

Water Simulation Implementation with Fast Fourier Transform

Vítor Hugo Magnus Oliveira
vhmoliveira@inf.ufrgs.br

Agosto 2024

1 Introdução

Este trabalho propõe a implementação de uma simulação realista da movimentação da água dos oceanos utilizando a Transformada Rápida de Fourier (FFT). Baseado nos conceitos introduzidos por Tessendorf em [2], este projeto explora como a matemática das funções espectrais pode ser aplicada para criar superfícies de água dinâmicas e visualmente convincentes no domínio da frequência.

Neste trabalho, discutimos as principais técnicas envolvidas na simulação, desde a geração do espectro inicial até a aplicação de efeitos visuais avançados, e apresentamos os resultados obtidos.

2 Descrição Técnica

A simulação foi implementada em C++ utilizando OpenGL, permitindo o desenvolvimento de programas otimizados que executam diretamente na GPU.

2.1 Funções Espectrais

As funções espectrais descrevem a distribuição da energia das ondas em função do vetor de onda \mathbf{k} . Essas funções são essenciais para modelar o comportamento das ondas de vento na superfície do mar, permitindo a simulação de oceanos realistas. A função espectral de Phillips foi a escolhida para a implementação, mais especificamente a sua versão modificada com a inclusão de um fator de atenuação adicional:

$$P(\mathbf{k}) = A \frac{\exp\left(-\frac{1}{(kL)^2}\right) \exp(-k^2 l^2)}{k^4} |\mathbf{k} \cdot \hat{\mathbf{w}}|^2,$$

onde:

- $\mathbf{k} = (k_x, k_y)$ é o vetor de onda, com $k = |\mathbf{k}| = \sqrt{k_x^2 + k_y^2}$, representando a magnitude do vetor de onda.
- A é uma constante de escala que depende das condições ambientais, como a intensidade e a direção do vento.
- L é o comprimento característico relacionado à maior onda possível para uma determinada velocidade do vento, V , onde $L = \frac{V^2}{g}$ e g é a aceleração devido à gravidade.
- $\hat{\mathbf{w}}$ é a direção do vento normalizada, indicando a orientação das ondas de maior energia.
- l é um fator de atenuação que controla a supressão de ondas pequenas.

O termo adicional é $\exp(-k^2 l^2)$, que reduz a energia das ondas para pequenos valores de k , evitando que ondas de comprimento muito longo dominem a simulação.

2.2 Gerando o Espectro Inicial

O espectro inicial representa uma distribuição de ondas que evoluirá ao longo do tempo para simular o comportamento dinâmico do oceano. Os coeficientes espectrais iniciais $\tilde{h}_0(\mathbf{k})$ são calculados com base no espectro $P(\mathbf{k})$ e na aplicação de números aleatórios:

$$\tilde{h}_0(\mathbf{k}) = \frac{1}{\sqrt{2}} (\xi_r + i\xi_i) \sqrt{P(\mathbf{k})},$$

onde:

- ξ_r e ξ_i são números aleatórios independentes, seguindo uma distribuição normal com média zero e variância de um. Estes representam as componentes real e imaginária do ruído gaussiano necessário para a simulação.

A geração do espectro inicial é realizada através de um *compute shader* escrito em GLSL. Este shader executa diretamente na GPU e calcula o espectro inicial para cada ponto no domínio da frequência.

2.3 Variação do Espectro com o Tempo

Para animar a superfície do oceano ao longo do tempo, o espectro de ondas é atualizado para cada vetor de onda \mathbf{k} . Essa evolução temporal é baseada na combinação das componentes de Fourier iniciais e suas complexas conjugadas, que são moduladas pela frequência angular das ondas:

$$\tilde{h}(\mathbf{k}, t) = \tilde{h}_0(\mathbf{k})e^{i\omega(\mathbf{k})t} + \tilde{h}_0^*(-\mathbf{k})e^{-i\omega(\mathbf{k})t},$$

onde:

- $\tilde{h}_0(\mathbf{k})$ é o espectro inicial das ondas.
- $\tilde{h}_0^*(-\mathbf{k})$ é o conjugado complexo do espectro inicial para o vetor de onda inverso.
- $\omega(\mathbf{k}) = \sqrt{gk}$ é a frequência angular da onda, com g representando a aceleração da gravidade e $k = |\mathbf{k}|$ sendo a magnitude do vetor de onda.

Para calcular a evolução temporal do espectro, também foi desenvolvido um *compute shader* em GLSL.

2.4 Cálculo das Normais no Domínio da Frequência

O cálculo dos vetores normais pode ser feito no domínio da frequência. Para isso, é preciso usar os componentes da derivada parcial da altura da superfície com relação às coordenadas espaciais x e y :

$$\frac{\partial h}{\partial x} = \sum_{\mathbf{k}} ik_x \tilde{h}(\mathbf{k}, t) e^{i\mathbf{k} \cdot \mathbf{x}},$$

$$\frac{\partial h}{\partial y} = \sum_{\mathbf{k}} ik_y \tilde{h}(\mathbf{k}, t) e^{i\mathbf{k} \cdot \mathbf{x}}.$$

No código, as normais no domínio da frequência são calculadas como:

$$\mathbf{n}_x = i\tilde{h}(\mathbf{k}, t) \cdot k_x,$$

$$\mathbf{n}_y = i\tilde{h}(\mathbf{k}, t) \cdot k_y,$$

onde i é a unidade imaginária, $\tilde{h}(\mathbf{k}, t)$ representa a altura espectral no tempo t , e k_x, k_y são os componentes do vetor de onda.

No domínio espacial, as normais da superfície são então dadas por:

$$\mathbf{n}(x, y, t) = \left(-\frac{\partial h}{\partial x}, -\frac{\partial h}{\partial y}, 1 \right).$$

Estas normais são utilizadas na simulação para calcular a direção de reflexão e refração da luz na superfície da água.

2.5 Cálculo de Choppiness (Agitação)

O efeito de "choppiness" ou agitação é uma técnica para adicionar realismo à simulação ao representar ondas mais agudas, que são comuns em mares mais turbulentos. Este efeito é obtido deslocando as coordenadas x e y dos vértices da superfície com base na inclinação das ondas:

$$x' = x - \lambda \frac{\partial h}{\partial x}, \quad y' = y - \lambda \frac{\partial h}{\partial y},$$

onde λ é um fator de intensidade que controla a agitação.

No código, os deslocamentos devido à agitação são calculados como:

$$Dz = -\frac{\mathbf{n}_y}{k} \cdot \text{choppiness},$$

$$Dx = -\frac{\mathbf{n}_x}{k} \cdot \text{choppiness},$$

onde \mathbf{n}_x e \mathbf{n}_y são as normais no domínio da frequência e $k = |\mathbf{k}|$ é o comprimento do vetor de onda. Este deslocamento é ajustado pela intensidade da agitação choppiness e aplicado para alterar as coordenadas dos vértices da superfície.

2.6 Fragment Shader para Iluminação de Superfície de Água

Por fim, foi desenvolvido um *fragment shader* que simula a iluminação de uma superfície de água. É considerada a iluminação do sol, a cor ambiente, e utiliza um modelo Blinn-Phong para os reflexos especulares. Além disso, inclui o efeito Fresnel para acentuar os reflexos em ângulos rasos, ajustando sua intensidade e suavidade.

A dispersão subsuperficial (SSS) também é implementada, simulando a maneira como a luz interage com as cristas das ondas e sob a superfície da água. A coloração da água é ajustada para um tom frio, com um efeito de sombra suave para maior naturalidade. Esse efeito foi baseado na fórmula desenvolvida em [1].

Finalmente, o shader aplica High Dynamic Range (HDR) para alcançar um efeito de iluminação saturado, especialmente na luz solar.

3 Resultados

Os resultados obtidos foram satisfatórios, permitindo a animação em tempo real de uma malha de vértices com movimentação baseada no deslocamento vertical e horizontal, calculados através da Transformada Inversa de Fourier. A simulação demonstra a dinâmica realista das ondas, com suavidade na atualização das superfícies e sem comprometimento no desempenho. As Figuras 1 e 2 ilustram o comportamento da simulação em diferentes instantes.

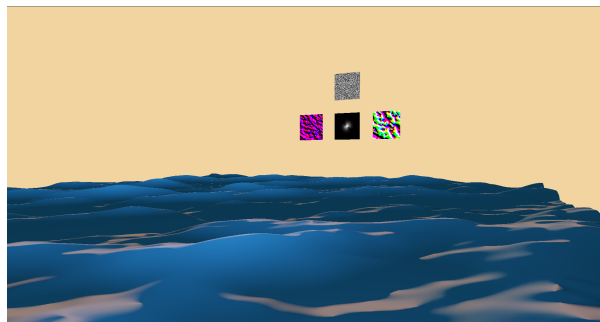


Figure 1: Simulação de superfície de água em um instante de tempo.

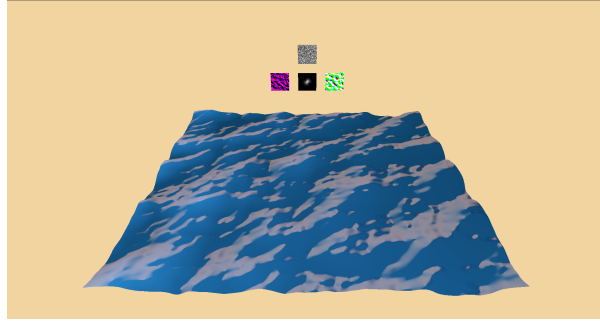


Figure 2: Outro instante da simulação com variação no deslocamento das ondas.

4 Conclusão

Neste trabalho, foi implementada uma simulação realista de superfícies de água utilizando a Transformada Rápida de Fourier (FFT). A partir dos conceitos apresentados por Tessendorf, foi possível gerar ondas oceânicas de maneira eficiente e em tempo real, aplicando técnicas como o cálculo das normais no domínio da frequência e o efeito de choppiness para aumentar o realismo da simulação.

Os resultados mostram que a abordagem adotada é capaz de produzir animações fluidas e visualmente atraentes, o que a torna adequada para aplicações em tempo real, como jogos e simulações interativas. Contudo, há espaço para aprimoramentos, especialmente no que diz respeito à inclusão de efeitos adicionais, como espuma, reflexões mais detalhadas e a implementação de uma Skybox para melhorar a ambientação.

No futuro, pretende-se explorar técnicas avançadas de renderização e pós-processamento, assim como otimizações de performance, para permitir a simulação de superfícies de água em maior escala e com maior complexidade visual.

5 Trabalhos Futuros

Para o futuro, planeja-se o aprimoramento do projeto com melhorias gráficas adicionais, tais como:

- Renderização de múltiplas malhas para simulação de diferentes corpos d'água.
- Adição de efeitos de espuma nas cristas das ondas para maior realismo.
- Inclusão de uma Skybox para ambientação.
- Implementação de reflexões do ambiente na superfície da água.
- Aplicação de técnicas avançadas de pós-processamento para refinar a qualidade visual.

References

- [1] Mark Mihelich and Tim Tcheblovkov. “Wakes, Explosions, and Lightning: Interactive Water Simulation in 'Atlas'”. In: *Game Developers Conference (GDC)*. 2019.
- [2] Jerry Tessendorf. “Simulating Ocean Water”. In: *Proceedings of the 2004 ACM SIGGRAPH Course Notes*. 2004.