

Adaptando um Algoritmo de Geração de Terrenos para a Renderização de Águas de Oceanos

Gustavo Wagner, Alberto Raposo, Marcelo Gattass

Tecgraf – Grupo de Tecnologias em Computação Gráfica
Departamento de Informática – Pontifícia Universidade Católica (PUC-Rio)
R. Marquês de São Vicente, 225 – Rio de Janeiro – RJ – Brasil
{gustavow, abraposo, mgattass}@tecgraf.puc-rio.br

Abstract. *We present a simple and efficient method to render ocean water in real-time adapted from height map visualization, normally used in terrain visualization. We also show how to animate the water surface to simulate the existence of waves.*

Resumo. *Apresentamos um método simples e eficiente de gerar oceanos em tempo real adaptado a partir de técnicas usadas para a visualização de mapas de altura, normalmente utilizadas na visualização de terrenos. Também mostramos como movimentar a superfície da água para simular a existência de ondas.*

1. Introdução

A qualidade da representação do mar é crucial para complementar o realismo em ambientes virtuais que necessitam desse tipo de representação, tais como os usados em simulações de operações da marinha e na área de exploração de petróleo [Corseuil et al. 2004].

A implementação apresentada neste artigo se divide em duas etapas. Primeiro geramos uma malha de triângulos com a resolução apropriada para cada ponto do oceano. Em seguida deslocamos cada vértice da malha sob efeito de uma função que simula o movimento de ondas.

2. Métodos para Geração da Malha

Para gerar a malha do oceano usamos uma técnica semelhante à usada para terrenos em [de Boer 2000]. As características da malha gerada são muito semelhantes nos dois casos. Em ambos queremos mais triângulos em regiões próximas à câmera e menos em regiões distantes, o que tem que ser feito com cuidado para evitar a formação de aberturas e T-Vertices entre as junções de cada região.

Com terrenos, também é necessário garantir que as regiões com maiores variações de relevo sejam criadas com mais triângulos do que regiões planas. Mas isso não é necessário para oceanos já que as variações de altura causadas pelas ondas são pequenas e bem distribuídas. Assim, podemos ignorar o relevo no critério de escolha de resolução, levando em consideração apenas a distância de cada região à câmera.

As regiões da superfície do oceano são criadas a partir de divisões sucessivas de uma única região inicial. Começamos com um grande retângulo, grande o suficiente

para que o usuário não consiga ver suas bordas e seguimos esses passos:

1. Se o retângulo atual estiver fora do campo de visão podemos ignorá-lo.
2. Se a resolução do retângulo atual for suficiente para a distância da câmera, podemos selecioná-lo para a renderização.
3. Caso contrário, devemos subdividi-lo em quatro, repetindo esse algoritmo para cada parte resultante.

Cada retângulo selecionado é então tesselado para formar um grid NxN. Como estamos variando a resolução para partes da mesma superfície, temos que dar tratamento especial para fronteiras entre retângulos de resoluções diferentes. Sempre que isso ocorrer, devemos remover os vértices que não fizerem parte do retângulo de menor resolução (vértices cinzas na figura 1a) da malha gerada para o retângulo de maior resolução. Essa operação é semelhante a mover o vértice a ser removido para um de seus vizinhos na borda e eliminar a face degenerada que será formada.

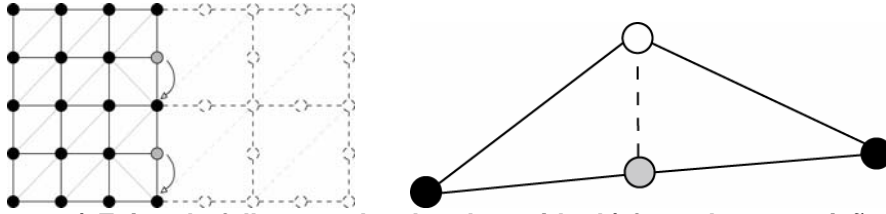


Figura 1. a) Evitando falhas nas bordas dos grids; b) fazendo a transição entre diferentes resoluções

Mudanças repentinas na resolução de um grid podem causar “pulos” na visualização dependendo do nível de detalhamento usado, o que podemos evitar fazendo uma transição suave entre cada resolução (Geomorphing). A medida que um grid se aproxima da mudança para uma resolução mais baixa devemos mover cada vértice que será apagado (vértice branco na Figura 1b) para perto da aresta que assumirá o seu lugar (vértice cinza). Por outro lado, quando esse grid se aproxima de uma resolução mais alta, temos que levar o vértice de volta à posição original.

3. Dando Movimento às Ondas

A geometria criada é então movimentada com Ondas de Gerstner [Tessendorf 2001], que descrevem o deslocamento de pontos na superfície do oceano. Cada vértice criado durante a fase de tesselação, independentemente da resolução do Grid em que está, é deslocado em x, y e z de acordo com as seguintes equações:

$$\vec{x} = \vec{x}_0 - (\vec{k}/k)A \sin(\vec{k} \cdot \vec{x}_0 - \omega t) \quad (1) \quad k = 2\pi/\lambda \quad (3)$$

$$z = A \cos(\vec{k} \cdot \vec{x}_0 - \omega t) \quad (2) \quad kA < \lambda \quad (4)$$

Nas equações (1) e (2), \vec{x}_0 e \vec{x} são vetores em \mathbb{R}^2 correspondendo, respectivamente, à posição original e posição deslocada de um determinado vértice. A variável z é a componente vertical do vértice deslocado. O vetor \vec{k} , denominado *wavevector*, representa a direção de deslocamento desta onda. A constante k é calculada a partir do comprimento de onda λ de acordo com a equação (3). A frequência desta

onda é representada por ω e a sua amplitude é dada por A . Para evitar a formação de ondas inconsistentes, devemos sempre respeitar a inequação (4).

O resultado obtido a partir da aplicação direta da equação acima é bem limitado, resultando em ondas repetitivas, que se movem sempre na mesma direção. Para melhorá-lo temos que fazer um somatório de várias ondas desse tipo, variando os parâmetros de formação de cada uma (Figura 2a):

$$\vec{x} = \vec{x}_0 - \sum_{i=1..N} (k_i/k_i) A_i \sin(k_i \cdot \vec{x}_0 - \omega_i t) \quad (5)$$

$$z = \sum_{i=1..N} A_i \cos(k_i \cdot \vec{x}_0 - \omega_i t) \quad (6)$$

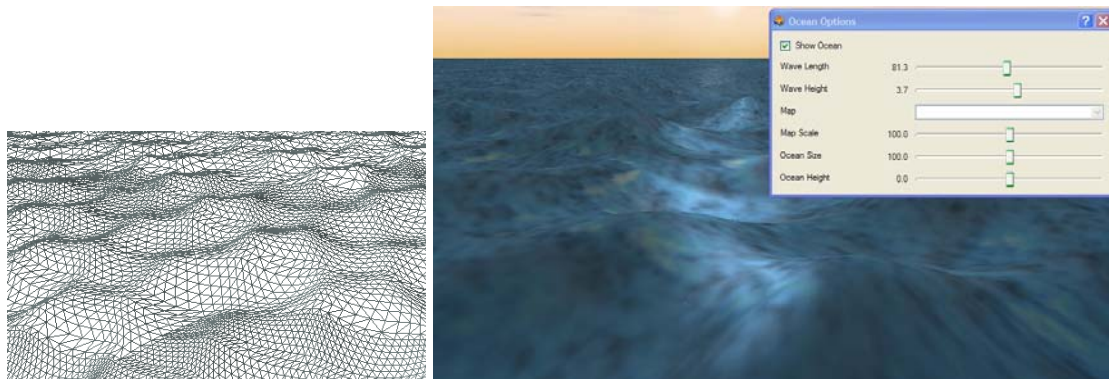


Figura 2. a) malha gerada; b) interface de configuração

4. Conclusões

O oceano implementado foi adaptado para o OpenSceneGraph [Osfield 2004], uma biblioteca aberta de grafos de cena. Permitimos a configuração de vários parâmetros da geração de ondas, como o seu comprimento e altura, na interface para dar ao usuário controle sobre a sua aparência (Figura 2b).

Fizemos testes com três cenas de tamanhos variados em um Pentium4 com 3GB de RAM e uma nVidia Quadro FX 1000. Os resultados da Tabela 1 mostram que o impacto do oceano na cena é relativamente pequeno, principalmente para cenas com uma quantidade expressiva triângulos.

Tabela 1. Impacto do oceano na performance

| Triângulos na Cena | Com Oceano | Sem Oceano |
|--------------------|------------|------------|
| 300 mil | 52,1 fps | 73,7 fps |
| 1,3 milhões | 18,3 fps | 22,4 fps |
| 2,3 milhões | 11,5 fps | 12,9 fps |

Referências

- de Boer, W. H. (2000) "Fast Terrain Rendering Using Geometrical MipMapping", <http://www.fipcode.com/tutorials/geomipmaps.pdf>
- Corseuil, E. T. L., Raposo, A. B. et al. (2004) "ENVIRON - Visualization of CAD Models in a Virtual Reality Environment". Proc. EG-VE 2004 – Eurographics Workshop on Virtual Environments.
- Osfield, R., Burns, D. (2004) "OpenSceneGraph". <http://www.openscenegraph.org>
- Tessendorf, J. (2001), "Simulating Ocean Water". In: Simulating Nature: Realistic and Interactive Techniques. SIGGRAPH 2001 Course Notes 47.