

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)
21301025	张佳讯	8	23	17	36	84

缺少对现有工作的梳理，参考文献较少



计算机图形学期末（论文）

基于 Gerstner 波的水面模拟

Water surface simulation based on Gerstner waves

学 院： 软件学院

专 业： 软件工程

学生姓名： 张佳讯

学 号： 21301025

指导教师：

北京交通大学

2024 年 6 月

## 学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

指导教师签名：

签字日期：      年    月    日

签字日期：      年    月    日

## 摘要

在游戏中，对于水面的模拟都是我一直比较好奇的点。所以基于 Gerstner 波实现了水面波动模拟。Gerstner 波可以生成更加尖锐的波峰和更广阔的波谷，比正弦波方法能够更真实地模拟粗犷的波浪。本文首先介绍了相关工作，包括 OpenGL、正弦波方法、Gerstner 波公式、法线计算方法和菲涅尔反射公式。然后解释了实现的具体方法，基于高度场的水面模拟思路以及顶点着色器和片段着色器的编写思路。最后，本文设计了实验，分析了波的数量和网格宽度对渲染效果和 FPS 的影响。结果表明，增加波的数量会使水面波动更复杂，FPS 会有所降低；增加网格宽度会扩大水面范围，FPS 会较大幅度降低。总的说，该方法有效实现了水面波动模拟。

**关键词：** OpenGL；水面模拟；Gerstner 波

## 目 录

摘要 .....	II
目 录 .....	III
1 引言 .....	1
2 相关工作 .....	2
2.1 OpenGL .....	2
2.2 正弦波 .....	2
2.3 GERSTNER 波 .....	3
2.4 法线计算 .....	3
2.5 菲涅尔（FRESNEL）反射 .....	4
3 方法描述 .....	5
3.1 基于高度场的水面模拟 .....	5
3.2 顶点着色器 .....	5
3.3 片段着色器 .....	6
4 实验设计 .....	7
4.1 实验设置与评价指标 .....	7
4.2 实验结果 .....	7
5 结论 .....	10
参考文献 .....	11
致 谢 .....	12

## 1 引言

渲染三维场景时经常会遇到需要渲染各种水体的情况，比如湖泊、河流、海洋等。渲染水面的方法有很多，比如简单的若干正弦波叠加就可以实现。但是这种方法不够真实。在流体力学中，有一种波称作 **Gerstner** 波。它可以模拟粗犷的海洋，生成较尖的浪头和较宽的浪槽。本文将介绍通过 OpenGL 实现的 **Gerstner** 波模拟水面效果方法。

## 2 相关工作

水面模拟最主要的一点是水波的模拟，最简单的模拟高度的做法，是最简单的简谐函数。但是仅是正弦波并不足以满足需求，还可以对其进行扩展。Gerstner 波能提供更有效的波形，Gerstner 波的最大特性就是：它们通过将顶点移向每个波峰而形成更尖的波峰。

### 2.1 OpenGL

OpenGL 是用于渲染 2D、3D 矢量图形的跨语言、跨平台的应用程序编程接口，常用于 CAD、虚拟现实、科学可视化程序和电子游戏开发。OpenGL 规范严格规定了每个函数该如何执行，以及它们的输出值。至于内部具体每个函数是如何实现的，将由 OpenGL 库的开发者自行决定。因为 OpenGL 规范并没有规定实现的细节，具体的 OpenGL 库允许使用不同的实现，只要其功能和结果与规范相匹配。

### 2.2 正弦波

正弦波是最简单模拟高度的做法。通过多个正弦波的叠加可以使得波形更加复杂来模拟水波。

需要一组参数来定义每个波浪。

。

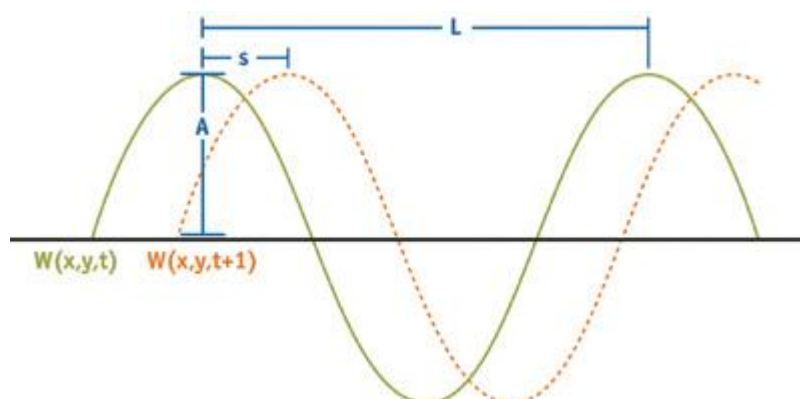


图 2-1 单波函数

波长（L）：世界空间中波之间的波峰到波峰距离。

振幅（A）：从水面到波峰的高度。

速度（S）：波峰每秒向前移动的距离。

方向（D）：垂直于波峰沿波前行进的水平矢量。

### 2.3 Gerstner 波

Gerstner 波函数为：

$$\mathbf{P}(x, y, t) = \begin{pmatrix} x + \sum (Q_i A_i \times \mathbf{D}_i \cdot x \times \cos(w_i \mathbf{D}_i \cdot (x, y) + \varphi_i t)) \\ y + \sum (Q_i A_i \times \mathbf{D}_i \cdot y \times \cos(w_i \mathbf{D}_i \cdot (x, y) + \varphi_i t)) \\ \sum (A_i \sin(w_i \mathbf{D}_i \cdot (x, y) + \varphi_i t)) \end{pmatrix}.$$

图 2-2 Gerstner 波函数

其中  $Q_i$  是控制波浪陡度的参数， $i$  是第  $i$  个波浪。如果  $Q_i$  是 0，则是一般的正弦波。当  $Q_i = 1/(w_i A_i)$  时，波峰最尖锐。

Gerstner 波可以和正弦波对比去理解。Gerstner 波的  $x$ 、 $z$  坐标向波峰靠拢， $y$  轴坐标做正弦波计算。所以看起来波峰更尖锐，波谷更广阔。从公式上看，全部正弦波叠加的公式为：

$$H(x, y, t) = \sum (A_i \times \sin(\mathbf{D}_i \cdot (x, y) \times w_i + t \times \varphi_i)),$$

图 2-3 正弦波叠加公式

Gerstner 波方程式为：

$$\mathbf{P}(x, y, t) = (x, y, H(x, y, t)).$$

图 2-4 Gerstner 波方程式

### 2.4 法线计算

法线的计算是计算每一个波对顶点位置的偏导数分量，再进行累加。计算公式：

$$\mathbf{N} = \begin{pmatrix} -\sum (\mathbf{D}_i \cdot x \times W A \times C0) \\ -\sum (\mathbf{D}_i \cdot y \times W A \times C0) \\ 1 - \sum (Q_i \times W A \times S0) \end{pmatrix},$$

图 2-5 法线计算公式

其中，

$$\begin{aligned}WA &= w_i \times A_i, \\S() &= \sin(w_i \times \mathbf{D}_i \cdot \mathbf{P} + \varphi_i t), \text{ and} \\C() &= \cos(w_i \times \mathbf{D}_i \cdot \mathbf{P} + \varphi_i t).\end{aligned}$$

图 2-6 法线计算公式参数

## 2.5 菲涅尔（Fresnel）反射

摄像机看到的水面上一点的颜色是反射环境的颜色和折射出来的水下颜色的加权之和。权值与入射角有关。如果你站在湖边，低头看脚下的水，你会发现水是透明的，反射不是特别强烈；如果你看远处的湖面，你会发现水并不是透明的，但反射非常强烈。这就是“菲涅尔效应”。菲涅尔方程描述的是在不同的表面角下表面所反射的光线所占的比率。入射光反射的比例  $R$ ：

$$R = \frac{1}{2} \left\{ \frac{\sin^2(\theta_t - \theta_i)}{\sin^2(\theta_t + \theta_i)} + \frac{\tan^2(\theta_t - \theta_i)}{\tan^2(\theta_t + \theta_i)} \right\}$$

图 2-7 菲涅尔公式

所以，计算水面颜色的简单公式为：

$$Color = R * ColorIncident + (1 - R) * ColorDeepWater$$



### 3 方法描述

使用 Gerstner 波来模拟水面波动，思路是根据公式计算顶点的更新位置和法线。先是随机生成波的参数，再根据波的参数逐个迭代计算每个顶点的位置。同时计算每个顶点的法线。将计算完的顶点位置和法线变换到世界坐标系，通过菲涅尔反射计算水面颜色再传给着色器进行渲染。

#### 3.1 基于高度场的水面模拟

Jerry Tessendorf[1]等人提出基于高度场  $y=h(x, z)$  的方法模拟水面。想要让水面随时间变化，公式变成： $y=h(x, z, t)$ 。

想要存储这个高度场，需要用离散的方法。可以用三维顶点坐标集合和顶点之间的连接关系集合来完成。即一个顶点和相邻的八个顶点之间构成四个正方形，每个正方形由两个三角形组成。

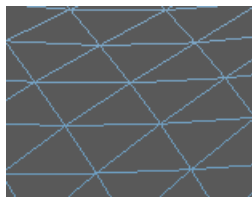


图 3-1 顶点连接关系

所以，只要考虑顶点集合，根据  $x, z$  求出  $y$  即可。

设置网格宽度为  $width$ ，则顶点数为  $width*width$ ，索引数为  $2*3*(width-1)*(width-1)$ 。

#### 3.2 顶点着色器

定义 Gerstner 波结构体来表示波的参数，包括波峰的锐度、波的最大高度、方向、波长、波速。之后生成随机参数传入着色器。

着色器根据 Gerstner 波函数更新顶点位置和法线。使用 Gerstner 波函数需要先算出频率和相位。频率的计算公式为

$$w = \sqrt{2 * \pi * g / l}$$

其中  $\pi$  是  $\pi$ ， $g$  是重力加速度， $l$  是波长。相位的计算公式为

$$fi = 2 * \pi * s / l$$

其中  $s$  为波速。

之后根据公式计算顶点位置和法线向量。最后使用模型矩阵将法线转换到世界坐标系，顶点位置转换到裁剪坐标系。

### 3.3 片段着色器

根据观察者的位置、光线的方向和法线向量计算入射角、反射角和折射角。再根据菲涅尔反射公式去计算入射光反射的比例  $R$ ，再由反射颜色和透射颜色混合得到最终的颜色。

## 4 实验设计

主要对波的数量，以及网格宽度对渲染效果和 FPS 的影响进行讨论和研究。

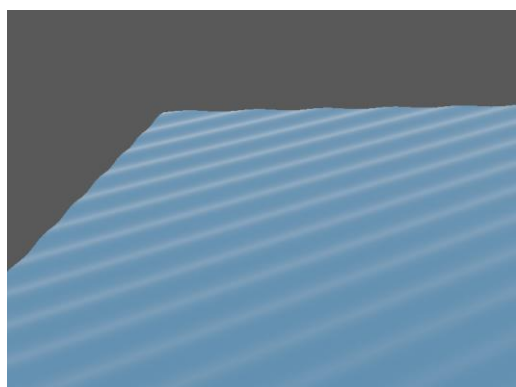
### 4.1 实验设置与评价指标

实验环境为 Windows 10 操作系统计算机，配备 NVIDIA GeForce RTX 3060 Laptop GPU 显卡，使用 OpenGL 4.6.0 NVIDIA 536.40、Vulkan 1.3.241、GLFW 3.2.1 和 Zlib 1.3.1 进行开发测试。

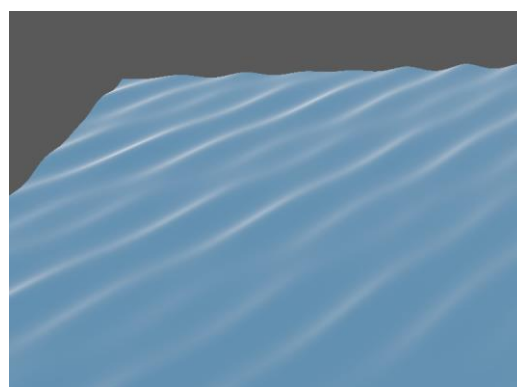
评价指标主要是每秒帧率（FPS）。

### 4.2 实验结果

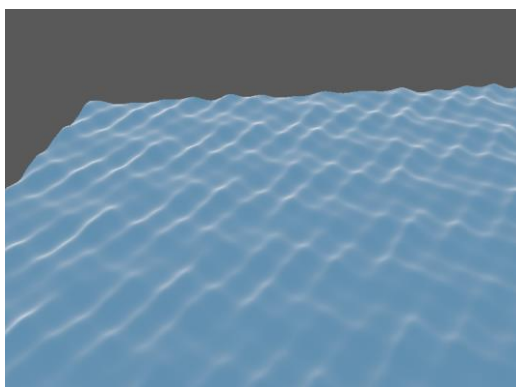
首先是在网格宽度为 200 的情况下变化波的数量进行效果和 FPS 比较。如下图



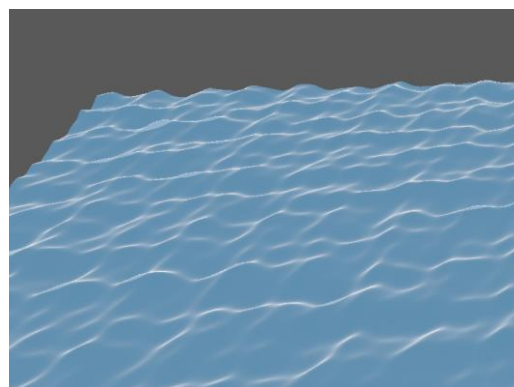
(a) 1 个波



(b) 3 个波



(c) 6 个波



(d) 10 个波

图 4-1 渲染效果图

表 4-1 波数量实验

WaveCount	FPS
1	161
3	2702
6	2691
10	2530

可以看到随着波数量的增加，水面波动变得更加复杂，FPS 也随之变小。

接下来是在波数量为 10 的情况下变化网格宽度进行效果和 FPS 比较。如下图

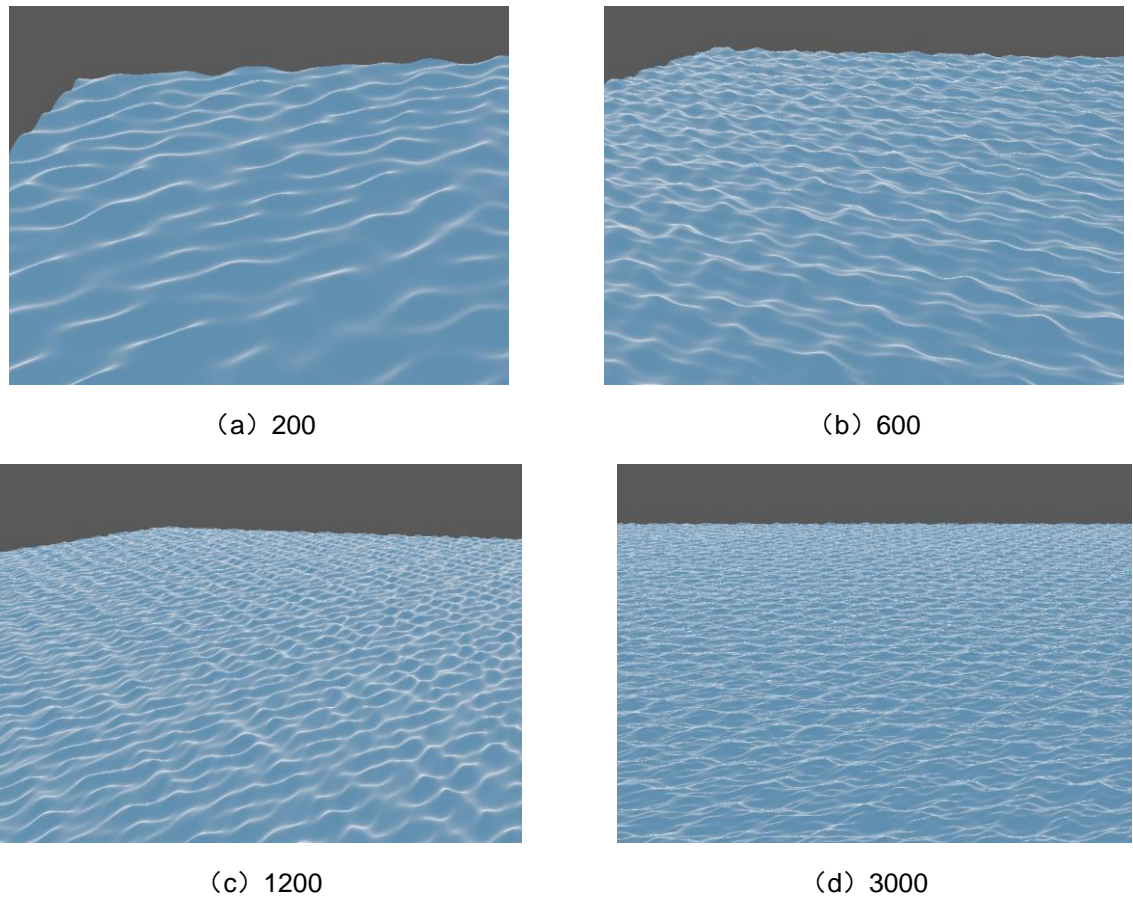


图 4-2 效果图

表 4-2 网格宽度实验

Width	FPS
200	2579
600	1008
1200	275
3000	49

可以看到随着网格宽度的增加，水面确实扩大了很多，FPS 变小的幅度也很大。

## 5 结论

通过 OpenGL 实现了基于 Gerstner 波的水面模拟。通过实验可以看出这种水面模拟的方法简单有效，并且看上去很逼真，即使水面很大，帧率也可以接受，是一种实现性价比很高的方法。但是还是不够真实，想要更真实一点，需要去实现基于 FFT 算法的水面模拟。

（本文实现代码见：<https://github.com/adaelon/water>）

## 参考文献

- [1] Tessendorf J . Simulating Ocean Water[J]. simulating nature realistic & interactive techniques siggraph, 2001.

## 致 谢

感谢陪我的朋友，图形学还是很有趣的！



