

# Filtragem Espacial

---

Prof. Antonio L. Apolinário Jr.  
Estagiaria Docente: Rafaela Alcantara

UFBA/IM/DCC/BCC - 2018.1

## Roteiro

---

- Motivação
- Funcionamento de um filtro
- Filtros Espaciais de Suavização
- Filtros Espaciais de Aguçamento
- Implementação em Three.js/WebGL

## Leitura de referencia

---

- Capítulo 3

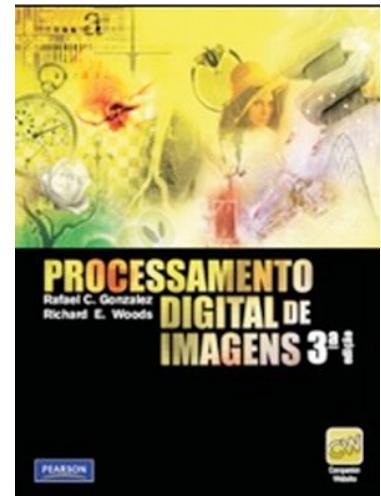
### **Processamento Digital de Imagens**

GONZALEZ, RAFAEL C.

WOODS, RICHARD E.

3<sup>a</sup> edição

Pearson Education do Brasil- 2010



## Leitura de referencia

---

- Capítulo 7

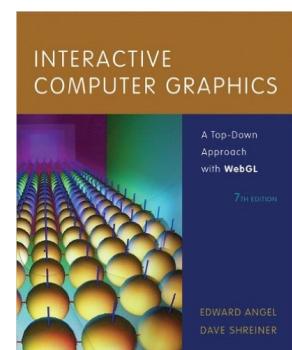
### **Interactive Computer Graphics - A top-down approach with WebGL**

7<sup>th</sup> Edition

Angel, Edward

Shreiner, Dave

Addison-Wesley. 2014.



- Capítulo 20

### **Computer Graphics : Principles and Practice Third Edition in C**

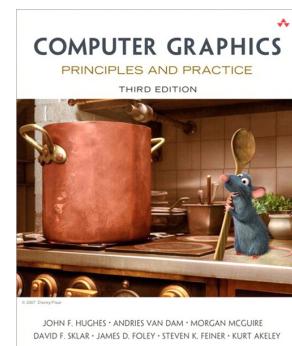
3rd Edition

John F. Hughes / Andries van Dam

Morgan McGuire / David F. Sklar

James D. Foley / Steven K. Feiner

Addison-Wesley. 2013.



## Leitura de referencia

---

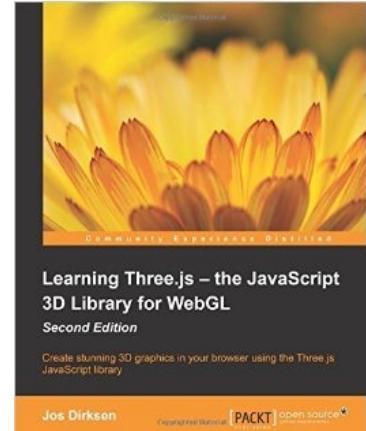
- Capitulos 11

### **Learning Three.js: The JavaScript 3D Library for WebGL**

Jos Dirksen

2<sup>nd</sup> Edition

Packt Publishing - 2013.



## Motivação

## Filtragem Espacial

---

- Transformações de intensidade modificam pontualmente (pixel) a imagem
- Analises mais elaboradas necessitam de mais informações a cerca da vizinhança de cada pixel
  - Exemplos:
    - Identificação de bordas
    - Eliminação de ruídos

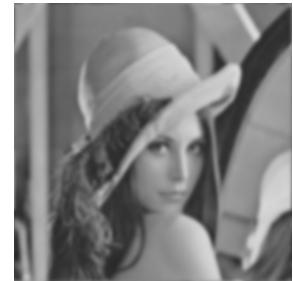
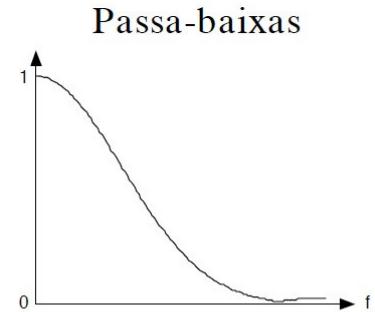
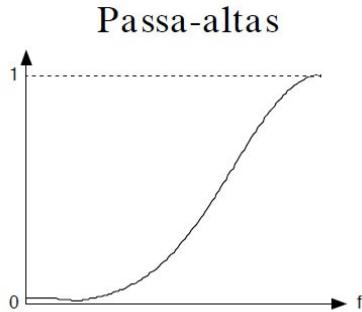
## Funcionamento de um Filtro Espacial

---

- Conceito de filtragem
  - Processamento de Sinais
    - Processo de aceitar/rejeitar certos componentes de freqüência de um sinal
  - Filtro passa-baixa
    - rejeita as altas freqüências do sinal
  - Filtro passa-alta
    - rejeita as baixas freqüências do sinal

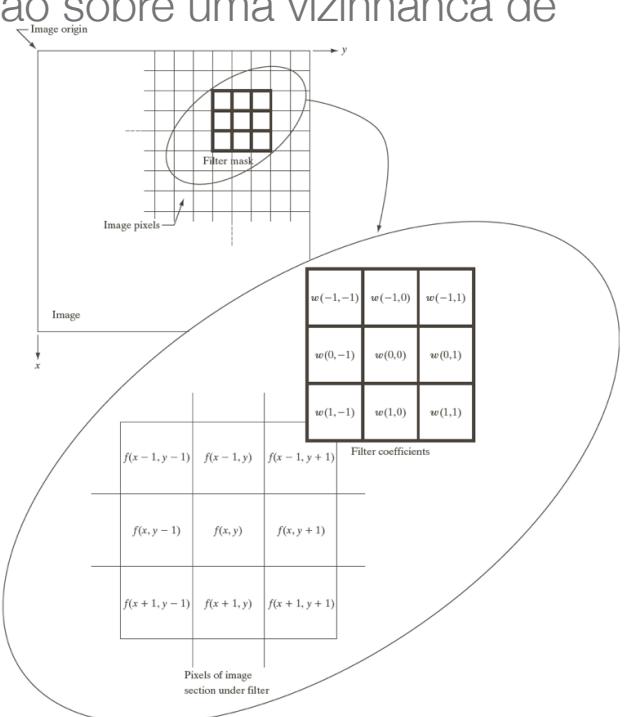
# Funcionamento de um Filtro Espacial

- Tipos de Filtros:



# Funcionamento de um Filtro Espacial

- Aplicação de uma certa operacão sobre uma vizinhanca de cada pixel da imagem



## Funcionamento de um Filtro Espacial

---

- Considerando:
  - Imagem M x N
  - Filtro m x n
    - Dimensão ímpar: m=2a+1, n=2b+1

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + w(-1,1)f(x-1,y+1) + \\ w(0,-1)f(x,y-1) + w(0,0)f(x,y) + w(0,1)f(x,y+1) + \\ w(1,-1)f(x+1,y-1) + w(1,0)f(x+1,y) + w(1,+1)f(x+1,y+1) +$$

$$g(x,y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s,t)f(x+s,y+t)$$

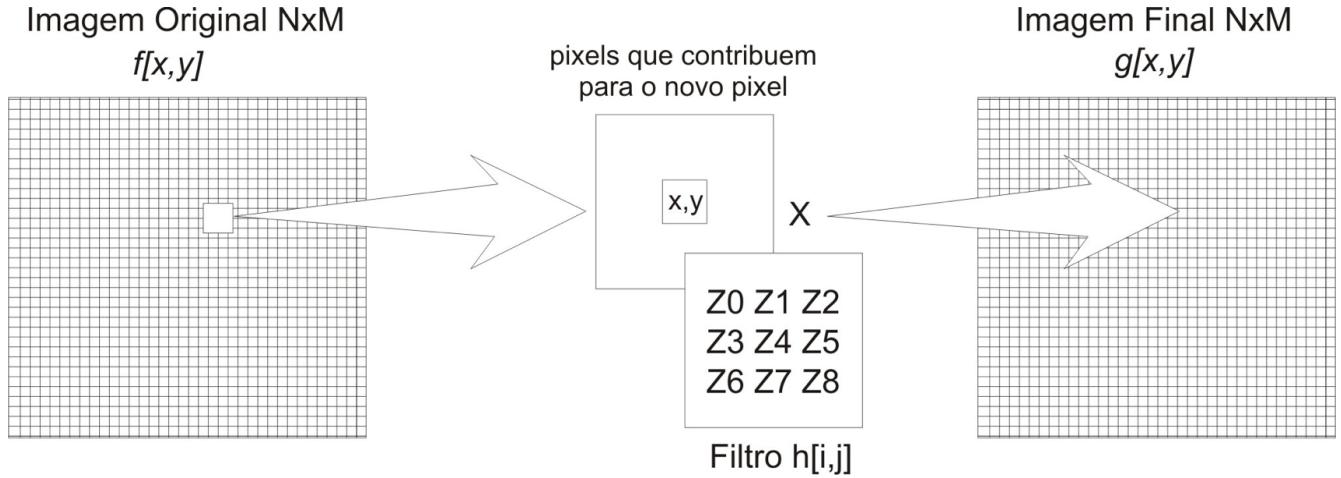
## Funcionamento de um Filtro Espacial

---

- Filtros Lineares
  - Máscaras que realizam somas ponderadas das intensidade de pixels ao longo da imagem
- Filtros não-lineares
  - Baseiam-se diretamente nos valores de pixels e não em pesos
- Mesma máscara aplicada ao longo da imagem
  - Filtro espacialmente invariante

# Funcionamento de um Filtro Espacial

- Resumo do processo:



## Convolução

- Convolução
  - filtro é pré-rotacionada de 180º
  - gera uma versão idêntica ao filtro quando aplicada a um impulso unitário

$$w(x,y) \bullet f(x,y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s,t) f(x-s, y-t)$$

- Representação vetorial:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_m z_m =$$

$$= \sum_{k=1}^m w_k z_k = \mathbf{w}^T \mathbf{z}$$

# Máscaras de Filtragem Espacial

- Gerar um filtro espacial linear  $m \times n$ :
  - definir os  $mn$  coeficientes da máscara
- Exemplo 1:
  - calcular a intensidade média na vizinhança  $3 \times 3$  do pixel

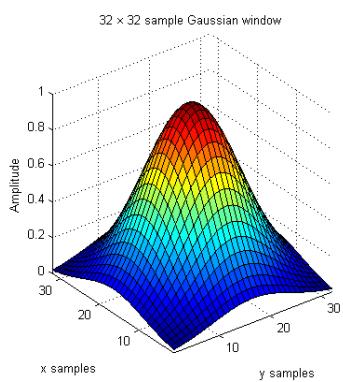
$$R = \frac{\sum_{k=1}^9 z_k}{9}$$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

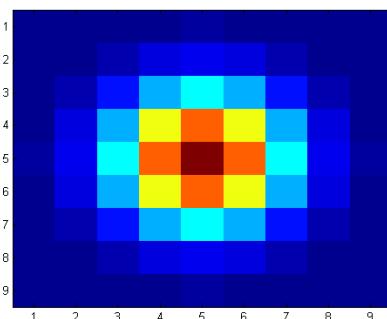
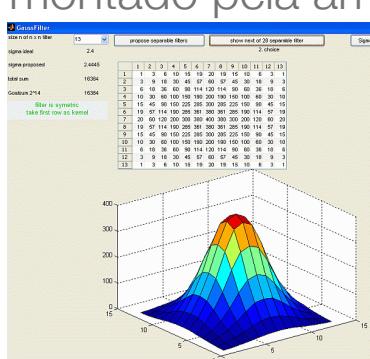
# Máscaras de Filtragem Espacial

- Exemplo 2:
  - Função Gaussiana

$$h(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



- O filtro pode ser montado pela amostragem de  $h(x,y)$ 
  - $w1=h(-1,1)$
  - $w2=h(-1,0)$
  - ....



## Máscaras de Filtragem Espacial

---

- Exemplo 3:
  - Filtro não linear:

$$R = \max(z_k) \quad k \in [0, mn]$$

## Filtros Espaciais de Suavização

## Filtros Espaciais de Suavização

---

- Utilizados para:
  - Borramento da imagem
  - Pré-processamento
    - Removendo pequenos detalhes
    - Conexão de pequenas descontinuidades em linhas e curvas
  - Redução de ruído

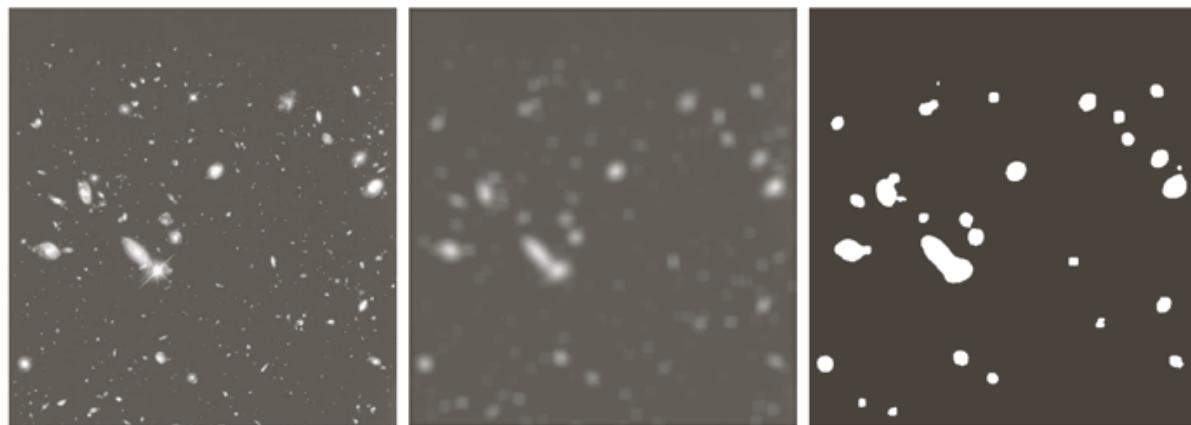
## Filtros Espaciais de Suavização

---

- Filtros lineares de suavização
  - Filtros de média ou Passa-baixa
    - Resposta é a média dos pixels na vizinhança da máscara de filtragem
  - Redução de transições abruptas
    - Podem ser causados por ruído
  - Redução de detalhes
  - Perda de nitidez da imagem
    - Efeito colateral ⇒ borrar bordas

## Filtros Espaciais de Suavização

- Filtros lineares de suavização (cont.)

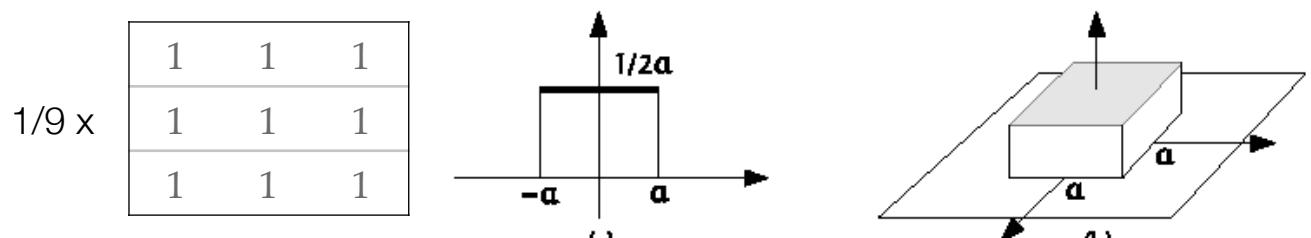


## Filtros Espaciais de Suavização

- Filtro nm de média aritmética simples:

$$R = \frac{\sum_{k=1}^{nm} z_k}{nm} = \frac{1}{nm} \sum_{k=1}^{nm} z_k$$

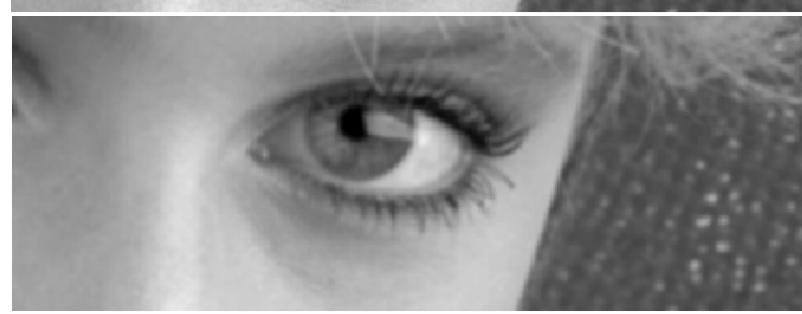
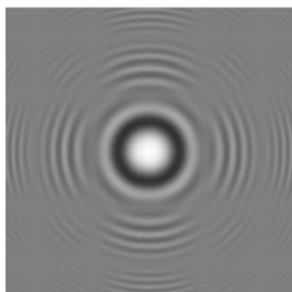
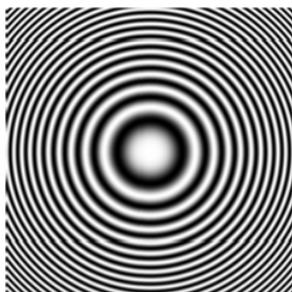
- Filtro Retangular ou Box
  - Exemplo: Box 3x3



## Filtros Espaciais de Suavização

---

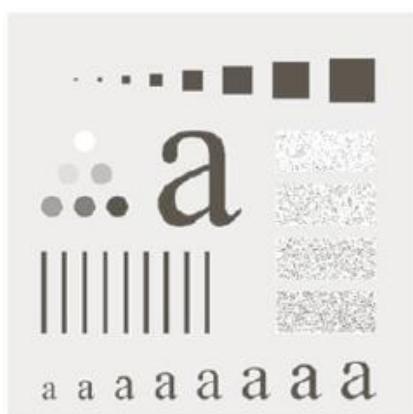
- Filtro retangular ou Box:



## Filtros Espaciais de Suavização

---

- Variação do tamanho da máscara



35x35

## Filtros Espaciais de Suavização

---

- Filtro mn de média ponderada:
  - Peso diferenciado para os pixels da máscara

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

## Filtros Espaciais de Suavização

---

- Filtro mn de média ponderada (cont.):
  - Exemplo:
    - pixel central com maior peso
    - demais ponderados por sua distância ao centro
    - reduzir o borramento no processo

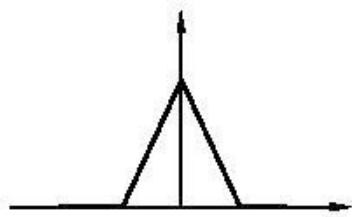
1/16 x

1	2	1
2	4	2
1	2	1

## Filtros Espaciais de Suavização

---

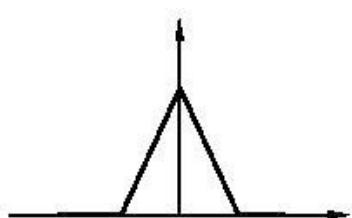
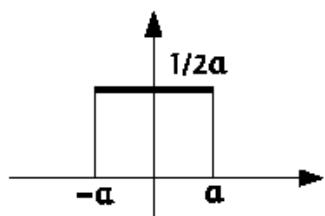
- Filtro mn de média ponderada (cont.):



## Filtros Espaciais de Suavização

---

- Filtros de média:



## Filtros Espaciais de Suavização

---

- Filtros de estatística de ordem
  - Filtros não lineares
  - Baseados na ordenação (classificação) dos pixels
  - Filtro de mediana
    - Produzem excelentes resultados na remoção de ruídos aleatórios.
    - Ruído “sal e pimenta”
    - Menor borramento da imagem

## Filtros Espaciais de Suavização

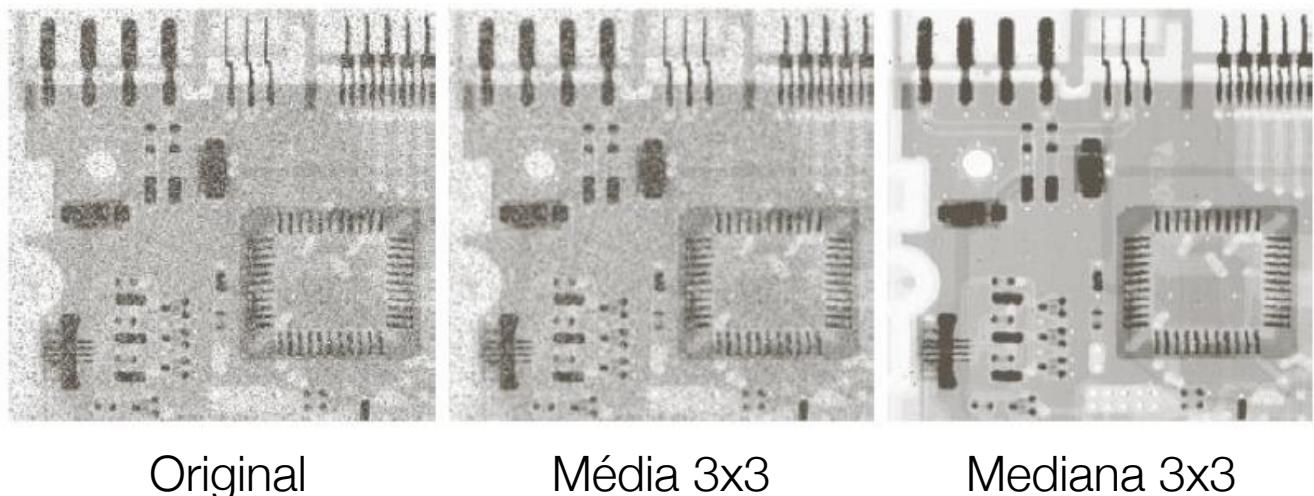
---

- Filtro de mediana (cont.)
  - Força que pontos distintos sejam mais semelhantes aos seus vizinhos
  - Agrupamentos isolados de área  $< m^2/2$  são eliminados por filtros  $m^2$
  - Agrupamentos maiores são menos afetados

## Filtros Espaciais de Suavização

---

- Filtros de mediana (cont.)
  - Exemplo:



## Filtros Espaciais de Suavização

---

- Filtros de estatística de ordem (cont.)
  - Mediana  $\Rightarrow$  50º percentil
  - Max  $\Rightarrow$  100º percentil
    - Filtro Max
    - identificar pontos mais claros
  - Min  $\Rightarrow$  0º percentil
    - Filtro Min
    - identificar pontos mais escuros

# Filtros Espaciais de Aguçamento

## Filtros Espaciais de Aguçamento

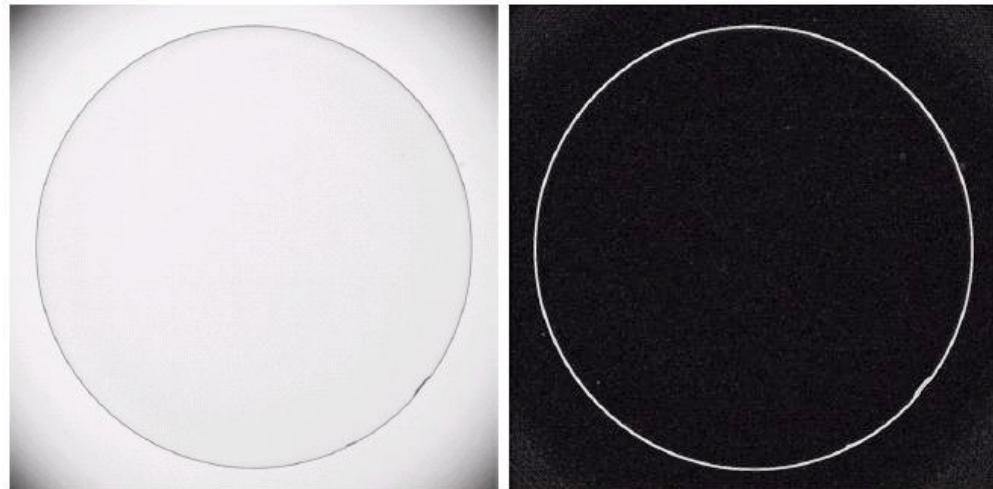
- Objetivo:
  - Realce das transições de intensidade
  - Aumento da nitidez da imagem



## Filtros Espaciais de Aguçamento

---

- Identificar as transições de intensidade



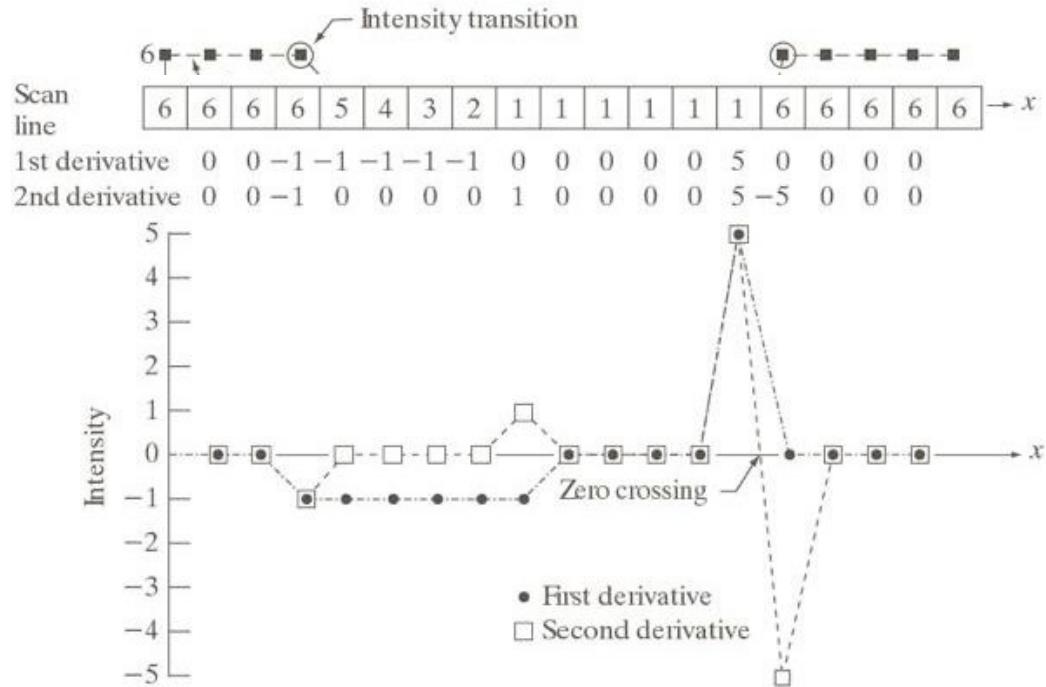
## Filtros Espaciais de Aguçamento

---

- Suavização
  - Realce de áreas contínuas
  - Atenuar diferenças
    - Média ⇒ Integração
- Aguçamento
  - Realce das descontinuidades
  - Acentuar bordas
    - Diferenças ⇒ Derivação

# Filtros Espaciais de Aguçamento

- Exemplo:



# Filtros Espaciais de Aguçamento

- Primeira derivada:
  - nula em regiões constantes
  - não nula em:
    - inicio de rampa ou degrau
    - ao longo de uma rampa
- Segunda derivada:
  - nula em
    - áreas constantes
    - ao longo de uma rampa de inclinação constante
  - não nula no inicio e no final de um degrau ou rampa

# Filtros Espaciais de Aguçamento

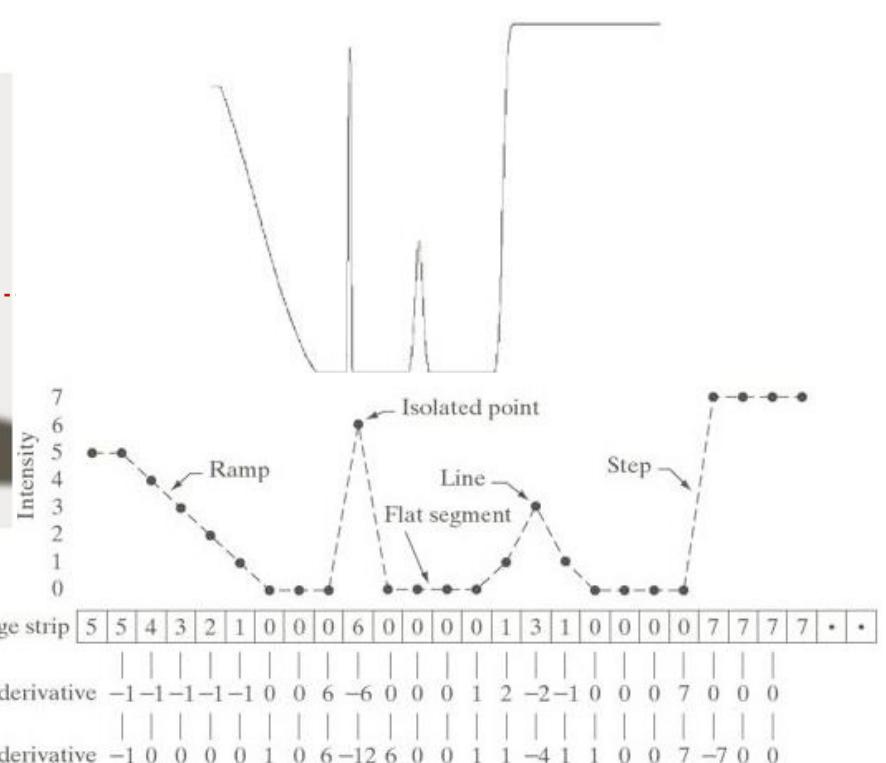
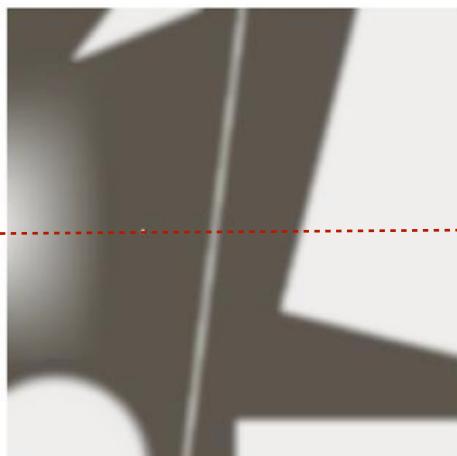
- Derivada de funções discretas
  - Diferenças entre amostras
  - Utilizando uma aproximação de primeira ordem da Série de Taylor, temos:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

# Filtros Espaciais de Aguçamento

- Exemplo:



## Filtros Espaciais de Aguçamento

---

- Da analise das derivadas podemos concluir:
  - a) O sinal da 2<sup>a</sup> derivada muda no inicio e fim de um degrau ou de uma rampa
  - b) Em uma transição do tipo degrau a 2<sup>a</sup> derivada aparece um cruzamento em zero (zero crossing)
  - c) Derivada primeira torna-se não nula ao longo de uma rampa

## Filtros Espaciais de Aguçamento

---

- Identificar arestas implica:
  - identificar rampas:
    - derivada 1<sup>a</sup> não nula
      - aresta da “largura” da rampa
    - derivada 2<sup>a</sup> não nula
      - dupla borda separada por zeros
  - identificar degraus:
    - zero crossing da derivada 2<sup>a</sup>

## Filtros Espaciais de Aguçamento

---

- Aguçamento utilizando a 2<sup>a</sup> derivada 2D:
  - Devemos considerar a orientação
    - filtro isotrópico
      - invariante a rotação
    - Operador Laplaciano 2D

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

## Filtros Espaciais de Aguçamento

---

- Aguçamento utilizando a 2a derivada 2D (cont.):

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\begin{aligned}\nabla^2 f = & f(x+1,y) + f(x-1,y) + \\ & + f(x,y+1) + f(x,y-1) - \\ & - 4f(x,y)\end{aligned}$$

## Filtros Espaciais de Aguçamento

---

- Máscara para o operador Laplaciano:

- isotrópico para rotações de 90°

0	1	0
1	-4	1
0	1	0

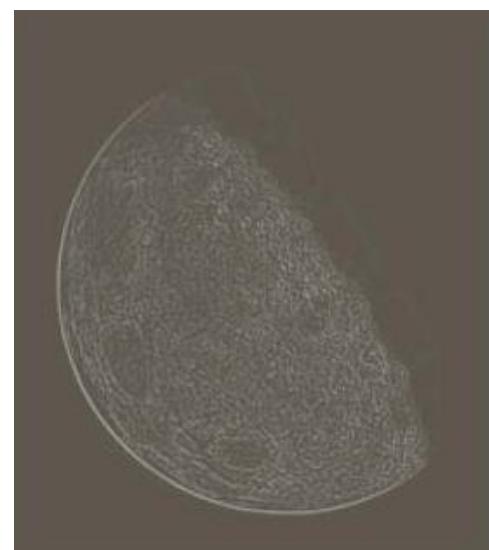
- Isotrópico para rotações de 45°
- inclusão das derivadas cruzadas

1	1	1
1	-8	1
1	1	1

## Filtros Espaciais de Aguçamento

---

- Exemplo: Aplicação do Laplaciano



## Filtros Espaciais de Aguçamento

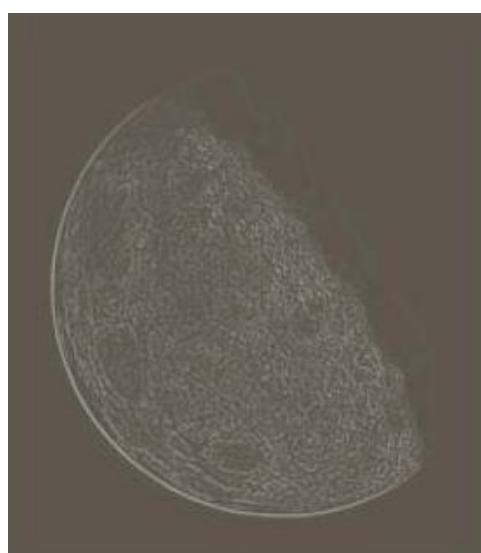
---

- Aplicação do Laplaciano pode gerar valores negativos:
  - normalizados para zero  $\Rightarrow$  regiões escuras
- Normalização para o intervalo  $[0, L-1]$

## Filtros Espaciais de Aguçamento

---

- Exemplo: Aplicação do Laplaciano com ajuste



## Filtros Espaciais de Aguçamento

---

- Características do Laplaciano
  - Operador linear
  - diferencial
    - realça as descontinuidades
      - tons de cinza
    - atenua regiões contínuas
      - tons escuros

## Filtros Espaciais de Aguçamento

---

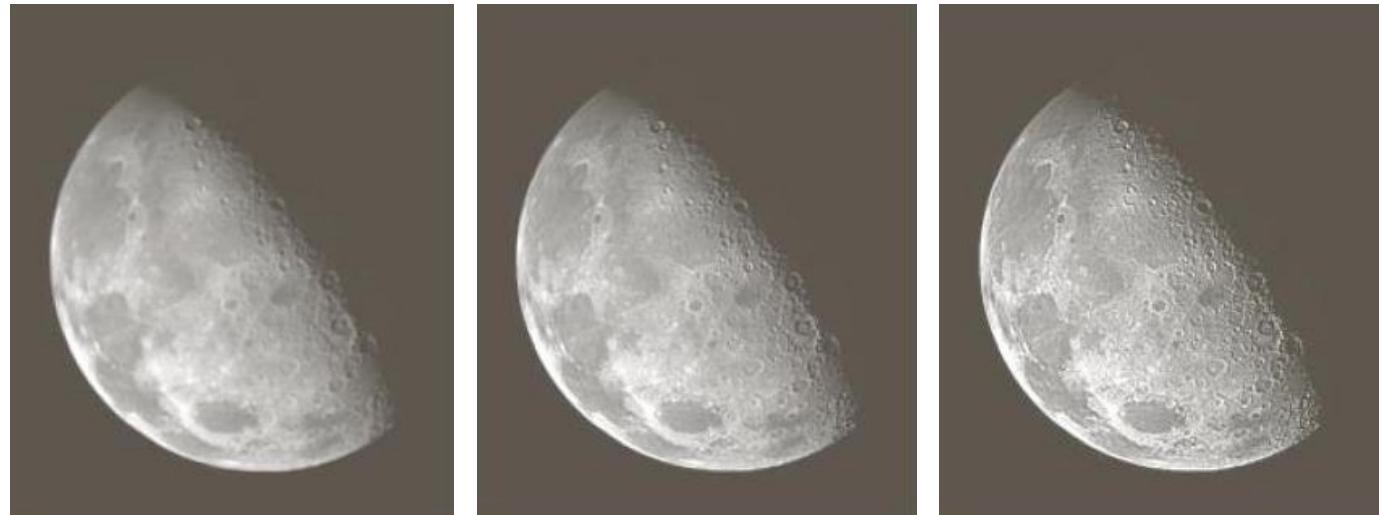
- Aguçamento:
  - Junção das linhas de borda e descontinuidades geradas pelo Laplaciano com as características de variação suave da imagem

$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)]$$

## Filtros Espaciais de Aguçamento

---

- Adicionando o Laplaciano isotrópico  $90^\circ$  e  $45^\circ$



## Filtros Espaciais de Aguçamento

---

- Aguçamento está associado ao aumento de nitidez na imagem
  - Aplicação de uma máscara de nitidez (*unsharp mask*)
    - Processo comum na industria gráfica
    - Consiste em:
      - subtrair uma versão não nítida da imagem original

## Filtros Espaciais de Aguçamento

---

- Máscara de nitidez envolve:
  - Borrar a imagem original
  - Subtrair a imagem borrada da original
    - resultado é a máscara de nitidez
  - Adicionar a máscara a imagem original

$$\bar{f}(x,y) = g_{suavização}(x,y)$$

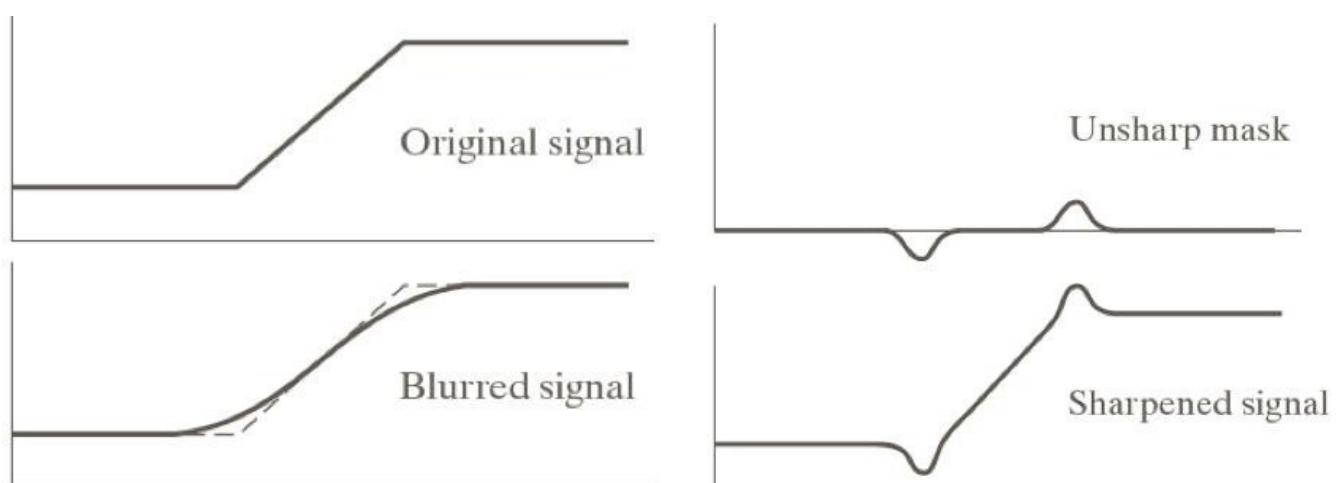
$$g_{máscara}(x,y) = f(x,y) - \bar{f}(x,y)$$

$$g(x,y) = f(x,y) + k * g_{máscara}(x,y)$$

## Filtros Espaciais de Aguçamento

---

- Funcionamento da máscara de nitidez:



## Filtros Espaciais de Aguçamento

---

- O valor da constante k define o quanto de aumento de nitidez será aplicado a imagem:
  - $k=1 \Rightarrow$  máscara de nitidez
  - $k>1 \Rightarrow$  filtragem high-boost
  - $k<1 \Rightarrow$  atenua a contribuição da máscara de nitidez

$$\bar{f}(x,y) = g_{suavização}(x,y)$$

$$g_{máscara}(x,y) = f(x,y) - \bar{f}(x,y)$$

$$g(x,y) = f(x,y) + k * g_{máscara}(x,y)$$

## Filtros Espaciais de Aguçamento

---

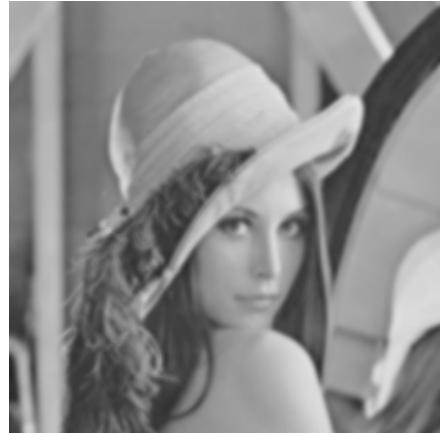
- Exemplo:



## Filtros Espaciais de Aguçamento

---

- Exemplo:



## Filtros Espaciais de Aguçamento

---

- Aguçamento não linear com derivada de 1<sup>a</sup> ordem
  - derivada de 1<sup>a</sup> ordem  $\Rightarrow$  magnitude do gradiente
  - aponta na direção da maior taxa de variação de  $f$  em  $(x,y)$

$$\nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

## Filtros Espaciais de Aguçamento

---

- O módulo do gradiente é dado por:

$$M(x,y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

- M é uma imagem de mesma dimensão de f
- As componentes de  $\nabla f$  são lineares, mas não seu módulo
- As componentes não são invariantes a rotações, mas o módulo é.
- Computacionalmente o modulo do gradiente pode ser aproximado por:

$$M(x,y) \approx |g_x| + |g_y|$$

## Filtros Espaciais de Aguçamento

---

- O gradiente discreto é definido por
  - $g_x = (z_6 - z_5)$
  - $g_y = (z_8 - z_5)$
- Roberts(65):
  - $g_x = (z_9 - z_5)$
  - $g_y = (z_8 - z_6)$

z1	z2	z3
z4	z5	z6
z7	z8	z9

## Filtros Espaciais de Aguçamento

---

- Usando Roberts:

$$M(x,y) = \left[ (z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$$

- Operadores gradientes diagonais de Roberts:

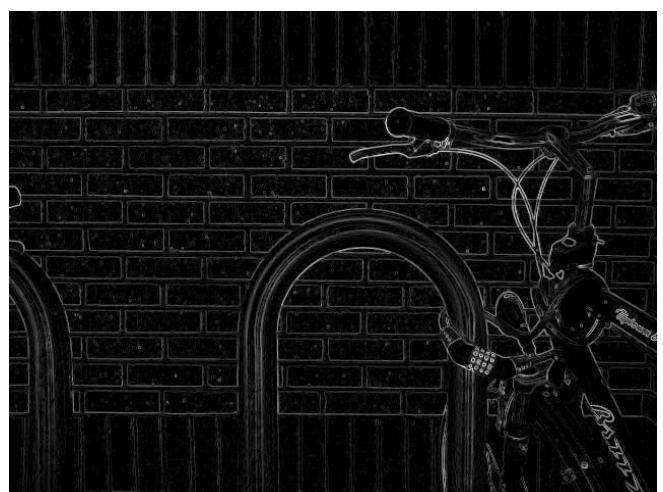
-1	0
0	1

0	-1
1	0

## Filtros Espaciais de Aguçamento

---

- Roberts



## Filtros Espaciais de Aguçamento

- Utilizando uma vizinhança 3x3 temos:
- Operadores de Sobel:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$M(x,y) = \left[ ((z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3))^2 + ((z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7))^2 \right]^{\frac{1}{2}}$$

$$M(x,y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

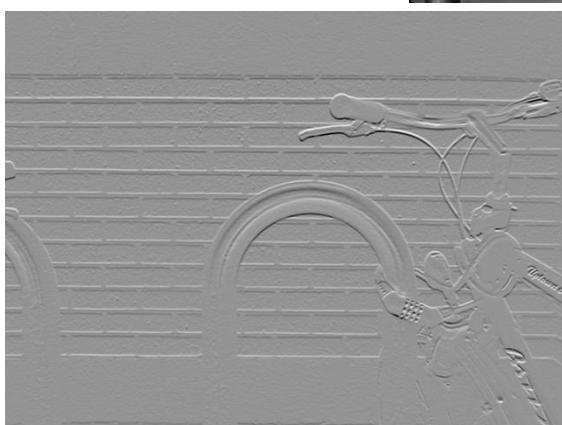
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

- Soma dos coeficientes de Sobel resulta nula:
  - geram resposta 0 em regiões de intensidade constante

## Filtros Espaciais de Aguçamento

- Sobel



## Filtros Espaciais de Aguçamento

---

- Sobel



## Filtros Espaciais de Aguçamento

---

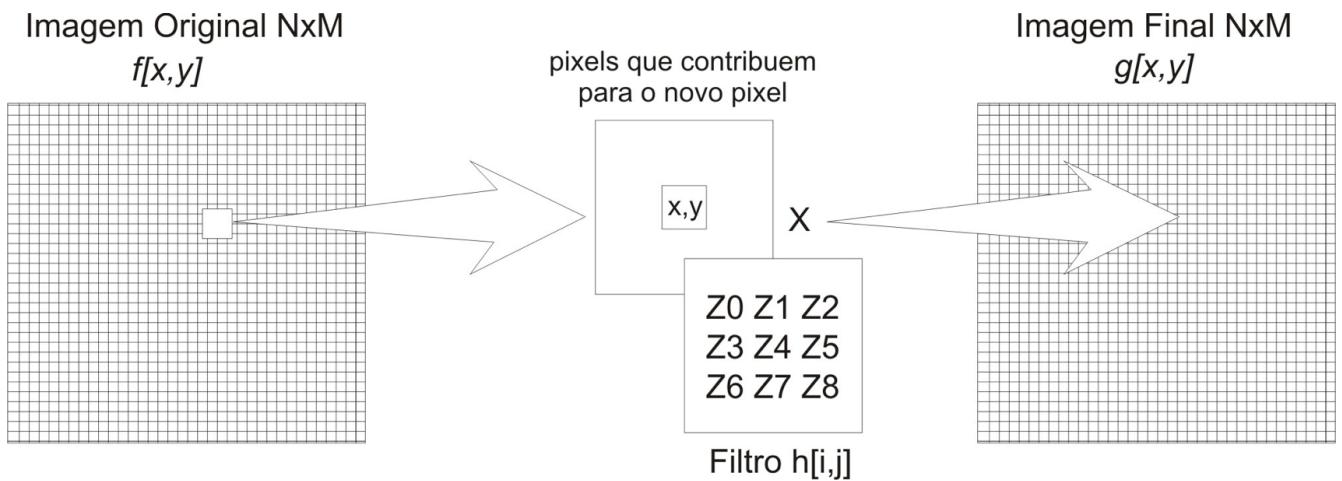
- Roberts x Sobel



## Aplicação em Three.js/WebGL

## Filtragem Espacial em WebGL

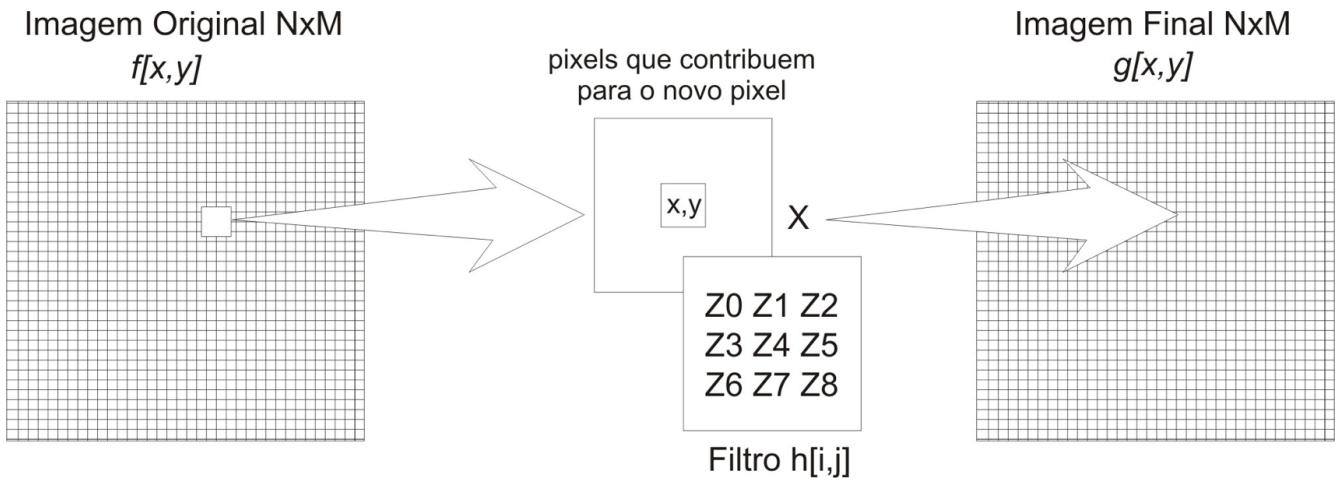
- Acesso as intensidades dos pixels em uma vizinhança
  - Dado um pixel como calcular a sua coordenada?



## Filtragem Espacial em WebGL

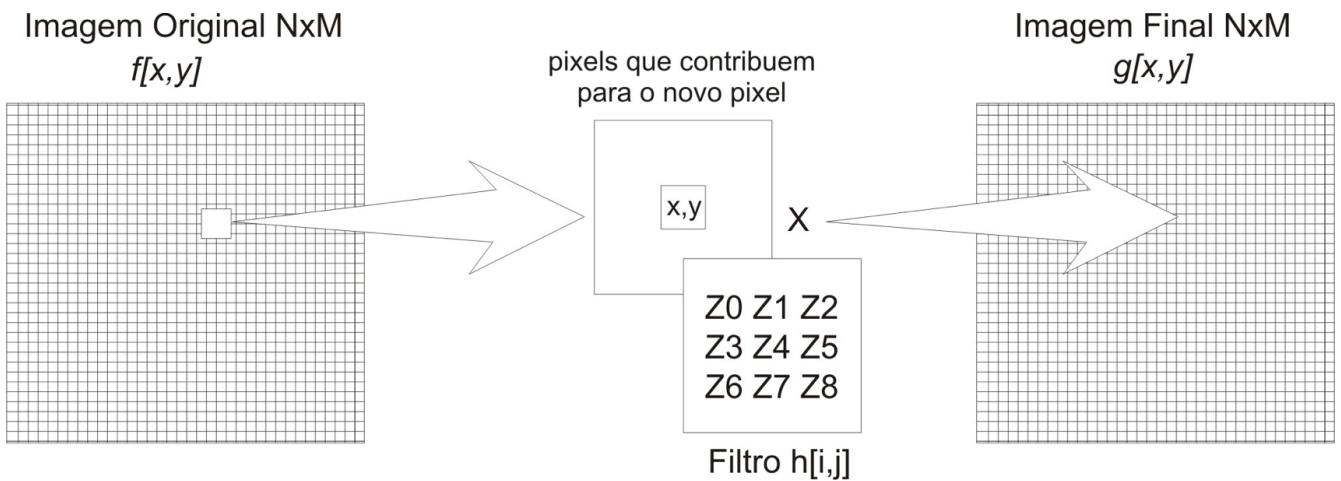
- Dado um pixel como calcular a sua coordenada?

**`varying vec2 vUv;`**



## Filtragem Espacial em WebGL

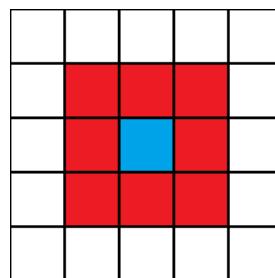
- Dado um pixel como calcular as coordenadas de seus vizinhos?



## Filtragem Espacial em WebGL

---

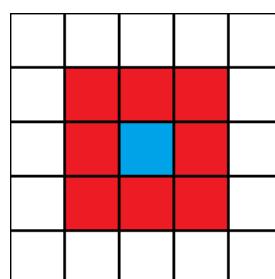
- Dado um pixel como calcular as coordenadas de seus vizinhos?



## Filtragem Espacial em WebGL

---

- Dado um pixel como calcular as coordenadas de seus vizinhos?
  - Calcular o "tamanho" do pixel
    - Considere uma imagem 5x5
    - coordenadas no intervalo [0.0, 1.0]
    - 1 pixel tem “dimensões”  $1/5 \times 1/5$



# Filtragem Espacial em WebGL

---

- Dado um pixel como calcular as coordenadas de seus vizinhos?
  - Em uma imagem qualquer:

```
uPixelSize= new THREE.Vector2( 1.0/texture.image.width,
                               1.0/texture.image.height)
```

- No *shader*:

```
uniform vec2 uPixelSize;
(...)
texture2D(uSampler, vUv+vec2(uPixelSize.x,uPixelSize.y))
```

# Filtragem Espacial em WebGL

---

- Resumindo:
  - Programa Javascript:

```
function onLoadTexture() {
    renderer.setSize(texture.image.width, texture.image.height);
    uniforms = {
        uSampler: { type: "t", value:texture },
        uPixelSize: { type: "v2", value: new THREE.Vector2(
                        1.0/texture.image.width,
                        1.0/texture.image.height) }
    };
    var matShader = new THREE.ShaderMaterial( {
        uniforms: uniforms,
        vertexShader: document.getElementById( 'suaviza-vs' ).textContent,
        fragmentShader: document.getElementById( 'suaviza-fs' ).textContent
    } );
    (...)
```

# Filtragem Espacial em WebGL

---

- Resumindo:

- *Shader:*

```
<script id="media-fs" type="x-shader/x-fragment">
    precision mediump float;

    uniform sampler2D uSampler;
    uniform vec2 uPixelSize;
    varying vec2 vUv;

    void main(void) {
        vec4 c = (
            texture2D(uSampler, vUv) +
            texture2D(uSampler, vUv + vec2(-uPixelSize.x, 0.0)) +
            texture2D(uSampler, vUv + vec2(uPixelSize.x, 0.0)) +
            texture2D(uSampler, vUv + vec2(0.0, -uPixelSize.y)) +
            texture2D(uSampler, vUv + vec2(0.0, uPixelSize.y)) +
            texture2D(uSampler, vUv + vec2(uPixelSize.x, uPixelSize.y)) +
            texture2D(uSampler, vUv + vec2(-uPixelSize.x, uPixelSize.y)) +
            texture2D(uSampler, vUv + vec2(uPixelSize.x, -uPixelSize.y)) +
            texture2D(uSampler, vUv + vec2(-uPixelSize.x, -uPixelSize.y))
        ) / 9.0;
        gl_FragColor = c;
    }
</script>
```

A seguir...

Técnicas de Renderização Baseadas em  
Mapeamentos