

doi: 10.7690/bgzd.2015.07.009

空海一体虚拟战场环境构建技术

侯学隆, 张海峰, 罗木生

(海军航空工程学院指挥系, 山东 烟台 264001)

摘要: 针对海军空海一体虚拟战场环境仿真平台演练需求, 构建了3种仿真计算模型, 即天空场景仿真模型、海浪场景仿真模型和海面光照模型。利用通用天空标准亮度分布模型, 进行亮度计算、颜色空间转换、颜色校正和顶点颜色映射, 实现不同经纬度地域、不同时间、不同天气状况下天空动态场景的实时仿真; 通过对方向谱进行离散采样, 提取单元波振幅、频率、传播方向等参数, 构造FFT计算模式快速合成大量单元波实时生成海浪; 建立海面光照模型, 应用GPU着色技术, 将动态天空的变化在海浪场景中得以体现, 表达了环境之间的依存性与随动性。仿真效果表明: 该技术途径可逼真模拟空海一体战场环境, 在工程上具有较强的实用价值。

关键词: 天空图像; 海浪仿真; 虚拟环境; 视景仿真

中图分类号: TP391.9 **文献标志码:** A

Building Technology on Virtual Synthetic Air-sea Battlefield Environment

Hou Xuelong, Zhang Haifeng, Luo Musheng

(Command Department, Naval Aeronautic & Astronautical University, Yantai 264001, China)

Abstract: According to counter simulation exercise requirements for naval multiple weapon platforms, sky scene simulation model, ocean wave simulation model and ocean surface optical model were built. First, a method for simulating dynamic real-time sky background image varied with different region, different time and different weather conditions was discussed. After sphere sky geometry model was built, any arbitrary sky element luminance could be computed based on CIE general sky standard luminance. Then, the sky background dynamic image could be generated by means of scaling luminance, converting color space, adjusting color gamma factor and mapping each vertex in sphere sky. Second, ocean waves could be simulated effectively by means of getting amplitude, frequency and direction of component wave based on ocean wave spectrum sampling. FFT algorithm was used in summing many component waves. Finally, dynamic sky scene could be shown well in ocean wave scene by constructing ocean surface optical model and using GPU painting technology. The result of simulation shows that the approach could simulate realistic synthetic air-sea battlefield environment, and possess very strong practical value in engineering.

Keywords: sky image; ocean wave simulation; virtual environment; scene simulation

0 引言

基于虚拟现实技术的虚拟环境生成技术应用于军事仿真演练和军事决策以后, 引起了军事演练观念和方式的变革, 推动了军事仿真演练的发展, 为构建虚拟演练环境, 建立作战实验室提供了一种相对传统方法更为有效的高解析仿真方法, 从而为解决军事训练、演练和军事决策提供了新的途径^[1]。西方军事大国历来重视虚拟战场环境的研发。例如, 在美国空军2005年度的“红旗”军演中, E-8飞机装有“Nellis Air Combat Training System”系统, 该系统采用真实数据逼真模拟了演习区域的大面积地形地貌, 6个国家128架战机在真实的空间进行机动, 飞行数据被传感器实时采集并在这个虚拟空间中实时再现, 同时进行着武器的虚拟攻击。该系统

构建的虚拟环境, 为飞机间的逼真对抗提供了良好的环境支撑, 成功保障了“红旗”对抗演练。

基于信息系统的体系作战已成为现代海上作战的基本模式, 其涉及多兵种、多兵力、多武器平台在多维空间交战, 开展海上作战仿真对抗演练必然要求构建一个能够满足多种武器平台“寄存”的虚拟环境, 必须将空间与海面场景无缝集成。因此, 笔者在解决空间、海面场景构建技术的基础上, 在开源图形引擎OSG上初步实现了空海一体大规模虚拟环境的综合集成。

1 天空场景仿真模型

天空光照计算一方面直接用于天空颜色生成, 另一方面为岸、海面仿真提供光源数据参与光照计算。文献[1-8]采用大气物理模型建立了大气散射和

收稿日期: 2015-03-16; 修回日期: 2015-04-15

基金项目: 总装军用仿真技术十二五预研资助项目(51304030205); 海军武器装备军内科研项目(2012-475)

作者简介: 侯学隆(1979—), 男, 江西人, 博士生, 讲师, 从事分布作战建模与仿真、分布式虚拟战场环境研究。

天空亮度模型，仿真效果逼真，但计算量大；文献[9-11]采用物理模型和测量数据实现了天空亮度仿真，由于需要实测大量测量数据，对实验条件要求高；文献[12-14]采用纹理贴图的方式模拟了较为逼真的天空景象，但难以反映不同地域和时辰下的动态效果。笔者利用 CIE2003 通用标准天空亮度分布模型，基于 Preeham^[15]的特征参数值，实现了天空彩色背景图像的动态实时仿真，并能较好反映不同经纬度地域、不同时间、不同天气状况对背景图像产生的影响。

1.1 天空几何模型

在航空器视景仿真中，通常只考虑离地球海拔高度为 35 km 以下的大气层。因此，天空几何模型通过以地球球面向外拓展 35 km 构建，在可视范围用半球表示。对于一个半球， p 是球面上一点， R 为球半径， γ 是指球面上的纬度， α 是 p 点按弧线在水平面的投影与正北的夹角。则 p 点可表示为：

$$f(R, \varphi, \theta) = \begin{cases} R \sin \gamma \cos(\pi/2 - \alpha) \\ R \sin \gamma \sin(\pi/2 - \alpha) \\ R \cos \gamma \end{cases} \quad (1)$$

对于一个半球， $\gamma \in [0, \pi/2]$ ， $\alpha \in [0, 2\pi]$ 。因此，可用以下参数表征的顶点网格来生成所需的球面：

$$\begin{cases} f(R, \gamma, \alpha) \\ f(R, \gamma + \Delta\gamma, \alpha) \\ f(R, \gamma, \alpha + \Delta\alpha) \\ f(R, \gamma + \Delta\gamma, \alpha + \Delta\alpha) \end{cases} \quad (2)$$

可以根据不同精度要求选择 $\Delta\gamma$ 和 $\Delta\alpha$ 对半球进行网格化以便构建三角形条带进行绘制。

1.2 天顶亮度及色度计算

太阳位置用 (z_s, α_s) 表示，如图 1 所示。其中 z_s (rad) 定义为太阳与天顶的夹角， $z_s \in [0, \pi/2]$ ； α_s (rad) 定义为太阳按弧线在水平面的投影与正北的夹角， $\alpha_s \in [0, 2\pi]$ 。太阳的位置与观察点的经纬度、日辰有关，可通过航海相关知识解算不同位置及时间下的太阳位置。

在给定太阳天顶角 z_s 和浑浊度 T 时，天顶的亮度 Y_{zenith} (K cd/m²) 可用下式计算，与太阳天顶角、浑浊度的关系见图 1。

$$Y_{zenith} = (4.0453T - 4.9710) \cdot \tan\left(\left(\frac{4}{9} - T/120\right)(\pi - 2z_s)\right) - 0.2155T + 2.4192 \quad (3)$$

天顶位置色度 (x_{zenith}, y_{zenith}) 与太阳位置及浑浊度有关，计算方式如下：

$$x_{zenith} = (T^2 T_1) \cdot M_{xzenith} \cdot \begin{pmatrix} z_s^3 \\ z_s^2 \\ z_s \\ 1 \end{pmatrix} \quad (4)$$

$$y_{zenith} = (T^2 T_1) \cdot M_{yzenith} \cdot \begin{pmatrix} z_s^3 \\ z_s^2 \\ z_s \\ 1 \end{pmatrix} \quad (5)$$

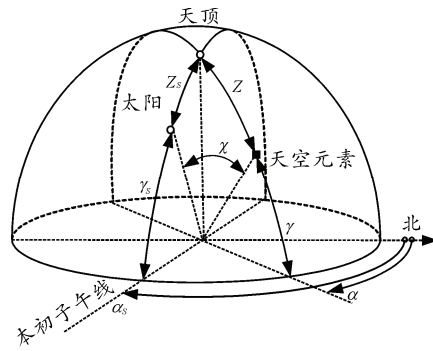


图 1 太阳位置及天空元素

1.3 天空任意点亮度及色度计算

在给定太阳位置的情况下，天空元素(即天空任意一个采样点)位置可用 (z, χ) 表征。其中 z (rad) 定义为天空元素与天顶的夹角， $z \in [0, \pi/2]$ ； χ (rad) 定义为天空元素与太阳间的角距离， $\chi \in [0, \pi]$ ，可通过下式计算：

$$\begin{cases} \chi = \arccos[\cos z_s \cdot \cos z + \sin z_s \cdot \sin z \cdot \cos \Delta\alpha] \\ z = \pi/2 - \gamma, \quad \Delta\alpha = |\alpha - \alpha_s| \end{cases} \quad (6)$$

式中： α (rad)、 α_s (rad) 分别为天空元素、太阳与正北之间的夹角。对于天空元素的亮度及色度坐标的计算，可从亮度分布特性入手。CIE2003 标准^[16]给出了任意天空元素的亮度 L_a 与天顶亮度 L_z 的比值，即亮度分布函数 $d(z, \chi, z_s)$ ：

$$d(z, \chi, z_s) = \frac{L_a}{L_z} = \frac{f(\chi) \cdot \varphi(z)}{f(z_s) \cdot \varphi(0)} \quad (7)$$

式中： φ 定义了亮度分级过渡函数，与天空元素的天顶角 z 大小有关； f 定义了散射指标特征函数，与天空元素与太阳间的角距离 χ 大小有关。 φ 、 f

的表达式为：

$$\begin{cases} \varphi(x) = 1 + A \cdot \exp(B / \cos x) \\ f(x) = 1 + C[\exp(D \cdot x) - \exp(D \cdot \frac{\pi}{2})] + E \cos^2 x \end{cases} \quad (8)$$

CIE2003 标准给出了 15 种不同天气下的特征参数 A、B、C、D、E 的具体值。为简化计算并与浑浊度 T 关联起来，可采用 Preetham 给出了亮度、色度对应的具体值^[15]。

综上所述：在固定 z_s 的情况下，任意天空元素的色度亮度 (Y_{sky})、色度 ($x_{\text{sky}}, y_{\text{sky}}$) 和可用下式计算：

$$\begin{cases} Y_{\text{sky}} = Y_{\text{zenith}} \cdot d_y(z, \chi, z_s) \\ x_{\text{sky}} = x_{\text{zenith}} \cdot d_x(z, \chi, z_s) \\ y_{\text{sky}} = y_{\text{zenith}} \cdot d_y(z, \chi, z_s) \end{cases} \quad (9)$$

由于 z_s 、 α_s 随时间变化，根据上式便可获得任意天空元素随时间变化的亮度和色度。

1.4 亮度压缩及 RGB 颜色生成

由于得出的亮度 Y_{sky} 值较大，需要将其缩小至 [0.0, 1.0] 区间，一个简单的方法是采用式 10 的指数法来实现。

$$Y_{\text{sky}} = 1 - \exp(-1.0 / K \times Y_{\text{sky}}) \quad (10)$$

式中 K 为调节因子，参考值范围 [20, 30]，典型值 25。

一旦计算出色度和亮度，就得到了 Y_{xy} 空间中的颜色。但计算机采用 RGB 颜色，因此必须实现颜色转换。具体来讲，分两步实施：首先将 Y_{xy} 转换到 CIE 中的 XYZ 三色值，然后由 CIE 中的 XYZ 三色值转换成 RGB 值。

$$\begin{cases} X = x_{\text{sky}} \cdot Y_{\text{sky}} / y_{\text{sky}} \\ Y = Y_{\text{sky}} \\ Z = (1 - x_{\text{sky}} - y_{\text{sky}}) \cdot Y_{\text{sky}} / y_{\text{sky}} \end{cases} \quad (11)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.53715 & -0.49853 \\ -0.969256 & 1.875991 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (12)$$

最后对 RGB 颜色进行一次 gamma 校正，如下式所示：

$$R = R^{1/\text{gamma}}, G = G^{1/\text{gamma}}, B = B^{1/\text{gamma}} \quad (13)$$

式中：gamma 因子取值范围 [2.3, 3.8]，典型值为 2.5。

2 海浪场景仿真模型

海浪场景是空海一体战场环境的主场景，其仿真效果直接关系到总体效果。目前，海浪仿真方法主要有：基于 Bump/Normal Mapping 的海浪仿真^[17]、基于物理模型的海浪仿真^[18-22]、基于构造波形函数的海浪仿真^[23-24]和基于波谱理论的海浪仿真^[25-31]。在空海一体虚拟战场中，对海浪仿真具有适应海况等级、逼真性、交互性、实时性、多分辨率和超大面积六大需求。基于海浪谱和快速傅里叶变换 (fast fourier transform, FFT) 方法可有效解决海浪仿真逼真度与实时性之间的矛盾。

2.1 海浪几何模型

将多个次摆线波叠加，并将其扩展到二维表面，可得海浪的波数表达式：

$$\begin{cases} x(x_0, t) = x_0 + \mu \cdot \sum_k \hat{k} A(k) \sin(k \cdot x_0 + \omega(k)t + \Phi(k)) \\ y(x_0, t) = y_0 - \sum_k A(k) \cos(k \cdot x_0 + \omega(k)t + \Phi(k)) \end{cases} \quad (14)$$

其中： μ 为波形控制因子； $\mathbf{x}=(x, z)$ 是以 $\mathbf{x}_0=(x_0, z_0)$ 为基准点随时间变化的水平坐标； y_0 为平均水面； $\mathbf{k}=(k_x, k_z)$ 为波数矢量，其幅值为 $k=||\mathbf{k}||$ ，单位矢量 $\hat{\mathbf{k}}=\mathbf{k}/k$ ， $\Phi(k)$ 为初始相位。

2.2 海浪的 FFT 表达式

假设在面积 $L_x \times L_z$ 的水平面建立 $N \times M (N=2i, M=2j; i, j$ 为正整数) 规则采样网格，共计 $N \times M$ 个采样点，对于深水浪，将上式变换如式 (15) 所示，推导过程参见文献 [32]。

$$\begin{cases} x = x_0(p - N/2) + \mu \cdot (-1)^{(p+q)} \cdot A \\ y = y_0 - (-1)^{(p+q)} \cdot B \\ z = z_0(q - M/2) + \mu \cdot (-1)^{(p+q)} \cdot A \\ A = R \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} i \cdot (-1)^{(n+m)} U_{kz} \cdot C \right) \\ B = R \left(\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (-1)^{(n+m)} \cdot C \right) \\ C = H(\mathbf{k}, t) \exp(i2\pi(\frac{np}{N} + \frac{mq}{M})) \\ H(\mathbf{k}, t) = \frac{1}{\sqrt{2}} (\lambda_r + i\lambda_i) A(\mathbf{k}) \exp(i\omega(k)t) \\ \exp(i\omega(k)t) = \cos(\sqrt{gk} \cdot t) + i \cdot \sin(\sqrt{gk} \cdot t) \end{cases} \quad (15)$$

式中： R 表示取复数的实部，

$\mathbf{k} = (k_x, k_z) = 2\pi(p - N/2)/L_x, 2\pi(q - M/2)/L_z$, $n, p \in [0, N-1]$, $m, q \in [0, M-1]$ 。对于上式, 便可分别使用 FFT 算法一次性将各采样点的高度值 y 和修正后的水平坐标 x, z 求出。

2.3 方向谱离散采样

由式 (15) 可知, $H(\mathbf{k}, t)$ 由振幅 $A(\mathbf{k})$ 、初相位 $\exp(i\varphi)$ 及相位 $\exp(i\omega(\mathbf{k})t)$ 3 个部分构成。对于振幅 $A(\mathbf{k})$, 一种简单的方法是在一个区间内随机产生。但随机产生会使仿真海浪的能量分布与真实海区的观测统计有较大差距, 并且仿真效果不能调节。获取 $A(\mathbf{k})$ 最好的方法是从海浪波数谱中提取, 文献[25]提出一种方法: 对 Phillips 波谱采样, 将波谱值的平方根直接作为振幅值。当采样数或风速值变化时, 该方法容易产生波高突变, 仿真效果难以控制。本文根据波谱理论采用积分法求取振幅, 进而推广到对任意波谱的采样。

给定波数谱 $E_k(\mathbf{k})$, 对于每一个采样区间有(当 $\Delta k_x, \Delta k_z$ 足够小时, 可近似计算):

$$\begin{cases} A(\mathbf{k}) = \sqrt{2 \int_{\Delta k_x} \int_{\Delta k_z} E_k(\mathbf{k}) dk_x dk_z} \approx \sqrt{2E_k(\mathbf{k})\Delta k_x \Delta k_z} \\ \Delta k_x = 2\pi/L_x, \Delta k_z = 2\pi/L_z \end{cases} \quad (16)$$

波数谱通常可由方向谱转化而来, 给定方向谱 $E_{\omega, \theta}(\omega, \theta)$, 为保证转化过程能量保持一致可按式^[28]进行:

$$\begin{cases} E_k(\mathbf{k}) = E_{\omega, \theta}(\omega_k, \theta_k) \frac{1}{2k} \sqrt{\frac{g}{k}} \\ \omega_k = \sqrt{gk}, \cos(\theta_k) = k_x / \|\mathbf{k}\| \end{cases} \quad (17)$$

方向谱的表达式通常由频谱和方向扩展函数组合而成, 即

$$E_{\omega, \theta}(\omega, \theta) = E_{\omega}(\omega) \cdot D(\omega, \theta) \quad (18)$$

其中: $E_{\omega}(\omega)$ 为频谱; $D(\omega, \theta)$ 为方向扩展函数; θ 为单元波的传播方向。不失一般性, 选择 P-M 频谱^[33]及 ITTC 方向扩展函数^[34]构造方向谱, 将上述方向谱转化成波数谱可得

$$E_{\text{PM-ITTC}}(\mathbf{k}) = \frac{\alpha}{\pi k^6 U_{10}^2} \exp\left(-\frac{0.688g^2}{k^2 U_{10}^4}\right) |\mathbf{k} \cdot \mathbf{U}_{10}|^2, \mathbf{k} \cdot \mathbf{U}_{10} \geq 0 \quad (19)$$

其中: $k = \|\mathbf{k}\|$; $U_{10} = \|\mathbf{U}_{10}\|$, 为海平面 10 m 处风速。

3 空海一体环境下的海面光照模型

要渲染高度真实的海浪, 除构造合适的波面几

何造型外, 还要充分考虑海水的光学效应。海水是良好的反射体与折射体, 海水的最终颜色是反射与折射的 Fresnel 合成结果。

3.1 反射

海水的反射主要考虑全局反射、局部反射和太阳镜面高亮反射。全局反射由半球形天空背景颜色构成, 根据第一节模型生成天空 RGB 图像, 通过帧缓存对象 (frame buffer object, FBO) 离屏渲染动态生成立方体纹理。在全局反射中, 可采用反射矢量作为立方体纹理坐标查找纹素来实现。如图 2 所示, 反射矢量按下式计算:

$$\begin{cases} \mathbf{E} = (\mathbf{C} - \mathbf{P}) / \|\mathbf{C} - \mathbf{P}\| \\ \mathbf{R} = 2(\mathbf{N} \cdot \mathbf{E})\mathbf{N} - \mathbf{E} \end{cases} \quad (20)$$

则全局反射为

$$C_{\text{globalrefl}} = \text{textureCube}(\text{skyMap}, \mathbf{R}) \quad (21)$$

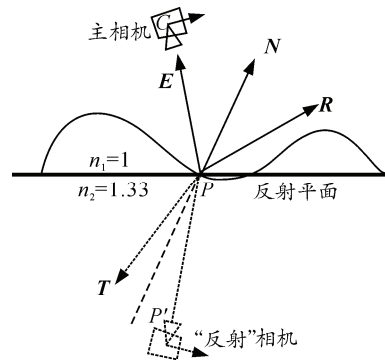


图 2 反射与折射示意图

全局反射实现了远处物体的静态环境映射, 并不能反射海面上的局部物体(如水面舰艇、小岛等), 因此动态局部反射应该根据需要实时绘制。其主要步骤如下:

1) 产生局部纹理对象 reflectMap, 配置 FBO、视口大小与当前相机观察面长宽比一致, 将反射纹理绑定到 FBO, 将 FBO 绑定到颜色缓冲区。

2) 设置“反射”相机, 如图 2 所示, “反射”相机与主相机关于反射平面对称。在 OpenGL 坐标系下, 设反射平面方程为 $y - h = 0$, 有 $\mathbf{M}_1, \mathbf{M}_2$:

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 2h \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{M}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (22)$$

如果主相机位置 (eye position) 为 C , 注视点 (Look-at point) 为 P_{ref} , 观察向上矢量 (up-vector) 为 \mathbf{U} , 则“反射”相机位置为 $\mathbf{M}_1 \cdot C$, 注视点为

$M_1 \cdot P_{\text{ref}}$, 观察向上矢量为 $M_2 \cdot U$ 。

3) 设置附加裁减面。由于反射纹理是反射面以上景物的镜像, 在渲染场景时, 存在3种情况: ① 物体完全位于反射面以下; ② 物体完全位于反射面以上; ③ 物体一部分在反射面以上, 其他部分在反射面以下。第1种情况不会产生倒影, 只涉及到折射问题, 在生成折射纹理需要渲染该物体。对于第2种情况, 只需渲染该物体即可。在渲染第3种类型的物体时, 如果反射面以下物体也加以渲染, 将获得错误的反射纹理, 因此在渲染该物体时要剔除反射面以下部分。附加裁剪面可用如下方程表示:

$$Mx + Ny + Jz + K = 0 \quad (23)$$

对于“反射”相机空间下的点 (x_e, y_e, z_e, w_e) , 满足如下条件将在 OpenGL 中绘制:

$$(M, N, J, K)M^{-1}(x_e, y_e, z_e, w_e)^T \geq 0 \quad (24)$$

其中 M 为当前模型视图矩阵。根据上述原理和反射平面方程, 采用 `glClipPlane` 设置附加裁剪面参数为 $(0, 1, 0, h)$, 便可将反射面以下物体从场景中剔除。

4) 绘制第2种和第3种类型的物体。由于纹理已和 FBO 绑定, 此时反射纹理已生成, 可直接使用。

5) 在适当时候将反射纹理 `reflectMap` 作为参数传递给片元着色器 (Fragment Shader) 以便进行纹理查询。反射纹理坐标采用投影映射方式获得, 如图2所示。设“反射”相机的模型视图矩阵为 M_m , 投影矩阵 M_p , 对于波面一点 $P(x, y, z, l)^T$, 在“反射”相机观察平面的投影点为 $P'(x, y, z, w)^T$ (齐次坐标), 有 $P' = M_p \cdot M_m \cdot P$, 则 P' 已经是归一化坐标, 即 $P' \cdot x / P' \cdot w$ 、 $P' \cdot y / P' \cdot w$ 、 $P' \cdot z / P' \cdot w$ 在 $[-1, 1]$ 。由于访问纹理矩形的坐标值用像素来表示, 设纹理矩形宽度为 tw , 高度为 th , 可得反射纹理坐标 $rtc(s, t)^T$ 。

$$rtc = \begin{bmatrix} 0.5tw & 0.5tw \\ 0.5th & 0.5th \end{bmatrix} \cdot \begin{bmatrix} P' \cdot x / P' \cdot w \\ P' \cdot y / P' \cdot w \end{bmatrix} \quad (25)$$

根据 rtc 查找反射纹理, 可得波面 P 点的反射图像。由于海浪上下起伏, 实际反射平面往往偏离预设反射平面 $y - h = 0$, 因此要对 rtc 进行扭曲, 以便获得“波光鳞鳞”的反射效果。具体方法是: 在原有值基础上, 加上与视点距离和波面法线相关的偏移量。

$$\begin{cases} rtc.s += RDS \cdot N_x / (P' \cdot z / P' \cdot w) \\ rtc.t += RDS \cdot N_y / (P' \cdot z / P' \cdot w) \end{cases} \quad (26)$$

其中: RDS 为反射扭曲控制因子; N_x 、 N_y 为法线 N

的 x 、 y 分量; 式中除以 z 坐标, 可以使扭曲随着视点距离增大而减少。由于反射纹理查询是在主相机的着色器下进行的, 所需的 M_m 与 M_p , 可在“反射”相机激活状态下获得并采用 `glUniform` 将该值传递给着色器。综上所述, 局部反射可由下式表达:

$$C_{\text{localrefl}} = \text{texture2D}(\text{reflectMap}, rtc.st) \quad (27)$$

最后, 对全局反射、局部反射与太阳光进行合成, 可得反射光为

$$C_{\text{reflection}} = C_{\text{globalrefl}} + C_{\text{localrefl}} + C_{\text{sunspecularlight}} \quad (28)$$

式中 $C_{\text{sunspecularlight}}$ 为海面太阳光反射的镜面高光, 可用 Phong 模型计算^[35]:

$$C_{\text{sunspecularlight}} = C_{\text{sun}} \cdot (R \cdot E)^{n_s \times \text{Depth}} \quad (29)$$

式中: C_{sun} 为太阳颜色, 可通过第一节模型计算得到; n_s 为镜面反射系数; Depth 为海面某采样点的深度值。

3.2 折射

折射的实现思路与反射基本相似。折射是在主相机状态下渲染折射纹理 `refractMap`, 然后在着色器中对纹理进行扭曲与映射操作。渲染折射纹理要注意以下几点: 1) 要设置合适的附加裁减面, 确保水上物体不被绘制; 2) 由于空气的折射率 (1) 小于水的折射率 (1.33), 导致水中物体看起来要比实际要短, 因此要进行扭曲操作; 3) 由于折射角会偏离入射角, 使得主相机“拍摄”的折射场景可能比实际要小, 这样会导致“纹理丢失”, 因此在渲染折射纹理要适当增加主相机的视场角。

折射纹理坐标的计算方法与反射类似, 与之不同的是: 一是使用的是修正后的主相机的模型视图矩阵和投影矩阵; 二是式 (25) 应该使用折射纹理的宽度与高度; 三是式 (26) 使用相应的折射扭曲控制因子。最后可用该坐标访问折射纹理产生折射效果:

$$C_{\text{localrefr}} = \text{texture2D}(\text{refractMap}, rtc.st) \quad (30)$$

对局部折射与水色进行混合, 可得最终的折射效果:

$$C_{\text{refraction}} = C_{\text{localrefr}} \cdot \alpha + (1 - \alpha) \cdot C_{\text{water}} \quad (31)$$

式中: α 为海水的透明度, C_{water} 为海水颜色。为了表达水下物体越深越不易看清效果, 借鉴 OpenGL 线性雾产生原理, 设计 α 随深度变小, 如下式^[36]所示:

$$\alpha = \exp(-\rho \cdot d) \quad (32)$$

式中： ρ 为衰减系数； d 为水下物体与海面采样点的距离。

3.3 颜色合成

最后，要对反射与折射进行合成。水的反射与折射对海水颜色的贡献比例遵循 Fresnel 定律，可由下式计算海水反射贡献因子：

$$\begin{cases} F(k) = \frac{(g-k)^2}{2(g+k)^2} \left[1 + \frac{(k(g+k)-1)^2}{(k(g-k)-1)^2} \right] \\ k = \mathbf{E} \cdot \mathbf{N}, \quad g = \sqrt{k^2 + 0.77} \end{cases} \quad (33)$$

由于上式复杂，GPU 计算比较费时，而 $k \in [0,1]$ ，可将其分成 512 份进行采样，获取的值存放在一维纹理 fresnelMap 中，在片元着色器中可以对其快速访问：

$$F(k) = \text{texture1D}(\text{fresnelMap}, k) \quad (34)$$

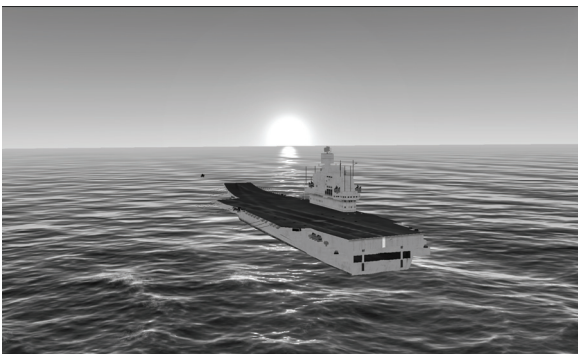
考虑有泡沫情况下，海水的最终颜色为：

$$\begin{cases} C_{\text{total}} = C_{\text{oceanwater}} \cdot s(h) + C_{\text{foam}} \cdot (1-s(h)) \\ C_{\text{oceanwater}} = C_{\text{reflection}} \cdot F(k) + C_{\text{refraction}} \cdot (1-F(k)) \\ s(h) = \text{Clamp}[(h-h_0)/(h_{\text{max}}-h_0)], s(h) \in [0,1] \end{cases} \quad (35)$$

式中： h_{max} 为海浪最大瞬时波高； h_0 为开始产生泡沫的海浪波高； h 为海浪当前瞬时波高。

4 仿真实验

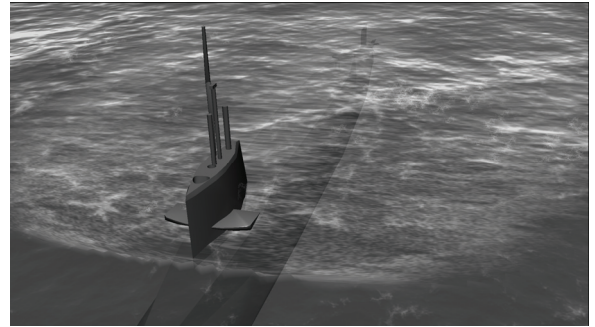
由上述分析，FFT 产生的海浪高度场具有平铺特性，可由“瓦块”拼接产生无边境效果。在当前主流 PC 机 (CPU 2.8G，内存 4GB，显存 2GB) 上，基于开源场景图形引擎 OSG，结合上述模型，采用 LOD 算法和 GPU 着色技术，实现了空海一体大规模动态逼真环境的实时仿真 (30FPS)，实验效果如图 3 所示。



(a) 日出效果



(b) 泡沫效果



(c) 海水透明度效果

图 3 仿真实效

实验证明：1) 天空动态图像不必每帧更新，在仿真时间与墙钟时间 1:1 情况下，1 min 更新一次太阳位置可保证图像平滑过渡；2) FFT 采用点为 64×64 时可实时 (30FPS 以上) 在线解算高度场，此时有 4 096 个单元波合成海浪，可同时兼顾海浪仿真的逼真度与实时性；3) 如果用 128×128 采样难以保证实时性，可离线生成系列高度场图，采用顶点纹理和法线图生成技术，基于时间系列进行插值，也可实现逼真的海浪动画，但经过一定周期后，波浪具有重复感。

5 结束语

笔者研究了一种空海一体化虚拟战场环境的仿真方法，初步实现空海一体战场环境的综合集成。以 CIE 光照模型为基础的天空背景图像生成技术可仿真不同地域、不同时间、不同能见度 (通过修改浑浊度 T 实现) 下的天空景象；以波谱理论和 FFT 算法为基础的海浪仿真方法可实现对多种波谱的采样与海浪合成，能够体现海区的浪高、浪频规律及海浪传播方向，可细致调节风速不会造成波高突变，较为细腻地反映风浪间的对应关系；以 GPU 着色技术为基础的海面光照模型将动态天空的变化在海浪场景中得以体现，表达了环境之间的依存性与随动性。下一步的研究工作是将岛岸、复杂气象效果 (云、大气散射光照) 及水下效果融入到空海

环境中, 满足海军空中、海面及水下作战平台及武器的仿真需求。

参考文献:

- [1] 赵沁平. DVENET 分布式虚拟环境[M]. 北京: 科学出版社, 2002: 1-15.
- [2] 杨晓东, 万旺根, 张开翼, 等. 多视角实时大气渲染算法研究[J]. 计算机工程与设计, 2010, 31(14): 3241-3243.
- [3] 张立民, 闫文君. 基于改进简化天空光照模型的天空绘制算法[J]. 计算机技术与发展, 2012, 22(2): 123-126.
- [4] 杜芳, 张炎. 基于GPU的地球大气散射现象可视化仿真[J]. 系统仿真学报, 2009, 21(2): 147-150.
- [5] 任程, 乔体洲, 戴树岭. 晴天天空与太阳的实时仿真[J]. 计算机辅助设计与图形学学报, 2009, 21(3): 305-310.
- [6] 黄建余. 白天天空背景亮度仿真研究[J]. 飞行器测控学报, 2008, 27(1): 61-64.
- [7] 雷励星, 范贤德, 徐振中. 面向飞行模拟的大气效果实时绘制技术[J]. 清华大学学报: 自然科学版, 2009, 40(7): 1003-1006.
- [8] 刘世光, 彭群生. 气象景观的真实感模拟技术综述[J]. 计算机辅助设计与图形学学报, 2008, 2(4): 409-414.
- [9] Eric Bruneton, Fabrice Neyret. Precomputed Atmospheric Scattering[C]. Eurographics Symposium on Rendering 2008 Steve Marschner and Michael Wimmer, 2008, 27(4): 1079-1086.
- [10] 景春元, 谭碧涛, 陈新. 白天全天空背景测量与分析[J]. 光学与光电技术, 2012, 10(3): 38-41.
- [11] 谭碧涛, 李艳娜, 尹飞. 白天全天空背景亮度特性研究[J]. 光电技术应用, 2010, 25(3): 1-2.
- [12] 谭碧涛, 李艳娜, 刘玮峰. 戈壁地区天空背景光谱研究[J]. 光电技术应用, 2010, 25(4): 1-5.
- [13] 王成, 戴树岭. 基于虚拟现实技术的地球太空环境仿真[J]. 系统仿真学报, 2010, 22(12): 2867-2871.
- [14] 陈业斌. 关于天空动态仿真技术研究及实现[J]. 计算机技术与发展, 2008, 18(6): 171-173.
- [15] Preetham A J, Shirley P, Smits B. A practical analytic model for daylight t[R]. Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, California, 1999: 91-100.
- [16] ISO 15469/CIE S011 (2003). Spatial distribution of daylight-CIE standard general sky[S/OL]. CIE Central Bureau Vienna, 2006.
- [17] 马骏, 朱衡军, 龚建华. 基于Cg和OpenGL的实时水面模拟[J]. 系统仿真学报, 2006(2): 395-400.
- [18] Kass M, Rapid M G. Stable fluid dynamics for computer graphics[J]. Computer Graphics, 1990, 24(4): 49-57.
- [19] Chen J X, Lobo N D V. Toward interactive-rate simulation of fluids with moving obstacles using Navier-Stokes equations [J]. Graphical Models and Image Processing, 1995, 57(2): 107-116.
- [20] 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.
- [21] Foster N, Metaxas D. Realistic animation of liquids[J]. Graphical Models and Image Processing, 1996, 58(5): 471-483.
- [22] Foster N, Fedkiw R. Practical animation of liquids[M]. SIGGRAPH 2001, 2001: 15-22.
- [23] Claes Johanson. Real-time water rendering[R]. Master of Science thesis Lund University 2004.
- [24] Peachey DR. Modeling waves and surf[J]. Computer Graphics (Siggraph'86), 1986, 20(4): 65-74.
- [25] Tessendorf J. Simulating Ocean Water[C]//Proc. of SIGGRAPH'99. Los Angeles, USA: ACM Press, 1999.
- [26] 邱宏安. 随机海浪模型的建立与分析[J]. 系统仿真学报, 2000, 12(3): 228-228.
- [27] 杨怀平, 孙家广. 基于海浪谱的波浪模拟[J]. 系统仿真学报, 2002, 14(9): 1175-1178.
- [28] Sebastien Thon, Djamchid Ghazanfarpour. Ocean waves synthesis and animation using real world information[J]. Computer Graphics, 2002, 26(4): 99-108.
- [29] 徐利明, 姜显明. 基于谱分析的实时波浪模拟[J]. 系统仿真学报, 2005, 17(9): 2092-2095.
- [30] Frechot J. Realistic Simulation of Ocean Surface Using Wave Spectra [Z](2008-11-07). <http://www.labri.fr/publications/is/2006/Fre06/>.
- [31] 侯学隆, 沈培志, 王文辉. 基于方向谱的海浪合成方法[J]. 系统仿真学报, 2010, 22(1): 130-134.
- [32] 侯学隆, 黄启来, 沈培志. 基于FFT的海浪实时仿真方法[J]. 计算机工程, 2009, 35(22): 258-258.
- [33] Pierson W J, Moskowitz L. A proposed spectral form for fully developed wind seas Based on the Similarity Theory of S. A. Kitaigorodskii[J]. Journal of geophysical research(S0148-0227), 1964, 69(24): 5181-5203.
- [34] International Towing Tank Conference(ITTC)[R]. Report of Seakeeping Committee, 1978(15th)and 1984(17th).
- [35] Chiu Yungfeng, Chang Chunfa. GPU-based Ocean Rendering[C]. ICME2006.
- [36] 蔡士杰, 宋继强, 蔡敏. 计算机图形学[M]. 3版. 北京: 电子工业出版社, 2005: 469.