S$  无关,仅与视点  $E$ 、参考点  $R$  和向上方向  $N$  有关,因此,只有当视点、视向或参考点改变时,才需要重新计算一次变换矩阵  $M_{InvProj}$  和  $M_{Proj}$ 。

### 3.2 球面自适应多分辨率网格生成

算法思想为:如图 5(a)所示,首先将屏幕均匀分割成正方形网格,然后将每一个网格点后向透视到世界空间,并求其与球面的近交点(离视点距离较近的交点),由所有的近交点即可生成球面多分辨率网格。

屏幕上任意一点  $P$  在世界坐标下与基球面的交点坐标为  $(x_p, y_p, z_p)$ 。对于屏幕上任意一点  $P_S$ ,

用式(8)、式(9)可求得以下世界坐标系下视点  $E$  和远点  $C$  的坐标,即

$$E_W = M_{InvProj} \times P_S(s, t, -1) \quad (8)$$

$$C_W = M_{InvProj} \times P_S(s, t, +1) \quad (9)$$

根据点  $E$  和  $C$  的坐标即可求得线段  $\overrightarrow{EC}$  的直线方程,即

$$\frac{x - x_C}{x_E - x_C} = \frac{y - y_C}{y_E - y_C} = \frac{z - z_C}{z_E - z_C} \quad (10)$$

再求线段  $\overrightarrow{EC}$  与球面的交点  $P$  的坐标,即

$$\begin{cases} \frac{x - x_C}{x_E - x_C} = \frac{y - y_C}{y_E - y_C} = \frac{z - z_C}{z_E - z_C} \\ x^2 + y^2 + z^2 = R^2 \end{cases} \quad (11)$$

由式(11)即可得到两个坐标值,其中距离视点较近的点便是点  $P$ 。

先将点  $P$  世界坐标转化成经纬度表示形式  $(\alpha_p, \beta_p, R_p)$ ,再根据式(5)即可求得  $t$  时刻点  $P$  的位置  $\hat{P} = (\alpha_{\hat{p}}, \beta_{\hat{p}}, R_{\hat{p}})$ 。由点  $\hat{P}$  组成的网格便为动态的海面网格。

这些网格虽然在屏幕上占有相同的像素,但在球面上,根据离视点的远近,这些网格的大小不同,即离视点越近,网格越小;反之,越大,可见这些网格是多分辨率的。

由于海面存在海浪实时运动,致使实际的海面可能高于或低于基球面,因此静止时,投影点位于屏幕边界处的点,动态时就可能超出或位于屏幕之内,从而造成由点  $\hat{P}$  组成的动态网格在屏幕的边界处就存在“裂缝”,即实际的海面网格点在屏幕上显示时,有部分超出屏幕,有部分缩进屏幕内,屏幕边缘部分未被全部覆盖(如图 6(b)所示)(图中虚线为屏幕边缘),而由点  $P$  组成的静态网格在屏幕的边界处就不存在“裂缝”(如图 6(a)所示)。

假设海浪的高度值位于  $(-h, h)$  之间,图 7 是球面的一个切面,图中 3 个圆(本文称之为外圆、基

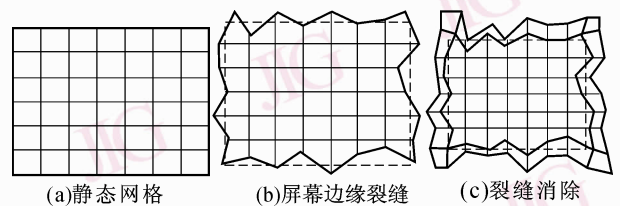


图 6 屏幕裂缝

Fig. 6 Screen slit

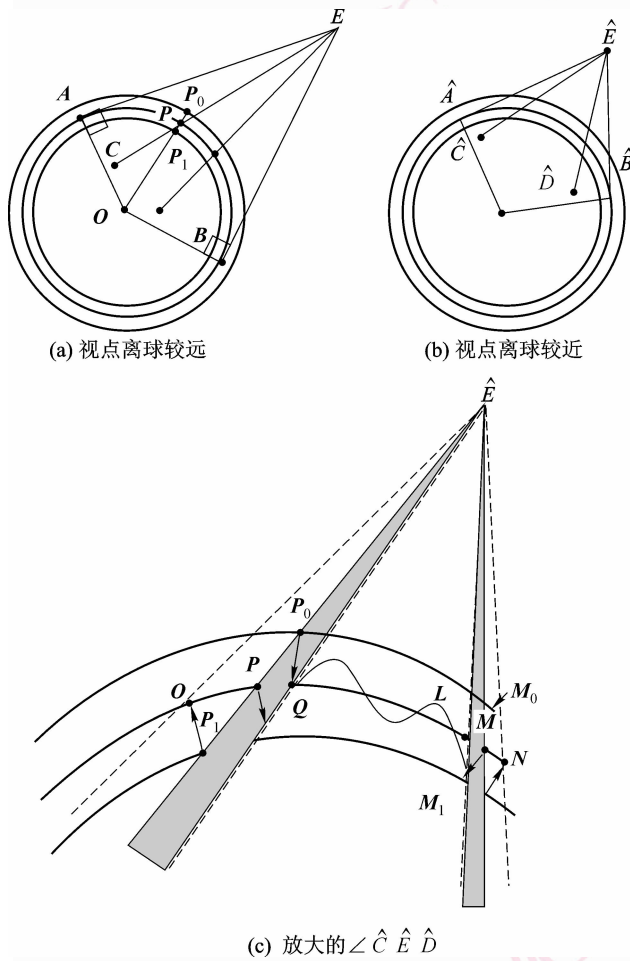


图7 过视点的地球切片

Fig. 7 Earth slice through viewpoint

圆和内圆,其对应的3个球称为“夹球”)半径差为  $h$ ,那么海面位于内外圆之间。屏幕“裂缝”现象产生的详细内在原因如下:(1)当视点离球较远时,此时视场范围就是切点与视点间的半圆部分,即  $\angle AEB$  内的部分(如图7(a)所示);(2)随着视点拉近,就看不到整个切点间的部分,只能看到切点间的一部分,即  $\angle \hat{C}\hat{E}\hat{D}$  内的部分(如图7(b)所示)。现在将  $\angle \hat{C}\hat{E}\hat{D}$  内的部分进行放大,其效果如图7(c)所示,  $\angle P\hat{E}M$  内部分可视。边界点为  $P、M$ , 直线  $\hat{E}P$  与外圆、基圆和内圆分别交于点  $P_0、P$  和  $P_1$ , 它们在外圆、基圆和内圆上的投影分别为点  $Q、P、O$ ; 直线  $EM$  与外圆、基圆和内圆分别交于点  $M_0、M$  和  $M_1$ , 它们在外圆、基圆和内圆上的投影分别为点  $L、M、N$ ; 圆弧  $\hat{P}M$  上的点所对应的海面,由于海面的动态性,其可能位于圆弧  $\hat{P}M$  上或下方,如某一时刻动态的点  $P$  或点  $M$  位于圆弧

$\hat{P}M$  的下方,便会在屏幕边缘产生“裂缝”(如图7(c)中的灰色区域)。

消除“裂缝”的思想为自适应扩大生成动态海面的静态海面范围。消除“裂缝”的具体执行方法为,通过世界坐标变换和投影变换先计算得到点  $Q、P、O、L、M、N$  在屏幕上的坐标,再找出屏幕坐标的极大、极小值;同理,先求出其他点坐标的极大、极小值,再根据点坐标的极大、极小值得到屏幕坐标的最大、最小值,以点  $A$  (坐标最小值、最小值)、点  $B$  (坐标最大值、最小值)、点  $C$  (坐标最大值、最大值)、点  $D$  (坐标最小值、最大值)4个点为顶点构建一个屏幕矩形,将屏幕矩形等分;然后计算与屏幕网格点对应的基球面点,再根据求得的基球面点计算海面高度值;最后生成动态海面,此时屏幕边缘始终处于生成的网格在屏幕上的投影之内(如图7(c)所示),即可消除因海面动态变化和屏幕采样范围不适引起的“裂缝”。

### 4 实验结果

基于数字地球的海浪模拟系统分成海浪生成、网格生成、陆海判断、光学模拟、海面绘制5个部分(图8)。进行海浪模拟时,首先利用海浪生成模块构造一个海面高度采样场;然后网格生成模块采用屏幕细分算法生成数字地球表面可视区域内的连续多分辨率网格;最后利用海陆判断模块对生成的网格点进行海陆判断,如果是陆地,则不进行计算;如果是海洋,则利用球面海洋模型对海面高度

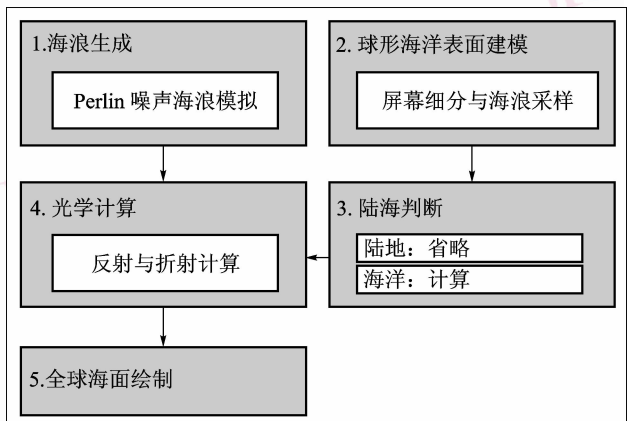


图8 基于数字地球的海浪模拟系统框架

Fig. 8 The framework of ocean wave simulation system based on digital earth

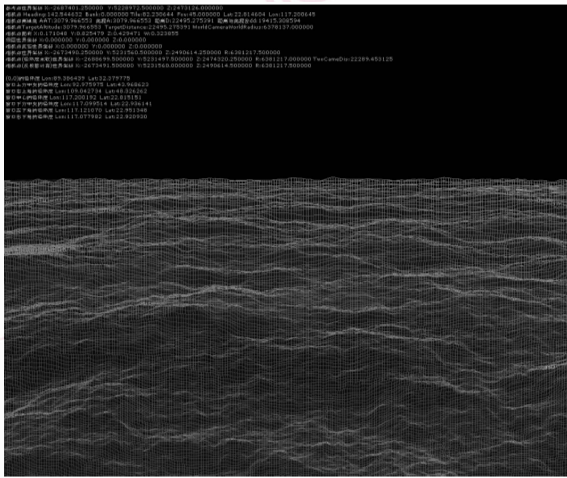
场进行采样,先得到该点的动态浪高值,然后利用光学模拟模块计算各海洋点的颜色,最后,由绘制模块实时对可视区域海洋表面进行绘制。各模块的计算,都在 GPU 上进行,以提高计算的效率和绘制的实时性。

根据本文给出的海浪建模方法和绘制算法,在 Pentium4 CPU 3.00 GHz、512 M 内存、RADEON X600 图形加速卡、WinXP 系统、Visual C++ 2005 和 Direct3D 环境下,以  $512 \times 512$  的网格进行采样,绘制帧率大约 10 fps。若以  $64 \times 64$  的网格采样,则可进行实时绘制,帧率大约 45 fps。

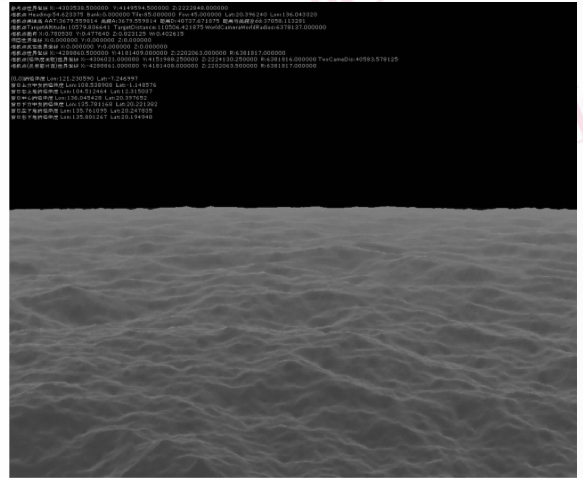
本文改进了文献[31]中的方法,生成了球形海面网格,其能在数字地球平台上进行海洋的模

拟。另外,本文方法比文献[16]的方法更实用,能生成无界多分辨率海洋场景,不仅能对局部海洋区域进行模拟和导航,而且能扩展到全球海洋区域。

图 9 是利用屏幕分割法生成的球形网格及海面效果,从图中可以看出,球形网格点在屏幕上是均匀显示的,此方法既避免了因视点变化而引起的计算量改变,也解决了屏幕边缘“裂缝”问题,同时能进行生成场景的多分辨率显示,可实现实时模拟;图 10 是利用本文方法在数字地球上生成的海陆一体化场景,效果很好,这说明本文的算法具有较强的实用性。



(a) 海面网格



(b) 海洋表面效果

图 9 基于数字地球的海面网格和效果

Fig. 9 Ocean surface grid and effect based on digital earth

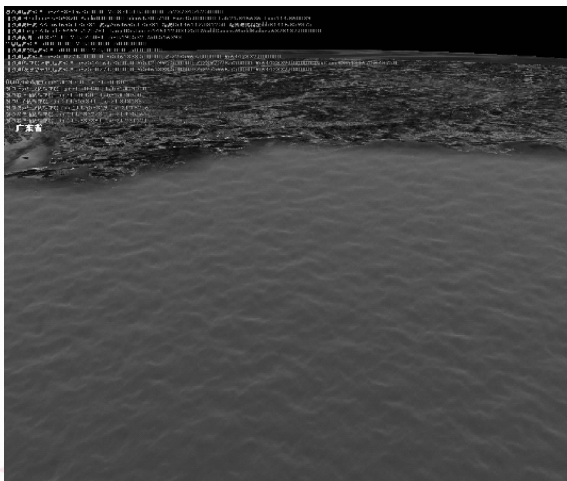


图 10 海陆一体化场景

Fig. 10 The integrated scene of ocean and land

## 5 结论

本文提出了一种考虑球面曲率的海浪模型,同时根据解析几何、海洋学和计算机图形学理论,推导出经度、纬度与海面高度场的定量表达式;定性说明了因经纬度变化而引起的高纬度“压缩”现象,并采用一种新的屏幕细分自适应算法计算和绘制出了不同视点、视向的多分辨率的真实感球面海浪场景,再现了数字地球上局部及全局的海浪效果,这是以往海浪模型和绘制方法所无法模拟的效果。与以往的工作相比,本文的工作有如下特点:

(1) 侧重点不同,其他文献的方法多是研究平面海浪建模与绘制,而本文则旨在研究球面海浪建

模与绘制,同时深入讨论了球面海浪建模与绘制中存在的问题,还分析了其原因,并提出了具体的解决方法和措施,绘制出了具有较强真实感的球面海浪效果;

(2) 本模型能绘制其他方法所不能绘制的效果,如不同边界条件下海陆一体化场景的绘制等;

(3) 本模型由于考虑了经纬度对海面建模的影响,从而避免了高纬度出现“压缩”现象;

(4) 本文提出了一种屏幕细分自适应网格生成算法,能绘制数字地球上的无界海洋,并能实现交互漫游;

(5) 本文的方法将计算限制在海洋的可视区域,使网格密度连续变化,其计算生成的海洋图像处处具有较高的分辨率,而且通过调整网格密度和模拟使用的噪声数,还可以灵活地控制质量代价比;

(6) 无论视点运动与否,本文方法的计算量相对固定,且绘制效率保持恒定,这对于模拟训练和科学研究都具有一定的借鉴意义。

下一步工作包括:(1)简化计算并采用并行计算等方法来提高绘制效率;(2)考虑天空场景的折射和反射,建立起陆、海、天一体化模型,用以模拟包括天空、陆地和海洋在内的自然环境。

### 参考文献 (References)

- Iglesias A. Computer graphics for water modeling and rendering: a survey[J]. *Future Generation Computer Systems*, 2004, **20**(8): 1355-1374.
- Su Fen-zhen, Du Yun-yan, Pei Xiang-bin, *et al.* Constructing digital sea of China with the datum of coastal line[J]. *Geo-Information Science*, 2006, **8**(1): 12-15, 20. [苏奋振, 杜云艳, 裴相斌等. 中国数字海洋构建基准与关键技术[J]. *地球信息科学*, 2006, **8**(1): 12-15, 20.]
- Blinn J F. Simulation of wrinkled surfaces[J]. *Computer and Graphics*, 1978, **12**(3): 286-292.
- Fishman B, Schachter B. Computer display of height fields[J]. *Computer and Graphics*, 1980, **5**(2-4): 53-60.
- Max N L. Vectorized procedural models for natural terrain: waves and islands in the sunset[J]. *Computer and Graphics*, 1981, **15**(3): 317-324.
- Gerstner F J. Theorie der wellen[J]. *Annalen der Physik*, 1809, **32**(8): 412-445.
- Rankine W J W. On the exact form of waves near the surfaces of deep water[J]. *Philosophical Transactions of the Royal Society, Series A*, 1863, **153**(4): 127-138.
- Fournier A, Reeves W T. A simple model of ocean waves[J]. *Computer and Graphics*, 1986, **20**(4): 75-84.
- Chen J X, Lobo N V. Toward interactive-rate simulation of fluids with moving obstacles using Navier-Stokes equations [J]. *Graphical Models and Image Process*, 1995, **57**(2): 107-116.
- Foster N, Metaxas D. Modeling water for computer animation [J]. *Communications of the ACM*, 2000, **43**(7): 60-67.
- Foster N. Realistic animation of liquids [J]. *Graphical Models and Image Processing*, 1996, **58**(5): 471-483.
- Premoze S, Ashikhmin M. Rendering natural waters [J]. *Computer Graphics Forum*, 2001, **20**(4): 189-200.
- Jensen L S, Goliás R. Deep-Water Animation and Rendering [EB/OL]. [http://www.gamasutra.com/gdce/Jensen/Jensen\\_01.html](http://www.gamasutra.com/gdce/Jensen/Jensen_01.html), 2001.
- Jerry Tessendorf. Simulating Ocean Water [EB/OL]. <http://home.gte.net/tssndrf/index.html>, 2001.
- Claes Johanson. Real-time Water Rendering [D]. Lund, Sweden: Lund University, 2004.
- Yuri Kryachko. Using Vertex Texture Displacement for Realistic Water rendering [EB/OL]. [http://developer.nvidia.com/object/gpu\\_gems\\_2\\_home.html](http://developer.nvidia.com/object/gpu_gems_2_home.html) Chapter 18, 2005.
- Mitchell Jason L. Real-Time Synthesis and Rendering of Ocean Water [EB/OL]. [http://ati.amd.com/developer/Mitchell-Real-Time\\_Synthesis\\_and\\_Rendering\\_of\\_Ocean\\_Water\(ATITR\\_Apr05\).pdf](http://ati.amd.com/developer/Mitchell-Real-Time_Synthesis_and_Rendering_of_Ocean_Water(ATITR_Apr05).pdf), 2005.
- Chiu Y F, Chang C F. GPU-based ocean rendering [A]. In: *Proceedings of IEEE International Conference on Multimedia and Expo [C]*, Toronto, Canada, 2006: 2125-2128.
- Yang Xu-dong, Pi Xue-xian, Zeng Liang, *et al.* GPU-based real-time simulation and rendering of unbounded ocean surface [A]. In: *Proceedings of the 9th International Conference on Computer Aided Design and Computer Graphics (CAD/CG 2005) [C]*, HongKong, 2005.
- Galin E, Chiba N. Realistic water volumes in real-time [A]. In: *Proceedings of Eurographics Workshop on Natural Phenomena [C]*, Vienna, Austria, 2006: 1-8.
- Xu Ying qing, Su Cheng, Li Hua, *et al.* Physics model based simulation of flowing water and wave [J]. *Chinese Journal of Computers (in Chinese)*, 1996, **19**(Supplement): 152-160. [徐迎庆, 苏成, 李华等. 基于物理模型的流水及波浪模拟[J]. *计算机学报*, 1996, **19**(增刊): 153-160.]
- Li Guang-xin, Ding Zhen-guo, Zhan Hai-sheng, *et al.* A modeling algorithm for realistic water surface simulation in virtual environment [J]. *Journal of Computer Research and Development*, 2004, **41**(9): 1580-1585. [李广鑫, 丁振国, 詹海生等. 一种面向虚拟环境的真实感水波面建模算法[J]. *计算机研究与发展*, 2004, **41**(9): 1580-1585.]
- Wang Lei, Ding You-dong, Zhan Jun. A method of rendering water wave based on perlin noise [J]. *Journal of Computer Research and Development*, 2005, **42**(Supplement): 207-210. [王磊, 丁友东, 张珺. 基于Perlin噪声绘制水面的方法[J]. *计算机研究与发展*, 2005, **42**(增刊): 207-210.]
- Wang Chang-bo, Wang Zhang-ye, Jin Jian-qiu. Real-time simulation of

- ocean wave based on cellular automata[A]. In: Proceedings of CAD/Graphics'2003[C], Macao, China, 2003:320-326.
- 25 Yin Y, Jin Y C, Ren H X. Wave simulation of visual system in marine simulator based on wave spectrums[A]. In: International Conference on Marine Simulation and Ship Maneuverability (Marsim'03) [C], Kanazawa, Japan, 2003.
- 26 Yan Lai-bin. Research and Implementation of Modeling and Real-time Rendering of Virtual Scene of Distribute Naval Battlefield[D]. Changsha: Graduate School of National University of Defense Technology, 2000. [ 鄢来斌. 虚拟海战场景建模与实时绘制技术的研究与实现[D]. 长沙:国防科技大学研究生院, 2000. ]
- 27 Xie Cui, Jin Yi-cheng, Li Zhi-hua, *et al.* Improved real-time ocean wave mesh model[J]. Journal of System Simulation, 2004, **16**(10): 2146-2148. [ 解翠, 金一丞, 李志华等. 改进的实时海浪网格模型[J]. 系统仿真学报, 2004, **16**(10): 2146-2148. ]
- 28 Pi Xue-xian. Modeling and Rendering of Large Scale Nature Scene [D]. Changsha: Graduate School of National University of Defense Technology, 2006. [ 皮学贤. 大规模自然场景建模与绘制[D]. 长沙:国防科技大学研究生院, 2006. ]
- 29 Arete Image Software[EB/OL]. <http://www.aretis.com>, 1996.
- 30 Perlin K. An image synthesizer [A]. In: Proceedings of ACM SIGGRAPH 85[C], San Francisco, CA, USA, 1985: 287-296.
- 31 Hinsinger D, Neyret F, Cani M P. Interactive animation of ocean waves [A]. In: Proceedings of ACM SIGGRAPH Symposium on Computer Animation [C], San Antonio, Texas, USA, 2002: 161-166.