

# Ondas Irreais

Gubi

24 de outubro de 2015

Vamos simular uma chuva fantasiosa em um planeta imaginário, gerando ondas irreais. Para compor um cenário um pouco mais surrealista, um material especial cobre toda a borda do lago retangular, absorvendo completamente as ondas que nela tocam.

Discos vagam pelo lago, com a mesma velocidade escalar, mas direções variadas, com choques perfeitamente elástico entre eles e as bordas. Estes discos, ao contrário das bordas, refletem a frente de onda nos pontos onde toca. Não há difração ou absorção. Daqui para frente chamaremos os discos de “esponjas”<sup>1</sup>.

## 1 O programa

O programa deve simular uma chuva neste lago. Cada gota produz uma onda irreal que se propaga de acordo com a função

$$h(\rho, t) = (\rho - v \cdot t) \cdot e^{-(\rho - v \cdot t)^2} \cdot e^{-t/10}$$

onde  $h$  é a altura da água com relação à superfície do lago em repouso. Os parâmetros são  $\rho$ , em coordenadas polares  $(\rho, \theta)$ , a partir do ponto de queda da onda e  $t$ , o tempo desde a queda da gota. Note que  $h$  não depende de  $\theta$ . A velocidade de propagação da onda é  $v$ .

Os parâmetros de simulação são fornecidos em um arquivo texto, passado como primeiro argumento. Os outros argumento é o número de processadores utilizado.

---

<sup>1</sup>**ATENÇÃO:** em aula comentei que as esponjas apagariam as ondas, mas podem aparecer anomalias estranhas demais, mesmo em um mundo surrealista. Mais detalhes na aula.

## 2 Arquivo de entrada

O arquivo de entrada possui os seguintes parâmetros, um por linha:

- Tamanho do lago, um par de valores entre parênteses:  $(larg, alt)$
- Tamanho da matriz, outro par de valores entre parênteses:  $(L, H)$
- Tempo (virtual) de simulação:  $T$
- Velocidade de propagação da onda:  $v$
- Número de iterações:  $N_{iter}$
- Probabilidade de surgimento de uma gota, por iteração:  $P$
- Número de esponjas:  $n_e$
- Diâmetro das esponjas:  $d_e$
- Velocidade escalar das esponjas:  $v_e$
- Semente para o gerador aleatório:  $s$

Este é um exemplo de entrada:

```
(1000,2000)
(500,100)
120
10
1000
.34
3
7
12.4
7623891
```

## 3 Simulação

Na inicialização, são geradas as  $n_e$  esponjas, cada uma com sua posição e velocidade, com as seguintes restrições:

- Uma esponja não pode ultrapassar as bordas do lago.
- A velocidade é um vetor bi-dimensional com módulo  $v_e$ .
- Para simplificar, uma esponja é eliminada se na geração, se sobrepuser a alguma já existente,

A simulação é feita por uma sequência de  $N_{iter}$  passos, chamados *timesteps*. Cada *timestep* equivale, portanto, a  $\frac{T}{N_{iter}}$  segundos.

Em cada passo estas operações devem ser efetuadas, nesta ordem:

1. Mover as esponjas.
2. Calcular o nível do lago em cada ponto (propagação das ondas).
3. Verificar, por sorteio, se uma nova gota deve aparecer e sortear sua posição.

## 4 Saída

A saída será um arquivo *PPM*, tipo P3 (colorido ASCII), contendo a imagem final do lago.

Considere a superfície do lago em repouso como altura 0 e representados em preto. Valores positivos devem ser representados em azul e negativos em vermelho.

Para determinar a intensidade de cor, use o seguinte algoritmo:

1. Seja  $h_{max}$  a altura máxima do líquido
2. Seja  $p_{max}$  a profundidade máxima
3. Seja  $\Delta = \frac{\max(h_{max}, -p_{max})}{255}$
4. Seja  $h(x, y)$  a altura do líquido no ponto  $(x, y)$
5. Seja  $(x_i, y_j)$  o ponto no lago correspondente ao pixel  $(i, j)$  da imagem.
6. Para cada pixel  $(i, j)$  da imagem, o valor da componente azul (ou vermelha, se  $h(x_i, y_j) < 0$ ), é dado por

$$\left\lceil \frac{h(x_i, y_j)}{\Delta} \right\rceil$$

7. Os pontos cobertos por uma esponja devem ser pintados de verde  $((0, 255, 0))$ .

## 4.1 Formato *PPM*

O formato é bastante simples:

- A primeira linha contém o tipo da imagem, com dois caracteres:

**P1** Bitmap, em modo ascii

**P2** *Greyscale*, modo ascii

**P3** Colorida, modo ascii

**P4** Bitmap, binária

**P5** *Greyscale*, binária

**P6** Colorida, binária

As linhas seguintes podem começar com ‘#’ e servem como comentários.

Em seguida vem um par de inteiros representando a largura e a altura da imagem. O próximo inteiro indica o valor máximo de uma componente, no nosso caso, será 255.

Em seguida são colocados os valores de cada pixel. Para o formato *P3* são triplas de inteiros, com as componentes RGB. No *P6* são triplas de *bytes*. Para mais informações, procure pelo formato *Netpbm* no *Google* ou similar.

**Que a velocidade esteja com vocês.**