Ocean Simulation

CSCI711 Final Project Report

Noya Cai, Tsingtao Zhang

# Problem/Project Description

*Give a brief overview of the project. What were your objectives for the project? What area of the image synthesis pipeline is emphasized?*

## Overview

This project aims to create a simulation of the ocean that is close to shore. The final stage of the project should include correct simulation of waves and caustics at the bottom of the ocean. Objects below the water as well as floating on the surface of the ocean should be correctly rendered in real-time according to time and light changing conditions.

## Pipeline

We use OpenGL and the ray tracer that we wrote in programming assignments as our rendering pipeline. The structure of our rendering method uses vertex shader and fragment shader to handle all the ray tracing parts, including the definition of objects and all the ray-object interactions.

# Approach

*A brief, high-level description of the project or system. If the project is an application, then a brief overview of inputs, outputs, and system controls should be described here. Detailed descriptions will come later in the user documentation in the Results section. If the project is a rendering, this section should include a description of the algorithms/techniques used in to implement the project. If the project involved writing programmable shaders, then a high-level description of the shader(s) (input, output, functionality) should be included.*

## Algorithms/Techniques

### Waves

Waves are mainly controlled by a procedure texture that calculates the normal of the wave surface. In this project, we use Gerstner waves to simulate a relatively quiet ocean surface. <https://www.tandfonline.com/doi/pdf/10.2991/jnmp.2008.15.S2.7>

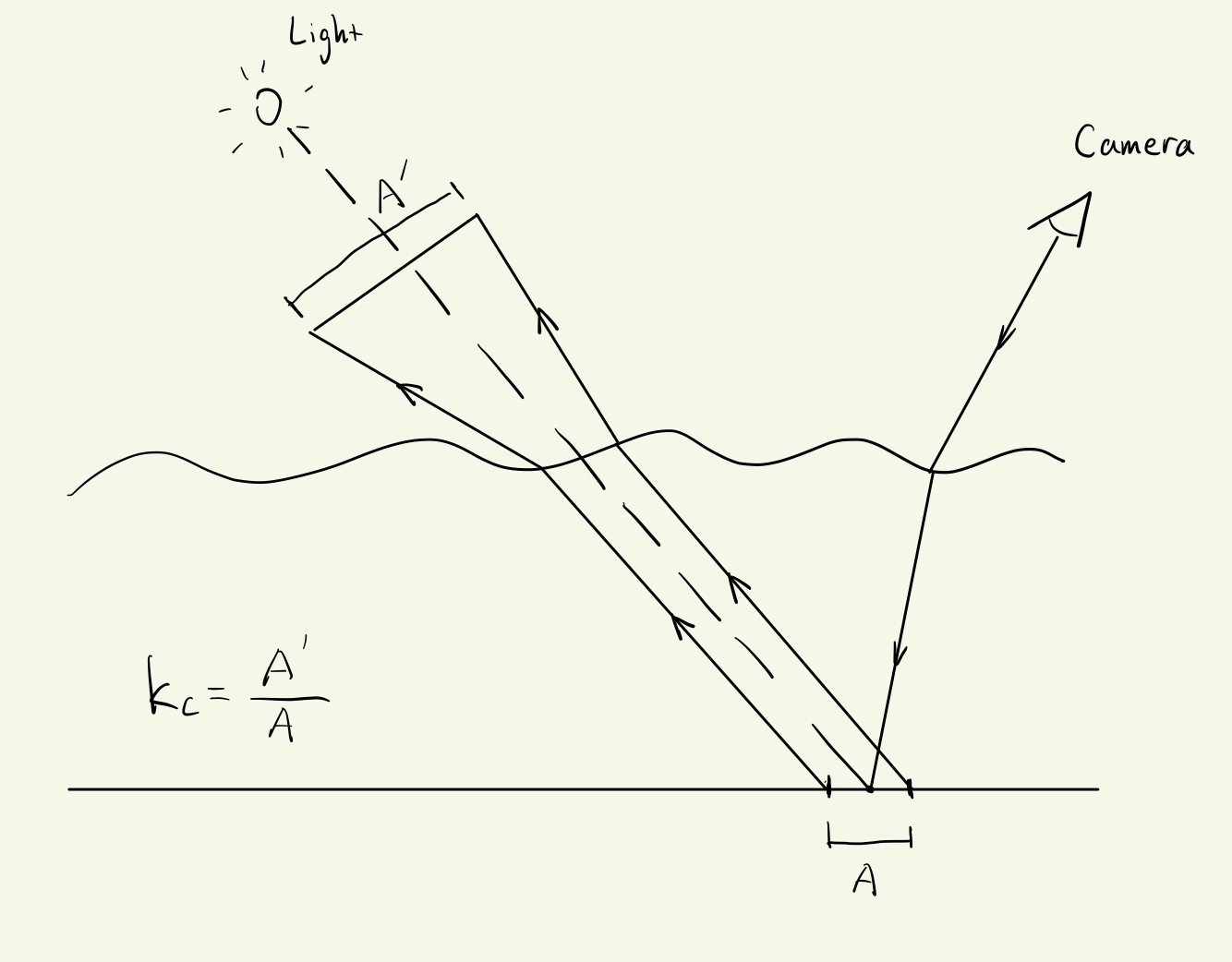
<https://developer.nvidia.com/gpugems/gpugems/part-i-natural-effects/chapter-1-effective-water-simulation-physical-models>

By adding a series of Gertner waves in different directions, we create a complex enough ocean surface. With the matching moving speed and the wave’s wavelength, the Gerstner wave results in great detail.

### Caustics

Inspired by [Evan Wallace](https://madebyevan.com/)’s [Rendering Realtime Caustics in WebGL](https://medium.com/@evanwallace/rendering-realtime-caustics-in-webgl-2a99a29a0b2c), the caustics can be referred to as the radiosity energy scattered in different densities, the curvy water surface works like a magnifying/minifying glass. Thus this radiosity energy can be represented as the ratio of two triangle’s area, one is above the water, the other is projected by refraction under the water.

And the area’s ratio can be attributed to the specular and diffuse coefficient (we’re using Phong shading), thus affecting the shading. See the image below showing this process in 2D.



## Shaders

We use vertex and fragment shaders in this program. No need to have any input other than what’s been set in OpenGL. However, if you want to tweak the scene, you can change the time, the object size and position, the wave’s aptitude, wavelength and moving speed, and the caustic definition and intensity. More details in technical documentation.

# Results

*The 3D Rendering and/or screen shots of the application. If the project is an application or a programmable shader, user documentation (see below) should be included here.*

[CSCI711Final.mp4](https://drive.google.com/file/d/1L65ZH91Dx2d1JTCu2CHZgv1UXK4tcnPv/view?usp=drive_link)

See the [user manual](https://docs.google.com/document/d/1eTO2nP2oxunjDa-riZYAem6zC_PRaaQoCJL8jadr9zQ/edit) here.

# Future Enhancements

*If you (or your team) were to continue working on this project, what would you like to add or fix? What things should be modified or added to improve the functionality? What different approach(es) should be taken to the different parts of the project?*

1. We can have Perlin noise affecting the wave layout, so instead of a uniform repeated pattern, the wave can have continuous randomness.
2. We can use the Monte-Carlo method to create a more realistic caustic effect. Right now all the triangles used to calculate the area ratio are built in the same way, we can make it vary in size and position, and based on all samples have more natural caustics.
3. We can cut off redundant code and optimize the RAM requirement.
4. Apply simulated buoyancy to the sphere.
5. Add wave spray to the waves and around objects that the wave is colliding with.