

1-4-2019

## Research and Implementation of Antisubmarine Combat Visual Simulation System

Zhang Xin

*School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China;*

Gao Yan

*School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China;*

Zhang Jian

*School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China;*

Changbo Wang

*School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Research and Implementation of Antisubmarine Combat Visual Simulation System

## Abstract

**Abstract:** This paper constructs visual simulation framework of antisubmarine combat. The sea surface mesh was modeled using projected grid and FFT technologies. *Both the sea surface and undersea are finely rendered. The submarine fine rendering algorithm was implemented by considering refraction, Fresnel reflection, Godray and underwater fog effect and using Shader technology, so as to build a realistic battlefield environment. The method of interpolation processing for data in WGS84 coordinates of latitude and longitude through local area network was proposed; the displacement interpolation and rotation interpolation were implemented by calculating the ratio between the current timestamp and both before and after timestamp; the position and orientation interpolation processing was implemented by using the longitude and latitude data; and the control instructions were parsed and executed to control airplanes, submarines and other entities, so as to achieve antisubmarine combat visual simulation.*

## Keywords

undersea scene, ocean rendering, shader, battle simulation

## Recommended Citation

Zhang Xin, Gao Yan, Zhang Jian, Wang Changbo. Research and Implementation of Antisubmarine Combat Visual Simulation System[J]. Journal of System Simulation, 2018, 30(4): 1302-1309.

## 反潜对抗视景仿真系统研究与实现

张昕, 高岩\*, 张健, 王长波

(华东师范大学计算机科学与软件工程学院, 上海 200062)

**摘要:** 构建反潜作战视景仿真框架, 采用海水投影网格与 FFT 框架对海面网格进行建模, 同时利用 Shader 技术考虑折射、菲涅尔反射、Godray 及水下雾化效应, 实现了实时海底精细绘制, 构建逼真的战场环境。提出了一种通过局域网对全球(WGS84)经纬度坐标数据插值处理的方法, 计算当前时间戳和前后 2 个时间戳的比例进行位移插值、旋转插值, 对坐标数据进行位置和朝向插值处理, 从而解析执行操控指令, 驱动飞机、潜艇等反潜对抗实体, 实现反潜对抗过程的视景仿真。

**关键词:** 海底场景; 海水绘制; 着色器; 作战仿真

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2018) 04-1302-08

DOI: 10.16182/j.issn1004731x.joss.201804012

## Research and Implementation of Antisubmarine Combat Visual Simulation System

Zhang Xin, Gao Yan\*, Zhang Jian, Wang Changbo

(School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China)

**Abstract:** This paper constructs visual simulation framework of antisubmarine combat. The sea surface mesh was modeled using projected grid and FFT technologies. Both the sea surface and undersea are finely rendered. The submarine fine rendering algorithm was implemented by considering refraction, Fresnel reflection, Godray and underwater fog effect and using Shader technology, so as to build a realistic battlefield environment. The method of interpolation processing for data in WGS84 coordinates of latitude and longitude through local area network was proposed; the displacement interpolation and rotation interpolation were implemented by calculating the ratio between the current timestamp and both before and after timestamp; the position and orientation interpolation processing was implemented by using the longitude and latitude data; and the control instructions were parsed and executed to control airplanes, submarines and other entities, so as to achieve antisubmarine combat visual simulation.

**Keywords:** undersea scene; ocean rendering; shader; battle simulation

## 引言

反潜是对潜入一定海区的敌潜艇进行搜索, 封锁, 限制或消灭等战斗行动。大致包括空中反潜和水面反潜。其中由于空中反潜具有速度快、航程

远, 机动灵活的优点, 已成为当前反潜作战主流。在空中反潜中, 反潜飞机利用声纳浮标、吊放声纳、磁探仪等进行反潜探测, 通过鱼雷、深弹等反潜武器进行反潜攻击。海洋环境的真实感渲染和构建仿真数据通讯控制平台是反潜对抗视景仿真系统所关注的两个重要问题。

在海洋环境的真实感渲染方面, Jerry Tessendorf 提出了一种基于数学模型的实时的水面绘制算法<sup>[1]</sup>, WANG Shunli 结合 Fournier 模型中



收稿日期: 2016-04-28 修回日期: 2016-07-09;  
基金项目: 国家自然科学基金(61532002), 上海市科委项目(16511101300);  
作者简介: 张昕(1988-), 男, 河南, 硕士生, 研究方向为虚拟仿真; 高岩(通讯作者 1973-), 男, 湖北, 博士, 副教授, 研究方向为虚拟仿真、计算机图形学。

<http://www.china-simulation.com>

• 1302 •

长轴的物理运动变化规律提出了一种近岸卷浪模型<sup>[2]</sup>, Masten, Watterberg 和 Mareda 提出利用 FFT 绘制海水海浪<sup>[3]</sup>, Johanson 提出基于 Projected Grid 的水面绘制算法<sup>[4]</sup>。但是, 这些算法并没有考虑海面下的绘制问题。对于海底的渲染, CHEN Chao 考虑海底地形与海底底质存在的不平滑性, 利用纹理对被划分的海底底质区域进行平滑过渡<sup>[5]</sup>, 但该算法并未对水下的光照效果建模。在构建仿真数据通讯控制平台方面, Chen Ge 构建了一个能够实时交互数据的海洋虚拟真实环境框架<sup>[6]</sup>。LI Tao 提出在作战仿真系统中建立统一的数据处理模型<sup>[7]</sup>。SONG Meng 利用 MFC 交互框架, 通过人机交互对海洋场景进行仿真<sup>[8]</sup>。WANG Bo 结合海洋战场的各种要素, 设计实现了能够详尽展现虚拟海洋战场态势的系统<sup>[9]</sup>。LI Chuan-qing 考虑到当前的仿真系统访问控制与业务逻辑联系过于密切的问题, 提出了一种将二者分离的仿真系统框架<sup>[10]</sup>。目前, 国内外有关反潜对抗可视化视景仿真系统的研究尚不多见。

本文构建局域网环境内的反潜对抗视景仿真框架, 提出实时海水(底)绘制新算法, 该算法设置水面和水下双相机, 根据前视点距海平面的高度差来自动切换相机, 在水面上, 考虑折射、菲涅尔反射效应计算海水的像素颜色, 绘制洋面场景及倒影效果; 在水面下, 考虑折射、Godray 与雾化效果, 并添加动态分页切换的海底地形, 绘制出逼真的海底效果。同时, 提出局域网经纬数据插值新方法, 该方法从服务器端接收 WGS84 经纬度坐标数据, 解析数据, 计算实体当前时间戳与前后两个时间戳的比例进行插值, 得到平滑的运动序列。进而解析执行操控指令, 驱动飞机、潜艇等反潜对抗实体, 实现反潜对抗过程的视景仿真。

## 1 系统总体结构

反潜对抗视景仿真系统模拟全球经纬度坐标系下, 飞机发现对抗潜艇的整个过程, 系统主要有

两大模块组成: 反潜对抗环境渲染模块, 反潜对抗仿真数据通讯模块。反潜对抗环境渲染模块实现了反潜对抗海洋环境的真实感渲染。建立投影网格与 FFT 结合的海水渲染基础框架, 利用 Shader 技术对海面进行光照渲染、对海底进行光影雾化渲染, 通过动态分页技术构造海底地形以及利用粒子系统产生特效。反潜对抗仿真数据通讯控制模块实现了利用服务器传输的数据得到实体运动序列的过程, 通过和服务器建立通讯, 传输服务器发送的数据, 解析服务器数据, 对仿真数据进行插值处理, 从而得到实体平滑的运动序列。

## 2 反潜对抗环境渲染模块关键技术

### 2.1 投影网格与 FFT 的框架

在进行反潜对抗海水环境模拟时, 需要构建海水渲染基础框架, 首先采用投影网格对海面进行建模。投影网格主要是将需要绘制点反投影到世界坐标系中, 通过高度场函数对绘制点对深度进行变换, 进而得到完整的点信息。

对于进行投影网格对海水建模主要进行如下步骤:

- (1) 在投影空间中创建一个垂直于摄像机方向的均匀分布网格点的平面。
- (2) 将该平面上的所有网格点经过视图矩阵和投影矩阵乘积的逆反投影到世界空间的一个基准面上。
- (3) 由海浪生成算法计算得到出的高程值数据作用于世界空间基准面上的每个投影点, 进行垂直方向上的扰动。
- (4) 渲染投影平面。

在海平面网格建立完成后, 还需要让海平面产生扰动效果。通过使用 FFT(快速傅里叶变换)产生扰动<sup>[11]</sup>, 海浪推导的过程如式(1)所示, 效果如图 1 所示。

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

式中： $R$ 表示复数实部， $\mathbf{K}=(k_x+k_z)= (2\pi(p-N/2)/L_x, 2\pi(q-M/2)/L_z)$ 。

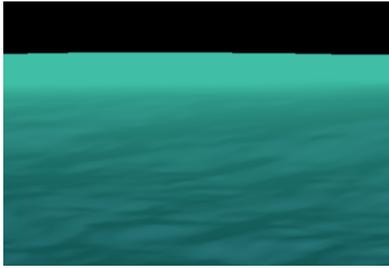


图1 投影网格与FFT结合效果  
Fig. 1 Effect using projected grid and FFT

## 2.2 海水渲染绘制

### 2.2.1 海水渲染绘制流程

在进行投影网格与FFT的框架搭建的情况后，在进行海面渲染同时，提出建立海底进行光影雾化模型并利用Shader技术实现。为了达到更逼真的效果，需要对海面加入水面倒影，以达到真实绘制渲染。对于海水的渲染主要分3个步骤进行。

(1) 投影网格与FFT的框架搭建完成后，利用反射，折射，光照亮度，以及浪花和泡沫叠加产生海面渲染效果，同时加入倒影效果。

(2) 考虑环境光照，漫射光照和菲涅尔反射共同作用，以及下雾浓度与视线距离、海水颜色以及透明度结合，产生逼真的海底光影雾化效果。

(3) 采用双相机对海面、海底切换。

### 2.2.2 海面渲染

海面渲染主要包括海面光照渲染和水面倒影绘制。海面光照渲染主要考虑菲涅尔反射、反射光照颜色、折射光照颜色、以及浪花颜色进行渲染。菲涅尔反射主要是根据观察者所看到的观察表面来调节反射率，效果为视角侧看或平看平面的时候，反射光泽的柔和效果比较明显，产生菲涅尔反射需要计算菲涅尔反射系数，菲涅尔反射系数计算如下：

$$f_{FT} = h * \cos \cos(f_{Dir}, f_N) \quad (2)$$

式中： $f_{Dir}$ 为从观察者视线向量； $f_N$ 为表面法向量； $h$ 表示加权系数； $f_{FT}$ 为菲涅尔反射系数。

水面绘制公式如下：

$$f_{out} = f_{Cl} + f_{Cfl} * f_{FT} + f_{Cfr} * f_{FT} + f_{Cfm} \quad (3)$$

其中 $f_{Cl}$ 表示局部光照颜色， $f_{Cfl}$ 表示反射光照颜色，视线在水面该点的反射方向看到的颜色， $f_{Cfr}$ 表示折射光照颜色，视线在水面该点的折射方向看到的颜色， $f_{FT}$ 表示菲涅尔反射系数， $f_{Cfm}$ 表示浪花颜色。

对于海面除了进行光照渲染外还对海面进行倒影绘制，对于水面倒影绘制，主要使用两个相机，一个相机，主要负责对太阳进行渲染，另一个相机，定义为镜面投影相机，用于在对第一个相机进行纵向镜像投影，从而获取场景的逆场景，通过渲染到纹理，进行镜面贴图，实现水面的倒影效果。

### 2.2.3 海底光影雾化模型

在进行水下光影效果的绘制时，除了考虑海下表面原颜色、环境光照强度、漫射光照强度、海下反射光照强度、海水菲涅尔反射系数外，还需考虑两种情况进行处理。一种是考虑没有菲涅尔反射的情况下，环境光照作用下表面颜色绘制，这一种主要对于海水的非海底部分。另一种是考虑环境光照，漫射光照和菲涅尔反射共同作用之后的表面颜色绘制，环境光照和漫射光照主要是通过给定的参数结合水深得到。计算非海底部分和海底部分光影

强度公式如下:

$$f_{nfr} = f_{col} * f_{AFa} \quad (4)$$

$$f_{fr} = 1 - f_{FT} * (f_{col} * f_{AFa} + f_{col} * f_{Dif}) + f_{FT} * f_{Rfl} \quad (5)$$

其中  $f_{col}$  表示海下表面原颜色,  $f_{AFa}$  表示环境光照强度,  $f_{Dif}$  表示漫射光照强度,  $f_{Rfl}$  表示海下反射光照强度,  $f_{FT}$  表示菲涅尔反射系数。

水下的雾化效果产生主要是将光影效果、视线距离、颜色、透明度结合。在水下, 需要设定视线能看到的最大距离, 当大于最大的视线距离时, 设定雾化值为恒定值, 当视线能看到的距离在最大距离以内, 利用视角到观察表面的距离的线性关系计算雾化度, 即距离越远颜色越深, 透明越小, 物体越难看清。同时, 雾化效果需结合光影效果产生。雾化强度线性公式如下:

$$f_{fog}(d) = d * f_{cor} * f_{ir} \quad (6)$$

式中:  $d$  表示从观察点到物体的距离,  $f_{cor}$  表示颜色的参数,  $f_{ir}$  表示雾透明度。

#### 2.2.4 海面、海底切换

对于进行海面、海底切换, 我们需要定义双相机, 一个相机主要负责对海面进行渲染, 另一个相机负责对海底进行渲染, 当前视点距海平面的高度差为正值时, 对海面进行渲染, 切换海面相机。为负值时, 对海底进行渲染, 切换海底相机。通过海面、海底切换进行渲染公式如下:

$$f_l(H) = \begin{cases} f_{out}H > 0 \\ f_{fr}H \leq 0 \end{cases} \quad (7)$$

$$f_{fog}(H) = \begin{cases} 0H > 0 \\ f_{fog}(d)H \leq 0 \end{cases} \quad (8)$$

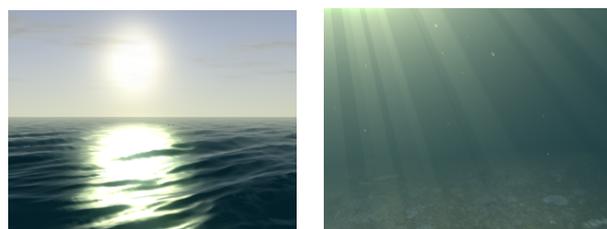
其中  $H$  表示当前视点距海平面的高度,  $f_{out}$  表示海面渲染,  $f_{fr}$  表示海底光影渲染,  $f_{fog}(d)$  表示海底雾化。  $f_l(H)$ ,  $f_{fog}(H)$  表示对海面、海底光照处理, 雾化处理。

水面效果、海底效果如图 2(a)、图 2(b)所示。

### 2.3 海底地形与渲染特效

在进行完海水 Shader 技术渲染绘制后, 还需要反潜对抗环境中增加海底地形和渲染特效。海底

地形的绘制, 利用获取了在某海域的真实的数据了生产高度图, 通过 `ogre terrain` 组件对海底的大片地形进行了分页绘制, 地形的平铺方向、地图大小、纹理等均可进行多项配置。在实现方面主要通过 `ogre PageManger` 管理器下的 `PagedWorld` 类下的 `PagedWorldSection` 进行分页处理。根据地形表面和相机的距离, 调用 `loadPage()` 完成地形块加载, 调用 `unloadPage()` 完成地形块卸载。海底地形效果如图 3 所示。



(a) 水面效果

(b) 海底效果

图 2 海水渲染绘制效果

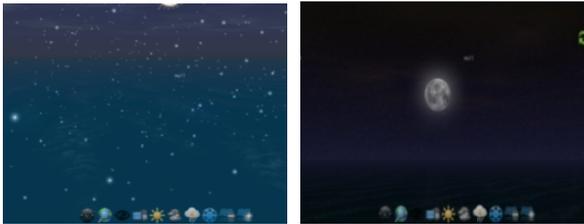
Fig. 2 Ocean rendering result



图 3 海底地形效果

Fig. 3 Seabed rendering result

渲染特效主要利用粒子系统来实现, 首先编写粒子脚本代码, 代码编写结构主要包括粒子发射器和粒子影响器, 在粒子发射器中编写粒子的发射速率、生命周期等。然后将脚本嵌入到 `ogre` 中。通过粒子系统制作多种特效, 如 3D 中的雪, 喷射气泡, 爆炸效果等, 这些特效能很好的模拟出海洋各种环境下的特效, 效果见图 4(a)。另外在粒子系统渲染外, 还有基于 `Caelum` 天空渲染组件, 根据指定的日期和时间对天空进行渲染, 以得到逼真的反潜对抗天空效果, 效果见图 4(b)。



(a) 雪天粒子效果 (b) 天空渲染效果

图4 渲染特效效果图

Fig. 4 Particle system rendering result

### 3 反潜对抗仿真数据通讯模块关键技术

#### 3.1 模块框架流程

通过仿真数据通讯控制可以传输大量的对抗实体运动数据从而模拟整个反潜作战过程。当系统要进行实时通讯时,考虑传输接口、线程等问题需要与服务器控制台建立通讯,然后通过服务器传输飞机、鱼雷、潜艇、航路信息等类型的数据,系统根据数据包类型将接收到的数据解析分类,计算当前时间戳和前后时间戳得到的比例关系进行插值处理等具体过程的处理以实现通讯,如图5所示。

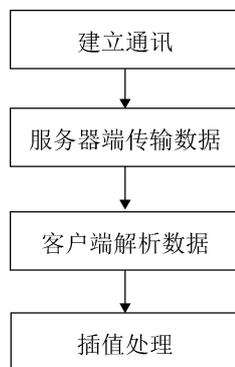


图5 数据通讯的流程图

Fig. 5 Data communication diagram

#### 3.2 建立通讯

系统建立通讯连接需要考虑传输接口、线程等问题,建立通讯具体步骤如下:

(1)当需要进行数据通讯时,搜索本地空闲的端口,确定这一端口为指定传输数据的端口。

(2)使用独立的一个线程持续接受服务器发送

的数据。

(3)如果有数据传输则接收并解析,若无数据则线程阻塞,不影响系统的其他模块使用。

#### 3.3 服务器端传输数据

由于需传输大量的实时数据,为了能够快速的数据传输,由于UDP通讯协议传输资源消耗小,处理速度快,利于实时进行通讯,所以本系统采取UDP通讯协议实现数据传输。在进行数据传输时需定义局域网传输数据包类型。数据包主要包括实体数据、报文长度、数据包类型、接收者ID、发送者ID。数据包的格式如下:

```

struct StPacket
{
    char buffer[ PACKET_SIZE];
    intnBufLen; //报文长度
    DATATYPE nDataType; //数据包类型
    intnReceiverID; //接收者 ID
    intnSenderID; //发送者 ID
};
  
```

其中,buffer字段存储主要的反潜对抗实体数据,如实体的位置、动作、状态等,nDataType字段为数据包类型,也是实体类型,如深弹、飞机、鱼雷、潜艇、航路信息等类型。

飞机数据是反潜对抗飞机实体的各种属性参数,主要包括:飞机即时所在的位置信息,时间戳,飞机的类型,主动声纳数量(主动声纳是指声纳主动发射声波“照射”目标,而后接收水中目标反射的回波时间),被动声纳数量(被动声纳技术是指声纳被动接收舰船等水中目标产生的辐射噪声和水声设备发射的信号),飞机所携带的深弹数量,飞机巡航时的飞行速度等。

航路信息主要是确定反潜对抗实体航行信息,主要包括:确定航路的总时间,在航行中实际航路点个数,当航行需要转弯时转弯的半径,实体所在位置的经纬度信息等。

反潜对抗中潜艇实体数据主要包括:获取系统

当前时间, 潜艇的当前的状态(水面、水下、海底、是否被击沉), 潜艇当前航行的情况参数, 赋值变速计时参数, 潜艇的生源级别, 潜艇的光反射系数, 潜艇运动后的最新深度, 潜艇运动后的最新航向和航速, 潜艇使用气幕弹标志, 潜艇使用潜艇模拟器标志, 潜艇模拟器计时时间, 潜艇模拟器的经度, 潜艇模拟器的纬度, 潜艇模拟器的深度等。

反潜对抗中鱼雷实体数据主要包括: 鱼雷的数目, 反潜鱼雷提供的鱼雷参数, 鱼雷信息时间戳, 鱼雷当前的经度, 鱼雷当前的纬度, 鱼雷航向、航速等。

反潜对抗中深弹实体数据主要包括: 深弹投放的数量, 深弹的序号, 深弹的状态包括未爆炸和爆炸, 深弹的俯视角, 深弹的经纬度数据, 深弹的空间状态包括水上和水上, 深弹的运行时间等。

此外还定义了时间戳字段, 时间戳指能唯一地标识某一刻的时间信息。时间戳字段主要记录系统当前的时间。

### 3.4 客户端解析数据

当接收到的数据包后, 需要对数据进行解析, 首先需根据数据包中 `nDataType` 字段(如深弹、飞机、鱼雷、潜艇)对 `buffer` 字段数据存入其对应的实体的结构体中, 再将 `buffer` 字段中各实体经纬度数据和指令数据保存于统一的结构体 (`StEntityInfo`) 中, 将统一的结构体保存于定义 `std::map<int, std::list<StEntityInfo*>>` 类型的 `map` 数据结构中, 以方便以后统一处理此类数据, 此外, 为以后进行插值处理, 计算数据包 `nDataType` 字段为时间戳的 `buffer` 字段数据得到开始仿真时间戳, 将其存入 `std::map<int, double>` 类型 `map` 数据结构。开始仿真时间戳计算只需进行一次计算, 公式如下:

$$T_s = t_d + t_l - t_c \quad (9)$$

式中:  $T_s$  表示开始仿真时间戳,  $t_d$  表示仿真延迟时间,  $t_l$  表示系统当前显示时间,  $t_c$  表示接收服务器端发送的时间戳。

具体的存放策略为:

(1)第一次接收到数据包, 搜索 `map` 数据结构如果不存在数据包中实体仿真 ID, 就在 `map` 数据结构中创建以此实体仿真 ID 为信息队列。

(2)如果 `map` 数据结构存在此实体仿真 ID 并且存在此实体仿真 ID 的信息队列, 同时由于系统资源有限, 将实体信息队列的长度看成缓冲区, 只当实体信息量小于 200 帧数据量, 才能此实体信息直接插入此实体仿真 ID 的信息队列。

### 3.5 插值处理

插值处理主要是为了通过位移、朝向的改变能够模拟出实体的运动。当接收到的实体数据都按 ID 存放完成以后, 对于需进行插值处理的实体, 需从仿真的信息队列找到此实体 ID 连续的两个实体信息时间戳, 根据计算当前时间戳(即系统当前显示时间与开始仿真时间戳的差值)和前后时间戳得到的比例关系在位移值、角度值进行位移插值、旋转插值。另外, 在进行插值处理时, 需预先将 GPS 的 WGS84 坐标转换为平面坐标(笛卡尔坐标系)得到绘制实体的位置信息, 再进行位移插值、旋转插值。具体的插值算法可以描述为:

(1)首先迭代遍历整个 `map`, 如果没有到达 `map` 结尾处就进入内循环进行具体插值处理。

(2)然后进行一次判断, 当当前时间戳小于要进行仿真的信息队列第二个实体信息时间戳并且大于要进行仿真的信息队列第一个实体信息时间戳时, 进行下一步位移插值、旋转插值。

(3)最后根据当前时间戳和前后时间戳算出比例进行位移插值、旋转插值。

插值算法描述公式如下:

$$\mu = \frac{t_c - t_f}{t_s - t_f} \quad (10)$$

式中:  $\mu$  表示根据当前时间戳和前后时间戳算出比例,  $t_c$  表示当前时间戳,  $t_f$  表示第一个实体信息时间戳。  $t_s$  表示第二个实体信息时间戳。

$$\begin{cases} x = \mu \cdot (l_{sX} - l_{fX}) + l_{fX} \\ y = \mu \cdot (l_{sY} - l_{fY}) + l_{fY} \\ z = \mu \cdot (l_{sZ} - l_{fZ}) + l_{fZ} \end{cases} \quad (11)$$

式中： $x$ ， $y$ ， $z$  分别表示进行位移插值时，当前实体的  $x$  轴， $y$  轴， $z$  轴位移坐标。 $l_{fX}$ ， $l_{fY}$ ， $l_{fZ}$ ，分别表示第一实体信息中  $x$  轴， $y$  轴， $z$  轴的位移坐标， $l_{sX}$ ， $l_{sY}$ ， $l_{sZ}$ ，分别表示第二实体信息中  $x$  轴， $y$  轴， $z$  轴的位移坐标。

$$\begin{cases} p = \mu \cdot (f_{sp} - f_{fp}) + f_{fp} \\ h = \mu \cdot (f_{sh} - f_{fh}) + f_{fh} \\ r = \mu \cdot (f_{sr} - f_{fr}) + f_{fr} \end{cases} \quad (12)$$

式中： $p$ ， $h$ ， $r$  分别表示进行旋转插值时，当前实体在  $x$  轴， $y$  轴， $z$  轴旋转分量。 $f_{fp}$ ， $f_{fh}$ ， $f_{fr}$ ，分别表示第一实体信息中  $x$  轴， $y$  轴， $z$  轴方向的旋转分量， $f_{sp}$ ， $f_{sh}$ ， $f_{sr}$ ，分别表示第二实体信息中  $x$  轴， $y$  轴， $z$  轴方向的旋转分量。

#### 4 反潜对抗实时对抗场景构建

对于反潜对抗实时对抗场景构建，需要对出潜艇、飞机等不同的反潜对抗实体进行建模以及实现对抗实体在进行对抗的对抗场景，首先对对抗实体(如飞机、潜艇、鱼雷、声呐等)进行建模，模型原格式为 .flt(openflight 文件)，之后转换为 .mesh 及 .material 文件供 OGRE 解析。然后在已构建的战场环境中，根据统一的结构体中指令数据，结合粒子特效，通过局域网数据通讯，呈现出反潜对抗实时对抗场景，如图 6 所示。

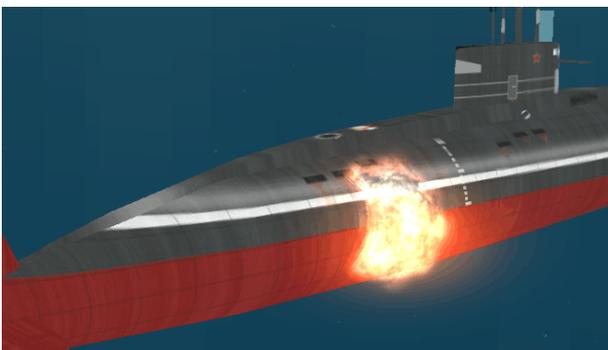


图 6 潜艇被击中场景图

Fig. 6 Result of the destroyed submarine

#### 5 系统仿真实现以及对比

本系统在 PC 机(CPU 2.5G, 内存 8G, 显存 2GB)上运行,通过采用 ogre 引擎下进行 C++编程,通过 Shader 技术对海面、海底进行渲染以及利用局域网实时对接收各实体的位置,姿态,状态等信息进行处理,实现了反潜对抗视景的仿真(50FPS)。并应用 VC2008 SP1 及 MFC 进行人机交互主界面的开发,系统水上、水下效果图如图 7(a)、图 7(b)所示。



(a)系统水上效果图



(b)系统水下效果图

图 7 系统水上、水下效果图

Fig. 7 Rendering result of sea and undersea

文献[12]构建结构类似的仿真系统,采用 FFT 框架进行海洋模拟,但原文只是对海面进行渲染并没有对海底也进行渲染,本系统在对海面进行渲染的基础上,对海底也进行渲染。原文通过海节点坐标定位全球的坐标位置,并没有将实体应用于系统内,本系统通过在 WGS84 坐标系定位全球的坐标位置,并将实体应用于系统内,通过局域网通讯使实体位置和朝向发生改变。

## 6 结论

本文主要采用 ogre 引擎,在 FFT 和投影网格技术框架下对海面、海底进行渲染,添加海底地形和渲染特效,构建了逼真的反潜对抗作战环境。采用局域网数据通讯通过建立通讯,传输数据,解析数据,插值处理的过程,得到实体运动序列进而模拟对抗实体进行对抗的场景。该系统可以为海军进行反潜军事训练提供三维场景再现,解决海军军事训练环境、人员等因素存在限制的问题。目前,系统还存在一些不足之处,如考虑对场景绘制进行优化解决场景出现大量实体时系统帧率低的问题和视点在水面上与水面下切换时会出现“空当”的问题。

### 参考文献:

- [1] Jerry Tessendorf. Simulating Ocean Water[C]. SIGGRAPH, 2001.
- [2] 王顺利,康凤举,徐建华. 近岸卷浪的建模与可视化研究[J]. 系统仿真学报, 2014, 26(10): 2395-2399.  
WANG Shunli, KANG Fengju, XU Jianhua. Modeling and Visualization of Curved Waves Near Seashore[J]. Journal of System Simulation, 2014, 26(10):2395-2399.
- [3] Mastin G A, Watterberg P A, Mareda J F. Fourier synthesis of ocean scenes[J]. IEEE Computer Graphics and Applications(S0272-1716), 1987, 7(3): 16-23.
- [4] Johanson C. Real-time water rendering-Introducing the projected grid concept[M]. Sweden:Department of Computer Science, Lund University, Master dissertation, 2004.
- [5] 陈超,王文珂,王怀辉,等.一种海底地形与底质的三维融合可视化方法[J]. 系统仿真学报, 2012, 24(9): 1936-1944.  
CHEN Chao, WANG Wen-ke, WANG Huai-hui, et al. 3D Visualization Method for Submarine Bottom Terrain and Characteristics [J]. Journal of System Simulation, 2012, 24(9): 1936-1944.
- [6] Chen Ge, Li Bo, Tian Fenglin, et al. Design and implementation of a 3D ocean virtual reality and visualization engine[J]. Journal of Ocean University of China(S1672-5182), 2012, 11(4): 481-487.
- [7] 李涛,柳林,皮学贤. 面向作战仿真的信息融合系统建模方法研究[J]. 系统仿真学报, 2014, 26(3): 549-555.  
LI Tao, LIU Lin, PI Xue-xian. Research on Information Fusion System Modeling Technologies Oriented to Military Simulation[J]. Journal of System Simulation, 2014, 26(3): 549-555.
- [8] 宋蒙,苏天赟,王国宇,等. 海洋环境三维场景关键技术研究[J]. 中国海洋大学学报(自然科学版), 2014, 44(8): 108-113.  
SONG Meng, SU Tian-yun, WANG Guo-yu, et al. Research of the key Technology of Three Dimensional Marine Environment[J], Periodical of Ocean University of China, 2014, 44(8): 108-113.
- [9] 王勃,康晓予,张莉. 虚拟海战场态势三维显示系统设计及实现[J]. 系统仿真学报, 2012, 24(1): 214-218.  
WANG Bo, KANG Xiao-yu, ZHANG Li. Design and Implementation of 3-D Situation Display System for Virtual Sea Battlefield[J]. Journal of System Simulation, 2012, 24(1): 214-218.
- [10] 李传庆,黄传红,孙福刚. 一种分布式视景仿真软件系统通用框架[J]. 系统仿真学报, 2013, 25(增1):376-378.  
LI Chuan-qing, HUANG Chuan-hong, SUN Fu-gang. Research and Application of Distributed Visual Simulation Software General-purpose Framework[J]. Journal of System Simulation, 2013,25(S1): 376-378.
- [11] 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.  
HOU Xue-long, HUANG Qi-lai, SHEN Pei-zhi, Ocean Wave Real-time Simulation Method Based on FFT[J], Computer Engineering, 2009, 35(22): 258.
- [12] 明德烈,徐秋程,李向春. 面向全球应用的海洋仿真系统的实现研究[J]. 系统仿真学报, 2012, 24(8): 1741-1745.  
MING De-lie, XU Qiu-cheng, LI Xiang-chun. On Realization of Visualization System for Global Ocean Simulation[J]. Journal of System Simulation, 2012, 24(8): 1741-1745.