CS519 Shaders Final Project

A simple Water simulation

Li Li

1 source files

There are:

Source file	what are they doing
mainwindow.cpp and mainwindow.h	activate the program from user
glwidget.cpp and glwidget.h	handle creating windows in a certain OS
camera.cpp and camera.h	do stuff to manipulate the camera
waterengine.cpp and waterengine.h	where truly the water creating and rendering happened
vector.h	telling how to do vector operation

2 main steps

- **Reading** My program is based on *GPU Gems: Chapter 1. Effective Water Simulation from Physical Models.* Most of it is talking at water simulation.
- **Single Wave Function** As shown in figure1. We calculate Wavelength (L) ,Amplitude (A), Speed (S) and Direction (D). In my implementation, D is random, A is set as 0.03f. L is from 0.3 to 0.8. Speed calculate as $S = 0.05 \times \sqrt[2]{\pi/L}$. And last part steepness is $Steepness = 5.0 \times (random\text{-}a\text{-}float \times 2.0 + 1.0)$.
- **Unit Normal Map** This is what I want to calculate in first pass program, an than apply them to second pass-program to the geometry to do a local bump mapping. See figure2 for whole flow chart.
- **In Waterunit frag shader**, base on the equation (figure3).I calculated A, ω ,ϕ , D ,k to get the Normal of a certain point. Then it emits this normals to a normal map. Each one is compute comes from 50 waves.
- **Compute Gerstner Waves vertex in waterrender vertex shader**, these are similar but only comes from 6 waves. See figure4,5,6 for the equation. Than it output the final geometry light vector to a vertex, and eye vertex to the vertex, as well as its texture parameter to following fragment shader.
- **Final decision to the color of water: mixing specular and fresnel effect** The shader grab N from normal map that pass in as texture. And than obtain specular vector. For fresnel, it blend oceanbue and skyblue, and it is base on the view vector to it.$fresnel = R_0 + (1.0 - R_0) * 1.0 - -normalize(viewv) \cdot N^{5.0}$

3 some results

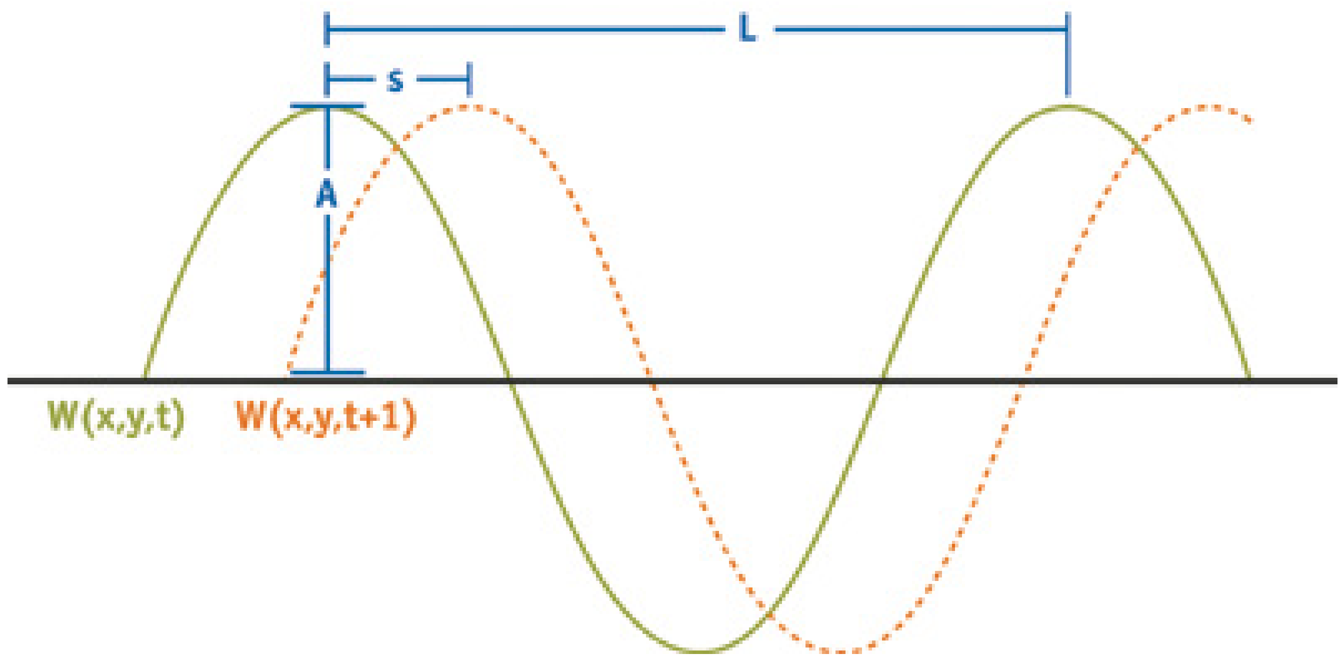

Figure 1: A single wave

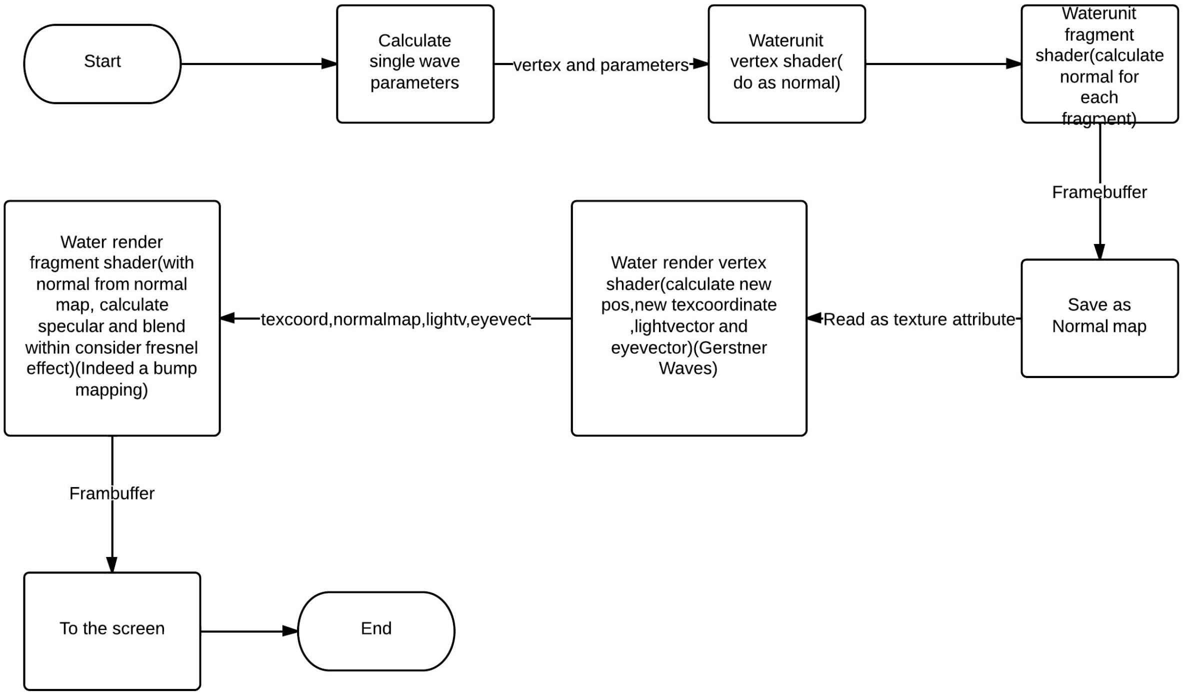

Figure 2: A flow chart

$$\frac{\partial}{\partial x}\left(W_i\left(x,y,t\right)\right) = k \times \mathbf{D}_i.x \times w_i \times A_i \times \left(\frac{\sin\left(\mathbf{D}_i \cdot (x,y) \times w_i + t \times \varphi_i\right) + 1}{2}\right)^{k-1}$$
$$\times \cos\left(\mathbf{D}_i \cdot (x,y) \times w_i + t \times \varphi_i\right).$$

Figure 3: normal equation

$$\mathbf{B} = \begin{pmatrix} 1 - \sum\left(Q_i \times \mathbf{D}_i.x^2 \times WA \times S()\right), \\ -\sum\left(Q_i \times \mathbf{D}_i.x \times \mathbf{D}_i.y \times WA \times S()\right), \\ \sum\left(\mathbf{D}_i.x \times WA \times C()\right) \end{pmatrix},$$

Figure 4: bi-normal equation Gerstner Waves

$$\mathbf{T} = \begin{pmatrix} -\sum\left(Q_i \times \mathbf{D}_i.x \times \mathbf{D}_i.y \times WA \times S()\right), \\ 1 - \sum\left(Q_i \times \mathbf{D}_i.y^2 \times WA \times S()\right), \\ \sum\left(\mathbf{D}_i.y \times WA \times C()\right) \end{pmatrix},$$

Figure 5: tangent equation Gerstner Waves

$$\mathbf{N} = \begin{pmatrix} -\sum\left(\mathbf{D}_i.x \times WA \times C()\right), \\ -\sum\left(\mathbf{D}_i.y \times WA \times C()\right), \\ 1 - \sum\left(Q_i \times WA \times S()\right) \end{pmatrix},$$

Figure 6: normal equation Gerstner Waves

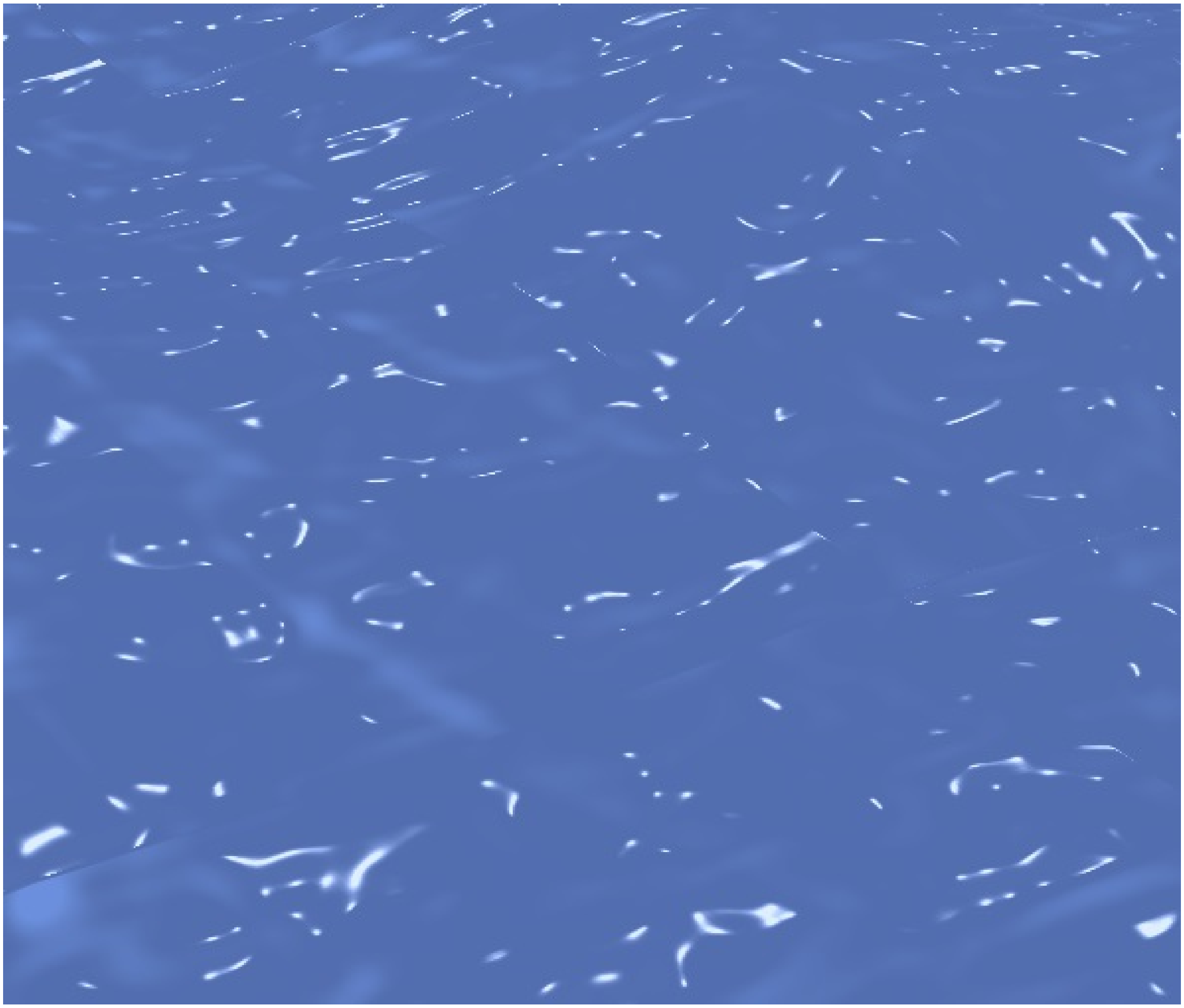

Figure 7: this is to show how a unit water looks like, not what waterunit.frag gives out(that is normal maps nor a frame buffer)

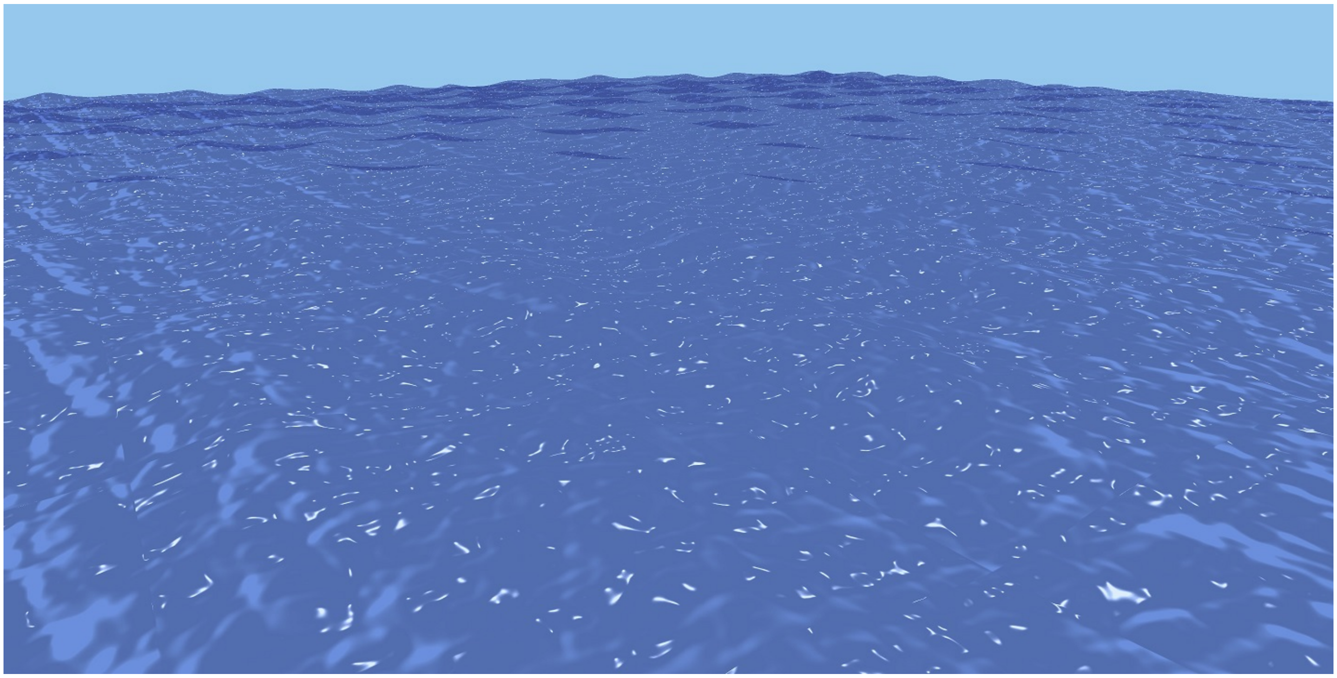

Figure 8: from a far pos look at the water

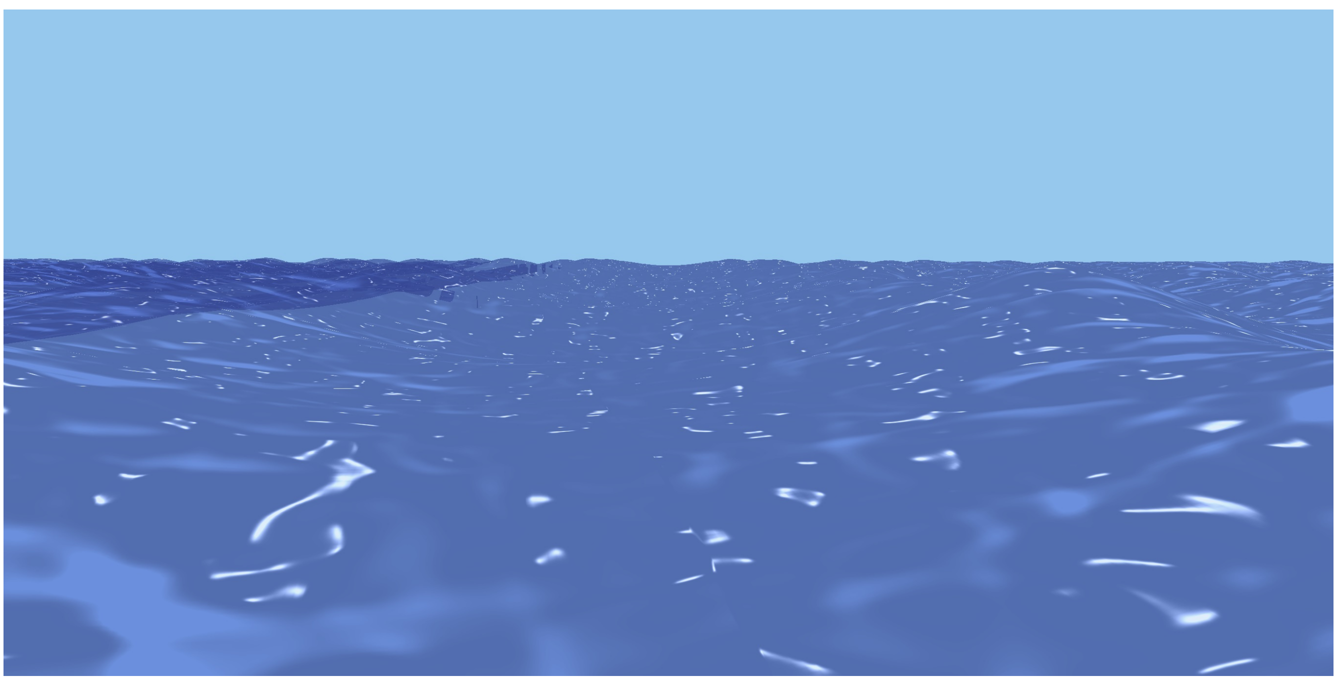

Figure 9: from a close pos look at the water

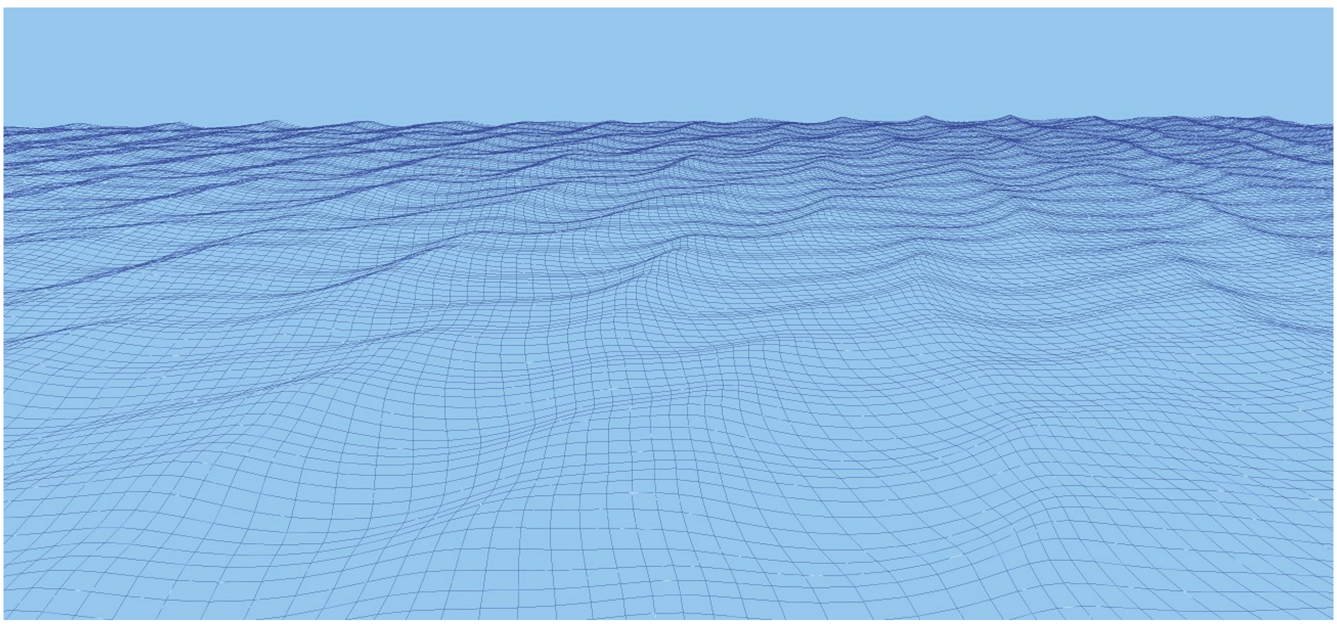

Figure 10: show how the geometry of the water looks like