软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(Suppl.(2)):247−257

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于球面投影网格的大规模复杂水面模拟*

熊 元[1], 刘世光[1,2]

[1](天津大学 计算机科学与技术学院,天津 300072)

[2](天津市认知计算与应用重点实验室,天津 300072)

通讯作者: 刘世光, E-mail: lsg@tju.edu.cn

摘 要: 针对目前大规模复杂水面模拟中存在的效率不高、碰撞检测较为复杂等问题,提出了一种海洋尺度复杂水面模拟解决方案.首先,提出了一种球面投影网格方法实现大规模动态水面波动效果的模拟.与传统的投影网格方法相比,该方法不需要重新构造与球面直接相交的投影体,具有更高的绘制效率并且适合图形硬件加速.其次,设计了高效的交互式复杂水面的模拟方法,包括水面和刚体交互作用的模拟及刚体与地形的快速碰撞模拟.此外,给出了通用的泡沫绘制和海岸线绘制方法.实验结果表明,该方法的模拟结果较为逼真,能达到较高的绘制速度(FPS>60),适用于计算机游戏、虚拟现实等实时环境.

关键词: 水面;实时;球面投影网格;泡沫;碰撞检测;GPU

中文引用格式: 熊元,刘世光.基于球面投影网格的大规模复杂水面模拟.软件学报,2014,25(Suppl.(2)):247−257. http://www.jos.org.cn/1000-9825/14042.htm

英文引用格式: Xiong Y, Liu SG. Large-Scale simulation of complex water scenes based on spherical projected grid. Ruan Jian Xue Bao/Journal of Software, 2014,25(Suppl.(2)):247−257 (in Chinese). http://www.jos.org.cn/1000-9825/14042.htm

Large-Scale Simulation of Complex Water Scenes Based on Spherical Projected Grid

XIONG Yuan[1], LIU Shi-Guang[1,2]

[1](School of Computer Science and Technology, Tianjin University, Tianjin 300072, China)

[2](Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300072, China)

Corresponding author: LIU Shi-Guang, E-mail: lsg@tju.edu.cn

Abstract: Current simulation methods for large-scale complex water scenes suffer from various problems such as low efficiency, and complicated collision detection. To remedy these problems, this paper presents a novel method for ocean-scale complex simulation based on spherical mapping projected grid. First, a novel spherical projected grid method is proposed for animating large scale water waves. Compared with conventional projected grid method, the new method does not need reconstructing projectors intersecting with the sphere, and thereby can achieve higher rendering efficiency which is also suitable for graphics hardware acceleration. Then, a new method is designed for simulating interactions including water-rigid body interaction and rigid body-terrain interaction. Additionally, a general bubble and coast line rendering framework is provided. Experiments demonstrate that the proposed method can realize realistic rendering results with high rendering rates (FPS>60), and can be applied in real time scenarios such as computer games and virtual reality.

Key words: water scenes; real time; spherical projected grid; foam; collision detection; GPU

水面是自然场景中的重要组成部分,是游戏场景、虚拟现实中不可或缺的元素之一.尽管计算机图形学领域中对于水面的绘制方法进行了大量研究,大尺度动态复杂水面的快速绘制仍是目前的难点问题.

本文的目标为设计一种海洋尺度的动态复杂水面的真实感绘制方法.我们提出基于球面的大范围水面投影网格生成方法,与传统的投影网格方法[1]相比,间接地在平面投影网格基础上将投影网格生成方法扩展到了

* 基金项目: 国家自然科学基金(61170118, 60803047); 天津市应用基础与前沿技术研究计划(14JCQNJC00100)
收稿时间: 2014-05-09; 定稿时间: 2014-08-19

球面上,再通过GPU(graphics processing unit)的加速运算,该方法取得了较高的模拟效率.图1为利用本方法实现的高空俯视地球表面角度的大范围水面绘制效果.

(a) 动态水面模拟效果 (b) 船体与水面及海岛的交互模拟效果

图 1 本文方法生成的高空俯视地球角度的海洋尺度复杂水面的效果

本文方法首先进行基于环境波和交互波的刚体动力模拟[2],然后将环境波参数传输到GPU端,在GPU上的实现球面投影和表面波动的绘制.

模拟水面作用(浮力):根据环境波和交互波方程确定刚体接触分布点在水面网格上的位置,通过双线线性插值可以计算刚体接触分布点在海水表面上的高度.

模拟刚体与地形碰撞:通过检测刚体与地形碰撞,得到地形与刚体的接触碰撞点,并计算刚体与地面接触点受力和碰撞反应.

创建水面的球面投影网格:将环境波和交互波模拟参数传输到GPU端,通过环境波和交互波方程计算的顶点波动结果,然后在世界坐标系中对顶点实施球面网格映射变换.

水表面的像素渲染:包括水面反射折射、泡沫等效果的绘制.

1 相关工作

对于海洋环境的真实感模拟,如果直接使用理论方法[3,4]进行计算,尺度是最大的困难[5].对于海面的运动,可以使用不同的统计模型模拟,比如使用快速傅里叶变换[4]可生成较好的环境波动画.由于重力波造成的波动,其方向频率谱具有随机海浪的特点,也可使用Perlin噪声模拟深海的环境波[1],或者使用简单的Gerstner waves[6].此外,图形学者对交互波的模拟也进行了研究[2,7].交互波指与物体相互作用产生的水面运动,可用来描述海面受到反应移动船只或对象(刚体)的反作用.

Yuksel等人[8]提出了基于波粒子的动态水面模拟新方法.这是一种无条件稳定的方法,它通过找到满足波动方程的解析解的波形函数来产生交互波.但是,由于波粒子方法表达特别的现象比如衍射时,需要首先获得一个能满足定解条件的波形函数,这相对于波动方程的有限差分数值解法(FDM)要困难许多;所以该方法适合模拟开放的水环境.该方法的计算开销随着波粒子数量增加而增大,而FDM方法的计算开销主要取决于交互接触点的数量;该数量是相对可控的,所以在效率上FDM方法更为高效.Cords和Staadt[9]通过在GPU实现移动栅格方法实现了波动区域的快速计算.该方法使用有限差分数值解法模拟交互波.但是,由于该方法中存在剪切波动场容易导致不稳定的波动场.文献[9]中刚体模拟的算法在CPU上执行,从GPU的读回CPU中数据的开销是昂贵的,我们的方法避免这种开销,从而在一定程度上提高了效率.水面对刚体交互的模拟可以用于描绘的逼真的动态物体如漂浮物、船体等和海面交互的场景.人们对水面和刚体的高效交互模拟绘制提出了不少改进策略[2,7−11].尽管人们在该领域进行了不少研究,大尺度动态海面的快速绘制方法及复杂海面交互作用的快速模拟方法仍值得深入探索.

近年来,计算机图形学领域中人们对近岸海岸线的绘制[3]和泡沫模拟[12−14]也展开了研究,比如通过检测水面的抖动值确定泡沫出现数量[13]的模拟方法、基于推流和解散原理[14]发展的粒子方法等可以实现较好的模

拟效果.但是,这些方法在算法效率、通用性以及绘制质量方面有待进一步提高.

2 交互式复杂水面模拟

通常,水面上存在船只等运动的物体,它们与水面不断交互,并且与水中的障碍物如海岛等发生交互作用.本文设计了快速的交互式复杂水面的模拟方法,主要包括水面和刚体交互作用的模拟及刚体与地形的快速碰撞模拟.

2.1 水面与刚体的交互

水面上物体的运动主要取决于它的受力情况.我们定义刚体为一个绝对固定的粒子系统,所有粒子具有同样的角速度以及速度;一个刚体对象被认为是作为一个无限数量的粒子的集合,每个粒子拥有一定的质量,这些粒子即为刚体的假想受力点.由于粒子的重心处于不同的位置,当刚体受力时,刚体将产生线速度;当刚体重心与粒子受力方向不在同一直线上时,刚体就会产生旋转[15].我们采用文献[2]中的方法对刚体转动计算进行简化,过程如下:

对于单个刚体转动,根据刚体的角动量定理的可得角加速度计算公式:

$$\frac{\mathrm{d}\omega}{\mathrm{d}t}=I^{-1}[N_{net}-\omega\times(I\omega)] \tag{1}$$

其中,I 是刚体的惯量张量,N_{net} 为对刚体作用的合力矩,ω 表示角速度.这些物理量均定义在世界坐标系空间中,我们假设刚体角动量和角速度方向始终相同.

假设惯量张量矩阵表示为

$$I_p=\begin{pmatrix} I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{pmatrix},$$

其中,I_{xx},I_{yy} 和 I_{zz} 为 I_p 的3个主转动惯量.

这个方程的一个解是 $I_{pe}=I_{xx}=I_{yy}=I_{zz}$,$I_{pe}$ 是标量,且 $I^{-1}=I_p^{-1}=I_{pe}^{-1}$,最终可得角加速度计算公式为

$$\frac{\mathrm{d}\omega}{\mathrm{d}t}=I_p^{-1}N_{net}=N_{net}/I_{pe} \tag{2}$$

对于式(2)中I_{pe}的计算,本文以几何上类似一个空心圆柱的船只作为模拟对象,用空心圆柱替代船只作为受力体参与动力模拟[2].我们计算船只所受浮力的向量和水面高度计算均来自对顶点纹理读回的采样.施加一个线性引擎力F_e作用于质心让物体比如船体加速,这个力始终与船身在世界坐标系空间中的方向一致,它的作用是维持船的速度的大小.根据不同的动态对象之间的距离,添加斥力F_c来模拟碰撞;假设有n个受力点,那么刚体所受合外力为

$$F=\sum_{i=1}^{n}(Fn_i+m_ig+Fd_i+Fc_i)+Fe \tag{3}$$

其中,m_i 为第i个部位的质量,Fn_i 为水面压力,Fd_i 表示水面阻力,Fc_i 是船只互斥力,Fe 为船只引擎推力.这样我们就可以计算位移加速度 $a=F/M$ (其中 M 为船体总质量),当受力平衡时,船的位移速度就恒定了.船体所受合力矩

$$N_{net}=\sum_{i=1}^{n}((Fn_i+m_ig+Fd_i+Fc_i)\times r_i) \tag{4}$$

r_i 为质心到每个接触点的向量(世界空间),这里 r_i 表示为质心到每个接触点的向量(世界空间).

2.2 物体与地形的碰撞检查

近年来的通用实时碰撞检查技术以分离轴算法[16]和GJK算法[17]为典型代表,本文针对地形特点设计一种高效的物体与地形快速动态碰撞检查方案.如图2所示,将地形物体接触顶点投射到四叉树的叶子CELL的单元格Tile上.基本算法思想如下:对于一个相交成功的CELL表,该方案通过采用模型各部位包围碰撞体检测四叉树

叶子CELL上的单元格TILE是否相交;如果相交,那么将包围碰撞体的接触顶点垂直投影于地形四叉树CELL上,而后通过点扫掠方法来生成一条测试线,即利用t和$t+\Delta t$时刻的两个投影点的位置形成测试线,粗略检测与CELL上的单元格Tile是否发生碰撞.最后在图元层次上检查包围碰撞体与地形的精确碰撞.

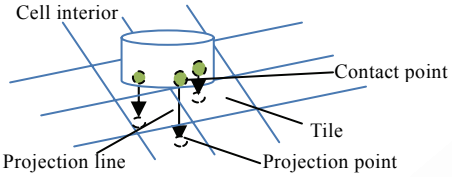


图 2　物体与地形碰撞检查算法

我们沿着测试线方向按照Bresenham直线算法检测可能发生碰撞的Tile(如图3所示).图3中实心点和空心点分别表示 t 时刻和 $t+\Delta t$ 时刻的投影点,通过测试线检测可能发生碰撞,使用双线性插值确定测试线的端点高度.图3中填充为点阵的格子表示可能发生碰撞的Tile,填充为实心的格子表示确定发生碰撞的Tile依次比较每一个单元格Tile在扫掠测试线上的高度,如果Tile高对是在测试扫掠测试线之上,即判定发生了碰撞,由于碰撞检测只包括很少的浮点比较,因此计算量较小.在低精确度的测试中,该方法对动态碰撞检查有较好的效果.最后,我们通过接触面三角形对单元格三角形(图元层次)的相交检查,得到高精确度的相交测试结果.本文中的物体使用空心圆柱受力接触面(较少的三角形数量)作为船体多边形替代体参与测试,并采用三角形射线的代数相交检测方法,可以快速确定碰撞接触位置.

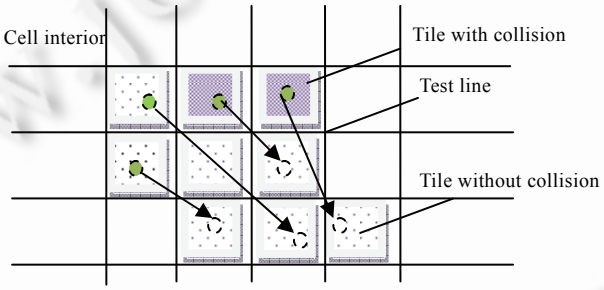


图 3　点扫掠算法

具体地,对于一块均匀的分辨率为512×512的地形,我们将树高度设置为8层,CELL叶子节点大小2×2,图元密度为每个CELL 包含16个三角形,这样虽然加大了一些上存储消耗,但降低了CELL叶子节点上图元密度与物体包围体尺寸大致匹配,提高了测试效率.

3　交互式复杂水面模拟

直接球面投影网格方法[1]中求交情形复杂、运算量较大,在 GPU 上的运算效率不高.针对这些问题,我们设计了一种改进的球面投影网格算法 ISPG(improved spherical projected grid).该方法实现的关键是利用齐次坐标对平面点进行投影,通过半球映射计算顶点位移.

3.1　传统的投影网格方法

首先,回顾文献[5]中传统的平面投影方法.假设已经通过平面H构造了投影体(projector),快速计算投影平面点的方法如下:

已知平面H的方程为 $N \cdot X = h$,那么线面的齐次坐标交点计算表达式为

$$\frac{y+\Delta y\cdot t}{w+\Delta w\cdot t}=h \quad => \quad t=\frac{w\cdot h-y}{\Delta y-\Delta w\cdot h} \tag{5}$$

其中,N为平面法向量,X为投影点,h为平面高度,y为视点齐次坐标y值,w为视点齐次坐标的权值,Δy 和 Δw 分别是方向齐次坐标y值和w值得变化量,t表示待解的线性系数.

由此,可得到投影平面点的齐次坐标为 $X_{\text{hom}}=(x+\Delta x\cdot t,y+\Delta y\cdot t,z+\Delta z\cdot t,w+\Delta w\cdot t)$, 再通过在Vertex Shader上的双线性插值得到齐次坐标投影平面点X.

3.2 直接衍生球面投影算法

由该算法直接衍生得到的球面投影算法,需要求解球面与射线的交点,即给出一个边界球的球心 c 和半径 r ,通过 $\|p-c\|-r=0$ 求边界球上是否存在交点 p .如图4所示,文献[5]中利用透视体构造一个与球面全相交的投影体(spherical projector).

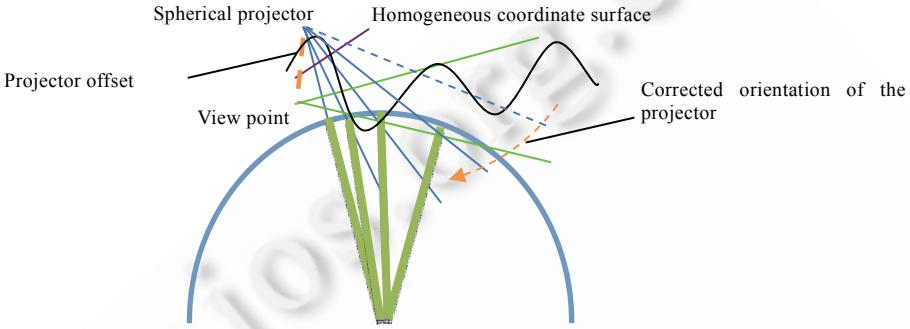


图 4 直接衍生的投影网格方法[5]中球面网格生成示意图

具体步骤如下.

1.与平面投影算法相同,算法避免水表面波动幅度大于视点,首先将Projector向正Y轴偏移,使得

$$h_{Projector}>H_{wave}.$$

2.求解球面与投影体棱线的交点(一次测试可能存在2个,取正方向的),我们仍然按照平面投影算法调整Projector方位forward,故应该满足 $dot(forward,Y)<0$,否则镜面翻转forward,使投影体基本朝向棱线交点.如果投影体棱边与球面交点少于2时,会出现投影失败的情形,那么我们尝试将投影体张角FOV减小值 λ 去逼近球面,实验中我们取0.5,使Projector的方位如:

$$forward=Normalize(c-Eye).$$

然后重复第2步,直至相交或者迭代次数超过阈值.

3. 与平面投影算法类似,通过计算minmax矩阵求得齐次坐标平面上的投影区间,在Vertex Shader上对区间坐标进行插值得到内部点的齐次坐标,从内部点投射射线(球面相交算法),由此构造球面网格.我们发现由于射线与球面相交计算指令较多,导致该方法在Vertex Shader上的计算效率不高.

3.3 改进的球面投影网格方法

本文改进的球面投影网格算法可视为对投射平面算法扩展,此映射为平面 $dot(N,X(x,y,z))=h$ ($h\geqslant r$,r为球面半径)与 $x_S^2+y_S^2+z_S^2=r^2$ 的半球面在开区间内构成一个同胚映射.

我们将属于这个开区间内的有限域上的平面投影点 X 映射到球面,映射表达为

$$ispg(X)=X_{sphere},X\in R \tag{6}$$

如图5所示,该映射将传统的平面投影网格转换为球面投影网格,在绘制时uv参数坐标保持不变.投射平面算法容易调整投影体使得投影体棱线与之相交,而且通过平面作为辅助,我们不必直接调整投影体方位和张角与球面的产生交点,从而节省通过齐次平面上的点投射球面的计算开销;也避免了校准投影体球面产生的交点造成

的额外开销,以及由于球面半径相对较小造成与投影体棱线相交检测失败、导致无法定位球面顶点的情形.具体计算如下:

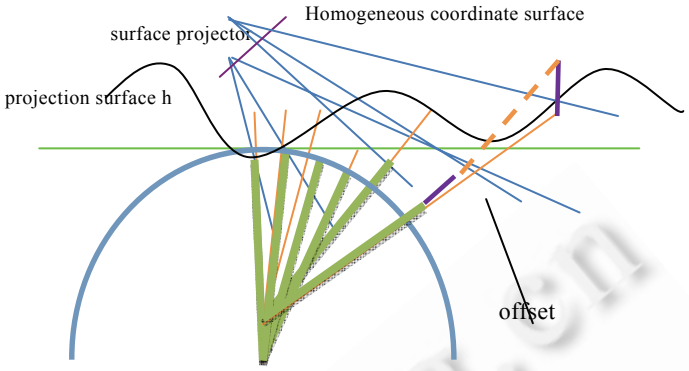


图 5 本文方法在投影体空间中进行球面网格绘制过程示意图

已知球面中心 c 和半径r,那么球心到齐次坐标投影平面点 X 的方向为

$$D=\frac{X-c}{\|X-c\|} \tag{7}$$

球面上的映射点的坐标为

$$X_{sphere}=c+D\cdot radius \tag{8}$$

与文献[5]中提到的直接构造投影的方法不同,我们的方法是一种间接方法.它不需要重新构造一种与球面直接相交投影体,也不需要测试齐次坐标平面上点与球面的投射交点;而是在生成投影网格的基础上,通过半球映射投影平面区域构造球面局部.

3.4 GPU加速

相对于构造直接相交投影体的算法,本文方法不会产生不必要的条件分支,适合GPU加速运算.实验结果验证了该方法的简便性和高效性.GPU加速计算的Shader伪代码如下:

```
half tmpy=worldSpaceVertex.y;
half3 center=half3(_WorldSpaceCameraPos.x,-t_radius,_WorldSpaceCameraPos.z);
half3 centerdir=normalize(worldSpaceVertex–center);
worldSpaceVertex=center+centerdir·(t_radius+tmpy);
```

其中,变量t_radius为CPU传输的球面半径参数.本文的算法结合了平面投影网格生成和半球面映射算法,同样也可以将半球面映射算法应用于大范围的地表甚至云层球面绘制的shader上,相同点是在表面顶点位置计算之后进行球面映射计算.

4 泡沫和海岸线绘制

本文提出一种通用的泡沫绘制解决方案.我们结合波的岸边衰减反射[2],实现了波峰和完整海岸线的实时绘制.其中,结合交互波在海岸边缘进行着色实现海岸线模拟,深水泡沫则是对交互波在渲染水面上形成的波峰基础上进行着色实现.此外,借助视空间的深度图或者转换到视空间的地形高度图,结合对水面交互波(衰减后)的高度进行采样,构造两种水泡深度曲线实现对水面进行着色,计算示意图如图6所示.

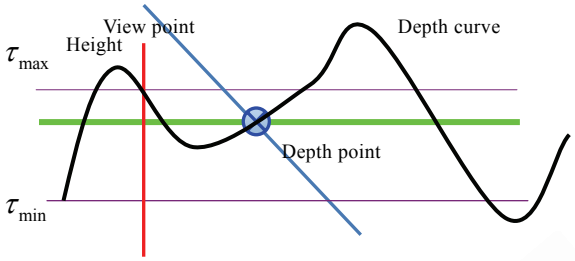


图 6 泡沫和海岸线深度曲线示意图

假设用户提供的泡沫渲染混合因子的最大高度阈值和最小高度阈值分别为 τ_{max} 和 τ_{min} ;对于深度图的情况,我们首先线性化深度曲线值,并将其转换为高度值,即:

$$height = LinearEyeDepth(depth) \tag{9}$$

其中*height*,*depth*分别为高度值和深度值.

然后,计算泡沫混合因子*foamBlendFactors*,

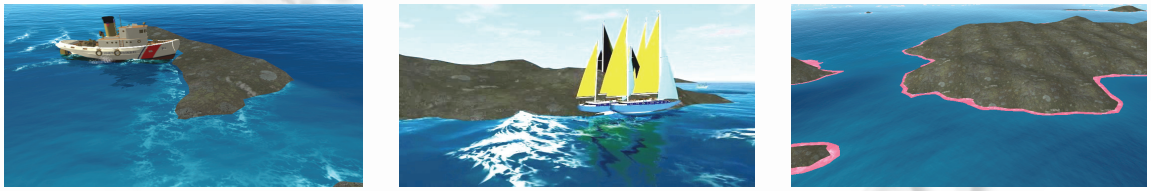
$$foamBlendFactors = \frac{\tau_{min} - height}{\tau_{min} - \tau_{max}} \tag{10}$$

接下来,根据泡沫混合因子对泡沫纹理和水面颜色进行叠加混合得到最终的颜色*color*,

$$color = watercolor + \sum_i foamBlendFactors_i \cdot foam\ color \tag{11}$$

其中,$foamBlendFactors_i$为第i条深度曲线的泡沫混合因子,*watercolor*和*foam color*分别为水面和泡沫的颜色.

海岸线计算和深水泡沫的映射计算类似,二者的区别是深度曲线构造方法不同,阈值参数数值也可以不同;深水泡沫是用水面交互波构造深度曲线,而浅水泡沫是利用深度图和岸边衰减的交互波构造深度曲线.图7展示了深水泡沫和海岸线的绘制效果.



(a) 深水泡沫(俯视) (b) 深水泡沫 (近距离观察) (c) 海岸线

图 7 深水泡沫和海岸线的绘制效果

5 实验结果

基于本文上述方法,我们模拟了不同的水面与动态物体的交互场景.实验环境为配置Intel Core I5 3.20GHz CPU,NVidia Geforce GTX650 GPU 的PC机,编程环境为DirectX 9.0c,OGRE 引擎以及Unity引擎.

图8为采用本文碰撞检查和碰撞反应方法实现的船体与海岛碰撞作用模拟效果序列.本实验中采用空心圆柱受力接触面作为船体替代体参与碰撞(接触点为4个).

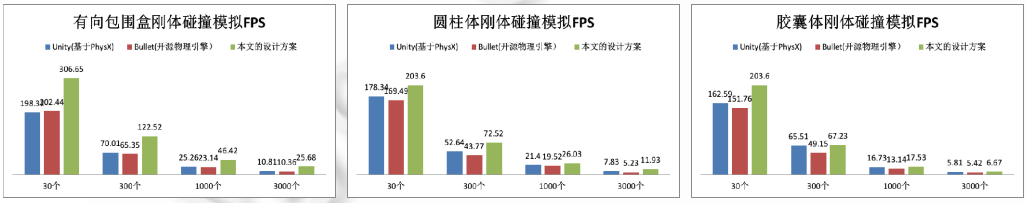
近年来的实时碰撞检查技术以文献[16]提出的分离轴算法和文献[17]提出的GJK算法为典型代表,而一些有名的游戏物理引擎中的碰撞检测模块也采取这些算法并进行优化,比如Bullet 引擎中GJK 算法的Voronoi Simplex Slover 引用了文献[17]中的算法.基于上述分析,我们将本文设计的地形和船体的碰撞模拟算法与Unity引擎(基于PhysX GJK算法)及开源物理引擎Bullet(使用文献[17]中的GJK算法)中的碰撞检测方法进行了比较.其中,我们对模拟效率进行了比较,在两个物理引擎中选取与本文实验条件相同的情形下,不同碰撞数量和碰撞体分别为有向包围盒(10个接触点),圆柱体(88个接触点),胶囊体(412个接触点),进行与地形发生碰撞实

验.另外,我们还对穿透率进行了比较,穿透标准为所有接触点符合 $h_c < h_{terrain}$,即接触点高度小于地形高度.具体比较结果如图9所示.其中,我们的方法的模拟效率明显高于其他两种方法,而且在碰撞接触点较少而数量较多时,提升效率更为明显;在使用接触点较少的碰撞体的情况下,我们方法的穿透率较低.

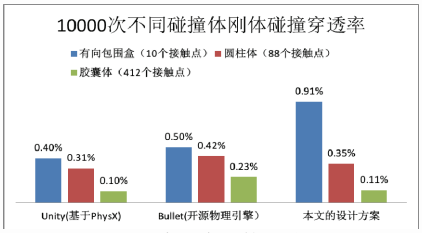
(a) 船只侧面碰撞

(b) 船只底部碰撞,帧时间间隔为 3ms

图 8 船体与海岛的碰撞反应方法绘制结果



(a) 不同碰撞体的地形碰撞检测比较

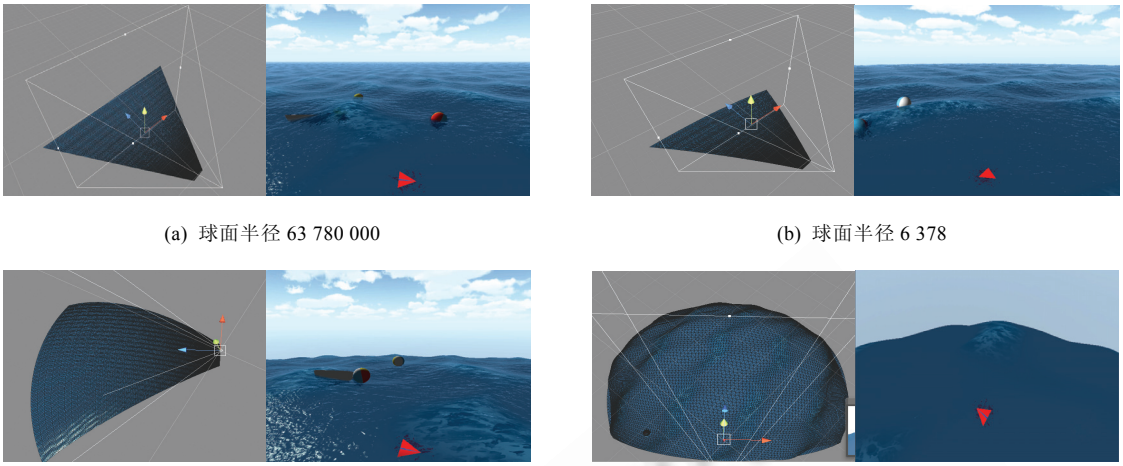


(b) 穿透率比较测试

图 9 不同方法之间的地形碰撞检测效率和穿透率比较

在球面投影网格的实验中,我们设定的实验参数为:网格分辨率128×128,视点高度=5,投影平面*FARPLANE* =1000.不同半径下的海面绘制结果如图10所示,其中,左列为投影空间示意图,右列为相应的绘制结果.平均绘制帧率大于60帧/s.

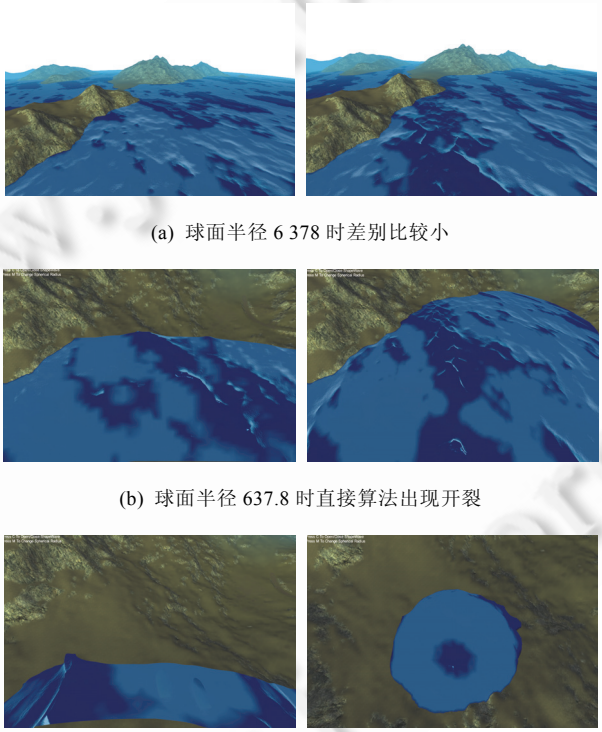
此外,我们对本文提出的ISPG算法和直接构造投影体方法[5]进行了对比实验.对比效果如图11所示,左列是绘制效果,右列是球面投影网格绘制效果,视点高度为100.在球面半径较大时,本文算法产生的投射网格会延伸较远,而直接算法则相对收缩;半径较小时,直接构造投影体算法会发生明显的开裂和消失等现象,原因可能是局部或者整个球面相交检测失败,正如第4.3节所述,由于我们的方法采用了半球映射,完整地还原了通过平面相交检测算法的单一性,从而可较好地避免这种现象.



(a) 球面半径 63 780 000　　(b) 球面半径 6 378

(c) 球面半径 637.8　　(d) 球面半径 63.78

图 10　不同球面半径下的球面投影网格绘制效果图



(a) 球面半径 6 378 时差别比较小

(b) 球面半径 637.8 时直接算法出现开裂

(c) 球面半径 63.78 时直接算法出现严重开裂甚至会出现消失的情况

图 11　不同球面半径下直接球面题构造算法[5]

我们将本文提出的球面投影算法和直接构造投影体放大的算法效率进行了对比(UNITY平台).如图12所示,对于分辨率高的投影球面网格,我们的球面投影算法比直接构造投影体算法效率更高,并且在某些极限情况下,我们的球面投影算法更稳定,不会产生为避免球面相交失败导致的迭代情形.

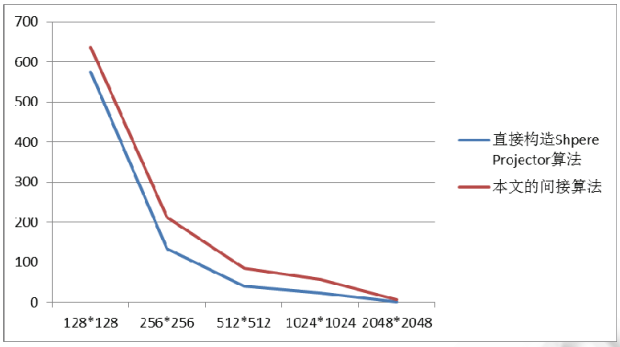


图 12 不同顶点数量时我们的球面投影方法与直接构造投影体方法的效率(fps)比较

6 结束语

本文提出了一种基于球面投影的大规模复杂水面模拟的新方法.其中包括水面与动态物体的交互模拟、地形对物体的碰撞检查及海面泡沫绘制的新方案.本文算法充分利用了多核CPU和GPU的优势,发挥了异构的并行计算能力,达到了较高的渲染速度.从性能测试结果分析,渲染速度可以满足实时交互应用的需要,在游戏、虚拟现实等领域具有较好的应用前景.在未来的工作中,我们将在大范围水面绘制的基础上,实现更为逼真的水面泡沫和局部卷浪效果,并尝试运用水平集方法进行动态碰撞检测以进一步提高检测精度.

致谢 我们向对本文的工作给予支持和建议的同行表示感谢.

References:

[1] Johanson C. Real-Time water rendering: Introducing the projected grid concept [MS. Thesis], Lund University, 2004.

[2] Liu SG, Xiong Y. Fast and stable simulation of virtual water scenes with interactions. Virtual Reality, 2013,17(1):77−88. [doi: 10.1007/s10055-013-0222-0]

[3] Pi XX, Yang XD, Li SK, Song JQ. Simulation of wave and water surface near the seashore. Chinese Journal of Computers, 2007, 30(2):324−329 (in Chinese with English abstract).

[4] Tessendorf J. Simulating ocean water. In: SIGGRAPH Course Notes. 2001. [doi: 10.1.1.131.5567]

[5] Darles E, Crespin B, Ghazanfarpour D, Gonzato JC. A survey of ocean simulation and rendering techniques. Computer Graphics Forum, 2011,30(1):43−60. [doi: 10.1111/j.1467-8659.2010.01828.x]

[6] Finch M. Effective water simulation from physical models. GPU Gems, 2004.

[7] Chentanez N, Müller M. Real-Time simulation of large bodies of water with small scale details. In: Proc. of the Eurographics/ ACM SIGGRAPH Symp. on Computer Animation, 2010. 197−206. [doi: 10.2312/SCA/SCA10/197-206]

[8] Yuksel C, House DH, Keyser J. Wave particles. ACM Trans. on Graphics, 2007,26(3):99−107. [doi: 10.1145/1275808.1276501]

[9] Cords H, Staadt O. Real-Time open water environments with interacting objects. In: Proc. of the Eurographics Workshop on Natural Phenomena. 2009. 35−42. [doi: 10.2312EG/DL/conf/EG2009/nph/035-042]

[10] Kass M, Miller G. Rapid, stable fluid dynamics for computer graphics. ACM Trans. on Graphics, 1990,24(4):49−55. [doi: 10.1145/ 97880.97884]

[11] Thürey N, Rüde U, Stamminger M. Animation of open water phenomena with coupled shallow water and free surface simulations. In: Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation, Eurographics Association. 2006. 157−164. [doi: 10.1145/1218064.1218086]

[12] Darles E, Crespin B, Ghazanfarpour D. Accelerating and enhancing rendering of realistic ocean scenes. In: Proc. of the WSCG. Czech Republic, 2007. [doi: 10.1.1.83.1806]

[13] Dupuy J, Bruneton E. Real-Time animation and rendering of ocean whitecaps. In: Proc. of the SIGGRAPH Asia, Technical Briefs. 2012, article 15. [doi: 10.1145/2407746.2407761]

[14] Ihmsen M, Akinci N, Akinci G, Teschner M. Unified spray, foam and air bubbles for particle-based fluids. Visual Computer, 2012, 28(6-8):669−677. [doi: 10.1007/s00371-012-0697-9]

[15] Baraff D. An introduction to physically based modeling: Rigid body simulation II-nonpenetration Constraints. In: SIGGRAPH Course Notes. 1995. [doi: 10.1.1.119.7282]

[16] Eberly DH. Game Physics. 2nd ed. CRC Press, 2010.

[17] Ericson C. Real Time Collision Detection. CRC Press, 2005.

附中文参考文献:

[3] 皮学贤,杨旭东,李思昆,宋君强.近岸水域的波浪与水面仿真.计算机学报,2007,30(2):324−329.

熊元(1984－),男,广西柳州人,工程师,主要研究领域为计算机游戏,流体模拟.
E-mail: 196221347@qq.com

刘世光(1980－),男,博士,教授,CCF 高级会员,主要研究领域为计算机图形学,计算机动画,虚拟现实,多媒体处理.
E-mail: lsg@tju.edu.cn