OPENGL 实验

真实水面模拟

学号	姓名	论文规范性（10）	问题分析与调研（30）	方案创新性（20）	实验结果分析与讨论（40）	结课论文总成绩（100）
21301148	葛宣成	7	26	17	35	85

摘要

本文在上次大作业“系外行星模拟漫步观察”的基础上，进一步做了更改,尝试基于 Perlin 噪声,网格绘制,反射折射计算等方式,进一步实现了真实的水面纹理绘制。实验表明，该方法满足水面模拟要求，可以绘制出较为真实，且渲染流畅的水面，可以很大程度上保证水面渲染的动态性。

关键词

Perlin 噪声；水面；OpenGL；实时渲染；反射；

目录

图目录

第1章 引言

在三维游戏和计算机图形学领域，无论各种电影还是各种游戏,凡是需要用到计算机图形学和虚拟现实的地方,几乎都会存在大规模水面的绘制,比如:河流、湖泊等。因此,大规模水面的实时绘制是十分重要[1]。水面的形成受到自然界许多因素的影响，呈现出丰富、多变的水面场景。进行水面模拟主要解决的问题有两个：一是要求真实感，二是尽量降低计算量以实现真实感的实时绘制要求[6]。

第2章 相关工作介绍

在现在常见的实践项目中，水面绘制大致可以被分为以下四种方法。第一种是贴图置换技术，这是最简单的水面模拟方法。实现思路很简单，通过将一张纹理图像进行平移或置换，可以模拟出简单的水波效果。这种方法实现容易，计算量低，但效果较为单一，无法表现复杂的水波动态；第二种是网格绘制的方法，这主要通过物理模拟（例如波动方程）来计算网格顶点的位置，从而实现动态水面效果。这种方法需要对网格顶点进行实时计算，以模拟出波浪的动态效果。这样的方法主要基于外观来模拟流体,并不是按照水在真实世界的物理状态来模拟,但是,这样可以满足用户对实时性渲染的需求[3]。例如Peachey[4]采用正弦波和余弦波叠加的方法模拟波浪的波形轮廓；第三种方法主要采用动态凹凸纹理映射的方法，动态凹凸纹理映射主要借助法线贴图来实现，通过调整表面法线来模拟水波效果。这种方法和上面提到的波动方程模拟比起来，计算量小的很多，这种方法适用于需要实时渲染的场景，能够在较低的计算成本下实现逼真的水面效果，但是相比之下可操作性和真实性会有一定损失；第四种方法是进行真实的粒子模拟，部分追求质感的大型游戏采用了这种方式，通过大量粒子的运动模拟流体行为，实现水面效果。这是一种基于纯物理方法的绘制,即解 Navier-stokes(纳维斯托克斯)方程,但由于 Navierstokes 方程十分复杂,到现今为止,在计算机图形学领域内,也无人能够完整地解出 Navier-stokes 方程。现在各种解 Navier-stokes 方程的方法一般都是对 Navier-stokes 方程本身做了较大的简化。即便如此,用 Navier-stokes 方程实时绘制大规模的水面场景也是十分复杂和效率低下的[2]。这种方法具有高度的真实感，但计算量较大，适用于需要高度真实感的场景，如电影特效和高端游戏。

本文在参考以上思路的基础上，主要采用第二种方法，也就是通过波动方程结合噪声纹理绘制的方式，来尝试对真实的水面进行模拟。

第3章 方法描述

在我们本次实验中，我们结合上文提到的网格绘制法来进行真实水面的模拟工作。

我们在模拟过程中，涉及到了波动方程、Perlin[5]噪声和菲涅尔效应。大致来看，为了模拟真实的水面效果，本实验采用了基于网格绘制的方法。首先，我们在主程序中初始化一系列由顶点和索引组成的三角形网格，在三角形网格中，我们计算每个点和对应纹理的匹配关系，接着在顶点着色器代码中，我们撰写波动方程，用于计算每个顶点的动态位移。接下来，在片段着色器中使用 Perlin 噪声生成细节纹理，并结合法线贴图计算动态法线，以增加水波的细节，在最后着色部分，我们计算光线在水面上的反射和折射方向，并结合菲涅尔效应生成逼真的光学效果，通过计算光线在水面上的反射和折射，模拟出水面的透明度和光线变化。下面我们一一来介绍各个部分：

3.1 水面网格模拟

在程序的主循环代码中，我们初始化对应的网格，接着，再一一设置网格的对应的水面的大小，纹理，形状，发现方向等等。在具体实践中，我们先初始化所有的法线方向朝上，接着在后续的实验中，我们可以通过进一步改变法线的方向来帮助我们进行进一步模拟。

3.2 波动方程设计

由于真实的水波运动满足复杂的偏微分方程，求解这些方程非常困难。我们采用简化的数学模型和近似方法来模拟水波，为了能够模拟一个在二维平面（X-Z）上的波动，我们可以设计一个简单的波动方程：

$$\text{wave} = \sin(x \cdot \text{waveFrequency} + t \cdot \text{waveSpeed}) \cdot \text{waveHeight} + \cos(z \cdot \text{waveFrequency} + t \cdot \text{waveSpeed}) \cdot \text{waveHeight}$$

在这个方程中，我们通过正弦和余弦函数模拟波浪效果，并结合时间变量实现动态变化。

3.3 Perlin 噪声

在实践中我们需要噪声来帮助进行纹理的模拟。

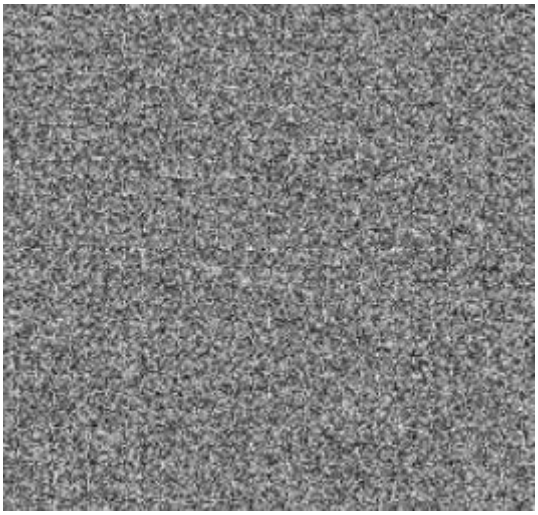

图 3.1 随机噪声

上图 3.1 为生成的随机噪声，我们可以看到用随机函数生成的噪声纹理太过嘈杂，不适合用于直接模拟水面纹理。因此，用这种噪声来模拟上述噪声难度太大了，因此我们需要一种新的水面模拟方法。

Ken Perlin 于 1985 年提出了一种自然噪声生成算法，即 Perlin 噪声，被广泛地应用于计算机图形学。为了增加水波的细节和真实感，我们使用 Perlin 噪声生成细节纹理。生成 Perlin 噪声首先需要伪随机的噪声函数和合适的插值函数，伪随机噪声函数，即相同的输入值，经过两次调用噪声函数应得到同样的输出结果。其核心思想是通过多个频率和振幅的噪声函数叠加，生成多层次的噪声效果。

Perlin 噪声的公式如下：

$$\mathrm{PerlinNoise}(x)=\sum_{i=0}^{n}\left(\frac{\mathrm{persistence}^{i}}{2^{i}}\right)\mathrm{Noise}\left(2^{i}\cdot x\right)$$

其中，persistence 是控制每次迭代影响力的衰减因子，i 是当前迭代的次数。

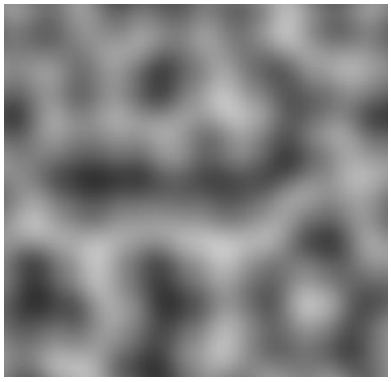

图 3.2 柏林噪声

图 3.2 为我们生成的二维柏林噪声图，由图可以发现，二维的柏林噪声相比于随机噪声,它在大尺度上表现出平滑的过渡，而在小尺度上则具有随机性。在实验中，我们将 Perlin 噪声生成的噪点图叠加到水面波动的基本形状上，来模拟真实的水面纹理细节。

3.4 菲涅尔效应

我们在进行水面渲染的时候，还需要考虑到菲涅尔效应。在真实世界中，除了金属之外，其它物质均有不同程度的菲涅尔效应。

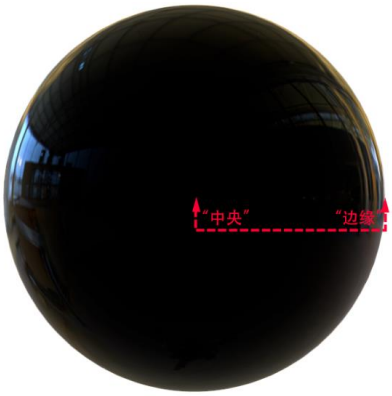

图 3.3 菲涅尔效应示意图

简单的讲，就是视线垂直于表面时，反射较弱，而当视线非垂直表面时，夹角越小，反射越明显。如图 3.3，圆球中心的反射较弱，靠近边缘较强，且过度关系被折射率影响。如果不使用菲涅尔效应的话，则反射是不考虑视点与表面之间的角度的。在水面渲染中，菲涅尔效应描述了光在介质界面上的反射和折射强度随入射角度变化的现象，在水面渲染中，我们可以通过调整计算光的反射和折射比例，从而生成更为真实的视觉效果。

菲涅尔公式如下：

$$F(\theta) = F_0 + (1 - F_0) \cdot (1 - \cos(\theta))$$

其中，$F(\theta)$是入射角为θ时的反射率，F_0是垂直入射时的反射率。

通过使用菲涅尔公式，我们可以计算出水面上每个像素点的反射和折射比例，从而生成逼真的水面光学效果。在代码实现中，这个公式用于结合反射颜色和折射颜色，最终生成水面的渲染结果。

第4章 实验设置与结果分析

4.1 实验环境

本次实验在普通 PC 机上实现的，基本软件实现平台为：Windows 11，实验环境包括：Visual studio Code，OpenGL3.3。显卡为 AMD Radeon RX 5700。

4.2 实验渲染

我们在原来的“天体行星漫步模拟观察系统”的基础上进行修改来进行展示，我们首先更改了天空盒来将模拟地点变为地球，接着我们通过绘制一张纹图案，并在该图案上进一步进行实验仿真。

为了给我的水面提供一个颜色参考，我切换了使用到的天空盒，将发生的地点环境从宇宙切换到地。我们使用 Learn OpenGl 中提供的天空盒提供的框架，将水面固定在天空盒中，并将视点设置在水面上。将水面固定在天空盒中，并将视点设置在水面上。接着，为了让水面可以随着我们摄像机的视点和角度对应变化，我们将对应的变量指标设置为 OpenGl 中的视点光线反射点，从而能保证水面的任意一点都可以构成对应的反射向量，接着初始化法线向量之后，在进行对应的反射计算。接着在生成纹理的时候，我们需要获得基础纹理，附加纹理，接着将不同频率和振幅的噪声叠加起来得到新的纹理，接着通过在循环中采样的方式，得到动态的水波纹理效果。

4.3 实验结果分析

我们按照上面的实验进行设置，得到了如下结果：

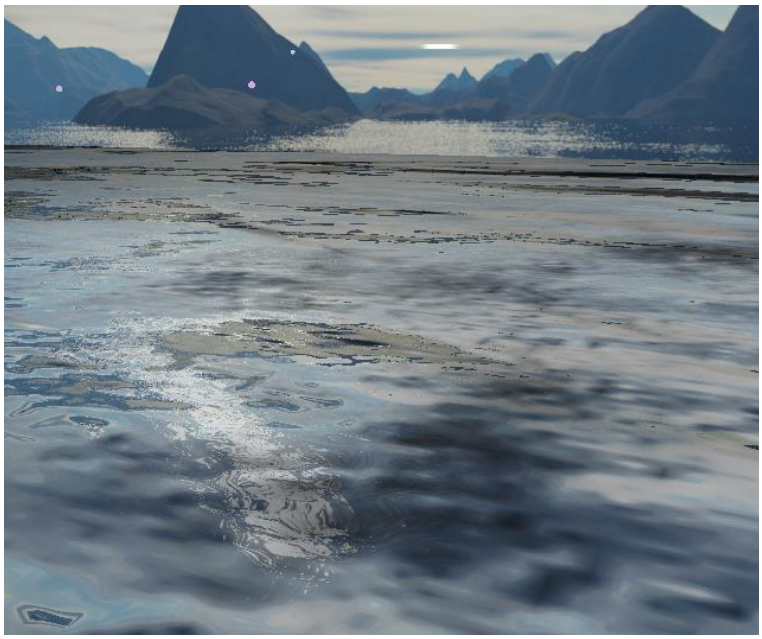

图 4.1 实验结果图

如图 4.1，水面的颜色与天空盒的环境颜色接近，比较好的实现了反射的效果，确保了一定的真实性；当斜看的时候，水下物体显得些许扭曲模糊，仔细观察可以发现水面是凹凸不平的，图 4.1 展示了动态波浪效果，水面看起来更加生动。同时，我们可以看到太阳光照的反射光，通过模拟光源的位置和强度，可以实现我们实现的水面的漫反射和镜面高光效果，其中光照方向我设置的和天空盒贴图的光照方向一致。

总体来说，本实验实现了水面的反射，波动，与不同光照下的水面模拟效果。

第5章 结论及展望

在未来的工作中，我希望可以在我们的行星模拟系统的基础上结合我学习到的水面模拟，进一步优化我们的银河设计。

致谢

感谢老师给我们减少作业。

参考文献：

[1] YAO Zhiqiang, LI Jiansheng, CHEN Jingwei, et al.Realization of flying simulation based on 3D modeling software and OpenGL[J].Science of Surveying and Mapping, 2008 (3) :1623-1627 (in Chinese) .[姚志强, 李建胜, 陈景伟, 等.基于 3DSMAX 和 OpenGL 的飞行仿真的实现一种基于 GPU 的实时水波模拟方法[J].测绘科学, 2008 (3) :1623-1627.]

[2] LIU Xiaoling, YANG Hongyu, GUO Huqi.Real-time simulation of rain and snow in large-scale scene based on GPU particle system[J].Computer Engineering and Design, 2012, 33 (6) :2399-2341 (in Chinese) .[刘小玲, 杨红雨, 郭虎奇.基于 GPU 粒子系统的大规模雨雪场景实时模拟[J].计算机工程与设计, 2012, 33 (6) :2399-2341.]

[3] HU Zhenhua.Rain rendering based on programmable GPU[D].Hangzhou:Zhejiang University, 2010 (in Chinese) .[胡振华, 基于可编程 GPU 的雨天特绘制[D].杭州:浙江大学硕士学位论文, 2010.]

[4] Peachey D R. Modeling waves and surfaces. Computer Graphics, 1986,20(4): 65-74

[5] K Perlin．An image synthesizer．Computer Graphics，1985，19(3)：287～296

[6] 梁梦洁. 基于 OpenGL 的真实感三维海面模拟[D].西安电子科技大学,2013.