



Texturas Procedimentais



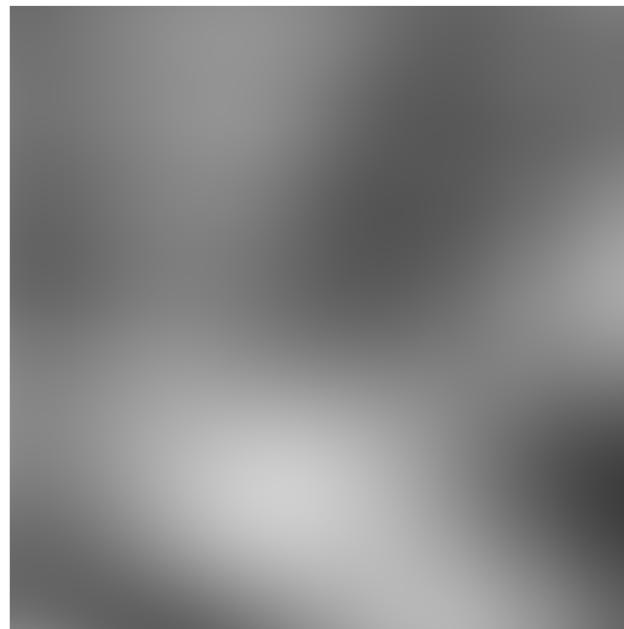
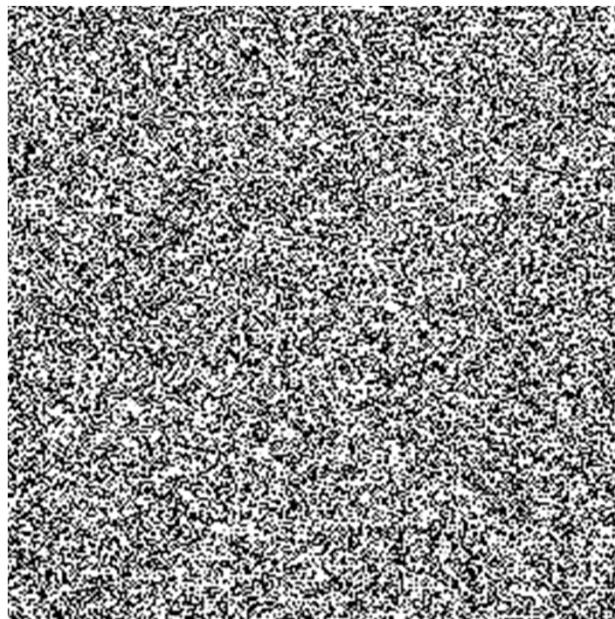
Texturas Procedimentais

- Resumo
 - Noise based
 - Cellular textures
 - Reaction-Diffusion



Noise

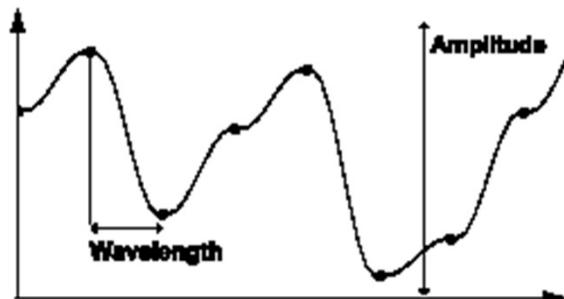
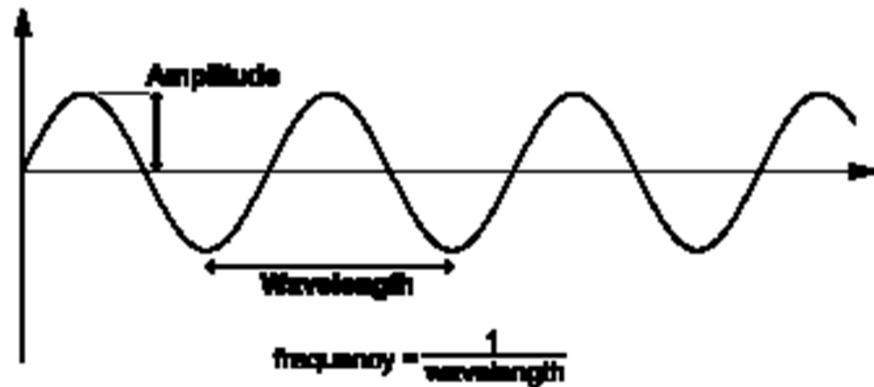
- Fractal Brownian Motion (fBM)
 - Perlin's Smooth Noise





Noise

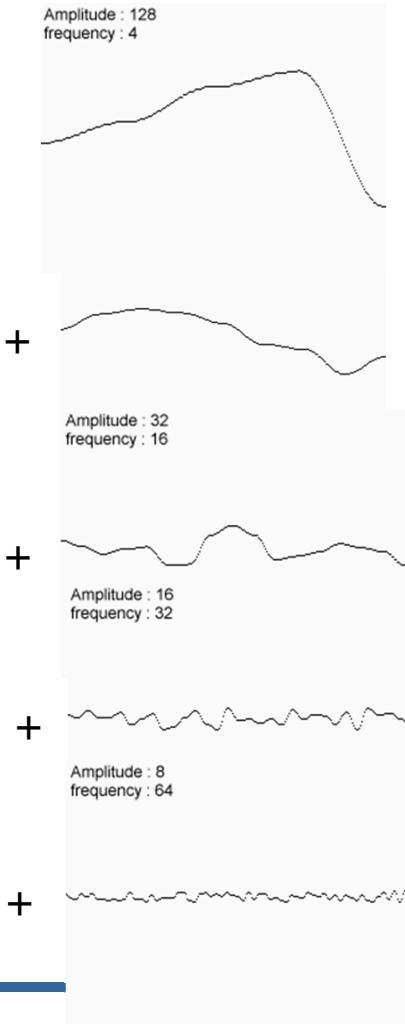
- Ruído: Branco e "Cor de Rosa"
- Noção de
 - Frequência
 - Amplitude



imagens do site de Hugo Elias



Noise



- Soma de componentes

$$\text{altura}(x) = \sum_{i=1}^n \text{noise}(x \times 2^i) \frac{\text{amplitude}}{2^i}$$

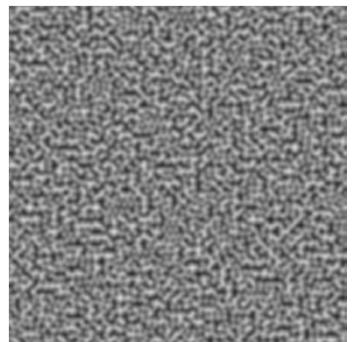
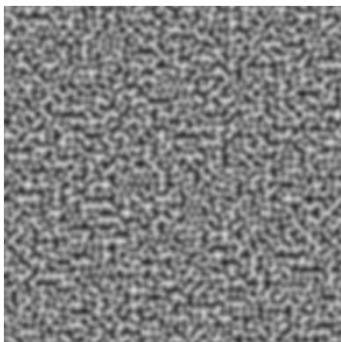
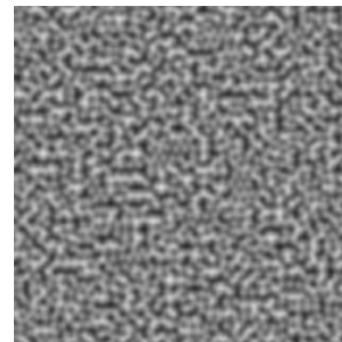
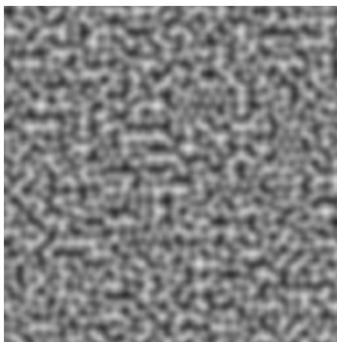
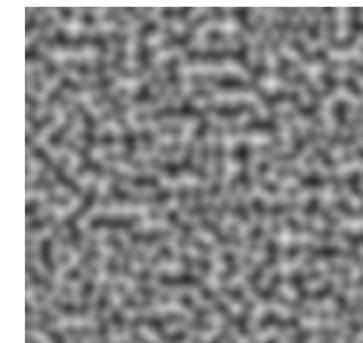
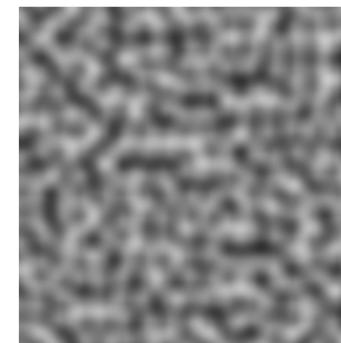
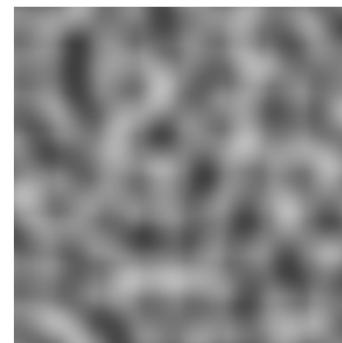
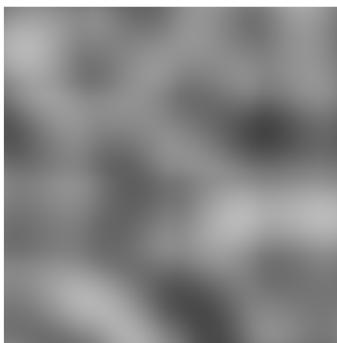
imagens do site de Hugo Elias



Noise

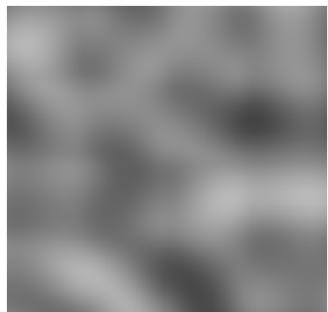
- Smooth Noise - Frequência

imagens do site de Hugo Elias





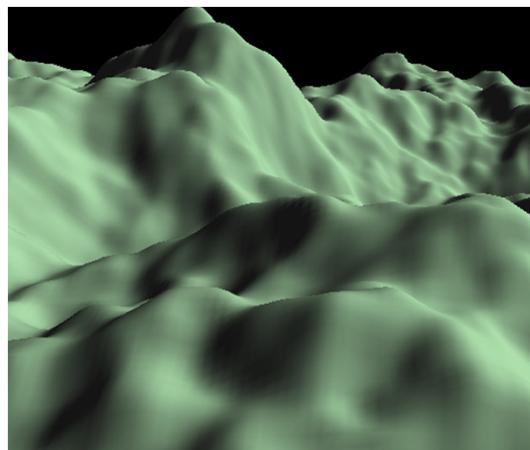
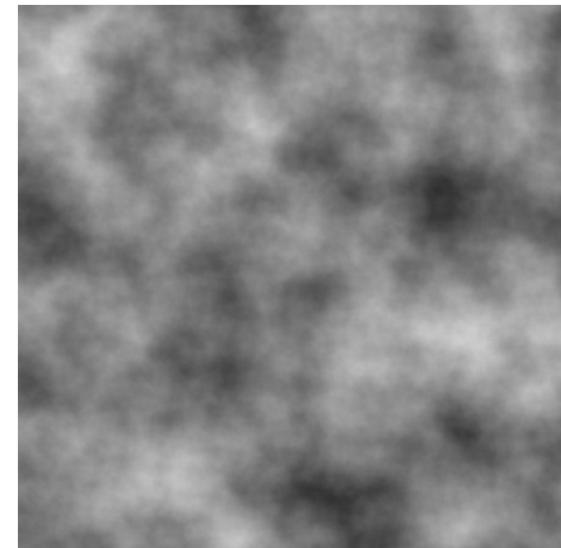
Noise



+ ... +



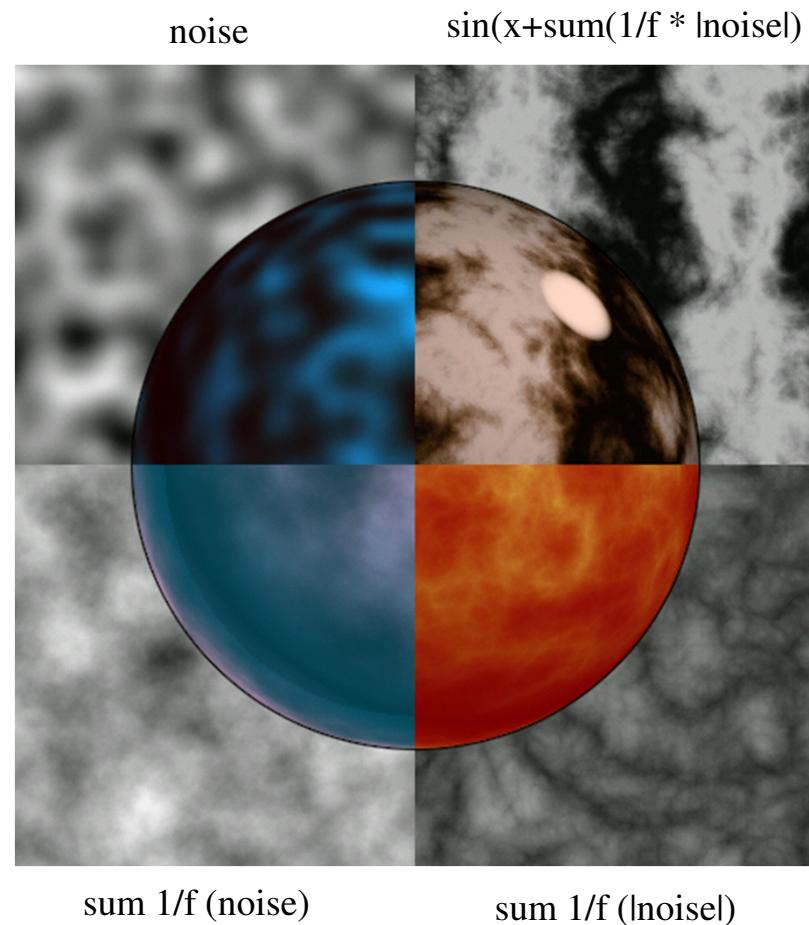
=





Noise

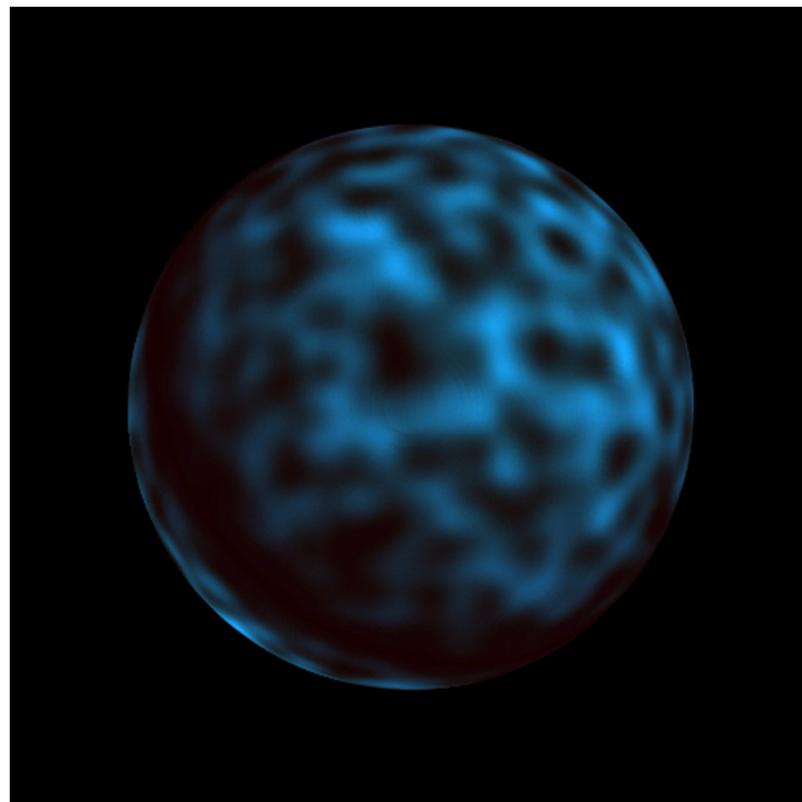
- Alguns efeitos:





Noise

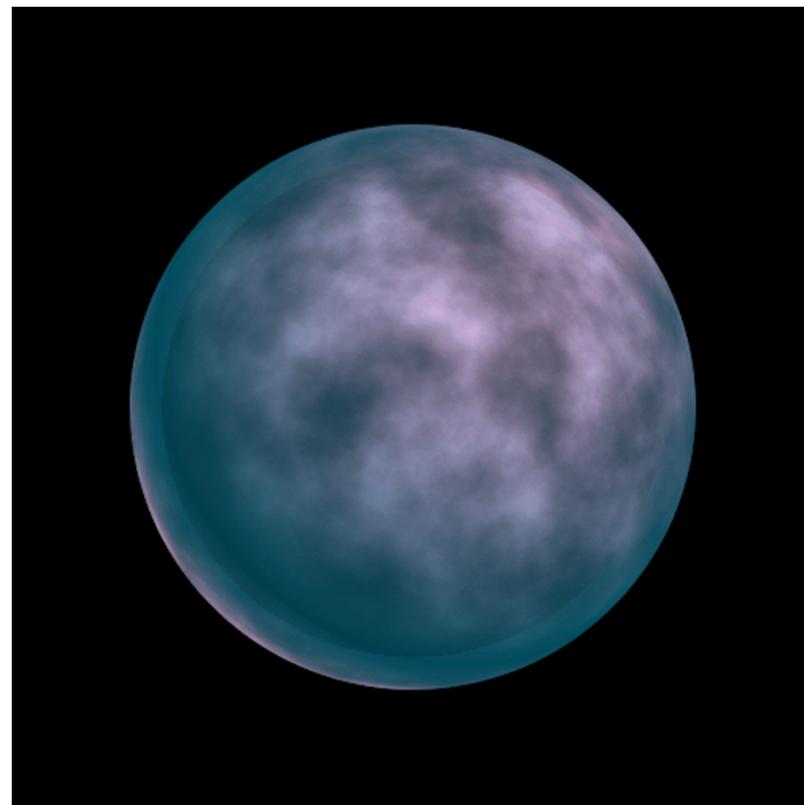
- simples
 - permite adicionar irregularidade a superfícies
 - alterar a componente especular para dar a ilusão de material desgastado
 - bump mapping





Noise

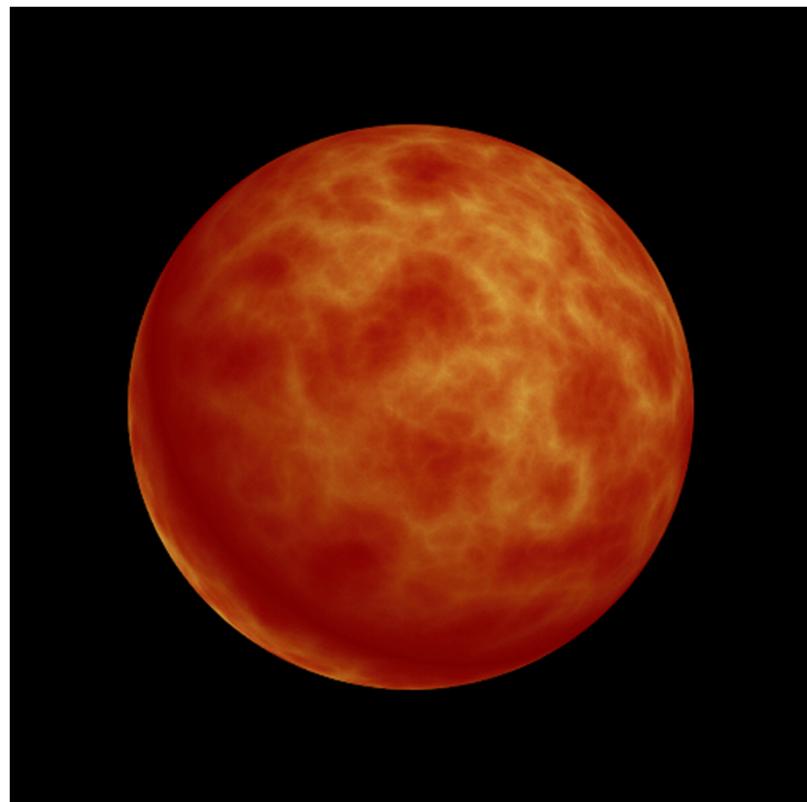
- sum 1/f noise
 - $n(p) + \frac{1}{2} * n(p*2) + \frac{1}{4} * n(p*4) + \dots$
 - nuvens
 - terrenos





Noise

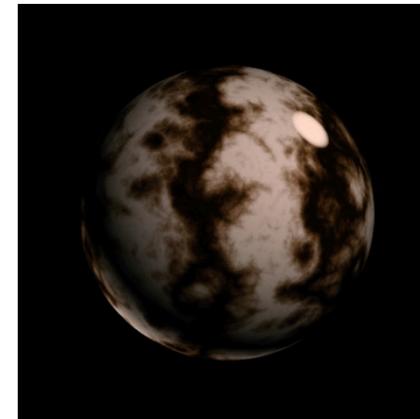
- sum $1/f |\text{noise}|$ aka turbulence
 - $|\text{n}(p)| + \frac{1}{2} * |\text{n}(p*2)| + \frac{1}{4} * |\text{n}(p*4)| + \dots$
 - cria discontinuidades no gradiente
 - lava





Noise

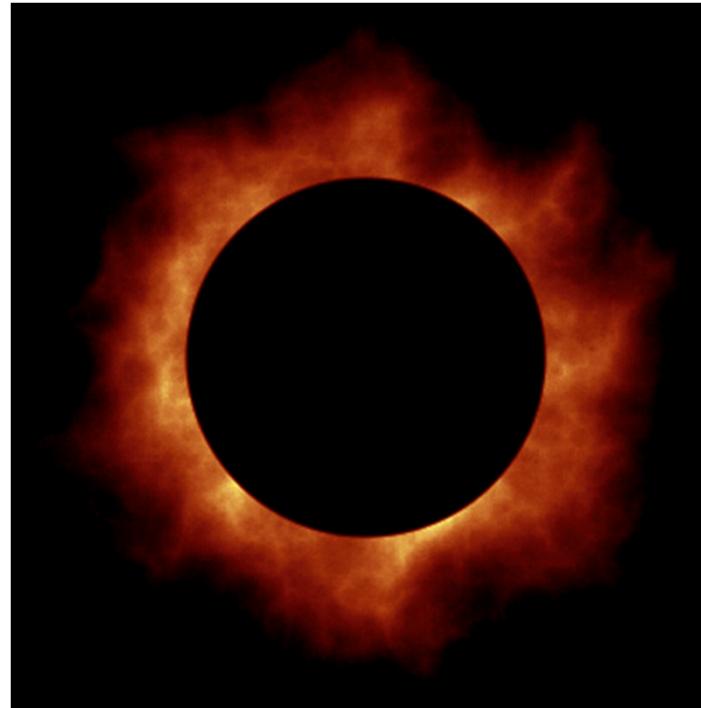
- $\sin(x + \text{sum } 1/f \text{ |noisel|})$
 - Aplicação da turbulência apresentada anteriormente para alterar a fase da função





Noise

- Fogo





Cellular Textures

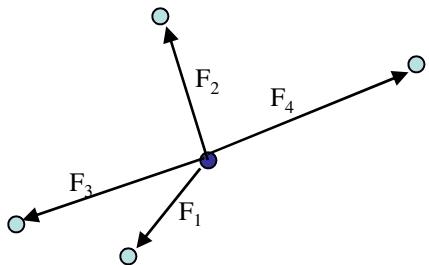
- Steven Worley
 - Processo de geração de texturas baseado em partição espacial e funções de distância





Cellular Textures

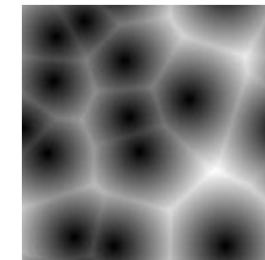
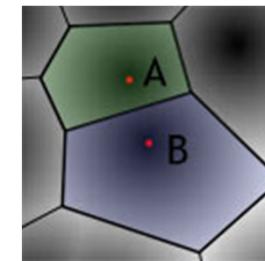
- Geração de pontos aleatórios em 2D
 - F_1 = distância ao ponto mais próximo
 - F_2 = distância ao segundo ponto mais próximo
 - ...
 - F_n = distância ao enésimo ponto mais próximo





Cellular Textures

- Atribuir uma cor a cada ponto gerado
- Colorir os pixels com a cor do ponto mais próximo (F1)
- Ou atribuir uma intensidade de acordo com a distância ao ponto mais próximo (F1)





Cellular Textures

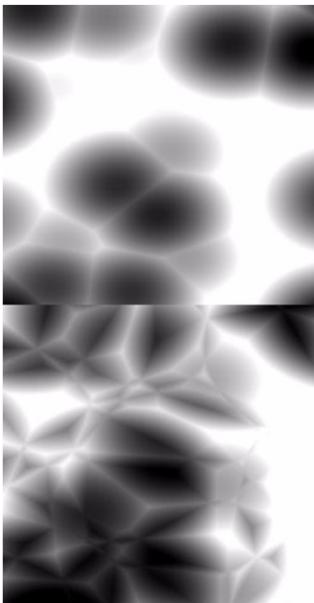
- Combinar funções

$F2 - F1$

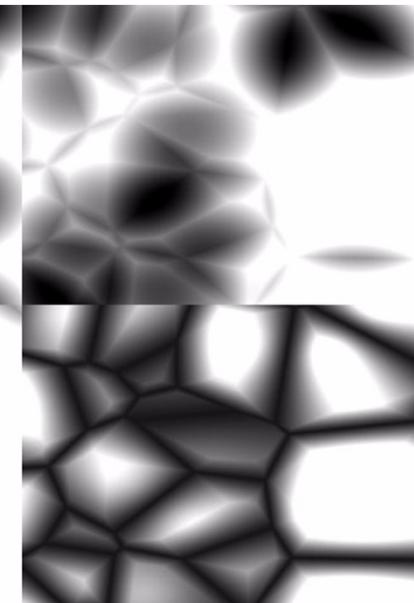
- Combinações com 4 funções

$C1F1 + C2F2 + C3F3 + C4F4$

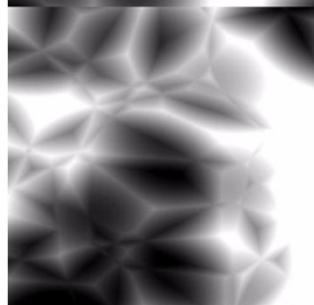
F1



F2



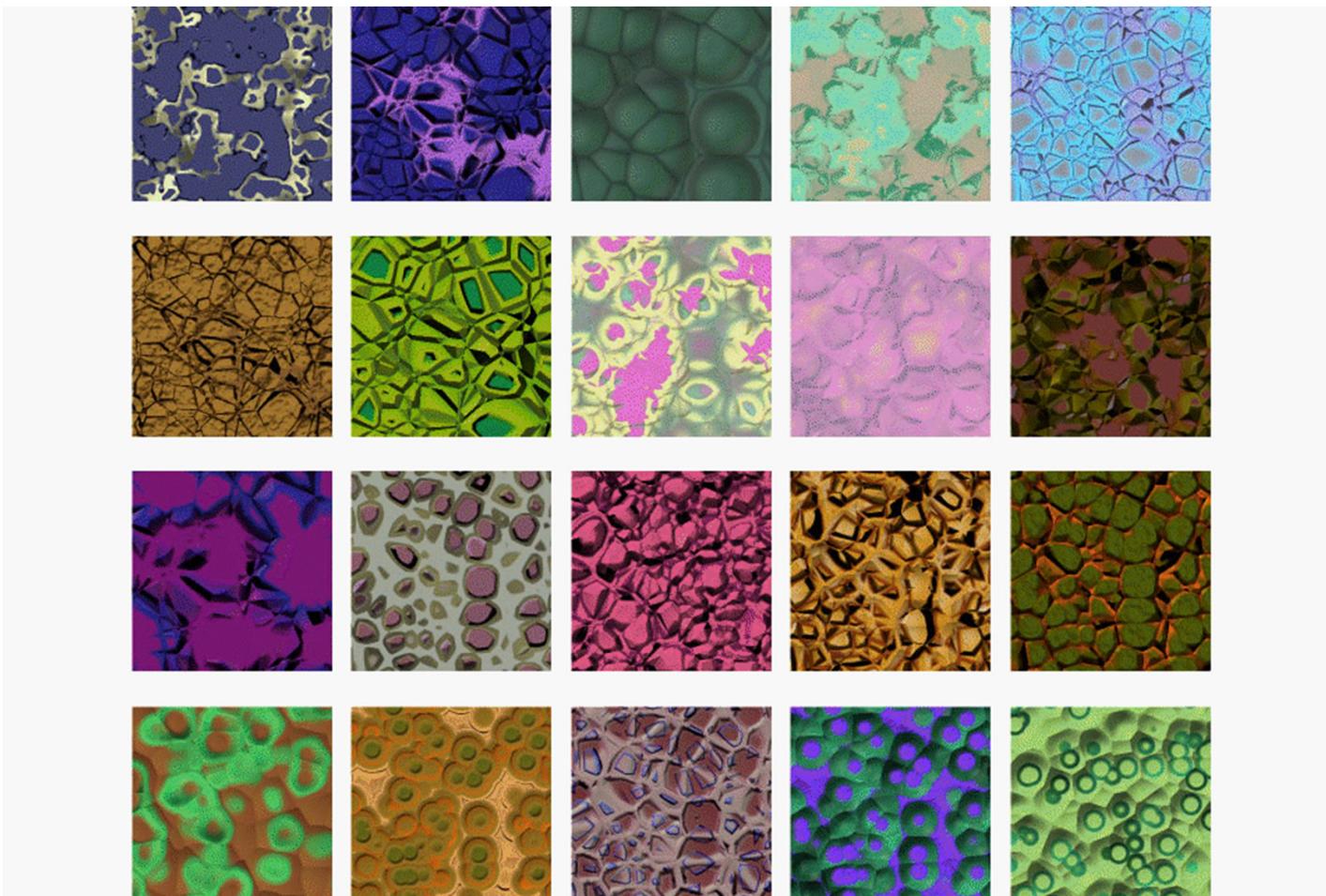
F3



$F2-F1$



Cellular Textures





Cellular Textures

- Versão Fractal
 - Combinar várias texturas a várias escalas (tal como com Noise)

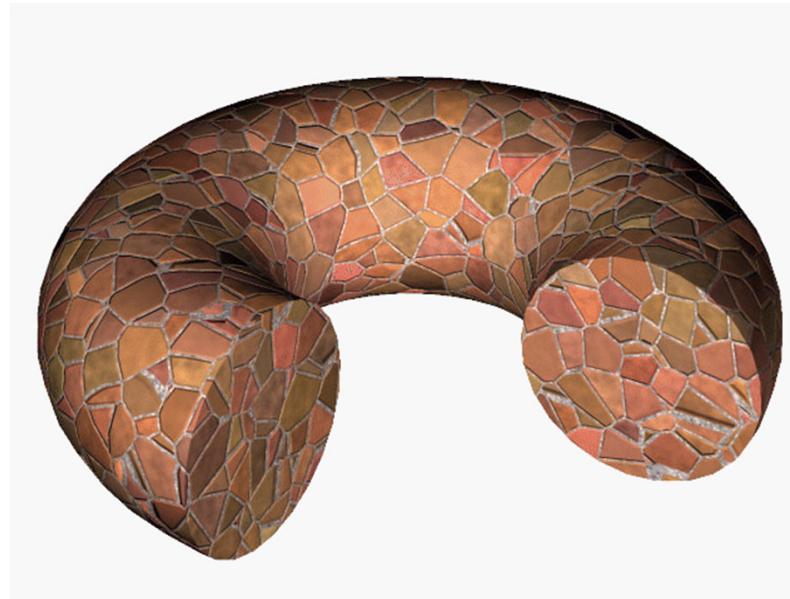
$$C = \frac{\sum F_n(2^i p)}{2^i}$$





Cellular Textures

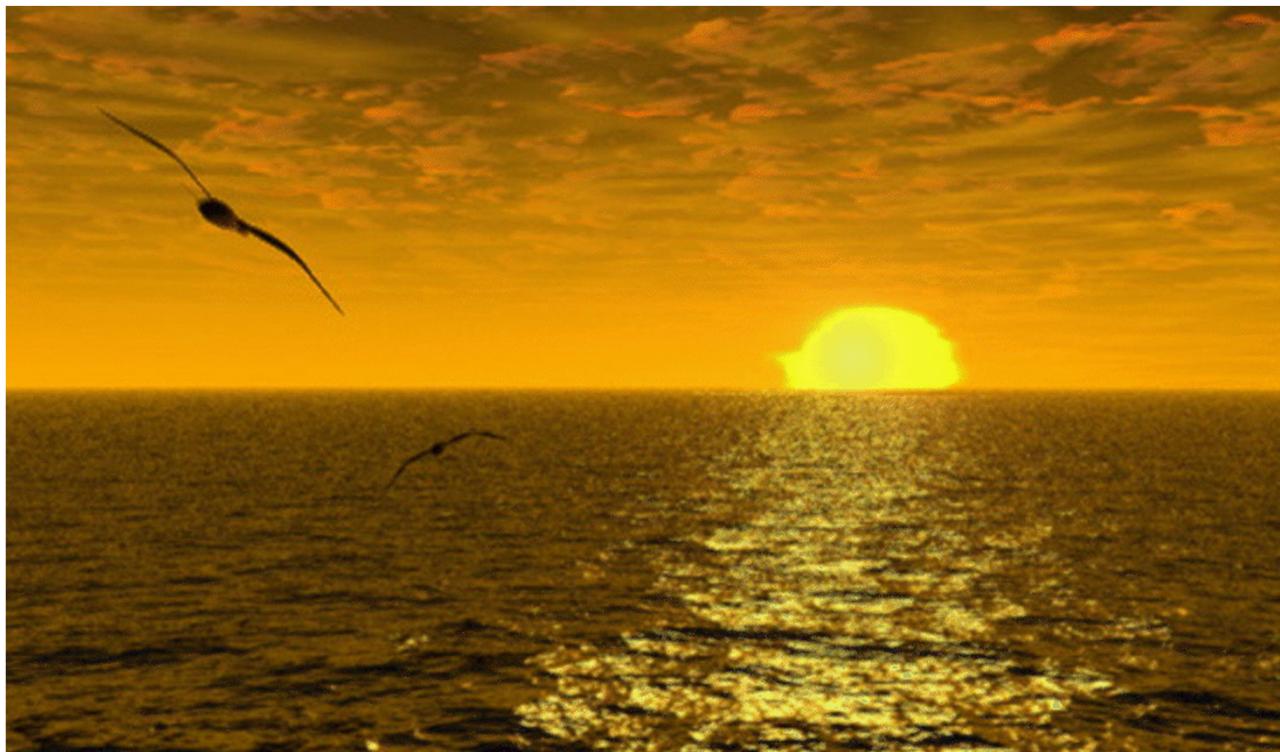
- Cor constante para F1 + fractal noise, F2-F1 bump mapping.





Cellular Textures

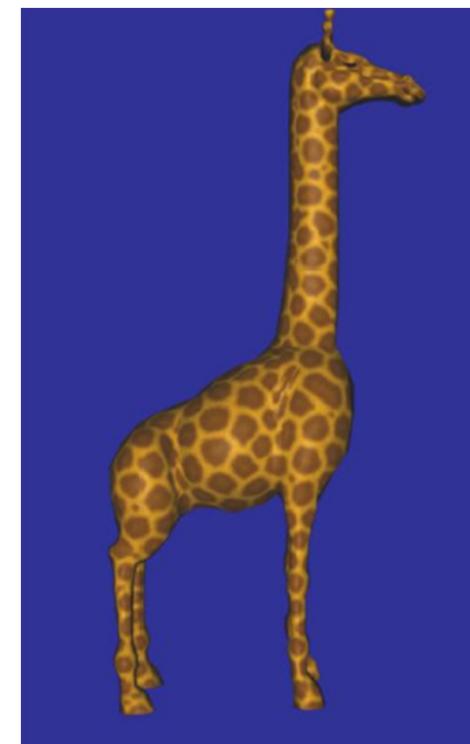
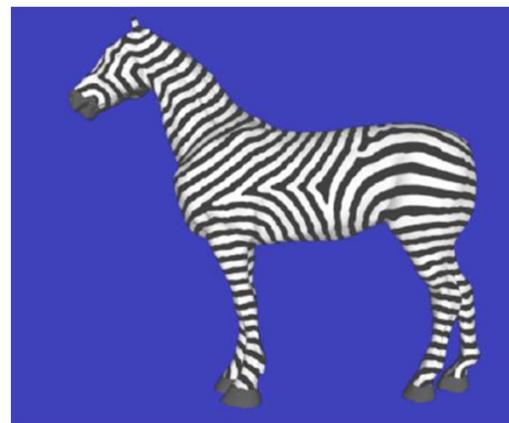
- F2-F1 para bump mapping + environment map





Reaction-Diffusion

- Modelo matemático para geração de padrões naturais.
- Simulação da propagação e reacção de químicos numa superfície.





Reaction-Diffusion

- Considerando dois químicos, a e b temos a seguinte formulação matemática:

$$\frac{\partial a}{\partial t} = F(a, b) + D_a \nabla^2 a$$

$$\frac{\partial b}{\partial t} = G(a, b) + D_b \nabla^2 b$$

- $F(a, b)$ - Função das concentrações locais dos químicos a e b .
- D_a - velocidade de difusão de a .
- $\nabla^2 a$ - medida da concentração de a , relativamente à concentração de a na vizinhança do ponto



Reaction-Diffusion

- Formula discreta proposta por Turing para geração de padrões:

$$\Delta a_{ij} = s(16 - a_{ij}b_{ij}) + D_a(a_{i-1,j} + a_{i+1,j} + a_{i,j+1} + a_{i,j-1} - 4a_{ij})$$

$$\Delta b_{ij} = s(a_{ij}b_{ij} - b_{ij} - \beta_{ij}) + D_b(b_{i-1,j} + b_{i+1,j} + b_{i,j+1} + b_{i,j-1} - 4b_{ij})$$

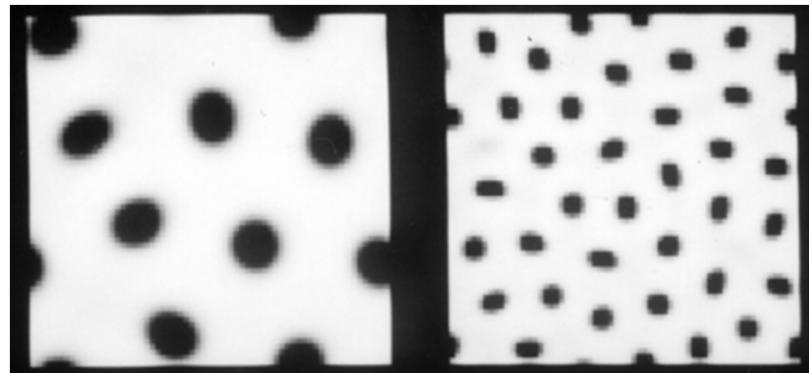
- o valor de β_{ij} é um valor aleatório destinado a introduzir irregularidade na propagação do químico b .
- Os valores de D_a e D_b propostos por Turing são respectivamente 0.25 e 0.0625, o que significa que a se propaga mais rapidamente que b .



Reaction-Diffusion

- exemplos (concentração de b branco = max, preto = min):
 - A concentração de a é maior nas zonas pretas.
 - valores iniciais $a=b=4$

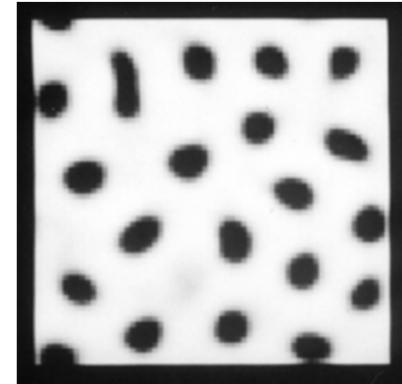
$s = 0.05$



$\beta_{ij} = 12 \pm 0.1$

$s = 0.2$

$s = 0.2$

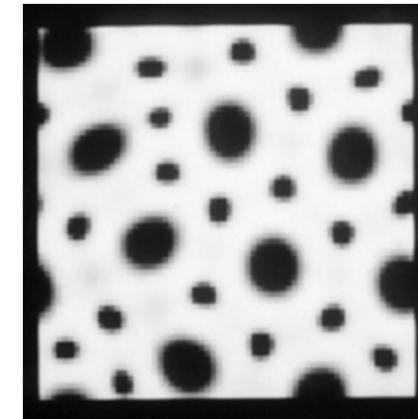


$\beta_{ij} = 12 \pm 3$



Reaction-Diffusion

- Composição de efeitos
 - passo 1: $s=0.05$
 - formação das manchas grandes
 - congelar células onde $b_{ij} \in [0,4]$
 - manter concentrações de a e b para as células congeladas (manchas negras grandes)
 - passo 2: $s = 0.2$
 - As manchas congeladas participam nos cálculos para as zonas não congeladas mas os seus valores não são alterados. A grande concentração de a nas zonas pretas evita a formação de manchas novas perto das manchas antigas.

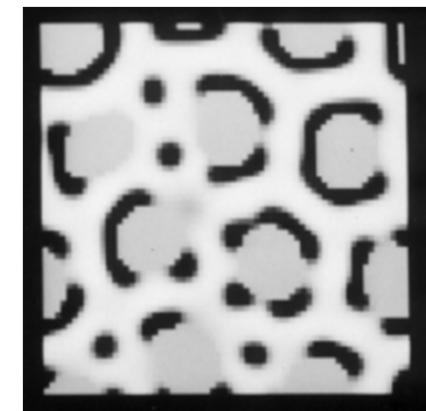




Reaction-Diffusion

- Composição de Efeitos

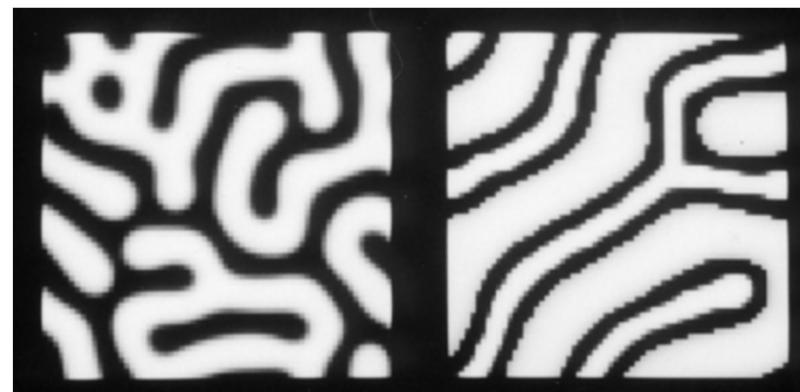
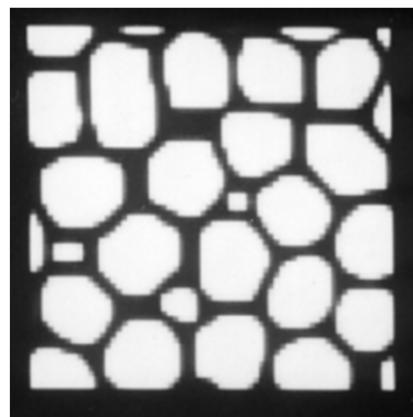
- passo 1: $s = 0.05$
 - congelar células onde $b_{ij} \in [0,4]$
 - alterar concentrações das áreas congeladas para $a=b=4$
- passo 2: $s = 0.2$
 - neste caso formam-se manchas negras perto das manchas antigas (a cinza) uma vez que a concentração de a não é suficientemente grande para inibir a formação de novas manchas na vizinhança.





Reaction-Diffusion

- Sistema de Meinhardt





Refs

- Ebert, Musgrave, Peachey, Perlin and Worley, *Texture and Modelling*, 2nd edition
- Greg Turk, *Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion*, *Computer Graphics*, Vol. 25, No. 4, pp. 289-298, (SIGGRAPH '91)