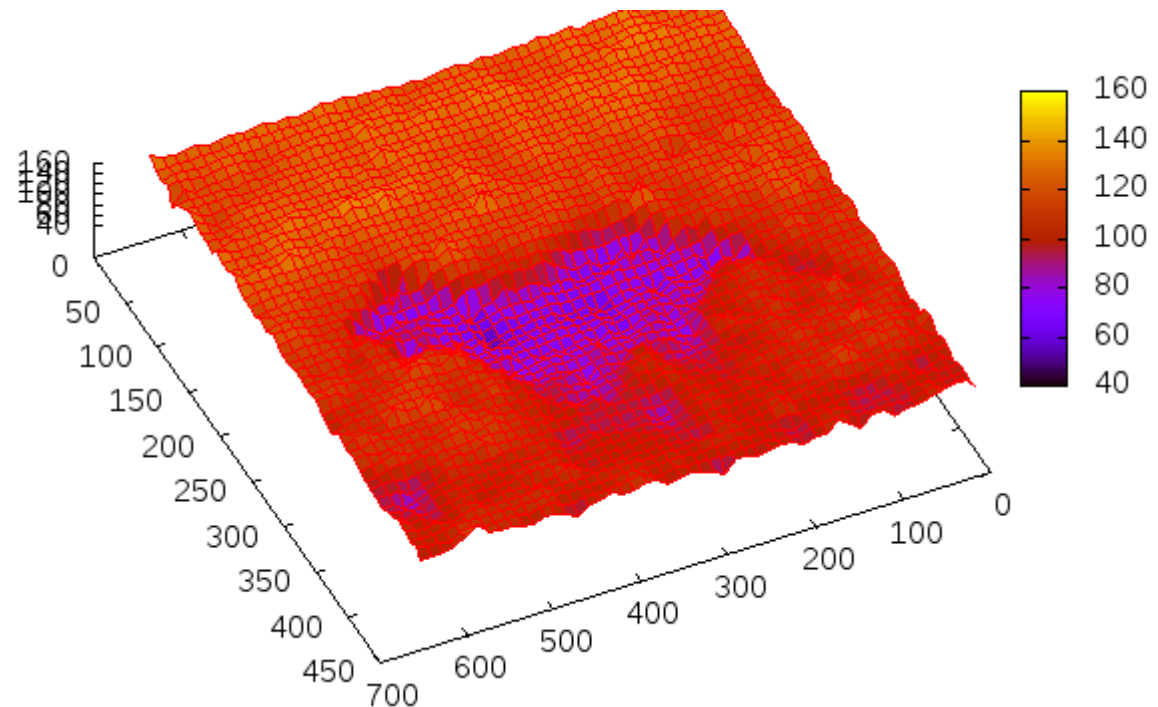
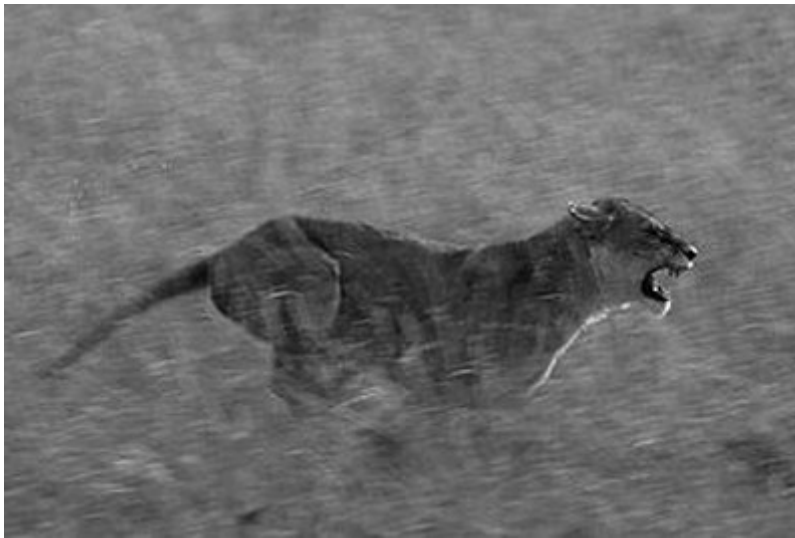


# Processamento digital de imagens

# Imagem digital: representação

- Armazenamento: matricial (mais comum)
- Função:  $f: X^2 \rightarrow Y_n$
- Podemos pensar em uma imagem digital também como a representação de uma função contínua



# Operações em imagens

- $g(x) = T(f(x,y))$
- Operadores pontuais  
atua em função de apenas um pixel da imagem
- Filtragem  
atua em função da vizinhança/região ao redor de um pixel

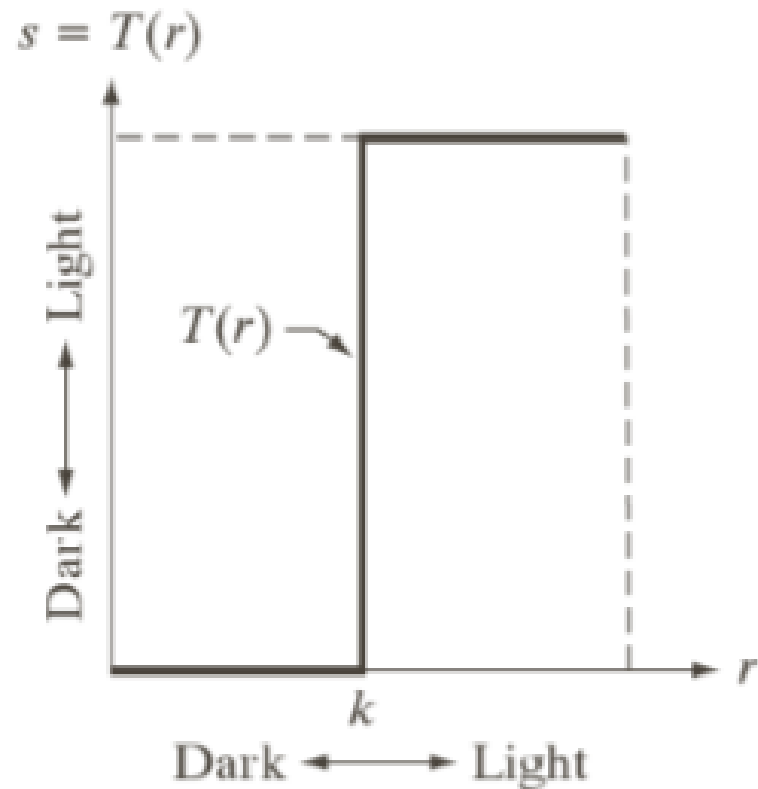
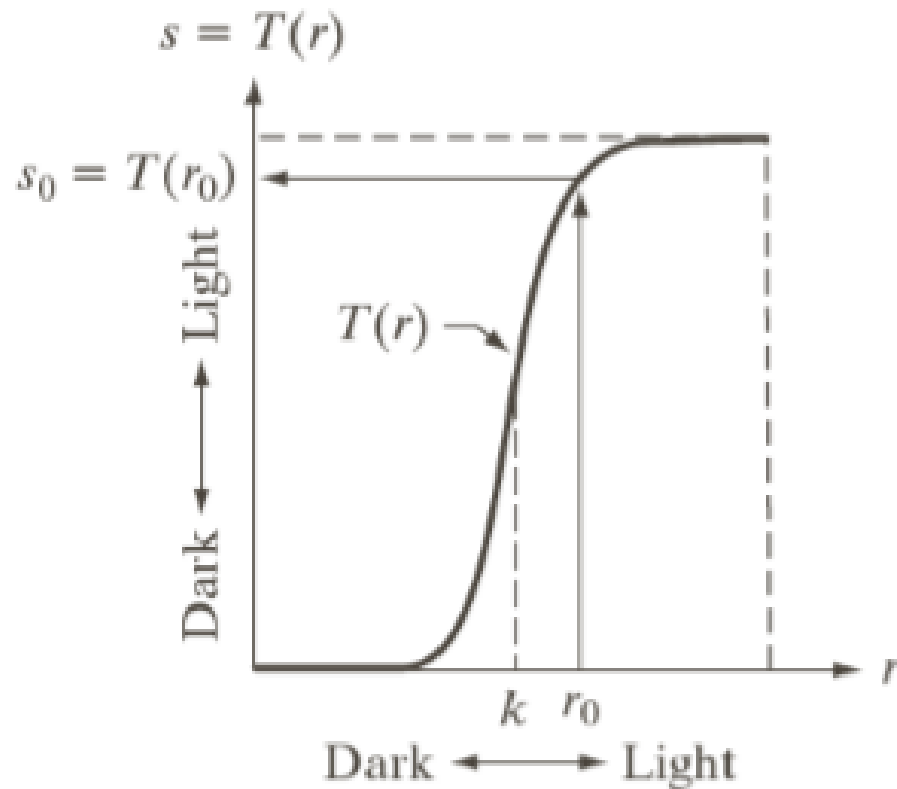
- As operações pontuais e os filtros são base para diversas aplicações:
  - restauração;
  - melhoria da imagem (tanto para processamento como para visualização);
  - extração de features;
  - atenuação de ruídos;
  - compressão de imagens;
  - etc



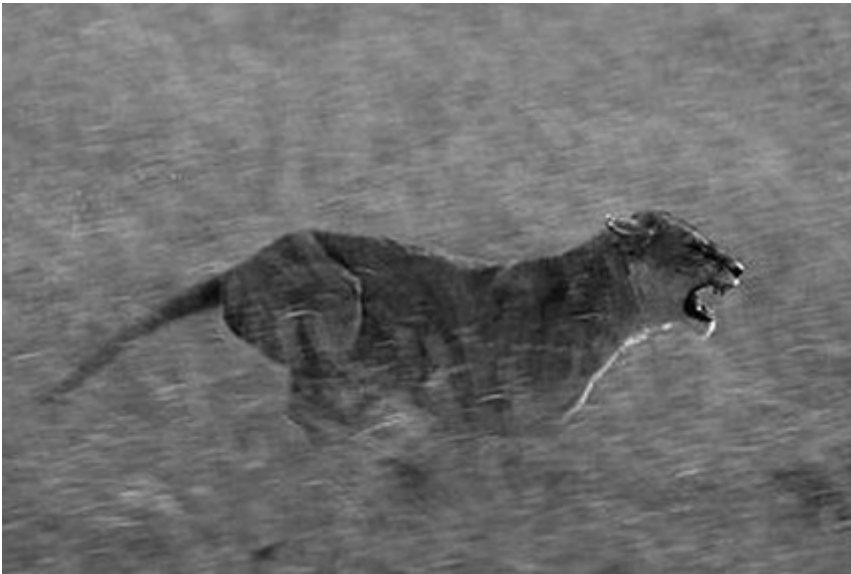
# Operadores pontuais

# Transformação de intensidade

- $s = T(r)$
- Exemplos (ajuste de contraste e limiar)

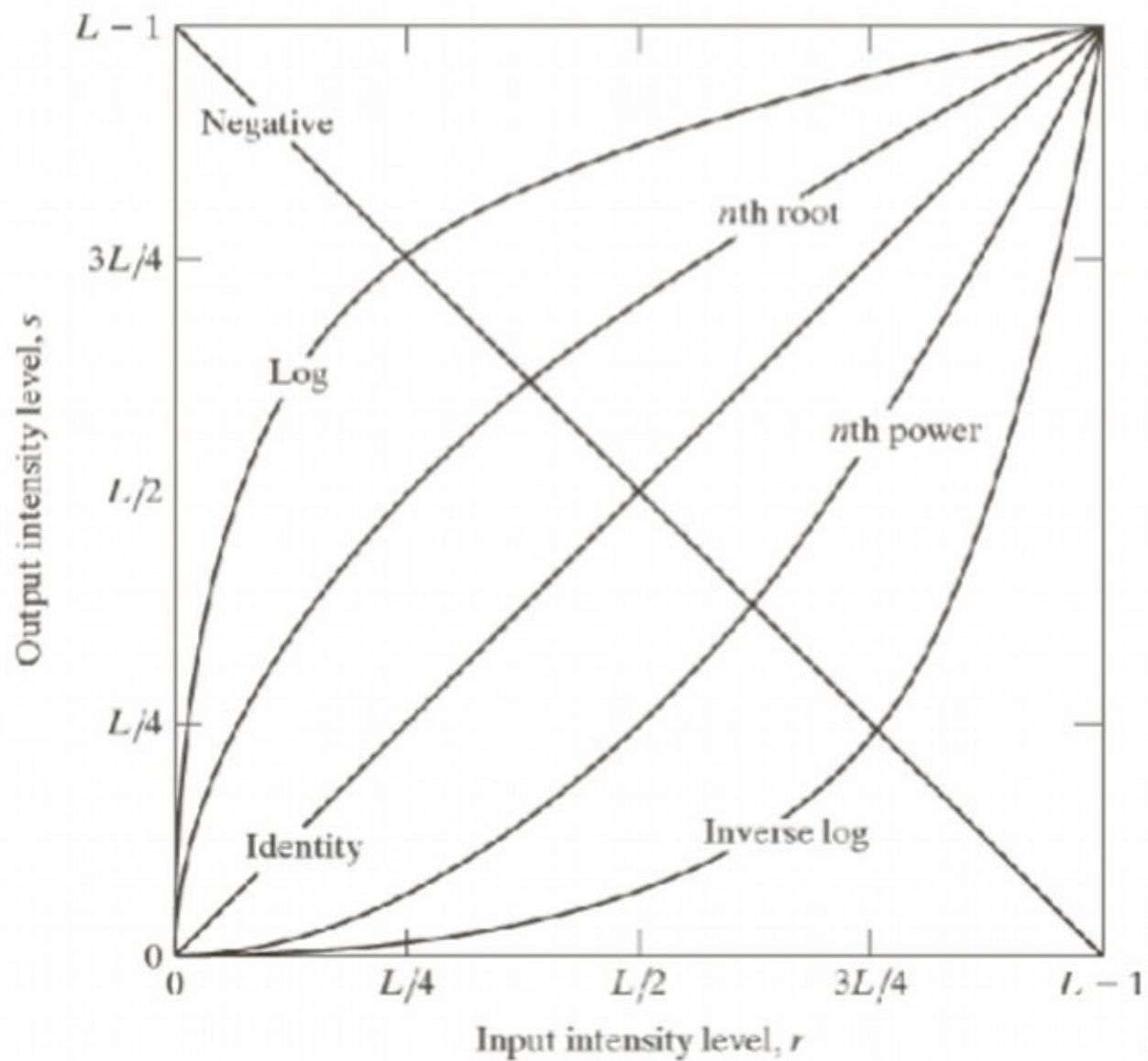


# Thresholding (limiar)



"Serengeti Lion Running saturated" by Taken by Schuyler Shepherd (Ununinium272). Contrast and saturation edited by norro - Canon 350D, 100-400mm f/4.5-5.6 L IS. Contrast and saturation edited by norro. Licensed under CC BY-SA 2.5 via Commons - [https://commons.wikimedia.org/wiki/File:Serengeti\\_Lion\\_Running\\_saturated.jpg#/media/File:Serengeti\\_Lion\\_Running\\_saturated.jpg](https://commons.wikimedia.org/wiki/File:Serengeti_Lion_Running_saturated.jpg#/media/File:Serengeti_Lion_Running_saturated.jpg)

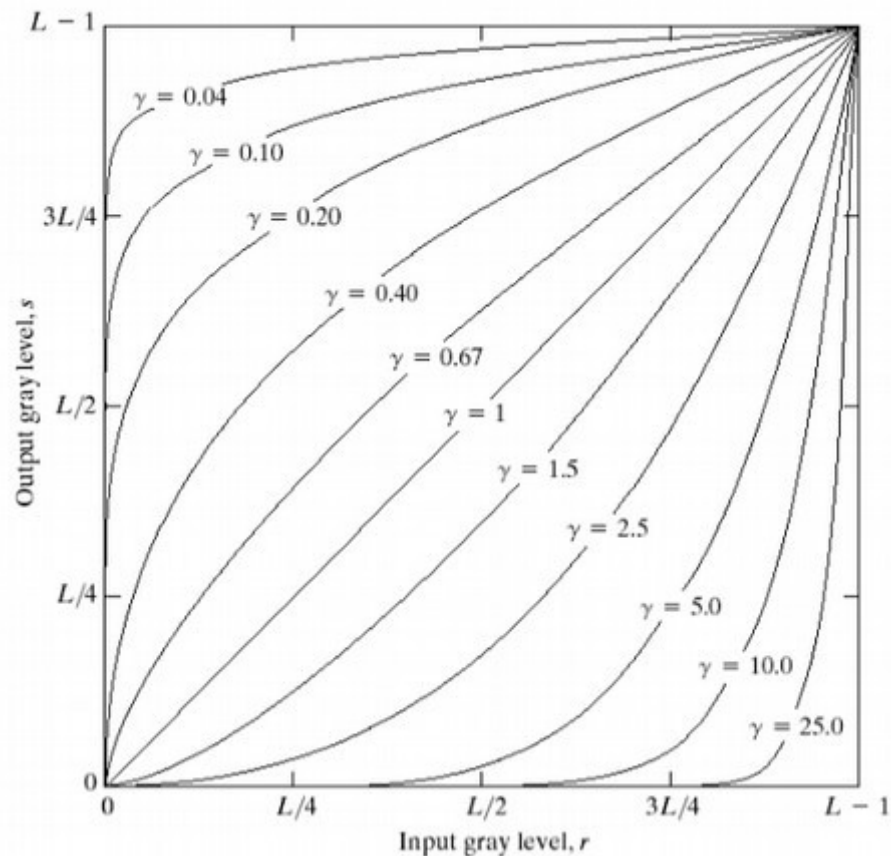
# Transformação logarítmica





# Correção gamma

- $s = cr^\gamma$
- `xgamma -gamma <gamma>` (no Linux)



Intensidade mínima

Intensidade máxima

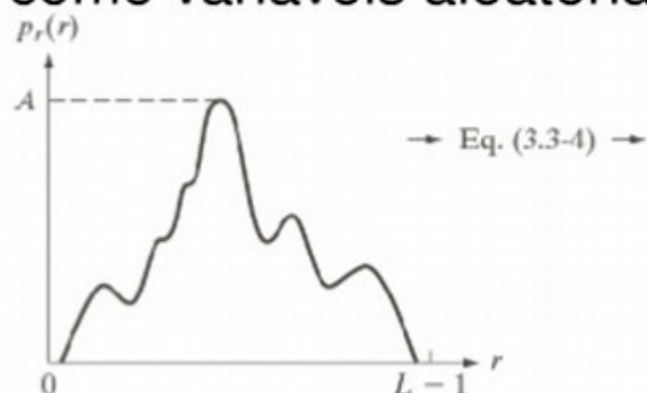
## Equalização de histogramas

É aplicada uma transformação de intensidade  $s = T(r)$ , onde  $0 \leq r \leq L - 1$ , tal que a função densidade probabilidade (PDF) de  $p_s(s)$  é (aproximadamente) constante.

Probabilidade da intensidade  $s$

# Equalização de histograma

Para obter a transformação de intensidade, considere  $p_s(s)$  e  $p_r(r)$  como variáveis aleatórias contínuas.



Probabilidade depois da equalização

Probabilidade antes da equalização

a b

**FIGURE 3.18** (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels,  $r$ . The resulting intensities,  $s$ , have a uniform PDF, independently of the form of the PDF of the  $r$ 's.

É possível provar que para obter uma distribuição contínua de  $p_s(s)$ , devemos aplicar a seguinte transformação de intensidade:

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw$$

# Equalização de histograma

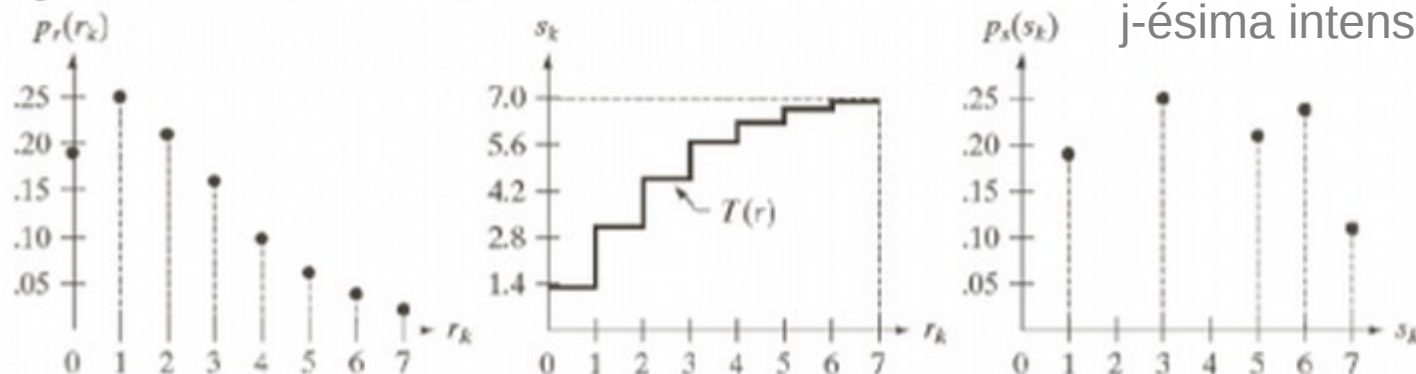
Para o caso de uma variável aleatória discreta (caso de imagens digitais), temos a seguinte transformação:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j$$

arredondado para o inteiro mais próximo, o que resulta em uma distribuição de probabilidade, na maioria das vezes,

**aproximadamente constante**

Número de ocorrências da j-ésima intensidade



a b c

**FIGURE 3.19** Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

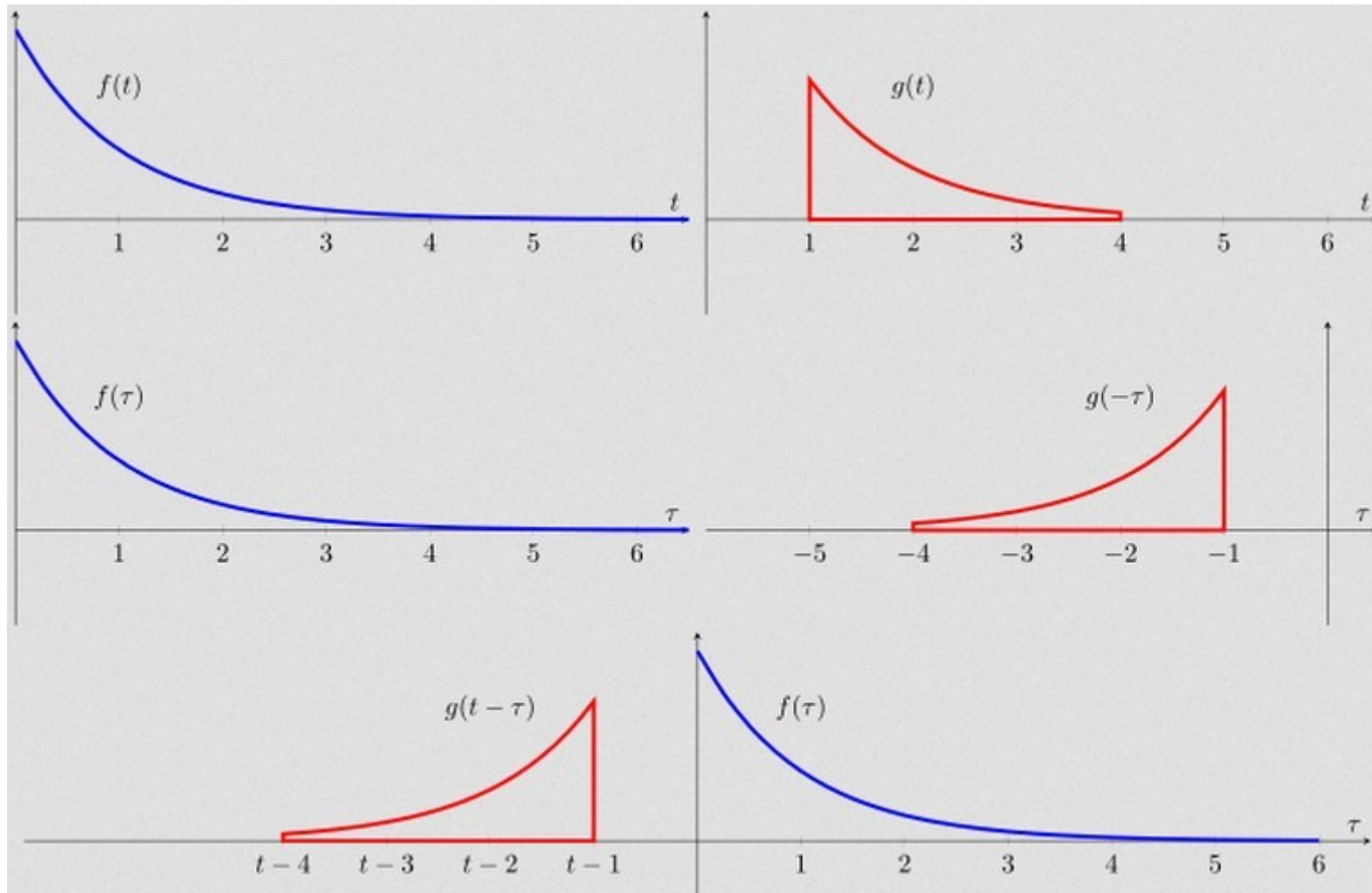


Filtragem

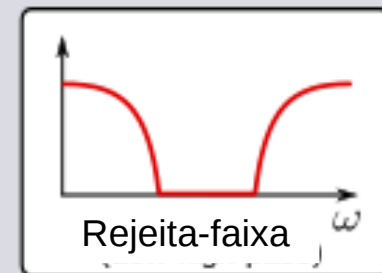
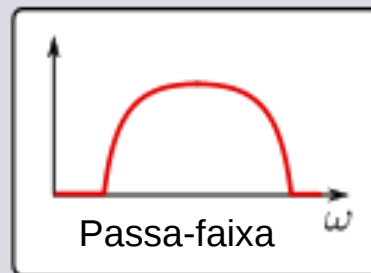
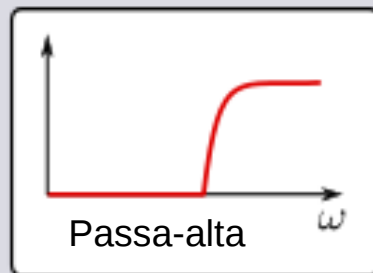
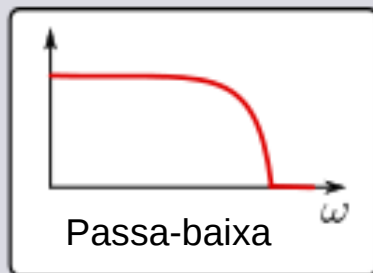
- Um filtro transforma um sinal de saída em um sinal de entrada
- Um filtro pode ser usado para reduzir ou destacar aspectos de um sinal
- Aplicação prática através de **convolução**

# Convolução

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$



# Filtragem: em termos de frequência

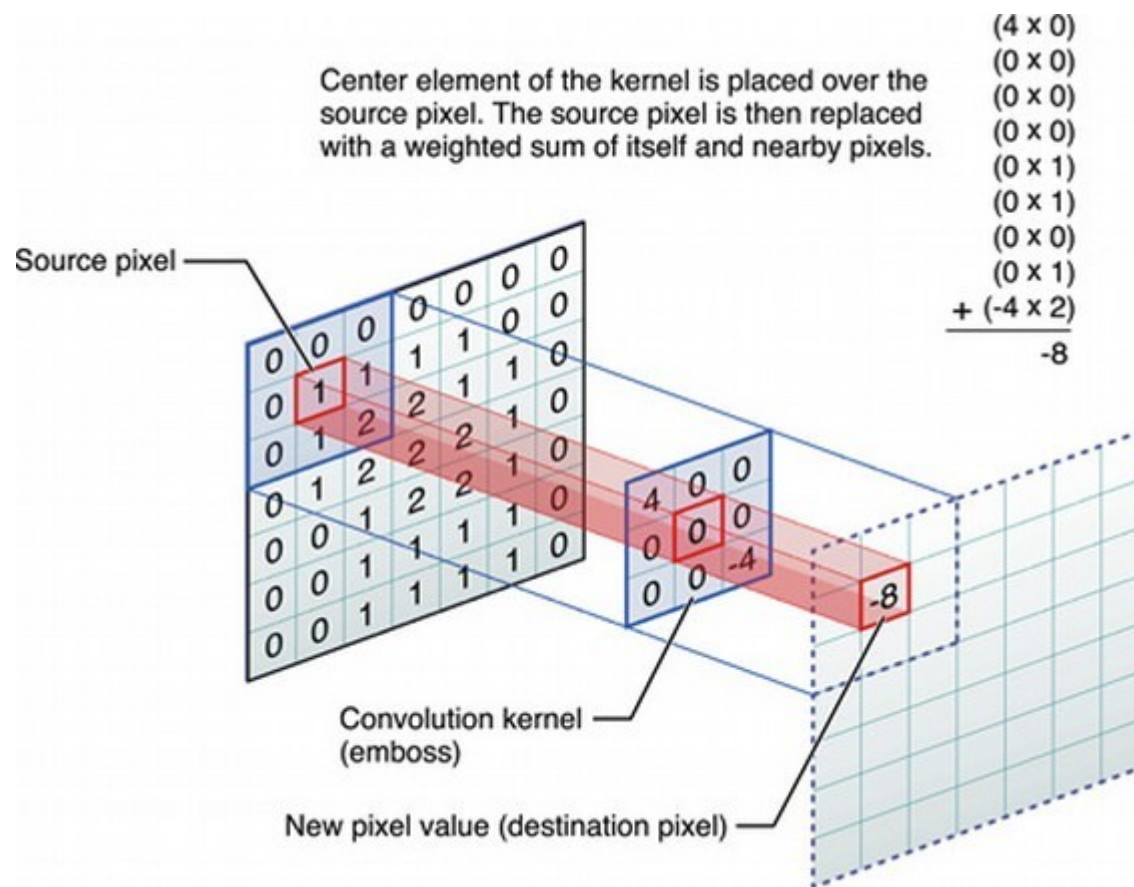


**Passa-baixa:** atenuam as altas frequências, então há um efeito de borramento (blurring)

**Passa-alta:** atenuam as baixas frequências, então há um efeito de detecção de bordas (edges)



# Filtragem: em termos espaciais



$$g(x, y) = \sum_{s=-m/2}^{m/2} \sum_{t=-n/2}^{n/2} w(s, t) f(x + s, y + t)$$
$$= w(x, y) \otimes f(x, y)$$

Imagem sobre convolução em:

[https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/Art/kernel\\_convolution.jpg](https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/Art/kernel_convolution.jpg)



Exemplo: filtro da média

## Problemas nas bordas

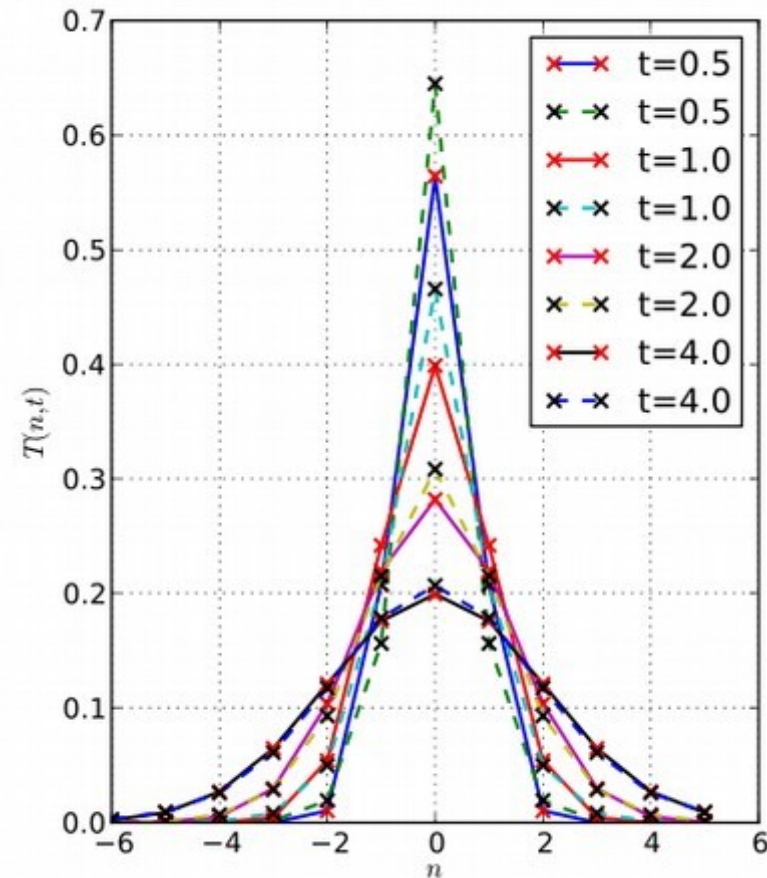
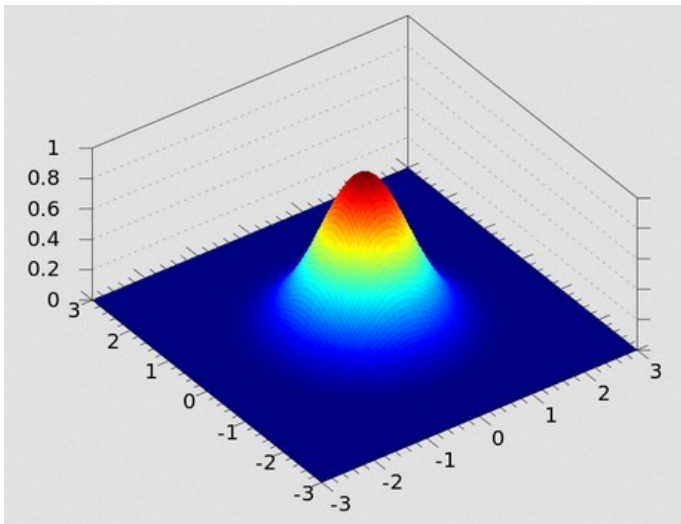
- Opções de solução: (1) começar no pixel onde a janela esteja nos limites (2) desconsiderar pontos da janela fora dos limites da imagem (3) wraparound



# Filtros de suavização

# Filtros de suavização

- Filtro da média
- Filtro da mediana
- Filtro gaussiano



Para discretizar, escolha um tamanho do kernel  
Distribua amostras igualmente espaçadas  
no formato de um grid

Divida todos pela menor das amostras  
e arredonde para o inteiro mais próximo

Multiplique a máscara por  $1/\text{somatório das amostras}$

$$\frac{1}{16}$$

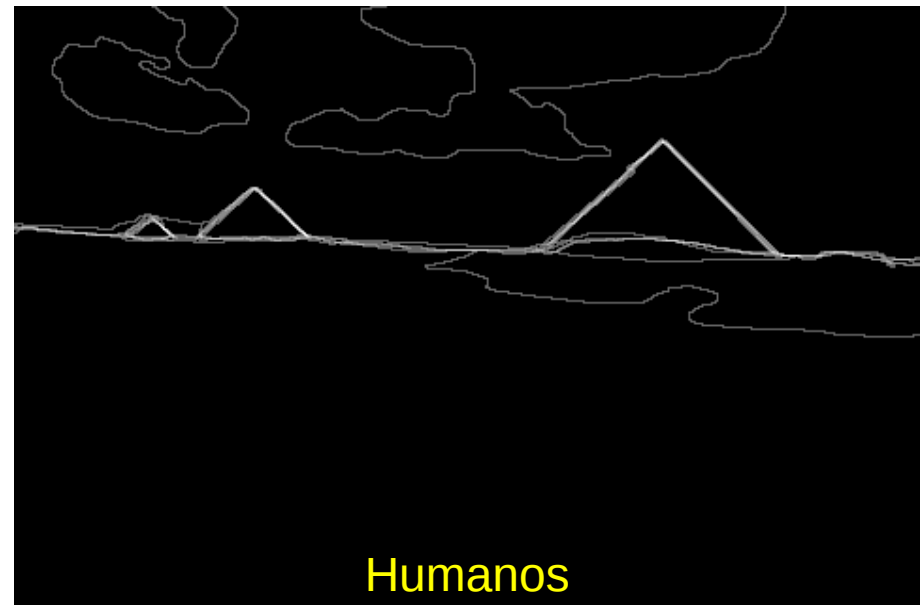
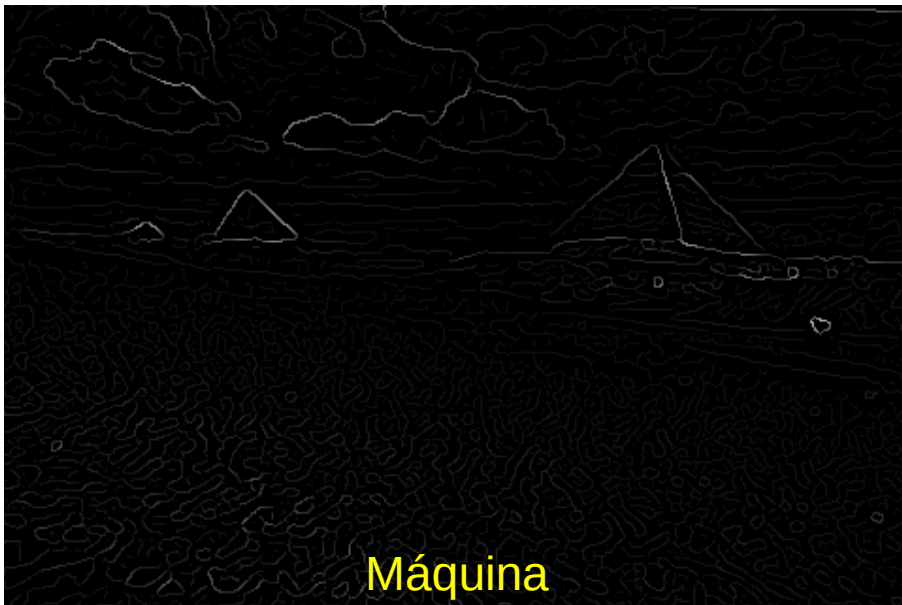
1	2	1
2	4	2
1	2	1

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$



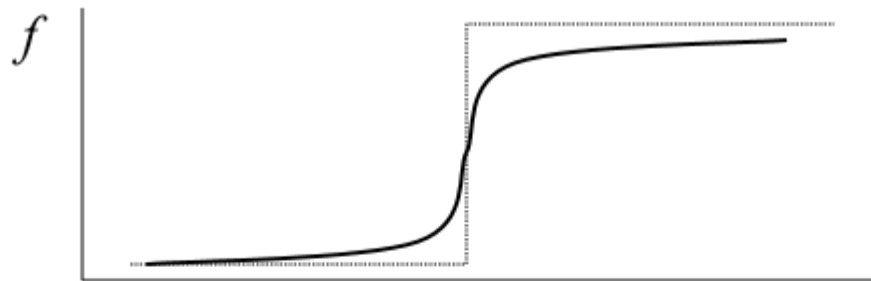
Filtros para detecção de bordas (edges)

- Detecção de mudanças abruptas na intensidade em decorrência de:
  - descontinuidades na profundidade
  - mudanças na orientação ou material da superfície
  - efeitos de iluminação
- Bordas (edges) x fronteiras (borders)
- Não é um problema bem posto (soluções não são únicas)



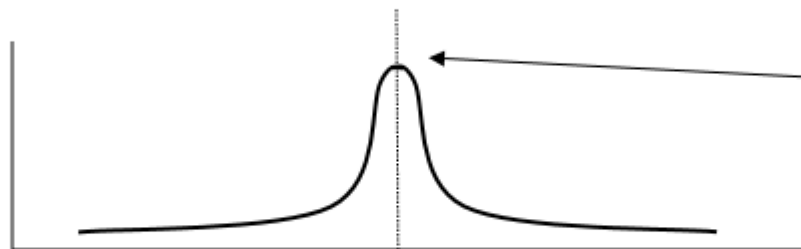
# Detecção pela derivada

Smoothed  
step edge



First  
derivative

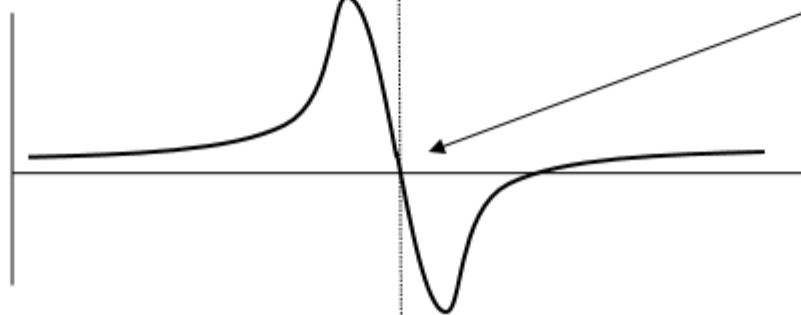
$$\frac{\partial f}{\partial x}$$



Peak magnitude at  
location of edge

Second  
derivative

$$\frac{\partial^2 f}{\partial x^2}$$



Zero crossing at  
location of edge



# Aprox. a derivada numericamente

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

-1	+1
----	----

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

+1	-2	+1
----	----	----


# Detector de bordas sobel e laplaciano

$$\frac{\partial}{\partial x} \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \frac{\partial}{\partial y} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

*Sobel operators*

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

*Laplacian operator*



## Orientação das bordas

Gradiente

$$\nabla \mathbf{f} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

Magnitude do gradiente

$$|\nabla \mathbf{f}| = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

Ângulo do gradiente

$$\theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

# Exemplo



# Detector de bordas Canny

- 1) Aplica um filtro gaussiano para suavização da imagem (redução de ruídos)
- 2) Encontre a magnitude do gradiente para cada pixel da imagem
- 3) Aplica *non-max suppression*
  - a) verifique os dois vizinhos na direção do gradiente (discretizado em 0, 45, 90, 135)
  - b) se a magnitude do gradiente do pixel atual não for maior que os dois vizinhos, o pixel é suprimido
- 4) Aplica *hysteresis* com *double thresholding* ( $t_1$  e  $t_2$ )
  - a) classifique cada pixel sobrevivente à etapa anterior como:
    - pixel de borda fraco:  $t_1 < |g| < t_2$
    - pixel de borda forte:  $|g| > t_2$
  - b) pixels onde  $|g| < t_1$  **ou** pixels de borda fracos desconectados de qualquer pixel de borda forte (busca por ex.: vizinhança de 8) são suprimidos