

Computer Graphics

Ocean rendering

資工大一 塗季芸 B08902071

資工大一 戴可威 B08902080

1. Abstract

水是個高深莫測的物質，除了固態、液態、氣態上的外型變化外，更有折射、反射等特性，在 **computer graphics** 領域中，水的模擬是不斷被討論的問題，從電影鐵達尼號的海面到海底總動員中的海底世界，我們可以看見模擬技術不斷演進，能夠創造更為細緻、逼真的畫面。

關於水的模擬有許多種，有水滴、水柱、靜態水面、水底光線等，其中我們對有波浪的海面感到特別好奇，想了解其背後所用的技術，加上考量我們現在大一，尚未具備機器學習、硬體感測等能力，因此希望應用這學期在 **ICG** 課程中所學，結合 **homework1** 中使用的 **shaders**，針對海面及波浪，嘗試找出效果比較好的模擬方法。

本報告將解釋我們尋找方法及嘗試的過程，期間曾遇到的困難與解決方式，並將著重於討論波浪的水面高度模擬及做成動畫的方式，這個 **project** 是以 **WebGL** 實作完成。

2. Method

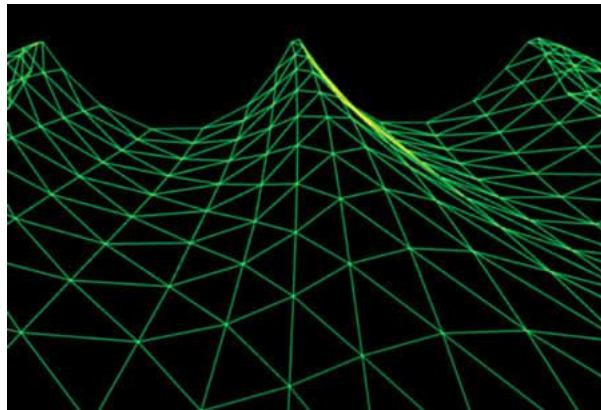
2.1 wave model

在閱讀相關資料時，我們發現現行模擬海面主要的 **wave model** 有兩種，第一個是根據 **Tessendorf** 所發表的論文，利用白噪音及 **Fast Fourier Transform (FFT)** 計算海面的高度及法向量，此方法雖然

能夠達到較好的視覺效果，實作上卻比較困難，運算量也相當龐大，第二個是 **Gerstner wave**，此方式能夠排除上述困難，也能套用在更多樣的水面型態，因此我們決定在這次 project 根據 Gerstner model 來實作。

2.2 Gerstner Wave

最簡單的波是 sin 波，但單純 sin 波通常無法模擬出海面的不規則性，且 sin 波曲線圓滑，但海波的波峰其實會略為陡峭（如下圖），sin 波在此時就不能模擬真實海面的情景。Gerstner Wave 正是產生較陡峭波峰的算法，其效率高也不複雜。



Gerstner Wave 計算出的波的位置：

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

波的 Normal :

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

其中，

$$\mathbf{WA} = w_i \times A_i,$$

$$S0 = \sin(w_i \times \mathbf{D}_i \cdot \mathbf{P} + \varphi_i t), \text{ and}$$

$$C0 = \cos(w_i \times \mathbf{D}_i \cdot \mathbf{P} + \varphi_i t).$$

在真實世界中，海面也不會僅僅只有一種波構成，因此我們用很多 Gerstner Wave 來疊加出我們想要的波形（即算式中的 sigma）。參數的決定則使用下列方法：

- Q_i : steepness。決定一個 steepness Q ，則 $Q_i = Q / (w_i A_i * \text{numWaves})$
- L : 波長（用於計算其他參數）：決定一個 median wavelength 後，random 選擇介於 $[\text{median wavelength} / 2, \text{median wavelength} * 2]$ 間的數值
- w : 頻率。
$$w = \sqrt{g \times \frac{2\pi}{L}},$$
- A : 振幅。選擇一個 median amplitude，則 $A_i = \text{median amplitude} / \text{median wavelength} * w_i$

- **D**：方向。選擇一個方向和角度，在這個角度之間 **random** 取方向的值。

而在我們的實作中，我們使用 6 個 **Gerstner wave** 疊加出我們的海面。

2.3 Real time?

我們曾在網路上看到一個 **real time** 的水面模擬，可以讓使用者創造漣漪及波浪，覺得這樣的效果相當好，原本希望能夠做出類似的 **interactive interface**，然而跟助教討論後，發現以我們的模擬方法，很難在 **real time** 下改動 **model** 的 **vertex** 跟 **normal**。於是改成事先做好 1000 個不同時間下的 **model**，在 **webgl** 的 **animate function** 中每一個 **frame** 都換成下一個時間點的 **model**，並且循環播放，創造波浪移動的效果。

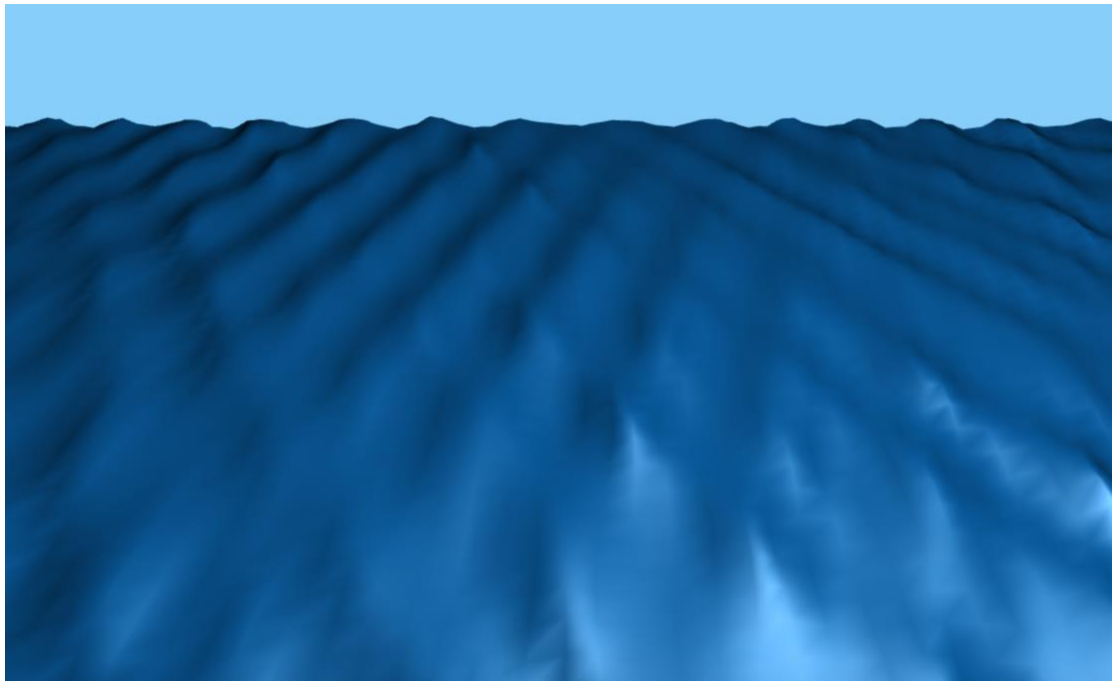
3 Conclusion

在這份專題中，我們了解幾個海面模擬常用的 **model**，選擇運用 **Gerstner wave model**，疊加多個波來計算波浪位置，並且以不同 **frame** 套用不同 **model** 的方式形成連續動態的動畫，在 **WebGL** 實作出海面的模擬。

4 Future works

我們希望未來能嘗試不同的 **wave model**，比較它們的波浪視覺效果，也希望能改善動畫實作方式，達到 **real time**。另外我們想更進一步研究一些特殊效果，如：風吹、漣漪，嘗試將其加入目前的模型中。

5 Result



6 Reference

Jerry Tessendorf, 'Simulating Ocean Water'

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.9102&rep=rep1&type=pdf>

Damien Hinsinger, Fabrice Neyret, Marie-Paule Cani, 'Interactive Animation of Ocean Waves'

<http://www-evasion.imag.fr/Publications/2002/HNC02/wavesSCA.pdf>

Evan Wallace's work, WebGL water

<http://madebyevan.com/webgl-water/>

Alan Fournier, William T Reeves, 'A Simple Model of Ocean Waves',
SIGGRAPH'86

<https://dl.acm.org/doi/epdf/10.1145/15886.15894>

Effective Water Simulation from Physical Models

https://developer.download.nvidia.com/books/HTML/gpugems/gpugems_ch01.html