

# PROCESSAMENTO DIGITAL DE IMAGENS

## PDI – Aula 10

Universidade Federal do Rio Grande do Norte  
Unidade Acadêmica Especializada em Ciências Agrárias  
Escola Agrícola de Jundiaí  
Tecnologia em Análise e Desenvolvimento de Sistemas  
**Profa. Alessandra Mendes**

# Filtragem Espacial

# Filtragem Espacial

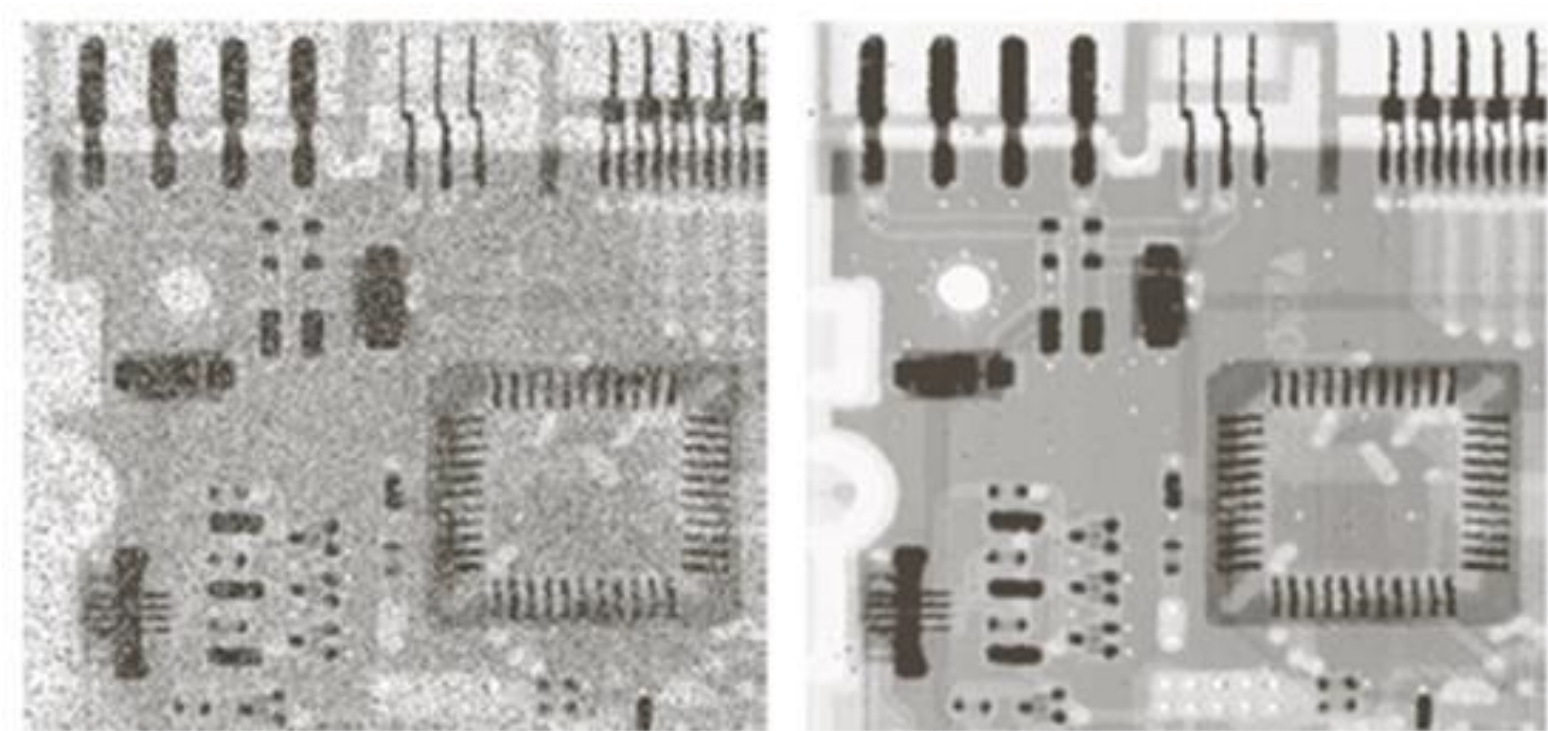
---

- ▶ A filtragem espacial é utilizada num grande variedade de aplicações, como aprimoramento da imagem, redução de ruído, aguçamento, detecção de borda, etc.
- ▶ A palavra *filtro* foi emprestada do processamento no domínio da frequência.
- ▶ Filtros espaciais são mais versáteis, pois é possível construir filtros não lineares.
- ▶ Se a operação realizada sobre os pixels é linear, o filtro é chamado de filtro espacial linear, caso contrário, é chamado filtro não-linear.

# Filtragem Espacial

---

- ▶ Exemplo: remoção de ruído



# Filtragem Espacial

---

- ▶ A filtragem espacial realiza **operações diretamente na imagem** usando **filtros espaciais** (também chamados de máscaras, *kernels*, *templates*, ou janelas).
- ▶ O processo de filtragem consiste em:
  - ▶ Uma **vizinhança**;
  - ▶ Uma **operação** pré-definida que é executada sobre os pixels da imagem que interceptam a vizinhança.
- ▶ A filtragem cria um **novo pixel** com coordenadas iguais ao do centro da vizinhança e **cujo valor é resultado da operação de filtragem**.

# Filtragem Espacial

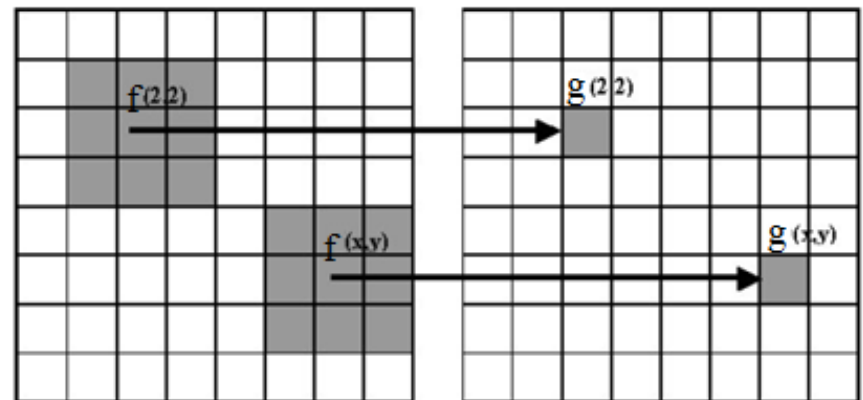
---

- ▶ O processamento sobre uma vizinhança consiste em:
  - 1) Definir um **ponto** central  $(x,y)$ ;
  - 2) Executar uma **operação** que envolva apenas os pixels da vizinhança pré-definida sobre o ponto central;
  - 3) Considerar o **resultado** da operação como sendo a resposta do processo no ponto  $(x,y)$ ;
  - 4) **Repetir** o processo para todos os pontos da imagem.
- ▶ O processo de mover o ponto central cria novas vizinhanças para cada pixel na imagem de entrada.  
**Esta operação é referida como processamento de vizinhança ou filtragem espacial.**

# Processo de Filtragem

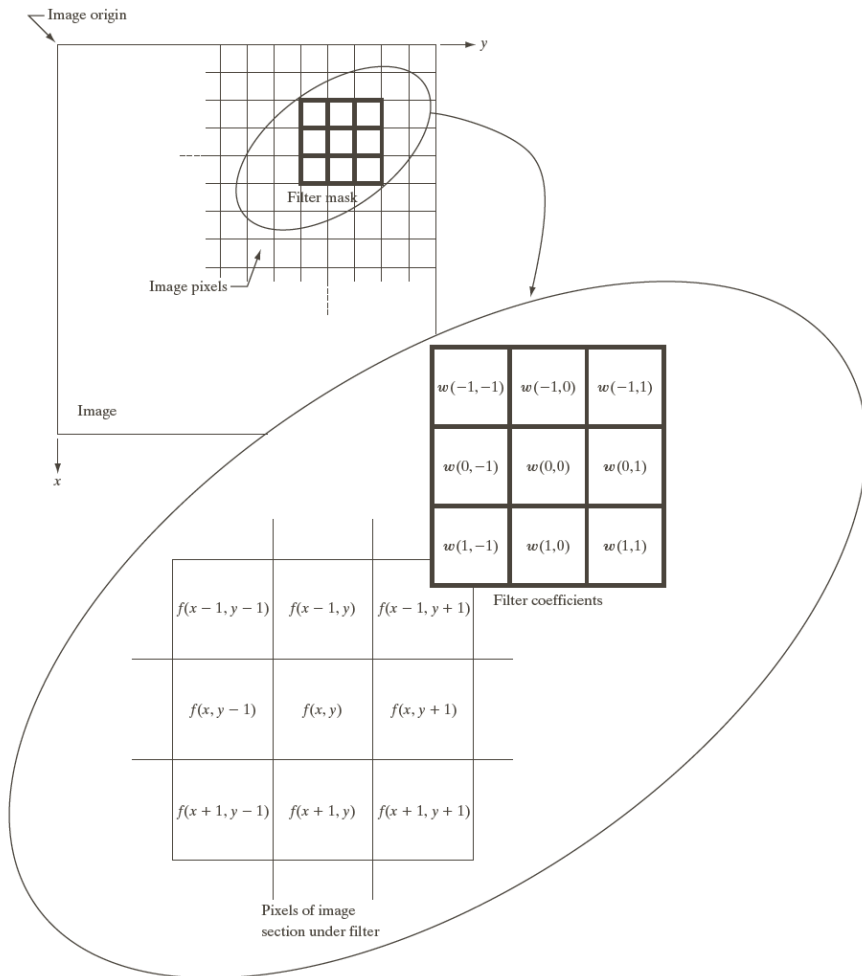
- ▶ *Cada elemento da máscara é multiplicado pelo valor do pixel correspondente na imagem  $f$*
- ▶ *A soma desses resultados é o novo valor do nível de cinza na nova imagem  $g$* 
  - ▶ Exemplo:  $w$  é uma janela de  $n \times n = k$  pixels.
  - ▶ O processo de filtragem para cada pixel na imagem  $g(x,y)$  será dada por:

$$g(x, y) = \sum_{i=1}^k w_i \cdot f(x, y)$$



# Filtragem Espacial

- ▶ Exemplo de mecânica de filtragem espacial linear usando máscara 3x3.
- ▶ A forma escolhida para denotar as coordenadas dos coeficientes da máscara de filtragem simplifica a escrita de expressões para filtragem linear.





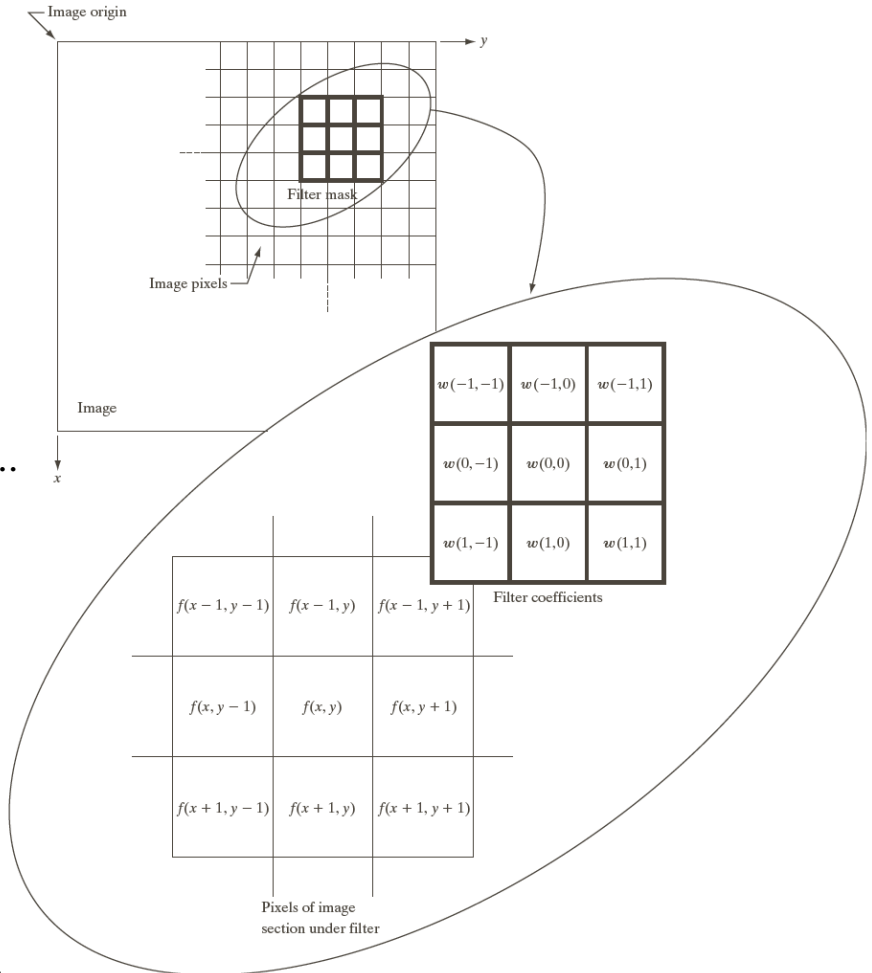
# Filtragem Espacial

- ▶ Em qualquer ponto  $f(x,y)$ , a resposta  $g(x,y)$  da filtragem é a soma do produtos entre os coeficientes e as intensidades dos pixels.

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

- ▶ Observa-se que o coeficiente central do filtro,  $w(0,0)$ , alinha com o pixel na posição  $f(x,y)$ .

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



# Correlação e Convolução

---

- ▶ Correlação e convolução, operações matemáticas em funções, são conceitos próximos da filtragem espacial.
- ▶ **Correlação** é o processo de **mover uma máscara de filtro sobre uma imagem e computar a soma dos produtos em cada posição**, exatamente como explicado anteriormente.

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

- ▶ A **convolução** difere da correlação pela **rotação do filtro de 180°**.

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

# Correlação e Convolução

- ▶ Os pontos (a,b,c,d,e,f,g,h,i) são os valores dos níveis de cinza na mesma vizinhança de  $f(x,y) = e$ , comparando com a máscara.
- ▶ Os valores  $w_1$  à  $w_9$  são os “pesos”, ou seja, os valores dos níveis de cinza em cada posição da máscara.

Imagem -  $f(x,y)$

|   |   |   |  |
|---|---|---|--|
|   |   |   |  |
| a | b | c |  |
| d | e | f |  |
| g | h | i |  |
|   |   |   |  |

x

y

Máscara

$K = 3 \times 3 = 9$

|       |       |       |
|-------|-------|-------|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

$$g(x, y) = \sum_{i=1}^k w_i \cdot f(x, y)$$

- ▶ O valor do pixel  $g(x,y)$  na nova imagem será dado por:

$$g(x, y) = w_1 \cdot a + w_2 \cdot b + w_3 \cdot c + w_4 \cdot d + w_5 \cdot e + w_6 \cdot f + w_7 \cdot g + w_8 \cdot h + w_9 \cdot i$$

# Correlação e Convolução

- Mecânica de filtragem em uma imagem com filtro de núcleo  $N \times N = 3 \times 3$ , onde  $I_k(i)$  representa os pixels vizinhos ao  $i$ -ésimo pixel da imagem e  $k$  é um índice linear que varre a região da vizinhança segundo uma convenção de linha ou coluna.

$$f_i = \sum_{k=1}^9 w_k I_k(i) =$$

$$(-1 \times 10) + (-1 \times 11) + (-1 \times 8) + (-1 \times 40) + (8 \times 35) + (-1 \times 42) + (-1 \times 38) + (-1 \times 36) + (-1 \times 46) = 14$$

|       |       |       |
|-------|-------|-------|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

=

|    |    |    |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8  | -1 |
| -1 | -1 | -1 |

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 12 | 11 | 12 | 13 | 13 | 9  |
| 10 | 8  | 10 | 11 | 8  | 13 |
| 32 | 36 | 40 | 35 | 42 | 40 |
| 40 | 37 | 38 | 36 | 46 | 41 |
| 41 | 36 | 89 | 39 | 42 | 39 |
| 42 | 37 | 39 | 43 | 45 | 38 |

# Correlação e Convolução

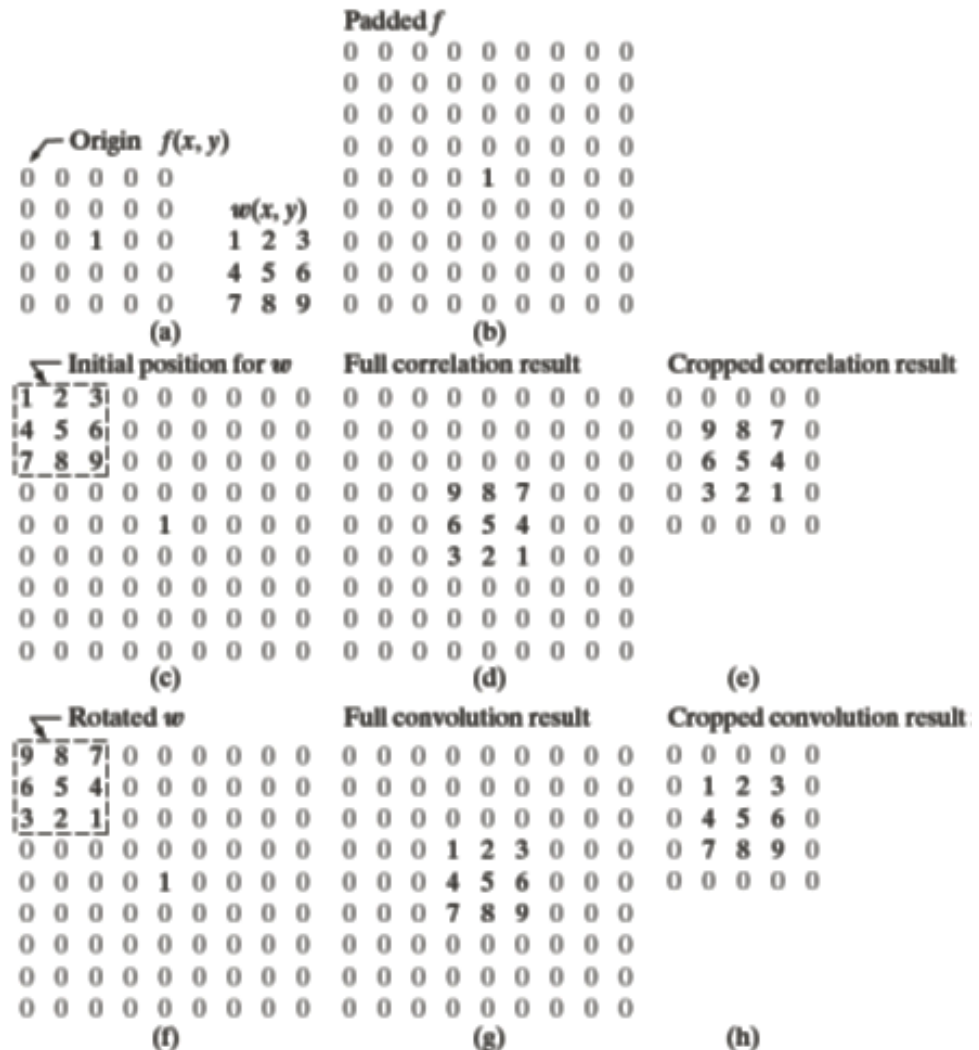
---

## ► Detalhamento do processo:

- 1) **Definir** o núcleo do filtro;
- 2) **Deslizar** o núcleo sobre a imagem de modo que o pixel central do núcleo coincida com cada pixel-alvo da imagem;
- 3) **Multiplicar** os pixels sob o núcleo pelos correspondentes valores (pesos) no núcleo e somar os resultados;
- 4) Para cada pixel-alvo, **copiar** o valor resultante na mesma posição de uma nova imagem (filtrada).

# Correlação e Convolução

- ▶ **Correlação** (linha do meio) e **convolução** (última linha) de um filtro 2D com um impulso unitário discreto 2D.
- ▶ Os 0's são mostrados em cinza para facilitar a análise visual.



# Correlação e Convolução

## ► Convenção:

- Máscaras de organização par (2 x 2, 4 x 4 , ..... ) o resultado é colocado sobre o **Primeiro Pixel**.
- Máscaras de organização ímpar ( 3 x 3, 5 x 5, ..... ) o resultado é colocado sobre o **Pixel de Centro**.
- Nas máscaras simétricas, as operações de correlação e convolução são idênticas. Exemplo:

|   |                 |   |   |   |   |                   |   |   |   |   |   |   |   |   |
|---|-----------------|---|---|---|---|-------------------|---|---|---|---|---|---|---|---|
|   | Imagem Original |   |   |   |   | Imagem Final      |   |   |   |   |   |   |   |   |
| Template  |                 |   |   |   |   |                   |   |   |   |   |   |   |   |   |
| <table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table> | 1               | 0 | 0 | 1 | 1 | 1                 | 3 | 3 | 4 | 2 | 5 | 7 | 6 | * |
| 1   | 0               |   |   |   |   |                   |   |   |   |   |   |   |   |   |
| 0   | 1               |   |   |   |   |                   |   |   |   |   |   |   |   |   |
|   | 1               | 1 | 4 | 4 | 3 | 2                 | 4 | 7 | 7 | * |   |   |   |   |
|   | 2               | 1 | 3 | 3 | 3 | 3                 | 2 | 7 | 7 | * |   |   |   |   |
|   | 1               | 1 | 1 | 4 | 4 | *                 | * | * | * | * |   |   |   |   |
| $T(i,j)$  | $f(x,y)$        |   |   |   |   | $T(i,j) * f(x,y)$ |   |   |   |   |   |   |   |   |

Os valores marcados com \* não podem ser calculados.

# Limites da Imagem

---

- ▶ Os limites da imagem devem ser propriamente tratados:
  - ▶ **Ignorar** os pixels para os casos em que a operação não possa ser realizada – borda não processada;
    - ▶ Possivelmente atribuindo um valor fixo (podendo ser zero) aos resultados não calculáveis;
  - ▶ **Expandir** a imagem criando linhas colunas e preenchendo:
    - ▶ Com zeros antes do cálculo da imagem final;
    - ▶ Com a replicação dos pixels das bordas;
    - ▶ Com o espelhamento dos pixels da borda (simetria).
  - ▶ **Utilizar uma máscara modificada** nas regiões de borda, o que aumenta complexidade da operação.



# Limites da Imagem

- ▶ Exemplo: Atribuição de valor fixo – zero – aos resultados não calculáveis.

Template

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Imagem

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 0 | 1 | 3 | 4 | 0 |
| 1 | 1 | 3 | 2 | 0 |
| 0 | 0 | 4 | 5 | 6 |
| 1 | 0 | 7 | 8 | 0 |

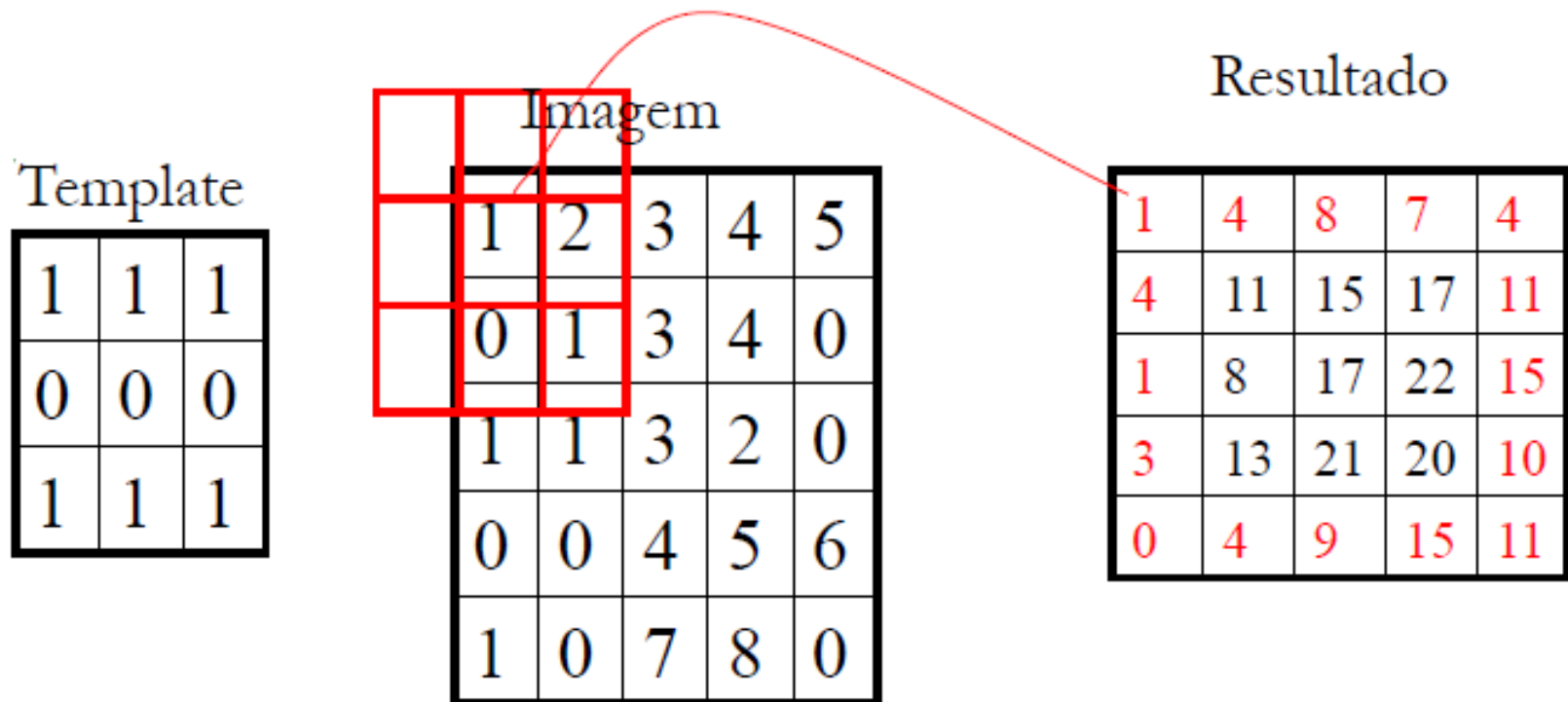
Resultado

|   |    |    |    |   |
|---|----|----|----|---|
| 0 | 0  | 0  | 0  | 0 |
| 0 | 11 | 15 | 17 | 0 |
| 0 | 8  | 17 | 22 | 0 |
| 0 | 13 | 21 | 20 | 0 |
| 0 | 0  | 0  | 0  | 0 |

Primeiro Ponto ==>  $(1 \times 1) + (1 \times 2) + (1 \times 3) + (0 \times 0) + (0 \times 1) + (0 \times 3) + (1 \times 1) + (1 \times 1) + (1 \times 3) = 11$

# Limites da Imagem

- ▶ Exemplo: Preenchimento com zeros antes do cálculo da imagem final.



**Primeiro Ponto** ==>  $(1 \times 0) + (1 \times 0) + (1 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 2) + (1 \times 0) + (1 \times 0) + (1 \times 1) = 1$

# Observações

---

- ▶ O **custo computacional** da correlação/convolução espacial é **alto**.
- ▶ Se a Imagem é de tamanho  $M \times M$  e a máscara  $N \times N$ , o número de multiplicações é de  $M^2 \cdot N^2$ , ou seja, se a Imagem é de  $512 \times 512$  e a máscara é de  $16 \times 16$ , são necessárias 67.108.864 multiplicações.
- ▶ A **alternativa é transformar a Imagem e a máscara para o domínio da frequência** (Fourier) e multiplicar elemento a elemento.
- ▶ A transformação **só é justificável se a máscara for maior que  $32 \times 32$** , devido ao custo da Transformada de Fourier.

# Filtragem no domínio espacial

---

- ▶ Os métodos de filtragem que trabalham no domínio espacial **operam diretamente sobre os pixels**, normalmente utilizando operações de **convolução** com máscaras.
- ▶ O uso de **máscaras** nas imagens no domínio espacial é usualmente chamado de filtragem espacial e as máscaras são chamadas de **filtros espaciais**.
- ▶ Filtragem:
  - ▶ Filtros espaciais de suavização – **passa-baixa**
  - ▶ Filtros espaciais de aguçamento – **passa-alta**
  - ▶ Filtros derivativos

# Filtragem no domínio espacial

---

## ► Filtragem:

$$g(x_i, y_i) = T[f(x_i, y_i)]$$

Onde:

$f(x_i, y_i)$  é a imagem de entrada a ser filtrada,

$g(x_i, y_i)$  é a imagem na saída, processada, e,

$T$  é um operador sobre  $f$ , definido em alguma vizinhança do *pixel* de posição  $(x_i, y_i)$ .

# Filtros passa-baixa

---

## ► Objetivos:

- **Suavizar** a imagem pela **redução das variações** nos de níveis de cinza que dão à aparência de “serrilhado” nos patamares de intensidade.
- **Atenuar as altas frequências**, que correspondem às transições abruptas.
- **Minimizar ruídos**.

## ► Tipos:

- Lineares: filtros de média ou média ponderada
- Não lineares: filtros de estatística de ordem (mediana, ordem, moda)

# Filtro passa-baixa: Linear

---

- ▶ A saída de um filtro espacial linear de suavização é a **média dos pixels contidos na vizinhança** da máscara de filtragem (filtro de média).
- ▶ Ao substituir o valor de cada pixel pela média, o resultado é uma imagem com **perda de nitidez**.
- ▶ Aplicações mais evidentes: **redução de ruído** e **suavização**.
- ▶ Efeito indesejado: **borramento das bordas**.
- ▶ Uma máscara de média é tal que seus pesos são positivos e a soma é igual a 1.
- ▶ Exemplos:  $\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$     $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$     $\frac{1}{32} \begin{bmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{bmatrix}$     $\frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

# Filtro passa-baixa: Linear

- Média (não ponderada) com máscara =  $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$f(x,y)$

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 20 | 30 | 24 | 34 | 60 | 80 | 89 | 90 | 12 | 00 |
| 23 | 24 | 56 | 67 | 88 | 99 | 00 | 00 | 00 | 00 |
| 12 | 23 | 35 | 65 | 66 | 77 | 88 | 99 | 00 | 00 |
| 11 | 22 | 99 | 99 | 99 | 99 | 99 | 98 | 88 | 88 |
| 12 | 12 | 12 | 22 | 22 | 44 | 55 | 65 | 77 | 88 |
| 11 | 44 | 55 | 76 | 87 | 55 | 66 | 33 | 33 | 33 |
| 12 | 33 | 44 | 55 | 66 | 77 | 88 | 00 | 00 | 00 |

$g(x,y)$

|  |    |    |  |  |  |  |  |  |  |
|--|----|----|--|--|--|--|--|--|--|
|  |    |    |  |  |  |  |  |  |  |
|  | 25 | 40 |  |  |  |  |  |  |  |
|  |    |    |  |  |  |  |  |  |  |
|  |    |    |  |  |  |  |  |  |  |
|  |    |    |  |  |  |  |  |  |  |
|  |    |    |  |  |  |  |  |  |  |
|  |    |    |  |  |  |  |  |  |  |

$$g(0,0) = (20 + 30 + 24 + 23 + 24 + 56 + 12 + 23 + 35) / 9 = 24,77$$

$$g(0,1) = (30 + 24 + 34 + 24 + 56 + 67 + 23 + 35 + 65) / 9 = 39,77$$



# Filtro passa-baixa: Linear

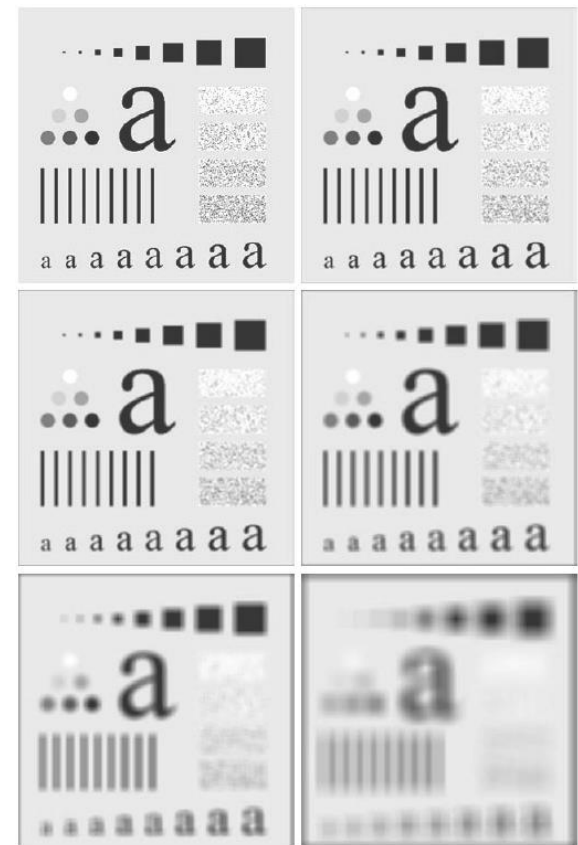
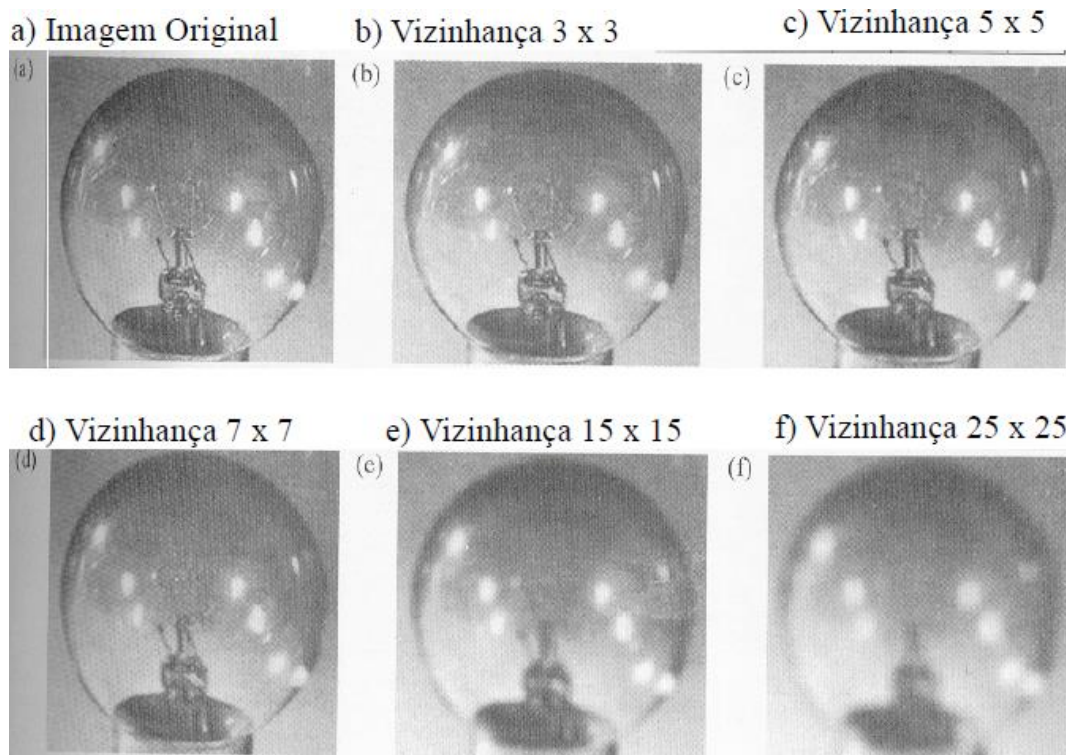
- Média ponderada com máscara =  $\frac{1}{8} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

|          |    |    |    |    |    |    |    |    |    |          |  |  |  |  |  |  |  |  |  |
|----------|----|----|----|----|----|----|----|----|----|----------|--|--|--|--|--|--|--|--|--|
| $f(x,y)$ |    |    |    |    |    |    |    |    |    | $g(x,y)$ |  |  |  |  |  |  |  |  |  |
| 20       | 30 | 24 | 34 | 60 | 80 | 89 | 90 | 12 | 00 |          |  |  |  |  |  |  |  |  |  |
| 23       | 24 | 56 | 67 | 88 | 99 | 00 | 00 | 00 | 00 |          |  |  |  |  |  |  |  |  |  |
| 12       | 23 | 35 | 65 | 66 | 77 | 88 | 99 | 00 | 00 |          |  |  |  |  |  |  |  |  |  |
| 11       | 22 | 99 | 99 | 99 | 99 | 99 | 98 | 88 | 88 |          |  |  |  |  |  |  |  |  |  |
| 12       | 12 | 12 | 22 | 22 | 44 | 55 | 65 | 77 | 88 |          |  |  |  |  |  |  |  |  |  |
| 11       | 44 | 55 | 76 | 87 | 55 | 66 | 33 | 33 | 33 |          |  |  |  |  |  |  |  |  |  |
| 12       | 33 | 44 | 55 | 66 | 77 | 88 | 00 | 00 | 00 |          |  |  |  |  |  |  |  |  |  |

$$g(0,0) = ((20*0)+(30*1)+(24*0)+(23*1)+(24*4)+(56*1)+(12*0) + (23*1)+(35*0))/8 = (0+30+0+23+96+56+0+23+0)/8 = 28,5$$

# Filtro passa-baixa: Linear

## ► Exemplos de aplicações de filtros de média:



# Filtro passa-baixa: Linear

- ▶ Exemplos de aplicações de filtros de média:



○  $1/9 x$

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

=



○  $1/25 x$

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

=



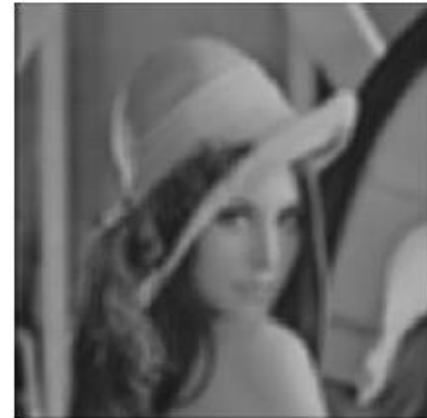
# Filtro passa-baixa: Linear

---

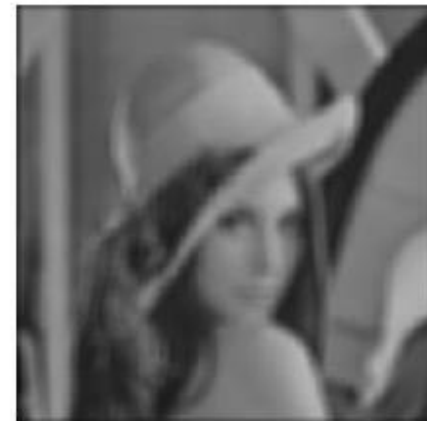
- ▶ Exemplos de aplicações de filtros de média:



○  $7 \times 7$  =



○  $9 \times 9$  =



# Filtro passa-baixa: Não Linear

---

- ▶ São filtros cujas **respostas** se baseiam na **ordenação dos pixels** contidos na área da imagem coberta pelo filtro.
  - ▶ O mais conhecido é o **filtro de mediana**, que substitui o valor de um pixel pela mediana dos valores da vizinhança.
  - ▶ São particularmente eficazes para **a redução de ruído impulsivo** (sal e pimenta).
  - ▶ A mediana  $\mathcal{E}$  de um conjunto de valores é um valor tal que **metade** dos valores do conjunto é **menor** que ele e a outra **metade** é **maior**.
    - ▶ Ex: vizinhança = 10,23,20,20,15,24,24,25,100;  $\mathcal{E} = 23$ .
  - ▶ Para realizar a filtragem, **ordena-se** os valores dos pixels da vizinhança, **calcula-se** a mediana e **atribui-se** ao pixel central.

# Filtro passa-baixa: Não Linear

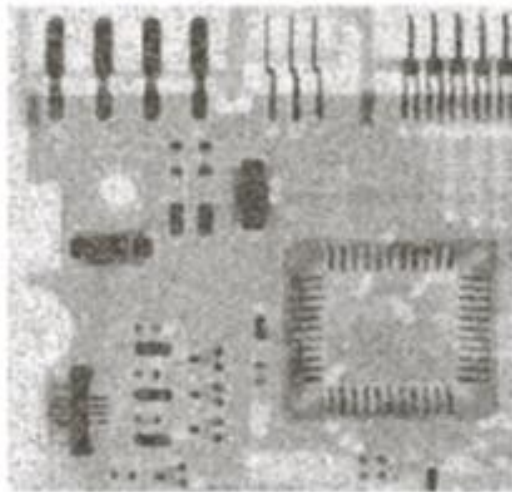
---

- ▶ Outros filtros não lineares:
  - ▶ Filtro de **ordem**: as intensidades dos pontos da vizinhança do pixel  $f(x,y)$ , dentro de uma janela da imagem, são ordenadas e é tomado o **valor de uma ordem qualquer** desta ordenação, como máximo, mínimo ou 3ª ordem, como novo valor para  $g(x,y)$ .
  - ▶ Filtro de **moda**: as intensidades dos pontos da vizinhança do pixel  $(x,y)$ , dentro de uma janela da imagem, são ordenadas e é tomado como novo valor para  $f(x,y)$ , o **valor  $g(x,y)$  mais frequente** da vizinhança.

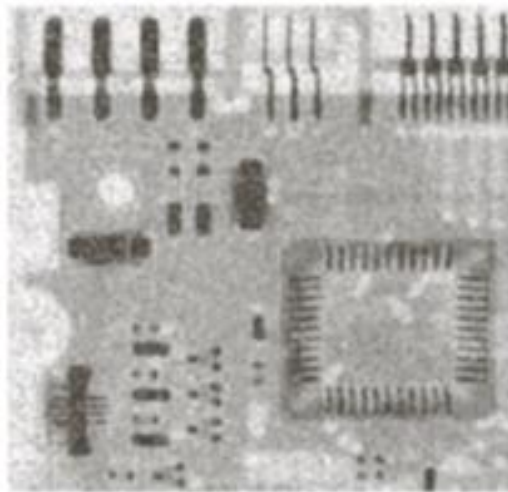
# Passa-baixa linear x não linear

---

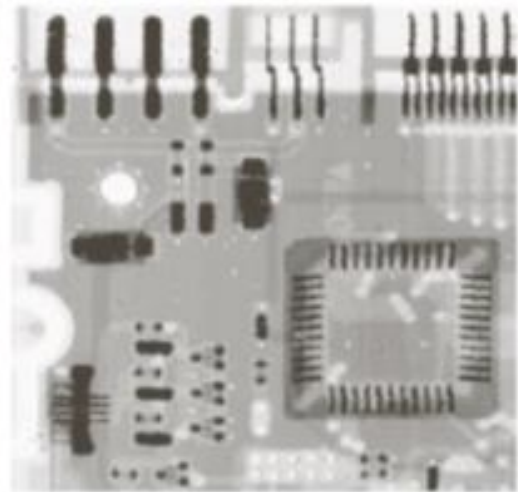
## ► Exemplos:



(1)



(2)



(3)

1. Imagem de raios-X de circuito impresso corrompido por ruído sal-e-pimenta.
2. Redução do ruído com uma máscara de média 3x3.
3. Redução do ruído com uma máscara de mediana 3x3.



# Filtros passa-alta

---

- ▶ Objetivos:
  - ▶ **Atenuam** ou **eliminam** as **baixas frequências**, realçando as altas frequências.
  - ▶ Usados para realçar os **detalhes** na imagem (intensificação, aguçamento ou “*sharpening*”).
  - ▶ **Destacam características** como bordas, linhas, curvas e manchas.
  - ▶ Tornam **mais nítidas as transições** entre regiões diferentes (como os contornos), realçando o contraste.
  - ▶ A máscara deve ter **pesos** cuja **soma** seja igual a **zero**.
- ▶ Filtros Laplaciano, máscara de nitidez e filtragem alto-reforço (*high-boost*).



# Filtro passa-alta: Laplaciano

---

- ▶ É um filtro linear e **isotrópico**
  - ▶ A resposta independe da direção das descontinuidades da imagem à qual o filtro é aplicado, ou seja, aplicar o filtro e rotacionar a imagem produz o mesmo resultado que rotacionar a imagem e aplicar o filtro.
- ▶ É um operador **diferencial**
  - ▶ **Realça** as **descontinuidades** de intensidade e **atenua** as regiões com níveis de intensidade de variação mais **suave**;
  - ▶ Produz uma imagem na qual as **descontinuidades** aparecerão em **níveis de cinza** e o **fundo** escuro e **uniforme**.
  - ▶ **Adicionando-se** a imagem laplaciana à original, **recupera-se** o fundo (cuidado com o sinal).

# Filtro passa-alta: Laplaciano

- ▶ Máscaras de filtragem usadas para implementar o laplaciano.

|   |    |   |    |    |    |
|---|----|---|----|----|----|
| 0 | 1  | 0 | 0  | -1 | 0  |
| 1 | -4 | 1 | -1 | 4  | -1 |
| 0 | 1  | 0 | 0  | -1 | 0  |

- ▶ Máscaras utilizadas para implementar uma extensão do laplaciano, incluindo as diagonais.

|   |    |   |    |    |    |
|---|----|---|----|----|----|
| 1 | 1  | 1 | -1 | -1 | -1 |
| 1 | -8 | 1 | -1 | 8  | -1 |
| 1 | 1  | 1 | -1 | -1 | -1 |

# Filtro passa-alta: Laplaciano

---

- ▶ É importante levar em consideração qual a definição de laplaciano que está sendo utilizada:
  - ▶ Máscara com **centro negativo**:  $c = -1$ ;
  - ▶ Máscara com **centro positivo**:  $c = 1$ .

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

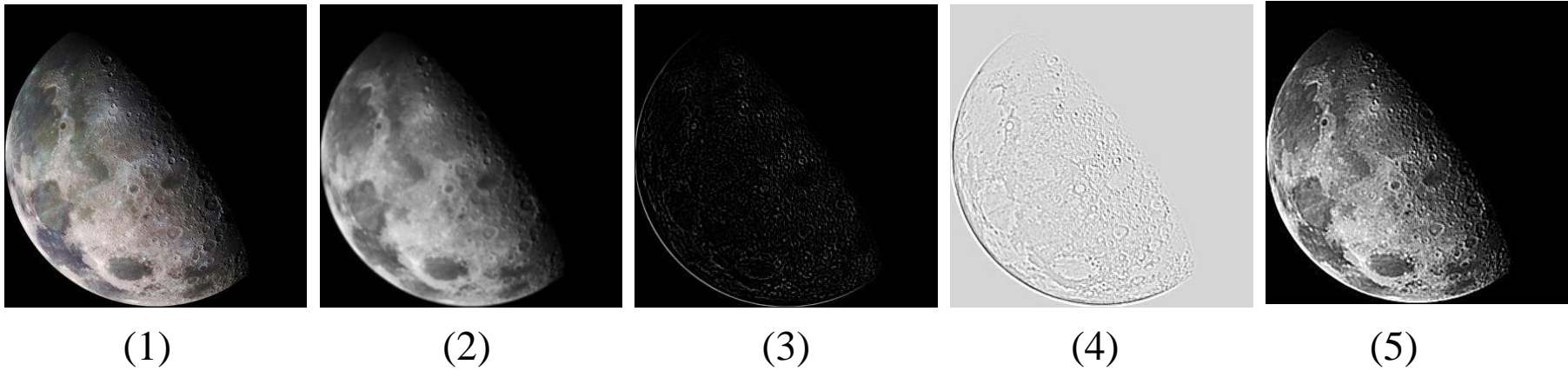
Onde  $f(x, y)$  e  $g(x, y)$  são as imagens de entrada e saída  $c$  é uma constante.

- ▶ Uma forma típica de **ajustar a escala de uma imagem laplaciana** é somar seu valor mínimo a ela para levar o novo mínimo a zero e ajustar o resultado para o intervalo total de intensidades  $[0 - 255]$ .

# Filtros passa-alta: Laplaciano

---

## ► Exemplos:



1. Imagem Original
2. Imagem suavizada (média)
3. Imagem laplaciana (filtro  $[-1 \ -1 \ -1; -1 \ 8 \ -1; -1 \ -1 \ -1]$ )
4. Imagem laplaciana ajustada
5. Imagem final

## Filtro passa-alta: máscara de nitidez e filtragem alto-reforço (*high-boost*)

---

- ▶ Consiste em subtrair uma versão não nítida (borrada, suavizada) de uma imagem da sua imagem original;
- ▶ Este processo é chamado de **máscara de nitidez** (*unsharp mask*) e consiste em:
  - ▶ **Borrar** a imagem original;
  - ▶ **Subtrair** a imagem borrada (suavizada) da original (o resultado é chamado de máscara);
  - ▶ **Somar** a máscara à imagem original.

# Filtro passa-alta: máscara de nitidez e filtragem alto-reforço (*high-boost*)

---

- ▶ Máscara de nitidez:

- ▶ Considerando a imagem borrada como  $\overline{f}(x, y)$

- ▶ A máscara de nitidez é expressa como:

$$g_{mask}(x, y) = f(x, y) - \overline{f}(x, y)$$

- ▶ Em seguida é adicionada uma porção ponderada da máscara à imagem original

$$g(x, y) = f(x, y) + k.g_{mask}(x, y)$$

- ▶ onde um peso  $k$  positivo foi incluído para generalização.

- ▶ Se  $k = 1$ , tem-se a máscara de nitidez definida anteriormente;

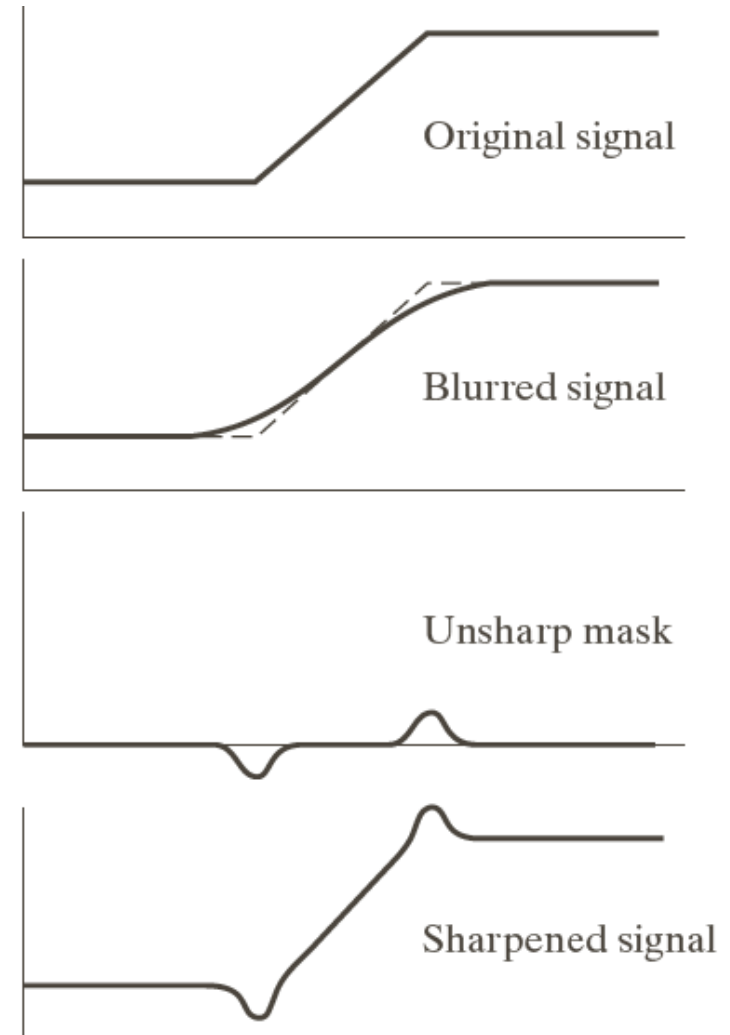
- ▶ Se  $k > 1$ , o processo é referido como filtragem *high-boost*;

- ▶ Se  $k < 1$ , atenua-se a contribuição da máscara.

# Filtro passa-alta: máscara de nitidez e filtragem alto-reforço (*high-boost*)

## ► Ilustração 1D do mecanismo de máscara de nitidez:

1. Sinal original
2. Sinal borrado com o original mostrado em pontilhado
3. Máscara de nitidez
4. Sinal realçado obtido somando 3 a 1.



# Filtro passa-alta: máscara de nitidez e filtragem alto-reforço (*high-boost*)

---

## ► Exemplo:

1. Imagem original
2. Imagem borrada
3. Máscara de nitidez
4. Resultado da aplicação da máscara de nitidez com  $k = 1$
5. Resultado do alto-reforço com  $k > 1$  ( $k=4.5$ )

(1)



(2)



(3)



(4)



(5)





# Filtro passa-alta

## ► Observações:



$$* \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$



Normalizado

$$* \frac{1}{9} \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$



# Filtros Derivativos

---

- ▶ Detectam **bordas** em todas as direções;
- ▶ São muito **sensíveis** (detectam) ruídos e pequenos detalhes.
- ▶ Filtros de **primeira derivada**: o operador de primeira ordem mais comum é o Gradiente.
  - ▶ A força da Borda (*edge strength*) é dada pela Magnitude do Gradiente.
- ▶ Exemplos de operadores de Gradiente
  - ▶ Prewitt, Sobel.

# Filtros Derivativos

## ► Operador de **Prewitt**

|    |    |    |
|----|----|----|
| -1 | -1 | -1 |
| 0  | 0  | 0  |
| 1  | 1  | 1  |

Bordas  
Horizontais

|    |    |    |
|----|----|----|
| 1  | 1  | 1  |
| 0  | 0  | 0  |
| -1 | -1 | -1 |

|   |    |    |
|---|----|----|
| 0 | -1 | -1 |
| 1 | 0  | -1 |
| 1 | 1  | 0  |

|    |    |   |
|----|----|---|
| 0  | 1  | 1 |
| -1 | 0  | 1 |
| -1 | -1 | 0 |

Bordas  
Diagonais

|   |   |    |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
| 1 | 0 | -1 |

Bordas  
Verticais

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

|   |    |    |
|---|----|----|
| 1 | 1  | 0  |
| 1 | 0  | -1 |
| 0 | -1 | -1 |

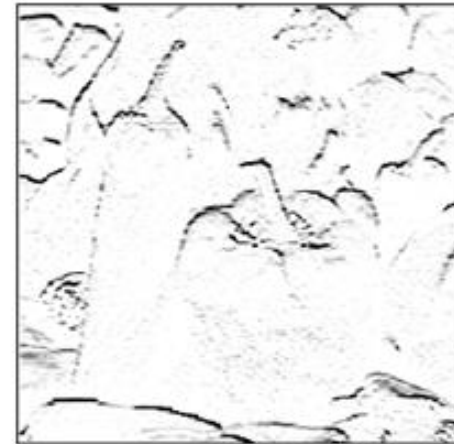
|    |    |   |
|----|----|---|
| -1 | -1 | 0 |
| -1 | 0  | 1 |
| 0  | 1  | 1 |

# Filtros Derivativos

## ► Operador de **Prewitt**



$$h_1 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



$$h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$



# Filtros Derivativos

## ► Operador de **Sobel**

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

Bordas  
Horizontais

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

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

Bordas  
Diagonais

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

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

Bordas  
Verticais

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

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

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

# Filtros Derivativos

## ► Operador de **Sobel**



$$h_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

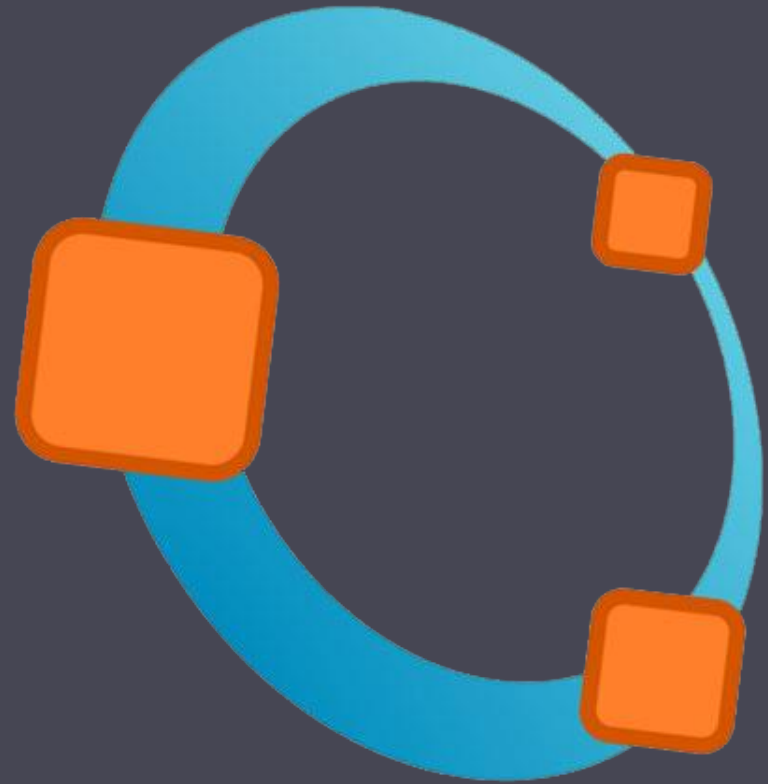


$$h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



# PRÁTICA EXTRA

- I. **Elabore o algoritmo para gerar as imagens do slide 40.**



Disponível no SIGAA